International Series in
Operations Research & Management Science

Katta G. Murty *Editor*

Case Studies in Operations Research

Applications of Optimal Decision Making

Springer

International Series in Operations Research & Management Science

Volume 212

Series Editor
Camille C. Price
Stephen F. Austin State University, Texas, USA

Associate Series Editor
Joe Zhu
Worcester Polytechnic Institute
Worcester, Massachusetts, USA

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

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

The book series *International Series in Operations Research and Management Science* encompasses the various areas of operations research and management science. Both theoretical and applied books are included. It describes current advances anywhere in the world that are at the cutting edge of the field. The series is aimed especially at researchers, doctoral students, and sophisticated practitioners.
The series features three types of books:

- *Advanced expository books* that extend and unify our understanding of particular areas.
- *Research monographs* that make substantial contributions to knowledge.
- *Handbooks* that define the new state of the art in particular areas. They will be entitled *Recent Advances in (name of the area)*. Each handbook will be edited by a leading authority in the area who will organize a team of experts on various aspects of the topic to write individual chapters. A handbook may emphasize expository surveys or completely new advances (either research or applications) or a combination of both.

The series emphasizes the following four areas:
Mathematical Programming: Including linear programming, integer programming, nonlinear programming, interior point methods, game theory, network optimization models, combinatorics, equilibrium programming, complementarity theory, multiobjective optimization, dynamic programming, stochastic programming, complexity theory, etc.
Applied Probability: Including queuing theory, simulation, renewal theory, Brownian motion and diffusion processes, decision analysis, Markov decision processes, reliability theory, forecasting, other stochastic processes motivated by applications, etc.
Production and Operations Management: Including inventory theory, production scheduling, capacity planning, facility location, supply chain management, distribution systems, materials requirements planning, just-in-time systems, flexible manufacturing systems, design of production lines, logistical planning, strategic issues, etc.
Applications of Operations Research and Management Science: Including telecommunications, health care, capital budgeting and finance, marketing, public policy, military operations research, service operations, transportation systems, etc.

Katta G. Murty
Editor

Case Studies in Operations Research

Applications of Optimal Decision Making

Springer

Editor
Katta G. Murty
Department Industrial and Operations Engineering
University of Michigan
Ann Arbor
Michigan
USA

The online version of this book (doi:10.1007/978-1-4939-1007-6) contains supplementary material, which is available to authorized users.

ISSN 0884-8289 ISSN 2214-7934 (electronic)
ISBN 978-1-4939-1006-9 ISBN 978-1-4939-1007-6 (eBook)
DOI 10.1007/978-1-4939-1007-6
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2014951678

© Springer Science+Business Media New York 2015
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

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

Foreword

History of Optimum Decison Making Techniques and Their Applications

During my High School education, I had a mathematics teacher who used to prove one theorem after another in his class. In that class I got the impression that his aim, and in fact the aim of mathematics as a subject, is to maximize the number of theorems proved. After seeing what felt like close to 100 theorems, I thought of the question, *what is the very first mathematical result realized by mankind?* The textbook did not discuss this aspect at all. So one day, I picked up enough courage to ask the teacher about it. He thought about it for a long time, and then said that he does not know the answer, but will discuss with other colleagues and let me know. My question went up all the way to the level of the Head Master, but none of the teachers knew the answer.

Thinking about that question now, I believe that the discovery of the first mathematical result realized by mankind goes back well over 100,000 years ago. In those days people used to live in caves, and they needed to fetch water from the nearby river for their living. So they faced the problem of finding the shortest route from the center of their cave entrance to the nearest point to it on the river bank (an instance of an unconstrained decision-making problem, as there are no constraints on the route taken), and they realized that this shortest route is the straight line joining the two points. This, I believe, is the first mathematical result realized by man.

In this way, I believe that the human urge to find optimum solutions to the decision-making problems that they faced in their daily living provided the motivation for the development of mathematics.

Solving a decision making problem optimally involves the following steps: (1) Construct an appropriate *mathematical model* for it, (2) then select an efficient *algorithm* for solving that model, and (3) finally implement the solution obtained by the algorithm.

History of Mathematical Modeling

For the first step above, both Chinese and Indians can feel proud that their ancestors pioneered the development of mathematical modeling over 2500 years ago. It involves representing the quantities that we want to determine by symbols—usually letters of the alphabet like *x, y, z* (the symbols representing the unknown quantities to be determined are nowadays called *unknowns*, or *variables*, or *decision variables*) then express the relationships between the quantities represented by these symbols in the form of equations, or other functional relationships. This process is called *modeling* or *mathematical modeling*. The earliest mathematical models constructed were *systems of linear equations*.

The Chinese text *Chiu-Chang Suanshu* (*Nine Chapters on the Mathematical Art* (a summary of this ancient Chinese text can be seen at the website: http://www-groups.dcs.st-and.ac.uk/history/HistTopics/Nine-chapters.html) composed over 2000 years ago describes the modeling process using a problem of determining the yield (measured in units called "tou") of an alcoholic drink made from rice. Rice grain produced by farmers contains three grades of grain: inferior, medium, and superior. Yield data from rice grain procured from three different farmers, Farmers 1, 2, and 3 are given. The composition of rice from these farmers (in terms of the percentage by weight of the three grades) and the yield from it is given in the following table:

Rice from farmer	Weight percent of grade			Yield of drink in tou/unit
	Inferior	Medium	Superior	
1	50	30	20	6
2	45	25	30	8
3	30	30	40	9

The problem considered is to determine the yield of the drink if it is made from pure inferior, medium, and superior grades of rice. Denote these quantities by the symbols:

$$\begin{aligned} x_1 &= \text{yield in tou from one unit of inferior grade rice} \\ x_2 &= \text{yield in tou from one unit of medium grade rice} \\ x_3 &= \text{yield in tou from one unit of superior grade rice} \end{aligned}$$

Then the mathematical model for determining the values of these variables is:

$$\begin{aligned} 50x_1 + 30x_2 + 20x_3 &= 6 \\ 45x_1 + 25x_2 + 30x_3 &= 8 \\ 30x_1 + 30x_2 + 40x_3 &= 9 \end{aligned}$$

Ancient Indian texts *Sulabha suutrah* (*Easy Solution Procedures*), and the *Bakshali Manuscript*, with origin during the same period describe the process in terms of models consisting of systems of two (three) linear equations in two (three) variables; for

information on these texts, and a summary see: http://www.tlca.com/adults/origin-math.html. One of the problems considered deals with a fruit seller in a farmer's market. She is selling the following bundles of fruit:

Bundle	Price
8 citrons + 5 mangoes	20
5 citrons + 4 mangoes	15

The problem is to determine how much she is charging for each individual citron, mango respectively. So, it is to detrmine the values of the following variables:

$$\begin{aligned} x_1 &= \text{ price in Rs. of an individual citron} \\ x_2 &= \text{ price in Rs. of an individual mango} \end{aligned}$$

Then the mathematical model for determining the values of these variables is:

$$\begin{aligned} 8x_1 + 5x_2 &= 20 \\ 5x_1 + 4x_2 &= 15 \end{aligned}$$

The Chinese and the Indians developed the *elimination method* for solving linear equation models, independently around that same period, more than 2000 years ago. This method was almost unknown in Europe until the German mathematician Gauss rediscovered it in the eighteenth century, hence it is now-a-days called the Gaussian elimination method.

History of the Word "Algorithm"

Methods for solving mathematical models are called *algorithms*. This word itself has a very strange connection to India. For many centuries, Muslim, Arabs, and Persians have invaded India. Many of these invaders settled down in India, but a few of them also returned to their home lands after some stay in India. One of those is a Persian, Muhammad ibn-Musa Alkhawarizmi, who left India after learning mathematics from Indian teachers in the ninth century CE. Then in 825 CE, he wrote a book *Kitab al-Jam'a wal-Tafreeq bil Hisab al-Hindi*, in Arabic, the lingua-franca in the Middle East in those days. This book appeared in a Latin translation under the title *Algoritmi de Numero Indorum*, meaning *Al-Khwarizmi Concerning the Hindu Art of Reckoning*; it was based on earlier Indian and Arabic treatises. Today this book survives only in its Latin translation, because all the copies of the original Arabic version have been lost or destroyed. The word *algorithm* (meaning procedures for solving algebraic systems) originated from the title of this Latin translation. Algorithms seem to have originated in the work of ancient Indian mathematicians on rules for solving linear and quadratic equations.

What Steps to Take Before Implementing an Algorithm in Practice

When an algorithm is developed for a decision-making problem, we need to make sure, either through a mathematical analysis of its performance, or through a computational test, that it outputs a guaranteed optimal or near optimal solution.

I will relate my experience once. I was talking with the DMs (Decision Makers) at a company on the benefits their company can gain by using optimal decision-making techniques for the decisions they make. They told me "Prof. Murty, for every decision we have to make, we use the following approach: we list all possible *actions* we can take for it, and estimate the profit we can get by taking each of those actions. Among all these actions, we select for implementation, one that is associated with the maximum expected profit. Because of this, we do not see how your optimal decision-making techniques can improve on our performance. If you like, you can try to convince us using an example."

I constructed the following example for them. It deals with a marketing company. The company divided their marketing area into six different zones, Z_1–Z_6, based on the background and other characteristics of people living there. For example the residents of zone Z_1 are administrators and other relatively well-off people. Zone Z_2 houses academic communities, etc. Then, they selected six experienced candidates, C_1–C_6 with different marketing backgrounds, and other experiences, etc.

Now they are faced with the very important decision-making problem of allocating one candidate as Director of Marketing Operations per zone. These are high level positions with the responsibility of maximizing the growth in company's profit.

Considering the background, experience, and other relevent characteristics of the candidates, and information obtained by market surveys, company's statisticians have come up with the following estimates of the profit to the company/year by the allocation of each candidate to (the Director of Marketing Operations position in) each zone.

	cij = Expected annual profit in \$million if candidate Ci is assigned to zone Zj					
Zone =	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
Candidate = $C1$	29	10	1	17	6	2
$C2$	36	31	26	32	28	27
$C3$	35	24	18	25	21	19
$C4$	30	11	3	20	8	4
$C5$	34	16	13	23	15	14
$C6$	33	12	5	22	9	7

Each candidate can be allocated to any of the six zones, so there are six possible actions associated with each candidate (allocating her/him to the position in zone $j = 1$ to 6). Since there are 6 candidates, there are $6 \times 6 = 36$ possible actions the company can take ((C_i, Z_j) = allocating C_i to the position in Z_j; $i, j = 1$ to 6). The above table gives the annual expected profit to the company associated with each of these 36 actions.

To apply the procedure described by the DMs at the company, we look for the action corresponding to the highest annual expected profit, it is (C_2, Z_1) yielding \$ 36 million in expected annual profit. After taking this action, we are left with candidates C_1, C_3 to C_6, and zones Z_2–Z_6. Among the actions available at this stage, the one corresponding to the maximum profit is (C_3, Z_4), leading to \$ 25 million in expected annual profits. Continuing the same way, leads to the solution of allocating candidates $C_2, C_3, C_5, C_6, C_4, C_1$ to the position in zones $Z_1, Z_4, Z_2, Z_5, Z_6, Z_3$ respectively, yielding total annual expected profit of $\$ 36 + 25 + 16 + 9 + 4 + 1 = 91$ million.

To check how good this solution is, we generate a random solution. It allocates candidates $C_1, C_2, C_3, C_4, C_5, C_6$ to the position in zones $Z_4, Z_3, Z_5, Z_6, Z_1, Z_2$ respectively; which yields an annual total expected profit of $\$ 17 + 26 + 21 + 4 + 34 + 12 =$ 114 million; higher than the annual expected profit corresponding to the solution produced by the company's procedure!

Actually, by developing a mathematical model for this problem, and solving it by an algorithm guaranteed to solve the problem exactly, it can be verified that the solution given by the company's procedure, instead of maximizing the total annual expected profit as required, in fact minimizes it!!!

The optimal solution maximizing the total annual expected profit allocates candidates $C_1, C_2, C_3, C_4, C_5, C_6$ to the position in zones $Z_1, Z_3, Z_6, Z_4, Z_5, Z_2$ respectively, leading to an annual expected profit of $\$ 29 + 26 + 19 + 20 + 15 + 12 =$ 121 million.

This shows that optimal decision-making problems are tricky, and procedures for solving them based on instinct and simple hunches may often lead to poor solutions. Hence, the importance of developing an appropriate mathematical model for them, and solving the model using an algorithm guaranteed to produce a good solution for implementation. In this book we discuss applications of this approach in a variety of areas.

IOE Dept., University of Michigan
Ann Arbor, MI-48109-2117, USA.
9 September 2012.

Katta Gopalakrishna Murty

Preface

In their OR (Operations Research) curricula, most universities place a lot of emphasis on teaching algorithms for solving a variety of standard mathematical models for solving optimum decision-making problems like linear programming, nonlinear programming, integer programming, etc. and the mathematical analysis of their convergence properties and computational complexity. But they do not put much emphasis on helping their students cultivate the essential skill of constructing an appropriate mathematical model for real world decision-making problems, before these algorithms can be applied to solve them.

Textbooks used in optimization courses do contain some simple modeling examples, but each of them is specially prepared to illustrate the application of a specific algorithm under discussion. But when students graduate and start working, they face the reverse situation, they face unstructured real world problems for which they have to construct appropriate mathematical models, and decide on which algorithms to use for solving them to get good solutions. From simple examples in their textbooks, and college courses, these students cannot learn how to model the unstructured real world problems that they will face. Since a fundamental goal of education is to train students to be able to comfortably handle the work they will face on their job after graduation, we have here a contradiction in present day practices.

One remedy for this situation is to provide students a textbook of a collection of real cases of applications of optimum decision-making techniques from a variety of application areas which can, by examples, help them develop the modeling skill, that students can also use for self-study and self-learning. This is the main purpose of this book.

Applications of OR and optimization techniques are usually quantitative and data intensive. Obtaining satisfactory solutions for them needs specialized data handling and data analysis strategies. Surprisingly, so far, OR departments have not yet developed for OR curricula collections of cases of data intensive quantitative work amenable to OR techniques for optimum decision making. I, and the contributing authors, hope that this book addresses this pressing need for such good and realistic cases

Style of the Book and Its Goals

Each chapter in the book will be a detailed case study of some challenging real world application of OR techniques, with all the data provided in a companion website that Springer has set up and maintains for this book. There will be two files: File 1 and 2 associated with each chapter. File 1 will only reside on the website for the book at Springer, with access given by Springer to all the readers of the book; it contains a brief description of the area of application, and of the problem, and the required outputs, and provides links to access all the data in the problem.

File 2 for the chapter is the book chapter. It contains a detailed description of the area of application (to make the readers familiar with the area), the scope for optimum decision-making techniques in the area, how to take advantage of any special structure in the data for the problem to construct a simpler mathematical model for it, different approaches for modeling and selecting algorithms for solving it, comparisons of the solutions obtained by these different approaches, and finally a recommendation on the best among them for implementation. If the problem is one that occurs periodically, the chapter describes how a DSS (Decision Support System) can be developed for solving it at the organization. As far as possible, the chapter should be in a style that students find engaging.

Methods for solving problems, considered in the book, will include not only ones like linear, nonlinear, integer programming with theoretically well established convergence criteria discussed in OR textbooks, but also simulation, heuristic methods like the greedy method with appropriately set up "greediness criterion;" which are not theoretically guaranteed to give the best solution, but have been verified to give good results in practice. In the book all methods including heuristics that give good solutions are considered.

So, the overall goals of this textbook are:

- To provide the readers with descriptions of the history and other background information on a variety of industries, and service or other organizations in which decision making is an important component of their daily operations. Also, in each of these organizations, give a list of possible applications of optimum decision-making techniques that can improve performance of these organizations.
  This information will provide very useful knowledge to students in their job interviews, and for them to settle in their jobs when they start working.
- To expose students to a variety of applications in a wide variety of areas.
  Since we clearly explain how each application is modeled and solved, and a variety of real applications are discussed, it helps the students develop the skill of modeling and solving the problems that they will face in their jobs.
- To teach students how to choose an appropriate model and algorithm to solve decision making problems.
- To teach students how to look for and take advantage of any special structure in the problem and the data for it, to simplify the model and algorithm used to solve it.

- To provide in each chapter some project exercises for students to practice what they have learnt in the chapter. These could be some variants of the problem discussed in the chapter. Faculty can also use these exercises as project assignments in their courses.

Brief Summaries of Chapters

Chapter 1 titled "Intelligent modeling essential to get good results" describes a project carried out at HIT (Hong Kong International Terminals) in the Port of Hong Kong by K. G. Murty in collaboration with colleagues from Hong Kong University of Science and Technology and HIT, to optimize decision making in daily operations at the terminal, with the aim of increasing the productivity of the terminal measured by the GCR (Gross Crane Rate, the number of lifts (moves of the shore cranes either downloading an incoming container from a vessel, or loading an outgoing container into a vessel))/shore crane hour. The only models discussed in published literature at that time are 0–1 models. They found that these models are inappropriate for the problem, since the data used in the model changes even before the software outputs the solution of the model. The second model tried for this problem is a multicommodity flow model which was easier to solve, but produced poor results. Finally a real time model using a substitute objective function technique gave excellent results, and the team received the Edelman finalist award of INFORMS for this work.

Chapter 2 on the Locomotive Fueling Problem (LFP) is a contest problem set up by RAS (Railway Applications Society) of INFORMS in 2010. It deals with a decision-making problem faced by a railroad company using diesel locomotives, to determine which yards should be selected as locomotive fueling yards, and how many refueling trucks should be hired at those yards to minimize fuel related operating costs of the company. The original contest problem can be seen at the website: http://www.informs.org/Community/RAS/Problem-Solving-Competition/2010-RAS-Competition.

Chapter 3 deals with the problem of organizing national elections in the Indian Republic, a societal application. Due to concerns about terrrorist disruptions at polling booths, there is a restriction that at each polling booth, 4 CPF (Central Police Force) personnel must be posted while polling is going on, and only 1.5 million CPF personnel are available for this. Hence polling over the whole country cannot be completed in a single day. The problem is to determine, subject to the specified constraints, the scheduling of different states of the country to different polling days, to minimize the number of polling days and the cost of moving CPF personnel from one set of states to the next over these polling days.

Chapter 4 deals with the decison making problems at a large Milk Producers Cooperative in India, the country that produces and consumes the maximum quantity of milk among all countries in the world.

Chapter 5 discusses an important application in the Health-care area. It deals with developing a DSS for use by a hospital to help new patients select for their care a PCP

(Primary Care Physician) from their PCP doctors, with the objective of equalizing the workload on these doctors; and for allocating appointment times to patients calling for service, with the aim of minimizing delays and waiting times.

Chapter 6 features decision-making problems faced by an NPO (Non-Profit Organization) distributing solar cooking stoves to poor families in tropical countries.

Chapter 7 is another societal application. It deals with the civil engineering problem of finding an optimum design for an earth dam across a river, to minimize the cost of constructing it, subject to safety constraints that the dam will not collapse during heavy flooding situations.

Chapter 8 features the optimum scheduling of a multiunit pumped strorage hydro power station during a short-term planning horizon, to maximize revenue at the prevailing market clearing price.

Chapter 9 deals with the problem of rebuilding the decaying water distribution infrastructure in municipalities to replace all the pipes using the same network topology to achieve present day pressure demands at junctions of the network at minimum cost.

Chapter 10 discusses an important application in the paper and paper board manufacturing industry. The major raw material in paper production is wood logs of a variety of species (conifers, casurina, eucalyptus, subabul, etc.). This chapter discusses the problem of making various decisions in the procurement of wood and the feeding of wood to the production line optimally.

Chapter 11 deals with the problem of finding an optimum design for the heat exchanger network to minimize the cost of hot and cold utilities used for bringing fluid streams in given sets of hot and cold streams to their target temperatures. This is a common problem in oil refineries and chemicals manufacturing industries.

Chapter 12 is a warehousing application. It deals with the problem of developing a DSS for a warehouse to store cuboidal boxes using storage space optimally.

Chapter 13 deals with the problem of determining the number of floating oil rigs to rent, route, and allocate for the drilling oil wells in the oil and gas fields in production in the North Sea to maximize the profit made by recovery and processing of oil/gas reserves.

Over the period of a 24-h day, the demand for electric power is higher during daylight hours than nighttime. The peak demand occurs during late afternoon; and its magnitude is nearly twice the demand level at 5:00 A.M., which is known as the base load level. Power companies usually have generators operating on cheaper fuel, producing power continuously throughout the day at base load level. It takes a long time to turn on these generators operating on cheaper fuel, that is why they are kept operating continuously once they are turned on. During daytime as demand for power increases (for airconditioning mostly, and other daytime uses), they turn on other generators running on more expensive fuels, to meet demand over the base load level, and switch these off in the evening as demand drops. If demand for power is at the same level throughout the day, meeting the demand would be much cheaper. The problem of meeting power demand over base load level economically is known as the "peak power problem." Chapter 14 discusses a strategy for solving this peak power problem to meet extra demand for power for airconditioning needs during hotter day time hours using thermal energy storage.

Chapter 15 on "Optimal flight planning and performance for a jet aircraft" discusses the optimum choice of pilot inputs: thrust level, elevator deflection, and bank angle by quantifying them, and determining how their values affect the flight conditions and its route; and then using this framework for determining optimal flight conditions.

While the freight railroad industry has been in existence for about 200 years, the procedure used for aggregating freight railcars into "blocks" based on their destination yards and other attributes, and then assembling blocks into trains has essentially remained the same. This "Train design problem" discussed in Chapter 16 is a very important and difficult combinatorial optimization problem encountered in the freight railroad industry. This case study is based on a simplified real-life problem set-up under the 2011 RAS Problem Solving Competition by RAS (Railway Applications Society) of INFORMS, see https://www.informs.org/Community/RAS/Problem-Solving-Competition/2011-RAS-Problem-Solving-Competition.

Chapter 17 discusses efficient tools for solving 2D cutting stock problems in the sheet-metal products industry, and 1D cutting stock problems in the paper industry.

Chapter 18 features the problem of managing inventories of blood products in a blood bank efficiently.

Chapter 19 discusses the application of optimization techniques for packing cells inside traction batteries used for operating forklifts in companies, to maximize their lives.

An "operating room" is a unit in a hospital where surgical procedures are performed, it is estimated that they generate over 42 % of hospital revenues. Chapter 20 discusses the important and complex problem of managing operating rooms in hospitals to minimize the idle time of both operating rooms and surgeons. The later part of this chapter on sequencing the various surgeries to be performed by a surgeon over the time of the day is based on the 5th AIMMS-MOPTA Optimization Modeling Competition 2013, see http://www.aimms.com.

Chapter 21 deals with the efficient scheduling of image acquisition, and image downlinking operations in satellite mission planning for earth observing satellites. It is a case study of mission planning operations of Canada's Earth Observing Synthetic Aperture Radar (SAR) satellite RADARSAT-2.

In some coastal regions, they have ferry service carrying people and automobiles between various ports in the region. Companies operating the ferrry service are often confronted with the problem of routing and scheduling the ferries to meet the travel demand between the various ports in the region during the planning horizon (typically from 5:00 A.M. to midninght in a day), while minimizing the cost of operations and any passenger dissatisfaction. Chapter 22 is a case study of such a ferry scheduling problem.

File 1s for the Various Chapters

For each chapter we have a File 1. It briefly explains the setting and the statement of the optimum decision-making problem discussed in the chapter, either provides all the data for the problem, or gives a link from where it can be accessed, and clearly describes the desired outputs.

For each chapter, File 1 is prepared so that instructors can modify some data elements in the problem, and then use it as a project exercise in their courses. File 1s for the various chapters can be accessed by users of the book on the book's website at Springer.com; the link to which is: http://www.springer.com/business+%26+management/operations+research/book/978-1-4939-1006-9

A Final Caution

Clearly, making decisions optimally brings competitive advantages to the performace of organizations. But it is so easy to introduce an inappropriate or wrong model, and wrong equations into the model. To illustrate this, I mention the following puzzle:

" Three Ladies went to a hotel to share a room. The clerk asked for $ 300. Each Lady paid $ 100, making up the $ 300.

The hotel keeper then decided to allow a discount for the day, charging only $ 250 for the room. He told the clerk to return $ 50 to the three Ladies. The clerk pocketed $ 20 for himself. He gave the remaining $ 30 back to the three Ladies. Each Lady took back $ 10. Therefore, each Lady paid $100 - 10 = \$ 90$ to the hotel.

$\$ 90 \times 3 = \$ 270 +$ the clerk's $\$ 20 = \$ 290$. Question: Where has the remaining $ 10 gone?"

Acknowledgments

My thanks to all the contributing authors; and my editors and supporters, Fred Hillier, Camille Price, and the Springer team for constant encouragement. Many of the figures in this book are taken from Wikipedia articles, Google.com articles, and other public web sources, and we gratefully acknowledge the owners. Finally I thank my wife Vijaya Katta for being a great companion.

murty@umich.edu
Katta Gopalakrisna Murty
9 September 2012

Contents

Contributors

Adel Alaeddini University of Texas at San Antonio, San Antonio, TX, USA

University of Michigan, Ann Arbor, MI, USA

Gemma Berenguer Purdue University, Lafayette, IN, USA

Alberto Borghetti DEI, University of Bologna, Bologna, Italy

Cristiana Bragalli DISTART, University of Bologna, Bologna, Italy

Weidong Chen Department of Industrial and Operation Engineering, University of Michigan, Ann Arbor, MI, USA

Wesley Cole McKetta Department of Chemical Engineering, University of Texas, Austin, TX, USA

Claudia D'Ambrosio CNRS LIX, École Polytechnique, Palaiseau, France

Majid M. Al-Gwaiz Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI, USA

Mary Goulet Duck University of Michigan Hospitals and Health Centers, Ann Arbor, MI, USA

Thomas Edgar McKetta Department of Chemical Engineering, University of Texas, Austin, TX, USA

Dag Haugland Department of Informatics, University of Bergen, Norway

Daniel Karapetyan ASAP Research Group, School of Computer Science, University of Nottingham, Nottingham, UK

Kriti Kapoor McKetta Department of Chemical Engineering, University of Texas, Austin, TX, USA

JongSuk Kim McKetta Department of Chemical Engineering, University of Texas, Austin, TX, USA

Jon Lee IOE Department, University of Michigan, Ann Arbor, MI, USA

Taeyoung Lee George Washington University, Washington, DC, USA

Harshal Lowalekar Operations Management and Quantitative Techniques Area, Indian Institute of Management Indore, Indore, Madhya Pradesh, India

Andrea Lodi DEI, University of Bologna, Bologna, Italy

Krishna T. Malladi Department of Mathematics, Simon Fraser University, Surrey, BC, Canada

Silvano Martello DEI, University of Bologna, Bologna, Italy

Harris McClamroch University of Michigan, Ann Arbor, MI, USA

Snezana Mitrovic-Minic MDA Systems Ltd., Richmond, BC, Canada

Katta G. Murty Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA

IOE Department, University of Michigan, Ann Arbor, MI, USA

SQC & OR Unit, Indian Statistical Institute, Hyderabad, India

G. S. R. Murthy SQC & OR Unit, Indian Statistical Institute, Hyderabad, India

A. L. N. Murthy SQC & OR Unit, Indian Statistical Institute, Hyderabad, India

Bodhibrata Nag Indian Institute of Management, Kolkata, India

Operations Management Group, Indian Institute of Management Calcutta, Calcutta,West Bengal, India

Omkar D. Palsule-Desai Indian Institute of Management, Indore, India

Abraham P. Punnen Department of Mathematics, Simon Fraser University, Surrey, BC, Canada

G. Raghupathy HES Infra Pvt Limited, Hyderabad, India

Amber Richter University of California, Berkeley, CA, USA

N. Ravichandran Production and Quantitative Methods, Indian Institute of Management Ahmedabad, Ahmedabad, Gujarat, India

Zuo-Jun (Max) Shen University of California, Berkeley, CA, USA

Paolo Toth DEI, University of Bologna, Bologna, Italy

Bjørn Peter Tjøstheim Statoil ASA, Postboks, Bergen, Norway

Chapter 1
Intelligent Modeling Essential to Get Good Results: Container Storage Inside a Container Terminal

Katta G. Murty

1 Introduction

The first steps in solving a decision making problem optimally are to construct an appropriate mathematical model for it, and then select an algorithm for solving that model. We will illustrate this with work on a project carried out at HIT (Hong Kong International Terminals) in Hong Kong Port [3] which has won the Edelman Finalist Award from CPMS (College for the Practice of Management Science) of INFORMS (Institute for Operations Research and Management Science) in 2004. But first we provide an account of how I became interested in "optimization."

2 My Introduction to Optimization

My first encounter with optimization was in 1960 in a talk given by a visiting US Professor at the Indian Statistical Institute, Kolkata, where I was a graduate student at that time. The American Professor talked about a newly evolving subject called *OR (Operations Research)* unheard of in India at that time. He talked about "algorithms" (a new term for me at that time), and a problem called the *TSP (traveling salesman problem)*, for which he said no one has been able to develop a reasonable algorithm until that time. That intrigued me so much that I started thinking about the TSP, and in about a year, while on a visit supported by a Fulbright grant to Case Institute of Technology (now called the Case Western Reserve University), I developed the Branch and Bound method for it [1].

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_1) contains supplementary material, which is available to authorized users.

K. G. Murty (✉)
Department of Industrial and Operations Engineering, University of Michigan,
Ann Arbor, MI 48109-2117, USA
e-mail: murty@umich.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_1, © Springer Science+Business Media New York 2015

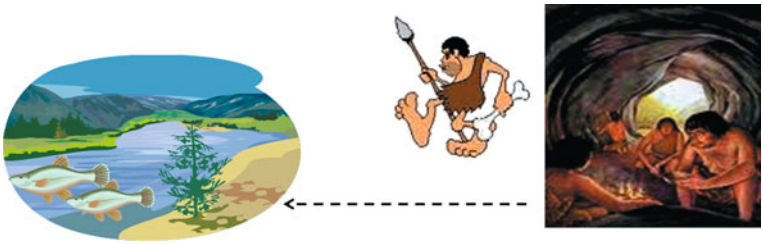


Fig. 1.1 Find shortest path from cave home to river: An unconstrained minimization problem

This work has helped to shift my research focus from statistics to optimization, and in 1965, I joined the University of California, Berkeley (UCB) as a graduate student for my Ph.D. When I met my advisor there, David Gale, for the first time, he asked me "Mr. Murty, what do you want to work on"? I told him "Prof. Gale, I want to work on optimization." Thinking about this today, I am so happy that I replied in this way at that time, as optimization has opened many opportunities for me. I also want to encourage young scholars planning their careers to seriously consider optimization.

I believe that the development of mathematics, and in fact of all sciences, has its roots in the human desire to optimize. So, before getting into the main topic, I will briefly survey the history of the development of optimization from ancient times (Fig. 1.1).

100,000 Years Ago . . . In those days, people used to live in caves. They need water from the river, so they faced the problem of going from their cave to the river for water. Seeing the nearest point on the river bank to their cave, they discovered the fundamental result that the shortest route from the center of their cave's entrance to that point on the river bank is the straight line joining them. This is perhaps the earliest mathematical result discovered by mankind; solution to an *unconstrained minimization problem* (as there are no constraints on the route under consideration).

Several Thousand Years Later . . . Things changed slowly in those days. Some thousands of years later, a family of tigers moved in and occupied a region in the middle of their shortest route from their cave to the river. Now they faced the new problem of finding a shortest route from their cave to the river avoiding the zone occupied by the tigers, an instance of a *constrained optimization problem* (Fig. 1.2).

Now . . . We have seen above the two types of optimization problems encountered in real-world decision-making. Now, OR is the branch of Science dealing with optimum decisions for situations that we face in real-world operations. The main strategy for solving these problems involves the following steps:

1. Construct a mathematical model for the decision problem (which involves making a list of all the relevant decision variables that play a role in the problem, and whose values need to be determined optimally; identifying the bounds and constraints on them from the way the system operates; and a list of all the objective functions that need to be optimized).
2. Solving the model using an efficient algorithm to find the optimum solution(s).
3. Making necessary changes, and implement the final solution obtained.

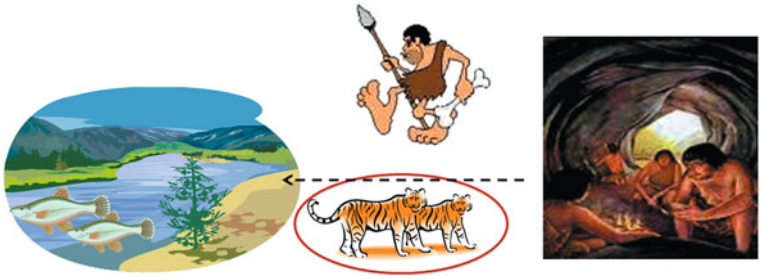


Fig. 1.2 Find shortest path from cave home to river, avoiding the area occupied by tigers! A constrained minimization problem

By now, OR Theory has developed efficient algorithms for solving several single objective decision-making models. But in many real-world applications, practitioners find that none of these well-solved models fit their application exactly; because real-world problems are usually multiobjective, and lack the nice structure of models discussed in theory. So, there is a big *gap between theory and practice*, and we need a bridge to cross this gap. In order to get good results in applications, it is essential to model the problems intelligently using heuristic modifications, approximations, substitute objective functions, relaxations, and hierarchical decomposition—these provide the bridge to cover the gap between theory and practice. We will illustrate this point using a project carried out at HIT in Hong Kong Port, and later at PSA (Port of Singapore Authority), two of the largest container terminals in the world [4, 5, 6] .

3 Brief Introduction to Container Shipping

Maritime shipping is broadly divided into two categories: one is shipping of bulk commodities like crude oil, iron ore, coal, etc. in specialized vessels, and the other called *container shipping* handling all other manufactured goods like textiles, etc.

In container shipping, customers pack all the goods that they want to send to overseas destinations in steel boxes called "containers" of which there are three different sizes in use today: 20-foot long containers called *TEUs (twenty-foot equivalent units)* common in old times, 40-foot long containers called FEU (forty-foot equivalent units, counted as 2 TEUs) which are becoming common today, and "refers" (refrigerated containers that must be held at cold temperatures, commonly 45-foot long). The customer then takes these containers (called *outbound* or *export containers (ECs)*) to the container terminal in his own truck (referred to as *ET (external truck)*) which enter the terminal through the *TG (terminal gate)*. The gate operator tells the driver where to take it in the *SY (storage yard)*. At the SY, a *YC (yard crane)* removes the container from the truck, and stores it, where it will remain until the vessel into which it is to be loaded arrives and docks at a berth in the terminal. At

Fig. 1.3 A docked container vessel and quay cranes (*QCs*) unloading inbound containers (*ICs*) from it and loading outbound containers into it. Also see an internal truck (*IT*) (that transports containers between the berth and the storage yard (*SY*)) loaded with an IC that it is taking to the SY

that time, the YC retrieves the container from its storage position and puts it on one of the terminal's trucks (these are owned by the terminal, hence called *ITs (internal trucks)* and used by them to shuttle containers between the quay and the SY) which takes it to the vessel on the berth. This IT with that container joins a queue under a *QC (quay* or *shore crane*) loading/unloading containers into/out of the vessel. When this truck moves to the front of the queue, the QC removes the container from it and loads it into the vessel. The truck that carried it then returns to the SY to bring another container to be loaded.

The handling of inbound containers or *ICs (import containers)* that are sent to this port by overseas customers follows a reverse sequence to that of ECs, they arrive on vessels, unloaded on berth by QCs, and loaded onto ITs which take them to be stored in the SY until its owner sends his ET to the terminal to pick it up, then it is retrieved from storage by a YC, transferred to the ET which leaves the terminal with it through the TG (Fig. 1.3).

SY is the section of the terminal used for the temporary storage of containers between land and sea transportation. Depending on the terminal's mode of operation, containers may be stored either on the trailer on which they arrived, or stacked one on top of the other. The SY is divided into rectangular regions called *storage blocks*. At HIT, the width line of each block is divided into seven lanes, six of them are storage lanes and the seventh is the truck passing lane for container trucks (ET or IT) to bring containers for storage in the block, or to take away containers stored in this block. Each storage lane is divided into 26 storage spaces, each for storing one stack of containers one on top of the other up to six containers high. Each block is allotted two or more YCs for storing arriving containers and for retrieving containers from storage in it (Fig. 1.5).

4 Key Service Quality Metrics

There are only about 40 major shipping lines in the world, and they all take their decision on which container terminal to patronize very seriously. Once a shipping line decides to stop using a terminal's services, there is very little that the terminal

Fig. 1.4 Containers in temporary storage at the storage yard (*SY*). At water's edge, see quay cranes (*QCs*) unloading import containers (*ICs*) from, and loading export containers (*ECs*) into, vessels. In SYs see yard cranes (*YCs*) that store and retrieve containers from storage

can do to win them back. Regional competition among terminal operators is fierce. Every day terminal managers rack their brains to improve service level to surpass their competitors, as efficiency is a matter of business survival (Fig. 1.4).

Each vessel has a lot of employees working on it to keep it sailing smoothly. When a vessel docks on its berth, the terminal begins *processing it*, i.e., unload import containers off it and load export containers onto it; once this processing is completed, the vessel leaves the port on its next voyage. The amount of time taken for this processing is called the *vessel turnaround time*. While the vessel is docked on the berth, its employees have no work, but the shipping line has to pay their wages. That is why shipping lines are very anxious to keep these vessel processing times at terminals as low as possible. So the *average vessel turnaround time* at a terminal is one of the most important service quality metrics used in the industry to rate terminals—at top rated terminals this varies from 8 to 12 h.

The average vessel turnaround time is directly influenced by another metric known as the *GCR (gross crane rate)* or *quay crane rate*, which is the average lifts achieved at the terminal per QC working hour; where a "lift" refers to either the unloading of an import container from, or the loading of an export container onto the vessel. Clearly the vessel turnaround time at a terminal is inversely proportional to its GCR, that is why the GCR is the most commonly used productivity measure to rate container terminals. The higher the GCR, the better the service quality, so it is a profit measure to be maximized. The gold standard for GCR is 40—good terminals have GCRs in the high 30s.

5 Business Environment in Hong Kong and in HIT in Mid-1990s

China's manufacturing and export boom was creating a remarkable demand for terminal services in Southern China at that time. The surge of exports from the region offered terminals there a golden opportunity; whichever terminal could offer the

Fig. 1.5 Six rows of stored containers and seventh (*rightmost*) row used as a lane for truck passage in a storage block. You can see the yard crane (*YC*) lowering a container retrieved from storage onto an internal truck (*IT*)

capacity and service quality would capitalize on this expanding market. However, with land scarce in Hong Kong, HIT could not simply expand its terminals to meet the growing demand. Also, it faced growing competition from several new terminals opening along China's southern coast. With favorable labor costs and governmental subsidies in China, these ports were poised to offer HIT's customers much cheaper prices than HIT could. So HIT realized that it had to provide premium service quality and establish itself as the industry's benchmark in order to survive.

However, the GCR at HIT was hovering in the upper 20s at that time; consequently their average vessel turnaround time was high. The situation entered a crisis mode in 1995 when they lost a customer, and faced a dire situation of losing market share in a growing market.

The *terminal road network* refers to a directed network representation of the road system within the terminal; with blocks, berths, TG, and road intersections as nodes, and road segments joining pairs of nodes as arcs. This terminal road network is virtually a closed system with the TG as the only access point for ETs; and the only vehicles operating on it are container trucks and the occasional YC moving from one block to another.

Traffic flow in a container terminal is akin to the circulation of blood in the human body: life depends on it. With congestion on the terminal road network, ITs get stuck in traffic, consequently QCs have to wait for them, pulling the GCR down. So, the GM of HIT realized that congestion was the Gordian Knot that needs to be untied before any other improvements can benefit HIT.

The operational practice at that time at HIT (and in all other busy container terminal around the world) was to allocate 6 ITs to serve each working QC. Each IT would shuttle back and forth between the SY and its allotted QC. Realizing that QCs were losing productivity due to waiting for ITs, HIT management decided to increase the allottment of ITs/QC from six to eight, hoping that it would fix the problem. Six months after this change, they found that the new policy made the situation even worse than before. It was counterproductive because increasing the number of trucks operating on the finite terminal road network system

naturally increases traffic congestion. HIT needed to reduce the number of working ITs, but increase the throughput of QCs through other improvements.

So, the GM of HIT decided to take the bull by the horns and reengineer HIT's processes using intensive computerization, and improving its decision-making in daily operations by engaging an outside consultant. I was visiting HKUST at that time. When the GM met me, he told me "Professor Murty, to describe a problem as hard, Americans use the expression "it is like rocket science." I think that congestion on our terminal road network is harder than rocket science. If you can, help us develop a new *DSS (decision support system)* for daily operations that would help reduce it. This will form the core of our program to enhance productivity and capacity."

6 My Involvement with the Problem

It became clear to me that reducing congestion requires a two-pronged approach: reduce the number of trucks operating on the terminal road network, and most importantly, routing them optimally. I did not know anything about solving road congestion problems at that time, so I hesitated that by accepting to work on it, I may be setting my foot into a very dangerous territory. But I was in Hong Kong, and just heard the Chinese proverb given below, that mustered my courage to take the plunge.

千里之行，始于足下

A long journey begins with the first step

The first thing that I did was to scan the literature for any publications in prestigious journals on alleviating traffic congestion in terminal road networks. Even though it is a common problem, I did not find any research publications on this problem at that time. So, we were on our own.

7 Reducing the Number of Vehicles Operating on the Terminal Road Network

The operational procedure at that time was having eight ITs allotted to serving each QC, these line up in a queue to serve only that QC. Thinking about this, I was reminded about an analogous problem that we discussed in a Queueing theory course at UCB, in which we analyzed a system with many servers working simultaneously. We compared two policies: Policy 1 would maintain a separate queue in front of each server, and arriving customers can join any queue that they like at the time of arrival;

Policy 2 maintains a single queue which every customer joins at the time of his/her arrival, and the customer at the front of the queue will be serviced by any server that becomes free next. I remembered learning that Policy 2 gives better performance results than Policy 1.

Applying this to the container terminal situation, Policy 1 corresponds to the existing practice of having a separate batch of eight ITs to serve each QC. Policy 2 corresponds to a new operational procedure that I called the *pooling system* in which all the ITs working in the terminal form a pool that collectively serves all the QCs. This pooling system needs a new *central dispatching unit* which will be communicating with each IT driver, and continuously monitoring the number of QCs lined up and waiting under each QC. It directs each IT returning from the SY to the quay, to join under the QC with the smallest number of ITs waiting under it at that time. The queueing theory result suggests that this pooling system will have better performance than the existing system.

So, I asked the HIT management if this pooling system can be implemented. They estimated that the required communication and monitoring equipment alone would require an initial investment of US$ 11 million, a substantial investment for this industry. But being in an urge to modernize, they decided to go ahead with it.

In a conference a couple of years later, the manager at the Port of Singapore Authority told me that they were also adopting this pooling system for ITs at their container terminals. Now every busy container terminal in the world has adopted this pooling system. This pooling system helped reduce the number of ITs operating in the terminal while simultaneously improving their service level to the QCs (see Sect. 8.3.3 for details).

8 Routing Container Trucks

While the pooling system helped improve performance, I realized that in order to achieve a significant dent in congestion, we need to route container trucks optimally. At that time, HIT had ten berths; over 200 ITs, each of them continuously on the move between the SY and the quay; 80,000 TEUs of constantly accessed storage space; and 10,000 truck passages through the TG daily. I realized that it is really impractical to develop a route that each IT driver is required to follow for each trip between the SY and the quay. Also, all the IT drivers know the terminal roads intimately, and know instantly what route to take to get to any destination point as fast as possible. So ideally we must make sure that route optimality to minimize congestion follows as a consequence of other decisions made in daily operations. I realized that the policy of allocating storage spaces in the SY to arriving containers is the key to achieve optimum routing, as that determines the routes that ITs take.

HIT organizes its work using a 4-h interval (half-a-shift) as the *planning period*. In each period, they work on planning how to make decisions during the next period.

8.1 A 0-1 Model; the First Model that We Tried

In the literature, I have seen several publications on this storage space allocation problem for arriving containers. All of them modeled the problem using binary variables of the form:

$x_{ijkl} = 1$ if the ith container from the jth QC is stored in the kth stack of block ℓ, 0 otherwise.

The model is large with many binary variables, and took too long to solve. At that time, I attended a container terminal decision-making session in a conference. One of the speakers said that he was able to reduce the solution time to an hour. In daily operations in a terminal, to retrieve a container at the bottom of a stack of five stored containers, the crane operator has to first move the top four containers in this stack to other stacks, this is a common operation called *reshuffling*. So a storage space which is unoccupied at an instant, may not be vacant even a few minutes later due to this reshuffling. The input data for this binary variable model is the set of vacant storage positions in the SY, it keeps changing constantly in the container terminal, so I realized that this binary variable model for storage space allocation to arriving containers is not only impractical, but also totally inappropriate.

8.2 A Multicommodity Flow Model; the Second Model that We Tried

So, we decided to develop a new model for this problem on our own. We then realized that the best strategy to minimize road congestion is to spread the container truck traffic on all the road segments evenly, i.e., to equalize the container truck flows on all the arcs of the terminal road network as much as possible. This lead us to the second model for the problem which is a multicommodity flow model. The data needed for the model is the expected number of containers flowing into and out of each node in the terminal node network. For instance, during the planning period:

- at each QC position on the quay, these data consist of how many import containers are expected to be unloaded and sent to the SY from there, and how many export containers are expected to be sent here from each block of the SY
- at each block, how many export containers are expected to be dispatched to each QC position, and how many import containers are expected to be retrieved for leaving the terminal through the TG
- at the TG, how many export containers are expected to arrive for entry into the terminal, etc.

The decision variables in this multicommodity flow model are:

f_{ij} = the total number of container trucks flowing on arc (i, j) in the terminal road network during the planning period.

The objective function to minimize is $\theta - \mu$, where θ = maximum$\{f_{ij}$: over all arcs $(i, j)\}$, and μ = minimum$\{f_{i,j}$: over all arcs $(i, j)\}$.

For each 4-h planning period, the multicommodity flow model is a large-scale linear program (LP), but we could solve it in a few minutes of cpu-time using the best available LP software system at that time, and the output led to the routes for container trucks to minimize congestion.

We ran into very serious difficulties trying to implement the optimum solution from this model. First, truck drivers resented being told what routes to take. Their union said "We all know the terminal road network well, and can find the best route to get to any destination point by ourselves based on current traffic conditions." Second, this model tries to optimize the handling of the total expected workload in the planning period of 4 h, in that process its solution handles this total workload evenly over the time interval of this period. So, inherently it is assuming that all this work is available at the terminal at the beginning of the period, and can be handled at a uniform rate over time. But in reality, in the arriving streams of ECs and ICs, each container has to be handled at the instant of its arrival, and the flow rates of these streams are quite uneven over time because of the stochastic nature of vessel arrivals. So this model also, is not really appropriate for our problem.

When we realized this, I became totally frustrated, and at a loss to know what to do next. Then the Telugu proverb given below came to my mind:

చీకటిని తిడుతూ కూర్చోకు, ఒక చిరుదీపం వెలిగించు

Don't just sit there cursing the darkness, light a small candle

8.3 A Different Linear Programming Model; the Third Model that We Tried

I made up my mind to find that candle to shed light on an approach for solving the congestion reduction problem. I realized that I needed to have more intimate knowledge about the various operations going on inside the terminal, and the consequences of various decisions. So, I spent good 3 weeks in the terminal just observing the operations. During this time I noticed that the:

Occupancy or *fill-ratio in a block* = (number of containers in storage in the block)/(number of storage positions in it)
varied significantly over blocks. At HIT, the number of storage positions in each block = 600. All the blocks with higher fill-ratios tended to have much more truck traffic around them compared to those with low fill-ratios.

This gave me the idea that equalizing the fill-ratios among blocks ensures equal distribution of traffic on terminal roads, minimizing congestion. Fill-ratios in blocks of course vary with time as containers are added to or retrieved from them. So, we needed to select a specific time-point in each planning period at which we try to equalize the fill-ratios in all the blocks, and we decided that this point of time will be the end of each planning period. This led to our next model for the problem, which is again an LP model but much smaller. The decision variables and data elements

in this model are the following (all the data elements are either known or can be estimated with good precision for the planning period):

x_i = *Container quota number* for block i in this period = number of arriving new containers in this period to be dispatched to block i for storage, a decision variable.

a_i = Number of stored containers that will remain in block i at the end of this period if no new containers are sent there for storage during this period, a data element.

N = Number of new containers expected to arrive for storage in this period, a data element.

B, A = Total number of blocks in the storage yard, number of storage positions in each block, data.

$w_i(t)$ = Number of container trucks waiting in block i to be handled by the yard cranes there at time point t, data.

$x_i^R(t)$ = Remaining container quota number for block i at time point t in the period = x_i − (number of new containers sent to block i for storage up to time point t in this planning period), data.

8.3.1 Fill-Ratio Equalization Policy

This policy determines the decision variables x_i for this period to make sure that the fill-ratios in all the blocks are as nearly equal as possible at one time point in the period, the end of the period. The fill-ratio in the whole yard at the end of this period will be $F = (N + \sum_i a_i)/(A \times B)$. If the fill-ratios in all the blocks at the end of this period are all equal, they will all be equal to F. So, this policy determines x_i to guarantee that the fill-ratio in each block will be as close to F as possible by the end of this period. The LP formulation of this problem is

$$\text{Minimize} \sum_i (u_i^+ + u_i^-)$$

$$\text{subject to } a_i + x_i - (u_i^+ - u_i^-) = A \times F \quad \text{for all } i$$

$$\sum_i x_i = N$$

$$x_i, u_i^+, u_i^- \geq 0 \quad \text{for all } i$$

The number of variables in this model is B = number of blocks in the SY, which is typically of the order of 100 or so for large, busy terminals. Also, because of the special structure of this LP, it can be shown that its optimum solution is given by a simple combinatorial scheme described below:

1. Rearrange blocks so that a_i increases with i.
2. Determine x_i in increasing order of i using:

$$x_1 = \text{Minimum}\{N, \text{Maximum}\{0, A \times F - a_1\}\}, \text{ and for } i \geq 2$$
$$x_i = \text{Minimum}\{\text{Maximum}\{0, A \times F - a_i\}, N - \textstyle\sum_{r=1}^{i-1} x_r\}.$$

Numerical Example Suppose there are $B = 9$ blocks in the terminal, each with $A = 600$ storage spaces. Suppose $N = 1040$ new containers are expected to arrive for storage in the planning period. Data on a_i are given below, with a_i already increasing with i. The fill-ratio over the whole yard at the end of this period will be $F = (\sum a_i + 1040)/(9 \times 600) = (2570 + 1040)/(5400) = \approx 0.67$, and $A \times F \approx 400$.

The combinatorial scheme described above determines x_i in the order $i = 1, 2, \ldots$, to bring $a_i + x_i$ to 400 until all 1040 new containers are allotted to a block for storage. The results are shown below.

i	a_i	x_i
1	100	300
2	120	280
3	150	250
4	300	100
5	325	75
6	350	35
7	375	0
8	400	0
9	450	0
Total		1040

Note that this policy only determines the container quota number for each block, not the identity of which containers will be stored in each block. That will be determined by the dispatching policy discussed below.

8.3.2 Arriving Container Dispatching Policy

Irrespective of how the container quota numbers x_i are determined, congestion will be created at a block if we send a consecutive sequence of arriving container trucks to it in a short time interval. To avoid this possibility, it is necessary to allow the yard crane in that block enough time to handle a truck sent there before sending the next one.

Hence this policy dispatches each arriving truck (at the terminal gate, and each berth) at time point t in the period to a block i satisfying: $w_i(t) = \text{Min}\{w_j(t) : j$ satisfying $x_j^R(t) > 0\}$, i.e., a block with remaining positive quota that has the smallest number of trucks waiting in it.

8.3.3 Performance Benefits

The policies discussed above were implemented, and they resulted in significant improvements in performance. With the pooling system, optimized storage space allocation, dispatching and routing, congestion has decreased as evidenced by a 16 % decrease in IT cycle time. Actually, we also worked on some other decision-making problems in daily operations in developing the DSS, for details see [3, 4].

Implementing this DSS for decision-making in daily operations resulted in a 30 % improvement in the GCR at HIT, and a reduction in the average vessel turnaround time from over 13 h to 9 h. The average number of ITs deployed/QC decreased from 8 to 4. Since this project was initiated, HIT's annual throughput has gone from about 4 million TEUs in 1995 to about 6 million in 2002, which they continued to handle using the same labor force, resulting in significant savings.

9 The Substitute Objective Function Technique

Here instead of minimizing congestion measured directly by either the maximum flow amount on an arc in the terminal road network, or $\theta - \mu$ defined earlier, we were able to control it by minimizing the sum of absolute deviations in the fill-ratios of the various blocks from their average. The latter objective function has a strong influence on the former, which is why the substitution produced excellent results. This is the idea behind the substitute objective functions approach developed in [2] for other routing problems. By finding suitable objective functions to substitute, this approach can be used to handle complex large scale optimization problems in many different areas. The main requirements are that the substitute objective function has a strong influence on the original, and optimizing it should be simpler than the original problem.

Many other problems in industry and business deal with processing arriving streams of items, orders, etc. When each arrival has to be processed at the time of arrival, traditional approaches based on batch processing (i.e., planning for a batch of expected arrivals in a period) do not usually work well. In such cases, developing dynamic real-time strategies to handle each arrival based on the prevailing conditions at that time, as we developed here for arriving containers, may offer a good approach.

10 The Lessons that I Have Learnt

As you know there is this eternal conflict between theoreticians and practioners:

> Theoreticians say that practitioners do *practically nothing*
> Practitioners say that theoreticians do *nothing practical*.

I consider myself belonging to both camps, and I learnt two things from my experience: one is that constructing an intelligent model for problems encountered in applications is often quite challenging. The second is that implementing the solution from the model requires listening very carefully to the viewpoints of all the stakeholders and requires being very tactful as explained in Chap. 3 of the recent book [4].

Summary OR academic curricula in general emphasize the teaching of techniques and theory, and do not pay much attention to cultivating good modeling skills in their students. With significant fractions of OR graduates opting for applications-oriented careers these days, I think it is important that modeling find a prominent place in OR curricula. The case study discussed above points out the importance of intelligent modeling to get good results in applications. The aim of this book is to provide a collection of realistic cases that will help the students in developing the modeling skill, to teach them how to choose an algorithm (either one that has been theoretically proven to converge; or a heuristic verified to yield good solutions, and is easier to implement in the real-world setting where it will be used) for solving the model, and how to take advantage of any special structure in the problem to simplify the model and the algorithm used to solve it.

11 A Practical Exercise

Consider a container terminal with a storage yard consisting of 100 blocks each with storage space to hold 600 containers, numbered serially 1 to 100. For $i = 1$ to 100, during a particular 4-h planning period, if no newly arriving containers in this period are sent to block i for storage, the number of stored containers in it is expected to be respectively:

320, 157, 213, 96, 413, 312, 333, 472, 171, 222;
439, 212, 190, 220, 372, 101, 212, 251, 86, 79;
295, 138, 343, 281, 372, 450, 100, 183, 99, 505;
99, 254, 330, 279, 300, 150, 340, 221, 79, 119;
43, 71, 219, 363, 98, 500, 413, 259, 182, 391;
360, 447, 181, 233, 414, 30, 333, 427, 251, 83;
144, 404, 76, 84, 196, 336, 411, 280, 115, 200;
117, 284, 263, 477, 431, 297, 380, 327, 155, 360;
360, 290, 350, 157, 116, 141, 82, 116, 99, 78;
220, 182, 96, 301, 121, 278, 372, 119, 282, 310.

The terminal estimates that in this period 15,166 new containers will be unloaded from docked vessels and dispatched to the storage yard for storage. Determine an optimum plan to allocate these new containers to blocks for storage to minimize congestion on the roads inside the terminal.

References

1. Murty, K. G., Karel, C., & Little, J. D. C. (1962). Branch and bound method for the traveling salesman problem. Unpublished 1962 technical report. http://www-personal.umich.edu/~murty/.
2. Murty, K. G., & Djang, P. A. (1999). The US Army national guard's mobile simulators location and routing problem. *Operations Research, 47*(2), 175–182.

3. Murty, K. G., Wan, Y.-W., Liu, J., Tseng, M. M., Leung, E., Lai, K.-K., & Chiu, H. W. C. (2005). Hongkong international terminals gains elastic capacity using a data-intensive decision-support system. *Interfaces, 35*(1), 61–75.
4. Murty, K. G. (2010). *Optimization for decision making: Linear and quadratic models*. Springer (2010, ISBN 0884-8289).
5. Petering, M. E. H., & Murty, K. G. (2009). Effect of block length and yard crane deployment systems on overall performance at a seaport container transshipment terminal. *Computers and Operations Research, 36,* 1711–1725.
6. Petering, M. E. H., Wu, Y., Li, W., Goh, M., & de Souza, R. (2009). Development and simulation analysis of real-time yard crane control systems for seaport container transshipment terminals. *OR Spectrum, 31,* 801–835.

Chapter 2
Diesel Locomotive Fueling Problem (LFP) in Railroad Operations

Bodhibrata Nag and Katta G. Murty

1 Introduction

With many modes of transport (trucking, rail, and shipping where this option is available) now-a-days there is intense competition in the commodities transport industry, and in order to survive in the business, companies have to keep their charges low. Consequently they have to keep their operating costs low.

About 75 % of transport by railroads in the world is based on diesel locomotives; the remaining 25 % is mostly running on electrified track. While, almost all the goods transported by rail in Europe are on electrified track, the situation in the US is the reverse with almost all of it powered by diesel locomotives. One of the major components in the operating cost of diesel powered rail transport industry is the cost of fuel. This case study deals with minimizing the cost of fuel and the cost of contracting trucks that deliver the fuel to the locomotives used in goods transport powered by diesel locomotives. The cost of fuel is highly location dependent (due to local taxes and transportation costs between supply and demand points), locomotive fueling problem (*LFP*) discussed in this chapter is a critical problem in railroad operations. Given: the set of yards, the set of trains to operate, the locomotive assignments to trains and the fuel cost and capacity data; this problem deals with finding the fueling plan for the various trains to minimize the total cost of fueling the locomotives.

The case study is a simplified real-life problem constructed and set up for the "*Problem Solving Competition-2010*" organized by the *Railway Applications Section (RAS)* of INFORMS (Institute for Operations Research and Management

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_2) contains supplementary material, which is available to authorized users.

B. Nag (✉)
Operations Management Group, Indian Institute of Management Calcutta,
Calcutta, West Bengal, India
e-mail: bnag@iimcal.ac.in

K. G. Murty
IOE Department, University of Michigan, Ann Arbor, MI 48109-2117, USA
e-mail: murty@umich.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_2, © Springer Science+Business Media New York 2015

Fig. 2.1 Locomotive fuel tank being loaded with fuel (*Source*: http://www.locomotiveservice.com/index.php)

Science). The statement of the problem and all the data sets in it can be seen at http://www.informs.org/Community/RAS/Problem-Solving-Competition/ 2010-RAS-Competition or at the website for the book in http://extras.springer.com. Kamalesh Somani (Kamalesh_Somani @CSX.com) of CSX Transportation, and Juan C. Morales (Juan.Morales@BNSF.com) of BNSF Railways contributed to this problem and the data sets in it.

In this problem, as in most of the countries, cost/gallon of diesel varies from yard to yard. We describe 3 different algorithms that we used to solve this problem and highlight the summary of solutions obtained by each of them for comparisons of these algorithms.

2 Brief Description of the Problem in the Case Study

The problem deals with N [= 214] trains hauled by L [= 214] locomotives on a railroad network consisting of Y [= 73] yards over a 2 week *planning horizon*. The yard to yard distances, over the railroad network is given; the average yard to yard distance is 285.66 miles with standard deviation of 44.54 miles, median and mode of 300 miles. All locomotives are assumed to be identical in performance.

Each train visits a sequence of yards (referred to as *route* in the paper). For example, the route for the train T10 is the sequence (Y43, Y16, Y11, Y2, Y3, Y29, Y28, Y23) of yards, where Y43, Y23 are the origin, destination yard; and Y16, Y11, Y2, Y3, Y29, Y28 are all intermediate yards in that order in this route. A few characteristics of the trains included in the case are:

- All the 214 trains operate daily. Thus 214 trains originate every day from respective originating yards.
- 52 trains reach the destination yard the same day it leaves the originating yard. The remaining 162 trains reach the destination yard the next day.
- 135 trains ply between two yards only. Of these 135 trains, only 49 trains reach destination the same day (examples are trains T2 and T4); the remaining 86 trains reach destination the next day(examples are trains T1 and T3).

- 35 trains traverse only one intermediate yard between origin and destination yards. Of these 35 trains, 11 trains reach the intermediate and destination yards the next day (examples are trains T35 and T51) and 21 trains reach the intermediate yard on the starting day, but reach the destination yard the next day(examples are trains T36 and T52).
- 20 trains traverse two yards between origin and destination yards. Of these 20 trains, 14 trains reach the first intermediate yard the next day (examples are trains T13 and T14), 18 trains reach the second intermediate yard the next day (examples are trains T13, T14, T16, and T94) and all trains reach their destination the next day.
- 16 trains traverse three yards between origin and destination yards. Of these 16 trains, 7 trains reach the first intermediate yard the next day (examples are trains T7 and T8), 10 trains reach the second intermediate yard the next day (examples are trains T7, T8, T33, and T40), 15 trains reach the third intermediate yard the next day (examples are trains T7, T8, T33, T34, and T39) and all trains reach their destination the next day.

Each route may be hauled by a different locomotive on different days; the allocation of locomotives to routes is given as data. For hauling each train in this case study, only one locomotive is used. Each locomotive may haul different trains on different days.

All the routes operated in the case study problem can be grouped into a set of *pairs*, each pair operating between a pair of yards in the forward and reverse directions; but the set of yards visited in the two directions for a *route-pair* may be different. Every route pair has a dedicated pair of locomotives operating it. When we refer to a *yard* on a route, we mean either the origin or destination yards of the route, or an intermediate yard where the train has a scheduled stop. An example is locomotives L1, L2 hauling trains T1 on route (Y25, Y19) and T2 on route (Y19, Y25); with L1, L2 hauling T1, T2 respectively on days 1, 3, 5, 7, 9, 11, 13; and L2, L1 hauling T2, T1 respectively on days 2, 4, 6, 8,10, 12, 14 of the planning horizon. Another example is locomotives L5, L6 hauling trains T5 on route (Y36, Y60, Y62), T6 on route (Y62, Y36) on alternate days.

The locomotive of each train can be refueled by fueling trucks positioned at any of the yards on its route except the destination yard, but the total number of refuelings on any route should be ≤ 2. All locomotives have the same fuel capacity of 4500 gallons; and the fuel consumption (3.5 gallons of fuel per mile) on any route is independent of the route and the locomotive operating the train. Fuel consumption for a locomotive traveling between any two yards can be determined using the given table of inter-yard distances. Fuel cost at different yards varies between $ 2.90 to 3.56/gallon (with average of $ 3.13 and standard deviation of $ 0.17). Every time a locomotive is refueled, a setup cost of $ 250 has to be paid in addition to the price of the fuel loaded. Fuel is dispensed by fueling trucks positioned at yards, see Figs. 2.1, 2.2, each having a maximum capacity of 25,000 gallons/day and involving a one-time contracting cost of $ 8000 for the 2 week planning horizon (the contracting cost is $ 4000 per week per truck). The problem statement allows each locomotive

Fig. 2.2 A locomotive being refueled by a fueling truck (*Source*: http://www.locomotiveservice.com/index.php)

to start on the very first trip with any feasible amount of fuel (referred to as "*initial fuel*" in this paper) without any cost incurred; the locomotive should be left with the same amount of fuel after completing the last trip in the planning horizon.

2.1 Desired Outputs

We need to determine: (a) which yards will serve as fueling points for the locomotives (these yards, where refueling trucks are contracted are called "*committed yards*" in this paper), (b) which yard will be used to refuel the locomotive hauling each train, (c) the days and the amount of fuel loaded at each yard used as refueling point for each train, (d) the number of fueling trucks contracted at each yard, (e) and the amount of fuel(in gallons) in each locomotive tank at the beginning of the planning horizon. We need to minimize the total cost = fuel costs + fueling truck contracting costs + setup costs for refueling.

3 Discussion on Methods Used for Solving this Problem

Typically when faced with problems like this, OR specialists will try to build a mathematical model for it (the appropriate model for this problem will be an MIP (mixed integer programming) model). For solving an MIP model, the company needs to have access to a software package like CPLEX, but many railroad companies may not have access to such packages.

So we started looking for an approach which is much simpler to solve the case study problem, gives comparable results, and scales up easily to problems of the size encountered in real world applications. We developed a *greedy algorithm* for the LFP, which meets all these requirements (see [1] for a description of greedy methods), which we discuss below. However, for the sake of comparison, we developed a

mathematical model for this case study problem and solved it using CPLEX; we discuss this model and the results from it in Sect. 6.

The first step in developing a greedy method for solving this problem is to develop a "*greediness criterion*" for the decisions to be made in it. Keeping the objective function to be minimized in the problem in mind, there are two greediness criteria that we can use for selecting a yard p as a refueling yard for a route:

$GC1_p$ = fuel cost incurred at yard p for the route, if yard p is selected as a refueling yard

$GC2_p$ = (fuel cost incurred at yard p for the route) + (incremental truck contracting cost at this stage, if yard p is selected as a refueling yard for this route)

In the following section, we will discuss the greedy algorithm developed for the problem based on *GC1*.

4 Description of the Greedy Algorithm, Algorithm1, Using *GC1* as the Greediness Criterion

The decisions in this algorithm are made in the specific order given below, one route at a time.

i. Partitioning the set of routes into various labeled categories There are several pairs of yards $\{y_i, y_j\}$ such that (y_i, y_j) and (y_j, y_i) are both routes and this pair of routes are operated by a pair of locomotives dedicated to these routes only, each locomotive hauling one of these pairs alternately on alternate days. Route pairs of this type with no intermediate stops in either direction are classified into a set or *category* labeled R_1. Routes (Y19, Y25) and (Y25, Y19) operated by locomotives L1 and L2 belong to category R_1. The set R_2 consists of the remaining routes with at least one scheduled stop in one or both the directions.

R_1 is again partitioned into:

- R_{11} (for these, full locomotive tank capacity is insufficient to cover the round-trip distance from origin to destination and back),
- R_{12} (for these a full locomotive tank capacity is sufficient to cover the round-trip distance from origin to destination and back).

In the case study problem, $R_{11} = \emptyset$, R_{12} contains 59 % of all the routes with 33 % of all the mileage in the problem.

R_2 consists of route pairs, with at least one stop at an intermediate yard in the forward or reverse direction or both, each operated by a dedicated pair of locomotives hauling in each direction alternately. R_2 is again partitioned into:

- R_{21} (for these, a full locomotive tank capacity is sufficient to cover the entire round trip pair; this contains 28 % of all the routes covering 33 % of all the mileage in the problem), and

- R_{22} are the remaining (in each route pair here, at least one refueling is needed in each direction; this set contains 12 % of all the routes covering 34 % of all the mileage in the problem).

Routes (Y10, Y7, Y2, Y12, Y30) and (Y30, Y12, Y10) are operated by locomotives L11 and L12. The round trip distance from yard Y10 back to yard Y10 through the yards Y7, Y2, Y12, Y30, and Y12 is 941 miles. Each locomotive thus requires 3293.5 gallons with fuel consumption at the rate of 3.5 gallons of fuel per mile. Since the locomotive tank capacity is 4500 gallons, the entire round trip pair can be covered with a single filling in the round trip. This route-pair belongs to the R_{21} category.

Routes (Y43, Y41, Y56, Y57, Y51) and (Y51, Y57, Y56, Y41, Y43) are operated by locomotives L7 and L8. The round trip distance from yard Y43 back to yard Y43 through the yards Y41, Y56, Y57, Y51, Y57, Y56, and Y41 is 2010 miles. Each locomotive thus requires 7035 gallons with fuel consumption at the rate of 3.5 gallons of fuel per mile. Since the locomotive tank capacity is 4500 gallons, the entire round trip pair cannot be covered with a single filling in the round trip. This route-pair belongs to the R_{22} category.

Therefore we will adopt the policy of refueling locomotives serving routes in categories R12 and R21 at most once in each round trip pair of routes and those in categories R11 and R22 at least once on each origin to destination route in this category.

ii. Identifying refueling yards for routes in each Category in Algorithm1 Here we discuss how this method selects the refueling yards to be used on each route; but not the actual refueling plan for each locomotive which will be discussed later in sub-section (iv). Refueling yards are selected using the greediness criterion *GC1* defined in Sect. 3 for the yards.

Clearly each route in $R_1 (= R_{12}$ in the case study problem) can only be refueled on any day at the origin yard of route hauled on that day, if we choose to refuel it on that day. Also, each locomotive on these routes in R_{12} need not be refueled every day.

We will now discuss how the refueling yards are selected on each route in this algorithm, for each category of routes separately.

Category R_{11} For the routes belonging to set R_{11} in a general problem, there is no choice other than to refuel the locomotive at the origin yards of all trains hauled over the planning horizon on these routes. Thus the origin yards of all routes in set R_{11} will be refueling yards and refueling trucks are committed to be positioned at these yards.

Category R_{12} Define a set called the "current" set R'_{12} initially $= R_{12}$, $S'_{12} =$ set of origin yards for routes in R'_{12} and $Y'_{12} =$ yard with the cheapest fuel cost among those in S'_{12}. For each route in R'_{12}, for which Y'_{12} is the origin yard, fix Y'_{12} as the committed refueling yard, and delete those routes from the set R'_{12} from further consideration. If R'_{12} is now $\emptyset$, go to the next category. Otherwise update S'_{12}, Y'_{12} using the current R'_{12} and repeat this step.

To illustrate, in the first iteration, S'_{12} ={Y1, Y2, Y3, Y6, Y10, Y13, Y14, Y15, Y17, Y18, Y19, Y21, Y22, Y23, Y24, Y25, Y26, Y27, Y30, Y32, Y34, Y36, Y37, Y38, Y40, Y41, Y43, Y44, Y45, Y46, Y47, Y48, Y49, Y50, Y52, Y53, Y54, Y56, Y57, Y58, Y59, Y60, Y62, Y63, Y64, Y65, Y66, Y67, Y69, Y70, Y73} and Y'_{12} =Y60. In the second iteration, S'_{12} = {Y1, Y2, Y3, Y10, Y13, Y14, Y15, Y17, Y18, Y19, Y21, Y22, Y23, Y24, Y25, Y26, Y27, Y30,Y32,Y34,Y38,Y40,Y41,Y43,Y44, Y45,Y46,Y47,Y48,Y49,Y50,Y52,Y53,Y54,Y57,Y58,Y59,Y63,Y64,Y65,Y66,Y67, Y69,Y70,Y73} and Y'_{12} =Y32, etc.

Proceeding in this manner, the other committed yards obtained in ascending order of fuel cost are Y34, Y25, Y17, Y53, Y54, Y52, Y64, Y1, Y38, Y44, Y23, Y13, Y46, Y50, Y49, Y41, Y24, Y66, Y47, Y59, Y15, Y22, Y30, Y3, Y10, Y63 and Y58. At least one of the yards of the 63 R_{12} route pairs are covered by these 29 committed yards. Both yards of a few route pairs associated with yards {Y47, Y15}, {Y3, Y30}, {Y23, Y66}, {Y60, Y41}, {Y41, Y32},{Y60, Y15}, {Y1, Y15}, {Y3, Y13}, {Y15, Y30}, {Y15, Y38}, and {Y34, Y59} are committed yards; in such cases, fueling is always done at the yard with lesser fuel cost (for example at yard Y30 for route pairs associated with yards Y3 and Y30).

Category R_{21} For each route pair in this category, on every route if there are no previously committed yards amongst the set of yards visited by it, the yard having the cheapest fueling cost on it will be fixed as the committed refueling yard. There are no instances of this type in the case study; all R_{21} category route pairs have at least one previously committed yard amongst the set of yards visited by them.

If there is a committed yard Y_k on this route-pair (i.e. the yard Y_k has been committed as a refueling yard earlier and has the cheapest fuel cost amongst all the committed yards on this route-pair) and if Y_k has the cheapest fuel cost on this route, then commit this yard Y_k as the refueling yard for the route pair. On the other hand, if the cheapest fuel cost yard on this route-pair is Y_p not in the committed list, compute s_{kp} = saving in fuel cost by fueling this route-pair at Y_p rather than Y_k, = (fuel cost at yard Y_k − fuel cost at yard Y_p) × (fuel requirement for the roundtrip route pair over the planning horizon). If $s_{kp} \leq 8000$, which is the total truck contracting cost, then this route will be fueled at the previously committed fuel station Y_k. Else it will be refueled at the yard Y_p which will now be added to the committed refueling yard list. This procedure is repeated for all category R_{21} route-pairs.

For example, the route pair (Y68, Y64, Y20, Y23) and (Y23, Y20, Y64, Y68) corresponding to trains T43 and T44, have previously committed yards Y23 and Y64. The fuel cost at the previously committed yards Y23 and Y64 are $ 3.04 and 2.98 respectively; thus Y64 is chosen for comparison. The cheapest fuel cost on this route is $ 2.92 at Y68. The total distance covered in round trips by the route pair over the planning horizon is 12,460 miles, for which the fuel requirement is 43,610 gallons. With a difference of $ 0.06 in fuel cost per gallon between yards Y64 and Y68, s_{kp} works out to $ 2616.60. Since $s_{kp} < 8000$, this route pair is refueled at previously committed yard Y64.

All the 30 R_{21} category route pairs in the case study are thus refueled at previously committed yards and there is no addition to the committed refueling yard list comprising of 29 yards.

Category R_{22} In each of the route pairs in this set, at least one refueling is needed in each direction. We will now discuss how the refueling yards are selected in each direction of a general route pair in this category.

Fact 1 *It turns out that one refueling in each direction is sufficient in the case study problem, hence greedy method selects just one refueling yard in each direction of the route pairs.*

We check the following for each couple of yards $\{y_i, y_k\}$ where y_i, y_k belong to different direction routes in this pair. If the mileage of the route from y_i to y_k or y_k to y_i is greater than what can be covered by one locomotive tank capacity, discard this pair. If the mileage of the portion of the route pair from y_i to y_k,or y_k to y_i can both be covered by one locomotive tank capacity, then call this pair of yards as a *feasible couple* (Fact 1 implies that there will be at least one feasible couple of yards), and compute f_{ik} = (fuel consumption on the portion of the route-pair from y_i to y_k)(cost of fuel per gallon at y_k) + (fuel consumption on the portion of the route-pair from y_k to y_i)(cost of fuel per gallon at y_i). Among all feasible couples $\{y_i, y_k\}$, choose that optimal couple corresponding to the lowest value of f_{ik} as the pair of committed refueling yards for the route-pair under consideration. It may be noted here that the optimal couple may be $\{y_i, y_i\}$, which implies that re-fueling may be done at the same yard in different directions; for example, it is optimal to refuel at yards Y3 and Y3 for the route pairs (Y23, Y28, Y29, Y3, Y2, Y16, Y43) and (Y43, Y16, Y11, Y2, Y3, Y29, Y28, Y23) associated with trains T9 and T10. This procedure is repeated for all route-pairs in category R_{22}.

For example, fueling trucks positioned at yards Y41 and Y51 spaced 752 *route-miles* apart (from Y41 to Y56 to Y57 to Y51) can serve the route pairs (Y43, Y41, Y56, Y57, Y51) and (Y51, Y57, Y56, Y41, Y43) corresponding to trains T7 and T8. While Y41 is a previously committed yard, Y51 is added to the committed refueling yard list. Similarly yards Y29, Y11, Y33, Y8, Y20, and Y6 are added to the committed refueling yard list to cater the 14 R_{22} category route pairs in the case study.

iii. Determining initial fuel *FI*, at origin yards for routes, in Algorithm1 If the originating yard for a route at the beginning of the planning cycle is a committed yard, the initial fuel amount *FI* in the locomotive for that route is zero. If the originating yard for a route at the beginning of the planning cycle is not a committed yard, the initial fuel amount *FI* in the locomotive for that route is the fuel required to reach the first committed refueling yard on it from the origin.

iv. Fueling Plan of Locomotives & Number of trucks contracted at Committed Yards in Algorithm1 Once a setup cost for refueling the locomotive is incurred, the greedy method tries to take the full advantage of it by filling the locomotive to full capacity. This principle helps us to determine the fuel loaded at each refueling stop of each locomotive.

So fill up the locomotive tank at the first refueling yard on the route, to the full capacity of the locomotive fuel tank. At subsequent committed refueling yards on this route, refuel to the full capacity of the locomotive fuel tank for Category R_{11} and R_{22} routes. For category R_{12} and R_{21} routes, refuel to full tank capacity at every $\left\lceil \frac{\textit{Locomotive tank capacity}}{(\textit{fuel consumption rate in gallons per mile})(\textit{round trip mileage})} \right\rceil$ round trips.

At the last refueling in the planning horizon, refuel each locomotive only to the extent that a balance fuel amount equal to the initial amount of fuel *F1* for this route will be left in its fuel tank at the end of that trip.

For example, for route pair (Y25, Y19) and (Y19, Y25), the fuel truck(s) is/are committed to be positioned at yard Y25. Counting the days of the planning horizon as days 1 to 14, locomotive L1 operates train T1 on route (Y25, Y19) on days 1, 3, 5, 7, 9; and train T2 on route (Y19, Y25) on days 2, 4, 6, 8, and 10 with refueling. On day 11 locomotive L1 will be starting at the origin yard Y25 to operate the route (Y25, Y19); it will be left with only 475 gallons of fuel in it, which is insufficient to cover the roundtrip consisting of the route (Y25, Y19) on this day and the route (Y19, Y25) the next day. So before starting at Y25 on day 11 locomotive L1 needs to refuel; and since only 4 more days are left in the planning horizon and the total fuel needed for the routes it has to cover in these days is 1610 gallons, fuel amount of $1610 - 475 = 1135$ gallons is refueled into this locomotive at yard 25 on day 11. Then it covers train T1 on days 11, 13; and train T2 on days 12, 14; and will be left with an empty fuel tank (same as initial fuel) at the end of the planning horizon.

The other locomotive operating this route pair {(Y25, Y19), (Y19, Y25)} is L2 operating train T2 on the route (Y19, Y25) on day 1. Y19 is not a fueling yard, so this locomotive needs an initial fuel amount of 403 gallons = fuel required to reach the fueling yard Y25 on this route from its origin yard Y19 on day 1. Locomotive L2 gets 4500 gallons filled its fuel tank at yard Y25 on the days 2, 4, 6, 8, 10, and on train T1 on route (Y25, Y19) on days 3, 5, 7, 9. On day 11 at the origin yard Y25 on train T1 on the route (Y25, Y19) it will have a fuel amount 475 gallons only left, not enough to cover the round trip {(Y25, Y19),(Y19, Y25)}, so it has to refuel on this day. Again since only 4 days are left in the planning horizon, and the fuel amount needed to cover the remaining trips in the planning horizon is 1135 gallons; on day 11 at the origin yard Y25 an amount of 1135 gallons is filled in its fuel tank. Locomotive L2 then operates trains T1 on the route (Y25, Y19) on days 11, 13, and train T2 on route (Y19, Y25) on days 12, 14; and will have 403 gallons of fuel at the end of the planning horizon (same as the initial fuel amount at the origin yard on day 1).

Using the fueling plan obtained above and the given information on yards visited on each route on each day of the planning horizon, the fuel dispensed at each committed yard on each day of the planning horizon is computed. The number of fueling trucks contracted at any committed yard y_i is determined by the formula

$\left\lceil \frac{p_i}{q} \right\rceil$ where p_i is the maximum of the daily fuel dispensed at the committed yard y_i obtained in the above procedure, and $q = 25{,}000$ gallons is the maximum capacity of each fueling truck.

Fig. 2.3 A fueling truck refueling a locomotive (*Source*: http://www.bei-benedict.com/locomotive-refueling.php)

4.1 Highlights of Solution Obtained by Algorithm1

The solution consists of 36 committed refueling yards, and the total number of trucks contracted at them is 43. See Fig. 2.3. Number of contracted trucks varies from 1 to 3 at different committed yards; number of contracted trucks is 1 at 30 committed yards, and 2 at 5 committed yards. The total cost in the solution is $ 11.5 million, of which the cost of fuel is $ 10.84 million, operating costs of fueling trucks is $ 0.34 million and fueling setup cost is $ 0.32 million. The utilization of the fueling trucks over the planning horizon varies from a minimum of 3 to 60 %. Algorithm1 was coded on C language, which takes about 0.5 s for compiling and solution on a 1.6 GHz computer.

Considering the mileage of each route, and the lowest fuel cost of the yards visited on each route, the total fuel cost in any solution is guaranteed to be $\geq$ $ 10.7 million. The total fuel cost in the solution obtained in Algorithm1 is $ 10.75 million. The fuel cost is 94 % of the total cost in our solution. From this we can conclude that the solution obtained by Algorithm1 is nearly optimal.

5 Description of the Greedy Algorithm, Algorithm2, Using *GC2* as the Greediness Criterion

Partitioning of set of routes into various categories of routes is the same as in Algorithm1, Sect. 4.

The aim of Algorithm2 is to minimize the total cost (cost of fuel dispensed + refueling setup cost + cost of contracting fuel trucks) as well as ensure maximum utilization (*utilization* is defined as the ratio of the mean amount of fuel dispensed by the truck to its maximum daily refueling capacity) of the fuel trucks at the previously committed yards. As an illustration we will use a made-up example not related to our case study, involving a pair of routes in which the pair of yards (P, Q) is a feasible

couple, with two fuel trucks already committed at yard P and the utilization of this truck being only 28 % (the trucks dispense 40,500 gallons on the first day, 36,000 gallons on second day, 24,192 gallons on fifth day, 24,192 gallons on sixth day, 3864 gallons on seventh day, 3864 gallons on eighth day, 24,192 gallons on ninth day, 24,192 gallons on tenth day, 1625 gallons on 11th day, 5440 gallons on 13th day and 5440 gallons on 14th day of the planning horizon; the average fuel dispensed daily by the two trucks over the planning horizon is 13,822 against the maximum daily dispensing capacity of 50,000 gallons giving a utilization of 28 %). Algorithm2 examines whether it is feasible to meet the refueling demands of the route pair using the existing two fuel trucks at P or it is cheaper to commit a fuel truck at Q. In case the refueling demands of the route pair is 44,100 gallons over the planning horizon (4500 gallons on first day, 9000 gallons on second day, 4200 gallons on fifth day, 8400 gallons on sixth day, 4200 gallons on ninth day, 8400 gallons on tenth day, 1800 gallons on 13th day and 3600 gallons on 14th day), then refueling demands can be catered by daily dispensing capacity of the existing two fuel trucks at yard P; in such a case, it would not make any sense to commit a fuel truck at Q unless the fuel cost at Q is much lower than P (for example, if the fuel cost at Q is cheaper by $ 0.20, the fuel cost savings for the 44,100 gallons will be $ 8820 and the net saving after considering the truck contracting cost of $ 8000 would be $ 820 by deciding to commit a fuel truck at Q). Algorithm2 thus differs from Algorithm1, where the aim was to choose yards with cheapest fuel cost in order to minimize the cost of fuel alone.

This algorithm differs from Algorithm1 of Sect. 4 in the greediness criterion used for selecting refueling yards for various routes. Also in this algorithm, in contrast to Algorithm1, since the selection of the refueling yard on each route is based on the greediness criterion *GC2* defined in Sect. 3, it requires determining the amount of fuel to be dispensed at each yard on each day of the planning horizon on that route before the selection can be made. In Algorithm2 also, refueling yards are selected for route pairs one at a time sequentially, in the order R_{11}, R_{12}, R_{21} and R_{22}. At any stage, let T denote the set of committed refueling yards selected up to that stage.

For Category R_{11} the procedure for selection of refueling yards for routes is the same as in Algorithm1. Since the locomotive has to be refueled to full capacity at the origin yard of each route in R_{11}, T will be the set of all yards on routes in R_{11}.

Then prepare a two-way table G of fuel requirements at yards y in T on day $j \in \{1 \ldots 14\}$ in the planning horizon, where entry g_{yj} in this table is the fuel dispensed at yard y on day j for the routes processed already. Prepare a new column $H = (h(y)$: $y \in \{1,2,\ldots,74\}$ = set of all yards) where the entry $h(y)$ = number of contracting fuel trucks needed at this yard y to dispense this fuel over the planning horizon.

For Category R_{12}, order the route pairs in it in decreasing order of round-trip mileage; we will process the routes in this order.

We will select the refueling yard for each route pair r in R_{12} by the following procedure. For each of the terminal (i.e., origin or destination) yards y on this route, define $h_2(y)$ = the fuel + refueling setup cost + truck contracting cost of dispensing all the fuel commitments at this yard y for the route pair r under consideration. The truck contracting cost is to be considered only when either (1) yard y is not a

member of the set T or (2) though yard y is a member of the set T the total fuel requirement (calculated by considering the fuel requirements g_{yj} at the yard y on day j for routes processed already and the fuel requirement for the each route pair r under consideration) on any given day in the planning horizon exceeds the daily fuel dispensing capacity of the trucks available. Choose the terminal yard y corresponding to the minimum value of $h_2(y)$ as the refueling yard for this route pair and update the column H and set T accordingly. A similar principle is used in processing routes in R_{21}, R_{22}.

For the case study, there are no R_{11} routes; thus $h(y) = 0$ for all yards and $T = \emptyset$. Hence we start with R_{12} category routes, ordering the route pairs in decreasing order of round-trip mileage. First, we consider the route pair (Y45, Y17) and (Y17, Y45) with 1200 round-trip miles corresponding to trains T64 and T65; the fuel cost is \$ 2.96 and 3.16 at Y17 and Y45 respectively. Since $h_2(45) = \$\ 102{,}906$ and $h_2(17) = \$$ 97026, Y17 is chosen as the refueling yard for this route pair; thus $h(17)$ is now updated to \$ 97,026 and $T = \{Y17\}$. Continuing in this manner, we obtain $T = \{Y1, Y2, Y13, Y15, Y17, Y22, Y23, Y24, Y25, Y30, Y32, Y34, Y41, Y44, Y48, Y49, Y50, Y57, Y60, Y64, Y66\}$ for all the R_{12} category route pairs. R_{22} category route pairs are handled similarly.

Highlights of solution obtained by Algorithm2 The solution consists of 29 committed refueling yards and total number of trucks contracted at them is 37 (number of contracted trucks varies from 1 to 3 at different committed yards; number of contracted trucks is 1 at 31 committed yards and 2 at 4 committed yards). The total cost in the solution is \$ 11.7 million, of which the cost of fuel is \$ 11.07 million, operating costs of fueling trucks is \$ 0.3 million, and fueling setup cost is \$ 0.35 million. The utilization of the fueling trucks over the planning horizon varies from a minimum of 5 to 63 %. Algorithm2 was coded on C language, which takes about 1.5 s for compiling and solution on a 1.6 GHz computer.

6 The Mixed Integer Programming (MIP) Model for the Case Study Problem and its Solution Obtained Using CPLEX Software

There are many different ways to model this problem using an MIP model. The main decisions to be made in this problem are: the set of yards which will serve as refueling yards, the yard(s) where each locomotive will refuel, the days on which this refueling will take place and the quantity of fuel loaded at each refueling, and the number of fuel trucks contracted at each refueling station. The problem of making all these decisions to minimize the total cost = fuel cost + setup costs for refueling + truck contracting costs can be modeled as an MIP (mixed integer programming) model, but it leads to a model with many 0-1 variables, and may take a long time to solve.

Thus MIP model can be simplified by using the partition of the 214 routes for the trains into the three categories R_{12}, R_{21}, and R_{22} discussed above; and using the information:

- Each locomotive serving the routes in category R_{12} needs to be refueled only periodically as discussed in the previous section,
- Each locomotive serving routes in category R_{21} needs to be refueled only once on each round trip,
- Each locomotive serving routes in the category R_{22} needs to be refueled only once in each direction of each roundtrip, and
- Each locomotive is filled to capacity at each refueling.

Also we notice that the fuel cost is the major element in the total cost; and all the other costs put together are only a small fraction of the fuel cost. If we take the fuel cost as the major part of the objective function to minimize, the model becomes even much simpler.

The solutions obtained in Algorithms 1, 2 had 36, 37 committed yards with 30, 31 among them having only one fueling truck contracted at them. So, it seems that we can approximate the total truck contracting cost at an optimum solution by 8000 (number of committed yards in the solution). Using this, we consider the simpler problem of minimizing the [fuel costs + 8000 (number of committed yards)] subject to all the constraints.

Here is the MIP model for this simplified version of the problem. In the following when we talk about "refueling on a route p", it refers to "refueling of a locomotive serving route p". Since each route pair is served by a unique pair of locomotives, given the planning horizon of 14 days, we conclude that both locomotives serving any route pair, will log equal distance by the end.

Using the fueling plan and the given information on yards visited on each route on each day of the planning horizon, we determine the (a) the days and the amount of fuel loaded at each yard used as refueling point for each train, (b) the number of fueling trucks contracted at each yard, (c) and the amount of fuel(in gallons) in each locomotive tank at the beginning of the planning horizon.

Indices used:

p: this index is used for routes, $p = 1$ to 214
j: this index is used for yards, $j = 1$ to 73
k: this index is used for routes in the forward or reverse direction between a pair of terminal yards for trains in R_{22} category, $k = 1,2$
c: this index is used for feasible couples for trains in R_{22} category

Notation for data elements:

S_p: set of yards on route p
C_p: set of feasible couples on route p in R_{22} category; for example, there are 14 feasible couples for the locomotive serving route pairs (Y23, Y28, Y29, Y3, Y2, Y16, Y43) and (Y43, Y16, Y11, Y2, Y3, Y29, Y28, Y23): (Y23, Y43), (Y23,

Y16), (Y28, Y43), (Y28, Y16), (Y28, Y11), (Y29, Y16), (Y29, Y11), (Y29, Y2), (Y3, Y2), (Y3, Y3), (Y2, Y3), (Y2, Y29), (Y16, Y29), (Y16, Y28)

F_{pc}: set of yards in feasible couple c on route p in R_{22} category (for example, (Y23, Y43) is the set of yards in feasible couple $c = 1$)

t_j: fuel cost at yard j in terms of \$/mile= (\$/gallon) (3.5 gallons/mile)

m_p: total distance covered by the locomotives serving route p in entire 2-week planning horizon

Decision Variables: All the decision variables are binary:

$y12_{pj} = 1$, if refueling for category R_{12} route p is carried out at yard j
=0, otherwise

$y21_{pj} = 1$, if refueling for category R_{21} route p is carried out at yard j
= 0, otherwise

$y22_{pjk} = 1$, if refueling for category R_{22} route p in direction k is carried out at feasible yard j
= 0, otherwise

$y22_{pcj} = 1$, if the yard j in a feasible couple c refuels category R_{22} route p
= 0, otherwise

$w_{pc} = 1$, if refueling for category R_{22} route p is carried out at yards in feasible couple c
= 0, otherwise

$x_j = 1$, if refueling trucks are committed at yard j
= 0, otherwise

The Objective Function to minimize is:

$$\text{Minimize} \sum_{p \in R12, j \in S_p} y12_{pj} t_j m_p + \sum_{p \in R21, j \in S_p} y21_{pj} t_j m_p +$$
$$\sum_{p \in R22, c \in C_p} \sum_{j \in F_{pc}} (w_{pc} y22_{pcj} t_j m_p) + \sum_{j=1}^{73} 8000 x_j$$

Subject to the Constraints:

1. $\sum_{j \in s_p} y12_{pj} = 1, \quad for\ all \quad p \in R12$
2. $y12_{pj} \le x_j, \quad for\ all \quad p \in R12, \quad j \in S_p$
3. $\sum_{j \in s_p} y21_{pj} = 1, \quad for\ all \quad p \in R21$
4. $y21_{pj} \le x_j, \quad for\ all \quad p \in R21, \quad j \in S_p$
5. $\sum_{j \in F_{pc}} y22_{pcj} \le w_{pc} + 1, \quad for\ all \quad p \in R22, \quad c \in C_p$
6. $\sum_{c \in C_p} w_{pc} = 1, \quad for\ all \quad p \in R22$
7. $\sum_{c \in C_p} \sum_{j \in F_{pc}} (w_{pc} y22_{pcj}) = 2, \quad for\ all \quad p \in R22$
8. $y22_{pcj} \le x_j, \quad for\ all \quad p \in R22, \quad c \in C_p, \quad j \in F_{pc}$

Explanation of the constraints: Constraint (1) comes from the fact that each locomotive serving on a route in category R_{12} is to be refueled at one of the two terminal nodes between which the locomotive goes back and forth, that refueling trucks are to be committed at that yard for refueling. The binary variable $y12_{pj}$ decides the yard where refueling of the particular route p will take place even though refueling trucks

may be located at both the terminal nodes. Constraint (2) ensures that refueling truck is actually available at the chosen terminal node. Constraints (3) and (4) come from the corresponding facts for locomotives serving routes in category R_{21}. Constraints (5), (6), and (7) ensures that for category R_{22} only a feasible couple of yards as defined in Section 4 is chosen as a pair of refueling stations for each route pair in R_{22}. Constraint (8) is similar to earlier constraints (2) and (4).

Highlights of solution obtained by MIP Model The solution consists of 29 committed refueling yards and total number of trucks contracted at them is 40. Number of contracted trucks varies from 1 to 3 at different committed yards. The total cost in the solution is $ 11.56 million(which is only 0.5 % higher than that obtained by Algorithm1), of which the cost of fuel is $ 10.92 million, operating costs of fueling trucks is $ 0.32 million and fueling setup cost is $ 0.32 million.

The utilization of the fueling trucks over the planning horizon varies from a minimum of 3 to 65 %. The MIP model takes about 3 seconds for processing and solution using IBM ILOG CPLEX 12.1.0 on a 1.6 GHz computer.

7 Summary

As we have seen from the results obtained in Sects. 5 & 6, the results obtained in the greedy method depend on the greediness criteria used for making the decisions. Here is a summary for comparing all three methods:

	Greedy method with *GC1*	Greedy method with *GC2*	MIP model
Number of yards where refueling trucks are committed in the optimal solution	36	29	29
Total number of trucks contracted in the optimal solution	43	37	40
Total truck contracting cost in the optimal solution (million $)	0.34	0.30	0.32
Total fuel cost in the optimal solution (million $)	10.84	11.07	10.92
Total refueling setup cost in the optimal solution (million $)	0.32	0.35	0.32
Total cost in the optimal solution (million $)	11.50	11.71	11.56

The solution obtained using Algorithm1 with *GC1*, is comparable to that obtained from the MIP model. Also Algorithm1 is very simple and easy to implement, and scales directly to large scale problems in real world applications.

The solution obtained using Algorithm1 is comparable to the solutions obtained by the three winners of the INFORMS-RAS competition [4]. Mor Kaspi and Tal Raviv of Tel-Aviv University obtained a total cost of 11.4 million using a MILP

formulation (with specialized cuts and domination rules). Cristian Figueroa and Virot Chiraphadhanakul of MIT's Operations Research Center obtained a total cost of 11.4 million using a MIP model (with Lagrangean relaxation sub-models). Ed Ramsden of Lattice Semiconductor Corporation obtained a total cost of 11.5 million using a combination of heuristics and stochastic search algorithms.

In the case study problem, the greedy method based on the greediness criterion *GC1* gives better results than that based on *GC2*. Since fuel cost is over 94 % of the overall costs in the optimal solution, *GC1* based on fuel cost as greediness criteria works better than *GC2*. Also since the case study problem is a realistic example of real world problems, we can expect the similar performance in general in practice.

In real world applications it is very likely that there will be more trains on tracks and consequently, most contracted fueling trucks tend to be used to full capacity; and the difference in the total costs associated with *GC1* and the MIP-based technique similar to the one discussed in Sect. 6 may not be much. The greedy method based on either the greediness criteria *GC1* or *GC2* are very easy to implement and do not need software packages like CPLEX and take very little CPU time.

In most real world applications, partitioning the trains into categories like *R11*, *R12*, *R21*, *R22* like in the case study example; and identifying sections of the track where the locomotive of the train will undergo refueling possibly based on practical logistic considerations, may in fact be the preferred option of railroad managements. Then, if the company does not have access to an MIP software, the greedy method based on *GC1* may be the preferred approach for solving the problem. On the other hand if the company has access to MIP software, a model like the one in Sect. 6 can be used to model and solve this problem as explained in Chap. 3 of [2].

8 A Practical Exercise

Consider the same problem discussed in this chapter. Assume that all the data remains the same, except the fuel prices at the various yards, which have changed. The new prices at yards 1 to 73 are (2.93, 2.96, 2.99, 3.02, 3.05, 3.08, 3.11, 3.14, 3.17, 3.20, 3.23, 3.26, 3.29, 3.32, 3.35, 3.38, 3.41, 3.44, 3.47, 3.50, 3.56, 2.90, 2.93, 2.96, 2.99, 3.02, 3.05, 3.08, 3.11, 3.14, 3.17, 3.20, 3.23, 3.26, 3.29, 3.32, 3.35, 3.38, 3.41, 3.44, 3.47, 3.50, 3.53, 2.90, 2.93, 2.96, 2.99, 3.02, 3.05, 3.08, 3.11, 3.14, 3.17, 3.20, 3.23, 3.26, 3.29, 3.32, 3.35, 3.38, 3.41, 3.44, 3.47, 3.50, 3.53, 2.90, 2.93, 2.96, 2.99, 3.02, 3.05, 3.08, 3.11) respectively in that order. Solve the modified problem with this new data and discuss the output obtained.

Acknowledgments We thank RAS of INFORMS, and Kamlesh Somani, Juan C Morales for permission to use this problem as a case study problem for this chapter and for agreeing to maintain the statement of the problem for future readers at the site mentioned in [3]. This problem specified by the data sets here can be used as a classroom project problem illustrating the usefulness of the greedy method in real world applications.

References

1. Murty, K. G. (1994). *Operations research: Deterministic optimization models*. Englewood Cliffs: Prentice Hall.
2. Murty, K. G. (2010). *Optimization for decision making: Linear and quadratic models*. Springer 2010, ISBN 0884-8289.
3. Railway Applications Section (RAS) of INFORMS. Problem Solving Competition-2010. http://www.informs.org/Community/RAS/Problem-Solving-Competition/2010-RAS-Competition. Accessed 7 Nov 2011.
4. Railway Applications Section (RAS) of INFORMS. RAS-2010-ProblemSolvingCompetition (PDF file) downloaded from INFORMS website. http://www.informs.org/content/download/188885/1887915/file/RAS-2010-ProblemSolvingCompetition.pdf. Accessed 7 Nov 2011.

Chapter 3
Organizing National Elections in India to Elect the 543 Members of the Lok Sabha

Bodhibrata Nag and Katta G. Murty

1 Brief History of Governance and National Polls in India

Human civilization has been evolving and flourishing in India since ancient times, going back well over 10,000 years. In those days it was not a single unified country, but was several little countries, each a dynastic kingdom with each king being succeeded by his eldest son. In those days the main roles of a king used to be: regularly holding his court where performances by famous artists (music and dance performances, poetry recitations, gymnastic performances, etc.) are held, patronizing the arts and skilled trades, hearing complaints and pleas from his subjects, and settling disputes among his subjects; and finally expanding his kingdom by waging wars on neighboring kingdoms (Figs. 3.1 and 3.2).

There were two religions in India at that time, each with its own god, and followers of each religion were divided into various castes based on professions. Each had its own temples for worship (Fig. 3.3).

There used to be frequent disputes among the various castes each claiming to be superior to the other, and fights among the religions each claiming that their god is the true god.

In this climate, in the sixth century BCE (before current era), the crown prince of one of the kingdoms, whose name was Siddhartha Gautama Buddha, became totally disillusioned with the conflicts among various factions. He took the bold step of renouncing his title to the kingdom, and started preaching the social equality of all human beings, ignoring all castes and gods altogether, and a way of life based on

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_3) contains supplementary material, which is available to authorized users.

B. Nag (✉)
Operations Management Group, Indian Institute of Management Calcutta,
Calcutta, West Bengal, India
e-mail: bnag@iimcal.ac.in

K. G. Murty
University of Michigan, Ann Arbor, MI, USA
e-mail: murty@umich.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_3, © Springer Science+Business Media New York 2015

Fig. 3.1 Sculpture of a dancer (*Source*: http://www.kamat.com reproduced with permission from K. L. Kamat)

morality only, which attracted a huge following all over Asia and led to the creation of Buddhism.

Also at that time the remaining people in the two religions in India realized that their gods are two different forms of the same Supreme Being, and merged themselves into one combined religion called Hinduism. Ever since, Hinduism has maintained this spirit of accepting every other religion into its fold, considering their god as another form or "avatar" of that one Supreme Being.

Dynasties of kings continued to rule their individual kingdoms in India the same way. Then in the third century CE (current era) one king Asoka waged a big war on a neighboring kingdom in which his forces were barely victorious. When he visited the battlefield at the end of the war and saw all the corpses, torn limbs, and blood, he became disgusted with war and realized that the duty of a king should be to devise means by which the lives of his subjects become happier, and not that of waging wars for the sake of expanding the kingdom. He started building various amenities to improve the quality of life of his countrymen. For this he is now recognized as the world's first *social entrepreneur*.

In those days India used to be the center of attraction for people from all over the world, for traders, travelers, people seeking education and enlightenment, and for immigrants.

Then came the Muslim invaders who settled down and established their own dynastic kingdoms, and built their monuments like the well-known Taj Mahal (Fig 3.4).

Then in the seventeenth century CE the British arrived as traders at first, but when they saw an opportunity they seized it and set themselves up as alien rulers. Then a

Fig. 3.2 Sketch of a king and queen in their court sitting on their thrones watching a dance performance with their ministers in pre-British India. (Courtesy of Soma Raju Karuparthi gaaru)

Fig. 3.3 A temple in India (Photo of Ranganathaswamy temple in India courtesy of Keith E. Stanley, http://www.kestan.com/travel/tamilnadu/)

large portion of the territory in India was ruled by Governor-Generals appointed by this alien kingdom.

Indians had to undergo a long-drawn struggle to gain independence from Britain. Finally they were able to achieve it with no war but a peaceful struggle under the leadership of Mohandas Karamchand Gandhi. The freedom struggle united the whole country under one nation that became an independent nation in 1947 CE.

Three years after gaining independence from Britain, Bharat (India) became the Bharat Ganrajya (The Republic of India, or Indian Republic) by adopting the Constitution of India in 1950. Now the Indian Republic comprises 28 states and 7 Union Territories, which we will refer to as the 35 states of India hereafter. The Indian parliamentary form of government is federal in structure with legislative powers

Fig. 3.4 The Taj Mahal (*Source*: http://rajivawijesinha.wordpress.com/2010/08/13/historicbuildings-x-the-taj-mahal-at-agra-c-1650-ad/)

distributed between the Parliament of India and State Legislatures. The Parliament of India comprises two legislative bodies—the Rajya Sabha (Upper House, corresponding to the "Senate" in the USA, or the "House of Lords" in the UK), and the Lok Sabha (Lower House, corresponding to the "House of Representatives" in the USA, or the "House of Commons" in the UK). The 250 members of the Rajya Sabha are indirectly elected by legislators of states and Union Territories comprising the Union of India. The 543 members of the Lok Sabha are directly elected by universal adult franchise by the electorate of all the 35 states through the "National Elections," called *General Elections* in India. The term of office of each Lok Sabha is 5 years from the date of its first meeting, unless dissolved earlier due to the ruling party losing a vote on a no-confidence motion in the Lok Sabha. These General Elections were held for the first time in the history of India in 1951–1952 after the adoption of the constitution of India, and regularly after that as depicted in Table 3.1. In this chapter we will consider only the organization of General Elections for electing the members of the Lok Sabha in India.

The total membership of the Lok Sabha is distributed among the 35 states in such a manner that the ratio of the population to the number of seats allotted to any state is nearly the same. The geographical area of the state is then demarcated into a number of territorial constituencies (with geographical boundaries), equal to the number of seats allotted, such that the population of all constituencies in that state is nearly the same. Since there are large variations in population densities across states, constituencies vary largely in terms of geographical area—thus Ladakh (the constituency with largest area) covers 173,266 km^2 in contrast to Delhi-Chandni Chowk (the constituency with smallest area) which covers only 11 km^2. Each constituency has a large number of polling stations distributed across the constituency such that voters can reach the polling stations to cast their vote with minimum travel. The distribution of membership of the Lok Sabha and the total number of polling stations for each state is given in Table 3.2.

Table 3.1 General elections held in India

Lok Sabha	General elections	Date of first meeting	Date of dissolution
1	25 October 1951 to 21 February 1952	13 May 1952	4 April 1957
2	24 February to 14 March 1957	10 May 1957	31 March 1962
3	19–25 February 1962	16 April 1962	3 March 1967
4	17–21 February 1967	16 March 1967	27 December 1970
5	1–10 March 1971	19 March 1971	18 January 1977
6	16–20 March 1977	25 March 1977	22 August 1979
7	3–6 January 1980	21 January 1980	31 December 1984
8	24–28 December 1984	15 January 1985	27 November 1989
9	22–26 November 1989	18 December 1989	13 March 1991
10	20 May to 15 June 1991	9 July 1991	10 May 1996
11	27 April to 30 May 1996	22 May 1996	4 December 1997
12	16–23 February 1998	23 March 1998	26 April 1999
13	5 September to 6 October 1999	20 October 1999	6 February 2004
14	20 April to 10 May 2004	2 June 2004	18 May 2009
15	16 April to 13 May 2009	1 June 2009	–

The General Elections of India are the world's biggest election exercise. During the 2009 General Elections, a 717 million strong electorate exercised their franchise through 1.3 million electronic voting machines deployed in 834,000 polling stations spread across the length and breadth of India to elect 543 members of the Lok Sabha from among 8000 candidates contesting the elections. The only other comparable elections are the European Parliament elections with an electorate of 500 million and the US Congress elections with an electorate of 312 million.

The responsibility for conducting the elections to the Lok Sabha is vested in the Election Commission of India according to the provisions of Article 324 of the Constitution of India. The Election Commission operates through its secretariat based at New Delhi manned by about 300 officials. It is assisted at the state level by the Chief Electoral Officer of the State, who is appointed by the Election Commission in consultation with the state government. The Chief Electoral Officer is assisted by District Election Officers, Electoral Registration Officers, and Returning Officers at the constituency level. In addition, the Election Commission co-opts a large number of officials from the Central (or federal) and state governments for about 2 months during each General Election, for conducting the elections. About 5 million officials were deployed during the 2009 General Election.

Elections in the past have been marked by instances of voter intimidation through violence or harassment in various forms, as well as clashes between political opponents [1]. These incidences have been largely arrested through deployment of additional police forces during the polling process in order to bring peace, restore confidence in candidates and voters, and thereby ensure fair and free elections.

The Constitution of India mandates that maintenance of law and order is the responsibility of the states. Thus, while all states maintain police forces totaling about 1.5 million, the average police–population ratio for all the states is only 133 police per 100,000 [2] in comparison with average international ratio of 342 [3]. The

Table 3.2 Number of constituencies and polling stations in each state

Serial number	State	Number of members of the Lok Sabha	Total number of polling stations
1	Andhra Pradesh	42	66,760
2	Arunachal Pradesh	2	2057
3	Assam	14	18,828
4	Bihar	40	57,020
5	Goa	2	1339
6	Gujarat	26	42,568
7	Haryana	10	12,894
8	Himachal Pradesh	4	7253
9	Jammu & Kashmir	6	9129
10	Karnataka	28	46,576
11	Kerala	20	20,510
12	Madhya Pradesh	29	47,812
13	Maharashtra	48	82,598
14	Manipur	2	2193
15	Meghalaya	2	2117
16	Mizoram	1	1028
17	Nagaland	1	1692
18	Orissa	21	31,617
19	Punjab	13	18,846
20	Rajasthan	25	42,699
21	Sikkim	1	493
22	Tamil Nadu	39	52,158
23	Tripura	2	3008
24	Uttar Pradesh	80	129,446
25	West Bengal	42	66,109
26	Chattisgarh	11	20,984
27	Jharkhand	14	23,696
28	Uttarakhand	5	9003
29	Andaman & Nicobar Islands	1	347
30	Chandigarh	1	422
31	Dadra & Nagar Haveli	1	161
32	Daman & Diu	1	94
33	NCT of Delhi	7	11,348
34	Lakshadweep	1	40
35	Puducherry	1	856

Central Government therefore maintains *Central Police Forces* numbering about 800,000 under 7 different divisions, to complement the state police, whenever and wherever required. [4].

Since the state police are the arm of the state governments, allegations of partisan conduct of police in enforcing law and order during the campaign closing phases and during the day of elections are likely. It has therefore become universal practice to deploy *Central Police Forces* (CPF), in addition to state police at all polling stations during the General Elections (Fig. 3.5). However, only about a quarter of the CPF can be spared for deployment during the elections, which amounts to only about 200,000 personnel. Thus General Elections are spread over different days with each day covering only a few states, such that the required number of CPF personnel can be deployed across

all polling stations of all constituencies of those states. The days of elections are spread a few days apart to allow redeployment of paramilitary personnel and allow them to be familiar with their constituencies. For example, the 2009 General Election was conducted in five phases on 16 April, 23 April, 30 April, 7 May and 13 May.

The movement of CPF personnel from their bases to the polling stations in the different phases and their subsequent return to the bases is a gigantic exercise, requiring coordination between different agencies such as CPF operations, Election Commission and State Chief Electoral Officers, District Election Officers, Railways, airlines, and the Indian Air Force. In the 2009 General Election, 119 special trains, 65 sorties by Indian Air Force transport aircraft, 600 sorties by Indian Air Force helicopters, and Air India chartered flights were used for the cross-country movement of CPF personnel [5].

However the process of scheduling the elections and movement of police personnel is done manually by the Election Commission. This chapter proposes an optimization methodology to find an optimum plan for organizing the General Elections for all the 543 parliamentary constituencies in the minimum possible time, with the available CPF personnel, while minimizing the total cost for the police movement involved. The paper is organized as follows: the problem statement and description of all the data is in Sect. 2; representation, methodology, and algorithms for the problem are described in Sect. 3; followed by discussion of the solutions obtained in Sect. 4 and conclusions in Sect. 5.

2 The Problem Statement and Description of all the Data

The total number of polling stations of all the 543 parliamentary constituencies, spread over 35 states is 833,701. If four CPF personnel are deployed at each polling station, the total requirement of police personnel is 3.3 million if elections are to be held on the same day all over the country. However, since only about a quarter of CPF amounting to 200,000 personnel can be spared for deployment during the elections, additional reserve CPF and army personnel are also used to bring the number to 1.5 million for deployment during the elections. Even then it can be seen that it is not possible to conduct elections for all the 543 parliamentary constituencies on a single day. Thus elections will have to be conducted in phases, with CPF personnel movement between constituencies in the phase intervals. The proposed method assumes that the CPF personnel movement will be entirely by air, except the "last mile" movement between the airports and the constituencies.

2.1 The Constraints in the Problem and the Objective Function to be Optimized

While conducting the elections in phases, the Election Commission requires that the following constraints should be satisfied as far as possible.

2.1.1 Constraints to be Satisfied

a. In every state, the elections in all constituencies in it should be held on a single day.
b. As far as possible, states in which elections are held on a day must be contiguous.
c. The set of states in which elections are held on consecutive polling days should be contiguous.
d. General elections over the whole country should be completed using the smallest number of polling days.
e. At every polling station, four CPF personnel should be deployed on the election day.
f. The total number of CPF personnel available for deployment at polling stations on any polling day is at most 1.5 million.

The proposed method attempts to incorporate all these constraints in the model used to solve the problem.

2.1.2 Selection of the Objective Function to Optimize

In the original statement of the problem, the unit for measuring the objective function is stated in terms of people-miles. But the CFP personnel deployed move from a state to another state in next phase by air (as far as possible), and from the airport to the polling stations in the constituency by road vehicles; and the costs per person-mile by air and road are very different. It is not logical or reasonable to add people-miles by different modes of travel directly to get the objective function to be minimized; particularly since in the end all travel has to be paid for in units of money, *Indian Rupees* (₹), in the total travel cost of all the CFP personnel involved.

So, we use the objective function as the total cost of travel of all the CFP personnel involved in the General Elections. Air travel cost should be the cost of air travel of all the moves made by air for all these personnel. Road travel can be in terms of cost incurred for it, the total cost of mileage of all the road vehicles used for all these personnel.

2.2 The Data for the Problem

All the data for the problem can be accessed in the folder "Election Data" at the following website: http://extras.springer.com in the folder corresponding to the ISBN number of this book. Here is the description of the data made available at this website.

Table D1 at this website gives the number of polling stations in each constituency in each state, the name of the nearest airport to that constituency (CPF personnel deployed to polling stations in this constituency will use this airport to arrive in this constituency and depart from it to their next assignment), and the road distance of

this constituency to that airport. In the following Table 3.3 we show a portion of this information for one selected constituency in each state.

Table D2 on the website provides the air travel cost in Indian Rupees between various pairs of airports. Actual air travel cost of CPF polling personnel is hard to get exactly since some of it is on Indian Air Force Transport Aircraft, helicopters, and some on Air India chartered flights. So this data is based on estimated average cost in ₹ 10/mile (based on 2012 prices) on these different types of flights used, and the air distance data between pairs of airports [9].

Table D3 on the website provides the estimated cost in Indian Rupees of the road or train travel of CPF polling personnel from airport used to the polling stations in the constituency and back for each constituency. This data is based on estimated average cost in ₹ 3/mile by these modes of travel (road or rail) [7, 8].

Table D4 on the website provides for each state all its adjacent states. This data is shown in full in the following Table 3.4. Two states are defined to be *adjacent* if they share a common boundary line.

Table D5 on the website gives the cost in Indian Rupees of travel/person from the various bases from where CPF polling personnel come from, to each constituency, and Table D6 on the website, the number of personnel that come from each base.

Table D7 on the website gives the cost of travel/person from each constituency to each constituency across the country.

3 A Graph Representation of the Problem

A *graph* (also known as an *undirected network*) is a pair of sets $G = (N, A)$ where N is a set of *nodes* (also called *vertices*, these are numbered serially and referenced by their numbers), and A is a set of lines or *edges* (also called arcs in some books), each edge joining a pair of nodes. If an edge joins the pair of nodes i, j it is represented by the pair (i, j) (in some books the symbol $(i; j)$ is used instead); this edge is said to be incident at nodes i and j. Nodes i, j are said to be *adjacent* if there is an edge joining them. See Murty [6] for a discussion of networks and network algorithms. The network G is said to be *connected* if for every pair of nodes p, q in it, there is a path of edges in G connecting them; otherwise it is not connected. See Figs. 3.6, 3.7. Nodes are circles with its number entered inside the circle; each edge is a straight line joining the pair of nodes on it.

A *partial network* of G is a network $G_1 = (N_1, A_1)$ where the set of nodes N_1 is a subset of N, and the set of edges A_1 is the set of all edges of G that have both their incident nodes in N_1. G_1 is said to be the *partial network of* G *induced by* the subset of nodes N_1, it is connected if it forms a connected network. For example for the network G in Fig. 3.6, $G_1 = (\{1, 2, 3\}, \{(1, 2), (2, 3), (3, 1)\})$ is a connected partial network; and $G_2 = (\{1, 2, 3, 4\}, \{(1, 2), (2, 3), (3, 1)\})$ is a partial network which is not connected because there is no path connecting nodes 1 and 4 in it.

Let $\{N_1, N_2, \ldots, N_k\}$ be a partition node set N in G (i.e., their union is G, and every pair of subsets in this partition are mutually disjoint). For $t = 1$ to k, let

Table 3.3 Number of polling stations and nearest airport of select constituencies

Serial number	State	Constituency	Number of polling stations	Nearest airport	Distance to airport (miles)
1	Andhra Pradesh	Adilabad	1464	Ramagundam	85
2	Arunachal Pradesh	Arunachal East	851	Pasighat	52
3	Assam	Karimganj	1229	Silchar	27
4	Bihar	Purvi Champaran	1193	Muzaffarpur	46
5	Goa	South Goa	660	Dabolim Goa	0
6	Gujarat	Anand	1510	Vadodara	22
7	Haryana	Kurukshetra	1263	Chandigarh	52
8	Himachal Pradesh	Mandi	1921	Kulu	20
9	Jammu & Kashmir	Anantnag	1502	Srinagar	31
10	Karnataka	Dharwad	1455	Hubli	12
11	Kerala	Wayanad	988	Kozhikode	40
12	Madhya Pradesh	Bhind	1659	Gwalior	44
13	Maharashtra	Dhule	1624	Aurangabad	79
14	Manipur	Inner Manipur	970	Imphal	0
15	Meghalaya	Shillong	1326	Shillong	0
16	Mizoram	Mizoram	1028	Aizawl	0
17	Nagaland	Nagaland	1692	Dimapur	29
18	Orissa	Cuttack	1319	Bhubaneswar	13
19	Punjab	Jalandhar	1764	Ludhiana	34
20	Rajasthan	Sikar	1574	Jaipur	63
21	Sikkim	Sikkim	493	Darjeeling	30
22	Tamil Nadu	Viluppuram	1376	Pondicherry	19
23	Tripura	Tripura West	1558	Agartala	0
24	Uttar Pradesh	Rae Bareli	1576	Lucknow	46
25	West Bengal	Jalpaiguri	1560	Bagdogra	27
26	Chattisgarh	Raigarh	2166	Bilaspur	81
27	Jharkhand	Kodarma	2097	Gaya	43
28	Uttarakhand	Garhwal	1876	Dehradun	46
29	Andaman & Nicobar Islands	Andaman & Nicobar Islands	347	Port Blair	0
30	Chandigarh	Chandigarh	422	Chandigarh	0
31	Dadra & Nagar Haveli	Dadra & Nagar Haveli	161	Daman	15
32	Daman & Diu	Daman & Diu	94	Daman	0
33	NCT of Delhi	New Delhi	1540	Delhi	9
34	Lakshadweep	Lakshadweep	40	Agatti	0
35	Puducherry	Puducherry	856	Pondicherry	0

Table 3.4 Adjacent states of each state

State	Adjacent state(s)	State	Adjacent state(s)
Andhra Pradesh	Orissa, Chattisgarh, Maharashtra, Karnataka, Tamil Nadu	Arunachal Pradesh	Assam
Assam	West Bengal, Arunachal Pradesh, Nagaland, Manipur, Tripura, Meghalaya, Mizoram	Bihar	Uttar Pradesh, Jharkhand, West Bengal
Goa	Karnataka, Maharashtra	Gujarat	Maharashtra, Madhya Pradesh, Dadra & Nagar Haveli, Daman & Diu, Rajasthan
Haryana	Punjab, NCT-Delhi, Chandigarh, Himachal Pradesh, Uttar Pradesh, Rajasthan	Himachal Pradesh	Punjab, Jammu & Kashmir, Uttarakhand, Haryana, Chandigarh
Jammu & Kashmir	Punjab, Himachal Pradesh	Karnataka	Goa, Maharashtra, Andhra Pradesh, Tamil Nadu, Kerala, Lakshadweep
Kerala	Karnataka, Tamil Nadu, Lakshadweep	Madhya Pradesh	Maharashtra, Gujarat, Rajasthan, Uttar Pradesh, Chattisgarh
Maharashtra	Goa, Karnataka, Andhra Pradesh, Chattisgarh, Madhya Pradesh, Gujarat, Daman & Diu, Dadra & Nagar Haveli	Manipur	Nagaland, Mizoram, Assam
Meghalaya	Assam	Mizoram	Assam, Tripura, Manipur
Nagaland	Manipur, Assam	Orissa	Andhra Pradesh, Chattisgarh, Jharkhand, West Bengal
Punjab	Rajasthan, Haryana, Himachal Pradesh, Jammu & Kashmir, Chandigarh	Rajasthan	Gujarat, Madhya Pradesh, Uttar Pradesh, Haryana, Punjab
Sikkim	West Bengal	Tamil Nadu	Kerala, Karnataka, Andhra Pradesh, Puducherry
Tripura	Assam, Mizoram	Uttar Pradesh	Uttarakhand, Himachal Pradesh, Haryana, Rajasthan, NCT-Delhi, Madhya Pradesh, Chattisgarh, Jharkhand, Bihar
West Bengal	Orissa, Jharkhand, Bihar, Sikkim, Assam	Chattisgarh	Orissa, Andhra Pradesh, Maharashtra, Madhya Pradesh, Uttar Pradesh, Jharkhand

Table 3.4 (continued)

State	Adjacent states	State	Adjacent statesstates
Jharkhand	Orissa, Chattisgarh, Uttar Pradesh, Bihar, West Bengal	Uttarakhand	Himachal Pradesh, Uttar Pradesh
Andaman & Nicobar Islands	Tamil Nadu, Andhra Pradesh	Chandigarh	Punjab, Haryana, Himachal Pradesh
Dadra & Nagar Haveli	Gujarat, Maharashtra	Daman & Diu	Gujarat, Maharashtra
NCT of Delhi	Haryana, Uttar Pradesh	Lakshadweep	Kerala, Goa, Karnataka
Puducherry	Tamil Nadu		

Fig. 3.5 A polling station in India being watched by a CPF personnel (*Source*: Reproduced from ebook "Lok Sabha Election 2009: Reinforcing Indian Democracy" available at http://eci.nic.in/eci_main/dj/book_DJ.pdf)

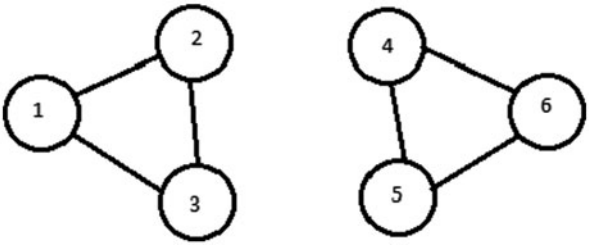


Fig. 3.6 A six-node, six-edge network which is not a connected network (for example there is no path consisting of edges between nodes 1, 4 in this network)

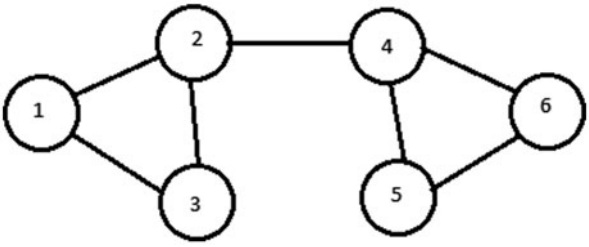


Fig. 3.7 A six-node, seven-edge network that is a connected network

$G_t = (N_t, A_t)$ be the partial network of G induced by N_t. Then $\{G_1, \ldots, G_k\}$ is said to be a *partition* of G. The *graph partitioning problem* is the general problem of finding a partition of a given graph G subject to specified constraints that minimizes a specified objective function.

Our polling problem can be viewed as an instance of the graph partitioning problem. To see this let $NI = \{1, \ldots, 35\}$ where node i represents the state of India with serial number i in Table 3.2. Let $AI = \{(i, j)$: nodes i, j correspond to adjacent states as given in Table 3.4. Then $(NI, AI) = GI$ is known as the *adjacency network* for states in India.

With 1.5 million CPF personnel available for deployment for the elections, and the required four CPF personnel at every polling station, on a single day elections can be held at 1,500,000/4 = 375,000 polling stations at most. Since there are 833,701 polling stations over the whole country, and 833,701/375,000 is strictly between 2 and 3,

we need at least 3 polling days to complete holding the elections over the whole country. (3.1)

So, on any single day, elections are held in a subset of states in the country. Suppose elections are held on k different days. We know that $k \geq 3$. Let NI_t be the set of states in which elections are held on the tth day for $t = 1$ to k, and let $GI_t = (NI_t, AI_t)$ be the partial network of GI induced by the subset of nodes NI_t. So, our problem boils down to finding the partition $(GI_1, \ldots, GI_k)$ of GI satisfying constraints (a) to (f) listed above in Sect. 2.1.1, and *sequence* these partial networks generated corresponding to the election days 1, 2, ..., k; while minimizing the total cost of moving the CPF personnel across the country to monitor the polling stations as required. So our problem actually involves a graph partitioning problem, and a sequencing problem.

In constraint (b) of Sect. 2.1.1, the meaning of the word "contiguous" is left somewhat vague. We will interpret it to mean that the partial network GI_t of GI induced by the set of states in which elections are held on the t-th day should be a connected network for each $t = 1$ to k. Also, we will interpret constraint (c) of Sect. 2.1.1 that the set of states in which elections are held on consecutive days should be "contiguous" to mean that there should be at least one edge in GI joining a node in NI_t to a node in NI_{t+1} for each $t = 1$ to $k - 1$.

3.1 A Hamiltonian Path Heuristic for the Problem

A *Hamiltonian path* in an undirected network G is a path consisting of edges in the network that contains all the nodes in the network. Given the lengths of all the edges in G, the *Hamiltonian path problem* in G is that of finding a shortest Hamiltonian path in G. A Hamiltonian path in G is a path in G that contains all the nodes in G; and a shortest Hamiltonian path is one whose length (the length of a Hamiltonian path is the sum of the lengths of the edges in it) is the shortest among all Hamiltonian paths. In some applications, we may require Hamiltonian paths beginning with a specified

node as the initial node. This can easily be transformed into the well-known *traveling salesman problem*.

Since our CPF polling personnel team has to cover each of the states in India satisfying the constraints listed above, the shortest Hamiltonian path in the network *GI* using the cost of traveling/person between states *i, j* as the length of edge (*i, j*), may provide useful information to develop a good solution to our problem. But there are important differences. No member of the CPF polling team visits all the states; everyone visits only a subset of the states, so our problem is really one of partitioning *GI* into connected partial networks $GI_1, \ldots, GI_k$ of *GI*, corresponding to election days 1 to *k* such that there is an edge joining a node in GI_t to a node in GI_{t+1} for all $t = 1$ to $k - 1$; and each member of the CPF polling team visits one node in each of GI_t for $t = 1$ to *k*.

Define a *segment of a Hamiltonian path* as the portion of this path between a pair of nodes on it. One way of generating partial networks of *GI* satisfying the contiguity requirements in our problem is to obtain a minimum cost Hamiltonian path in *GI*, and then divide it into segments of required size, and take the partial networks of *GI* to be the partial networks induced by the sets of nodes of the various segments. This is the basis for this method, which we will describe next. The method has two stages. Stage 1 determines the election day for each state with day 1 as the starting day of the elections.

Selecting the sequence of partial networks of *GI* to cover the various polling days as the sequence of segments along the shortest Hamiltonian path helps keep the cost of movements of the NPF polling team between consecutive polling days small, thus achieving our objective of minimizing the total cost of travel of this team in conducting the elections, while satisfying the contiguity requirements (b), (c) among the specified constraints of Sect. 2.1.1.

Stage 1: Determining the Election Day for Each State There may be up to 13 different airports that the CPF polling team will use to enter and leave a state. In reality then, each individual of this team has a separate problem to be solved to estimate correctly the cost incurred for his participation; but all the data for that individual is not known until the solution of this stage is determined, making it impractical to get into such precise detail.

So, as an approximation, we will assume that all members of the CPF team visiting a state will travel to and from this state use one airport, the one that majority of this team will use (this is a reasonable and good approximation).

So, *make the cost coefficient of edge* (*i, j*) *in GI to be* c_{ij} = *cost of air travel/person between the airports* (*determined as above*) *corresponding to states i, j.*

For each node i in GI define its weight w_i = *number of polling stations in state i.*

As the polling team moves from one state to the other, our original problem is actually many Hamiltonian path problems, one for each individual in the CPF polling team, depending on the airports they will use for each move. By using a single airport for each state, one that majority will use, we will adopt a single countrywide shortest Hamiltonian path as an approximation to the collection of all of them.

This Stage 1 involves two Steps; we will describe them now.

Step 1: Finding the Shortest Hamiltonian Path in GI We assume that the national elections in India always begin in the Nation's Capital, "NCT of Delhi" (state number 33). So, find a shortest Hamiltonian path in *GI* beginning with state 33 as the starting state, and covering all the airports selected above corresponding to the various states. Let this Hamiltonian path be H. The order of states on H is the order in which elections will be held in the solution determined by this method.

Renumbering of the states Renumber the states in the order of appearance on H beginning with number 1 for the state "NCT of Delhi," the starting state. *In the sequel, we will refer to the states by these numbers* (Fig 3.8 and Table 3.5).

Step 2: Determining the Election Day for Each State Define

$$\textbf{Weight of a segment of } H = \text{sum of the weights } w_i \text{ of nodes on it}$$

$$w_0 = \textbf{maximum number of polling stations that} \tag{3.2}$$

$$\textbf{the CPF polling team can handle in a day} = 1{,}500{,}000/4 = 375{,}000$$

In this step we will break up H into the smallest number of segments subject to the constraint that the weight of each segment is $\leq w_0$; each segment corresponds to an election day in the solution developed by this method. We discuss an algorithm for breaking up H into segments. This problem is known as the *tour segmentation problem.*

Algorithm 1: A Simple Heuristic Algorithm for Breaking up H Into Segments In this algorithm segments are formed one by one. Starting at node 1, move along H until the sum of the weights of states included so far is $\leq w_0$, and exceeds w_0 if the next state is included. At this time complete the current segment, remove it from H, and continue the same way with the remaining part of H.

Let k be the number of segments obtained by this heuristic. Number these segments 1 to k in the order of appearance on H. The solution obtained by this heuristic is to hold elections in all the states in segment p on day p, for each $p = 1$ to k.

We applied this algorithm and found the solution to the problem, in it $k = 3$. From Eq. (3.1), we conclude that this solution is optimal to the problem of breaking up H into the smallest number of segments subject to the constraint that the weight of each segment should be $\leq w_0$. According to this solution, the elections will be held according to the following schedule:

Day 1: NCT-Delhi, Haryana, Punjab, Jammu & Kashmir, Himachal Pradesh, Chandigarh, Uttarakhand, Uttar Pradesh, Bihar, Sikkim, Jharkhand, West Bengal, Meghalaya, Assam, Arunachal Pradesh, Nagaland, Manipur, Mizoram—total 18 states, 245 constituencies, 373,574 polling stations

Day 2: Tripura, Andaman & Nicobar Islands, Orissa, Chattisgarh, Andhra Pradesh, Tamil Nadu, Puducherry, Karnataka, Kerala, Lakshwadeep, Goa, Maharashtra, Dadra & Nagar Haveli, Daman & Diu, Gujarat—total 15 states, 244 constituencies,369,616 polling stations

Day 3: Madhya Pradesh, Rajasthan—total 2 states, 54 constituencies, 90,511 polling stations

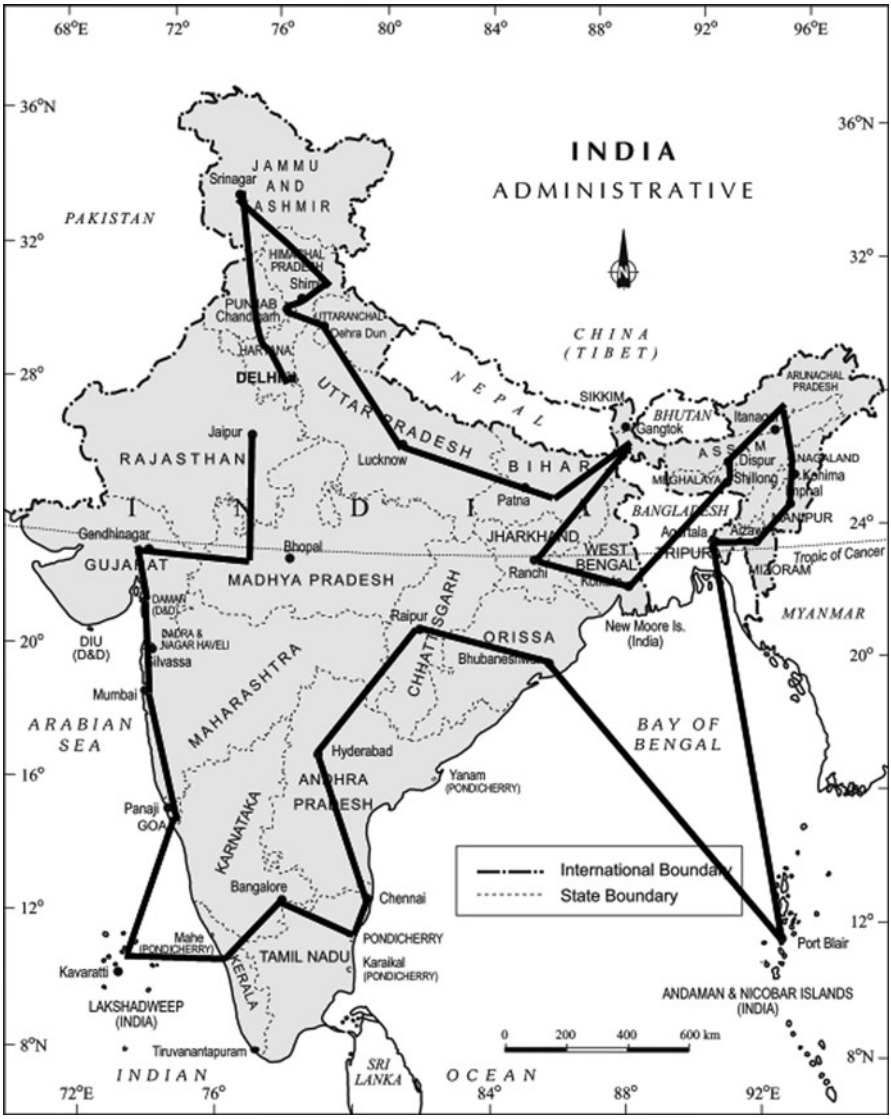


Fig. 3.8 Shortest Hamiltonian path covering the airports used by the CPF team for traveling to and from the states

Algorithm 2: A 0-1 model for breaking up H into segments For constructing this model we need an upper bound on the minimum number of segments into which H can be partitioned subject to the upper bound w_0 on the weight of each segment. From the results obtained from the solution given by Algorithm 1, we know that $k = 3$ is an upper bound that we need. We will describe this model denoting this upper bound by k (in our problem $k = 3$). Define variables

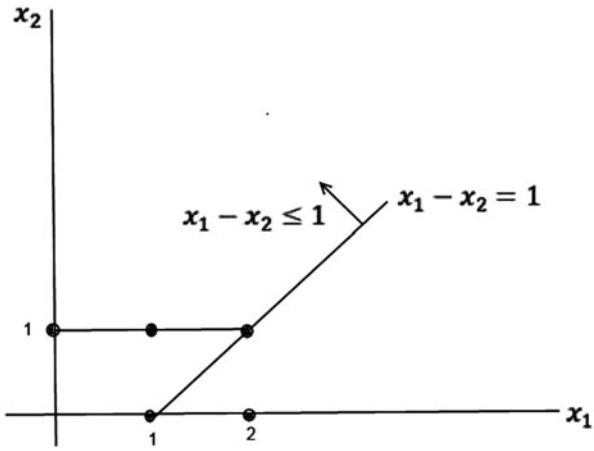


Fig. 3.9 Plot of feasible values for the integer variables x_1 ($= y_{i,j} + y_{i+2,j}$) and x_2 ($= y_{i+1,j}$). All feasible values $\{(x_1, x_2)$: $(x_1, x_2) = (0,0), (0,1), (1, 0), (1,1), (2,1)\}$ are plotted with a "•", and all of them satisfy $x_1 - x_2 \leq 1$

$y_{i,j} = 1$ if state i belongs to segment j, for $i = 1$ to 35, $j = 1$ to k; 0, otherwise.

For each j, we need to form constraints to guarantee that the set of all states i corresponding to $y_{i,j} = 1$ form a segment. For this, for each j and each $i = 1$ to 33 we must guarantee that:

$y_{i,j} = y_{i+2,j} = 1$ implies that $y_{i+1,j} = 1$ also.

This requires that if $y_{i,j} + y_{i+2,j} = 2$ then $y_{i+1,j} = 1$;

and if $y_{i,j} + y_{i+2,j} = 1$ then $y_{i+1,j} = 0$ or 1.

So, for the pair $(y_{i,j} + y_{i+2,j}; y_{i+1,j})$ all the values in the set $\{(0, 0), (0, 1), (1, 0), (1, 1), (2, 1)\}$ are possible, but (2, 0) is not.

Plotting these points on the 2-dimensional Cartesian plane with $y_{i,j} + y_{i+2,j}$ on the horizontal axis and $y_{i+1,j}$ on the vertical axis; we will now see see that $y_{i,j} + y_{i+2,j}$, and $y_{i+1,j}$ must satisfy the constraint $y_{i,j} + y_{i+2,j} - y_{i+1,j} \leq 1$. To see this clearly, denote:

$y_{i,j} + y_{i+2,j}$ by x_1, and

$y_{i+1,j}$ by x_2

and by plotting feasible values of these variables on the x_1, x_2-Cartesian plane we get Fig. 3.9.

From this we see that the system of constraints that the variables $y_{i,j}$ have to satisfy are:

$$\sum_{j=1}^{k} y_{i,j} \leq 1 \quad \text{for each } i = 1 \text{ to } 35$$

$$\sum_{i=1}^{35} w_i y_{i,j} \leq w_0 \text{ for each } j = 1 \text{ to } k$$

$$y_{i,j} + y_{i+2,j} - y_{i+1,j} \leq 1 \quad \text{for all } i = 1 \text{ to } 33, \text{and } j = 1 \text{ to } k$$

$$y_{1,1} = 1$$

$$\sum_{j=1}^{k} j\, y_{i-1,j} \le \sum_{j=1}^{k} j\, y_{i,j} \le 1 + \sum_{j=1}^{k} j\, y_{i-1,j} \quad \text{for all} \quad 2 \le i \le 35$$

$$\text{all } y_{i,j} = 0 \text{ or } 1 \quad \text{for all} \quad i = 1 \text{ to } 35, \quad \text{and} \quad j = 1 \text{ to } k$$

Here is an explanation for these constraints.

From the definition of the binary variables $y_{i,j}$ given above, the first constraint along with the fourth and fifth here guarantees that each of the states $i = 1$ to 35 is contained in exactly one segment.

The second constraint guarantees that the weight of each segment obtained is less than or equal to the maximum possible weight w_0 determined above in Eq. (3.2).

It is discussed above that the third constraint guarantees that for each j the set of all the states i corresponding to $y_{i,j} = 1$ form a segment.

The fourth constraint says that Segment 1 begins with state 1 where the elections begin in the country on the first day of polling.

The fifth constraint says that for each $i = 2$ to 35, either state i belongs to the same segment as state $i - 1$, or the next segment following it.

The last constraint says that all the variables are binary variables. Clearly these are all the constraints in the problem of breaking up H into segments.

From the numbering of the states (defined in Table 3.5), this guarantees that each segment is a portion of the Hamiltonian path H between two nodes, and that segments are numbered serially in the order in which they appear along H.

Now the actual number of segments into which H is partitioned in the solution obtained, is the serial number of the segment containing the last state 35 on H; so this is, $\sum_{j=1}^{k} j\, y_{35,j}$ which we have to minimize.

So the binary integer programming model for this tour segmentation problem is to minimize $\sum_{j=1}^{k} j\, y_{35,j}$ subject to the above constraints.

We solved this 0-1 integer programming problem and obtained an optimum solution, consisting of 3 segments. According to it, the elections will be held by the following schedule:

Day 1: NCT-Delhi, Haryana, Punjab, Jammu & Kashmir, Himachal Pradesh, Chandigarh, Uttarakhand, Uttar Pradesh, Bihar, Sikkim, Jharkhand, West Bengal, Meghalaya—total 13 states, 225 constituencies, 347,776 polling stations

Day 2: Assam, Arunachal Pradesh, Nagaland, Manipur, Mizoram, Tripura, Andaman & Nicobar Islands, Orissa, Chattisgarh, Andhra Pradesh, Tamil Nadu—total 11 states, 136 constituencies, 200,672 polling stations

Day 3: Puducherry, Karnataka, Kerala, Lakshwadeep, Goa, Maharashtra, Dadra & Nagar Haveli, Daman & Diu, Gujarat, Madhya Pradesh, Rajasthan—total 11 states, 182 constituencies, 285,253 polling stations.

Table 3.5 Arcs in the shortest Hamiltonian path H in Fig. 3.8 in order of appearance on H

Arc number i on H	Start node on arc i		End node on arc i		Length of arc i on H (miles)
	State/Union Territory number	Airport	State/Union Territory number	Airport	
$i = 1$	1 = NCT Delhi	Delhi	2 = Haryana	Delhi	0
2	2 = Haryana	Delhi	3 = Punjab	Ludhiana	178
3	3 = Punjab	Ludhiana	4 = Jammu & Kashmir	Srinagar	228
4	4 = Jammu & Kashmir	Srinagar	5 = Himachal Pradesh	Kulu	199
5	5 = Himachal Pradesh	Kulu	6 = Chandigarh	Chandigarh	86
6	6 = Chandigarh	Chandigarh	7 = Uttarkhand	Dehradun	70
7	7 = Uttarkhand	Dehradun	8 = Uttar Pradesh	Lucknow	298
8	8 = Uttar Pradesh	Lucknow	9 = Bihar	Muzaffarpur	279
9	9 = Bihar	Muzaffarpur	10 = Sikkim	Darjeeling	189
10	10 = Sikkim	Darjeeling	11 = Jharkhand	Ranchi	316
11	11 = Jharkhand	Ranchi	12 = West Bengal	Kolkata	201
12	12 = West Bengal	Kolkata	13 = Meghalaya	Shillong	305
13	13 = Meghalaya	Shillong	14 = Assam	Guwahati	46
14	14 = Assam	Guwahati	15 = Arunachal Pradesh	Zero	160
15	15 = Arunachal Pradesh	Zero	16 = Nagaland	Dimapur	116
16	16 = Nagaland	Dimapur	17 = Manipur	Imphal	77
17	17 = Manipur	Imphal	18 = Mizoram	Aizawl	108
18	18 = Mizoram	Aizawl	19 = Tripura	Agartala	91
19	19 = Tripura	Agartala	20 = Andaman & Nicobar Islands	Port Blair	847
20	20 = Andaman & Nicobar Islands	Port Blair	21 = Orissa	Bhubaneswar	751
21	21 = Orissa	Bhubaneswar	22 = Chattisgarh	Raipur	281
22	22 = Chattisgarh	Raipur	23 = Andhra Pradesh	Hyderabad	337

Table 3.5 (continued)

Arc number i on H	Start node on arc i		End node on arc i		Length of arc i on H (miles)
	State/Union Territory number	Airport	State/Union Territory number	Airport	
23	23 = Andhra Pradesh	Hyderabad	24 = Tamil Nadu	Chennai	322
24	24 = Tamil Nadu	Chennai	25 = Puducherry	Pondicherry	84
25	25 = Puducherry	Pondicherry	26 = Karnataka	Bangalore	165
26	26 = Karnataka	Bangalore	27 = Kerala	Kochi	229
27	27 = Kerala	Kochi	28 = Laksh-wadeep	Agatti	285
28	28 = Laksh-wadeep	Agatti	29 = Goa	Dabolim Goa	332
29	29 = Goa	Dabolim Goa	30 = Maha-rashtra	Mumbai	264
30	30 = Maha-rashtra	Mumbai	31 = Dadra & Nagar Haveli	Daman	93
31	31 = Dadra & Nagar Haveli	Daman	32 = Daman & Diu	Daman	0
32	32 = Daman & Diu	Daman	33 = Gujarat	Ahmedabad	182
33	33 = Gujarat	Ahmedabad	34 = Madhya Pradesh	Indore	211
34	34 = Madhya Pradesh	Indore	35 = Rajasthan	Jaipur	291

Like the Heuristic Algorithm 1, the solution obtained by Algorithm 2 also partitions the shortest Hamiltonian path H into 3 segments, which is the minimum possible. Now we will take the best among the solutions obtained from both the methods as the one to implement. Even though both correspond to the same value 3 for the number of segments into which H is partitioned; the one obtained by Algorithm 2 is better than that obtained by Algorithm 1, in other respects. For example, in the solution obtained by Algorithm 2, the number of constituencies and polling stations are more evenly distributed across the segments. Hence we will implement the solution obtained by Algorithm 2.

In the sequel k denotes the number of segments selected, which is 3.

Stage 2: Moving polling personnel from one state to the next during the elections After completing the elections in the states in segment j on day j, the batches of polling personnel have to be moved to the states in segment $j + 1$ for $j = 1$ to $k - 1$.

Construct a bipartite network F with states in segment j as the source nodes, and states in segment $j + 1$ as sink nodes. Join each source node p to each sink node q by a directed arc (p, q) with cost coefficient $c^2_{p,q}$ equal to plane-fare between states p,

q using the appropriate airports in each state to minimize the cost for this move (at this stage with information available we could use different airports than those used in Stage 1 to make the total cost for this stage small. Also at this stage notice that the airports used by different batches of polling personnel may be different in each state. Make *availability* (quantity to be shipped to the set of sink nodes) at each source node equal to the number of batches of polling personnel in the corresponding state on day j, and the *requirement* at each sink node equal to the number of batches of polling personnel required to hold elections at the corresponding state on day $j + 1$. An optimum solution of this bipartite minimum cost flow problem (a transportation problem) gives the moves to be made for this transitions of the polling personnel from day j to day $j + 1$ for each $j = 1$ to $k - 1$.

Moving CPF polling personnel from their bases to states in segment 1 on Day 1, and from states in the last segment back to their bases at the end of elections: This problem can also be modeled as a bipartite minimum cost flow problem exactly as above using the data given in Table D5 on the website, and solved.

4 Results

Solving the model, we obtain a solution with a cost of about ₹ 25 billion. The model takes about 21 seconds for processing and solution using IBM ILOG CPLEX 12.1.0 on a 1.6 GHz computer.

The optimal movement of Central Police Force personnel from Agartala base is given in Tables 3.6, 3.7, and 3.8 as an illustration. Here "number moved" is the number of police moving from a base/constituency to another constituency in the next segment.

"Number required" is the number of police personnel required at that constituency (given by four times the number of polling stations in that constituency).

Hence there are two situations occurring:

- Number moved exactly equals the number required which means that this set of movements is sufficient to meet its needs. For example, in base to segment 1, movement from Agartala Base to West Bengal-Dum Dum (5992/5992)
- Number moved is less than number required, which means that there are policemen coming in from other places to meet the requirement in this constituency. For example, in segment 2 to segment 3 movements, Karnataka-Chikkballapur requires 7292 police which is met by 1788 from Tamil Nadu-Arani and 5504 from Tamil Nadu-Viluppuram. Table 3.6, Table 3.7, Table 3.8

5 Discussion

The method demonstrated in Sect. 4, enables (a) scheduling of elections within the minimum number of segments; (b) sequencing the segments, such that the movement of Central Police Forces (measured in men-miles) is minimized; and (c) sourcing the appropriate number of personnel from the most convenient bases.

Table 3.6 Movement of Agartala based Central Police Force personnel from base to Segment 1

To West Bengal State constituencies (number moved/total number required)	1. Barrackpore (3164/5308) 2. Dum Dum (5992/5992) 3. Barasat (6128/6128) 4. Joynagar (5700/5700) 5. Jadavpur (6516/6516) 6. Kolkata Dakshin (7452/7452) 7. Kolkata Uttar (6584/6584) 8. Howrah (6600/6600) 9. Jhargram (7016/7016) 10. Purulia (6336/6336) 11. Bankura (6752/6752) 12. Bishnupur (6488/6488) 13. Burdwan Durgapur (6752/6752) 14. Asansol (6252/6252)
To Jharkhand State constituencies (number moved/total number required)	1. Dumka (396/6572) 2. Jamshedpur (6504/6504) 3. Singhbhum (5368/5368)

Table 3.7 Movement of Agartala based Central Police Force personnel from Segment 1 to Segment 2

Movement from West Bengal State constituencies to Tamil Nadu State constituencies (number moved/total number required)	1. Barrackpore to Chennai North (1288/4968) 2. Barrackpore to Chennai South (1876/5576) 3. Dum Dum to Mayiladuthurai (5292/5292) 4. Barasat to Chennai North (1536/4968) 5. Barasat to Chennai Central (4592/4592) 6. Jadavpur to Chidambaram (2944/5612) 7. Kolkata Dakshin to Arani (5548/5548) 8. Kolkata Uttar to Arakkonam (5388/5388) 9. Howrah to Viluppuram (5504/5504) 10. Howrah to Chidambaram (1096/5612)
Movement from West Bengal State constituencies to Orissa State constituencies (number moved/total number required)	1. Dum Dum to Bhadrak (700/6168) 2. Joynagar to Kandhamal (56/5152) 3. Joynagar to Kendrapara (336/6656) 4. Joynagar to Jagatsinghpur (5308/6844) 5. Jadavpur to Dhenkanal (3572/5500)

Table 3.7 (continued)

	6. Kolkata Dakshin to Aska (1904/5560) 7. Kolkata Uttar to Jajpur (1196/5596) 8. Jhargram to Balasore (5856/5856) 9. Purulia to Sambalpur (5604/5604) 10. Bankura to Nabarangpur (5828/5828) 11. Bishnupur to Mayurbhanj (6488/6488) 12. Burdwan Durgapur to Keonjhar (332/6584) 13. Burdwan Durgapur to Koraput (1612/6064) 14. Asansol to Keonjhar (6252/6584)
Movement from West Bengal State constituencies to Andhra Pradesh State constituencies (number moved/total number required)	1. Jhargram to Srikakulam (1160/7160) 2. Purulia to Aruku (732/6424) 3. Bankura to Aruku (924/6424) 4. Burdwan Durgapur to Kakinada (4808/5780)
Movement from Jharkhand State constituencies to Andhra Pradesh State constituencies (number moved/total number required)	1. Dumka to Aruku (396/6424) 2. Jamshedpur to Srikakulam (692/7160) 3. Jamshedpur to Kakinada (420/5780) 4. Jamshedpur to Narsapuram (5392/5392) 5. Singhbhum to Anakapalli (5368/6220)

The method assumes that there is only one set of movements from the bases, which is from the bases to constituencies where the first segments of elections are being held. Similarly there is only one set of movements from the segment where the final phases of elections are being held to the bases. There are no movements between the bases and any segment, where intermediate phases of elections are in progress.

The method can be modified to incorporate ground realities. For example, the requirement of polling personnel may vary across constituencies, depending on the perceptions of threat to maintenance of law and order. In that case, suitable data will have to be incorporated in both the models.

Table 3.8 Movement of Agartala based Central Police Force personnel from Segment 2 to Segment 3

Movement from Andhra Pradesh State constituencies to Maharashtra State constituencies (number moved/total number required)	1. Aruku to Akola (2052/6740) 2. Srikakulam to Buldhana (1852/6680) 3. Kakinada to Palghar (420/7696) 4. Narsapuram to Bhiwandi (5392/7304) 5. Kakinada to Palghar (3868/7696) 6. Kakinada to Bhiwandi (940/7304)
Movement from Andhra Pradesh State constituencies to Gujarat State constituencies (number moved/total number required)	1. Anakapalli to Bhavnagar (4272/6176) 2. Anakapalli to Rajkot (1096/6324)
Movement from Orissa State constituencies to Maharashtra State constituencies (number moved/total number required)	1. Nabarangpur to Yavatmal Washim (5828/7524) 2. Koraput to Amravati (1612/7180) 3. Bhadrak to Ramtek (700/8180)
Movement from Orissa State constituencies to Madhya Pradesh State constituencies (number moved/total number required)	1. Balasore to Rewa (5856/5808) 2. Sambalpur to Rewa (5604/5808) 3. Mayurbhanj to Shahdol (6488/6664) 4. Keonjhar to Rewa (6584/5808) 5. Kandhamal to Rewa (56/5808) 6. Kendrapara to Rewa (336/5808) 7. Jagatsinghpur to Rewa (5308/5808) 8. Dhenkanal to Rewa (3572/5808) 9. Aska to Rewa (1904/5808) 10. Jajpur to Balaghat (1196/7016)
Movement from Tamil Nadu State constituencies to Karnataka State constituencies (number moved/total number required)	1. Chennai North to Bangalore Central (1744/7068) 2. Chennai North to Bangalore South (956/6996) 3. Chennai North to Kolar (124/7560) 4. Chennai South to Bangalore North (1876/7804) 5. Mayiladuthurai to Hassan (4368/6000)

Table 3.8 (continued)

	6. Mayiladuthurai to Bangalore North (736/7804) 7. Mayiladuthurai to Shimoga (188/6820) 8. Chennai Central to Davangere (3204/6620) 9. Chennai Central to Tumkur (1232/6844) 10. Chennai Central to Bangalore Rural (156/8644) 11. Chidambaram to Tumkur (4040/6844) 12. Arani to Bangalore Rural (336/8644) 13. Arani to Chikkballapur (1788/7292) 14. Arakkonam to Bagalkot (32/6012) 15. Viluppuram to Chikkballapur (5504/7292)
Movement from Tamil Nadu State constituencies to Puducherry State constituencies (number moved/total number required)	1. Arani to Puducherry (3424/3424)
Movement from Tamil Nadu State constituencies to Goa State constituencies (number moved/total number required)	1. Arakkonam to North Goa (2716/2716) 2. Arakkonam to South Goa (2640/2640)

6 Conclusion

In this paper, a methodology is proposed and demonstrated for obtaining the optimal scheduling and logistics planning of the Indian General Elections. The method can be utilized for scheduling and planning any nationwide event requiring scarce resources.

7 Practical Exercises

1. Consider the problem discussed in this chapter with all the data remaining the same, except the total number of people in the CPF polling team which is 1 million instead of 1.5 million. Find the optimum solution of the problem with this change.
2. Consider the problem discussed in this chapter with all the data remaining the same, except that polling is required to begin on the first day with elections in the

first state in alphabetical order in Table 3.3, Andhra Pradesh. Find the optimum solution of the problem with this change.

3. Consider the problem discussed in this chapter with all the data remaining the same, except that six Central Police Force personnel are required for each polling station in the states of Bihar, Manipur, Assam, Maharashtra and Kerala. Find the optimum solution of the problem with this change.

References

1. Scharff, M. Policing election day: Vulnerability mapping in India 2006–2009. http://www.princeton.edu/successfulsocieties/content/data/policy_note/PN_id173/Policy_Note_ID173.txt. Accessed 6 Nov 2011
2. National Crime Records Bureau. (2010). *Crime in India 2009 statistics*. New Delhi: NCRB (Ministry of Home Affairs, Government of India).
3. Stefan Harrendorf, M. H. (2010). *International statistics on crime and justice*. Helsinki: European Institute for Crime Prevention & Control (HEUNI).
4. Bureau of Police Research & Development. Data on police organizations 2009. BPRD (Ministry of Home Affairs, Government of India). http://www.bprd.nic.in/index3.asp?sslid=547&subsublinkid=204&lang=1. Accessed 6 Nov 2011.
5. Election Commission of India. (2009). Lok Sabha election 2009—Reinforcing Indian democracy. New Delhi: Election Commission of India.
6. Murty, K. G. (1992). *Network Programming*. Englewood Cliffs: Prentice-Hall. http://ioe.engin.umich.edu/people/fac/books/murty/network_programming/.
7. Google. Google Maps. http://maps.google.co.in. Accessed 2 Oct 2011
8. iTouchMap.com. Latitude and longitude of a point. http://itouchmap.com/latlong.html. Accessed 2 Oct 2011
9. The Airport Authority. Every airport, everywhere. http://airport-authority.com/. Accessed 8 Nov 2011

Chapter 4
Procurement, Production and Marketing at Supply-Driven Milk and Milk Products Cooperative

Omkar D. Palsule-Desai and Katta G. Murty

1 History of Milk and Milk-Products

Milk is a nutritious liquid that is secreted by mammals to feed their young. Human beings have been consuming milk of several species of mammals such as cow, buffalo, sheep, goat, camel, reindeer, etc., as food for themselves ever since these mammals were domesticated by about 9000 BCE in various parts of the world. One of the great Indian epics, Mahabharata, whose composition is dated to around the same time, describes how Lord Krishna as a child growing up in a family of cowherds, used to love consuming the milk of cows, and milk products such as yogurt and butter. For more details on this great epic, refer [5]. By about 7000 BCE, large herds of cows, buffaloes, sheep, goats, camels, etc., were being grown for their milk, meat and skins in several parts of the world. Making cheese from milk was practiced by ancient Greeks and Romans, and the consumption of milk and milk products spread throughout the world. The ancient traveler Marco Polo visiting Mongolia in 1275 CE described in his records how Mongolians have been using powdered milk in those days.

Cattle were brought to the Americans in 1600 CE by the early European colonists, and by the eighteenth century CE, some population centers in the US had grown sufficiently to become attractive markets for large-scale dairy industry operations.

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_4) contains supplementary material, which is available to authorized users.

This chapter is based on one of the chapters of the thesis submitted by the first author at Indian Institute of Management Ahemdabad, India. The author is sincerely grateful to his thesis advisory committee, Dr. Devanath Tirupati, Dr. Pankaj Chandra and Dr. M.S. Sriram, for their guidance in conceptualizing and analyzing the problem context in the thesis originally.

O. D. Palsule-Desai (✉)
Indian Institute of Management, Indore, India
e-mail: omkardpd@iimidr.ac.in

K. G. Murty
IOE Department, University of Michigan, Ann Arbor,
MI 48109-2117, USA
e-mail: murty@umich.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_4, © Springer Science+Business Media New York 2015

Fig. 4.1 Lord Krishna and his friends stealing butter to eat. During their childhood, Lord Krishna and his friends were known for stealing butter from houses in their neighborhood. People in the neighborhood also affectionately and consciously encouraged these children to steal butter to eat. (Source: http://prabhukrishnaconsciousness.blogspot.in/2010_04_01_archive.html)

By mid-nineteenth century, condensed milk plants and companies were in operation, and the process of pasteurization of milk was discovered by Louis Pasteur in France. During the same time, glass milk bottles and jars were in use, and the process of making milk deliveries in bottles every morning also started.

Today, India is the top ranked country in the world in milk production with the annual output of approximately 125 million tonnes during 2011–2012. India's milk production accounts for approximately 18 % of the aggregate world output. The success story of India's milk production and marketing is primarily centered around the programmes such as *White Revolution* and *Operation Flood* that were based on forming cooperatives of milk producers. The aim of these programmes was to develop sustainable operations by eliminating middlemen and bringing procurement, production and marketing of milk and milk products under one umbrella for the benefits of the farmers (see [3]). Simultaneously developing both supply and demand markets was identified as a key business strategy to devise the sustainable operations (see [1]). Dr. Verghese Kurien (November 26, 1921–September 9, 2012) is credited with being the pioneer in organizing this effort. This "Milkman of India" introduced a business model, popularly known as 'Anand Pattern', that provides milk producing farmers access directly to consumers that they can individually neither afford nor manage. This business model builds a three-tiered organizational structure consisting of three kinds of players—village level cooperative societies, district level producers' cooperatives, and state level marketing agencies. For a detailed schematic representation of the organizational structure under the Anand pattern, see [2].

2 Milk and Milk Products Cooperative (MMPC) in India

This chapter is based on work carried out at a major Milk and Milk Products Cooperative (MMPC) in India over a period of several years. MMPC was one of the largest and the oldest cooperatives that were established in the middle of the twentieth century during Operation Flood. Currently, this cooperative manages the largest food products brand in India. It produces and supplies liquid milk and a variety of milk products such as butter, cheese, ice-cream, chocolates, *ghee* (clarified butter popular, particularly, in India), milk powder, etc. This cooperative is the largest exporter of dairy products in India and its major export markets are in the Middle East, countries in Africa and South East Asia.

MMPC is organized as a cooperative in the sense that its suppliers of milk are the owners (sole shareholders) of the cooperative, and it operates for the benefits of all the suppliers. During 2011–2012, the cooperative had 3.2 million producer farmers as its members, and its daily average procurement of raw milk from its suppliers was 10.6 million litres. The products made by the cooperative can be broadly classified into two categories: high return products (butter, cheese, ghee, ice-cream, chocolates), and low return products (unflavored and flavored liquid milk, milk powder). All the high return products are nonperishable and can be stored for long periods under refrigeration. Among the low return products, unflavored milk

has shelf life of only 2 days even under refrigeration, and flavored liquid milk has a storage life of 2 weeks under refrigeration. But milk powder is nonperishable and requires no refrigeration. In addition, milk powder can be reconstituted into liquid milk that can be used as a basic raw material for the cooperative any time. Liquid products such as unflavored and flavored milk are measured in litres (we will use the abbreviation "L" for "litres"); 3.78 L is equal to 1 US gallon. Solid products are all measured in terms of their weight in kilograms (kg); 1 kg is equal to 2.2 lb. The raw material requirements for each unit of an output product are non-identical; the details of which are given in the data discussed later.

The planning horizon is a year divided into two seasons: winter and summer; each approximately half a year long. Planning for the entire year is always carried out at the beginning of the winter season because the raw material supply peaks during that season, but demand has its peak in the summer season. Only the inventory of the nonperishable products can be carried from one season to the next.

Fig. 4.2 shows the structure of major activities of MMPC.

3 MMPC's Operating Policies

MMPC is supply-driven in a way that as per its by-laws the cooperative is obliged to buy all the raw milk supplied by its member farmers, and also it does not buy any milk from farmers who are not members of the cooperative. At the time of milk supply, each farmer is paid the procurement price that is declared by the cooperative for the season at the beginning of the season. Each producer farmer can choose to sell his/her produce to any buyer in the market, however, the member producers are required to sell a minimum quantity of milk annually to keep their membership active. In return, the cooperative provides a variety of benefits for betterment of the member families, e.g., medical services for members and their animals, cattle-feed on credit and at low prices, training and basic education, alternate ways to earn livelihood, etc. The cooperative cultivates loyalty of their farmers by providing them financial security through uniform and trustworthy procurement pricing for their milk supply year long.

The cooperative also cultivates customer loyalty by selling *high* quality food products at reasonable prices year-round, and ensuring that basic customer needs are met. While this cooperative competes in the market for high return products, one of its strategies is to not lose market share for any of the low return products. This is achieved through its policy of meeting minimum customer demand in each product category. Moreover, the cooperative's practice of devising customer focused strategies is reflected from the fact that it has expanded progressively and cautiously its product portfolio over the years from only liquid milk in the initial years to high return products such as ghee and ice-cream. Consumption of liquid milk daily is considered essential by domestic customers, and the cooperative tries to meet their need at all cost. For this they have established their own retail outlets at numerous locations even though it has not been observed to be a profitable proposition at many places.

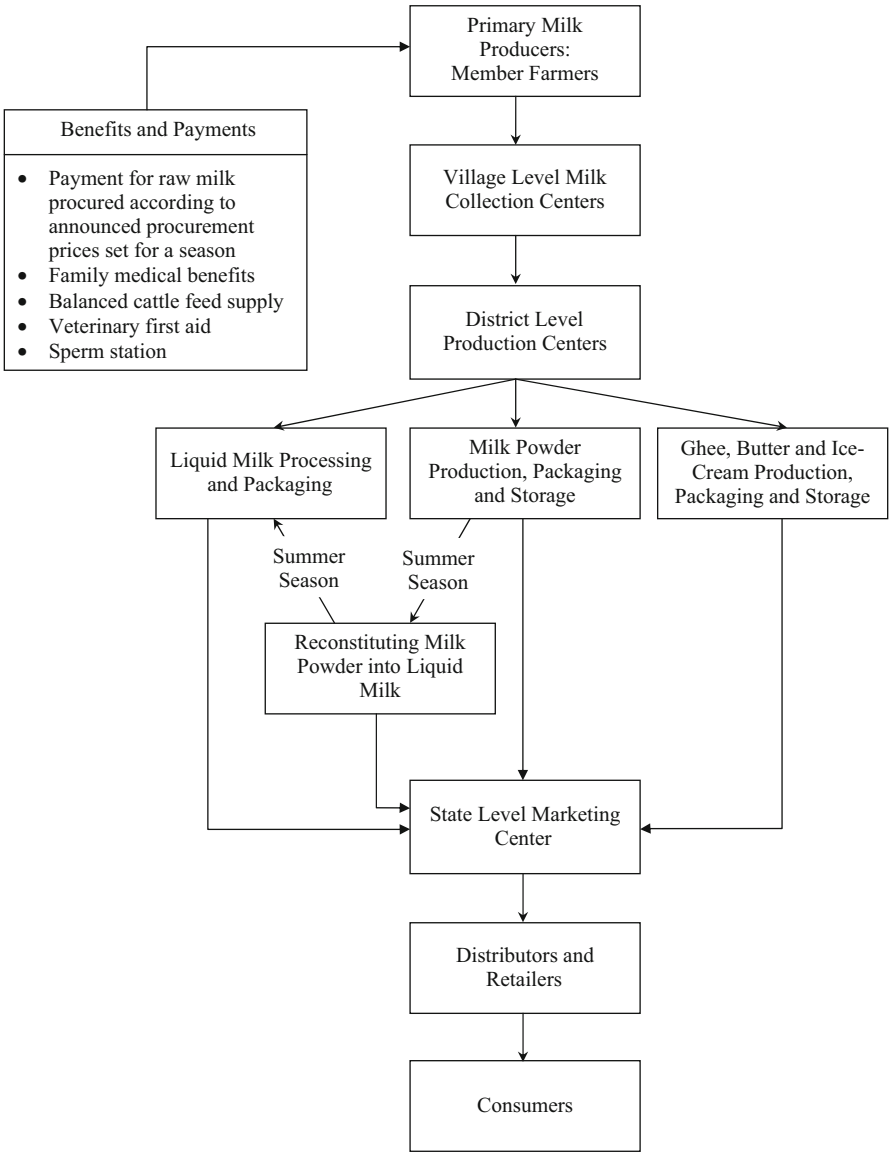


Fig. 4.2 Organizational structure and major activities in Anand Pattern. (Source: This figure is prepared on the basis of information provided in [3] and by AMUL, one of the largest milk cooperatives in India (http://www.amuldairy.com/index.php? option=com_content&view=article&id=75&Itemid=100)

Table 4.1 Procured and output products of MMPC

Product number	Product name	Profitability	Perishability	Units for measuring
0	Raw milk procured	N/A	Perishable	Litre
1	Liquid milk	Low	Perishable	Litre
2	Milk powder	Low	Nonperishable	Kilogram
3	Ghee	High	Nonperishable under refrigeration	Kilogram
4	Butter	High	Nonperishable under refrigeration	Kilogram
5	Ice-cream	High	Nonperishable under refrigeration	Kilogram

By the operational considerations, cheese—one of the major milk products—is subsumed in *Butter*

4 Procurement, Production and Marketing Decision Making at MMPC

In this chapter we will discuss the MMPC's problem for the planning year (2012) involving decision making in procurement, production and marketing of milk and milk products. In this regard, help the cooperative make aggregate plans for the year 2012. In a variety of output products that MMPC offers in the consumer market, the primary product categories that we focus on in this chapter are given in Table 4.1.

In the sequel, we use superscript k, $k = 1, 2, \ldots, 5$, to refer to the output products as numbered here, and $k = 0$ denotes raw milk procured and also measured in litres. We use subscript i, $i = 0, 1, 2, \ldots$, to denote year i, and subscript j, $j = 1, 2$, to refer to seasons in a year: $j = 1$ represents winter season and $j = 2$ denotes summer season.

The decisions to be made are: determining for each of the winter and summer seasons of the planning year

1. Procurement price of MMPC for raw milk
2. Sales price of MMPC for each of the output products in the market
3. Sales volume of each of the output products
4. Production level for each of the output products
5. Volume of milk powder to be reconstituted into liquid milk

In the sequel, *sales prices* refers to *prices in the retail market*. Similarly, *national average sales price* for liquid milk refers to average price in the retail market in the country. Also, *sales volumes* refers to those in the retail market for all output products.

In order to help the cooperative make these decisions, data are provided as follows[1]. The data for procurement, production and sales volumes, and production capacities, etc. are on seasonal basis.

[1] Data are compiled based on the information provided by the professional database agencies such as Centre for Monitoring Indian Economy Pvt. Ltd. (CMIE) (http://www.cmie.com/), the annual reports of well-known cooperatives in India, and the official reports published on the Indian Dairy Industry (http://www.indiadairy.com).

Table 4.2 Trend in milk procurement at MMPC and sales prices of liquid milk in India

Year	Season	MMPC procurement volume (million L)	MMPC procurement price (Rs./L)	National average sales price for liquid milk[a] (Rs./L)
2001	1	967.20	5.76	18.67
	2	644.80	7.37	18.67
2002	1	1038.50	5.92	19.32
	2	636.50	7.28	19.32
2003	1	1020.80	5.76	20.14
	2	835.20	7.35	20.14
2004	1	1053.28	5.96	21.06
	2	762.72	7.19	21.06
2005	1	1314.18	6.77	22.45
	2	771.82	8.15	22.45
2006	1	1369.80	6.75	24.11
	2	913.20	8.42	24.11
2007	1	1407.15	7.44	27.96
	2	977.85	9.69	27.96
2008	1	1564.84	8.95	29.08
	2	1133.16	11.31	29.08
2009	1	1957.76	9.49	29.83
	2	1101.24	12.10	29.83
2010	1	2154.10	10.85	31.24
	2	1159.90	13.55	31.24
2011	1	2133.81	12.88	33.57
	2	1253.19	15.91	33.57

[a]National average sales price for liquid milk—an output product in the consumer market—specifies the average of sales prices for liquid milk supplied by many milk and milk products producing companies in India. The Indian dairy industry is governed under a provision of the Essential Commodity Act, 1955, and the national average sales prices of liquid milk reflect economic implications of inflation for consumption of milk and milk products in India

Table 4.2 gives the data on the amount of raw milk (in litres) procured, procurement price (in Rs. per litre) of MMPC, and national average sales price of liquid milk—one of the output products produced from raw milk—in the consumer market since 2001. (Rs. refers to Indian currency: Rupee. The average currency exchange rate in the international market during January and June 2012 was US$ 1 equal to Rs. 52, or equivalently Rs. 1 equal to US$ 0.01922.) There are several milk and milk products producing companies in India. They all have the practice of keeping their sales prices for liquid milk fixed during each year, i.e., changing them only at the beginning of the winter season when plans for the coming year are finalized. What is given in column 5 of Table 4.2 is the sales price of liquid milk each year since 2001 averaged over all these companies.

Tables 4.3 and 4.4 give the data reflecting the trend in sales of the output products of MMPC. From Tables 4.2 and 4.3, we observe that over the years sales prices for liquid milk of MMPC have been *reasonably* higher than the average of sales prices of liquid milk in the country. MMPC has adopted the same pricing strategy for all of its output products. The main drivers of this strategy of charging reasonable

Table 4.3 Trend in sales of liquid milk of MMPC

Year	Sales volume (million L)	Sales price (Rs./L)
2001	431.97	23
2002	457.24	23
2003	491.99	24
2004	504.18	24
2005	535.70	25
2006	579.58	27
2007	601.45	31
2008	722.56	32
2009	915.14	33
2010	982.86	35
2011	1005.13	38

The sales of liquid milk of MMPC are typically identical in both winter and summer seasons in a year, and they can be considered to be equal in analysis

premium vis-à-vis national average sales prices are threefold: product quality, brand image and the role of state sponsored companies in the country. To explain further - MMPC has been able to extract price premiums from its customers by supplying high quality products over the years. This has made MMPC's brand one of largest and the highest valued food products brand in the country. Moreover, the national average sales prices for liquid milk include sales prices of many state sponsored milk and milk products producing companies in the country. These companies, in particular, supply their products that are affordable to all segments of the market.

Decision Making Problem at MMPC

As a part of its strategy to sell the products at reasonable prices to win customer loyalty, the cooperative sets sales prices for its products within a predetermined sales price range. The bounds for the sales prices for the output products are chosen by considering the sales prices in the preceding year and the expected sales prices of the competitors in the planning year. Table 4.5 gives the desired range (i.e., lower and upper bounds) for the sales prices of each of its output products for each season of the planning year. The table also provides data on the cooperative's target sales levels for each of the products for each season. The cooperative requires that the actual sales performance for a product is above the target set for it.

MMPC also wins member farmers' loyalty by ensuring that the seasonal procurement prices for raw milk are comparable from year to year. The cooperative sets the procurement prices for the planning year within prescribed procurement price bounds that are determined on the basis of procurement prices offered in the preceding year and the expected procurement prices of the competitors in the planning year. Table 4.5 gives the desired range for the procurement prices in the year 2012.

Table 4.4 Trend in sales of output products of MMPC

Year	Season	Milk powder sales volume (million kg)	Milk powder sales price (Rs./kg)	Ghee sales volume (million kg)	Ghee sales price (Rs./kg)	Butter sales volume (million kg)	Butter sales price (Rs. per kg)	Ice-cream sales volume (million kg)	Ice-cream sales price (Rs./kg)
2001	1	96.75	105.30	1.08	209.00	4.85	126.50	4.26	49.48
	2	64.50	134.55	0.72	161.50	3.23	97.75	5.43	59.54
2002	1	101.91	112.24	1.28	218.00	5.41	137.34	4.86	55.64
	2	62.46	137.92	0.78	170.63	3.32	107.50	6.77	63.96
2003	1	106.11	113.03	1.44	231.00	5.14	147.40	7.43	57.89
	2	76.84	146.29	1.18	184.33	4.20	117.62	7.58	70.62
2004	1	103.63	126.90	1.78	232.20	6.45	155.52	7.69	63.59
	2	66.26	147.67	1.29	191.25	4.67	128.09	8.96	73.87
2005	1	127.18	131.13	2.35	239.80	8.06	156.96	7.46	64.28
	2	74.70	157.81	1.38	186.29	4.73	121.93	10.86	72.44
2006	1	131.57	134.68	2.78	259.20	8.59	177.12	9.51	71.64
	2	87.71	167.98	1.85	211.20	5.73	144.32	12.11	84.29
2007	1	131.77	138.84	3.50	267.50	9.55	186.18	11.60	76.00
	2	91.57	180.69	2.43	224.82	6.64	156.47	14.12	89.70
2008	1	140.15	145.80	4.54	280.90	10.79	195.04	14.50	79.49
	2	101.49	184.37	3.29	243.04	7.82	168.75	16.88	95.90
2009	1	162.81	153.79	6.51	296.80	13.94	237.44	14.42	97.84
	2	91.58	196.04	3.66	250.13	7.84	200.11	21.99	113.86
2010	1	172.24	160.08	8.53	318.00	16.98	258.64	17.10	107.75
	2	92.75	199.85	4.60	266.57	9.15	216.81	27.32	122.99
2011	1	157.28	167.44	10.84	348.80	20.41	320.46	20.55	115.00
	2	92.37	206.79	6.36	270.96	11.99	248.95	29.95	131.86

Table 4.5 Sales price ranges and target sales levels for output products and range for procurement prices for raw milk in the planning year of MMPC

Product number	Product	Season	Minimum price (Rs./unit[a])	Maximum price (Rs./unit)	Target sales (million units)
0	Raw milk	1	12.50	13.00	N/A
		2	15.50	16.00	N/A
1	Liquid milk	1	39	42	525
		2	39	42	525
2	Milk powder	1	170	175	65
		2	205	207.5	40
3	Ghee	1	350	355	13
		2	270	275	8
4	Butter	1	315	325	22
		2	245	252.50	13.50
5	Ice-cream	1	115	117.50	22
		2	130	132.50	32.50

[a]For raw milk and liquid milk the unit adopted is litres (L) and for other products it is kilogram (kg)

Table 4.6 Capacity, cost, and raw material requirements for output products of MMPC

Product number	Output product	Production capacity[a] (million units[b])	Production cost (Rs./unit)	Inventory holding cost[c] (Rs./unit)	Raw milk required (L/unit)
1	Liquid milk	600	2.30	N/A	1
2	Milk powder	100	11.72	1.41	6.40
3	Ghee	25	32.20	3.86	14.80
4	Butter	27.50	26.45	3.17	10.50
5	Ice-cream	27.50	17.60	2.11	3.75

The cost of reconstituting milk powder into liquid milk is Rs. 1/kg. The expected national average sales price for liquid milk in India in 2012 is Rs. 37/L.
[a] Production capacity is in number of production units per season
[b] For liquid milk the unit adopted is litres (L) and for other products it is kilogram (kg)
[c] Inventory holding cost is per season

In the decision making problem at MMPC, the above described constraints in view of both member farmers and consumers are to be satisfied.

Table 4.6 gives the data on the cooperative's production capacities (in units per season) and cost parameters - production cost, inventory holding cost from one season to the next - for each of the output products during the planning year. The costs and capacity parameters are the same for both seasons of the planning year. The table also provides information on the litres of raw milk required to produce one unit of the output product.

The production costs mentioned in Table 4.6 above include various components such as cost of raw materials other than raw milk, labor costs, machine costs, packaging costs, etc.

The objective of MMPC is to maximize annual net profit (total sales revenue - raw material procurement cost - production cost - inventory holding cost - cost of reconstituting milk powder into liquid milk).

5 Approach for Constructing an Appropriate Mathematical Model for the Decision Making Problem at MMPC

The most important set of decision variables in this problem are the procurement prices for raw milk and sales prices for the output products in each season of the planning year. The annual sales revenue depends on sales prices and sales volumes, while the raw material procurement cost depends on procurement prices and consequent procurement volumes. In order to express the objective function of annual net profit of the cooperative as a function of these primary decision variables, we need to establish relationships between procurement volumes and procurement prices, and sales volumes and sales prices using the data provided.

One of the critical features of quantitative analysis is to adopt available data to gather relevant and accurate information that can be used to obtain further insights in the situation under study. In this regard, the first essential task in constructing a mathematical model for the MMPC's problem is to generate expressions for procurement and sales volumes as functions of procurement and sales prices, respectively, for each season of each year. We describe approaches for findings these functions in the following subsections.

5.1 Generating Functional Relationship Between Procurement Prices and Procurement Volumes of Raw Milk

Farmers who are members of the cooperative are required to sell specified minimum amount of raw milk to MMPC to retain their membership in the cooperative. However, they have the freedom to sell their raw milk produce beyond the minimum quantity to any buyer including MMPC. Clearly, the farmers would act in a way to maximize what they get from selling their raw milk produce. Let

$$\pi_{ij}^{0} = \text{Quantity of raw milk } (k = 0) \text{ procured by MMPC in season } j \text{ in year } i,$$
$$i = 0, 1, 2, \ldots, 10;\ j = 1, 2 \tag{4.1}$$

Consider the ratio for particular season j of year i:

$$\rho_{ij}^{0} = \frac{\text{Procurement price set by MMPC}}{\text{National average sales price of liquid milk}}, \quad i = 0, 1, 2, \ldots, 10;\ j = 1, 2 \tag{4.2}$$

If $\rho_{ij}^0 > 1$, the member farmers have no incentive to sell their raw milk produce to anyone other than MMPC. Even if the ratio is less than, but close to 1, out of their loyalty to the cooperative, and considering the other benefits the cooperative provides to them (such as medical facilities, education, etc. - see Fig. 4.2), they may still opt to sell most of their raw milk produce to MMPC. We can get an indication of how much of raw milk produce the farmers will sell to MMPC using how they behaved in this respect in the past. For this the data on milk procurement provided in Table 4.2 earlier can be used.

All prices are affected by year to year inflation. Because of the inflation, the numerical values of prices in two different years are not comparable. The issue is particularly important in India as the average inflation rate in India has increased from 3.671 % in 2001 to 8.87 % in 2011 with a peak of 12.11 % in 2010. However, the assumption that inflation will affect the procurement prices set by the company and the national average sales prices of liquid milk equally is quite reasonable. Under this assumption, the ratio ρ_{ij}^0 defined in (4.2) above is independent of year to year inflation. Also, considering the aspects of farmers' behavior discussed above, it is reasonable to assume that the volume of raw milk procurement by MMPC in any year depends on the ratio ρ_{ij}^0 defined in (4.2) above.

The data that we will use to express the procurement volume as a function of the procurement price set by the cooperative and the national average sales prices of liquid milk are given in Table 4.7. Using the data derived in the table, for each i and j we will fit π_{ij}^0 as a linear function of ρ_{ij}^0. In particular, we consider the model function $\pi_{ij}^0 = a_j^0 + b_j^0 \rho_{ij}^0 + g_j^0 \pi_{(i-1)j}^0$, for all i, j, where a_j^0, b_j^0 are parameters corresponding to seasons (winter and summer) to be estimated from the data. Parameter g_j^0 represents the growth in volume of raw milk procured from one year to the next across the seasons. Accordingly, this model assumes that volume of raw milk procured in a season can be predicted using the volume of raw milk procured in the same season in the preceding year. The assumption is reasonable since MMPC accepts raw milk supply only from its member farmers, and the individual farmer level information is well captured in the Anand Pattern (see [3]).

We use the least sum of absolute deviation technique - L_1-technique - (see [4]) for estimating the coefficients a_j^0, b_j^0 and g_j^0. It may be noted that this technique leads to the following linear programming model for estimating a_j^0, b_j^0 and g_j^0 in the decision making problem at MMPC:

$$\text{Minimize} \sum_{i=1}^{10} \sum_{j=1}^{2} \left(u_{ij}^0 + v_{ij}^0\right) \tag{4.3}$$

Subject to

$$\pi_{ij}^0 - a_j^0 - b_j^0 \rho_{ij}^0 - g_j^0 \pi_{(i-1)j}^0 + u_{ij}^0 - v_{ij}^0 = 0, \quad i = 1, 2, \ldots, 10;\ j = 1, 2 \tag{4.4}$$

$$u_{ij}^0, v_{ij}^0 \geq 0, \quad i = 1, 2, \ldots, 10;\ j = 1, 2 \tag{4.5}$$

In the model (4.3)–(4.5) above while the decision variables a_j^0, b_j^0, g_j^0 for $j = 1, 2$ are the parameters being estimated, the decision variables u_{ij}^0, v_{ij}^0 are the deviation

Table 4.7 Trend in milk procurement at MMPC

Year	i	Season j	MMPC procurement volume π_{ij}^0 (million L)	MMPC procurement price (Rs./L)	National average sales price for liquid milk (Rs./L)	ρ_{ij}^0
2001	0	1	967.20	5.76	18.67	0.31
		2	644.80	7.37	18.67	0.31
2002	1	1	1038.50	5.92	19.32	0.31
		2	636.50	7.28	19.32	0.31
2003	2	1	1020.80	5.76	20.14	0.29
		2	835.20	7.35	20.14	0.29
2004	3	1	1053.28	5.96	21.06	0.28
		2	762.72	7.19	21.06	0.28
2005	4	1	1314.18	6.77	22.45	0.30
		2	771.82	8.15	22.45	0.30
2006	5	1	1369.80	6.75	24.11	0.28
		2	913.20	8.42	24.11	0.28
2007	6	1	1407.15	7.44	27.96	0.27
		2	977.85	9.69	27.96	0.27
2008	7	1	1564.84	8.95	29.08	0.31
		2	1133.16	11.31	29.08	0.31
2009	8	1	1957.76	9.49	29.83	0.32
		2	1101.24	12.10	29.83	0.32
2010	9	1	2,154.10	10.85	31.24	0.35
		2	1159.90	13.55	31.24	0.35
2011	10	1	2133.81	12.88	33.57	0.38
		2	1253.19	15.91	33.57	0.38

variables. All other symbols π_{ij}^0, ρ_{ij}^0 are data elements in the model. Solving the LP leads to the estimates given in Table 4.8. We use Excel Solver in MS Office 2007 to estimate our model parameters. While estimating the model parameters - a_j^0, b_j^0 and g_j^0 - we assumed 2001 as the base year, i.e., $i = 0$. It may be noted that $a_j^0 = 0$ for $j = 1, 2$, and hence, the functional relationship between procurement price and procurement volume in a season of a particular year can be given as $\pi_{ij}^0 = b_j^0 \rho_{ij}^0 + g_j^0 \pi_{(i-1)j}^0$.

It may be noted from Table 4.8 above that $a_j^k \geq 0$ for $j = 1, 2$ and $k = 1, 2, \ldots, 5$. This is consistent with the reality that supply of raw milk from the member farmers and demand for the output products in any season are non-negative for any level of (procurement or sales) prices set by MMPC. While we did not explicitly include the constraint $a_j^k \geq 0$ for $j = 1, 2$ and $k = 1, 2, \ldots, 5$ in the model (4.3)–(4.5) given above, we considered it while solving the model in Excel Solver. Similarly the facts $a_j^k \geq 0, g_j^k \geq 0$ for $j = 1, 2, k = 0, 1, \ldots, 5$, and $b_j^k \leq 0$ for $j = 1, 2, k = 1, 2, \ldots, 5$, and $b_j^k \geq 0$ for $j = 1, 2, k = 0$ are consistent with our expectation based on reality.

Table 4.8 Parameter estimates

Product number k	Output product	Season j	a_j^k	b_j^k	g_j^k
0	Raw milk procured	1	0	301.99	0.98
		2	0	1444.60	0.48
1	Liquid milk	1	100.71	−85.42	1.06
		2	100.71	−85.42	1.06
2	Milk powder	1	57.34	−7.63	0.92
		2	196.81	−19.50	0.23
3	Ghee	1	1.30	−0.12	1.27
		2	2.26	−0.27	1.27
4	Butter	1	0	−0.10	1.26
		2	0	−0.09	1.25
5	Ice-cream	1	2.12	−0.91	1.26
		2	0	−0.06	1.21

The estimated parameters are such that $a_j^k \geq 0$ for $j = 1, 2$ and $k = 0, 1, \ldots, 5$

5.2 *Generating Functional Relationship Between Sales Prices and Sales Volumes of Output Products*

In parallel with generating functional relationship between procurement volumes and procurement prices for raw milk, in this section we generate functional relationship between sales prices and sales volumes for each of the output products in each season of a particular year.

India's population has been growing at approximately 1.6 percent per year since 2001. Due to population growth, demand for milk and milk products is expected to have a corresponding annual growth rate. Data given in Tables 4.3 and 4.4 show evidence of this clearly; this will also reflect in the sales volumes of milk and milk products of MMPC. As sales price increases, it is reasonable to expect sales volume to decrease. There arguments suggest the following model formulation for liquid milk ($k = 1$):

$$\pi_{ij}^1 = a_j^1 + b_j^1 \rho_{ij}^1 + g_j^1 \pi_{(i-1)j}^1, \quad i = 1, 2, \ldots, 10;\ j = 1, 2 \tag{4.6}$$

where, g_j^1 captures the fractional growth in the sales volume of liquid milk from one year to the next across the seasons. As mentioned earlier, the demand for liquid milk is constant throughout the year, and hence, it may be considered identical in both seasons. Thereby, we would expect estimates of the parameters - a_j^1, b_j^1, g_j^1, $j = 1, 2$ - for both winter and summer seasons in (4.6) above to be equal, i.e., $a_1^1 = a_2^1, b_1^1 = b_2^1, g_1^1 = g_2^1$.

For each of the other output products, $k = 2, 3, \ldots, 5$, the model function is

$$\pi_{ij}^k = a_j^k + b_j^k \rho_{ij}^k + g^k \pi_{(i-1)j}^k, \quad i = 1, 2, \ldots, 10;\ j = 1, 2 \tag{4.7}$$

The parameters above $\left(a_j^1, b_j^1, g_j^1\right)$ for liquid milk and $\left(a_j^k, b_j^k, g_j^k\right)$, $j = 1, 2$ and $k = 2, 3, \ldots, 5$, can be estimated by setting up L_1-technique problems as described

Table 4.9 Notation

Parameter	Description
j	Index for season; $j = 1, 2$
k	Index for product; $k = 0, 1, 2, \ldots 5$
	Data Elements
α^k	Litres of raw milk required for each unit[a] of output product k
β_j^k	Target sales levels of output product k in season j
$\underline{\gamma}_j^k$	Minimum price[b] for product k in season j
$\bar{\gamma}_j^k$	Maximum price[c] for product k in season j
χ^k	Production capacity of output product k in season j
ξ	Expected national average sales prices of liquid milk in the planning year
η^k	Unit production cost of output product k
λ_j^k	Unit cost of holding inventory of output product k at the end of season j
μ_2^2	Unit cost of reconstituting milk powder into liquid milk in summer season
	Decision Variables
y_{1j}^k	Sales price of output product k in season j
y_{2j}^0	Unit procurement price of raw milk in season j
x_{1j}^k	Units of output product k to be produced in season j
x_{2j}^k	Sales volume of output product k in season j
x_{3j}^k	Volume of output product k to be inventoried in season j
x_{42}^2	Quantity of milk powder to be reconstituted in liquid milk in summer season
x_{5j}^0	Units of raw milk allocated to liquid milk in season j

All decision variables have notation using x or y, and all other symbols denote either data elements or estimates of functional relationships obtained in the earlier sections

The values of parameters $\alpha^k, \chi^k, \eta^k, \xi$ for $k = 1, 2, \ldots, 5$ are identical for both summer and winter seasons, i.e., $j = 1, 2$, and hence, the season specific subscript j is dropped from these parameters.

[a] For raw milk and liquid milk the unit adopted is litres (L) and for other products it is kilogram (kg)

[b,c] For $k = 0$, price refers to procurement price, and for $k = 1, 2, \ldots, 5$, it refers to sales price

in the case of raw milk ($k = 0$) earlier. The parameter estimates obtained are given in Table 4.8.

5.3 *Model for Optimizing the Decision Making Problem of MMPC*

We assume that the procurement volume of raw milk and sales volumes of output products are equal to the functions in terms of procurement price of raw milk and sales prices of output products, respectively, obtained for them above. In this section we develop the optimization problem that characterizes the problem context discussed above. The complete notation used in this model is given in Table 4.9.

The decision making problem of MMPC that involves developing aggregate procurement, production and marketing plan for the planning year is as follows. (In our model, inventories for the output products at the beginning of the winter season of

the planning year are assumed to be zero.)

$$\text{Maximize} \sum_{j=1}^{2} \sum_{k=1}^{5} y_{1j}^{k} x_{2j}^{k} - \eta^{k} x_{1j}^{k} - \lambda_{j}^{k} x_{3j}^{k} - y_{2j}^{0} \left(a_{j}^{0} + b_{j}^{0} y_{2j}^{0} / \xi + g_{j}^{0} \pi_{10j}^{0} \right) - \mu_{2}^{2} x_{42}^{2} \tag{4.8}$$

Subject to

$$\underline{\gamma}_{j}^{0} \leq y_{2j}^{0} \leq \bar{\gamma}_{j}^{0}, \quad j = 1, 2 \tag{4.9}$$

$$\underline{\gamma}_{j}^{k} \leq y_{1j}^{k} \leq \bar{\gamma}_{j}^{k}, \quad j = 1, 2; \quad k = 1, 2, \ldots, 5 \tag{4.10}$$

$$y_{11}^{1} = y_{12}^{1} \tag{4.11}$$

$$x_{2j}^{k} \leq a_{j}^{k} + b_{j}^{k} y_{1j}^{k} / \xi + g_{j}^{k} \pi_{10j}^{k}, \quad j = 1, 2; \quad k = 1, 2, \ldots, 5 \tag{4.12}$$

$$x_{2j}^{k} \geq \beta_{j}^{k}, \quad j = 1, 2; \quad k = 1, 2, \ldots, 5 \tag{4.13}$$

$$x_{1j}^{k} \leq \chi^{k}, \quad j = 1, 2; \quad k = 1, 2, \ldots, 5 \tag{4.14}$$

$$\alpha^{1} x_{11}^{1} \leq x_{51}^{0} \tag{4.15}$$

$$\alpha^{1} x_{12}^{1} \leq x_{52}^{0} + \alpha^{2} x_{42}^{2} \tag{4.16}$$

$$x_{2j}^{1} \leq x_{1j}^{1}, \quad j = 1, 2 \tag{4.17}$$

$$\sum_{k=2}^{5} \alpha^{k} x_{1j}^{k} \leq a_{j}^{0} + b_{j}^{0} y_{2j}^{0} / \xi + g_{j}^{0} \pi_{10j}^{0} - x_{5j}^{0}, \quad j = 1, 2 \tag{4.18}$$

$$x_{5j}^{0} \leq a_{j}^{0} + b_{j}^{0} y_{2j}^{0} / \xi + g_{j}^{0} \pi_{10j}^{0}, \quad j = 1, 2 \tag{4.19}$$

$$x_{3j}^{k} = x_{3(j-1)}^{k} + x_{1j}^{k} - x_{2j}^{k}, \quad j = 1, 2; \quad k = 3, 4, 5 \tag{4.20}$$

$$x_{31}^{2} = x_{11}^{2} - x_{21}^{2} \tag{4.21}$$

$$x_{32}^{2} = x_{31}^{2} + x_{12}^{2} - x_{22}^{2} - x_{42}^{2} \tag{4.22}$$

$$x_{42}^{2} \leq x_{31}^{2} \tag{4.23}$$

$$y_{1j}^{k}, y_{2j}^{0}, x_{1j}^{k}, x_{2j}^{k}, x_{3j}^{k}, x_{42}^{2}, x_{5j}^{0} \geq 0, \quad j = 1, 2; \quad k = 1, 2, \ldots, 5 \tag{4.24}$$

As mentioned earlier, the objective function of the optimization problem describes the profit function (in Rs.) of MMPC in the planning year. The problem constraints can be described as follows:

- Constraints (4.9) and (4.10): ranges for procurement prices for raw milk and sales prices for the output products, respectively.
- Constraint (4.11): sales price of liquid milk is constant throughout the year.
- Constraint (4.12): sales volume of an output product cannot be more than the demand for the product in a season.
- Constraint (4.13): sales is not less than the minimum target sales levels set by the cooperative.

Table 4.10 Optimal procurement plan

Product	Price (Rs./L)		Volume (million L)	
	Winter	Summer	Winter	Summer
Raw milk	12.50	15.50	2188.82	1212.22

- Constraint (4.14): production volume of any output product is constrained by the production capacity available in a season.
- Constraint (4.15): the quantity of liquid milk produced in the winter season cannot be more than the raw milk allocated (x_{51}^0).
- Constraint (4.16): the quantity of liquid milk produced in the summer season cannot be more than the sum of raw milk allocated (x_{52}^0) and reconstituted liquid milk.
- Constraint (4.17): since liquid milk cannot be inventoried, its production volume of in any season cannot be less than its sales quantity.
- Constraint (4.18): the volumes of the output products other than liquid milk produced cannot be more than the raw milk available in a season after allocating a portion of the raw milk for liquid milk.
- Constraint (4.19): the quantity of raw milk allocated for liquid milk cannot be more than the volume of raw milk procured in a season.
- Constraint (4.20): inventory balance constraint for the output products - ghee, butter and ice-cream (assuming that $x_{30}^k = 0, k = 1, 2, \ldots, 5$).
- Constraints (4.21) and (4.22): inventory balance constraints for milk powder in the winter and summer season, respectively.
- Constraint (4.23): the quantity of milk powder reconstituted into liquid milk in the summer season cannot be more than the inventory of milk powder carried forward from the winter season.
- Constraint (4.24): all decision variables are non-negative.

6 Optimal Solution

It can be observed in the optimization problem of MMPC that the objective function is quadratic in nature (due to the presence of terms $y_{2j}^0 \left(a_j^0 + b_j^0 y_{2j}^0/\xi + g_j^0 \pi_{10j}^0\right)$ and $y_{1j}^k x_{2j}^k$), and the constraints are linear. Thereby, the problem is a quadratic program that we solve using Excel Solver in MS Office 2007. In Tables 4.10, 4.11, 4.12, and 4.13 we provide the optimal solution to the decision making problem at MMPC for the planning year 2012.

Table 4.11 Optimal plan for obtaining raw milk for liquid milk production

Allocated from procurement (million L)		Milk powder reconstituted (million kg)	
Winter	Summer	Winter	Summer
538.00	369.33	N/A	26.35

Table 4.12 Optimal production and sales plan for output products

Product	Sales price (Rs./unit[a])		Production volume (million units)		Sales volume (million units)		Inventory (million units)	
	Winter	Summer	Winter	Summer	Winter	Summer	Winter	Summer
Liquid milk	42.00	42.00	538.00	538.00	538.00	538.00	0	0
Milk powder	175.00	207.50	100.00	99.67	65.00	108.32	35.00	0
Ghee	355.00	275.00	22.17	0.00	13.85	8.32	8.32	0
Butter	325.00	252.50	27.50	9.70	23.70	13.50	3.80	0
Ice-cream	117.50	132.50	27.50	27.50	22.00	33.00	5.50	0

[a]For liquid milk the unit adopted is litres (L) and for other products it is kilogram (kg)

Table 4.13 Optimal profit

Winter season (Rs. million)	Summer season (Rs. million)	Aggregate (Rs. million)
17,377.05	33,178.67	50,555.72

6.1 Discussion on the Optimal Solution to the Decision Making Problem at MMPC

From Table 4.12, we observe that the optimal plan for MMPC requires the cooperative to completely utilize its production capacity in the winter season for three of the output products - milk powder, butter and ice-cream, while production capacity for liquid milk and ghee remains underutilized. The production capacity in the summer season is underutilized for all categories of the output products. It is not surprising to note that there are no inventories of the output products remaining at the end of the summer season. However, due to limited supply of raw milk in the summer season, the cooperative inventories all nonperishable output products in the winter season. Accordingly, the entire volume of raw milk procured in each season is utilized, and the entire volume of the output products is sold in the consumer market by the end of the summer season. This leads to zero inventory of any of the output products for the cooperative at the beginning of the winter season in the next planning year. It may be observed that the optimal sales prices for the output products are such that they extract as much consumer surplus (defined as the difference between the maximum price a consumer can pay and the actual price charged, so "extracting as much consumer surplus as possible" essentially reflects the firm's ability to increase its profit by charging higher prices while keeping in mind the consequent decrease in demand) as possible while ensuring that sales volumes are not more than the demand for the products in the market. Moreover, the sales volumes for the output products, except liquid milk, are strictly less than the consumer demand in the winter season.

Similarly, the sales volume for milk powder, butter and ice-cream is strictly less than the demand for the products in the summer season. It is also interesting to note that the cooperative ensures the supply of milk in the summer season by exactly meeting its minimum sales target for milk powder in the winter season and inventorying the excess production of milk powder that can reconstituted into liquid milk in the summer season as desired. From Tables 4.11 and 4.12, we note that the volume of raw milk allocated to produce liquid milk in the winter season is exactly equal to the sales level of the output product. However, in the summer season the cooperative allocates less raw milk than the sales level of liquid milk; the remaining volume of liquid milk is obtained from reconstituting milk powder carried forward from the winter season.

6.2 Estimating True Profits in Winter and Summer Seasons Considering Ability to Reconstitute Milk Powder into Liquid Milk

Recall that milk powder, which is one of the output products that are sold in the consumer market, can be inventoried from one season to the next. Milk powder can also be reconstituted through *reverse* production process into liquid milk that can be sold in the consumer market. However, no other output product offers such reconstitution alternative to obtain liquid milk. In this regard, the entire process of first producing milk powder from raw milk procured from the farmers, inventorying milk powder and later reconstituting it into liquid milk can be viewed as inventorying liquid milk that is perishable originally as a nonperishable product by incurring additional costs.

We know that the cooperative's policy is to fix the sales prices of each of the output products 1–5 for the entire year and the sales volumes depend on how the customer reacts to these prices. Products 2–5 are nonperishable and can be made any time during the year, stored in inventory and sold any time later in the year. However, product 1 (liquid milk) is perishable and can only be stored for a few days. Moreover, the supply of raw milk is higher in the winter season than their demand, while their supply is lower than the demand in the summer season. So to meet the demand in the summer season, the cooperative uses milk powder made in the winter season in the year or earlier, and reconstitutes it into liquid milk. From Table 4.6, we see in the planning year 2012 that MMPC incurs the unit cost of Rs. 14.13/kg. for production and inventorying of milk powder in the winter season and reconstituting it into liquid milk in the summer season purely for the sake of meeting the demand in the summer season. It may be noted that we deliberately ignore differences in milk procurement prices in both winter and summer seasons as these pricing decisions are not based on inventorying and reconstituting abilities of milk powder. Instead they are driven by the farmer oriented uniform procurement pricing strategy of the cooperative. Therefore, for getting the true profit in the winter season, we need to add the total cost of production, inventorying and reconstitution of milk powder,

which is equal to Rs. 372.33 million, to the profit in the winter season determined by the optimal solution, i.e., Rs. 17,377.05 million. Similarly, the true profit in the summer season is obtained by subtracting this cost from the profit in the summer season determined by the optimal solution, i.e., Rs. 33,178.67 million. Accordingly, the true profit for MMPC in the winter season is Rs. 17,749.38 million and that in the summer season is Rs. 32,806.34 million. This approach to obtain true profit in each season of the planning year does not alter the aggregate profit in the planning year.

7 Exercises

1. In this chapter we developed a mathematical model for MMPC - a cooperative whose operating policies demonstrate concerns for social welfare. In particular, we considered that the cooperative won loyalty of its member farmers by offering uniform and reasonable procurement prices for their produce throughout the year. Similarly, it won customer loyalty by supplying high quality products for reasonable and competitive prices, and also by ensuring that minimum demand in each season is met. Assuming that MMPC was a private firm without much concerns for farmers and/or consumers, develop an aggregate plan involving procurement, production and marketing decisions for the planning year, and reflect on the firm's profitability. You may incorporate the following changes in the model presented in this chapter.
   - Ignore the target (minimum) sales levels for the output products in both winter and summer seasons. (Maintain the ranges for the sales prices for the output products as they are based on competitive sales prices in the market.)
   - Ignore the minimum procurement prices for raw milk supplied by the member farmers in each season.
   - Consider both changes mentioned above together, and discuss on the private firm's optimal plan for the planning year.
2. In the recent years, MMPC has been consciously making efforts to create awareness among consumers for benefits of consumption of milk and milk products produced from *fresh* raw milk vis-à-vis products produced from reprocessed milk powder. In this regard, the cooperative has adopted a strategy of not selling more than 35 % of raw milk procured during the year in the form of milk powder. What are the implications of this strategy on the cooperative's profitability? What is the impact on the optimal solution if MMPC limits volume of raw milk that can be sold in the form of milk powder to 30 % during the year?
3. Refer to the optimal solution to the decision making problem at MMPC presented in this chapter. In view of the criticality of milk powder in matching supply of raw milk with demand for milk and milk products across the seasons, MMPC is interested in procuring additional capacity in any season to produce milk powder at a vendor. How much should the cooperative pay for each unit of capacity in

the winter season? Is the price same in the summer season too? If yes, then why; if not, then why?

4. Instead of milk powder, if MMPC is interested in buying production capacity for ghee, the output product with the highest value in the portfolio, how much should it be ready to pay for in both winter and summer seasons? What are the implications of each additional unit of capacity for the cooperative's profitability?
5. If the cooperative gets an opportunity to sell one more unit of liquid milk in the winter season, how much would its profit increase or decrease?
6. Would you expect the implications for the cooperative's profitability of selling one more unit of ghee in the winter season and in the summer season to be identical? Comment on the same in the case of butter.
7. Will the cooperative's profit increase or decrease, and how much, if it increases the sales prices of liquid milk in both winter and summer seasons by unit each?

References

1. Chandra, P., & Tirupati, D. (2003). *Business strategies for managing complex supply chains in large emerging economies: The story of AMUL*. Working paper, W.P. NO. 2002-05-06. Indian Institute of Management Ahmedabad.
2. Goldberg, R. A., Knoop, C. I., & Sunder, S. R. (1998). *Amul and India's National Dairy Development Board*. Harvard Business School Case 9-599-060.
3. Heredia, R. (1997). *The Amul India Story*. New Delhi: Tata-McGraw Hill.
4. Murty, K. G. (2009). *Optimization for Decision Making: Linear and Quadratic Models*. International Series in Operations Research & Management Science. New York: Springer.
5. Smith, J. D. (2009). *The Mahabharata*. London: Penguin. (Abridged edition).

Chapter 5
DSS (Decision Support System) for Allocating Appointment Times to Calling Patients at a Medical Facility

Adel Alaeddini and Katta G. Murty

1 Introduction

A variety of terms are used to describe medical facilities offering treatment to general patients—hospitals, clinics, etc. We will use the generic term *hospital* for such a facility. In this chapter, we will discuss one common decision making problem encountered in daily operations at such a facility. When a patient has a health condition that needs treatment, she or he calls her/his hospital for an appointment. The receptionist who receives that call asks the caller a few basic questions, like who her/his doctor is, and the reason for the call, etc. The receptionist then looks up the patient records on the desktop computer in front of her, and based on the data stored in that record, and information obtained from that phone call, she faces the decision making problem of selecting the date and time for the patient's appointment with the required caregiver in the hospital subject to various operational conditions and requirements to optimize important objectives described in more detail later. At most hospitals, receptionists have to deal with hundreds of such calls every day, and the solution for the decision making problem in each call has to be found during the short duration of the call. The name used for procedures for solving such decision making problems occurring sequentially over time with the requirement that the solution of each must be determined within a very short time of the problems occurring is *real-time decision making algorithms*. Clearly, receptionists at hospitals need a decision support system (DSS) installed on the desktop in front of them, which they can use to determine the appointment time with medical practitioners in the hospital to calling patients. This chapter deals with the problem of developing such a DSS.

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_5) contains supplementary material, which is available to authorized users.

A. Alaeddini (✉)
University of Texas at San Antonio, San Antonio, TX 78249, USA
e-mail: adel.alaeddini@utsa.edu; alaedini@umich.edu

A. Alaeddini · K. G. Murty
University of Michigan, Ann Arbor, MI, USA
e-mail: murty@umich.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_5, © Springer Science+Business Media New York 2015

Fig. 5.1 **a** Castor plants. **b** Castor beans. **c** *Piptoporus betulinus*

2 History of Medicine

All human societies had medical beliefs that provide explanations for birth, death, and disease. Throughout history, illness has been attributed to many things such as demons or evil spirits. These ideas are still around to some extent, although the rise of scientific medicine over the past millennium has altered or replaced mysticism in most cases.

Nubians, who were the early occupants of Egypt (during the time of Pharos, going back to 7000 BCE), had a system of medicine that was very advanced for its time and influenced later medical traditions. The Egyptians and Babylonians both introduced the concepts of diagnosis, prognosis, and medical examination. In the medieval era, surgical practices inherited from the ancient masters were improved and then systematized in Rogerius's *The Practice of Surgery*. During the Renaissance, understanding of anatomy improved, and the invention of the microscope would later lead to the germ theory of disease. These advancements, along with developments in chemistry, genetics, and lab technology (such as the X-ray) led to modern medicine.

2.1 Prehistoric Medicine

Prehistoric medicine is the term used to describe the practice of medicine before the invention of writing. People in prehistoric times used to suffer from different ailments than those common today. For instance, there is evidence that in those days many people suffered from osteoarthritis, probably caused by the lifting of heavy objects which would have been a daily and necessary task in their societies. Also, without proper cleanliness, antiseptics, and knowledge about germs, minor incidents such as cuts, bruises, and bone fractures, would have led to serious infections.

Preparations made from herbs and other plants, played a major role in the treatments of diseases in prehistoric cultures. For example, *castor bean oil* in the tropical regions, and preparations using the plants *Piptoporus betulinus* commonly found in alpine environments, have been used widely as basic laxatives by prehistoric people living in those regions (See Fig. 5.1); since they found that these preparations bring

on short bouts of diarrhea when ingested. In fact, traces of these materials were found among the possessions of mummified bodies. Castor bean oil is used as a laxative in some parts of India even today.

Applications of healing clays and other special types of soil used both externally and internally, can also be observed among wild and domesticated animals for treating wounds and other ailments. Observing animal behavior, prehistoric people adopted these practices as treatment procedures. Widespread use of such treatment procedures was observed among aboriginal people around the world, and even recorded in documents dated to preindustrial periods.

The practice of carrying out minor surgical procedures such as trepanning by the medicine men was also common among prehistoric societies across the globe. Medicine men (also witch-doctors, shamans) carried out such minor surgical procedures in the belief that they will remove the ill-effect of an evil spirit trapped within the body of the patient. In prehistoric societies, these medicine men had responsibility to maintain the health of their tribe by also distributing plant-based preparations, and providing supernatural treatments such as charms, spells, and amulets to ward off evil spirits.

2.2 Medicine in Antiquity Among Different Regions of the World

2.2.1 Babylon

Along with contemporary ancient Egyptian medicine, the Babylonians introduced the concepts of diagnosis, prognosis, physical examination, and medical prescriptions. In addition, their *Diagnostic Handbook* introduced the methods of therapy, etiology, the use of empiricism, logic, and rationality in diagnosis, prognosis, and therapy.

The oldest Babylonian texts on medicine date back to the old Babylonian period in the first half of the second millennium BCE. The most comprehensive Babylonian medical text, however, is the *Diagnostic Handbook* composed around 1000 BCE. The Diagnostic Handbook was based on a logical set of axioms and assumptions, including the modern view that through the examination and inspection of the symptoms of a patient, it is possible to determine the patient's disease, its etiology and future development, and the chances of the patient's recovery. The symptoms and diseases of a patient were treated through therapeutic means such as bandages, creams, and pills.

2.2.2 China

China also developed a large body of traditional medicine. Much of the philosophy of traditional Chinese medicine derived from empirical observations of disease and illness by Taoist physicians and reflects the classical Chinese belief that individual human experiences express causative principles effective in the environment at all

scales. These causative principles, whether material, essential, or mystical, correlate as the expression of the natural order of the universe. Traditional Chinese Medicine that is based on the use of herbal medicine, acupuncture, massage, and other forms of therapy has been practiced in China for thousands of years. The foundation of Chinese medicine can be found in the *Huangdi Neijing* usually dated to the late Warring States period (fifth century-221 BCE).

2.2.3 Egypt

At the time of Pharos, going back to around 7000 BCE, Egyptians had developed a large, varied, and fruitful medical tradition. Herodotus described the Egyptians as "the healthiest of all people, next to the Libyans," due to the dry climate and the notable public health system that they possessed. According to him, "the practice of medicine is so specialized among them that each physician is a healer of one disease and no more." Also, they perfected the process of embalming and mummifying the dead bodies of Pharos, and preserving these mummies that we can see and marvel even today after so many centuries.

The earliest known surgery was performed in Egypt around 2750 BCE. Imhotep in the third dynasty is sometimes credited with being the founder of ancient Egyptian medicine and with being the original author of the *Edwin Smith Papyrus*, detailing cures, ailments, and anatomical observations. The Edwin Smith Papyrus is regarded as a copy of several earlier works and was written circa 1600 BCE, it details the examination, diagnosis, treatment, and prognosis of numerous ailments.

Medical institutions, referred to as *Houses of Life,* are known to have been established in ancient Egypt as early as the first dynasty. By the time of the 19th dynasty, some workers enjoyed such benefits as medical insurance, pensions, and sick leave. The earliest known physician is also credited to ancient Egypt: Hesy-Ra, "Chief of Dentists and Physicians" for King Djoser during the third dynasty of Egypt. Also, the earliest known woman physician, Peseshet, practiced in Ancient Egypt at the time of the fourth dynasty. Her title was "Lady Overseer of the Lady Physicians." In addition to her supervisory role, Peseshet trained midwives at an ancient Egyptian medical school in Sais.

We get a lot of information on the practice of medicine in ancient Egypt from the research being carried out on deciphering the inscriptions on the famous pyramids, and many excavations of the tombs from these times in the Egyptians desert (See Fig. 5.2).

2.2.4 Greek and Roman

The first known Greek medical school opened in Cnidus in 700 BCE. As was the case elsewhere, the ancient Greeks developed a humoral medicine system where treatment sought to restore the balance of humors within the body.

Fig. 5.2 **a** Medical inscriptions from a Pyramid in Egypt. **b** Pyramids in Egypt. **c** Medical services in ancient Egypt

A towering figure in the history of medicine of those times was the physician Hippocrates of Kos (ca. 460 BCE—ca. 370 BCE), considered the "father of modern medicine." The Hippocratic Corpus is a collection of around seventy early medical works from ancient Greece strongly associated with Hippocrates and his students. Most famously, Hippocrates invented the *Hippocratic Oath* for physicians, which is still relevant and in widespread use today.

Hippocrates and his followers were the first to describe many diseases and medical conditions. He is given credit for the first description of clubbing of the fingers, an important diagnostic sign in chronic suppurative lung disease, lung cancer, and cyanotic heart disease. For this reason, clubbed fingers are sometimes referred to as "Hippocratic fingers." Hippocrates was also the first physician to describe *Hippocratic face* in Prognosis. Hippocrates began to categorize illnesses as acute, chronic, endemic, and epidemic. Another of Hippocrates's major contributions may be found in his descriptions of the symptomatology, physical findings, surgical treatment, and prognosis of thoracic empyema, i.e., suppuration of the lining of the chest cavity. His teachings remain relevant to present-day students of pulmonary medicine and surgery. Hippocrates was the first documented chest surgeon and his findings are still valid.

The Greek Galen was one of the greatest surgeons of the ancient world and performed many audacious operations—including brain and eye surgeries—that were not tried again for almost two millennia. Later, in medieval Europe, Galen's writings on anatomy became the mainstay of the medieval physician's university curriculum; but they suffered greatly from intellectual stagnation. In the 1530s, however, Belgian anatomist and physician Andreas Vesalius took on a project to translate many of Galen's Greek texts into Latin. Vesalius's most famous work, *De humani corporis fabrica*, was greatly influenced by Galenic writing and form. The works of Galen and Avicenna, especially *The Canon of Medicine* which incorporated the teachings of both, were translated into Latin, and the Canon remained the most authoritative text on anatomy in European medical education until the sixteenth century.

Fig. 5.3 **a** Part of Atharvaveda. **b** Palm-leaf manuscripts

The Romans invented numerous surgical instruments, including the first instruments unique to treatment of women, as well as the surgical uses of forceps, scalpels, cautery, cross-bladed scissors, and surgical needle. Romans also performed cataract surgery.

2.2.5 India

The Vedas (meaning "knowledge," or "wisdom," or "vision") of India are the oldest extant texts composed by humanity (earliest documents of the human mind), and they date back well over 6000 years. There are four of them, Rigveda, Yajurveda, Samaveda, and Atharvaveda. They are considered as the earliest record of the Indo-Aryan civilization. There is discussion of diseases and their cures, and rites for long life, in Atharvaveda (See Fig. 5.3).

At the time of their composition, alphabet and writing were not discovered yet, so they were handed down to posterity by oral tradition. Some people committed them to memory, and to remember them, they used to chant them daily. In order to make this chanting pleasant, the Vedas were composed in the form of verses. To make the chanting and listening of it enjoyable, they adopted a highly stylized version for this recitation. Prodigious energy was expended by ancient Indian culture in ensuing that these texts were transmitted from generation to generation with inordinate fidelity. The class of people who took on the responsibility of chanting the Vedas formed the "Brahmin caste" in India.

To transmit the Vedas to the next generation, they faced the problem of encouraging children to memorize and learn to chant them. When fully awake, children would rather play with their friends than pay attention to memorizing the Vedas or listening to their chanting. For this reason, they started the practice of chanting the Vedas before daybreak while the children are still in bed. This is to make sure that the children get accustomed to the sound of chanting of the Vedas while they are making up their mind to awaken. This practice made it easy for the children in Brahmin families to learn the style of chanting the Vedas, and to motivate these children to memorize them and chant them themselves. This led to the custom of Brahmin children becoming Brahmins themselves.

Alphabets were invented much later, and by about 1500 BCE the practice of preserving texts in palm-leaf manuscripts written using incisions with a pointed metal stylus started in India. Dried palm-leaves (of palmyra or talipot palms) served the role of paper in those days (See Fig. 5.3).

Then by about 800 BCE, developments in healing led to the formation of "Ayurveda," the world's oldest medical system. The word "Ayurveda" does not refer to a single text, but to the "science of life or longevity." It is a form of medicine relying heavily on herbal medicines and Yoga asanas and exercises. Around that time, the Indian scholar Charaka compiled the principles of Ayurveda in a text called "Charaka Samahita." Another medical scholar Sushruta (now known as the "father of surgery") who lived in Kasi (present day Benares), compiled a collection called "Sushruta Samahita" in which he describes many surgical instruments, surgical procedures, and classifies human surgery into eight categories.

2.2.6 Iran (Persia)

The practice and study of medicine in Persia has a long and prolific history. The Iranian academic centers like Jundishapur University (third century CE) were a breeding ground for mingling among great scientists from different civilizations. These centers successfully followed their predecessors' theories and greatly extended their scientific research through history.

In recent years, some experimental studies have indeed evaluated Medieval Iranian medical remedies using modern scientific methods. These studies raised the possibility of revival of traditional treatments on the basis of evidence-based medicine.

The medical history of ancient Persia can be divided into three distinct periods. The sixth book of Zend-Avesta contains some of the earliest records of history of ancient Iranian medicine. Some of the most notable of Persia's ancient physicians were Mani, Roozbeh, and Bozorgmehr. The second epoch covers the era of what is known as Pahlavi literature, where the entire subject of medicine was systematically treated in an interesting tractate incorporated in the encyclopedic work of Dinkart, which listed in an altered form some 4333 diseases. The third era begins with the Achaemenid dynasty, and covers the period of Darius I of Persia, whose interest in medicine was said to be so great that he re-established the school of medicine in Sais, Egypt, which previously had been destroyed, restoring its books and equipment.

2.3 Middle Ages

Most of the material in this section and the next two sections is taken from the article "the history of medicine" in Wikipedia (http://en.wikipedia.org/wiki/History_of_medicine). Persia's position at the crossroads of the East and the West frequently placed it in the midst of developments in both ancient Greek and Indian medicine. The first generation of Persian physicians trained at the Academy of Jundishapur. This evolved into the medieval Islamic Bipartisan hospitals.

Avicenna, considered among the most influential medical scholars in history, wrote The *Canon of Medicine* (1025) and *The Book of Healing* (1027), which remained standard textbooks in both Muslim and European universities until the seventeenth century. Avicenna's contributions include the distinction of mediastinitis from pleurisy and careful descriptions of skin troubles, sexually transmitted diseases, and nervous ailments, as well the use of ice to treat fevers, and the separation of medicine from pharmacology, which was important for the development of the pharmaceutical sciences.

Al-Kindi wrote De Gradibus, in which he demonstrated the application of mathematics to medicine, particularly in the field of pharmacology. This includes the development of a mathematical scale to quantify the strength of drugs, and a system that would allow a doctor to determine in advance the most critical days of a patient's illness.

Abu al-Qasim al-Zahrawi (Abulcasis), who some have called the father of modern surgery, wrote the Kitab al-Tasrif (1000), a 30-volume medical encyclopedia which was taught at Muslim and European medical schools until the seventeenth century. He used numerous surgical instruments, including some that are unique to women's treatments.

2.4 Renaissance to Early Modern Period

With the Renaissance came an increase in experimental investigation, principally in the field of dissection and body examination, thus advancing our knowledge of human anatomy. The development of modern neurology began in the sixteenth century with Vesalius, who described the anatomy of the brain and other organs; he had little knowledge of the brain's function, thinking that it resided mainly in the ventricles. Over his lifetime, he corrected over 200 of Galen's mistakes. Understanding of medical sciences and diagnosis improved, but with little direct benefit to healthcare. Few effective drugs existed, beyond opium and quinine. Folklore cures and potentially poisonous metal-based compounds were popular treatments in this period.

2.5 Nineteenth Century: Rise of Modern Medicine

Medicine was revolutionized in the nineteenth century and beyond, advances in chemistry and laboratory techniques and equipments replaced old ideas of infectious disease epidemiology replaced with bacteriology and virology. Bacteria and microorganisms were first observed with a microscope by Antonie van Leeuwenhoek in 1676 CE, initiating the scientific field microbiology.

Ignaz Semmelweis (1818–1865 CE) in 1847 CE dramatically reduced the death rate of new mothers from childbed fever by the simple expedient of requiring physicians to clean their hands before attending to women in childbirth. His discovery

predated the germ theory of disease. However, his discoveries were not appreciated by his contemporaries, and came into general use only with the discoveries of British surgeon Joseph Lister, who in 1865 proved the principles of antisepsis in the treatment of wounds.

Gregor Mendel (1822–1884 CE) published in 1865 CE his books on pea plants, which would be later known as Mendel's laws. Re-discovered at the turn of the twentieth century, they would form the basis of classical genetics. The 1953 discovery of the structure of DNA by Watson and Crick would open the door to molecular biology and modern genetics.

Semmelweis's work was supported by the discoveries made by Louis Pasteur. Linking microorganisms with disease, Pasteur brought about a revolution in medicine. He also invented with Claude Bernard (1813–1878 CE) the process of pasteurization still in use today. His experiments confirmed the *germ theory*. Claude Bernard aimed at establishing scientific methods in medicine; he published *An Introduction to the Study of Experimental Medicine* in 1865 CE. Beside this, Pasteur, along with Robert Koch (who was awarded the Nobel Prize in 1905 CE), founded bacteriology. Koch was also famous for the discovery of the tubercle bacillus (1882 CE) and the cholera bacillus (1883 CE) and for his development of Koch's postulates.

It was in this era that actual cures were developed for certain endemic infectious diseases. However, the decline in many of the most lethal diseases was more due to the improvements in public health and nutrition than to the medicine. It was not until the twentieth century that the application of the scientific method to medical research began to produce multiple important developments in medicine, with great advances in pharmacology and surgery.

2.6 Twentieth Century

During the twentieth century, large-scale wars were attended with medics and mobile hospital units that developed advanced techniques for healing massive injuries and controlling infections rampant in battlefield conditions.

Cardiac surgery was revolutionized in the late 1940s, as open-heart surgery was introduced. In 1954, Joseph Murray, J. Hartwell Harrison, M.D., and others accomplished the first kidney transplantation. Transplantations of other organs, such as heart, liver, and pancreas, were also introduced during the latter twentieethth century. The first partial face transplant was performed in 2005, and the first full one in 2010. By the end of the twentieth century, microtechnology had been used to create tiny robotic devices to assist microsurgery using microvideo and fiber-optic cameras to view internal tissues during surgery with minimally invasive practices. More detailed information about modern surgery can be found at: http://en.wikipedia.org/wiki/Surgery.

3 The Process a New Patient Uses to Select Her/His PCP

3.1 Different Types of Doctors in a Hospital

There are many specialties in medicine today. *Primary care physician (PCP)* is a medical doctor who provides both the first contact for a person with an undiagnosed health concern as well as the continuing care of varied medical conditions, not limited by cause, organ system, or diagnosis. Although some women may consider their obstetrician/gynecologist as their PCP, in most cases only doctors with the following three specialties are assigned as PCPs:

- *Family practice or general practice:* A family practitioner is qualified to care for the entire family. A family practitioner can be board-certified and have training in a variety of subjects including obstetrics and gynecology, pediatrics, internal medicine, and psychiatry. Typically when all members of a family of different ages sign up for service from a clinic, all of them will have a family practitioner as their PCP. When an individual by her/himself sign up for service from a clinic she or he typically will have a general practitioner in the clinic as her/his PCP.
- *Internal medicine:* An internist can diagnose and treat disease with medicine. An internist is not a surgeon. There are several subspecialties an internist can have, including: specializing in a particular organ, like the lungs or the kidneys, a particular disease, like diabetes, or a particular age group, like the elderly. Some patients, based on their health problems, opt to have an internist as their PCP.
- *Pediatrics:* A pediatrician specializes in the overall well-being of children. Most pediatricians treat children from birth until adolescence or about 14 years old. Patients in this age group usually have pediatricians as their PCP. Pediatricians can have subspecialties such as surgery or pediatric cardiology.

3.1.1 Skills Needed for Serving as a PCP

The skills of primary care physicians generally include basic diagnosis and treatment of common illnesses and medical conditions. Diagnostic techniques include interviewing the patient to collect information on the present symptoms, prior medical history and other health details, followed by a physical examination. Many PCPs are trained in basic medical testing, such as interpreting results of blood or other patient samples, electrocardiograms, or x-rays. If the patient's condition needs more complex and time-intensive diagnostic procedures and treatments, the PCP refers the patient to a specialist who has special training and experience in the diagnosis and treatment of that condition.

After collecting the data, the PCP arrives at a diagnosis and, with the participation of the patient, formulates a plan including (if appropriate) components of further testing, specialist referral, medication, therapy, diet or lifestyle changes, patient education, and follow-up results of treatment. PCPs also counsel and educate patients on safe health behaviors, self-care skills and treatment options, and authorize screening tests and immunizations.

Ideally, the PCP acts on behalf of the patient to collaborate with referral specialists, coordinate the care given by varied organizations such as hospitals or rehabilitation clinics, act as a comprehensive repository for the patient's records, and provide long-term management of chronic conditions. Continuous care is particularly important for patients with medical conditions that encompass multiple organ systems and require prolonged treatment and monitoring, such as diabetes and hypertension.

3.1.2 Specialties Among Medical Practitioners

Similar to primary care, various specialties in medicine are various branches of medical science. After completing medical school, physicians, or surgeons usually further their medical education in a specific specialty of by completing a multi-year residency to become a medical specialist. Medical specialties can be classified along several axes:

- *Surgical or internal medicine:* Surgery is the specialty in which diagnosis and treatment are achieved through surgical techniques. Internal medicine is a branch of medicine in which diagnosis and treatment does not involve opening up the body of the patient. Anesthesiology is classified as a surgical discipline, since it is vital in the surgical process, though anesthesiologists never perform a major surgery themselves.
- *Age range of patients:* The age range of patients seen by any given specialist can be quite variable. Pediatricians handle most complaints and diseases in children that do not require surgery, and there are several subspecialties (formally or informally) in pediatrics that mimics the organ-based specialties in adults. Pediatric surgery may or may not be a separate specialty that handles some kinds of surgeries in children.
- *Diagnostic or therapeutic:* A further subdivision is the diagnostic versus therapeutic specialties. While the diagnostic process is of great importance in all specialties, some specialists perform mainly or only diagnostic examinations, such as pathology, clinical neurophysiology, and radiology.
- *Organ-based or technique-based:* Many specialties are organ-based. Many symptoms and diseases come from a particular organ. Others are based mainly around a set of techniques, such as radiology, which was originally based around X-rays.

A detailed list of specialties that are common worldwide and those are recognized in North America can be found though the following link: http://en.wikipedia.org/wiki/Medical_specialist.

3.2 How a New Patient Begins the Process of Receiving Health Services

Typically the only way a patient can get specialized healthcare is to first see her/his PCP, who will then give her/him a referral to the specialist. Also with many health

insurance plans, one must select a primary care physician, or PCP, who is her/his "go-to" doctor for any health concerns. So it is important to select a physician who the patient trusts and in whom the patient has confidence.

Ideally, the time to establish a relationship with a PCP is when a patient is well, not when someone is sick or injured and needs medical help in a hurry. Even if her/his choice is limited to physicians within a health plan network, using certain steps outlined below will help ensure that the selected PCP is one who meets the requirements and provides quality healthcare:

Step 1: Assess Medical Needs and Preferences An important step in determining an appropriate PCP is the patient's self- assessment of her/his medical needs. The overall health status of the patient, family history or having chronic illness, such as diabetes or hypertension may suggest the need for a particular type of physician. For some patients it may be important that the physician emphasizes lifestyle changes, such as diet and exercise. The potential patient may also need to consider her/his preference (if any) over the gender, age, race, etc. of the PCP.

Step 2: Gather Information Normally, most people start by choosing their insurance plan and then picking a doctor on the insurance's list. People may also ask their friends, families, and coworkers for recommendations or referrals to PCPs. It is also not very uncommon for hospitals to offer a referral service that can provide patients with the names of staff doctors.

Once the patient gets the listing of doctors who are accepting new patients, further information may be gathered to identify the doctor who has the best match with the patient's needs. Some of the potential pieces of information to be gathered and their sources are listed below:

- *Expertise:* this information includes the university the doctor graduated from, number and types of special training she or he has had if any, and her/his years of experience. Such information about a doctor can be gained by calling health plan service representative or visiting the plans website. Another source for this information is state medical licensing boards and American Medical Association's website: http://www.healthgrades.com.
- *Board certification:* the information about a doctor can be gained from American Board of Medical Specialties (ABMS) website at http://www.certifacts.org/.
- *Complaints or disciplinary actions*: there are several sources such as state medical board, and local county clerk's office to find out about information on past and pending disciplinary actions, and malpractice lawsuits on a doctor. Sanctions information is also available for many states at http://www.healthgrades.com and http://www.docboard.org.
- *Recommendations and resources*: a very good way of gaining this information about doctors is asking friends, family members, and coworkers for personal recommendations. Another source may be physicians, nurses, and other healthcare workers whose opinions you trust. There are websites such as http://www.bestdoctors.com that will provide the names of physicians in different areas who have been recommended by other doctors in their database.

- *Access, physician's background, and office practices*: access information can be on standards for appointment wait time, typical visit length, routine for urgent appointments, driving distance, ease of parking, working days, and office hours. Physician background can be about the doctor's age, gender, the experience of the doctor with specific health condition, her/his policy regarding prescribing medications over the phone and giving phone advice. Finally the office practice information is about whether the health service is group practice or not, number of specialized services like blood test and other labs available in the office, and fees and billing procedures.

Most of this information about medical practitioners can be obtained from the sources given above, and by a telephone call to the hospital where he or she works. It is likely that the patient would not be able to speak with the physician directly, but an office manager, nurse, or medical assistant should be able to answer the majority of patient questions. Some offices might also allow patients to conduct a "get-acquainted" visit with the physician, so ask if this is an option.

Some patients may also want to schedule an appointment to meet with the physician to address their medical concerns, and evaluate the time physician spends for answering their questions, their behavior, etc.

Step 3: Evaluate and Choose As discussed above, patients evaluate their physicians on many different characteristics. A new patient trying to select his PCP at a hospital from a list of available candidates may find that each doctor in the list is excellent on some characteristics but poor in others. In fact it is very rare that a doctor in the list would turn out to be excellent on all the characteristics. When all the different characteristics are of some importance to some degree how does the patient select his PCP from the list in this case?

The branch of optimization dealing with such a class of problems is called multicharactersitic decision making (MCDM). One technique for solving these problems is the *scoring method* [14].

Summary of Scoring Method

1. *Alternatives, characteristics*: Determine all the available alternatives (among which one has to be chosen) for the decision, and all of the characteristics on which each of them has to be evaluated. In the PCP selection context, available alternatives are candidate doctors one of whom should be chosen as the PCP. Also characteristics are the criteria that the patient is using for comparing the doctors.
2. *Determine weights for each characteristic*: All characteristics are important, but usually some are more important than the others. The decision maker (DM, in our situation, it is the new patient) determines a *weight* for each characteristic as a measure of its importance for her/him. The weights are numbers between 0 and 1; and normally they are chosen to satisfy the property that the sum of the weights of all the characteristics considered is 1, and the higher the weight of a characteristic the more important that characteristic is.
3. *Determine the score of each alternative on each characteristic:* This involves evaluating each alternative and assigning a score for it on each characteristic.

Typically each score is a number between 0 and 100; the higher the score the better the alternative is on that characteristic. So, a score of 100 indicates that the alternative is "excellent" on that characteristic. On the other hand, a score of 0 indicates that the alternative is "worst possible" on that characteristic.

4. *Compute the combined score for each alternative:* Suppose there are n alternatives for the choice, and m characteristics on which each alternative is evaluated. Let:

$$w_j = \text{The weight of the } j \text{ th characteristic, } j = 1 \text{ to } m$$

$$a_{ij} = \text{Score of the } n \text{ th alternative on the } j \text{ th chatartetristic,}$$

$$i = 1 \text{ to } n, j = 1 \text{ to } m$$

$$S_i = \sum_{j=1}^{m} w_j a_{ij},\ i = 1 \text{ to } n.$$

Then, S_i is the combined score of alternative i,i=1 to n.

5. *Select the best alternative:* The alternative with the highest value for the combined score S_i is the best alternative, select that alternative. In fact, when the combined scores vector $(S_1, \ldots, S_n)$ is arranged in decreasing order left to right, their subscripts in that order, give the alternatives in decreasing order of desirability for selection.

For an illustrative numerical example see example 2 in Sect. 7.

Choosing a physician can be an overwhelming process unless you know where to start. But finding the right doctor for your healthcare needs can help to ensure greater continuity of care, better communication among your healthcare providers, and even peace of mind for you and your family in your future life.

4 Description of the Data Provided for Our Study

Our study is carried out based on the dataset from a Veterans Affairs (VA) medical center in the USA. While there are many patients being served by our hospital, our study on patient allocation to PCP is restricted to a sample of 28,866 patients among them. Data-table 1 provides information on the age, sex, and frequency of visits of each of these patients to their PCP in the year 2010.

Data-table 2 provides information on the age, sex of each patient in a random sample of 99, and for each appointment they have taken to visit their PCP during 2009 and 2010, whether they kept the appointment or failed to show up for the appointment. Both Data-tables 1 and 2 can be downloaded from http://extras.springer.com in the folder corresponding to the ISBN number of this book.

We assume that the patients included in each of Data-tables 1 and 2, is a representative random sample from the population of patients being served by their hospital.

5 The Process of Allocating Patients to PCP Doctors

In almost all hospitals, the patient access to healthcare can be measured by the time in days between patient's desired appointment date and the actual appointment date obtained for primary care, which is known as *appointment waiting time* [5, 8, 12], whose expected value we will denote by Q1.

At a hospital, a PCP doctor's *patient panel* is defined as the set of patients for whom she or he is the PCP; and the *panel size* is defined as the number of patients in it.

Typically the number of hours that a doctor commits to work/month at a hospital is not the same for all doctors. We will use the following symbols:

i = Index used to denote PCP doctors working at a hospital, $i = 1$ to n,

n = Number of PCP doctors working at the hospital,

t_i = Hours that PCP doctor i is committed to work at the hospital/month, typically assumed to be the same every month,

y_i = Size of the patient panel affiliated with doctor i.

The Process of Allocating Patients to PCPs in Current Practice One of the important factors that affect appointment waiting time among patient's affiliated with a PCP doctor is the composition of the patient panel of that doctor. In current practice, most healthcare providers have a strong desire to *equalize the workload of their PCP doctors***.** It is generally believed that this can be achieved by equalizing the panel size of all PCP doctors. But all the doctors do not work the same number of hours at the hospital, so many hospitals believe that equalizing the workload of their PCP doctors can be achieved by making the panel size of each of them proportional to the number of hours they work, i.e., make $[y_i/t_i]$ the same for all PCP doctors at the facility. If the upper bound for $[y_i/t_i]$ selected by the facility is denoted by u, then when the value for $[y_i/t_i]$ for doctor i reaches this bound u, the patient panel for this doctor is considered to be full, and no more patients will be added to this panel [8, 13]. This type of an approach for allocating patients to patient panels is known as a *constant bin size method it is based on the assumption that the workloads imposed by the various patients on her/his PCP are equal.*

Developing a New Procedure for Allocating to PCPs However, this "constant bin size" method usually fails to recognize the high degree of variation in healthcare work load among patients with different sex, age, etc. which drastically affects the visiting frequency and service load imposed by the panel. Consequently, a lot of variation occurs in appointment waiting time of the patient and utilization of physicians. When patient panels are determined by this constant bin size method, this variation introduces significant differences in the workload of their PCP doctors per hour they contract to work, and consequently decreases the healthcare access and increases the system and patient cost.

Various approaches have been explored for panel size determination [8, 9, 12, 13]. However, few of them optimally allocate panel members to each PCP based on a

data driven and scientific approach, using healthcare data such as work load and demographic data available in healthcare informatics systems.

In this chapter, we will discuss a simple real-time method based on past data from a medical center, to optimally determine the patient panel for each PCP to maximize access while maintaining high degree of resource utilization and continuity of patient care. The method has three components that will be discussed in detail in the forthcoming subsections:

1. Classification of patients into homogenous classes such that all patients in each class have the same expected number of visits/year to her/his PCP.
2. Estimation of patient workload, i.e., number of visits to the PCP doctor/year/patient in each class.
3. Developing an optimization procedure for panel determination to make sure that the expected value of the workload per hour they contract to work is the same for all PCP doctors.

5.1 Classification of Patients into Homogenous Classes

Here, we present an efficient method for classification of patients into homogenous classes based on our dataset from the VA hospital. Our aim is to classify patients at this VA hospital into various groups called *classes* such that all the patients within each class impose comparable workload in terms of number of visits/year for service at the hospital. Some of the most appropriate characteristics of the patient for this grouping are:

- Sex of the patient: M (male), F (female).
- The data included only patients of ages 18 year or older, that is why we only consider ages in this range in this chapter. To carry out this work at another hospital, the age range pertaining to that hospital must be used, for example it may include: infants and children also. In our study, age is classified into age groups as:
  - Adult (18–40 years of age), abbreviated as AD;
  - Middle Age (40–60 years of age), abbreviated as MA;
  - Senior (60–75 years of age), abbreviated as SE;
  - Elderly (over 75 years of age), abbreviated as EL.

1. By classifying patients using these two characteristics (Gender, Age), we get eight different Gender-Age groups *G-A groups*, namely F-AD, F-MA, F-SE, F-EL, M-AD, M-MA, M-SE, M-EL. In our study, we further classify each of these G-A groups into two subgroups named V1, V2 using the third characteristic: "Yearly number of visits from past data": i.e., frequency of visits to their PCP from the past data. We define:
   - V1: Set of patients in that G-A group for whom the annual number of their visits to their PCP is in the range of this number, chosen by a high frequency of patients in that G-A group;
   - V2: Set of all the remaining patients in this G-A group.

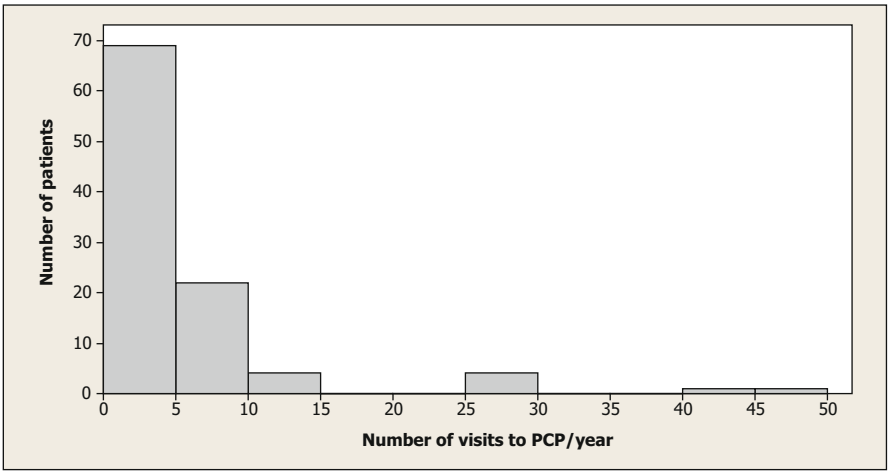


Fig. 5.4 Histogram for G-A group F-AD

To determine these sets V1, V2 in any G-A group, you need to draw the histogram of:

X = Annual number of visits to their PCP of a patient in this G-A group.

To draw this histogram for a G-A group, find the range of values of X among patients in this G-A group from Data-table 1 for the year 2010. Divide that range into a convenient number of small intervals all of equal width. For each interval determine its:

$$\text{Frequency} = \text{Number of patients in this G} - \text{A group for whom their} - \text{value lies in this interval.}$$

Then on a sheet of paper mark the various intervals of values of X on the horizontal axis, and on each interval erect a rectangle with its height proportional to its frequency along the vertical axis. The resulting diagram is the histogram of X corresponding to this G-A group.

As an example consider the G-A group F-AD, the values of X for patients ranged (Fig. 5.4) from 0 to 50. We divide this range into 10 intervals of width 5 each; and draw the histogram given below.

We see that the left most two intervals 0–5, 5–10 are those corresponding to the highest frequencies of patients. So, for this G-A group we define the set V1 to be the set of patients in this G-A group who visit their PCP ≤ 10 times/year, and V2 is the set of all other patients in this G-A group.

Similarly, for the M-AD G-A group, the histogram is shown in Fig. 5.5. Using the same procedure, we define V1 as the set of patients visiting their PCPs ≤ 3 times/year in this G-A group.

To save space, we will show the histograms of the other G-A groups numerically in an abbreviated fashion. ("Interval" refers to the "interval of number of visits to PCP/year, and "frequency" refers to the number of patients with their annual number of visits to PCP in this interval. See Tables 5.1, 5.2.)

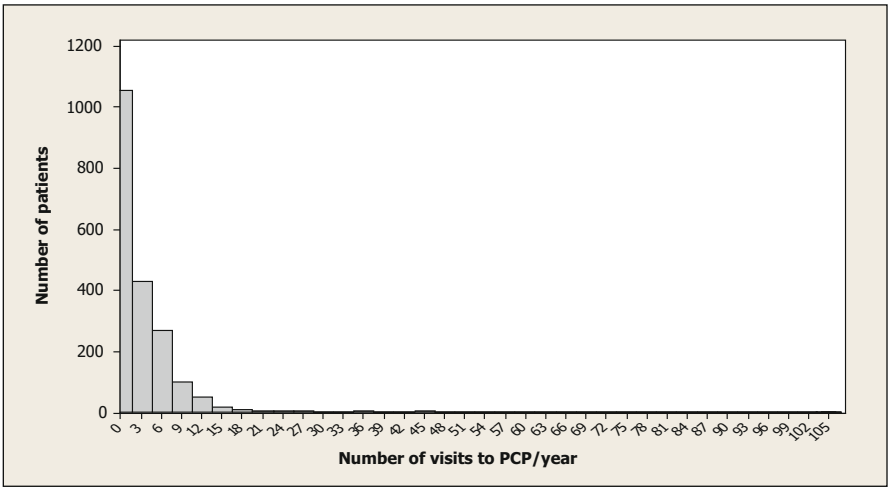


Fig. 5.5 Histogram for G-A group M-AD

Table 5.1 Numerical representation of the intervals and related frequencies for G-A groups F-MA, M-MA, and F-SE

F-MA		M-MA		F-SE	
Interval	Frequency	Interval	Frequency	Interval	Frequency
0–2	155	0–5	4100	0–5	49
2–4	50	5–10	4500	5–10	28
4–6	32	10–15	650	10–15	2
6–8	10	>15	200	>15	3
>8	30				

Table 5.2 Numerical representation of the intervals and related frequencies for G-A groups M-SE, F-EL, and M-EL

M-SE		F-EL		M-EL	
Interval	Frequency	Interval	Frequency	Interval	Frequency
0–6	5300	0–6	40	0–10	5500
6–12	4500	6–12	8	10–20	700
12–18	600	12–18	2	20–30	70
18–24	100	18–24	2	30–40	30
>24	150	>24	2		

From these, we derive the subgroups V1, V2 in each G-A group. In each G-A group V2 is the complement of V1. See Table 5.3.

At this hospital we found that these characteristics are the most appropriate for classifying patients into homogenous groups in terms of number of annual visits to their PCP. At other hospitals there may be different set of characteristics for this classification that has to be determined by examining the database at that hospital. Here in Table 5.3 is the final list of classes that we are going to use:

Table 5.3 Subgroup V1 of each G-A group

G-A Group	Definition of subgroup V1 defined earlier, for this G-A group
F-AD	Set of all patients with annual number visits to their PCP $\leq$ 10 times/year
M-AD	Set of all patients with annual number visits to their PCP $\leq$ 3 times/year
F-MA	Set of all patients with annual number visits to their PCP $\leq$ 2 times/year
M-MA	Set of all patients with annual number visits to their PCP $\leq$ 10 times/year
F-SE	Set of all patients with annual number visits to their PCP $\leq$ 10 times/year
M-SE	Set of all patients with annual number visits to their PCP $\leq$ 12 times/year
F-EL	Set of all patients with annual number visits to their PCP $\leq$ 6 times/year
M-EL	Set of all patients with annual number visits to their PCP $\leq$ 10 times/year

How to Classify a New Patient Seeking Service When a new patient calls to register for service at the hospital, the receptionist gets her/his sex, age, and information on the frequency of her/his annual visits to the doctor in the past, and classifies her/him into the appropriate class based on the information provided, with a temporary label "t". After a couple of years such patients with temporary labels are reclassified into their appropriate class based on actual records generated in this time.

5.2 Estimation of PCPs Workload/Patient per Year in Each Class

We will use the index p to denote various classes of patients. In our problem, p varies from 1 to 16. An estimate of the expected number of visits/year of patients in class p is:

$c_p =$ Average number of visits/year/patient in this class from our data in Data-table 5.1.The value of c_p for $p = 1$ to 16 is shown in Table 5.4 below.

5.3 The Objective Function to Optimize in Allocating Patients to PCPs

We are using the index i to denote PCP doctors at our hospital, where i varies from 1 to n =total number of PCP doctors at the hospital. The number of patients visits that a doctor can handle/year, denoted by:

D_i = Workload capacity of ith PCP doctor for patients visits/year, $i = 1$ to n.

Thus, D_i depends on the number of hours this doctor has contracted to work per year at the hospital. Typically this number is distributed equally among the 12 months of the year, which we assume in this chapter.

Most hospitals follow the practice of allocating a constant duration of time (typically between 15 and 30 min) for each patient visit to the PCP. So this number D_i for PCP doctor i is directly proportional to these contracted annual hours of work and can be determined directly. Let us denote by:

y_{pi} = Number of patients in class p affliated with PCP doctor i, $i = 1$ to n.

In general, we want to make sure that $\sum_{p=1}^{16} y_{pi}c_p$ representing expected annual patient visits experienced by PCP doctor i, is close to D_i. One approach for this is to make patient allocation to PCP doctors such that the integer variables y_{pi} minimize the objective function $\sum_{i=1}^{n}\sum_{p=1}^{16}\left\|y_{pi}c_p - D_i\right\|$.

Some research publications in the literature [13] discussed how this problem of allocating patients to PCP doctors can easily be transformed into a linear integer programming model and solved using widely available software for it.

Unfortunately, this type of model is totally inappropriate for solving this problem for the following important reasons:

a. The set of all patients to allocate to PCPs is never available as a group completely at any point of time. New patients arrive one by one as a continuous stream over time, and they need to be allocated to a PCP for care at the point of their arrival. So the patient allocation problem is actually a *real-time decision making problem*, and what is required for solving it is a real-time decision making algorithm.
b. Patient allocation decision is not really up to the hospital or to some software system. It is the patient who decides which medical team he selects to care for him.
c. Typically the system already exists, with several patient panels formed already, even though some may not be full, and the choice is to be made for each patient in the newly arriving patient stream.

So clearly what this allocation needs is a real-time algorithm, for new patients who arrive to register to service at the hospital. We discuss this real-time algorithm now.

5.4 A Real-Time Algorithm for Patient Allocation to PCP

When a new patient calls up for service from the hospital, the person attending to his call gets all relevant information from her/him, based on it classifies her/him in an appropriate class, and determines all the PCP doctors in the hospital who satisfy the following conditions at that time:

- The current expected annual workload of this doctor in terms of her/his expected annual number of patient visits from her/his present patient panel is less than her/his workload capacity

- The patient's health condition as provided by her/him makes this doctor a suitable PCP for that patient
- and gives that patient a list of their names in order of decreasing (workload capacity - current expected annual workload) and tells the patient to select one of them as her/his PCP.

When the patient makes the selection he or she registers in the patient panel for that PCP doctor, and updates the data for the doctor.

It should be noted that, the list of doctors and their workload capacity, as well as patient dataset and their related classifications should be updated in real-time or at least periodically to keep the effectiveness on the real-time allocation algorithm.

6 Appointment Scheduling for a Calling Patient

6.1 No-Show Events and Estimation of Their Probabilities

Sometime patients make appointments to visit their PCP, but do not show up for the appointments, such an event is called a *no-show event*.

If an appointment on the PCP doctors' schedule for a day turns out to be a no-show appointment, the doctor's time allocated for that appointment can be considered as idle time for that doctor, which could have been utilized to serve another patient. We will estimate the probability of occurrence of no-show events at the hospital and take it into account in the appointment scheduling algorithm.

No-show events have been a recurring problem [1], so hospitals have recently developed systems to remind patients of their appointment: A couple of days before the appointment date, a telephone call is made reminding the appointment date, and asking them to confirm their intention of keeping the appointment, or cancel it if they cannot keep it [6, 7]. This practice has been very effective in reducing the occurrence of no-show events. Now-a-days some hospitals also started the practice of charging patients for no-show-events, to reduce the occurrence of no-show events even further [3].

Due to these practices most general hospitals catering to the general public have very low occurrence of no-show events (typically probability less than 0.05), but some clinics catering to patients of particular health diseases, age groups, job categories, and those depending on others for transportation, may experience higher no-show probability ([2, 10, 11, 15]. No-show probability higher than 0.10 not only causes inconvenience to the hospital management but also have a significant impact on the revenue, cost, and resource utilization for almost all healthcare systems [3, 11]. Hence, accurate prediction of no-show probability is a cornerstone for any scheduling systems and nonattendance reduction strategy [1, 4]. We will use the following symbol as the estimate of the no-show probability at our hospital:

pr = Estimated fraction of no − show appointments from the data over the set of all patients at our hospital.

In the literature, many research publications analyzed this problem theoretically considering a variety of objectives: (Q1) Expected value of the appointment waiting time, where the appointment waiting time is defined as the number of days between the desired day and the assigned appointment day, (Q2) Expected value of the patient waiting time at the hospital on the appointment day for seeing the doctor, expected care team idle time cost, and scheduling systems revenue, etc.

No-show events are experienced in a wide variety of other service systems like airline flights, etc., that provide services to the public. Very satisfactory overbooking strategies have been developed in those industries to minimize the effects of no-show events. We will adopt a similar overbooking strategy for our problem; this is included in the algorithm discussed below.

We will estimate *pr*= probability of a patient appointment with her/his PCP will result in a no-show event, by the fraction of no-show appointments obtained from the data provided in Data-table 2. Its value is:

$pr = 0.204$.

6.2 A Real Time Algorithm for Appointment Scheduling with PCP

On any particular working day, each PCP doctor in a hospital declares how many hours she or he can devote for patients visits on that day. The hospital typically follows the practice of allocating a certain duration for each appointment, this is typically 15 to 30 min, based on the hospital's past experience. For PCP doctor i and any particular working day, let:

$$a_i' = \frac{\text{(time duration in minutes that doctor } i \text{ can devote for patient visits on that date)}}{\text{(duration selected by the hospital per patient visit in minutes)}}.$$

Then, the actual number of patient appointments scheduled for this doctor i on this day will be less than or equal to $a_i = [a_i'/(1 - pr)]$, and each visit is actually scheduled for the time interval:

$$a_i'' = \frac{\text{(time duration in minutes that doctor } i \text{ can devote for patient visits on that date)}}{a_i}$$

When a patient calls up for scheduling an appointment with her/his PCP, say doctor i, the recipient gets the information on the reasons for the visit, and finds the earliest date in the future when the a_i computed above for doctor i is greater than the number of appointments already scheduled for that day for her/him, and tells that date to the patient. There are two possibilities now:

- The patient agrees for the date, then the appointment is fixed and the doctor's data is updated accordingly
- If the patient's condition is such that it is not possible to wait until that data for the appointment, in this case the receptionist looks up some other PCP doctors

Table 5.4 Average no. of visits to PCP/year for patient in each class

Class $p =$	Class symbol	Age range	Sex	Range of visit frequency / year of patients	$c_p =$ Average frequency of visits / year / patient
1	M-AD-V1	(18–40]	Male	≤3	1.2205
2	M-AD-V2	(18–40]	Male	>3	25.2923
3	F-AD-V1	(18–40]	Female	≤10	3.1106
4	F-AD-V2	(18–40]	Female	>10	25.2923
5	F-MA-V1	(40–60]	Female	≤2	1.0585
6	F-MA-V2	(40–60]	Female	>2	6.3564
7	M-MA-V1	(40–60]	Male	≤10	3.1929
8	M-MA-V2	(40–60]	Male	>10	21.3801
9	M-SE-V1	(60–75]	Male	≤12	3.0342
10	M-SE-V2	(60–75]	Male	>12	33.3108
11	F-SE-V1	(60–75]	Female	≤10	2.3556
12	F-SE-V2	(60–75]	Female	>10	24.8231
13	M-EL-V1	(75–130]	Male	≤10	2.2273
14	M-EL-V2	(75–130]	Male	>10	29.4249
15	F-EL-V1	(75–130]	Female	≤6	1.3654
16	F-El-V2	(75–130]	Female	>6	17.3586

in the hospital who are able to see this patient at the earliest time and fixes the appointment with that doctor according to the patient's wishes.

Again similar to patient allocation algorithm to PCP's, the information of doctor's available time, and patient's probability of no-show should be updated in real-time or at least periodically to keep the effectiveness of the algorithm.

7 Solution to Our Problem, Illustrated with Examples

Based on the data, we classified the existing patients at the hospital into 16 homogenous classes shown in Table 5.4.

c_p = expected frequency of visits/year/patient in class p estimated from the data in Data-table1 is shown in Table 5.4 for all $p = 1$to 16.

Using this information we compute the current workload of each PCP doctor. For example if PCP doctor i has $(y_{i,1}, \ldots, y_{i,16})$ as his number of patients in classes 1 to 16 respectively in his current panel, then the estimate of the expected annual workload for patient visits of doctor i is, $W_i = \sum_{p=1}^{16} y_{pi} c_p$ say, for this PCP doctor i. The DSS displays his current workload W_i computed as above, and his workload capacity for patient visits/year D_i.

When a new patient calls the hospitals for enrolling in their service, the receptionist looks at the values of W_i and D_i for all doctors i, and gives the patient a sorted list of these doctors for whom $W_i < D_i$ and asks the patient to select one of them as her/his PCP.

Table 5.5 Workload and current patient panel of various PCP doctors

Patient class	Current no. of patients affiliated with PCP doctor i		
	$i=1$	2	3
	y_{p1}	y_{p2}	y_{p3}
p=1	92	101	73
P=2	69	100	54
⋮	⋮	⋮	⋮
p=16	37	2	49
Expected annual workload $W_i = \sum_{p=1}^{16} y_{pi} c_p$	1565	1330	900
Workload capacity D_i	1560	1560	1560
PCP doctors with annual workload < capacity in decreasing order of (capacity-workload): 3,2,...			

Example 1. What DSS Displays on the Receptionist's Screen to Help New Patient Select Their PCP We will illustrate what the system displays to the hospital's receptionist to help new patients select a PCP at the hospital. First, we will discuss current updated information on PCP doctors displayed. We will illustrate with information shown for PCP doctors $i = 1,2,3$. As an example suppose each of them has contracted to work a total of 26 weeks at the hospital, at the rate of 5 working days/week; and that each of them will spend 6 h/day on patient visits; then assuming that the hospitals allots 30 min/patient visit D_i =their workload capacity for patient visits/year $= 26 \times 5 \times 6 \times \left(\frac{60}{30}\right) = 1560$, for $i = 1,2,3$. The information on the PCP doctors displayed is summarized in Table 5.5.

When a new patient calls, the receptionist will give her/him the names of PCP doctors with capacity>workload from the bottom of this table, and let them choose one of these doctors as her/his PCP for service at the hospital.

For an illustration, we have $D_1 = D_2 = D_3 = 1560$ for doctors 1, 2, 3. The current expected workload estimated for these doctors 1, 2, 3 are 1565, 1300, 900, respectively. Then the patient panel of PCP doctor 1 is already full, but not for PCP 2, 3. Hence, doctors 3, 2 are the doctors whom the new patient can select as her/his PCP in decreasing order of (Capacity-workload).

Example 2. Illustration of How a New Patient Selects Her/His PCP from the List of Doctors Available Suppose the new patient is given the names of four doctors to select as her/his PCP. she or he has gathered the information discussed in Sect. 3.2 for this selection. The doctors characteristics evaluated, the patient's score of each doctor on each characteristics (higher the score, the better the doctor on those characteristics, scores between 0 to 100), and the weight the patient is using for measuring the importance of this characteristic (higher the weight, the more important it is for the selection, weights between 0 to 1 and summing to 1) are summarized in the following Table 5.6. Actually the patient scored the doctors in terms of word scores which are shown in Table 5.6, and they were later translated into numerical scores between 0 and 100, which are then used for the selection.

The characteristics listed here have been already discussed in Sect. 3.2.

Table 5.6 Score for the available doctors for PCP selection (Appointment waiting time is in days, driving distance in miles)

Characteristics of the doctor		Weight for selection	Evaluations for doctor i=				Score for doctor i=			
			2	3	4	5	2	3	4	5
1	Specialty	0.087	Internal medicine	Family medicine/ general practioner	Family medicine	Pediatrics	100	0	0	30
2	Sanctions	0.087	0	0	0	0	100	100	100	100
3	Lawsuits	0.087	0	0	0	0	100	100	100	100
4	Friends advise	0.087	Recommended	No comment	No comment	No comment	66.7	33.3	33.3	33.3
5	Old/current patients reviews	0.078	Very good	Fair	Very good	Very good	66.7	33.3	66.7	66.7
6	Years of experience	0.069	15	20	20	30	50	86.8	86.8	100
7	Board certification	0.065	Yes	Yes	No	No	100	100	0	0
8	Appointment waiting time	0.065	2	2	3	4	97.2	97.2	92.4	80.2
9	Driving distance	0.056	5	4	2	8	50	79	96.7	5
10	Office hours	0.052	4–8 h/day and 3–5 days a week	4–8 h/day and 3–5 days a week	4–8 h/day and 3–5 days a week	Less than 4 h/day or 3 days a week	50	50	50	0
11	Age	0.052	40	50	55	69	45.3	83.1	97.9	43.1
12	Education	0.043	Top Tier University	Top Tier University	Tier 2 University	Top Tier University	100	100	50	100
13	Gender	0.043	Female	Male	Male	Male	100	0	0	0
14	Ease of parking	0.035	Fair	Easy	Fair	Easy	50	100	50	100
15	No. of specialized service	0.035	1	1	2	0	33.6	33.6	56.1	0
16	Group/individual Service	0.035	Group service	Individual service	Group service	Individual service	0	100	0	100
17	Special training	0.026	No	Yes	No	Yes	0	100	0	100
Combined score for doctor i =							72.5	68	56.4	56.5

Table 5.7 Number of visiting slots available to PCP doctor *i*, by date

Date, beginning with the earliest availability	Number of visiting slots available (a'''_i) on that day
09 July 2012	1
10 July 2012	3

So we see that the available doctor 2, 3, 4, 5, in decreasing order of combined score is, doctors 2, 3, 5, 4, with combined scores of 72.5, 68, 56.5, and 56.4 respectively.

So with the list of available doctors for being selected as the PCP, and the weights for the characteristics and scores given, doctor 2 is the best choice for this patient, followed by doctor 3, 5, 4 in decreasing order of preference.

Example 3. Information that the DSS Provides the Receptionist to Help a Patient Schedule an Appointment with Her/His PCP On any particular date for PCP doctor *i,* the maximum number of patient visits we can schedule on that day is given by a''_i defined in Sect. 6.3. If already some appointments are fixed for this doctor on that day, we denote (a''_i-number of appointments made already) by a'''_i which is continually updated and displayed by day as time goes on.

Using this table, the receptionist will work with the calling patient to select the earliest date and time available for the appointment convenient to him See Table 5.7 for an example.

8 Staffing Decisions

The real-time patient allocation algorithms discussed above already make sure that the total expected workload on for PCP doctors in the hospital matches the workload that they have contracted for.

Decisions Dealing with Hiring of Medical Practitioners From the real-time patient allocation algorithm, we can easily get estimates of the total workload on PCP doctors of each type from existing patient body, and estimates of new patient arrivals. Using these pieces of information the hospital can determine which type of medical practitioners they should plan to hire and when. For example, when all the existing PCP doctors' patient panels are full, it is time to consider hiring.

Acknowledgement Fig. 5.1a–c are from the articles "Castor oil plant", and "*Piptoporus betulinus*" at AMPPURLStarthttp://en.wikipedia.org/wiki/Castor_beansAMPPURLEnd, and AMPPURLStarthttp://en.wikipedia.org/wiki/Piptoporus_betulinus.AMPPURLEnd, Fig. 5.2a–c is from the articles "Ancient Egyptian medicine," "Pyramid," and "Ancient Egyptian Medicine History" at AMPPURLStarthttp://en.wikipedia.org/wiki/Ancient_Egyptian_medicineAMPPURLEnd, http://en.wikipedia.org/wiki/Pyramids, and http://www.egyking.info/2012/05/medicine-in-egyptian-history.html. Fig. 5.3a, b is from the articles "The 'Tribal Medicine Project' (Part 1)" and "Palm-leaf manuscript" at http://savingoxfordmedicine.blogspot.com/2013/08/the-tribal-medicine-project-part-1.html and http://en.wikipedia.org/wiki/Palm-leaf_manuscripts. Please see these articles for detailed information on them.

References

1. Alaeddini, A., Yang, K., Reddy, C., & Yu, S. (2011). A probabilistic model for predicting the probability of no-show in hospital appointments. *Health Care Management Science, 14*(2), 146.
2. Barron, W. M. (1980). Failed appointments: Who misses them, why they are missed, and what can be done. *Primary Care, 7*(4), 563.
3. Bech, M. (2005). The economics of non-attendance and the expected effect of charging a fine on non-attendees. *Health Policy, 74*(2), 181.
4. Cayirli, T., & Veral, E. (2003). Outpatient scheduling in health care: A review of the literature. *Production Operations Management, 12*(4), 519.
5. Dixon, S., Sampson, F., O'Cathain, A., & Pickin, D. M. (2006). Advanced access: More than just GP waiting times? *Family practice, 23*(2), 233.
6. Dockery, F., Rajkumar, C., Chapman, C., Bulpitt, C., & Nicholl, C. (2001). The effect of reminder calls in reducing non-attendance rates at care of the elderly clinics. *Postgraduate Medical Journal, 77*, 37.
7. Gariti, P., Alterman, A., Holub-Beyer, E., Volpicelli, J. R., Prentice, N., & O'Brien, C. (1995). Effects of an appointment reminder call on patient show rates. *Journal of Substance Abuse Treatment, 12*(3), 207.
8. Green, L. (2007). Providing timely access to care: What is the right patient panel size? *The Joint Commission Journal on Quality and Patient Safety, 33*(4), 8.
9. Grumbach, K. (2004). Can health care teams improve primary care practice? *American Medical Association, 291*(10), 6.
10. Hixon, A. L., Chapman, R. W., & Nuovo, J. (1999). Failure to keep clinic appointments: Implications for residency education and productivity. *Family Medicine, 31*(9), 627.
11. Moore, C. G., Wilson-Witherspoon, P., & Probst, J. C. (2001). Time and money: Effects of no-shows at a family practice residency clinic. *Family Medicine, 33*(7), 522.
12. Murray, M. (2003). Improving timely access to primary care case studies of the advanced access model. *American Medical Association, 289*(8), 5.
13. Murray, M., & Davies, M. (2007). Panel size—How many patients can one doctor manage? *Archives of Internal Medicine, 171*(13), 10.
14. Murty, K. G. (2005). Chapter 2 in Junior level optimization models for decision making, Vol. 1. http://ioe.engin.umich.edu/people/fac/books/murty/optimodel/.
15. Rust, C. T., Gallups, N. H., Clark, S., Jones, D. S., Wilcox, W. D., & Adolesc, A. (1995). Patient appointment failures in pediatric resident continuity clinics. *Pediatric and Adolescent Medicine, 149*(6), 693.

Chapter 6
Transportation–Location Problem for a Solar Stove Distributing Nonprofit Organization

Gemma Berenguer, Amber Richter and Zuo-Jun (Max) Shen

1 Introduction

Solar cooking power gives low-income rural and semiurban families an alternative to their traditional cooking methods (usually wood or gas). The adoption of this green technology reduces some major negative effects of the use of traditional methods. In particular, several studies claim that solar cooking reduces the amount of acute respiratory infections in women and children, mitigates environmental degradation related to carbon emissions and deforestation, and promotes gender equality by empowering women as less hours per day are needed for wood collection and less income is required to purchase fuel.

This case study is a simplification of a real-life problem faced by a nonprofit organization (NPO) that distributed solar stoves (Fig. 6.1).The leaders of this organization aimed to make their production and distribution system more efficient and locally beneficial to the communities they serve by reducing shipping costs and enabling certain local production. The statement of the problem and all data sets can be seen at http://extras.springer.com in the folder corresponding to the ISBN number of this book. The data provided approximate the real-life situation that was under investigation but should not be taken as an exact representation of the real problem.

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_6) contains supplementary material, which is available to authorized users.

G. Berenguer (✉)
Purdue University, Lafayette, IN 47907-2056, USA
e-mail: gemmabf@purdue.edu

A. Richter · Z.-J. (Max) Shen
University of California, Berkeley, CA 94270-1777, USA
e-mail: amberr@berkeley.edu

Z.-J. (Max) Shen
e-mail: shen@ieor.berkeley.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_6, © Springer Science+Business Media New York 2015

Fig. 6.1 Solar stove in use

2 Description of the Problem

An NPO that distributes solar stoves in low-income countries is observing inefficiencies in the way they manage their supply chain. Currently, a private company produces the finished products in Mexico. From this production site, the NPO ships the assigned number of assembled solar stoves directly to the beneficiaries of each country. Hence, the current distribution is done on a per country basis in which each shipment goes from Mexico directly to the destination country.

The organization identifies costs as the major decision driver whereas time is not critical. For this reason, water is the transportation mode of choice for the solar stoves between Mexico and the African countries involved in this project. Once the cargo has arrived at the port of entries for the final destination countries, the rest of the solar stove transportation is done with trucks or other types of land vehicles.

While the modes of transport remain the same, the organization wonders whether there might be other ways of distributing the solar stoves that are less costly. One of the major cost components is the cost of shipping containers by sea. Each container has a maximum capacity of 2000 solar stoves. In the past, containers have been shipped under capacity because they were assigned to a single country. The organization wants to incorporate the possibility of consolidating shipments for different countries into the same container.

The solar stove is composed of four major components: the reflector, the glass lid, the black steel pot, and the glass bowl (depicted from top to bottom in Fig. 6.2). Technical requirements for the manufacture of the tempered glass and enamel-coated steel components have prevented local production in any of the destination countries and only the reflector in the version made with heavy cardboard covered with aluminum foil can be manufactured in these countries. Despite the initial high setup cost of reflector production and solar stove assembly plants, the organization strives to foster local production with the long-term objective of achieving self-sustained

Fig. 6.2 Solar stove components from top to bottom: reflector, lid, dark pot, and clear pot. (Source: http://solarcooking.wikia.com/wiki/HotPot)

organic growth of the solar cooking industry in each country. For this reason, the new design of the supply chain should include the possibility of shipping a set of components from the plant in Mexico and locally manufacturing a reflector and assembling the final product in the destination or an intermediate African country. A container transporting sets of components, where each set is composed of the glass lid and bowl and the black steel pot, has a maximum capacity of 6000 units since less space is required per unit. Local production in the African countries would also reduce import taxes since governments are open to applying import duty reductions or even exonerations when components, and not the final product, are imported. Import duties are included in the per unit shipping costs and this is one of the major reasons why shipping costs of the assembled final product are more expensive than shipping costs of the set of components.

We are studying the design of a supply chain composed of three tiers: the production plant in Mexico, an intermediate country (which is the destination of the container vessel), and the final destination country. Only a subset of the final destination countries will be candidate intermediate countries as they need to have a sea port of entry in order to receive a container vessel. Hence, there will be cases where the intermediate and final countries will be the same. According to these tiers, we define three types of flows (which are also pictured in Fig. 6.3):

- The first flow consists of assembled solar stoves that are shipped from the plant in Mexico to the intermediate country and from there to their final destinations.
- The second flow consists of sets of components that are shipped from the plant in Mexico to the intermediate country, where reflector production and final assembly take place. From this country the final product is shipped to its final destination.
- The third flow consists of sets of components that are shipped from the plant in Mexico to the intermediate country and then shipped to their final destinations, where reflector production and final assembly take place.

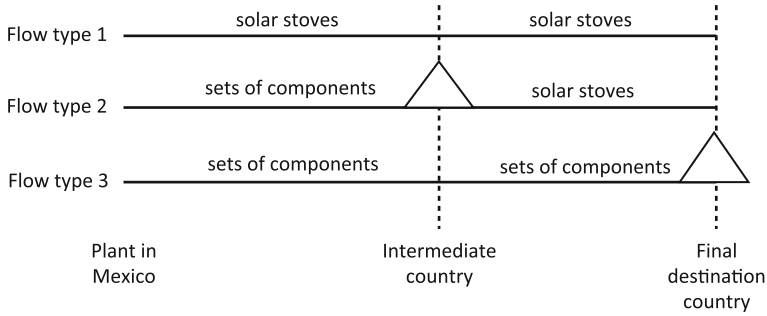


Fig. 6.3 Flow configurations for each final destination country and each candidate intermediate country, where the triangles represent established assembly plants

Note that by assigning an intermediate country we are allowing for the pooling of shipments with different final destinations. This will provide the option to easily fill containers from Mexico to the intermediate countries and thus save in shipping costs. Each container must be filled with a unique type of product, either assembled solar stoves or sets of components.

Costs Shipping costs are in the form of fixed costs per container from the plant in Mexico to the intermediate country, per unit costs from the plant in Mexico to the intermediate country, and per unit costs from the intermediate country to the final destination country. Due to differences in volume, weight, and import duties, *the shipping cost per solar stove is double the shipping cost per set of components*.

The NPO's major function is to coordinate and enable the distribution of solar stoves; it does not directly participate in solar stove production. Thus, besides purchasing the assembled solar stoves and sets of components from the manufacturer in Mexico, the organization supports local entrepreneurs in these intermediate and final destination countries by purchasing their reflectors and partially funding their assembly costs. A low-income country can only assemble sets of components and locally procured reflectors into solar stoves if the NPO establishes an assembly plant in that country. The set up cost per assembly plant for each country is determined based on each country's specific conditions in terms of its economy and the NPO's relationship with the local government and entrepreneurs.

The price of a solar stove manufactured entirely in Mexico is $ 40/unit while the price of a set of components manufactured in Mexico is $ 30/unit. Prices of reflectors and solar stove assembly costs are country-specific. Without including shipping costs, the lump-sum cost of the product assembled in a low-income country is higher than the cost of purchasing the solar stove from the manufacturer in Mexico. Hence, there is a trade-off between the savings obtained by shipping the sets of components instead of the solar stoves and the extra costs incurred by supporting local manufacturing in the intermediate and final destination countries.

Demand The number of solar stoves to be delivered per country is established by the NPO and depends on the adequacy of solar cooking in the country, the support of the local government and partners, and the grants obtained. *This is a static problem with a single period of one year.*

3 Decisions to be Made

The organization needs to determine: (i) the flow of solar stoves and sets of components from Mexico to the intermediate countries and to their final destinations, (ii) the location of assembly plants among the total set of intermediate and/or final destination countries, and (iii) the number of containers of solar stoves and of sets of components shipped to each intermediate country during the planning year.

The problem presented corresponds to the planning phase in which the NPO decides the best distributional strategy for the upcoming year, without an a priori budget to be spent. Once the plan is designed, with its respective costs estimates, the organization applies for the funds needed.

To summarize, the organization's goal is to solve the planning problem that consists of distributing the established amount of solar stoves while minimizing total costs. In summary, the NPO's total costs are composed of (a) shipping costs (including fixed shipping costs per container from Mexico to the intermediate country, per unit shipping costs from Mexico to the intermediate country, and per unit shipping costs from the intermediate country to the final destination country), (b) purchasing costs of solar stoves and sets of components produced at the plant in Mexico and purchasing costs of reflectors produced locally in intermediate or final destination countries, and (c) per unit assembly costs and fixed set up costs per assembly plants at the intermediate and final destination countries.

4 Modeling

This problem can be modeled in several different ways. We will show two different formulations that lead to equivalent optimization problems. Our purpose in showing two different formulations and demonstrating their equivalence is (1) to reinforce the fact that there is no single "right" way of modeling a problem and (2) to improve the active reader's ability to check their formulation for accuracy or to demonstrate that their formulation is equivalent to the formulations provided here and is thus addressing all of the necessary aspects of the problem. For both formulations we assume that the set I is the set of intermediate countries that consist of the six countries that have sea ports of entry: Ghana, Democratic Republic of the Congo (R.D. Congo), Cameroon, Senegal, Kenya, and Tanzania. The set J is the set of final destination countries, which is composed of all 12 African countries. Note that $I \subset J$. The parameters of the problem are defined as follows:

Parameters (Data Elements) for each $i \in I$, $j \in J$

p_j	Price of a reflector produced in country j in dollars
k_i	Cost of shipping a container from the plant in Mexico to country i in dollars per container
f_j	Fixed cost of establishing an assembly plant in country j in dollars
a_j	Per unit assembly cost in country j in dollars per unit
d_i	Shipping cost per solar stove from the plant in Mexico to country i in dollars per solar stove

d_{ij}	Shipping cost per solar stove from country i to country j in dollars per solar stove
$\hat{d}_i (= \frac{d_i}{2})$	Shipping cost per set of components from Mexico to country i in dollars per set of components
$\hat{d}_{ij} (= \frac{d_{ij}}{2})$	Shipping cost per set of components from country i to country j in dollars per set of components
h_j	Demand at destination country j for solar stoves during planning year
M	Very large integer positive number

All these data are given in File 1 for this chapter.

4.1 First Model

The most appropriate way to approach this problem is to model it as an integer linear programming (ILP) model. The ILP formulation described in this section represents the flows of materials as they have been described in Sect. 2 and depicted in Fig. 6.3. Thus, the first type of flow (y_{ij}) is the flow of assembled solar stoves that are shipped from the plant in Mexico to intermediate country i and from there to final destination country j. The second type of flow ($y2_{ij}$) is the flow of sets of components that are shipped from the plant in Mexico to intermediate country i, where they are assembled into solar stoves and then shipped to final destination country j. The last type of flow ($y3_{ij}$) is the flow of sets of components that are shipped from the plant in Mexico to intermediate country i and from there to final destinations country j, where the final assembly takes place.

Decision Variables for each $i \in I, j \in J$

$$z_j = \begin{cases} 1, & \text{if an assembly plant is established at country } j \\ 0, & \text{otherwise} \end{cases}$$

$y_{ij} =$ Flow of solar stoves from the plant in Mexico to country i and from there to country j

$y2_{ij} =$ Flow of sets of components from the plant in Mexico to country i and flow of solar stoves from i to country j

$y3_{ij} =$ Flow of sets of components from the plant in Mexico to country i and from there to country j

Optimization Problem

$$\begin{aligned} Min \sum_{i\in I}\sum_{j\in J} 40y_{ij} + \sum_{i\in I}\sum_{j\in J}(30+p_i+a_i)y2_{ij} + \sum_{i\in I}\sum_{j\in J}(30+p_j+a_j)y3_{ij} \\ + \sum_{j\in J} f_j z_j + \sum_{i\in I} k_i \lceil \frac{\sum_{j\in J} y_{ij}}{2000} \rceil + \sum_{i\in I} k_i \lceil \frac{\sum_{j\in J} y2_{ij} + \sum_{j\in J} y3_{ij}}{6{,}000} \rceil \\ + \sum_{i\in I} d_i (\sum_{j\in J} y_{ij}) + \sum_{i\in I} \hat{d}_i (\sum_{j\in J} y2_{ij} + \sum_{j\in J} y3_{ij}) \end{aligned}$$

$$+\sum_{i\in I}\sum_{j\in J} d_{ij}(y_{ij}+y2_{ij})+\sum_{i\in I}\sum_{j\in J}\hat{d}_{ij}y3_{ij} \quad (6.1)$$

$$s.t. \quad \sum_{i\in I}(y_{ij}+y2_{ij}+y3_{ij}) \geq h_j \qquad \textit{for all } j\in J, \quad (6.2)$$

$$\sum_{j\in J} y2_{ij} \leq Mz_i \qquad \textit{for all } i\in I, \quad (6.3)$$

$$\sum_{i\in I} y3_{ij} \leq Mz_j \qquad \textit{for all } j\in J, \quad (6.4)$$

$$z_j \in \{0,1\},\ y_{ij}, y2_{ij}, y3_{ij} \in \mathbb{Z}^+ \qquad \textit{for all } i\in I, j\in J. \quad (6.5)$$

First of all, note that z_i *for all* $i \in I$ is the subset of the set of variables z_j *for all* $j \in J$ that refers only to the intermediary countries. The objective function (6.1) is composed of the sum of all costs: costs of purchasing solar stoves, sets of components and reflectors, assembly costs (per unit and fixed costs), and shipping costs (per container and per unit). The first set of constraints (6.2) states that the total amount of solar stoves shipped to country j must be greater than or equal to the demand of solar stoves at that country. The second and third sets of constraints (6.3) and (6.4) state that the flow of sets of components that arrive at the intermediate country i or at the final country j, respectively, can only be positive if there is an assembly plant established at that country. Finally, constraints (6.5) are the nonnegativity and integrality constraints.

Note that the ceiling functions $\lceil \frac{\sum_{j\in J} y_{ij}}{2{,}000} \rceil$ and $\lceil \frac{\sum_{j\in J} y2_{ij}+\sum_{j\in J} y3_{ij}}{6{,}000} \rceil$ are nonlinear functions. We can easily linearize them by substituting for the auxiliary variables t_i and t_i' that represent the number of containers of solar stoves and of sets of components shipped to country i, respectively. Subsequently, constraints (6.7), (6.8), and (6.9) should be added to our formulation as follows.

$$Min \sum_{i\in I}\sum_{j\in J} 40y_{ij}+\sum_{i\in I}\sum_{j\in J}(30+p_j+a_j)y2_{ij}+\sum_{i\in I}\sum_{j\in J}(30+p_j+a_j)y3_{ij}$$
$$+\sum_{j\in J} f_j z_j+\sum_{i\in I} k_i(t_i+t_i')+\sum_{i\in I} d_i(\sum_{j\in J} y_{ij})+\sum_{i\in I}\hat{d}_i(\sum_{j\in J} y2_{ij}+\sum_{j\in J} y3_{ij})$$
$$+\sum_{i\in I}\sum_{j\in J} d_{ij}(y_{ij}+y2_{ij})+\sum_{i\in I}\sum_{j\in J}\hat{d}_{ij}y3_{ij} \quad (6.6)$$

$$s.t. \quad \sum_{j\in J} y_{ij} \leq 2000t_i \qquad \textit{for all } i\in I, \quad (6.7)$$

$$\sum_{j\in J}(y2_{ij}+y3_{ij}) \leq 6000t_i' \qquad \textit{for all } i\in I, \quad (6.8)$$

$$t_i, t_i' \in \mathbb{Z}^+ \qquad \textit{for all } i\in I, \quad (6.9)$$

(6.2), (6.3), (6.4), (6.5).

4.2 Second Model

Another way of modeling the problem is to partition each flow into two different sets of variables from the plant in Mexico to the intermediate country and from the intermediate country to the final destination country. The decision variables in this case are defined as follows:

Decision Variables for Each $i \in I, j \in J$

$$z_j = \begin{cases} 1, & \text{if an assembly plant is established at country } j, \\ 0, & \text{otherwise} \end{cases}$$

$x_i =$ Flow of solar stoves from the plant in Mexico to intermediate country i
$x'_i =$ Flow of sets of components from the plant in Mexico to intermediate country i
$w_{ij} =$ Flow of solar stoves from intermediate country i to final destination country j
$w'_{ij} =$ Flow of sets of components from intermediate country i to final destination country j
$q_i =$ Number of solar stoves assembled at intermediate country i
$t_i =$ Number of containers of solar stoves sent from the plant in Mexico to intermediate country i
$t'_i =$ Number of containers of sets of components sent from the plant in Mexico to intermediate country i

Optimization Problem

$$Min \sum_{i\in I} 40x_i + \sum_{i\in I} 30x'_i + \sum_{i\in I} (p_i + a_i)q_i + \sum_{i\in I}\sum_{j\in j} (p_j + a_j)w'_{ij} + \sum_{j\in J} f_j z_j + \sum_{i\in I} k_i(t_i + t'_i) + \sum_{i\in I} d_i x_i + \sum_{i\in I} \hat{d}_i x'_i + \sum_{i\in I}\sum_{j\in J} d_{ij} w_{ij} + \sum_{i\in I}\sum_{j\in J} \hat{d}_{ij} w'_{ij} \tag{6.10}$$

$$s.t. \qquad x_i + q_i = \sum_{j\in J} w_{ij} \qquad \text{for all } i \in I, \tag{6.11}$$

$$x'_i - q_i = \sum_{j\in j} w'_{ij} \qquad \text{for all } i \in I, \tag{6.12}$$

$$\sum_{i\in I} (w_{ij} + w'_{ij}) \geq h_j \qquad \text{for all } j \in J, \tag{6.13}$$

$$\sum_{i\in I} w'_{ij} \leq h_j z_j \qquad \text{for all } j \in J, \tag{6.14}$$

$$q_i \leq M z_i \qquad \text{for all } i \in I, \tag{6.15}$$

$$x_i \leq 2000 t_i \qquad \text{for all } i \in I, \tag{6.16}$$

$$x'_i \leq 6000 t'_i \qquad \text{for all } i \in I, \tag{6.17}$$

$$z_j \in \{0,1\},\ t_i, t_i', x_i, x_i', w_{ij}, w_{ij}', q_i \in \mathbb{Z}^+, \quad \textit{for all } i \in I, j \in J. \tag{6.18}$$

The sets of constraints (6.11) and (6.12) are the flow balance constraints of solar stoves and sets of components, respectively, at each intermediate country. The set of constraints (6.13) is equivalent to constraints (6.2) in the first model. The set of constraints (6.14) states that an assembly plant must be built at a final destination for that country to receive sets of components. The set of constraints (6.15) states that an assembly plant must be built at an intermediate country for that country to be able to assemble sets of components into solar stoves. The sets of constraints (6.16) and (6.17) are equivalent to constraints (6.7) and (6.8) in the first model. Finally, the set of constraints (6.18) defines the decision variables of the model.

Result We prove that both models are equivalent by defining the following equivalence between decision variables:

$$x_i = \sum_{j \in J} y_{ij},\ x_i' = \sum_{j \in J} (y2_{ij} + y3_{ij}),\ q_i = \sum_{j \in J} y2_{ij} \quad \textit{for all } i \in I,$$

$$w_{ij} = y_{ij} + y2_{ij},\ w_{ij}' = y3_{ij} \quad \textit{for all } i \in I, j \in J.$$

5 Solution Methods

The models developed in the previous section are ILP models. Unlike linear programming models, integer programming models can be time consuming to solve to optimality [1]. General purpose optimization software for ILP, and more extensibly for mixed-integer linear programming (MILP), are usually based on branch-and-bound approaches. The amount of computer time needed to find an exact optimum solution to an MILP using these approaches increases as the number of integer variables in the model grows. This is challenging for certain types of problems with a large number of nodes in the supply chain but it does not represent a major challenge in the context of our problem. According to all these conditions, the software package that we have chosen to use to solve our models is mentioned in Sect. 5.1.

Nonetheless, we note that we are analyzing a one-time strategic decision with a 1-year lifespan. Therefore, this problem will only need to be solved once during this period of time. For this reason, it may not be cost-effective for our NPO to hire an expert in operations research and purchase a software license. In this case, the problem can be solved using a heuristic algorithm that may not necessarily be guaranteed to find an optimal solution. We develop two heuristics in Sect. 5.2 as alternative solution methods to be used if the NPO does not have access to the appropriate professional and/or software license. It is advantageous, if an exact solution method is unavailable, to solve the problem with more than one heuristic to be able to find the best solution possible.

5.1 Exact Algorithm

It is ideal for our NPO to solve this problem to optimality. General purpose software is capable of solving this problem to optimality as the problem is relatively small and has a network flow structure that the software can take advantage of. For our case, we have employed GAMS/CPLEX (version 12.4) to solve our problem. Branch and cut and dynamic search are the methods most generally used by these solvers for this type of problem.

5.2 Heuristic Algorithms

In this section, we develop two heuristics as alternative solution methods to be used if the NPO does not have access to a software license. These heuristics are similar to those developed by [2] for the uncapacitated fixed charge facility location problem. We conclude these algorithms with the addition of a last step procedure that develops an improved solution from the one obtained initially.

We observe that one of the most challenging decisions to be made in order to solve the problem is to determine the subset of binary variables z_j equal to 1 in the optimum solution. For this reason, in the heuristic methods discussed below for our problem, the main strategy used is to determine good choices for the subset of binary variables z_j which should be set to 1. In addition, we observe that the only capacity restriction in this problem is on the capacity of the containers; there are no capacity restrictions on production or assembly levels at any of the plants nor on transportation quantities between intermediate countries and final destination countries. Thus, this problem can be greatly simplified by disregarding this capacity requirement at first; later we will take this capacity requirement back into account during the last step procedure. Thus, for now, we will assume that all solar stoves (or their components) assigned to a final destination country will all flow through the same intermediate country, or in other words, single sourcing. Similarly, all of the stoves going through the designated intermediate country to the final destination country will be of the same type of flow, where the three possible types of flows are as described in Sect. 2 and depicted in Fig. 6.3.

Within this framework, let us construct two greedy improvement heuristics to solve our problem. We begin by defining some additional notation. Let us approximate the cost of serving final destination country j from intermediate country i through flow type $k = 1, 2$, and 3 by J_{ij}^k as defined below:

$$J_{ij}^1 = (40 + \frac{k_i}{2000} + d_i + d_{ij})h_j \tag{6.19}$$

$$J_{ij}^2 = (30 + \frac{k_i}{6000} + \hat{d}_i + d_{ij} + p_i + a_i)h_j \tag{6.20}$$

$$J_{ij}^3 = (30 + \frac{k_i}{6000} + \hat{d}_i + \hat{d}_{ij} + p_j + a_j)h_j \tag{6.21}$$

Thus, we approximate the cost of serving final destination country j from intermediate country i by using a linear continuous approximation of the fixed container costs in addition to the other per unit manufacturing, assembly, and shipping costs. From Fig. 6.3 we see that for flow type 2 to be possible, an assembly plant must be established at intermediate country i. Using the notation established in Sect. 4, this is when $z_i = 1$. Similarly, for flow type 3 to be possible, an assembly plant must be established at final destination country j, or when $z_j = 1$. Then,

Result We can represent the approximated minimum cost to serve final destination country j from intermediate country i by

$$min[J_{ij}^1, (z_i + M(1 - z_i))J_{ij}^2, (z_j + M(1 - z_j))J_{ij}^3] \tag{6.22}$$

and the approximate cost to serve all final destination countries by

$$\sum_{j \in J} \{min_{i \in I}[min(J_{ij}^1, (z_i + M(1 - z_i))J_{ij}^2, (z_j + M(1 - z_j))J_{ij}^3)] + p_j z_j\} \tag{6.23}$$

Heuristic 1: The ADD Algorithm

With this notation and these approximate costs established, we can now describe the greedy add heuristic. As this heuristic is an improvement heuristic, we must first find a feasible solution from which to begin. We do this by assuming that no assembly plants are established (i.e., by setting all z_j equal to 0) and solving Eq. (6.23) for the approximate cost to serve all final destination countries. Let us denote this as our current approximate total cost TC. For each final destination country j, the intermediate country i which satisfies the minimum in the sum in Eq. (6.23) is the current best source country for final destination country j. Thus, in this case, since all the z_j equal 0, for each final destination country j the current best source intermediate country is the intermediate country i which has the minimum value of J_{ij}^1.

Now that we have a feasible solution to improve upon, we can implement the greedy add heuristic. This heuristic adds assembly plants to the current solution until no assembly plant can be added which will reduce the current approximate total cost. We call this heuristic the "greedy" add heuristic as in each iteration we choose to add the assembly plant which reduces the approximate total cost the most while keeping all other previously selected plants fixed. To determine where an assembly plant should be added, we calculate the cost savings if we add an assembly plant at each country j as the maximum between zero and the difference between TC, the current approximate total cost, and the value achieved by Eq. (6.23) when evaluating it with $z_j = 1$ and all other z variables fixed to their value in the current solution. The country for which adding an assembly plant achieves the greatest cost savings is the country where we chose to establish the next assembly plant. The heuristic continues until no more assembly plants can be added to decrease the approximate total cost. From this final total approximate cost we can extract the best source country and flow type for each final destination country. The heuristic is summarized in Fig. 6.4.

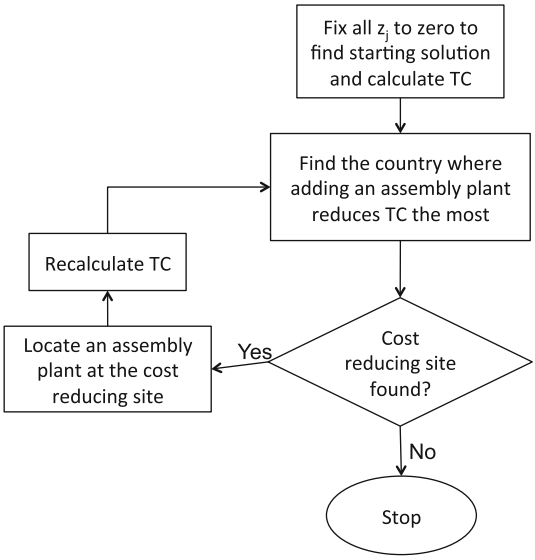


Fig. 6.4 Greedy add heuristic

Heuristic 2: The DROP Algorithm

The greedy drop heuristic is carried out in a similar manner to the greedy add heuristic. An initial feasible solution is found when assembly plants are located at each country (i.e., by setting all z_j equal to one) and solving Eq. (6.23) for the approximate cost to serve all final destination countries. The greedy drop heuristic then can be applied. In each iteration, the heuristic removes (or "drops") assembly plants from the current solution until no assembly plant can be removed to reduce the approximate total cost. We call this heuristic the "greedy" drop heuristic as in each iteration we choose to remove the assembly plant which reduces the approximate total cost the most while keeping all other previously selected plants fixed. The heuristic continues until no more assembly plants can be removed to decrease the approximate total cost. The heuristic is summarized in Fig. 6.5.

Last Step of the Heuristic

The Add and Drop greedy heuristics described above do not allow for multiple sourcing nor do they fully take into account the effects of the fixed container costs. Thus, there is still some room for improvement of the solutions. We will now develop a simple procedure to improve the solution found by the heuristics. To keep it simple, we will preserve the best flow types and assembly plant establishment decisions made by the heuristics and reflected in the solution found by the heuristics.

The last step procedure can be described as follows. For reducing the number of containers used, define K as the set of all intermediate countries that received containers with unused capacity in the current solution. Start with the intermediate country i in this set which has the greatest amount of unused capacity available in its less than full container. For this intermediate country i, we will try to find a

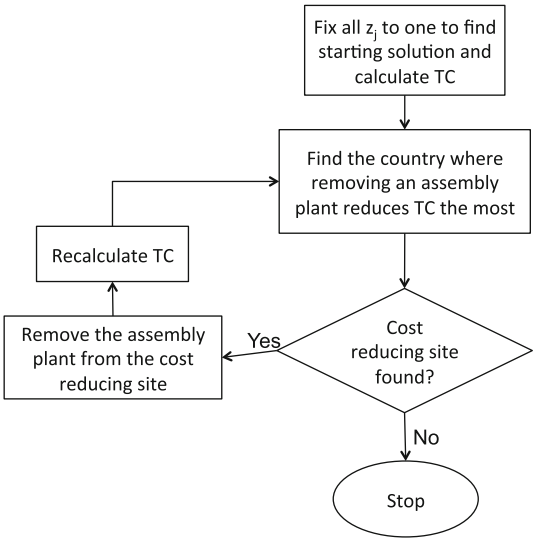


Fig. 6.5 Greedy drop heuristic

way to move the contents of the less than full container to another less than full container to try to reduce shipping costs. We will do this as follows. For each of the final destination countries j sourced by intermediate country i in the current solution whose demand is greater than or equal to the contents of the less than full container, calculate the total cost of sourcing that much of its demand from another intermediate country which:

1. Has enough unused space in its less than full container, and
2. Has the ability to service j with the same type of flow while keeping all other decisions in the solution fixed

If any of these moves provides a solution which has total cost less than the current total cost, make the move that provides the greatest reduction in total cost (we note that here we are referring to the actual total cost, as calculated in Eq. (6.6) or (6.10), and not the approximate total cost we used during the greedy add and drop heuristics). Then, remove i from K, update the capacity statistics of the intermediate countries, and move on to the next intermediate country in K which has the greatest amount of unused capacity available in its less than full container. If none of the moves provides a cost reducing solution, remove country i from K, and move on to the next intermediate country in K which has the greatest amount of unused capacity available in its less than full container. Continue until the set K is empty. This procedure is summarized in Fig. 6.6.

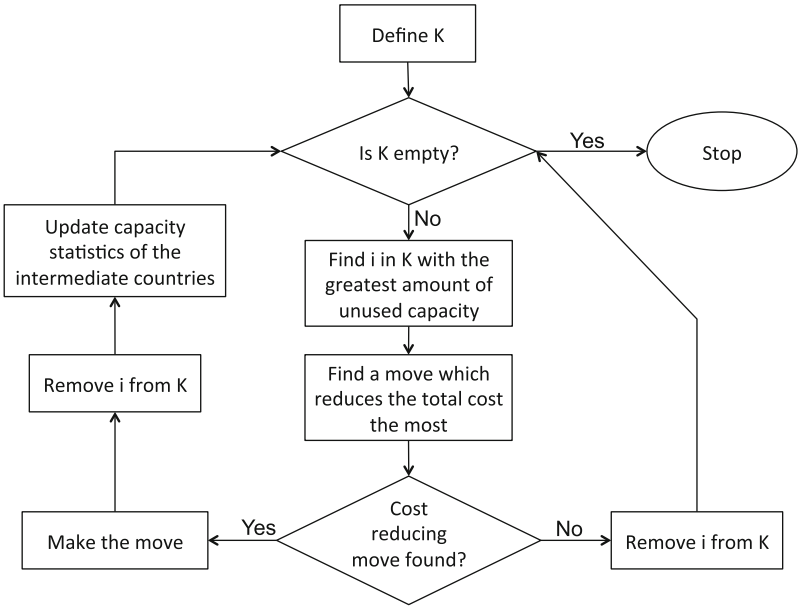


Fig. 6.6 Last step procedure

6 Solutions Obtained

6.1 Solution of Exact Algorithm

GAMS/CPLEX provides the optimal solution of the problem in less than one second of CPU time. The details of the optimal solution are explained in the following list:

- The optimal total cost is $ 5,696,450.
- Assembly plants need to be placed in Cameroon, Kenya, and Tanzania. The number of solar stoves assembled at the assembly plants at intermediate or final destinations is 12,500 in Cameroon, 15,000 in Kenya, and 20,500 in Tanzania.
- There is a total of 22 containers of solar stoves shipped in full from Mexico to D.R. Congo (7), Ghana (3), and Senegal (12).
- There is a total of eight containers of sets of components shipped in full from Mexico to Cameroon (2), D.R. Congo (2), Kenya (1), and Tanzania (3).
- Four countries receive units from multiple intermediate countries:
  - Cameroon receives 500 sets of components from D.R. Congo and 7500 sets of components from its own sea port.
  - Tanzania receives 2500 sets of components from D.R. Congo and 5500 sets of components from its own sea port.
  - Uganda receives 6000 solar stoves from Kenya and 1500 solar stoves from Tanzania.
  - Zimbabwe receives 2000 solar stoves from D.R. Congo and 8000 from Tanzania.

Table 6.1 Greedy add heuristic iterations

Iteration	Total approximate cost	Assembly plant added	Cost savings
1	$ 5,781,750	Kenya	$ 68,916.67
2	$ 5,712,833.33	Tanzania	$ 14,083.33
3	$ 5,698,750	Cameroon	$ 3,708.33
4	$ 5,695,041.67	N/A	N/A

- The rest of the countries receive units from a unique source (intermediate country)
  - Burkina Faso receives 7000 solar stoves from Senegal.
  - Chad receives 4500 solar stoves from Cameroon.
  - D.R. Congo receives 12,000 solar stoves from its own sea port.
  - Ghana receives 6000 solar stoves from its own sea port.
  - Kenya receives 9000 sets of components from D.R. Congo.
  - Mali receives 10,000 solar stoves from Senegal.
  - Rwanda receives 3000 solar stoves from Tanzania.
  - Senegal receives 7000 solar stoves from its own sea port.

6.2 Solutions of Heuristic Algorithms

Both heuristics can be conveniently implemented manually using © Microsoft Excel. It turns out that, due to the simplicity of the cost approximations and independence between countries in the greedy add and greedy drop heuristics, both heuristics find the same solution pre-last step. A summary of the major details of this solution is given below:

- The corresponding total cost for this solution is $ 5,758,375, which is within 1.087 % of the optimal cost.
- The solution has the same three assembly plants as the optimal solution.
- From the heuristic's design, all countries receive units from a unique source (intermediate country). We show only those flows that are different from the optimal solution:
  - Cameroon receives 8000 solar stoves from its own sea port.
  - Tanzania receives 8000 solar stoves from its own sea port.
  - Uganda receives 7500 solar stoves from Kenya.
  - Zimbabwe receives 10,000 solar stoves from Tanzania.

A summary of the iterations for each heuristic is provided in Tables 6.1 and 6.2. The total approximate cost refers to the value obtained from Eq. (6.23).

Next, we note that in the solution found by the heuristics, there are several cases of less than full containers which are summarized in Table 6.3. We also note that all of these cases are for containers of sets of components and all containers containing solar stoves are used to full capacity. Since the solution found by the heuristics only has less than full containers of sets of components, we have only developed

Table 6.2 Greedy drop heuristic iterations

Iteration	Total approximate cost	Assembly plant removed	Cost savings
1	$ 5,707,541.67	Senegal	$ 1500
2	$ 5,706,041.67	Mali	$ 1500
3	$ 5,704,541.67	Burkina Faso	$ 1500
4	$ 5,703,041.67	Chad	$ 1500
5	$ 5,701,541.67	D.R. Congo	$ 1500
6	$ 5,700,041.67	Zimbabwe	$ 1500
7	$ 5,698,541.67	Rwanda	$ 1500
8	$ 5,697,041.67	Uganda	$ 1500
9	$ 5,695,541.67	Ghana	$ 500
10	$ 5,695,041.67	N/A	N/A

Table 6.3 Summary of cases of less than full containers in the solution found by the add and drop heuristics

Intermediate country	Content type	Unused capacity	Capacity utilized
Cameroon	Sets of components	5500	500
D.R. Congo	Sets of components	3000	3000
Kenya	Sets of components	4500	1500
Tanzania	Sets of components	3000	3000

Table 6.4 Last step procedure iterations

Iteration	Total cost	K	Country with greatest unused capacity	Best cost reducing move
1	$5,758,375	{Cameroon, D.R. Congo, Kenya, Tanzania}	Cameroon	Five hundred to Chad now from Kenya instead of Cameroon
2	$5,740,500	{D.R. Congo, Kenya, Tanzania}	Kenya	Two thousand to Uganda now from Tanzania instead of Kenya
3	$5,712,600	{D.R. Congo, Tanzania}	D.R. Congo	None
4	$5,712,600	{Tanzania}	Tanzania	None

the last step procedure for the case of a single type of less than full container. A similar procedure can be developed for the case when containers of solar stoves and containers of sets of components are less than full.

In Table 6.4 we show a summary of the iterations for the last step procedure implemented on the solution found using the add and drop heuristics. We obtain a solution of $ 5,712,600, which is within 0.284 % of the optimal cost. This improved solution has some cargo redistributed, as described in the last column of Table 6.4, from the solution provided by the add and drop heuristics but is otherwise the same.

6.3 Brief Comparison of Different Solutions

In summary, we have obtained three different solutions that are rather similar to each other though they have a definite ranking in terms of which one is best. We describe

Table 6.5 Solutions obtained from different methods

	Exact algorithm	Heuristics	Heuristics with final step
Objective	$ 5,696,450	$ 5,758,375	$ 5,712,600
% error	0	1.087 %	0.284 %
Multiple sourcing	Cameroon, Tanzania, Uganda, Zimbabwe	0	Chad, Uganda
Less than full containers	0	4	2
Spare container capacity	0	17.4 %	4.3 %

them as follows. The solution provided by the add and drop heuristics is the best solution given that no multiple sourcing is allowed and no special consideration to less than full containers is given. The addition of the last step procedure is key to improving the solution since it reduces the amount of less than full container units substantially by allowing multiple sourcing. Thus, this procedure takes care of the two considerations that were disregarded in the heuristics. Better than both of these solutions in terms of the objective value is the solution obtained using the exact algorithm, though the gap between the other two solutions and the optimal solution is small. Table 6.5 shows the major characteristics of the three solutions obtained and confirms that the solutions are relatively close to each other.

7 Summary

In this chapter, we describe a transportation–location problem of an NPO that aims to efficiently distribute solar stoves to a number of different African countries. Local production costs and shipping costs are the main conflicting objectives. We describe the problem with two different models that are equivalent. The first model describes the flows of materials from the initial node (the plant in Mexico) to the end nodes (the final destination countries) of the supply chain, while the second model partitions the flows into two subflows (from the plant in Mexico to the intermediate countries and intermediate countries to the final destination countries).

We solve the problem in two different ways. For our exact algorithm, we employ the most advanced operations research software tools to find the optimal solution of our problem. On the other hand, we note that the NPO might not have the same human and computer resources to address the problem as academics or private for-profit companies might have. For this reason, we suggest two improvement heuristics as alternative methods that are not guaranteed to produce an optimal solution but do provide a good enough solution in practice. The last step procedure embedded in the heuristic algorithm is key since it looks for an improved solution by allowing multiple sourcing while reducing the amount of less than full containers. If the appropriate resources are available, it is best to use the solution method that is guaranteed to find an optimal or close to optimal solution.

Other heuristic algorithms can be designed that can provide good enough solutions. The quality of other greedy heuristics will depend on the greediness criteria

employed. Furthermore, in practice, other factors might need to be considered in order to come up with the best solution for this organization's problem. For example, it might be important to consider aspects related to transportation reliability in which some routes between different countries might be unreliable, more expensive or more time consuming than the parameters estimated a priori or assumptions reflected in our model. Another factor to further evaluate is local production, where quality issues might arise which could lead to spikes in production costs. The models we have developed in this chapter may not appropriately rank solutions in terms of these factors, and thus there is a great advantage to finding multiple optimal or close to optimal solutions in order to rank them in terms of other less important or less quantifiable factors. If factors such as these are deemed sufficiently important and quantifiable, it may be appropriate to take them into account in the modeling of the problem.

8 Project Exercises

1. Assume that the NPO has a limited budget of $ 5,000,000 while all other problem parameters remain the same and that the objective is now to maximize the number of recipients of solar stoves in the destination countries subject to this budget constraint. Model this problem as an ILP, explain your objective function and constraints, and solve the problem using optimization software.
2. For the problem variation described in question 1, briefly describe an idea for a heuristic to solve the problem without optimization software.
3. Intuitively, describe the effect on the optimal objective value and optimal solution of increasing the cost of a solar stove manufactured entirely at the plant in Mexico from $ 40 to 60 while all other problem parameters remain the same. Solve the problem with this change to test your intuition. Was your intuition correct?
4. Intuitively, describe the effect on the optimal objective value and optimal solution of increasing the capacity for containers of fully assembled stoves from 2000 to 4000 while all other problem parameters remain the same. Solve the problem with this change to test your intuition. Was your intuition correct?

References

1. Bertsimas, D., & Tsitsiklis, J. N. (1997). *Introduction to linear optimization*. Belmont: Athena Scientific.
2. Daskin, M. S. (1995). *Network and discrete location: Models, algorithms, and applications*. New York: Wiley.

Chapter 7
Designing Earth Dams Optimally

G. S. R. Murthy, Katta G. Murty and G. Raghupathy

1 Introduction

A dam is a barrier that impounds water or underground streams. Dams generally serve the primary purpose of retaining water, while other structures such as floodgates or levees (also known as dikes) are used to manage or prevent water flow into specific land regions. Hydropower and pumped-storage hydroelectricity are often used in conjunction with dams to generate electricity. A dam can also be used to collect water, or for storage of water which can be evenly distributed among locations. See the sites http://en.wikipedia.org/wiki/Dam and http://osp.mans.edu.eg/tahany/dams1.htm for a detailed discussion on dams.

An earth dam is a dam built with highly compacted soils. It is classified as a type of embankment dam, being built in the shape of an embankment or wedge which blocks a waterway. In addition to soil, other materials such as rock and clay are also used in earth dams. Earth dams have been built by various human societies for centuries as they are most cost effective and are built from materials available in nearby locations. Safety of earth dams is a crucial issue (http://www.wisegeek.com/what-is-an-earth-dam.htm). Failure of moderate or large size earth dams can cause severe damage to life and property of the people living in the nearby areas. Overtopping,

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_7) contains supplementary material, which is available to authorized users.

G. S. R. Murthy (✉)
SQC & OR Unit, Indian Statistical Institute, Street No. 8, Habsiguda,
Hyderabad 500007, India
e-mail: murthygsr@gmail.com,

K. G. Murty
Department of Industrial and Operations Engineering,
University of Michigan, Ann Arbor, MI 48109-2117, USA
e-mail: murty@umich.edu,

G. Raghupathy
HES Infra Pvt Limited, Hyderabad 500034, India
e-mail: raghupathi.hes@gmail.com

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_7, © Springer Science+Business Media New York 2015

seepage, and structural failures are the problems encountered due to poor designing and maintenance.

Engineering design of an earth dam is a crucial issue from the view point of safety, and economy of construction cost. Evaluation of safety of a given design involves intensive computation. Despite the advances in soft computing and development of software for analyzing design safety, their utilization in practical applications appears to be limited. Perhaps accessibility, cost considerations, awareness, and user-friendliness could be some of the reasons for this. Under the present practices, the emphasis is laid more on ensuring safety, and the economic aspects are often given little or no consideration while designing the dams. The concept of looking for an optimal design itself appears to be a novel idea for many designing engineers in practice.

This work deals with producing optimal designs for earth dams. It has evolved from the efforts of a senior engineer (Raghupathy, one of the authors of this chapter) who was working as Superintending Engineer, Department of Irrigation, Government of Andhra Pradesh, India, and was responsible for designing earth dams. Following a scientific approach, it aims at formulating the problem of designing earth dams as an optimization problem. The problem is formulated as a nonlinear program with dam cross-sectional area—which represents the major portion of cost of construction—as the objective function and safety factor of the design as the main constraint. In order to study the problem, it was necessary to develop a computer program to compute the factor of safety for a given design. The engineers find this program extremely useful in the preparation and analysis of earth dam designs and are using it as a decision support system.

The outline of this chapter is as follows. Section 2 presents a brief interesting overview of history of dams. Section 3 presents a short discussion on different types of dams and safety aspects. Section 4 introduces the description of earth dams and basic terminology to set up the stage for the optimization problem in question. Section 5 presents concept and definition of factor of safety of an earth dam. Section 6 deals with the computational aspects of safety and some innovative ideas of computing factor of safety. Section 7 provides mathematical modeling for optimizing earth dam designs and for computing the factor of safety. It also discusses issues in obtaining optimal solutions to the formulations, presents a methodology for determining good designs, and the results of the application of the methodology to a live earth dam design. Section 8 summarizes the chapter. Section 9 lists some exercises for learning experience, which is followed by references.

2 Brief History of Dam Building

Dams are barriers constructed across streams (above ground or underground) to impound water or the underground streams. Any discussion on the history of dams is incomplete without a mention of beavers (industrious furry animals of the rodent family who live under water) that build astonishing water impounding structures across streams (see Fig 7.1) and rivulets using tree branches which they cut themselves, chopped wood, twigs, and mud; to provide themselves with comfortable ponds to live in.

Fig. 7.1 A hard working beaver building a dam. For details see the source "Images for beaver dams" at https://www.google.co.in/search?q=beaver+dams

The earliest human dam builders may have received their inspiration from beavers; human dam construction dates back to earlier than 5000 BCE and was practiced by several civilizations across the globe; the earliest perhaps in Mesopotamia, Middle East and India.

The Jawa dam in Jordan, 100 km Northeast of Amman, (a gravity dam, 9 m high, 1 m wide stone wall supported by 50 m earth rampart) is dated to 3000 BCE.

In ancient India, the Indus valley civilization in Mohenjodaro and Harappa going back even earlier than 3000 BCE built a dam on the Saraswati (also referred to as the Ghaggar-Hakra) river and flourished to a high state of development, but that civilization scattered and vanished without any trace, for reasons not fully understood today; one of the contributing factors is perhaps the drying up and eventual disappearance of the river Saraswati.

In third century BCE, an intricate water storage and management system consisting of 16 reservoirs, dams, and various channels for collecting, storing, and distributing water was built in modern day India at Dholovira in Gujarat state in western India.

Romans introduced the then novel concept of large reservoir dams which could secure water supply for urban settlements year round including the dry season, by pioneering the use of water-proof hydraulic mortar and Roman concrete for much larger dam structures than previously built, such as the Lake Homs dam in Roman Syria in 284 CE, with a capacity of 90 million m^3 and is in use even today.

The Kallanai Dam (Fig 7.2) across the main stream of the Kaveri river in Tamilnadu, South India, constructed of unhewn stone and 300 m long, 4.5 m high, and 20 m wide, dated to the second century CE, considered the oldest water diversion structure in the world, is still in use for diverting the waters of the Kaveri across the fertile delta region for irrigation via canals.

Dujiangyan, finished in 251 BCE, a large earthen dam in modern-day northern Anhi province in China, created an enormous irrigation reservoir 100 km in circumference, a reservoir still present today (Fig. 7.3).

Fig. 7.2 The Kallanai dam in India. For details see the source "Images for Kallanai dam" at https://www.google.co.in/search?q=Kallanai+dam+pictures

Fig. 7.3 Dujiangyan dam in China. For details see the source "Images for Dujiangyan Dam" at https://www.google.co.in/search?q=Dujiangyan+Dam+pictures

Also, in China is the Three Gorges dam, a hydroelectric dam made of concrete and steel, spanning the Yangtze river by the side of the town of Sandouping in Hubei province, the world's largest power station in terms of installed capacity (22,500 MW). It is also intended to increase the Yangtze river's shipping capacity and reduce the potential for flood downstream by providing flood water storage space. Before the construction of this dam, the Yangtze river floods used to be an annual phenomenon ravaging that area in China, and after witnessing these floods in 1954, then Chairman Mao Zedong authored a poem titled "Swimming" and initiated plans to construct this dam. The dam body was completed in 2006, and the dam project was completed and fully functional as of 4 July 2012. This project is a historic engineering, social, and economic success. Project plans also included a unique method for moving ships, the ships will move into docks located at the lower and upper ends of the dam, and then cranes with cables would move the ships from one

Fig. 7.4 Itaipu dam in South America, considered as one of the seven modern wonders of the world. For details see the source "Images for Itaipu dam" at https://www.google.co.in/search?q=itaipu+dam+pictures

Fig. 7.5 The Hoover dam. For details see the source "Images for hoover dam" at https://www.google.co.in/search?q=hoover+dam+pictures

dock to the next. The ship lift is expected to be completed in 2014. The dam has raised the water level in the reservoir to 1725 m above sea level by October 2010. The expected annual electricity output will be over 100 TWh.

The Itaipu is a hydroelectric dam on the Parana river located on the border between Brazil and Paraguay, its name comes from that of an isle that existed near the construction site (in the local Guarani language, the word "Itaipu" means "the sound of a stone"). This dam is the largest operating hydroelectric facility in terms of annual energy generation, generating 94.7 TWh in 2008; its installed capacity of 14,000 MW is second to that of the Three Gorges dam. This dam is considered to be one of the seven modern wonders of the world; it is a concrete and steel dam (Fig. 7.4).

The era of large dams was initiated after Hoover dam which was completed on the Colorado river near Los Vegas in 1936. By the end of the twentieth century, there were an estimated 800,000 dams worldwide, of which about 40,000 are over 15 m high (Fig. 7.5).

2.1 The Collapse of a Dam

Ancient people who lived in the present day Egypt at the time of the Pharoas, constructed the Sadd-el-Kafara dam at Wadi-al-Garawi (25 km South of Cairo), a 102 m long at the base and 87 m wide structure, around 2800 BCE, as a diversion dam for flood control; unfortunately this structure had a very short life as it was destroyed by very heavy rains a few years after its construction. When a dam gets washed away, its effects could be devastating resulting in great damage to life and property of the people living in the downstream areas of the dam. For this reason, it is of utmost importance to ensure that the dams are safe. This chapter deals with one of the important aspects of safety of a particular type of dams—the earth dams.

3 Types of Dams

There are several types of dams. The types may be classified by type of materials used for construction, the height of the dam, the purpose for which the dam is built, and so on. Broadly, dams can be divided into two types: embankment dams and concrete dams. Embankment dams are earth or rock-filled; while gravity, arch, and buttress dams are made of concrete. The type of dam for a particular site is selected on the basis of technical and economical data and environmental considerations. Factors that determine the type of a dam are topography, geology, foundation conditions, hydrology, earthquakes, and availability of construction materials. The foundation of the dam should be as sound and free of faults as possible. Narrow valleys with shallow sound rock favour concrete dams. Wide valleys with varying rock depths and conditions favor embankment dams. Earth dams are the most common type of dams (http://encyclopedia2.thefreedictionary.com/dam).

3.1 Earth Dams

In this chapter we are mainly concerned with earth dams. Earth dams are constructed of well compacted earth, whose cross-section shows a shape like a bank, or hill. A uniformly rolled-earth dam is entirely constructed of one type of material but may contain a drain layer to collect seep water. A zoned-earth dam has distinct parts or zones of dissimilar material, typically a locally available shell with a watertight clay core composed of an impermeable material to stop water from seeping through the dam. Since they exert little pressure on their foundations, they can be built on hard rock or softer soils. Compared to concrete dams, earth dams are most economical. To give a rough idea, the cost of a concrete dam is about 20 times that of an earth dam of comparable size.

3.2 Safety of Dams

Throughout history, there have been several instances of dam failure and discharge of stored water, usually causing considerable loss of life and great damage to property. The main causes of dam failure include spillway design error (South Fork dam), geological instability caused by changes to water levels during filling or poor surveying (Vajont dam, Malpasset), poor maintenance, especially of outlet pipes (Lawn Lake dam, Val di Stava dam collapse), extreme rainfall (Shakidor dam), and human, computer, or design error (Buffalo Creek Flood, Dale Dike Reservoir, Taum Sauk pumped storage plant, (http://en.wikipedia.org/wiki/List_of_dam_failures)). As the dam failures caused by the structure being breached or significantly damaged are generally catastrophic, it is extremely important to analyze the safety of a dam properly. During a recent flood, the reservoir level of the famous Nagarjun Sagar dam in Andhra Pradesh, India, reached alarming heights threatening dam failure. The efficient management and timely actions by the concerned engineers averted a possible catastrophe of unimaginable proportions.

4 Earth Dam Description and Terminology

One of the most important steps while building a dam is to ensure that it is safe well beyond doubt, as any failure of the dam results in calamities. Safety depends on the materials used, the method and quality of construction, and above all, the engineering design of the dam. In fact, the design parameters can be chosen to offset, to a reasonable extent, the lacunae in the qualities of materials, method and construction. The measure for safety is called the *factor of safety* and has a precise mathematical definition. It depends upon the design parameters, the material qualities, and implicitly on the method of construction. The optimization problem being considered in this chapter aims at deriving the design parameters in the most economical way without compromising the factor of safety. To facilitate a good understanding of the problem and its solution, it is essential to have a detailed description of an earth dam, its structure and design parameters, the mathematical definition of factor of safety, and its evaluation and the underlying assumptions. This section describes the earth dam and its design parameters and the underlying assumptions made for determining the factor of safety. The next two sections are devoted for the precise mathematical definition of factor of safety and the methods for evaluating the same.

4.1 Earth Dam Description

Throughout this chapter, the units of measurement of various quantities will be as follows:

Units for Measurement All the linear measurements (length, width, depth, and height) will be in meters (m); areas in square meters (m^2); volume in cubic meters

Fig. 7.6 Picture of an earth dam

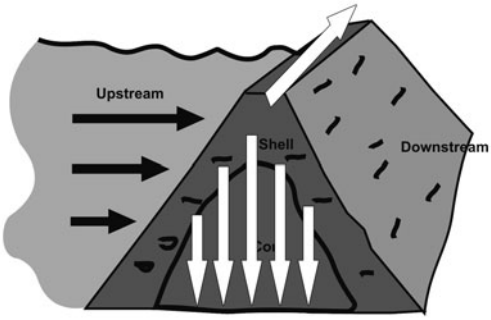


Fig. 7.7 Cross-section of an earth dam

(m^3); time units in seconds; force units in kgF (one kgF is 9.8 Newtons). Weight of an object will be measured by its gravitational force, its unit is kgF. In this chapter we use densities of material for computation of factor of safety. These densities are the weight densities. *Weight density* is defined as the weight per unit volume and the unit will be kgF/m^3.

As described earlier, an earth dam is constructed of compacted earth in a structured way, usually between two elevated places (see Fig 7.6). Figure 7.7 is drawn to show the picture of cross-sectional view of an earth dam. In this picture, the top arrow is along the length of the dam. The left hand side of the dam in the picture is that side of the dam where the water is impounded by the dam. The horizontal arrows indicate the natural flow of water. This side of the dam is called the *upstream of the dam*. The right side of the dam in the picture, horizontally opposite to the upstream, is called the *downstream of the dam*. The front portion of the picture shown by the vertically downward arrows is called a *cross-section of the dam* at a point along the length of the dam. Broadly speaking, earth dams comprise two components—the core and the shell. The interior portion of the dam is called the *core* and the portion of the dam surrounding it is called the *shell* (see Fig 7.7). The purpose of the core is to arrest the seepage of water from upstream side to downstream side. For this purpose, the core is constructed using materials of low permeability such as clay. In certain earth dams, the core may not be present.

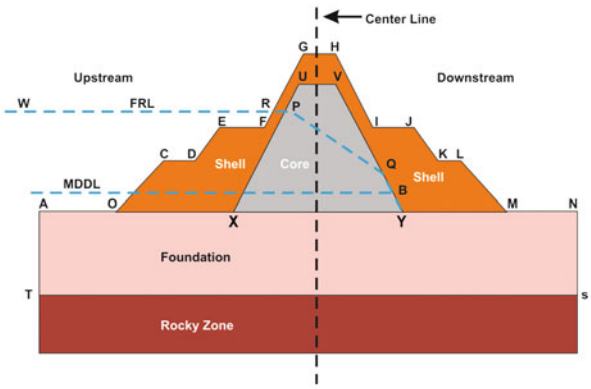


Fig. 7.8 Basic structure of earth dam, cross-sectional view. The terms in this figure are explained in Sect. 4.3

4.2 Cross-Section of Earth Dam

In Fig 7.7, the top portion of the dam is shown as a path. Technically speaking, the elevation of the upper most surface of a dam, usually a road or walkway, excluding any parapet wall, railings, etc, is called the top of the dam. The vertical height from the ground level to the top of the dam is called the *dam height*. The road or walkway at the top of the dam is used for passages, inspections of the dam, and so on. When the dam height is large, one needs to have accessibility to various points on the dam. For this purpose, roads or walkways are also built at different altitudes on the two slanting surfaces—the upstream surface and the downstream surface—of the dam. One such road/walkway is visible on the upstream side in Fig 7.6. As per the conventional norms, a dam is supposed to have a road/walkway at every 15m altitude. However, this is not a hard and fast rule. For instance, the Khandaleru dam in Andhra Pradesh, India, is 50 m high, yet it has only one road at the top of the dam and no road/walkways anywhere else on the dam.

If we look at the cross-sectional view of a dam (that is, a person viewing from the ground at a point orthogonal to the water flow direction so that the core and shell are visible), then the picture of the cross-section would look like the one in Fig 7.8. Any road/walkway on the slanting surfaces of the dam is called a *berm*. Note that the road/walkway at the top of the dam is not called a berm. For the dam in Fig 7.8, there are four berms—two on the upstream side and two on the downstream side. These correspond to the edges CD, EF, IJ, and KL in the figure.

Imagine a dam of 1 km length. We can slice this dam into 1000 vertical slices along the length of the dam, each of 1 m top length. Let a be the area of cross-section of the slice at the center point along the length of the dam. Then the volume of this slice is a cubic meters. The total volume of the material of the entire dam is computed as a function of the cross-sectional area at the center point and the depths at various points along the length of the dam. Therefore, it is the cross-sectional area of the dam at its center point that determines the volume of the material used for building the dam. In fact, the engineers design the cross-section of a dam at its center point to

design the dam. It is the cross-sectional view that provides all the design parameters and these are described below.

4.3 Basic Terminology of an Earth Dam

When an engineer designs an earth dam, she/he has to specify certain dimensions. These dimensions are described below. We shall call them the design parameters. In order to understand these parameters, we must first understand the basic structure and terminology of an earth dam. These are described with the help of Fig 7.8. Firstly, the cross-section is viewed as a 2D object in the xy plane. Here, the x axis is parallel to the direction of the water flow and is assumed to be aligned with the ground. Origin of the 2D plane is taken as the starting point of the dam on the upstream side (point O in the picture). Some of the terms defined below are not the engineering terms but are coined to facilitate the description.

Base Line of the Dam: It is the x axis assumed to be aligned with the ground level. It is the horizontal line AN in the picture.

Base of the Dam: It the line segment OM in the picture. It is on this line segment that the dam is erected. The width of the line segment OM is called the *base width of the dam*.

Top of the Dam: The horizontal line segment GH is called the top of the dam. The width of the line segment GH is called the top width of the dam.

Center Line of the Dam: The perpendicular bisector of the line segment GH is called the central line of the dam. It divides the plane into two halves. The half that impounds water is called the upstream of the dam and the other half is called downstream of the dam.

Berms: The horizontal line segments CD, EF, IJ, and KL are called the berms, and their widths are called respective berm widths.

Core: The trapezium XUVY is called the core of the dam. It comprises material that has low permeability. The line segments XU, UV, VY, and XY are called the core edges, and their union is called the core border. The width of the core top, UV, is called the core top width, and the width of the core base, XY, is called the core base width. The distance between the core top and the core base is called the core height.

Shell: The area of the polygon OCDEFGHIJKLM excluding the core is called the shell. The line segments OC, CD, DE, EF, FI, IJ, JK, KL, LM, and OM are called the shell edges.

Foundation of the Dam: The rectangle ATSN is called the foundation of the dam, and its depth, length of AT, is called depth of the foundation of the dam.

Full Reservoir Level (*FRL*): Dams are designed to hold water safely up to a certain level, called the full reservoir level, which normally takes into account the flood situations. When the water level is getting close to *FRL* or rises above *FRL*, the water is released through an outlet channel called the spill way. In the figure, FRL is the height of the horizontal line WRP from the ground level (equals to the y coordinate of W or R or P).

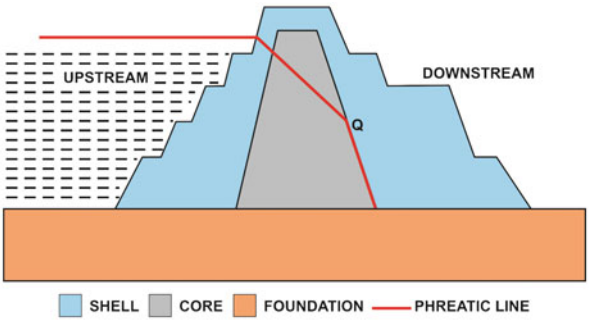


Fig. 7.9 Picture of a heterogeneous dam

Phreatic Line: By definition, it is the upper flow line, or the free surface of the seepage, along which the pressure in the soil water is equal to the atmospheric pressure. It is assumed to be the union of line segments WP, PQ, and QY in the figure. In reality, the line segment PQ in the phreatic line is a curve rather than a straight line. For the purpose of computation of factor of safety, PQ is also assumed to be a line segment. In many of the designs, the slope of PQ is assumed to be -0.25. However, other criteria are also in practice. For instance, the IS Code 7894-1975 [1] defines the point Q as the point of intersection of the downstream core edge and the horizontal line at $y = FRL/2$.

Minimum Draw Down Level (*MDDL*): It is the level at which water is always maintained. This is also known as sill level.

Rocky Zone: The portion below the foundation is called the rocky zone or impervious zone.

Moist or Dry Zone: Moist zone is the portion of shell and core that is above the phreatic line. Moist zone is also referred to as *Dry zone* in technical terms.

Buoyant Zone: This is the area of the dam bounded by *MDDL*, the line segment BY and the x axis.

Saturated Zone: This is the area of the dam which is neither moist nor buoyant. It is the area of the dam bounded by the *MDDL* line and the phreatic line. Saturated zone is also referred to as *wet zone*.

Homogeneous Dams: We shall call a dam homogeneous if the structures of shell and core are symmetric around the central line of the dam. Otherwise, it will be called a heterogeneous dam. While Fig 7.8 depicts a homogeneous dam, Fig 7.9 is an example of a heterogeneous dam.

4.4 Design Parameters of an Earth Dam

As mentioned earlier, the safety and economics of an earth dam depend upon the cross-section of the dam. Therefore, designing a dam essentially means the determination of cross-sectional parameters of the dam. Of course, the determination of materials and methods to be used in the construction of the dam is equally important for designing the dam. But for the purpose of the optimization problem considered

in this chapter, it is assumed that these factors are identified and fixed so that their determination is kept out of the scope of this project. Their role in our problem is only to use their strength characteristics in terms of friction and cohesiveness. We shall now identify the design parameters of the cross-section of an earth dam. Since we are dealing with only the cross-section of the dam, it will be convenient to use the 2D coordinate system to represent various points and objects in the design. Before proceeding further, we need the following definitions.

4.5 Number of Berms

In a heterogeneous dam, the number of berms on the upstream side need not be same as the number of berms on the downstream side. Let n_U denote the number of berms on the upstream side and let n_D denote the number of berms on the downstream side. For the dam in Fig 7.8, n_U and n_D are both equal to 2. And for the dam in Fig 7.9, $n_U = 4$ and $n_D = 3$.

4.6 Berms, Slopes and Their Numbering

Berms and their widths have already been defined earlier. We shall number the berms as Berm 1, Berm 2, …, starting from the upstream side from ground level to the top, and then from the top to the bottom on the downstream side. Thus, for the dam in Fig 7.8, Berm 1 is CD, Berm 2 is EF, Berm 3 is IJ, and Berm 4 is KL. Notice that each berm has a slanting edge attached to it and below the berm. For example, Berm 1 in Fig 7.8, CD, has the slanting edge CO below it. Similarly, the berms EF, IJ, and KL are adjacent to the slanting edges ED, JK, and LM respectively. In addition, the top two slanting edges are always adjacent to the top of the dam. These are GF and HI in Fig 7.8. Notice that each slanting edge corresponds to a right angled triangle in which the slanting edge is the hypotenuse. One edge of this triangle is in the vertical direction and the other in the horizontal direction. The horizontal edge will be called the base of slanting edge. The widths of horizontal and vertical edges of the triangle will be called the width and height of the slanting edge respectively. For the purpose of this chapter, designing an earth dam means specifying the berm widths, widths and heights of the slanting edges, and the core bottom width (XY in Fig 7.8). In other words, the decision variables of the earth dam design optimization problem are the widths and heights mentioned in the previous sentence. These decision variables shall be denoted by the vector $u = (u_1, u_2, \ldots, u_k)$.

Normally, the dam height, dam top width, the core height, and core top width are fixed and the designer has no choice over these parameters. In view of this, the coordinates of u are taken in the following order. Take u_1 as the left most width, u_2 is the width next to it, and so on. If u_k is the width of the right most slanting edge of the dam, take u_{k+1} as the width of the core, take u_{k+2} as the height of the left

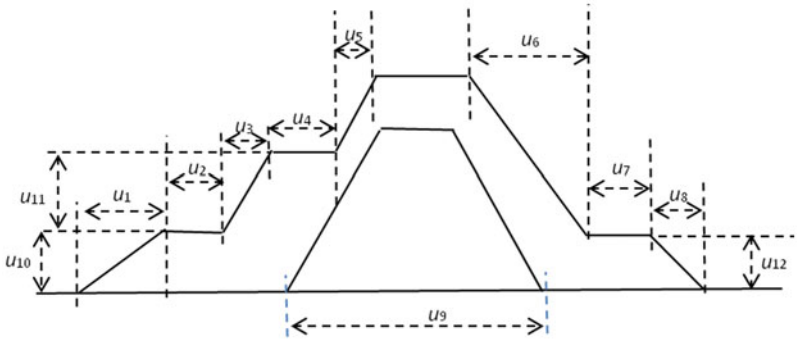


Fig. 7.10 The design parameters of earth dam

most slanting edge of the dam, u_{k+3} as the height of the next slanting edge, and so on. While considering the slanting edge heights, the heights of the slanting edges adjacent to dam top are ignored. The reason for this is that the dam height is fixed.

The decision variable u for a three-berm dam (total number of berms is three) shown in Fig 7.10 has 12 variables, that is, $u = (u_1, u_2, \ldots, u_{12})$. Notice that u_8 is the width of the right most slanting edge of the dam. Next, u_9 is the width of the core bottom. Having exhausted all the widths, u_{10} represents the height of left most slanting edge of the dam, and u_{11} is the height of the immediate right slanting edge. As the dam height is fixed, say h, the height of the upstream slanting edge adjacent to dam top is equal to $h - u_{10} - u_{11}$. Therefore, this height is not a decision variable. Similarly, the height of the downstream slanting edge adjacent to the dam top is also fixed as the slant height of the right most slanting edge (u_{12}) is taken as a decision variable.

Thus, we have identified and defined all the decision variables of the earth dam design optimization problem. The variables are: n_U, n_D, and u.

5 Factor of Safety with Respect to a Slip Circle—Theory, Definitions, and Computation

The material or the mass of an earth dam has a tendency to slide or slip downward due to its weight (the gravitational force on the mass). This may lead to a partial or complete failure of the dam. The force that actuates the downward movement of the mass is a function of the weight of the dam determined in a particular direction. The weight of the mass depends upon the condition of the dam. For instance, when the water is up to *FRL*, then the weight of the dam is heavy compared to the situation where the water is only up to *MDDL*. Similarly, if there is an earthquake, there is an additional force shaking the dam structure adding to the downward movement. In view of this, the safety of the dam is evaluated under different conditions. For

the purpose of this chapter, we shall confine ourselves to only one of those conditions. The forces causing downward movement are typically termed as *driving* or *actuating forces*. As has been pointed out just now, the nature and quantum of these forces depend on various conditions and the dam properties such as the material characteristics, the quality of construction, and so on. A dam has a natural tendency to offer *resistance* to the driving forces. The resistance arises from the characteristics of the materials used, the method of construction, the external environment (which changes the characteristics of the soil etc.), and the design parameters which were identified in the previous section. The resistance arises from the frictional characteristics and the cohesiveness of the materials used and the design characteristics. The safety of a dam is a function of the driving forces and resistances. The factor of safety is computed using the driving forces and resistance and their resulting moments. If the moment of resistance is greater than the moment of driving forces, then the dam is expected to be stable and safe. The factor of safety is calculated as the resisting moment to the driving moment.

Several methods have been proposed to determine the factor of safety. These include Culman wedge block Method, Swedish circle or Fellenius method, Janbu method, Bishop method, Morgenstern–Price method, Spencer method, Bishop–Morgenstern method, Taylor method, Barnes method, Lowe–Karafiath method, Sarma method, etc. Each method has its own assumptions([3], http://www.vulcanhammer.net/utc/ence361/f2001/361-sl14.pdf). Of these, Fellenius or Swedish circle or ordinary method [2, 7] is widely accepted and used in practice. This method is developed by Wolmar Fellenius as a result of slope failures in sensitive clays in Sweden. It is the simplest method of slices. It reduces the force resolution of the slope to a statically determinate structure (http://en.wikipedia.org/wiki/Embankment_Dam). Despite the advances in evaluating the stability and the availability of analysis software ([8], http://tailingsandminewaste.org/courses/2D_3D%20Description.pdf), it is observed that the *slip circle method* is the method that is by and large used to evaluate the safety of an earth dam. The method is described in this section.

5.1 Slip Circle

In the slip circle method, it is assumed that a part of the dam slides away along the arc of a circle. The circle is called a slip circle. Its center lies in the 2D plane above the cross-section of the dam. It cuts the shell outer line at two points to form a chord. The chord divides the circle into two arcs—the major (bigger) arc and the minor (smaller) arc. By definition, the slip circle must be such that the major arc is above the chord and the minor arc must have a nonempty intersection with the cross-section of the dam. A picture of a slip circle is shown in Fig 7.11.

According to the theoretical assumptions, if the moment of the weight of the mass of the dam above the minor arc exceeds the resistance offered by the dam, then the mass above the minor arc slides away along the minor arc causing a failure of the

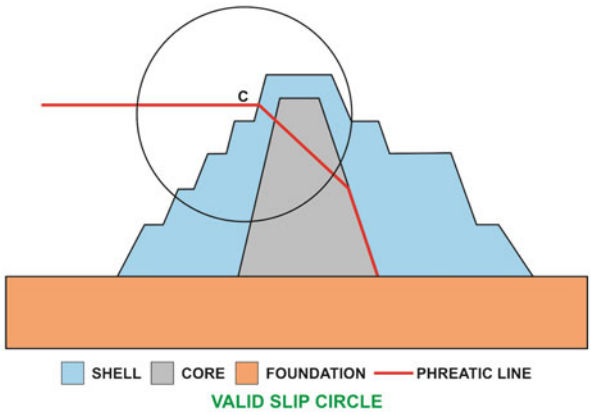


Fig. 7.11 A slip circle

dam. If a slide occurs, the mass slips into the upstream side of the dam provided the slope of the chord of the slip circle is positive; otherwise, the mass slips into the downstream side. The dam is expected to be safe with every possible slip circle. From engineering and physics knowledge, it is known that failure cannot occur with respect to certain slip circles. Such a slip circle is called an invalid slip circle and it is assumed that factor of safety with respect to an invalid slip circle is infinity. There are certain rules to verify whether a given circle is valid or not. For a circle to be a valid slip circle, the conditions to be satisfied by it, besides the ones mentioned above, are: it is not a very small circle (i.e., its radius is larger than a specified parameter ℓ), the circle does not extend beneath the foundation bottom, and it does not intersect any vertical line in the cross-section (i.e., a line parallel to the straight line joining the midpoints of the core base and the core top (line segments XY, UV in Fig 7.8)) at two distinct points both lying inside the cross-section of the dam.

If the center of the slip circle lies in the upstream side, then it is called *upstream slip circle*; and if the center lies in the downstream side, then the circle is called *downstream slip circle*. The slope of any upstream slip circle chord is positive and that of a downstream slip circle chord is negative. In what follows, reference to a slip circle means that it is a valid slip circle unless stated otherwise.

We are concerned with only the minor arcs of the slip circles and whenever we refer to arc, unless stated otherwise, we mean that it is the minor arc that is being referred.

5.2 *Weight of Mass of the Dam at a Point on the Arc*

Consider a slip circle arc and a slice of the dam of small width δx and depth 1 m above the arc at the point $(x,\ y)$ on the arc. Here width is measured along the cross-section and depth along the length of the dam. The slice can be decomposed into seven different components, namely, shell slice in the dry (also referred to as moist) zone ($h_{ds}(x)$), shell slice in the wet (also known as saturated) zone ($h_{ws}(x)$), shell slice in

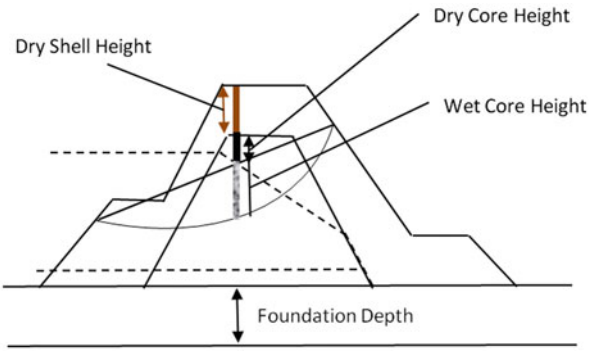


Fig. 7.12 Weight composition of material at a point on the arc

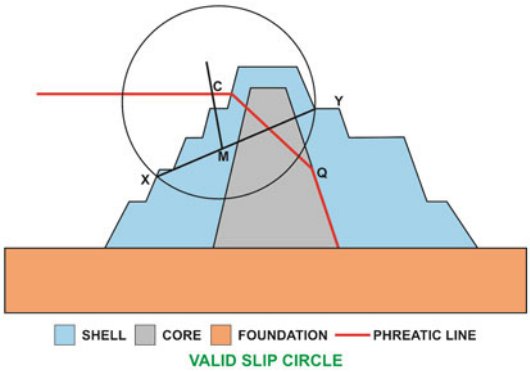


Fig. 7.13 Slip circle with chord *XY*

the buoyant zone ($h_{bs}(x)$), core slice in the dry zone ($h_{dc}(x)$), core slice in the wet zone ($h_{wc}(x)$), core slice in the buoyant zone ($h_{bc}(x)$), and the slice of foundation ($h_f(x)$). The notation in the parentheses against each of these components is the approximate height (actually height at x which is taken as the approximate height of the slice) of the respective slice component. For the instance in Fig 7.12, only three of these components have positive heights, namely, $h_{ds}(x)$, $h_{dc}(x)$, and $h_{wc}(x)$. The heights of other components are zero. These heights depend on the arc and the position (x, y) on the arc. As the shell, core, and foundation materials are different, they have different densities and their densities also depend on the zone (recall that densities are weight densities as mentioned earlier). If d_{ds}, d_{ws}, d_{bs}, d_{dc}, d_{wc}, d_{bc}, and d_f are the densities of the seven components mentioned above (in the same order), then the expression $d_f h_f(x) + d_{wc} h_{wc}(x) + d_{dc} h_{dc}(x) + d_{ws} h_{ws}(x) + d_{ds} h_{ds}(x)$ multiplied by the width δx of the slice gives the approximate weight of the slice above the arc. The densities depend on the material and the zone in which the material is present.

5.3 Driving Forces and Resistance at a Point on the Arc

Consider the slip circle with center at C in Fig 7.13. The chord XY is formed by the points of intersection of the circle with the shell outer line. Since the sliding takes place only on the minor arc, our interest and discussion are confined only to the

minor arc which is below the chord. Hence, any reference to the arc means it is with respect to the minor arc only as stated earlier. Let $(x,\ y)$ be an arbitrary point on the arc. Functionally, let $w(x,\ y)$ denote the weight of the material vertically above at the point $(x,\ y)$. For brevity, we write simply w for $w(x,\ y)$. From the previous discussion, we have:

$$w(x, y) = d_f h_f(x) + d_{bs} h_{bs}(x) + d_{ws} h_{ws}(x) + d_{ds} h_{ds}(x) + d_{bc} h_{bc}(x) + d_{wc} h_{wc}(x) + d_{dc} h_{dc}(x), \tag{7.1}$$

where d_f, d_{bc}, d_{wc}, d_{dc}, d_{bs}, d_{ws}, and d_{ds} are the material densities of the foundation, buoyant core, wet core, dry core, buoyant shell, wet shell, and dry shell respectively; and $h_f(x)$, $h_{bc}(x)$, $h_{wc}(x)$, $h_{dc}(x)$, $h_{bs}(x)$, $h_{ws}(x)$, and $h_{ds}(x)$ are the vertical height components of foundation, buoyant core, wet core, dry core, buoyant shell, wet shell, and dry shell above the point $(x,\ y)$ respectively.

The weight w is the force always acting vertically downward. This force is resolved into two components: (i) the driving force resulting from the weight w and (ii) the force resulting from friction. We shall denote the driving force due to weight at $(x,\ y)$ by $DF(x,\ y)$, and the resistance resulting from friction by $FF(x,\ y)$. $DF(x,\ y)$ is the component of $w(x,\ y)$ in the tangential direction and $FF(x,\ y)$ is the component of $w(x,\ y)$ in the normal direction with respect to the slip circle. Note that these components are determined at each point $(x,\ y)$ on the arc. If $\alpha(x,\ y)$ is the angle (acute) between the weight (force vector acting vertically downward) and the normal to the slip circle at the point $(x,\ y)$, then $DF(x,\ y) = w(x,\ y) \sin\alpha(x,\ y)$ and $FF(x,\ y) = w(x,\ y) \cos\alpha(x,\ y) \tan\phi(x,\ y)$, where $\tan\phi(x,\ y)$ is the coefficient of friction. Here, $\phi(x,\ y)$ is called the angle of friction and it varies with material and the zone. For the purpose of this work, the coefficients of friction are taken as inputs.

5.4 Cohesion

In addition to friction, there is the cohesive component which adds to the resistance. The cohesion acts along the arc. The amount of resistance offered by cohesion at a point $(x,\ y)$ on the arc is measured by the cohesion coefficient. The cohesion coefficient, defined as the force per unit area, depends on the material at the point $(x,\ y)$ and the zone to which the point belongs. Consider the arc of a small length, say μ m, so that it is entirely in a zone composed of same material. The strip of the arc along the unit length of the dam has an area of μ m^2. Therefore, the resistance offered by cohesion of this strip is equal to $c \times \mu$, where c is the coefficient of cohesion. The unit of c is kgF/m^2. The cohesion coefficients are denoted by c_f for material in the foundation, c_{ds} for dry shell material, c_{ws} for wet shell material, c_{bs} for buoyant shell material, c_{dc} for dry core material, c_{wc} for wet core material, and c_{bc} for buoyant core material. We shall denote the cohesion at the point $(x,\ y)$ by $CF(x,\ y)$. Thus, if $(x,\ y)$ is in the foundation, then $CF(x,\ y) = c_f \mu$; if $(x,\ y)$ is in dry shell zone, then $CF(x,\ y) = c_{ds}\mu$, and so on.

5.5 Effect of Earthquakes

When earthquake is taken into account for computing the factor of safety, its effects are added to the actuating force and resistance. Technically, it is assumed that earthquake acts in the horizontal direction and its magnitude is taken as a fraction of the weight of the object. This fraction is called the *earthquake coefficient* which is a unitless number. The components of earthquake at $(x,\ y)$ are: $w(x,\ y)Cos\alpha(x,\ y)C_{EQ}$ to the driving force and $w(x,\ y)C_{EQ}\sin\alpha(x,\ y)\tan\phi(x,\ y)$ to the resistance, where C_{EQ} is the earthquake coefficient. The total driving force (TDF) and the total resistance (TRF) at $(x,\ y)$ are given by $TDF(x,\ y) = w(x,\ y)\{\sin\alpha(x,\ y) + C_{EQ}\cos\alpha(x,\ y)\}$ and $TRF(x,\ y) = CF(x,\ y) + w(x,\ y)\{\cos\alpha(x,\ y) + C_{EQ}\sin\alpha(x,\ y)\}\tan\phi(x,\ y)$.

5.6 Definition of Factor of Safety

The driving moment at point $(x,\ y)$ on the arc is defined as $r.TDF(x,\ y)$ and the resisting moment at $(x,\ y)$ is defined as $r.TRF(x,\ y)$, where r is the radius of the slip circle. The total driving moment on the arc, $TDM(Arc)$, is defined as the integral of $r.TDF(x,\ y)$ over the arc, that is the line integral of $r.TDF(x,\ y)$ over the arc of the slip circle. Similarly, the total resisting moment on the arc, $TRM(Arc)$, is defined as the integral of $r.TRF(x,\ y)$ over the arc, that is the line integral of $r.TRF(x,\ y)$ over the arc of the slip circle. Thus, the cumulative driving and resisting moments on the arc are given by

$$TDM(Arc) = \int_{Arc} r.TDFds \quad \text{and} \quad TRM(Arc) = \int_{Arc} r.TRFds, \tag{7.2}$$

where the integrals ($\int_{Arc}$) represent the line integrals over the arc (http://en.wikipedia.org/wiki/Line_integral#Definition).

The factor of safety of the dam with respect to the slip circle with center C and radius r is defined by

$$FS(C,\ r) = (TRM(Arc))/(TDM(Arc)) \tag{7.3}$$

It may be noted that the factor of safety is a unitless number. Higher the factor of safety, the better it is.

5.7 Computation of Factor of Safety

From the previous section, it is clear that computation of factor of safety involves evaluation of line integrals of functions that have no simple closed form expressions. In practice, these integrals are determined using numerical approximation. The usual approach is to divide the structure of the dam above the arc into vertical slices of

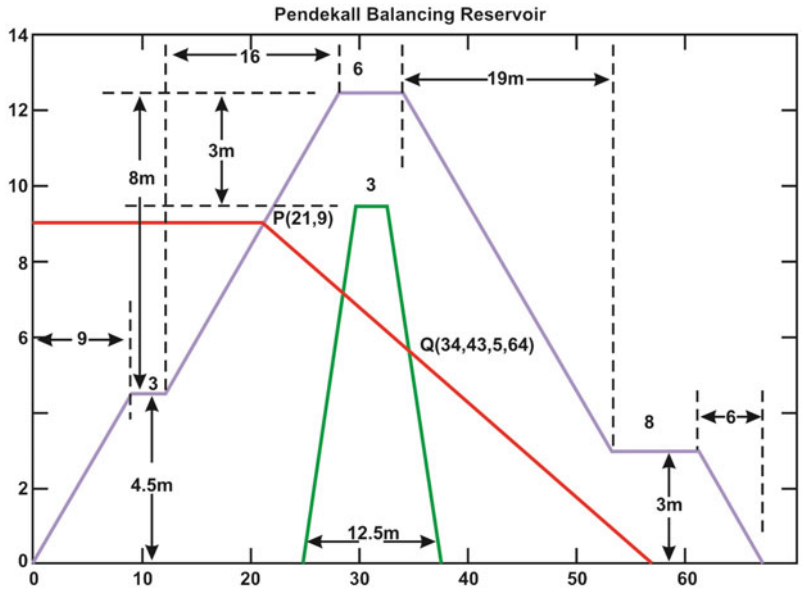


Fig. 7.14 Cross-sectional design of the Pendekal dam

small widths, determine the weight components of each slice, compute the driving and resisting moments for each slice, and then sum them up to get the total driving and total resisting moments. The accuracy of this procedure depends upon the number of slices considered for the computation. The Indian standard code IS 7894-1975 recommends about 15–20 slices to be considered for evaluation of the factor of safety. Computation of weight of the material, $w(x,\ y)$, requires determination of vertical height components $h_f(x), h_{ds}(x), h_{ws}(x), h_{bs}(x), h_{dc}(x), h_{wc}(x)$, and $h_{bc}(x)$. This in turn requires finding points of intersection of the phreatic line with various line segments of the cross-section and the arc of the slip circle. Once these heights are determined, they need to be multiplied by their densities. As per the recommended procedure, the densities used for computing driving forces may be different from those used for computing resistance. The reason for this is when the densities cannot be determined precisely, lower values are taken for the resistances to give benefit of doubt for the factor of safety computation. For this reason, the densities are specified separately for the driving forces and resistance computations. As an illustration, we shall now explore the computational aspects of factor of safety with respect to a specific slip circle for the example below.

5.8 Example

The Pendekal dam, a heterogeneous dam with two berms—one in the upstream and the other in the downstream, is located near Ananthapur in Andhra Pradesh, India. Its cross-sectional design is shown in Fig 7.14. For this dam, the dam height is 12.5 m,

Table 7.1 Densities and coefficients of friction and cohesion

Zone	Densities (kgF/m^3)		Cohesion (kgF/m^2)	Coefficient of friction
	Driving	Resistance		
Dry shell (ds)	2006	2006	2900	0.325
Dry core (dc)	2000	2000	2000	0.141
Wet shell (ws)	2054	1054	2900	0.325
Wet core (wc)	2010	1010	2000	0.141
Foundation (f)	1073	1073	3200	0.231
Buoyant shell (bs)	1054	1054	2900	0.325
Buoyant core (bc)	1010	1010	2000	0.141

Coefficient of earthquake (C_{EQ}): 0.06
Read parameters in the following fashion: $d_{ws} = 2.25$ for resistance, $d_{dc} = 1.76$ for driving, $c_f = 1.90$, etc. Last column gives tan ϕ values

Table 7.2 Effect of number of slices in computation of factor of safety

S.No.	No. of slices	Driving force (kgF)	Cohesive force (kgF)	Resistance (kgF)	Factor of safety
1	15	225.117	153.292	203.015	1.583
2	20	215.197	150.721	197.719	1.619
3	50	218.063	152.037	198.239	1.606
4	100	219.563	152.123	198.513	1.597
5	200	220.138	153.333	198.641	1.599
6	500	220.860	153.552	198.931	1.596
10	5000	220.318	153.644	198.745	1.599

dam top width is 6 m, core height is 9.5 m, core top width is 3 m, the *FRL* is 9 m, the foundation depth = 10 m, and *MDDL* = *3* m. The design vector has nine variables. The design vector for the constructed dam is given by

$$u = (9, 3, 16, 19, 8, 6, 12.5, 4.5, 3). \tag{7.4}$$

These dimensions are shown in Fig 7.14. Table 7.1 gives list of required inputs for computing the factor of safety. Note that the densities are different for driving force and resistance.

Let us compute the factor of safety for this example with respect to the slip circle with center at (11.66, 13.36) and radius 21.35. Also, we shall examine the effect of number of slices on the accuracy of factor of safety. A computer program is developed to carry out these computations. Table 7.2 presents the factor of safety with different number of slices used.

Table 7.3 gives factor of safety for different slip circles. This information is useful in throwing some light on the extent of error if 50 slices are used. From these observations, it sounds reasonable to use 50 slices for computing the factor of safety.

The above procedure is useful for computing the factor of safety with respect to a given slip circle. But the problem is to find the minimum factor of safety for upstream and downstream separately over all possible valid slip circles. As the number of valid slip circles is infinite, the usual practice is to select, based on experience, a few slip

Table 7.3 Extent of error if 50 slices are used

Slip circle	Number of slices	Circle center x_0	Circle center y_0	Radius (m)	Factor of safety	Error
1	50	48.179	8.646	28.180	1.961	0.002
	5000	48.179	8.646	28.180	1.959	
2	50	37.975	8.770	17.977	2.443	0.046
	5000	37.975	8.770	17.977	2.490	
3	50	40.170	4.519	30.170	2.325	0.004
	5000	40.170	4.519	30.170	2.321	
4	50	26.421	3.006	28.579	2.314	0.022
	5000	26.421	3.006	28.579	2.292	

circles which are likely to be critical and evaluate the factor of safety with respect to only those slip circles. If the factor of safety is satisfactory with respect to these selected slip circles, then the design of the dam is accepted to be safe. The IS code suggests that about 12 circles be analyzed for examining the stability of a dam.

With the power of computing available today, a large number of slip circles can be analyzed in no time. In fact, we developed an approach that analyzes infinitely many circles. This is plausible based on a phenomenon that we have observed while working on this project. The approach and the phenomenon are elaborated below.

6 Factor of Safety of an Earth Dam

In the preceding section, we have looked at the factor of safety with respect to a specific slip circle. However, the dam failure may occur due to any slip circle with low factor of safety. For this reason, there is a need to find the minimum factor of safety over all possible slip circles. In practice, the factor of safety is split into two parts—the upstream factor of safety and the downstream factor of safety. These are defined below.

6.1 Upstream and Downstream Factors of Safety

Two *factors of safety* are defined—one for the upstream and the other for the downstream. The upstream factor of safety of a dam, FS_U, is defined as the minimum of $FS(C,\ r)$ over all upstream slip circles $(C,\ r)$. Similarly, the downstream factor of safety, FS_D, is defined as the minimum of $FS(C,\ r)$ over all downstream slip circles $(C,\ r)$. In practice, more importance is given to the downstream factor of safety compared to the upstream factor of safety. The conventional lower specifications are: 1.3 for the upstream factor of safety and 1.5 for the downstream factor of safety.

6.2 The Chord Based Approach for Evaluating Factors of Safety

The procedure described earlier is useful in computing the factor of safety with respect to a given slip circle. But to compute the factors of safety, FS_U and FS_D,

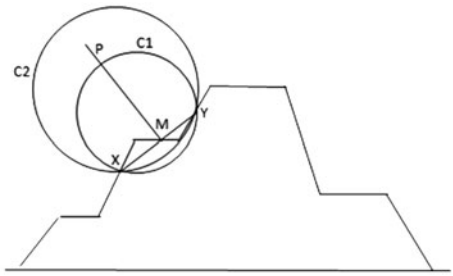


Fig. 7.15 Common slips circles of a chord

one must explore all valid slip circles. Notice that the arc of any slip circle has a chord associated with it. But this chord is common to infinitely many slip circles. Figure 7.15 exhibits two slip circles with a common chord XY. Every slip circle associated with a chord will have its center lying on the perpendicular bisector of the chord. Therefore, exploring all slip circles is equivalent to exploring all valid chords. By exploring a valid chord, we mean the following: take a chord of a valid slip circle and explore all slip circles that are associated with the chord. There is a substantial reduction in the effort required if this approach is used. The reduction is due to a special feature observed while working on this project. The special feature is that the factor of safety of slip circles associated with a chord is a *near* convex function. The details are explained below.

Notation for a Chord Throughout this chapter, we shall use the following notation for a chord. Let $X = (x_1,\ y_1)$ and $Y = (x_2,\ y_2)$ be the end points of the chord. Note that for a given design, y_i is a function of $x_i,\ i = 1, 2$. Therefore, for a given design, a chord is uniquely determined by the x-coordinates of its end points. For this reason, we use the representation $Chord(x_1,\ x_2)$ for the chord XY with $x_1 < x_2$.

6.3 Convex Nature of $FS(C,\ r)$ on a Chord

All slip circles associated with a chord will have their centers lying on the perpendicular bisector of the chord and they are uniquely determined by the distance between their centers and the chord. That is, if C_1 and C_2 are two slip circles associated with a chord (see Fig 7.15) with the distances from their centers to the chord as d_1 and d_2 respectively, then C_1 and C_2 are same if, and only if, $d_1 = d_2$. Now fix a valid chord of length $2\ L$ and consider the slip circle of radius r associated with the chord whose center is at a distance d from the chord. Figure 7.16 shows the chord, arc, radius, and the distance for this slip circle. Notice the lengths of the edges of the right-angled triangle ABC in the figure. Since the length of the chord is fixed, r is a function of d given by $r(d) = \sqrt{L^2 + d^2}$. Let $\theta(d)$ denote the angle ACB and define the parameter $\lambda(d) = d/r(d)$. Note that $\lambda(d) = \cos\theta(d)$. As d tends to infinity, the angle $\theta(d)$ goes to zero and $\cos\theta(d)$ goes to 1. This means as d approaches infinity, $\lambda(d)$ approaches 1. Also, note that for any given $\lambda \in (0, 1)$, there is a unique d such that $\theta(d) = \lambda$. Now consider the factor of safety of slip circles associated with the chord as a function of the parameter λ and denote it by $f(\lambda)$. The behavior of the function

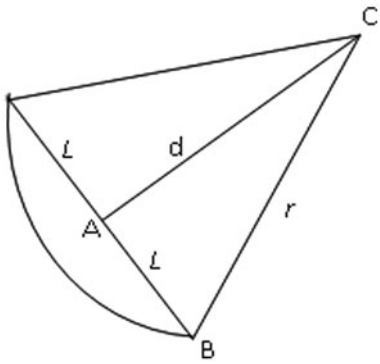


Fig. 7.16 Relationship between radius and distance to chord

$f(\lambda)$ is observed empirically by plotting the function for some selected chords of the Pendekal dam example (see Fig 7.17). The interesting and useful observation is that the function is approximately convex. We can exploit this observation in computing the minimum factor of safety with respect to the family of circles associated with the chord. This is done as follows. Determine the smallest λ, say λ_0, and compute the minimum of $f(\lambda)$ in the interval $[\lambda_0, 1)$. The question is: What is λ_0? This is determined as follows. One of the rules for a valid slip circle is that it cannot have a vertical line intersecting the arc at two distinct points inside the dam. This means that the y-coordinate of the center of the slip circle must be greater than or equal to the y-coordinates of the two end points of the chord. Let the chord be PQ, and let Q be the end point of the chord with the largest y-coordinate (whose value we will denote by y_0). Therefore, the center of the slip circle with smallest d is the intersection point of the perpendicular bisector of PQ and the parallel line $y = y_0$. Let d_0 be this smallest distance. Let r_0 be the radius of the corresponding slip circle. Then $\lambda_0 = d_0/r_0$ is the required smallest λ.

Next consider $\lambda_2 = d_2/r_2$ and $\lambda_3 = d_3/r_3$, where $r_2 < r_3$. Since $\lambda = d/r = \cos\theta$ for $0 < \theta < \pi/2$, there is a unique d for every λ. Choose λ_1 sufficiently close to 1 and let d_1 be the distance corresponding to λ_1. To find out the minimum of $f(\lambda)$ in the interval $[\lambda_0, 1)$ we use the following approach. Let $d_2 = (d_1 + 2d_0)/3$ and let $d_3 = (2d_1 + d_0)/3$. If $f(\lambda_2) < f(\lambda_3)$, then set $d_1 = d_3$; otherwise set $d_0 = d_2$. Repeat this process until d_0 and d_1 are sufficiently close. For a strict convex function, this procedure leads to the minimum.

Since the function $f(\lambda)$ can have local minima, we compute the function at a number of points in the interval $[\lambda_0,\ \lambda_1)$ where λ_1 is a number close to 1. Our algorithm for computing minimum of $f(\lambda)$ in the interval $[\lambda_0,\ 1)$ is given below.

Algorithm for Computing Minimum of $f(\lambda)$: Recall that the x-axis is assumed to be aligned with the ground level. It is the horizontal line AN in Fig 7.8.

Step 0: Fix λ_1 close to 1 and less than 1. Fix a small positive number, ϵ to be used for incrementing d.

Step 1: Determine d_0 corresponding to λ_0 and d_1 corresponding to λ_1.

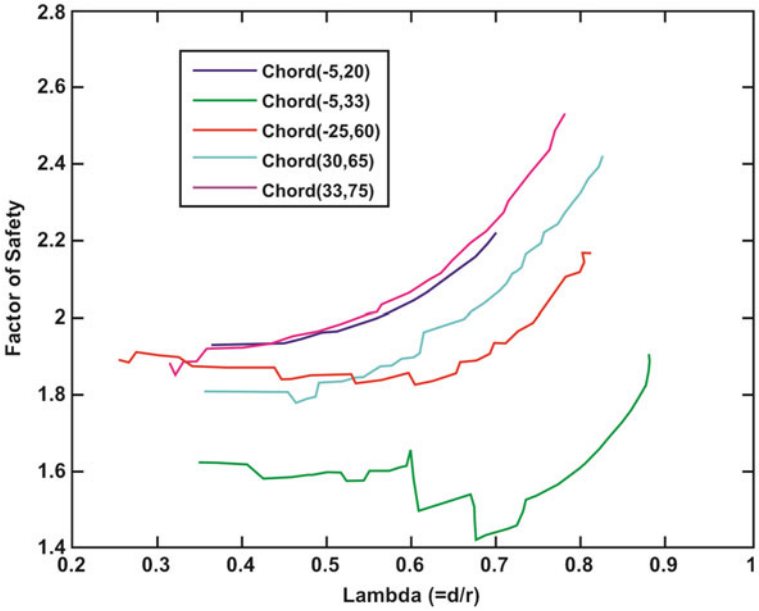


Fig. 7.17 Factor of safety as a function of lambda (λ)

Step 2: Let $d_2 = (d_1 + 2d_0)/3$ and $d_3 = (2d_1 + d_0)/3$. If $f(\lambda_2) < f(\lambda_3)$, then set $d_1 = d_3$; otherwise set $d_0 = d_2$.

Step 3: If d_0 and d_1 are sufficiently close, take $f(\lambda_0)$ as the minimum of $f(\lambda)$, and stop. Otherwise go to Step 2.

This algorithm is applied to the Pendekal dam design and it is found to be very efficient. The graph of factor of safety for the $Chord(28, 59)$ is shown in Fig 7.18. It took 35 iterations (examination of 35 slip circles) to arrive at the minimum. The time taken is about 4 s. The factor of safety for this chord is 1.334 and the critical slip circle of the chord has its center at (12.69, 21.56) with a radius of 24.08.

6.4 Evaluation of FS_U and FS_D

So far, we have a procedure for computing the minimum factor of safety on a family of slip circles associated with any given chord. From here we should have a procedure to compute the upstream and downstream factors of safety. This calls for exploring all valid chords. This can be formulated as a mathematical programming problem involving two decision variables (this will be shown in the next section). But for simplicity, we shall explore the minimum factor of safety on a large number of chords and then take the minimum of all those minimums. It should be noted that the procedure considers a subset of all possible slip circles and hence the factors of

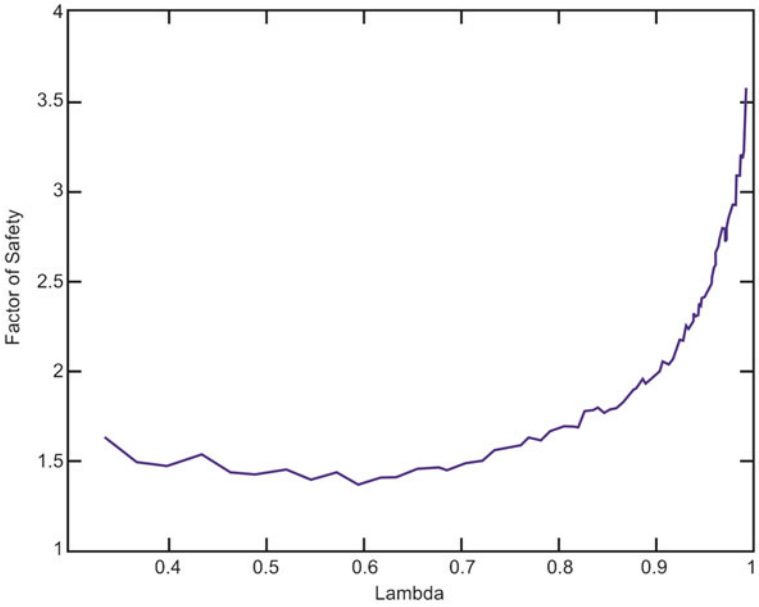


Fig. 7.18 Factor of safety as a function of lambda (λ)

safety evaluated are near approximations and not exact theoretically. Our procedure for doing this is given below.

6.5 *Procedure for Computing FS_U and FS_D*

We shall describe the procedure for computing FS_U. A similar procedure is used for computing FS_D.

Consider the $chord(a_1, a_2)$ with end points $X = (a_1, b_1)$ and $Y = (a_2, b_2)$. Recall that $a_1 < a_2$. If $b_1 < b_2$, then the chord corresponds to an upstream slip circle; or a downstream slip circle if $b_1 > b_2$. For ease of notation, let us define the nonnegative function $shell(x)$ as the height of the shell at x. Note that $shell(x) = 0$ for all $x \leq 0$ and for all x greater than the base width of the dam. Note that $b_1 = shell(a_1)$ and $b_2 = shell(a_2)$. Fix a lower bound B_L for $a_2 - a_1$. It may be noted that $a_2 - a_1$ is the length of the orthogonal projection of the chord PQ where $P = (a_1, shell(a_1))$ and $Q = (a_2, shell(a_2))$ onto the horizontal axis. Since small slip circles can be ignored, chords of small length can also be ignored. For this reason, we may set a minimum length, say L_0, and consider only chords whose length is at least L_0. Instead of setting a lower limit on chord length (L_0), we may set a lower limit on the length of the projection of the chord. Thus, we can choose B_L to be a practically meaningful figure. Once B_L is set as the lower limit, all chords of length less than B_L will be ignored. The starting point of a chord may have its x-coordinate (a_1) well before the origin. A practically meaningful value for this is taken as L where L is

-0.25 times the base width of the dam. If a chord is an invalid chord, its minimum factor of safety is taken as infinity.

According to this procedure, we shall take

$$FS_U = \min_{ij} FS_{ij},$$

where FS_{ij} is the minimum factor of safety over the $Chord(x_1^i,\ x_2^j)$, $x_1^i = L + i\delta x_1$, $x_2^j = x_1^i + B_L + j\delta x_2$, and $i,\ j$ are nonnegative integers. Here δx_1 and δx_2 are used as increments for the chord first and second parameters respectively.

Let s_{Left} and s_{Right} denote the x-coordinates of the dam top left and right points respectively (points G and H in Fig 7.8). It may be noted that x_1^i being the initial x-coordinate of an upstream chord, we must have $x_1^i \le s_{Left}$. If $x_2^j > s_{Right}$, then the $Chord(x_1^i,\ x_2^j)$ is flatter than $Chord(x_1^i,\ s_{Right})$ and has higher factor of safety. For this reason, it suffices to consider only j such that $x_2^j < s_{Right}$. The steps for determining FS_U are given below.

Step 0: Choose and fix $L,\ \delta x_1$ and δx_2. Set $i = 0,\ j = 0$ and $FS_{Min} = 1000$.
Step 1: Set $x_1 = L + i\delta x_1,\ x_2 = \min\{x_1 + B_L + j\delta x_2,\ s_{Right}\}$.
Step 2: If $x_2 \le s_{Right}$, then find out the minimum factor of safety on the $Chord(x_1,\ x_2)$. If this minimum is less than FS_{Min}, then reset FS_{Min} equal to this new minimum. Set $j = j + 1$ and go to Step 1.
Step 3: If $x_1 \le s_L$, then set $i = i + 1,\ j = 0$ and go to Step 1.
Step 4: Take $FS_U = FS_{Min}$ as the upstream factor of safety of the dam. Stop.

Applying the above algorithm to the Pendekal dam design, it is found that the factors of safety of the design, evaluated over 105 chords spread across the dam are: 1.270 for the upstream and 1.506 on the downstream. It should be noted that the implemented design does not meet the minimum requirement of 1.3 on the upstream. The failure occurs on the $Chord(-1.288,\ 34)$. The failure slip circle on this chord has its center at (11.6, 19.67) with radius of 23.52 m. The chord length is 34.18 m and the distance between the chord and the center is 17.42 m. Also, one of the engineers pointed out, based on an independent analysis, that the design does not meet the factor of safety requirements.

7 Mathematical Model for the Design Optimization

It is understood that at present the problem of designing an earth dam is merely viewed from the angle of safety and once a design is reached with required factors of safety, the book is closed. In this work, we have formulated the problem as a constrained optimization problem. The main objective of the problem is to minimize the area of cross-section of the dam which is related to minimizing the cost of material for building the dam. The main constraint is that of ensuring the factors of safety. The existing standards are: the upstream factor of safety should be at least 1.3 and the downstream factor of safety should be at least 1.5. As mentioned in the previous section, the problem of finding the factor of safety of a given design can itself be

formulated as a mathematical programming problem. We however determine the factor of safety approximately by partial enumeration. We first present this model and then proceed to consider the main problem of optimizing the design.

7.1 Model for Obtaining Factor of Safety of a Given Design

We shall describe the model for obtaining the upstream factor of safety, FS_U. Similar model can be used for FS_D as well. For this model, we shall assume that the factor of safety over any $Chord(a_1,\ a_2)$ can be determined using the algorithm described earlier. In light of this, the problem of determining FS_U boils down to determining the factor of safety over all upstream chords. Let $g(a_1,\ a_2)$ denote the minimum factor of safety over the $Chord(a_1,\ a_2)$. Then the problem of determining the upstream factor of safety is given by

Problem 1

$$\begin{aligned} \text{Minimize} \quad & g(a_1,\ a_2) \\ \text{subject to} \quad & \\ a_1 \quad & < a_2, \\ shell(a_1) \quad & < shell(a_2) \\ a_2 \quad & > 0. \end{aligned}$$

The first constraint in the above problem means that we are looking for a chord of positive length. The second constraint means that the $Chord(a_1,\ a_2)$ is an upstream chord, and the third constraint, $a_2 > 0$, together with the other constraints ensures that chord intersects the dam structure. For practical purposes, we can convert the strict inequalities in the above problem by introducing a small ϵ positive as follows:

Problem 2

$$\begin{aligned} \text{Minimize} \quad & g(a_1,\ a_2) \\ \text{subject to} \quad & \\ a_1 \quad & \leq a_2 + \epsilon, \\ shell(a_1) \quad & \leq shell(a_2) + \epsilon \\ a_2 \quad & \leq \epsilon. \end{aligned}$$

It may be observed that the constraints are piecewise linear (as *shell* is a piecewise linear function) and the objective function is nonlinear.

7.2 Model for Optimal Design

We shall now look at the main problem of this work. The problem is to determine the design vector u. The number of berms in the upstream side, n_U, and the number

of berms on the downstream side, n_D, are also decision variables. The two main constraints are that the two factors of safety should be above their minimum requirements. Clearly, these are nonlinear constraints. Besides, there will be boundary constraints on the variables and the slopes. For instance, it may be required that a slope u_i/u_j should be at most $\frac{1}{2}$, where u_i and u_j are the height and width of a slanting edge. Such a constraint can be treated as linear constraint $2u_i - u_j \leq 0$.

The core material is generally costlier compared to the cost of shell material. Therefore, in the formulation, it is justified to assume different costs to these two materials. Let C_{Shell} and C_{Core} be the unit costs (say, in rupees per m^3) of shell and core materials respectively. Let $f_{Shell}(n_U, n_D, u)$ and $f_{Core}(n_U, n_D, u)$ be the cross-section areas (in m^2) of 1-m thick slice of shell and core respectively. We now present the formulation of the problem.

Problem 3

$$\begin{aligned} &\text{Minimize} && C_{Shell} f_{Shell}(n_U,\, n_D,\, u) + C_{Core} f_{Core}(n_U,\, n_D,\, u) \\ &\text{subject to} && \\ &FS_U(n_U,\, n_D,\, u) && \geq 1.3, \\ &FS_D(n_U,\, n_D,\, u) && \geq 1.5, \\ & && u \in S, \text{ and } n_U \text{ and } n_D \text{ are nonnegative integers.} \end{aligned}$$

Here the set S is the feasible region with respect all constraints other than the factor of safety constraints.

7.3 Solving the Problem

If the problem is formulated in the above fashion allowing the n_U and n_D as decision variables, it becomes a complex problem to solve. As n_U and n_D will be small numbers, one approach is to solve the problem for different combinations of these two variables, and then choose the best among them. In practice, one may often be interested in finding a solution for fixed n_U and n_D. Therefore, it is proposed that the problem of finding an optimal design be treated as a two-stage optimization problem. In the first stage, optimal designs are obtained for each possible combination of fixed number of berms. Once this is done, the second stage optimization is that of picking up the best of the first stage optimal solutions.

Now consider the problem of finding an optimal design for a fixed number of berms. The function of factor of safety, involving the line integrals, has no closed form solution, and can only be evaluated using numerical methods. This turns out to be the major hurdle in solving the problem formulated above. The only solution to this problem is to use software such as MATLAB where a provision is given to solve such problems. The MATLAB has a function called *fmincon* which is used for finding solutions to optimization problems including nonlinear programming

problems. Consider the optimization problem:

$$\text{Minimize } f(x) \text{ subject to } g(x) \leq 0 \text{ and } h(x) \leq 0, \tag{7.5}$$

where f, g, and h are all nonlinear functions of the vector x of decision variables. To use *fmincon*, it is enough if there is computer program that provides the values of these functions at any given x. All that is required is that the computer program be written in MATLAB using its m-files. By writing $g(n_U, n_D, u) = 1.3 - FS_U(n_U, n_D, u)$, we can write an m-file for g and supply that for the upstream factor of safety constraint. Since this is the only option left to us to find an optimal solution to the above formulated problem, we developed an m-file that takes the input as the design parameters and produces a 3D vector (c, f_U, f_D) as output, where c is the cost of the material for the given design, f_U is the upstream factor of safety margin, and f_D is the downstream factor of safety margin. An attempt was made to find out an optimal design for the Pendakal dam. It so happened that the *fmincon* was not producing any output. Failing to resolve this problem, an alternative approach was adopted to find out an optimal design. Though this approach does not assure optimal solution, it is found effective at least in improving an existing design. The method is summarized briefly in the steps below and is applied to the Pendekal dam example in Sect. 7.5. This approach uses optimizing one variable at a time,while keeping all other variables fixed at present values.

7.4 Sequential 1D Optimization Method for Optimal Design

Step 1: Start with an initial design vector u. If u is feasible, then find out that component of u which yields a good profit (material cost reduction) and change only that variable with an appropriate margin. Evaluate the new design and if it feasible, repeat Step 1.

Step 2: If u is an infeasible design, identify the chord where the factor of safety is low and identify a component that is responsible for the weak factor of safety. Modify the component suitably and check for feasibility. If the modified design is feasible, go to Step 1. Otherwise repeat Step 2.

Repeat the above procedure until a satisfactory design is obtained.

7.5 Application to the Pendekal Dam

We shall now apply the procedure to analyze and improvise the Pendekal dam design. To facilitate the approach a table is designed to follow the steps in the algorithm. Before explaining the steps of the solution, let us formulate the problem for the Pendekal dam. This is explained with the help of the diagram in Fig 7.19.

Pendekal dam is a two-berm design. There are nine decision variables, in this design. Let u denote the vector of these decision variables. We shall call u as the design vector. The dam height is 12.5 m, dam top width is 6 m, core height is 9.5 m,

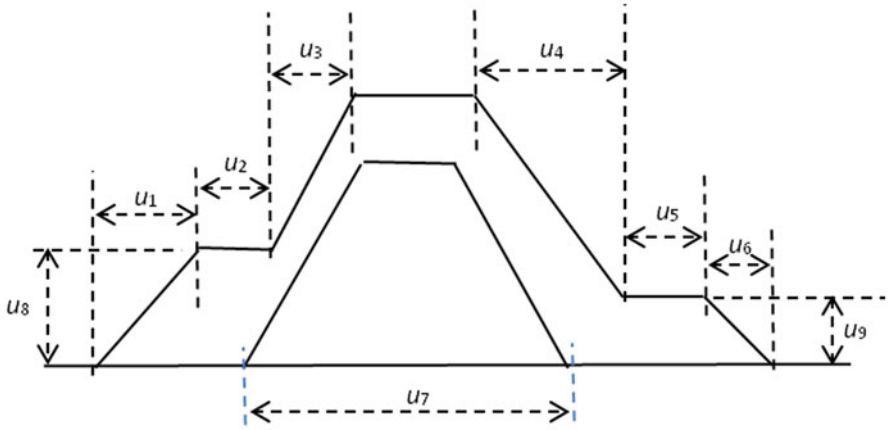


Fig. 7.19 Design variables of the Pendekal dam

core top width is 3 m, and *FRL* is 9 m. The height variables u_8 and u_9 cannot exceed *FRL*. The cost of core material for this dam is approximately 25 % more than the cost of its shell material. The total cost of the material of the dam is about 2000 million rupees (that is, about US$ 100,000). Let $A_S(u)$ and $A_C(u)$ be the areas of shell and core cross sections. Then the objective function is $A_S(u) + 1.25A_C(u)$. Since our approach starts with an initial design, the actual existing design in this case, at each stage we modify the decision variables within the allowable limits. With this in mind, the only constraints are the constraints of factor of safety and that u_8 and u_9 cannot exceed 9.5. The steps of our approach are listed below.

Solution for Pendekal Design

1. Set up the table for the approach. This is shown in Table 7.4. The items in the table are self explanatory.
2. Start with the first row and place the initial solution. The initial solution for this example is $u^0 = (9, 3, 16, 19, 8, 6, 12.5, 4.5, 3)$.
3. Evaluate the design for its factors of safety. In this case, the factors of safety are: $FS_U = 1.271$ and $FS_D = 1.541$. The critical circle has its center (12.84, 21.50) and has a radius of 25.11 m. The circle belongs to the $Chord(1, 33)$. Using this information, the natural way to increase the upstream factor of safety is by making the left side of the dam flatter. This can be done by increasing u_3^0.
4. Increase the u_3^0 from 16 to 18. The new design vector $u^1 = (9, 3, 18, 19, 8, 6, 12.5, 4.5, 3)$ is shown in the second row of the table. This new design turns out to be feasible with $FS_U = 1.366$ and $FS_D = 1.853$. This is marked under the column heading "Design Validity" in the table. The cost of the design has gone up from 443.41 to 460.41. This means an increase of 3.83 % over the original cost (see the last column of Table 7.4), but the new design is feasible whereas the original one is not.

Table 7.4 Search for an optional design using the sequential one-dimentional search method

	Design Parameters									Factor of Safety		Critical Circle					
S.No.	u1	u2	u3	u4	u5	u6	u7	u8	u9	Up Stream	Down Stream	xo	yo	Radius	Cost	Design Validity	Savings Percentage
1	9	3	16	19	8	6	12.5	4.5	3	1.271	1.541	12.84	21.50	24.11	443.41	Infeasible	0.00
2	9	3	18	19	8	6	12.5	4.5	3	1.366	1.853	11.12	26.01	27.44	460.41	Feasible	-3.83
3	9	3	17	19	8	6	12.5	4.5	3	1.349	1.850	13.29	19.84	22.92	451.91	Feasible	-1.92
4	9	3	17	15	8	6	12.5	4.5	3	1.321	1.663	13.29	19.84	22.92	420.91	Feasible	5.08
5	9	3	17	14	8	6	12.5	4.5	3	1.321	1.588	13.29	19.84	22.92	413.16	Feasible	6.82
6	9	3	17	10	8	6	12.5	4.5	3	1.255	1.384	13.29	19.84	22.92	382.16	Infeasible	13.81
7	9	3	17	12	8	6	12.5	4.5	3	1.349	1.559	12.69	21.56	24.08	397.66	Feasible	10.32

Note: The slip circle parameters shown in the Table correspond to the smaller of the two factors of safety

5. Next, it is examined if a jump of two units (16 to 18) was really necessary. So, the design $u^2 = (9, 3, 17, 19, 8, 6, 12.5, 4.5, 3)$, shown in row 3 of the table is checked for its feasibility. $FS_U = 1.349$ and $FS_D = 1.850$ for this design is also feasible and its cost is 1.92 % more than the original cost.
6. Since the downstream factor of safety is more than required, a new design may be possible with lesser cost. A new design is explored by reducing u_4^2 from 19 to 15. The new design $u^3 = (9, 3, 17, 15, 8, 6, 12.5, 4.5, 3)$, shown in row 4 of Table 7.4 is explored. The new design is feasible and has a cost reduction (savings) of 5.08 %. Further exploration in this direction of reducing u_4^3 is shown in the subsequent rows of the table.
7. When u_4^3 is reduced to ten, the design becomes infeasible as the factor of safety fails (in this case, both factors fall below their allowable limits). The last row of Table 7.4 shows that when $u_4^6 = 12$, the design is feasible and the overall reduction in the cost is 10.32 %. As the factors of safety are close to their permissible limits, the search for optimal design is stopped at the 7th iteration and the design given below (u^6) is taken as a satisfactory design.

$$u^6 = (9, 3, 17, 12, 8, 6, 12.5, 4.5, 3). \tag{7.6}$$

Thus, starting from an existing infeasible design, we are able to produce a satisfactory design with substantial reduction in the material cost. Each iteration took less than 2 min of running time whereas using *fmincon* of MATLAB no output was seen even after 2 h of computer running time.

8 Computer Programs Prepared for Decision Support

As a part of this work, a computer program is developed in Microsoft Excel and also in MATLAB. The programs are developed using the procedures described above. The MATLAB program is developed for using it as a tool to optimize the design using *fmincon* function and m-files of MATLAB. On the other hand, the Excel program is more like a decision support system. Besides determining the factors of safety, the program can be effectively used for studying a host of slip circles, evaluating the factor of safety on a given chord, finding out the chord to which a slip circle belongs to, finding out the factor of safety with respect to a slip circle which is at a given distance from a given chord and so on. This program has turned out to be extremely useful in analyzing the existing or any given designs. Interested readers may contact G. S. R. Murthy at murthygsr@gmail.com for further details on the Excel program and a copy of it.

The excel program has six sheets and it is a macro-based program. The sheets are titled "Specs," "SlipCirlce," "ChordAnalysis," "DamAnalyzer," "Optimizor," and "Parameters." Of these, the last sheet titled "Parameters" is meant for the program and has no relevance to the user interaction and hence can be ignored by the user. The functions of the other sheets are briefly described below.

Specs The Excel program is meant for analyzing the factor of safety of a given design. This program can be used effectively to analyze earth dams having up to four berms on either side. This sheet is meant for providing all the necessary inputs to compute the factors of safety. It is designed in such a way that the user can look at the graph of the design and provide the inputs against the dimensions in the graph. The user must specify: (i) number of berms (it takes the same number of berms for upstream and downstream but by specifying negligible berm widths one can analyze designs having different number of berms on upstream and downstream sides), (ii) FRL height, core height, dam height, (iii) densities for driving force and resistance, coefficients of cohesion, and friction for different zones (iv) all widths and heights of berms and slants, and (v) widths of core top and bottom and foundation height. Once the data are provided, the user should update the data by clicking on the "Yellow Push Button." User can modify the data partially or fully to analyze different designs. This is particularly useful to study the effect of changing design variables while searching for optimal designs.

SlipCircle This sheet is meant for finding the safety over a given slip circle. The circle is identified by chord parameters and the distance from the center of the circle to the chord. The reason for designing the inputs this way is that the user can get the factors of safety of a number of slip circles whose common chord is the chord specified. It takes five numbers as input, namely, x_1, x_2, d, and δd (Delta d) and number of circles required (say n). Here $x_1 and\ x_2$ are the chord parameters of $Chord(x_1, x_2)$, d is the distance from center to the chord. After giving these inputs, the user should click on the push button titled "Slip Circle Analysis." Upon clicking, n slip circles are analyzed and the following outputs are displayed for each of these circles: (i) distance from the center to the chord

($= d_i = d + (i-1)\delta d$ for the i^{th} circle), (ii) center coordinates and the radius (r_i) of the circle, (iii) driving force, cohesive force, resistance, factor of safety, and the parameter $\lambda_i = r_i/d_i$. If the input d is less than the smallest possible value for a valid slip circle, then the program will produce the output starting with smallest possible d. In this sheet, an option is provided to find out the chord parameters for a given slip circle whose center coordinates and the radius are known. With this option, the user can analyze the slip circles whose parameters are known in either format (through chord and distance or center coordinates and radius).

ChordAnalysis This sheet provides the option of finding the minimum factor of safety of all circles associated with the chord. The program uses the procedure described in Sect. 6.3. It must be remembered that the minimum factor of safety determined here is an approximation. The inputs for this module are just the chord parameters. The output is displayed for all the circles encountered in the procedure and the output format is similar to that of "SlipCircle" sheet format.

DamAnalyzer This sheet is meant for finding out the upstream and downstream factors of safety of the dam. The user is asked for four numbers as inputs, namely, L, R, DeltaX1 (for δx_1), and DeltaX2 (for δx_2) can specify the range of chord parameters optionally. The program will explore the chords with parameters between L and s_R for the upstream and the chords with parameters between s_L and R for the downstream with increments δx_1 and δx_2 using the procedure described in Sect. 6.5 (see the steps in that procedure).

Optimizer This sheet provides a convenient format for searching for an optimal solution manually. The table in this sheet is self-explanatory and the user can change one or more of the design variables simultaneously and evaluate.

9 Summary

In this chapter we have looked the problem of designing earth dams as an optimization problem. Majority of the dams constructed are earth dams as they are most economical. Factor of safety is an important and mandatory aspect of designing earth dams. The factor of safety is evaluated using slip circle method. This method is most widely used and recommended method for evaluating the factor of safety. As per the mandatory recommendations, the factors of safety should be evaluated under different conditions. In this chapter we have examined only one of these conditions.

It is understood that there is no good software available to the engineers to design earth dams. At present, couple of excel based software programs are being widely used for evaluating the manually developed designs. These programs are not interactive and they only evaluate at some fixed slip circles and the results for those circles are displayed. It is not possible to use the software to evaluate the factor of safety for a slip circle that an engineer wants to explore. Toward this end, the software developed under this work is highly interactive and can be used to explore various possibilities.

The problem of finding the critical circle is a complex one and involves exploring infinitely many slip circles. In this work we have introduced a new approach, the chord based approach, using which the search process is reduced to that of finding the critical chord instead of critical slip circle. In this approach, for each chord, the factor of safety is approximately a convex function of the parameter λ which is the ratio of distance from the center of slip circle to the chord to the radius of the slip circle. Using this observation, the critical circle with respect to each chord is efficiently determined.

The problem of designing the dam is formulated as a nonlinear program with nonlinear objective function and linear and nonlinear constraints. The constraints pertaining to the minimum factors of safety have no closed forms and have to be evaluated using only numerical methods and computer programs. An attempt was made to solve the design optimization problem using the *fmincon* function of MATLAB but with no success. As an alternative, a nonlinear programming algorithm was adopted and this appears to be promising. In Sect. 7 we discussed a way of solving the problem approximately. But compared to the present methods of computing factor of safety, our approach is far superior in that it explores a huge number of slip circles.

The methodology is illustrated with a live example using Pendekal dam. It is found through the analysis using the software developed that the actual design falls marginally short of upstream factor of safety. The optimal design for this dam determined using a coordinate search method proposed in this work exhibited approximately 10 % savings.

Acknowledgement The authors wish to thank two B. Tech. students, Vamsidhar Reddy and Aditya, who were assigned parts of this when they were doing their internship with one of the authors. We appreciate their efforts and contribution in this work.

Problems

1. Consider the Pendekal example. Doing the following exercises will help in understanding the project better (you may use the Excel files "ForPendekal" and "ForChevella" placed in the web for doing these exercises).
   a) What is the width of the dam?
   b) Divide the width into 15 equal parts and find out the corresponding x coordinates.
   c) For each of these x coordinates, find out the corresponding points $(x,\ y)$ on the shell, core, phreatic line, MDDL line, and on a slip circle of your choice.
   d) For each point $(x,\ y)$, determine the zone in which the point lies.
   e) Using the coefficients given in the table in sheet titled "Specs," find out the weight of each slice generated by your points.
   f) At each of the 15 points $(x,\ y)$ on the arc of your slip circle, find out the angle between the vertical line and the normal to the circle, and determine the normal and tangential components of the weight of the slice. Using the

formulae described in Sect. 5 of this chapter, compute the driving forces and resistance at each of the 15 points. Also compute the Cohesion.

g) Compute the factor of safety with respect to your slip circle.

2. A slip circle is considered invalid if the arc of the circle intersects a vertical line at two distinct points inside the dam. Consider the $Chord(1, 28)$. Determine the slip circle whose center $(x,\ y)$ is outside the dam and lies on the perpendicular bisector of the chord at a distance of two units from the chord. Is this an invalid slip circle? Determine the valid slip circle associated with this chord, that is, find out the center and radius, whose center is nearest to the chord.
3. In Sect. 7 of this chapter, we have used a nonlinear programming algorithm to search for an optimal solution. In this approach we have changed one variable at a time. It is possible to change more than one variable at a time. In fact, we can choose a direction d and explore the design change in that direction. To fix the ideas, Let x^0 be the initial design and let d be a nonzero vector whose dimension is same as that of x^0. Consider the new design $x^0 + d$. Examine the new design. Instead of exploring the problem in one direction, we may choose a set of directions, say, $d^1, d^2, \ldots, d^k$ and shift the initial point to the best of $x^0 + d^j,\ j = 1, 2, \ldots, k$. Once the initial point is shifted, we can repeat the process with the new initial point.

$$D = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \end{bmatrix}$$

One way of selecting the set of directions is to use orthogonal arrays which are extensively used in Design of Experiments[4, 5, 6]. Each column of the matrix D above is extracted from L_{12} orthogonal array. Let D_j denote the jth column of D. Explore the new designs given by $x^0 + \frac{1}{2}d^j,\ j = 1, 2, \ldots, 12$ where x^0 corresponds to the design given in first row of Table 7.4. Which of the new designs are good? Can you rank them?

4. There is a proposal to construct an ED, called the Chevella ED, near Hyderabad, Andhra Pradesh, India. One of the proposed designs is shown in the Fig. 7.20. Due to limited availability of core material, the core dimensions are also fixed as core top width as 3 m and bottom width as 23.3 m. Find the factors of safety for this design. Since the core dimensions are fixed, the objective boils down to minimizing the volume of shell material which in turn amounts to minimizing the shell area or equivalently the dam cross-sectional area. The input data and constraints for designing this dam are as follows:

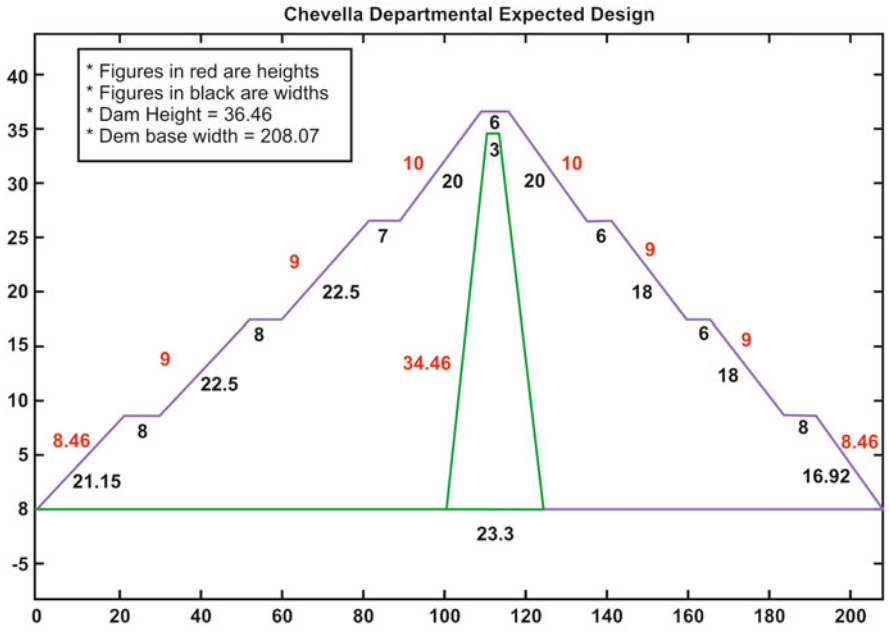


Fig. 7.20 A proposed design for Chevella ED

The fixed parameters are: dam height = 35.185, core height = 31.185 m, FRL = 30.185 m, foundation depth is 10 m, dam top width = 6 m, and MDDL = 20.3 m.

The constraints are: (i) the slopes of shell slanting edges must be at least one, (ii) the dam base width must not exceed 220 m, (iii) each berm should be at least 3 m wide.

Densities and Coefficients of Friction and Cohesion: These are given in the Excel file titled "ForChevell." Find an optimal design with two berms on each side. Compare your design with that given in Fig. 7.20.

References

1. BIS. (1975). Indian standard code of practice for stability analysis of earth dams, IS 7894, pp. 10–13.
2. Fellenius, W. (1936). *Calculation of the stability of earth dams*. Transactions of the 2nd International Congress on Large Dams, ICLD, Washington, DC, pp. 445–459.
3. Hammouri, N., Malkawi, A., & Yamin, M. (2008). Stability analysis of slope using the finite element method and limiting equilibrium approach. *Bulletin of Engineering Geology and the Environment, 67*(4), 471–478 (Copyright Springer-Verlag).
4. Park, S. H.. (1996). *Robust design and analysis for quality engineering*. London: Chapman & Hall.
5. Peace, G. S. (1993). *Taguchi methods*. New York: Addison–Wesley.

6. Phadke, M. S. (1989). *Quality engineering using robust design*. Englewood Cliffs: Prentice-Hall.
7. Sarma, S. K. (1979). Stability analysis of embankments and slopes. *Journal of the Geotechnical Engineering Division, ASCE, 105*(GT12), 1511–1524.
8. Sinha, B. N.. (2008). Advance methods of slope-stability analysis for earth embankment with seismic and water forces. http://www.civil.iitb.ac.in/dns/IACMAG08/pdfs/Q04.pdf.

Chapter 8
Optimal Scheduling of a Multiunit Hydro Power Station in a Short-Term Planning Horizon

Alberto Borghetti, Claudia D'Ambrosio, Andrea Lodi and Silvano Martello

1 Introduction

In many countries, an electricity market is implemented so that, for each period of the day that corresponds to a market session, the revenue of an electric generating station is given by the product between the amount of electric energy produced by it in the period and the corresponding market clearing price.

In general, the market operator determines the market clearing price on the basis of the bids provided by the market participants (electricity producers and consumers or traders). There is a consistent literature on electricity markets that describes the various mechanisms (see, e.g., [18, 20]). When a generating company has a small fraction of the production capacity compared with the entire system, we can assume that it is not able to influence the market results with its own bid, i.e., it behaves as a price-taker. The decision of the generating company is then limited to the scheduling of its power stations, i.e., to the decision of the power output level of every station during each of the periods that constitute the planning horizon.

The current chapter deals with a generating company that owns just one hydroelectric power station for the production of electricity by using the potential energy of water stored in a reservoir (normally created in a river valley by a dam) through

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_8) contains supplementary material, which is available to authorized users.

A. Borghetti (✉) · A. Lodi · S. Martello
DEI, University of Bologna, 40136, Bologna, Italy
e-mail: alberto.borghetti@unibo.it

A. Lodi
e-mail: andrea.lodi@unibo.it

S. Martello
e-mail: silvano.martello@unibo.it

C. D'Ambrosio
CNRS LIX, École Polytechnique, 91128, Palaiseau, France
e-mail: dambrosio@lix.polytechnique.fr

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_8, © Springer Science+Business Media New York 2015

water turbines and electric generators connected to the electric power transmission network (the *grid*). The stored energy depends on the product between the water volume and the difference in height between the level in the reservoir and in the tailrace, i.e., the level of the water outflow from the turbines. This height difference is called *gross head*, as some potential energy is lost by transferring the water from the reservoir to the turbines. The lost energy, called *head loss*, is directly proportional to the square of the water velocity and to the pipe's length. In order to exploit the change of the electricity value in different periods of the day, some hydro stations, called *pumped-storage stations*, are also equipped with pumps that use low-cost, off-peak electric power to move water from a lower elevated reservoir to the reservoir at a higher elevation, which feeds the generating station.

The structure of every hydro power station is rather unique due to the specific characteristics of valleys and rivers. However, a typical scheme includes, besides the already mentioned reservoir:

- One or more turbines located in the so-called powerhouse, each coupled with its own electrical generator.
- A man-made hydraulic system, i.e., a collection of components and conduits. Such a system is used to control the water flow by transferring water from the reservoir to the turbines and from the turbines to the tailrace (and vice versa, in the case of pumps) with minimum loss of energy. At least the final part of the hydraulic system feeding each turbine is constituted by an almost vertical enclosed pipe, called *penstock*, that is affected in general by hydraulic losses due to the high water speed inside.

As mentioned earlier, more than one turbine generator group or unit may be located in the same power station. In pumped-storage stations, special turbines can be used also as pumps when pulled to rotate in the reverse direction by an electric motor. Otherwise, the water is pumped by an appropriate hydraulic machine. In any case, during pumping, the entire production of electricity to the grid by the station is usually stopped.

We consider here a price-taker generating company that wants to optimize the operation of a pump-storage multiunit hydro power station for a given planning horizon, typically 1 day or 1 week. The problem is to determine the commitment and the power generation of the plant so as to maximize the revenue given by power selling. In this chapter, we use the euro (€) as monetary unit.

All the turbines of the plant are assumed to be fed by the same reservoir. The considered planning horizon is limited to a short term in order to assume that natural inflows to the reservoir and prices are known as they are previously forecast without taking into account the stochastic aspects of the problem.

Several approaches have been proposed for the solution of this problem. (The reader is referred to [16] for a detailed overview.) In [17] the problem was formulated as a simple linear programming (LP) model by neglecting costs and constraints relevant to startups and shutdowns of the units. In [6] a nonlinear programming (NLP) model with some simplified assumptions was introduced. Ad hoc heuristics were proposed by several authors, such as Sinha et al. [19] and Orero and Irving [15]. In [14] a

multistage optimization algorithm was proposed for the development of the optimal bidding strategies of an individual pumped-storage unit owner in a competitive electricity market. In [10] the large-scale mixed-integer NLP problem of determining the optimal scheduling of hydro power plants in a hydrothermal interconnected system is considered: the authors use Lagrangian relaxation decomposition strategies and a sequential quadratic programming algorithm to solve nonlinear subproblems. Various *mixed-integer linear programming* (MILP) approaches have been presented in the literature, see, e.g., [1, 4, 7, 8, 9, 11].

Although we limit our analysis to the deterministic solution of the scheduling of a single power station, the problem is especially interesting both because of its practical relevance and of the difficulties induced by its nonlinear aspects. Namely, an accurate model needs to take into account the nonlinear relationship between the unit electrical power output and the corresponding water flow from the reservoir. Moreover, it should also represent the so called *head effect*, i.e., the influence on power production of the water level in the reservoir.

In the current chapter, we focus on the modeling of this nonlinear characteristic. We show how it can be efficiently and accurately dealt with by using MILP techniques, and we compare the results produced by such an approach with those obtained by directly solving the "natural" *mixed-integer nonlinear programming* (MINLP) model. Indeed, the high efficiency of modern MILP software tools, both in terms of solution accuracy and computing time, encourages their use also in the solution of nonlinear problems. The proposed MILP model allows one to accurately represent the main technical and operating characteristics of a pump storage multiunit hydro power plant, and turns out to be computationally solvable for a planning horizon of one week.

For the general structure of the MILP model, we follow the one proposed in [9]. We show that this basic MILP model can be enhanced by taking into account some additional characteristics of the hydro units, such as ramp transition constraints and pump storage operating mode. In addition, we briefly describe a more sophisticated modeling of the head effect through a specialized approximation methodology that improves the overall accuracy of the linear approximation.

The chapter is organized as follows. The next section details the adopted nomenclature. In Sect. 2 we provide a brief description of the problem in the case study. In Sect. 3 we introduce MINLP and MILP models. In Sect. 4 three models are computationally compared through experiments on real-world instances of the problem. A summary is finally given in Sect. 5.

1.1 Nomenclature

1.1.1 Sets

$T = \{1, \ldots, \bar{t}\}$ = set of time periods considered;
$J = \{1, \ldots, \bar{n}\}$ = set of turbine/pump units.

1.1.2 Parameters

$I_t =$ predicted water inflow in period t ($t \in T$) [m^3/s];
$\Pi_t =$ unit price of the power generated/consumed in period t ($t \in T$) [€/MWh];
$\tau =$ period duration expressed in hours [h];
$C_j =$ start-up cost of unit j as a turbine ($j \in J$) [€];
$D_j =$ start-up cost of unit j as a pump ($j \in J$) [€];
$\underline{Q}_j, \overline{Q}_j =$ min and max flow value in turbine j ($j \in J$) (when the turbine is on) [m^3/s];
$\overline{P}_j =$ max power produced by turbine j ($j \in J$) [MW];
$Q^-, Q^+ =$ water flow max ramp down and ramp up (in m^3/s) per hour [m^3/(s h)];
$\underline{V}, \overline{V} =$ min and max water volume in the basin [m^3];
$V_0 =$ water volume in the basin in period 0 [m^3];
$V_{\bar{t}} =$ target (final) water volume in the basin [m^3];
$\overline{S} =$ max water spillage [m^3/s];
$W_j =$ water needed to start-up turbine j ($j \in J$) [m^3/s];
$Y_j =$ water needed to start-up pump j ($j \in J$) [m^3/s];
$E_j =$ energy needed to start-up pump j ($j \in J$) [MWh];
$Q_{j0} =$ flow in turbine j in period 0 ($j \in J$) [m^3/s];
$G_{j0} =$ status of turbine j in period 0 ($j \in J$) [1 on, 0 off];
$U_{j0} =$ status of pump j in period 0 ($j \in J$) [1 on, 0 off];
$Q_j^- =$ flow pumped by pump j ($j \in J; Q_j^- < 0$) [m^3/s];
$P_j^- =$ power consumed during pumping by pump j ($j \in J; P_j^- < 0$) [MW];
$\underline{\Theta} =$ min released water in each period [m^3/s].

1.1.3 Variables

$q_{jt} =$ water flow in unit j in period t ($j \in J, t \in T$), with $q_{j0} = Q_{j0}$ [m^3/s];
$v_t =$ water volume in the basin in period t ($t \in T$), with $v_0 = V_0$ [m^3];
$p_{jt} =$ power generated or consumed by unit j in period t ($j \in J, t \in T$) [MW];
$s_t =$ spillage in period t ($t \in T$) [m^3/s];
$w_{jt} =$ shutdown phase of turbine j in period t ($j \in J, t \in T$) {1 shutdown, 0 oth.};
$\widetilde{w}_{jt} =$ start-up phase of turbine j in period t ($j \in J, t \in T$) {1 started up, 0 oth.};
$g_{jt} =$ status of turbine j in period t ($j \in J, t \in T$), with $g_{j0} = G_{j0}$ {1 on, 0 off};
$y_{jt} =$ shutdown phase of pump j in period t ($j \in J, t \in T$) {1 shutdown, 0 oth.};
$\widetilde{y}_{jt} =$ start-up phase of pump j in period t ($j \in J, t \in T$) {1 started up, 0 oth.};
$u_{jt} =$ status of pump j in period t ($j \in J, t \in T$), with $u_{j0} = U_{j0}$ {1 on, 0 off}.

Some additional parameters and variables, introduced to linearize the model, are defined in Sect. 3.

2 Brief Description of the Problem in the Case Study

We consider a hydro power plant with a single turbine (turbine 1) fed by a reservoir of maximum and minimum capacities equal to 33×10^6 m^3 and 15×10^6 m^3, respectively. The maximum flow taken by the turbine is 42 m^3/s. The turbine can be also used as a pump with a constant power consumption of 21.5 MW corresponding to a 27 m^3/s pumped flow and a start-up energy consumption equal to 2 MWh. The planning horizon is one day, divided in 12 intervals of 2 h each. The initial reservoir water content is equal to about 21×10^6 m^3 with the turbine not in operation. A constant inflow equal to 5.32 m^3/s is predicted. For each of the periods, the electricity market price is forecast. The values of the model parameters are listed below (using the units of measurement specified in Sect. 1.1).

$$\bar{t} = 12$$
$$\overline{n} = 1$$
$$I_t = 5.32 \text{ for all } t \in T$$
$$\Pi = \{50.17, 35.17, 35.15, 57.17, 90.00, 146.94, 95.00, 95.00, 90.61, 60.39, 95.62, 60.25\}$$
$$\tau = 2 \text{ for all } t \in T$$
$$C_1 = 75$$
$$D_1 = 75$$
$$\underline{Q}_1 = 8.5$$
$$\overline{Q}_1 = 42$$
$$Q^- = 70$$
$$Q^+ = 70$$
$$\underline{V} = 15,000,000$$
$$\overline{V} = 33,000,000$$
$$V_0 = V_{\bar{t}} = 21,078,580$$
$$\overline{S} = \infty$$
$$W_1 = 0.0583333$$
$$Y_1 = 0$$
$$E_1 = 2$$
$$Q_{10} = 0$$
$$G_{10} = 0$$
$$U_{10} = 0$$

$$Q_1^- = -27$$
$$P_1^- = -21.5$$
$$\underline{\Theta} = 0$$

Following [10], the relationship between power p (in megawatt) produced by turbine j=1 in period t is described by the following polynomial function (superscripts h and k denoting exponents) in two variables: the value of water flow q (in cubic meter per second) and that of water volume v in the basin (in cubic meter):

$$p_{1t} = \frac{9.81}{1000} \cdot q_{1t} \cdot \sum_{h=0}^{6} \left(L_h q_{1t}^h \left(\sum_{k=0}^{6} K_k v_{1t}^k - \underline{L} - R_0 q_{1t}^2 \right) \right)$$

with
$R_0 = 0.01$;
$L = (L_h : h = 0, 1, \ldots, 6) = \{4.09863600116008, -1.25535942295343,$
$0.160530264942775, -9.76201903589132 \cdot 10^{-3}, 0.000309429429972963,$
$-4.92928898248035 \cdot 10^{-6}, 3.11519548768 \cdot 10^{-8}\}$;
$\underline{L} = 385$;
$K = (K_k : k = 0, 1, \ldots, 6) = \{307.395,\ 3.88 \cdot 10^{-5},\ -4.37 \cdot 10^{-12},\ 2.65 \cdot 10^{-19},$
$-8.87 \cdot 10^{-27},\ 1.55 \cdot 10^{-34},\ -1.11 \cdot 10^{-42}\}$,
where the numerical coefficients L_h and K_k refer to the specific turbine used for this example.

3 Mathematical Models

The next sections formally define the mathematical models that will be computationally evaluated in Sect. 4. Namely, in Sect. 3.1 we introduce the linear objective function, while the linear constraints, common to both the MINLP and MILP models, are introduced in Sect. 3.2. Finally, the nonlinear constraints and two of their possible linearizations are discussed in Sect. 3.3.

3.1 Linear Objective Function

In the following, we will denote the parameters by upper capital letters, and the variables by lower capital letters.

Start by observing that the two typical ways to handle the pump start-up of a unit j can be obtained through the parameters as follows:

- If unit j is started up as a pump by another turbine then no energy is consumed, and hence we have $E_j = 0$ on input, but the water spillage must be considered.

- If instead unit j is started up through the energy provided by the external power network, then the input has $Y_j = 0$.

Variables q, v, p, and s must obey obvious bounding constraints. Namely, for all $t \in T$ and $j \in J$, we must have

$$\begin{aligned} Q_j^- &\le q_{jt} \le \overline{Q}_j; \\ \underline{V} &\le v_t \le \overline{V}; \\ P_j^- &\le p_{jt} \le \overline{P}_j; \\ 0 &\le s_t \le \overline{S}. \end{aligned}$$

In addition, for any period $t \in T$, the values of variables q and p for turbine/pump unit j are determined by the three possible cases that can occur:

TP10: If unit j generates power, i.e., $g_{jt} = 1$ and $u_{jt} = 0$, then $q_{jt} > 0$ and $p_{jt} > 0$;
TP01: If unit j pumps water, i.e., $g_{jt} = 0$ and $u_{jt} = 1$, then $q_{jt} < 0$ and $p_{jt} < 0$;
TP00: If unit j is inactive, i.e., $g_{jt} = u_{jt} = 0$, then $q_{jt} = p_{jt} = 0$.

In the proposed model we are interested in the maximization of the profit provided by the sum, over all periods, of the revenues produced by power selling, decreased by the sum of the start-up costs (if any) of the turbine/pump units. In other words, we have a linear objective function

$$\max \sum_{j \in J} \sum_{t \in T} \left(\tau \, \Pi_t \, p_{jt} - C_j \, \widetilde{w}_{jt} - (D_j + \Pi_t E_j) \widetilde{y}_{jt} \right), \tag{8.1}$$

in which the first term can have a negative value in the periods when the unit works as a pump.

Concerning the constraints, we can distinguish a set of linear constraints and a set of nonlinear constraints, which will be linearized in order to be able to solve the overall model through MILP techniques.

3.2 Linear Constraints

We first introduce the linear constraints that model the relationships between flow, volume, and status of the turbine/pump units, and then discuss their logical meaning:

$$v_{\bar{t}} - V_{\bar{t}} = 0 \tag{8.2}$$

$$v_t - v_{t-1} - 3600 \, \tau \left(I_t - \sum_{j \in J} q_{jt} - s_t \right) = 0 \text{ for all } t \in T \tag{8.3}$$

$$q_{jt} - (Q_j^- \, u_{jt} + \underline{Q}_j \, g_{jt}) \ge 0 \text{ for all } j \in J, t \in T \tag{8.4}$$

$$q_{jt} - (Q_j^- u_{jt} + \overline{Q}_j \, g_{jt}) \leq 0 \text{ for all } j \in J, t \in T \tag{8.5}$$

$$\sum_{j \in J} (q_{jt} - q_{j(t-1)}) + \tau Q^- \geq 0 \text{ for all } t \in T \tag{8.6}$$

$$\sum_{j \in J} (q_{jt} - q_{j(t-1)}) - \tau Q^+ \leq 0 \text{ for all } t \in T \tag{8.7}$$

$$s_t - \sum_{j \in J} \left(W_j \, \widetilde{w}_{jt} + Y_j \, \widetilde{y}_{jt} \right) \geq 0 \text{ for all } t \in T \tag{8.8}$$

$$\sum_{j \in J} q_{jt} + s_t - \underline{\Theta} \geq 0 \text{ for all } t \in T \tag{8.9}$$

$$g_{jt} - g_{j(t-1)} - (\widetilde{w}_{jt} - w_{jt}) = 0 \text{ for all } j \in J, t \in T \tag{8.10}$$

$$\widetilde{w}_{jt} + w_{jt} \leq 1 \text{ for all } j \in J, t \in T \tag{8.11}$$

$$u_{jt} - u_{j(t-1)} - (\widetilde{y}_{jt} - y_{jt}) = 0 \text{ for all } j \in J, t \in T \tag{8.12}$$

$$\widetilde{y}_{jt} + y_{jt} \leq 1 \text{ for all } j \in J, t \in T \tag{8.13}$$

$$g_{jt} + u_{kt} \leq 1 \text{ for all } j, k \in J, t \in T \tag{8.14}$$

$$\sum_{j \in J} u_{jt} \leq \overline{n} - 1 \text{ for all } t \in T. \tag{8.15}$$

Constraint (8.2) ensures that, at the end of the considered planning horizon $\overline{t}$, the water volume in the basin is equal to the desired target $V_{\overline{t}}$. Constraints (8.3) impose, for each $t \in T$, the water conservation between period $t-1$ and period t. Constraints (8.4) establish lower bounds on the flows in the turbines (see the three cases TP10, TP01, and TP00 discussed in Sect. 3), while constraints (8.5) similarly establish upper bounds on the same quantities. Constraints (8.6) and (8.7) impose that the variation of the flow between two consecutive time periods does not exceed the maximum ramp-down and ramp-up, respectively. Constraints (8.8) ensure that the water spillage is sufficient to start up a pump or a turbine. Constraints (8.9) impose that the overall amount of water (total water flow plus spillage) released in each period obeys the lower bound on the released water. Constraints (8.10) and (8.11) establish the rules to switch on/switch off the turbines. The companion constraints (8.12) and (8.13) establish the same for the pumps. Constraints (8.14) simultaneously impose that: (a) if a turbine is on then no pump can be on, and (b) if a pump is on then no turbine can be on. Constraints (8.15) are only effective if the turbines are used to start up the pumps: in this case at least one pump must be off because there is no turbine available to start up the last pump.

3.3 Nonlinear Constraints and their Linearization

It is known that the performance of a hydro turbine depends on the rate of water discharge and on the net hydraulic head. In turn, the value of the net head depends on the water level in the reservoir, the tailrace level and the penstock losses, which

are function of the water flow. Thus, the power generated from a hydro unit is related to the water flow and the reservoir characteristics.

3.3.1 Nonlinear Formulation

For a generic hydro generator unit, the power output p is expressed as a nonlinear function φ of the water flow q and the water volume v in the reservoir. Such a function includes the nonlinear relationship that links the net head value to the water volume and the water flow, as well as the electric loss of the generator, i.e.,

$$p = \varphi(q, v). \tag{8.16}$$

Of course each unit is characterized by a specific φ function and an example of (8.16) can be found in [15].

Note that, unfortunately, even for a prefixed volume $\widetilde{v}$, the power production is a nonlinear and nonconcave function of the water flow. The net head variation can only be ignored for relatively large reservoirs, where the power generation only depends on the water flow.

The MINLP model that will be tested in Sect. 4 directly completes the system (8.1)–(8.15) by (8.16) in its nonlinear and nonconcave form.

3.3.2 Linearized Formulation

An accurate approximation of φ is crucial to obtain effective MILP models. A natural linearization of (8.16) can be obtained by considering a parametric number of water volumes in the reservoir, say $\widetilde{v}^{\,1}, \widetilde{v}^{\,2}, \ldots, \widetilde{v}^{\,k}$ and interpolating, through piecewise linear approximation [13], function

$$p = \varphi|_{\widetilde{v}^{\,r}}(q) \tag{8.17}$$

for each $\widetilde{v}^{\,r}$. This approach, originally proposed in [9] for a *fixed number* of water volumes, was later generalized in [4] and requires to amend model (8.1)–(8.15) through the addition of new variables and constraints. We show in the following how this can be obtained in our case.

We consider $\overline{r}$ volume intervals and $\overline{z}$ coordinates (*breakpoints*) on the axis of water flow. We divide the range of values of the water volume in the reservoir into $\overline{r}$ intervals, and represent the rth interval $[H_{r-1}, H_r]$ by its midpoint, v^r, for $r = 1, \ldots, \overline{r}$. This is the discretization that we will use for water volume in the reservoir. Let $R = \{1, \ldots, \overline{r}\}$ and $Z = \{1, \ldots, \overline{z}\}$. Moreover, we consider the following additional parameters:

- $[H_{r-1}, H_r)$ = extreme water volumes for interval r ($r \in R$) [m^3];
- Q_{ji} = flow in turbine j at breakpoint i ($j \in J, i \in Z$) [m^3/s];
- P_{jir} = power from turbine j at breakpoint i, for interval r ($j \in J, i \in Z, r \in R$) [MW];

- $\Delta P_{jr} = \max_{i \in Z}\{P_{ji\bar{r}} - P_{jir}\}$ $(j \in J, r \in R)$ [MW],

where ΔP_{jr} represents for turbine j the maximum power difference between intervals r and $\bar{r}$.

The following new variables need to be introduced:

- d_{tr} = membership status of volume v_t wrt interval r
  $\{1$ if $H_{r-1} \le v_t < H_r, 0$ otherwise$\}$ $(t \in T, r \in R)$;
- z_{jti} = contiguity status of q_{jt} wrt to discretized flow Q_{ji}
  $\{1$ if $Q_{j(i-1)} < q_{jt} \le Q_{ji}$ or $Q_{ji} \le q_{jt} < Q_{j(i+1)}, 0$ otherwise$\}$
  $(j \in J, t \in T, i \in Z)$;
- λ_{jti} = weight, in the piecewise linear approximation, of breakpoint i for turbine j in period t $(j \in J, t \in T, i \in Z)$, where $\lambda_{jti} \in [0, 1]$ for all j, t and i.

The linearization of the power production function (8.16) by a parametric number of water volumes is produced by the constraints

$$q_{jt} - \sum_{i \in Z} Q_{ji}\lambda_{jti} - Q_j^- u_{jt} = 0 \text{ for all } j \in J, t \in T \tag{8.18}$$

$$\sum_{i \in Z} \lambda_{jti} - g_{jt} = 0 \text{ for all } j \in J, t \in T \tag{8.19}$$

$$\lambda_{jti} - z_{jti} \le 0 \text{ for all } j \in J, t \in T, i \in Z \tag{8.20}$$

$$z_{jti} + z_{jtk} \le 1 \text{ for all } j \in J, t \in T, i, k \in Z : i < k - 1 \tag{8.21}$$

$$\sum_{r \in R} d_{tr} = 1 \text{ for all } t \in T \tag{8.22}$$

$$p_{jt} - \sum_{i \in Z} P_{jir}\lambda_{jti} - P_j^- u_{jt} - \Delta P_{jr}(1 - d_{tr}) \le 0 \text{ for all } j \in J, t \in T, r \in R \tag{8.23}$$

$$v_t - \sum_{r \in R} H_{r-1} d_{tr} \ge 0 \text{ for all } t \in T \tag{8.24}$$

$$v_t - \sum_{r \in R} H_r d_{tr} \le 0 \text{ for all } t \in T, \tag{8.25}$$

The water flow q_{jt} of turbine/pump j in period t according to the three possible cases discussed in Sect. 3.1 is regulated by constraints (8.18)–(8.21). Precisely,

- If $u_{jt} = 0$, the pump is off and the flow is either zero (if the turbine is off as well, case TP00) or a convex combination of breakpoint flows (case TP10);
- Otherwise, the pump is on (so $g_{jt} = 0$ from (8.14)) and there is a constant negative flow Q_j^- (case TP01).

Indeed, it is easy to see that constraints (8.20)–(8.21) are deactivated by $g_{jt} = 0$ in (8.19), which implies that the TP10 and TP00 are directly modeled by (8.18). *This is because* $g_{jt} = 0$ *implies all* λ_{jti}*'s being 0 as well (due to* (8.19)*), and simply* $z_{jti} \geq 0$ *(due to* (8.20)*), thus no contiguity constraint* (8.21) *is active for the pair* j, t. If instead $g_{jt} = 1$, constraints (8.19) impose that the breakpoint weights sum up to one. Constraints (8.20) are classical indicator constraints, allowing any λ_{jti} to be nonzero if and only if binary variable z_{jti} is one. It follows that constraints (8.20) and (8.21) together ensure that at most two weights can take a positive value and if this occurs for exactly two weights then they must be contiguous.

Similarly, constraints (8.22) and (8.23) are used to express the power p_{jt} of turbine/pump j in period t for volume interval r, in the same three cases. Precisely, constraints (8.22) trivially say that exactly one binary variable d_{tr} takes the value one. Hence, in the unique volume interval, say $\widetilde{r}$, for which $d_{t\widetilde{r}} = 1$, the last term of (8.23) takes the value 0 and (8.23) itself assumes the same form as (8.18), but with powers instead of flows. As a consequence, and by using the same arguments, constraints (8.23) model the three possible cases, the only difference being the "$\leq$" sign instead of "=." However, this has no effect because of the objective function (8.1) *that maximizes the power production thus leading to satisfy the constraint* (8.23) *as equality. For all the other volume intervals* $r \neq \widetilde{r}$*, for which* $d_{tr} = 0$*, constraint* (8.23) *is deactivated, i.e., always satisfied, because its last term takes the value* ΔP_{jr}*, which is by definition the largest possible value that* $p_{jt} - \sum_{i \in Z} P_{jir} \lambda_{jti} - P_j^- u_{jt}$ *can take.* Finally, Eqs. (8.24) and (8.25) define, for each time period t, the two extreme water volumes of the interval where the computed volume v_t lies.

The complete MILP model, denoted as "MILP1" in Sect. 4, is then

$$\max \sum_{j \in J} \sum_{t \in T} \left(\tau \, \Pi_t \, p_{jt} - C_j \, \widetilde{w}_{jt} - (D_j + \Pi_t E_j) \widetilde{y}_{jt} \right),$$

subject to: (8.2)–(8.15);
(8.18)–(8.25);

$$\begin{aligned}
Q_j^- \leq q_{jt} \leq \overline{Q}_j \quad & \text{for all } j \in J, t \in T; \\
\underline{V} \leq v_t \leq \overline{V} \quad & \text{for all } t \in T; \\
P_j^- \leq p_{jt} \leq \overline{P}_j \quad & \text{for all } j \in J, t \in T; \\
0 \leq s_t \leq \overline{S} \quad & \text{for all } t \in T; \\
0 \leq \lambda_{jti} \leq 1 \quad & \text{for all } j \in J, t \in T, i \in Z; \\
w_{jt}, \widetilde{w}_{jt}, g_{jt}, y_{jt}, \widetilde{y}_{jt}, u_{jt} \in \{0, 1\} \quad & \text{for all } j \in J, t \in T; \\
d_{tr} \in \{0, 1\} \quad & \text{for all } t \in T, r \in R; \\
z_{jti} \in \{0, 1\} \quad & \text{for all } j \in J, t \in T, i \in Z.
\end{aligned}$$

In the above MILP model, for any volume v_t belonging, say, to the rth interval $[H_{r-1}, H_r)$, the power production is approximated through a prefixed (static) value

P_{jir} that depends on the turbine and the breakpoint according to (8.23). Of course, the accuracy obtainable in this way heavily depends on the number $\overline{r}+1$ of water volumes H_r. As it will be shown in the next section, if such a number is too high, it substantially affects the computational effort required to solve the MILP, mostly because of its size.

To overcome the above issue, a second, enhanced, and more complicated linearization has been proposed in [4]. The approximation is obtained by keeping $\overline{r}$ at an effective (low) value, thus allowing for a reasonable-size MILP model, but correcting the estimated power production through two-dimensional considerations. Essentially, in the enhanced model, for a volume v_t, belonging, say, to interval $[H_{r-1}, H_r)$, the power production is given by a weighted combination of the values computed for the two extremes H_{r-1} and H_r, instead of being selected as a point on a single piecewise linear function.

Going into the details of such linearization is outside the scope of the present chapter, so the interested reader is referred to [4] for all the details. However, the performance of the enhanced linearization, leading to a MILP model denoted as "MILP2," will be reported in the next section so as to point out how beneficial spending time in improving a linearization can be.

4 MINLP Solution vs. MILP Solution

In this section we present computational results on the case study of Sect. 2. In particular, the models presented in the previous section were tested by running the MINLP solvers Bonmin [2] and Couenne [3] (for the MINLP model) or the MILP solver IBM-CPLEX [12] (for the MILP models) under mathematical programming modeling language AMPL Version 20061102. Note that Bonmin implements a heuristic algorithm for nonconvex MINLPs, while Couenne implements an exact algorithm. IBM-CPLEX implements an exact algorithm for MILP problems. We will distinguish between the simple and the enhanced MILP models by calling them one- and two-dimensional piecewise linear approximations ("MILP1" and "MILP2"), respectively.

The tests were executed by sequentially running the codes on a single processor of an Intel Core2 CPU 6600, 2.40 GHz, 1.94 GB of RAM. For each test, a time limit of 7200 s was imposed.

In Table 8.1, the first column gives the name of the model solved, for the MILP models (namely, "MILP1" or "MILP2"), or the type of solver used, for the MINLP model (namely, "Bonmin" or "Couenne"). The second column provides the number of breakpoints used to linearize the nonlinear functions (only for the MILP models). The third column shows the objective function (o.f.) value in euro of the best solution found by each solver within the time limit. (In the case of MILP models the real objective function value was recomputed after the end of each run.) The fourth column gives the value of the upper bound (UB) in euro provided by the solver when the time limit was reached. (Note that the upper bound provided by Bonmin

Table 8.1 Table

	No. of breakpoints	O.f. value	UB at t.l.	Sol. CPU time	Total CPU time
MILP1	5	14,456.5	–	0.060	0.060
MILP1	10	14,463.9	–	0.130	0.180
MILP1	20	14,512.7	–	0.360	0.380
MILP1	30	14,518.0	–	1.000	1.170
MILP2	5	14,457.2	–	0.076	0.076
MILP2	10	14,463.9	–	0.624	0.624
MILP2	20	14,530.1	–	3.680	8.360
MILP2	30	14,533.1	–	62.940	70.910
Bonmin	–	14,127.6	–	14.520	23.650
Couenne	–	7814.4	46,980.0	587.300	t.l.

is not valid, while the one provided by CPLEX or Couenne is.) The fifth column reports the number of CPU seconds needed to find the best solution. Finally, the last column reports the total number of CPU seconds needed (or "t.l.," if the time limit was reached).

The results were obtained by slightly modifying a couple of constraints and coefficients in order not to create numerical problems for the solvers. In practice, providing very tiny coefficients like $K_6 = -1.11E-42$ can be source of instability and numerical problems. To avoid this, we modified the units of $\underline{V}$, $\overline{V}$, V_0, and $V_{\bar{t}}$ by dividing them by 10,000. Thus, constraint (8.3) becomes

$$v_t - v_{t-1} - 0.3600\,\tau \left(I_t - \sum_{j \in J} q_{jt} - s_t \right) = 0 \quad \text{for all } t \in T$$

and the definition of p_{1t} becomes

$$p_{1t} = \frac{9.81}{1000} \cdot q_{1t} \cdot \sum_{h=0}^{6} \left(L_h q_{1t}^h \left(\sum_{k=0}^{6} K_k (10000^k v_t^k) - \underline{L} - R_0 q_{1t}^2 \right) \right).$$

This demonstrates that the modeling ability of the user is fundamental for making the optimization problem at hand tractable by the solver.

From the figures in Table 8.1, it is clear that the use of a MINLP model, more accurate than the MILP ones, does not pay off in this application. The heuristic solver Bonmin can find a solution that is slightly worse (more than 2 %) than the one found for the MILP1 model with five breakpoints (lowest accuracy) in a CPU time that is two orders of magnitude bigger. The performance of Couenne is even worse: it finds a solution whose value is half the one found for the MILP1 model in a CPU time that is four orders of magnitude bigger. The gap between lower and upper bounds is significant. Moreover, the upper bound does not change after the first 35 s and the lower bound does not improve after 587.3 s.

On the contrary, the performance of the two MILP models is impressive. As the number of breakpoints grows, the accuracy of the approximation improves. However, in the MILP1 model case, the CPU time needed to find the optimal approximated

solution is about 1 s even for the most difficult instance (30 breakpoints). The performance of the MILP2 model is either comparable or one order of magnitude worse with respect to the MILP1 model for what concerns the CPU time, from 0 to 0.12 % better for what concerns the objective function value of the optimal solution found.

To conclude, in this specific optimization problem, the quality of the approximation provided by the linearization is very effective and we can take advantage of the better performance of MILP solvers with respect to the MINLP solvers to find satisfactory heuristic solutions.

5 Summary

We have considered the problem of determining the commitment and the power generation of a single-reservoir pump storage hydro power plant. We have shown how to effectively model the problem by means of piecewise linear approximation of nonlinear and nonconcave constraints, so as to take advantage of the current effectiveness of MILP software tools. Two MILP models with different levels of complexity have been computationally tested and compared with the natural MINLP formulation. The experiments have shown that: (1) in this particular application, piecewise linear approximation is superior to MINLP, and (2) a more sophisticated (though more complicated) approximation might lead to substantial benefits.

6 Practical Exercise

Consider the same problem discussed in the previous sections. Assume that all the data remain the same, except the unit price of the power generated/consumed and the inflows. In particular, $\Pi = \{48.06, 47.56, 47.55, 54.20, 105.00, 110.63, 78.61, 94.91, 188.11, 199.13, 79.63, 58.49\}$ and $I_t = 3.06$ for all $t \in T$. Solve the modified problem with this new data and discuss the output obtained.

References

1. Baillo, A., Ventosa, M., Ramos, A., Rivier, M., & Canseco, A. (2002). Strategic unit commitment for generation companies in deregulated electricity markets. In B. F. Hobbs, M. Rothkopf, R. P. O'Neill & H. P. Chao (Eds.), *The next generation of electric power unit commitment models*. Boston: Kluwer Academic.
2. COIN-OR. (2014). Bonmin: https://projects.coin-or.org/Bonmin.
3. COIN-OR. (2014). Couenne. https://projects.coin-or.org/Couenne.
4. Borghetti, A., D'Ambrosio, C., Lodi, A., & Martello, S. (2008). An MILP approach for short-term hydro scheduling and unit commitment with head-dependent reservoir. *IEEE Transactions on Power Systems, 23,* 1115–1124.

5. Carrión, M., & Arroyo, J. M. (2006). A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem.*IEEE Transactions on Power Systems, 21,* 1371–1378.
6. Catalão, J., Mariano, S., Mendes, V., & Ferreira, L. (2006). Parameterisation effect on the behaviour of a head-dependent hydro chain using a nonlinear model. *Electric Power Systems Research, 76,* 404–412.
7. Chang, C., & Waight, J. (1999). A mixed integer linear programming based hydro unit commitment. In *Power Engineering Society Summer Meeting*, July 18–22, 1999
8. Chang, G., Aganagic, M., Waight, J., Medina, J., Burton, T., Reeves, S., & Christoforidis, M. (2001). Experiences with mixed integer linear programming based approaches on short-term hydro scheduling. *IEEE Transactions on Power Systems, 16,* 743–749.
9. Conejo, A., Arroyo, J., Contreras, J., & Villamor, F. (2002). Self-scheduling of a hydro producer in a pool-based electricity market. *IEEE Transactions on Power Systems, 17,* 1265–1272.
10. Finardi, E., Da Silva, E., & Sagastizabal, C. (2005). Solving the unit commitment problem of hydropower plants via Lagrangian relaxation and sequential quadratic programming. *Computational & Applied Mathematics, 24,* 317–341.
11. Garcia-Gonzalez, J., Parrilla, E., & Mateo, A. (2007). Risk-averse profit-based optimal scheduling of a hydro-chain in the day-ahead electricity market. *European Journal of Operational Research, 182,* 1354–1369.
12. IBM. (2014). ILOG CPLEX. http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/.
13. Keha, A. B., de Farias, I. R., & Nemhauser, G. L. (2004). Models for representing piecewise linear cost functions. *Operations Research Letters, 32,* 44–48.
14. Lu, N., Chow, J., & Desrochers, A. (2004). Pumped-storage hydro-turbine bidding strategies in a competitive electricity market. *IEEE Transactions on Power Systems, 19,* 834–841.
15. Orero, S., & Irving, M. (1998). A genetic algorithm modelling framework and solution technique for short term optimal hydrothermal scheduling. *IEEE Transactions on Power Systems, 13,* 501–518.
16. Padhy, N. (2004). Unit commitment—a bibliographical survey. *IEEE Transactions on Power Systems, 19,* 1196–1205.
17. Piekutowki, M., Litwinowcz, T., & Frowd, R. (1994). Optimal short-term scheduling for a large-scale cascaded hydro system. *IEEE Transactions on Power Systems, 9,* 805–811.
18. Sheblé, G. B. (1999). *Computational auction mechanisms for restructured power industry operation*. Berlin: Springer.
19. Sinha, N., Chakrabarti, R., & Chattopadhyay, P. (2003). Fast evolutionary programming techniques for short-term hydrothermal scheduling. *IEEE Transactions on Power Systems, 18,* 214–220.
20. Soft, S. (2002). *Power system economics: Designing markets for electricity*. New York: IEEE/Wiley.

Chapter 9
Optimizing the Design of Water Distribution Networks Using Mathematical Optimization

Cristiana Bragalli, Claudia D'Ambrosio, Jon Lee, Andrea Lodi and Paolo Toth

1 Introduction

Decaying infrastructure in municipalities is becoming a problem of increasing importance as growing populations put increasing stress on all service systems. In tough economic times, renewing and maintaining infrastructure has become increasingly difficult. As an example, many municipal water networks were installed several decades ago and were designed to handle much smaller demand and additionally have decayed due to age.

For example, consider the case of Modena, a city northwest of Bologna in the Emilia-Romagna region of Italy. We can see in Fig. 9.1 how the population has increased by more than a factor of 2.5 over the last century. The Modena water distribution network comprises 4 reservoirs, 272 junctions, and 317 pipes. Its complexity can be gleaned from Fig. 9.2.

The aim is to replace all the pipes using the same network topology at minimum cost to achieve pressure demands at junctions of the network. Pipes are only available

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_9) contains supplementary material, which is available to authorized users.

C. Bragalli (✉)
DISTART, University of Bologna, 40136 Bologna, Italy
e-mail: cristiana.bragalli@unibo.it

C. D'Ambrosio
CNRS LIX, École Polytechnique, 91128 Palaiseau, France
e-mail: dambrosio@lix.polytechnique.fr

J. Lee
IOE Department, University of Michigan, Ann Arbor, MI 48109-2117, USA
e-mail: jonxlee@umich.edu

A. Lodi · P. Toth
DEI, University of Bologna, 40136 Bologna, Italy
e-mail: andrea.lodi@unibo.it

P. Toth
e-mail: paolo.toth@unibo.it

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_9, © Springer Science+Business Media New York 2015

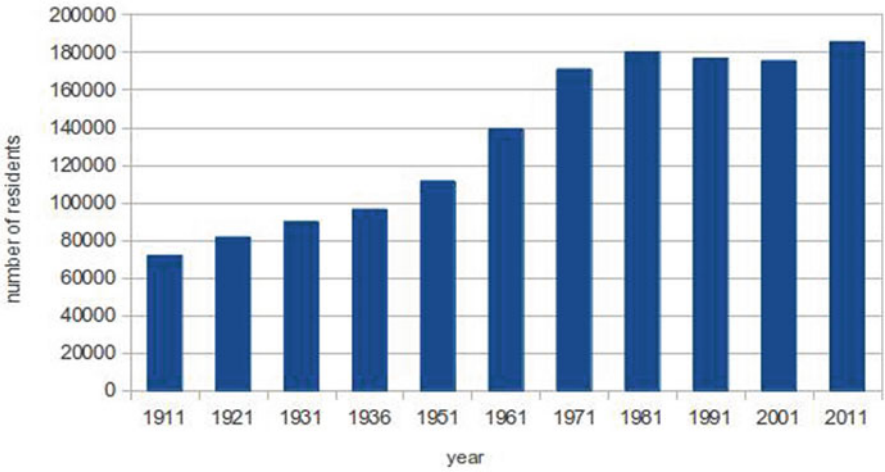


Fig. 9.1 Population of Modena

Fig. 9.2 Water distribution network of Modena

from commercial suppliers that produce pipes in a limited number of diameters. Reservoirs pressurize the network while most pressure is lost due to friction in pipes (some pressure is also lost at the junctions). Our goal is to model the problem using continuous variables as flow rates in the pipes, and pressures at the junctions; and discrete variables for the diameters of the pipes. Noting that pressure loss due to friction behaves nonlinearly, this puts us in the domain of mixed integer nonlinear programming (MINLP). Thus, we will try to obtain a model which is tractable by standard MINLP solvers.

In fluid dynamics, *hydraulic head* equates the energy in an incompressible fluid with the height of an equivalent static column of that fluid. The hydraulic head of a fluid is composed of pressure head and elevation head. The *pressure head* is the internal energy of a fluid due to the pressure exerted on its container and the *elevation head* is the energy given by the gravitational force acting on a column

of fluid, in this case water. As a convention, the units of hydraulic, pressure, and elevation head are meters, i.e., each unit represents the energy provided by a column of water of the height of 1 m. The water in its way from the reservoir to the rest of the junctions loses energy. This loss is called *head loss*. Head loss is divided into two main categories, "major losses" associated with energy loss per length of pipe, and "minor losses" associated with bends and relatively small obstructions. For our purposes, considering that we are interested in networks covering relatively large geographic areas, it suffices to ignore minor losses.

2 Nomenclature

In this section we introduce the sets and parameters, i.e., the data input of our problem.

2.1 Sets

The water network will be represented as a directed graph $G = (N, E)$ where, N is the set of nodes that represent the junctions and E is the set of pipes. Moreover, we define the set of reservoirs S as a subset of N. Finally, for ease of notation, we define $\delta_-(i)$ ($\delta_+(i)$) as the sets of pipes with an head (tail) at junction i.

2.2 Parameters

In the following, we introduce the notation for the data elements that we call parameters.
For each junction but the reservoirs $i \in N \setminus S$ we have:

- elev(i) = physical elevation of junction i, i.e., the height of junction i ($[m]$).
- dem(i) = water demand at junction i ($[m^3/s]$).
- $\text{ph}_{min}(i)$ = lower bound on pressure head at junction i ($[m]$).
- $\text{ph}_{max}(i)$ = upper bound on pressure head at junction i ($[m]$).

Moreover, for each reservoir $i \in S$, we have $h_s(i)$, i.e., the fixed hydraulic head at junction i.

Finally, for each pipe $e = (i, j) \in E$ the following parameters are needed:

- l(e) = length of pipe e ($[m]$).
- $v_{max}(e)$ = upper bound on the velocity of water in pipe e ($[m/s]$).
- $k(e)$ = physical constant depending on the roughness of pipe e.
- $\mathcal{D}(e, r)$ = r-th diameter that can be chosen for pipe e ($[m]$).
- $\mathcal{C}(e, r)$ = cost of the r-th diameter that can be chosen for pipe e ([€/m]).

Note that all the parameters described above are data input and are used to define constraints and objective function of the model.

3 The Mathematical Model for the Problem

We proceed further in describing the problem by specifying a formulation of it.

3.1 Decision Variables of the Problem

First, we specify the variables which are also the expected output of our problem:

- $Q(e) =$ flow in pipe e, for all $e\ in E$ $[m^3/s]$. This is the volume of water which passes through pipe per unit time.
- $D(e) =$ diameter of pipe e, for all $e \in E$ $[m]$.
- $H(i) =$ hydraulic head of junction i, for all $i \in N$ $[m]$.

For modeling purposes, each pipe e has a nominal orientation, and negative flow corresponds to water flow in the direction opposite to the nominal orientation of the pipe.

The goals of the problem are to choose, for each edge, the diameter of the pipe that has to be installed in order to implement the drinkable water distribution network by minimizing the installation costs and satisfying physical and operational constraints.

3.2 Constraints in the Problem

Each reservoir has a fixed hydraulic head that is specified:

$$H(i) = h_s(i), \quad \text{for all } i \in S. \tag{9.1}$$

Typically, there are inequalities imposed that bound the velocity of the water in each pipe. Because the water flow is given by the product between the cross-sectional area of the pipe $\pi(D(e)/2)^2$ and the velocity, for simplicity the velocity bounds can be written as bounds on the flow:

$$-\frac{\pi}{4} v_{max}(e) D^2(e) \leq Q(e) \leq \frac{\pi}{4} v_{max}(e) D^2(e), \quad \text{for all } e \in E. \tag{9.2}$$

Notice how these equations are nonlinear in the diameter variables $D(e)$, but we can think of them as linear in the cross-sectional area $\pi(D(e)/2)^2$.

Next, we have the usual flow-conservation equations:

$$\sum_{e \in \delta_-(i)} Q(e) - \sum_{e \in \delta_+(i)} Q(e) = \text{dem}(i), \quad \text{for all } i \in N \setminus S, \tag{9.3}$$

where, $\text{dem}(i)$ is the demand for water at junction i, $\delta_-(i)$ denotes the set of pipes oriented *into* junction i, and $\delta_+(i)$ denotes the set of pipes oriented *out of* junction i. We only have these equations for junctions that are not reservoirs.

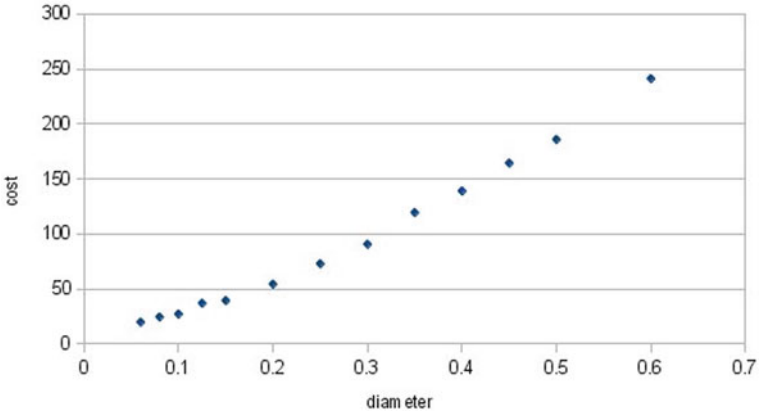


Fig. 9.3 Diameter set for Fossolo instance: diameter (m) and cost per meter (€/m)

Next, we impose upper and lower limits on the pressure head at each junction:

$$\mathrm{ph}_{\min}(i) \leq H(i) - \mathrm{elev(i)} \leq \mathrm{ph}_{\max}(i), \quad \text{for all } i \in N \setminus S. \tag{9.4}$$

Head loss along each pipe is modeled via a nonlinear function of the diameter of the pipe and the flow in the pipe. Typically, one uses the so-called Hazen–Williams equation, which is an empirical formula relating the head loss caused by frictions to the physical properties of the pipe:

$$H(i) - H(j) = \frac{\mathrm{sgn}(Q(e))|Q(e)|^{1.852} \cdot 10.7 \cdot \mathrm{l}(e) \cdot \mathrm{k}(e)^{-1.852}}{D(e)^{4.87}}, \quad \text{for all } e = (i, j) \in E, \tag{9.5}$$

where, we consider the pipe e to be oriented from junction i to junction j. Notice how following intuition head loss is proportional to the length $\mathrm{l}(e)$ of the pipe e. The dependence on the flow $Q(e)$ and the diameter $D(e)$ is nonlinear.

For each pipe e, the available diameters belong to a discrete set of r_e elements. For $e \in E$:

$$d_{min}(e) := \mathcal{D}(e, 1) < \mathcal{D}(e, 2) < \cdots < \mathcal{D}(e, r_e) =: d_{max}(e) \,,$$

and so, we have the equations:

$$D(e) \in \{\mathcal{D}(e, 1), \mathcal{D}(e, 2), \ldots, \mathcal{D}(e, r_e)\}, \quad \text{for all } e \in E. \tag{9.6}$$

For each pipe $e \in E$, there is a cost function $C_e()$ having a discrete specification as a (typically rapidly) increasing function of diameter. That is, $\mathcal{C}(e, r) := C_e(\mathcal{D}(e, r))$, $r = 1, \ldots, r_e$, where:

$$\mathcal{C}(e, 1) < \mathcal{C}(e, 2) < \cdots < \mathcal{C}(e, r_e) \,.$$

Our objective to be minimized is:

$$\sum_{e \in E} C_e(D(e))\, \mathrm{l}(e). \tag{9.7}$$

Our water-network optimization problem is now completely specified as the minimization of (9.7), subject to (9.1)–(9.6).

The mathematical model is reported below:

$$\begin{aligned}
&\min \sum_{e \in E} C_e(D(e))\, l(e) \\
&H(i) = h_s(i), \quad \text{for all } i \in S \\
&-\frac{\pi}{4} v_{max}(e) D^2(e) \le Q(e) \le \frac{\pi}{4} v_{max}(e) D^2(e), \quad \text{for all } e \in E \\
&\sum_{e \in \delta_-(i)} Q(e) - \sum_{e \in \delta_+(i)} Q(e) = \mathrm{dem}(i), \\
&\text{for all } i \in N \setminus S \\
&\mathrm{ph}_{\min}(i) \le H(i) - \mathrm{elev}(i) \le \mathrm{ph}_{\max}(i), \\
&\text{for all } i \in N \setminus S \\
&H(i) - H(j) = \frac{\mathrm{sgn}(Q(e)) |Q(e)|^{1.852} \cdot 10.7 \cdot \mathrm{l}(e) \cdot \mathrm{k}(e)^{-1.852}}{D(e)^{4.87}}, \\
&\text{for all } e = (i, j) \in E \\
&D(e) \in \{\mathcal{D}(e, 1), \mathcal{D}(e, 2), \ldots, \mathcal{D}(e, r_e)\}, \quad \text{for all } e \in E.
\end{aligned}$$

4 Solver

Because of the presence of discrete choices (the pipe diameters) and intrinsic nonlinearity (the dependence of head loss on flow and on pipe diameter), we are in the realm of MINLP . This is a very active area of current research (see [8], for example). We chose to use the open-source code `Bonmin` (Basic Open-source Nonlinear Mixed INteger programming) as our MINLP solver (see [3]). This choice was guided by the facts that (i) it is accessible from the modeling language `AMPL` (see [5]), (ii) it is freely distributed under an open-source license that has fairly liberal terms, (iii) we are able to modify the source code in case that is needed, and (iv) the primary developer, Pierre Bonami, is a very good friend (so *he* can modify the source code for us, in case that would be helpful).

Among other algorithms, `Bonmin` includes an NLP-based branch-and-bound implementation (see [2]). `Bonmin` is primarily aimed at searching for a global optimum of a formulation having a convex relaxation. Our formulation is not convex,

in particular because of the nonlinear Eqs. 9.5, so our use of `Bonmin` is as a heuristic. We utilized several features of `Bonmin` that are designed to facilitate such a use. In fact, though these features have rather general value for nonconvex formulations, these features were implemented in `Bonmin` due to our needs for this project.

4.1 Decreasing Nonconvexity

Because `Bonmin` is really aimed at MINLP formulations having a convex relaxation, it is natural to expect that it will perform better as a heuristic on nonconvex formulations for which the nonconvexity is not too severe. We can soften nonconvexities by making a substitution of variables. Specifically, we can define the cross-sectional area variables:

$$A(e) := \pi(D(e)/2)^2, \quad \text{for all } e \in E.$$

Then, the nonlinear flow bounds (9.2) become linear flow bounds:

$$-v_{max}(e)A(e) \le Q(e) \le v_{max}A(e), \quad \text{for all } e \in E. \tag{9.8}$$

Moreover, the nonlinear head loss Eqs. (9.5) become:

$$H(i) - H(j) = \frac{\text{sgn}(Q(e))|Q(e)|^{1.852} \cdot 10.7 \cdot \text{l}(e) \cdot k(e)^{-1.852} \left(\frac{\pi}{4}\right)^{2.435}}{A(e)^{2.435}},$$
$$\text{for all } e = (i, j) \in E. \tag{9.9}$$

So, we consider the cross-sectional area variables $A(e)$ to satisfy

$$A(e) \in \{\mathcal{A}(e, 1), \mathcal{A}(e, 2), \ldots, \mathcal{A}(e, r_e)\}, \quad \text{for all } e \in E, \tag{9.10}$$

where,

$$\mathcal{A}(e, r) := \pi(\mathcal{D}(e, r)/2)^2, \quad \text{for all } e \in E,\ r = 1, \ldots, r_e.$$

In (9.9), sgn(y) for any number y denotes the sign ($+$ or $-$) of y. Denoting $x = Q(e)$, the term $\text{sgn}(Q(e))|Q(e)|^{1.852}$ in (9.9) can be represented as the function $f(x)$, where $f(x) = x^p$ ($p = 1.852$) when x is nonnegative; and $f(x) = -(f(-x))$ when x is negative. In the next section, we will describe how we approximate $f(x)$ by a smooth function.

4.2 Smoothing

The MINLP code `Bonmin` makes use of the NLP code `Ipopt` (see [7]) to solve continuous relaxations. Like many codes for NLP, it is aimed at smooth formulations.

Unfortunately, the absolute value term in the head loss constraints (9.5) is nondifferentiable (at 0) but not so bad. Thus, to accommodate `Ipopt`, we smoothed the nondifferentiability. In order to do that effectively, our main goal is not to provide a fully accurate approximation near 0 because it is well known that Hazen–Williams Eq. (9.5) is in itself a poor approximation of the real pressure loss for small values of the flow. Instead, we smooth away the mild nondifferentiability by defining the head loss equation in a piecewise manner, so as to have accurate evaluations of the function. We insure the smoothness by matching function values as well as first and second derivatives at the breakpoints. Fortunately, piecewise constraints can be modeled in AMPL (see §18.3 of [5]).

More precisely, let $f(x) = x^p$ ($p = 1.852$) when x is nonnegative, and $f(x) = -f(-x)$ when x is negative (x is standing in for $Q(e)$). This function misbehaves at 0 (the second derivative does not exist there). Choose a small positive δ, and replace f with a function g on $[-\delta, +\delta]$. Outside of the interval, we leave f alone. We will choose g to be of the following form: $g(x) = ax + bx^3 + cx^5$. In this way, we can choose a, b, c (uniquely) so that f and g agree in value, derivative, and second derivative at $x = |\delta|$. So we end up with a smooth-enough antisymmetric function. It agrees in value with f at 0 and outside $[-\delta, +\delta]$. It agrees with f in the first two derivatives outside of $[-\delta, +\delta]$.

Formally, it is easy to prove that:

Proposition 1 *The unique polynomial* $g(x) = ax + bx^3 + cx^5$ *having* $f(x) = g(x)$, $f'(x) = g'(x)$ *and* $f''(x) = g''(x)$ *at* $x = |\delta|$ *is:*

$$
\begin{aligned}
g(x) = &\left(\frac{3\delta^{p-5}}{8} + \frac{1}{8}(p-1)p\delta^{p-5} - \frac{3}{8}p\delta^{p-5} \right) x^5 \\
&+ \left(-\frac{5\delta^{p-3}}{4} - \frac{1}{4}(p-1)p\delta^{p-3} + \frac{5}{4}p\delta^{p-3} \right) x^3 \\
&+ \left(\frac{15\delta^{p-1}}{8} + \frac{1}{8}(p-1)p\delta^{p-1} - \frac{7}{8}p\delta^{p-1} \right) x \, .
\end{aligned}
$$

Proof Via simple calculation one simply has to equate: (i) $g(\delta) = a\delta + b\delta^3 + c\delta^5 = \delta^p = f(\delta)$, (ii) $g'(\delta) = a + 3b\delta^2 + 5c\delta^4 = p\delta^{p-1} = f'(\delta)$, and (iii) $g''(\delta) = 6b\delta + 20c\delta^3 = p(p-1)\delta^{p-2} = f''(\delta)$. This is now a square linear system in the a, b, c variables. We solve it (symbolically), using *Mathematica* (see [9]). Finally, we just observe that f and g are antisymmetric, so we have the same a, b, c for $x = -\delta$. □

Figure 9.4, drawn for $\delta = 0.1$, shows that g provides a good approximation of f. Indeed the quintic curve fits very well on $(-\delta, +\delta)$, and of course it matches up to second order with the true function f at $\pm\delta$. This is all no surprise because we are operating in a small interval of 0, and the function that we approximate is not pathological. The NLP solver `Ipopt` responds well to this technique, as does our MINLP solver `Bonmin`.

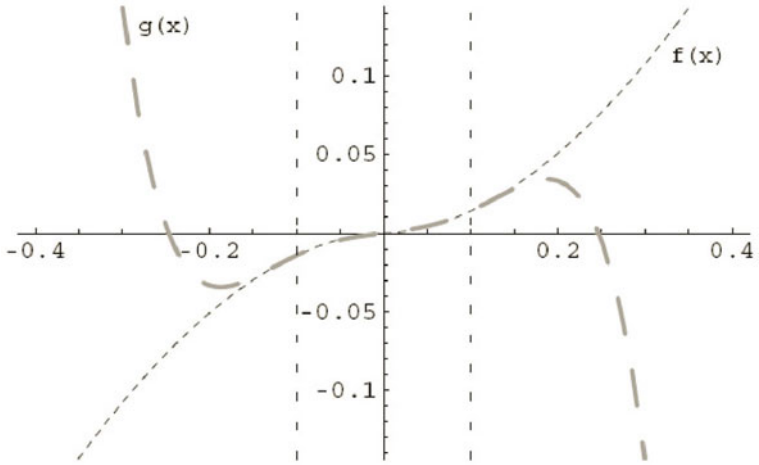


Fig. 9.4 Smoothing f near $x = 0$.

Our experience is that the inaccuracy in using this smoothed function is minimal compared to the other inaccuracies (e.g., numerical and modeling inaccuracies).

4.3 Continuous Cost Function

The NLP solver passes useful branching information to the MINLP solver by working with a good fully-continuous model. Because the cost data is truly discrete, we need to fit a continuous function to the data points. We use and advocate weighted fits to minimize relative error. For example, our least-squares fit for pipe e minimizes

$$\sum_{r=1}^{r_e}\left[1-\left(\frac{\sum_{j=0}^{t_e}\beta(j,e)\left(\mathcal{A}(e,r)^2\right)^j}{\mathcal{C}(e,r)}\right)\right]^2,$$

where, t_e is the desired degree and $\beta(j,e)$ are the coefficients of the polynomial[1] approximating C_e. Note that, for our purposes, a feature was added to `Bonmin` to accommodate one objective function for calculating lower bounds to guide the branch-and-bound search (for us, that is the fitted objective function) and another to calculate actual objective values to improve the incumbent solution (for us this just amounts to using the real objective function determined by the discrete cost data).

[1] [1]Note that the least-square minimization is itself a nonconvex NLP that we solve to local optimality by using the open-source NLP solver `Ipopt` (see, [7]) which we use as an NLP solver throughout our work (see, §4.5).

4.4 SOS

It would be nice if Bonmin could directly handle branching for imposing constraints like (9.10), but at the present time, this is not the case. So, we simply define additional binary variables $X(e,r)$, where $X(e,r) = 1$ indicates that cross-sectional area $\mathcal{A}(e,r)$ is selected for pipe e ($r = 1,\dots,r_e$, $e \in E$). Then, we use the "SOS type-1" branching (see, [1]) that is available in Bonmin. As standard, we use AMPL suffixes to pass along the SOS information needed by the solver: .sosno ("SOS number") is used to group variables into separate SOS constraints and .ref ("reference value") is used to indicate the value symbolized by a variable. In this way, for $e \in E$, in AMPL we naturally set:

$$X(e,r).\text{sosno} := e \text{ , for } r = 1,\dots,r_e \text{ ,}$$

and

$$X(e,r).\text{ref} := \mathcal{A}(e,r) \text{ , for } r = 1,\dots,r_e \text{ .}$$

4.5 Heuristic Features of Bonmin

NLP-based branch-and-bound is only a heuristic for a formulation having a nonconvex continuous relaxation, because relaxations are not solved to global optimality. But Bonmin includes several options tailored to improve the quality of the solutions it provides in such a context.

First, the NLP-solver Ipopt may end up in different local optima when started from different starting points. The two options num_resolve_at_root and num_resolve_at_node allow for solving the root node or each node of the tree, respectively, with a user-specified number of different randomly-chosen starting points, saving the best solution found. Note that the function to generate a random starting point is very naive: it chooses a random point (uniformly) between the bounds provided for the variable.

Secondly, because the solution given by Ipopt does not truly give a lower bound, the user can adjust the fathoming rule to continue branching even if the solution value to the current node is worse than the best-known solution value. This is achieved by setting allowable_gap, allowable_fraction_gap and cutoff_decr to negative values.

5 Example

In this section, we describe a real-world instance as an example. The data are taken from a neighborhood of Bologna called Fossolo. We have 37 junctions (of which 1 reservoir, identified by the index 37), and 58 pipes. The topology of the network is represented in Fig. 9.5 and provided in detail in Table 9.3.

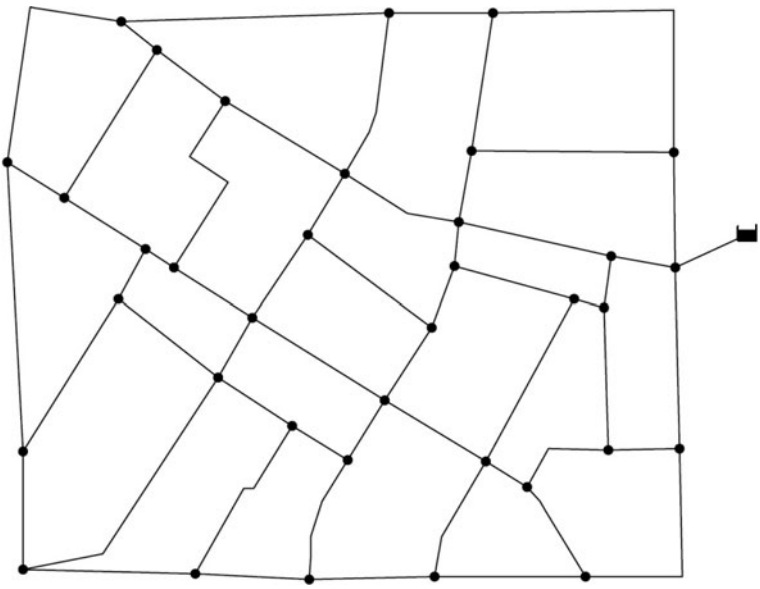


Fig. 9.5 Fossolo network

The hydraulic head at the reservoir is always fixed. In our example it is fixed to 121.0 m. The value of $v_{max}(e)$ is set to 1 m^3/s and the roughness coefficient is set to 100 for all $e \in E$. The lower bound on the pressure head $ph_{min}(i)$ is equal to 40 m and the upper bound $ph_{max}(i)$ is equal to $121 - eval(i)$ meters for all $i \in N$. The values of the elevation and the demand for each junction besides the reservoir are reported in Table 9.1. The network topology and the length of the pipes are reported in Table 9.3.

Concerning the available diameters for each pipe, we have 13 different possibilities. Table 9.2 reports for each $e \in E$ the value of $(\mathcal{D}(e,r), \mathcal{C}(e,r))$ for all $r = 1, \ldots, r_e$.

6 Solution Found by the MINLP Model

The solution found by `Bonmin` is depicted in Fig. 9.6 where the size of each diameter is proportional to the thickness of the arc. The diameters are expressed in meters, and the diameter is equal to 0.06 for the pipes without explicit number, i.e., the minimum diameter permissible for this data set.

The analysis of this solution shows a configuration in which the size of the selected diameters decreases from the reservoir toward the parts of the network farther away from the inlet point. This characteristic of the allocation of diameters to pipes plays

Table 9.1 Elevation (meters) and water demand (m^3/s) for instance Fossolo

i	$elev(i)$	$dem(i)$	i	$elev(i)$	$dem(i)$
1	65.15	0.00049	19	62.90	0.00188
2	64.40	0.00104	20	62.83	0.00093
3	63.35	0.00102	21	62.80	0.00096
4	62.50	0.00081	22	63.90	0.00097
5	61.24	0.00063	23	64.20	0.00086
6	65.40	0.00079	24	67.50	0.00067
7	67.90	0.00026	25	64.40	0.00077
8	66.50	0.00058	26	63.40	0.00169
9	66.00	0.00054	27	63.90	0.00142
10	64.17	0.00111	28	65.65	0.00030
11	63.70	0.00175	29	64.50	0.00062
12	62.64	0.00091	30	64.10	0.00054
13	61.90	0.00116	31	64.40	0.00090
14	62.60	0.00054	32	64.20	0.00103
15	63.50	0.00110	33	64.60	0.00077
16	64.30	0.00121	34	64.70	0.00074
17	65.50	0.00127	35	65.43	0.00116
18	64.10	0.00202	36	65.90	0.00047

Table 9.2 Diameter set (meters) and relative cost per meter (€/m) for instance Fossolo for each pipe

r	$\mathfrak{D}(e,r)$	$\mathfrak{C}(e,r)$
1	0.060	19.8
2	0.080	24.5
3	0.100	27.2
4	0.125	37.0
5	0.150	39.4
6	0.200	54.4
7	0.250	72.9
8	0.300	90.7
9	0.350	119.5
10	0.400	139.1
11	0.450	164.4
12	0.500	186.0
13	0.600	241.3

in favor of a correct hydraulic operation of the network and has a beneficial effect on water quality, see, e.g., the discussion in [10].

This characteristic could be noticed in the solution of MINLP for different instances, see Bragalli et al. [4] for details.

Note that the proposed solution is not guaranteed to be a global optimum of the problem. However, because of the intrinsic difficulty of the problem at hand, the proposed solution is, from a practical viewpoint, a very good quality feasible solution. Other approaches based on heuristic algorithms, mixed integer linear programming, or nonlinear programming models are not effective for medium/large instances. In this application, modeling the problem in the most natural way seems to be the most successful approach.

Table 9.3 Network topology and length of the pipes (meters) for instance Fossolo

$e=(i,j)$	i	j	$l(e)$
1	1	17	132.76
2	17	2	374.68
3	2	3	119.74
4	3	4	312.72
5	4	5	289.09
6	5	6	336.33
7	6	7	135.81
8	7	24	201.26
9	24	8	132.53
10	8	28	144.66
11	28	9	175.72
12	9	36	112.17
13	36	1	210.74
14	1	31	75.41
15	31	10	181.42
16	10	11	146.96
17	11	19	162.69
18	19	12	99.64
19	12	4	52.98
20	2	18	162.97
21	18	10	83.96
22	10	32	49.82
23	32	27	78.5
24	27	16	99.27
25	16	25	82.29
26	25	8	147.49
27	3	11	197.32
28	11	26	83.3
29	26	15	113.8
30	15	22	80.82
31	22	7	340.97
32	5	13	77.39
33	13	14	112.37
34	14	20	37.34
35	20	15	108.85
36	15	16	182.82
37	16	29	136.02
38	29	30	56.7
39	30	9	124.08
40	17	18	234.6
41	12	13	203.83
42	19	20	248.05
43	14	21	65.19
44	21	6	210.09
45	21	22	147.57
46	22	23	103.8
47	24	23	210.95
48	23	25	75.08
49	26	27	180.29
50	28	29	149.05
51	29	33	215.05
52	32	33	144.44
53	33	34	34.74
54	31	34	59.93
55	34	35	165.67
56	30	35	119.97
57	35	36	83.17
58	37	1	1.0

7 Conclusions

More details concerning our methodology and results can be found in [4].

8 A Practical Exercise

In this section, we describe a small but interesting example taken from the literature, in particular it was introduced by Fujiwara and Khang [6]. The network is a simplified version of the water network of the city of Hanoi, Vietnam.

We have 32 junctions (of which 1 reservoir, identified by the index 1), and 34 pipes. The topology of the network is represented in Fig. 9.7.

The hydraulic head at the reservoir is always fixed. In our example it is fixed to 100.0 m. The roughness coefficient is set to 130 and the elevation of each junction is 0. The lower bound on the pressure head $ph_{min}(i)$ is equal to 30 m and the upper bound $ph_{max}(i)$ is equal to 100 m for all $i \in N$. The values of the demand for each

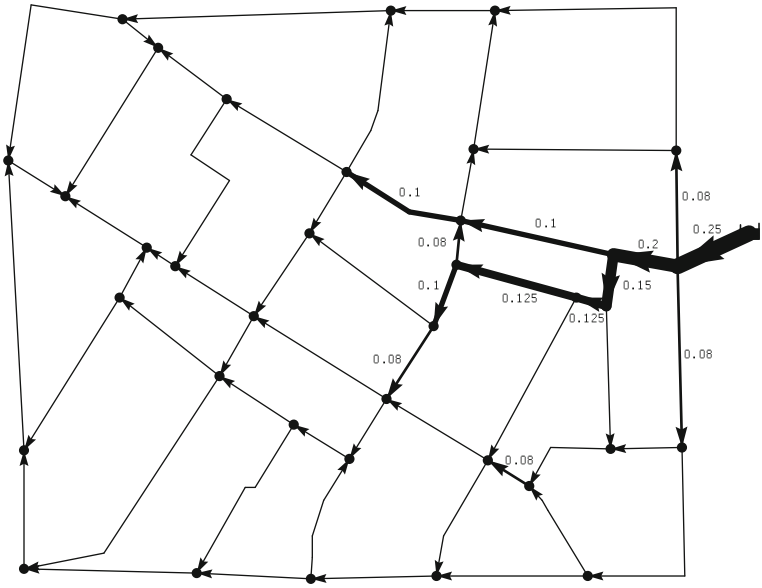


Fig. 9.6 Solution for Fossolo network: the size of each diameter is proportional to the thickness of the arc. The diameter (expressed in meters) is equal to 0.06 m for the pipes without explicit number

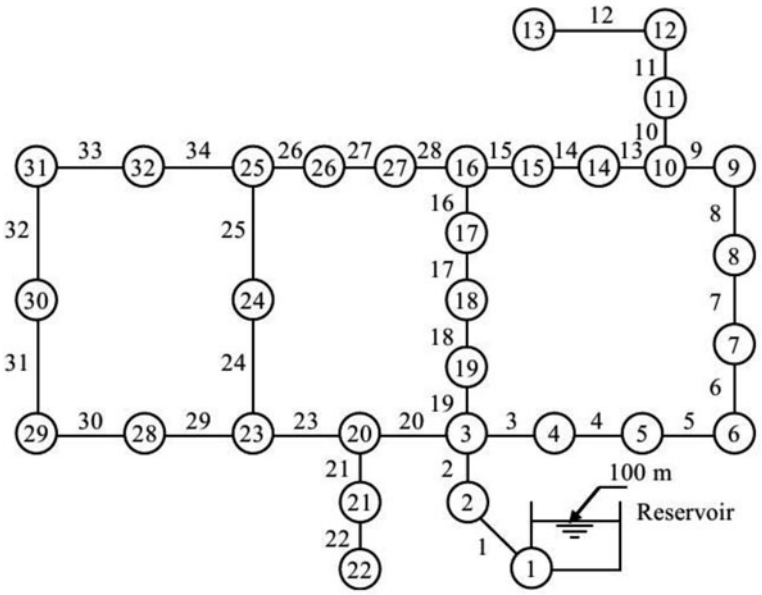


Fig. 9.7 Hanoi network

Table 9.4 Diameter set (meters) and relative cost per meter (€/m) for instance Hanoi for each pipe

r	$\mathfrak{D}(e,r)$	$\mathfrak{C}(e,r)$
1	0.3048	45.73
2	0.4064	70.40
3	0.5080	98.39
4	0.6096	129.33
5	0.7620	180.75
6	1.0160	278.28

Table 9.5 Water demand (m^3/s), topology, pipe length (m), and maximum speed of the water within the pipes (m^3/s) for instance Hanoi

i	$dem(i)$
2	0.24722
3	0.23611
4	0.03611
5	0.20139
6	0.27917
7	0.375
8	0.15278
9	0.14583
10	0.14583
11	0.13889
12	0.15556
13	0.26111
14	0.17083
15	0.07778
16	0.08611
17	0.24028
18	0.37361
19	0.01667
20	0.35417
21	0.25833
22	0.13472
23	0.29028
24	0.22778
25	0.04722
26	0.25000
27	0.10278
28	0.08056
29	0.10000
30	0.10000
31	0.02917
32	0.22361

$e=(i,j)$	i	j	$l(e)$	$v_{max}(e)$
1	1	2	100	7.0
2	2	3	1350	7.0
3	3	4	900	3.0
4	4	5	1150	3.0
5	5	6	1450	2.5
6	6	7	450	2.5
7	7	8	850	2.0
8	8	9	850	2.0
9	9	10	800	2.0
10	10	11	950	2.0
11	11	12	1200	2.0
12	12	13	3500	2.0
13	10	14	800	2.0
14	14	15	500	2.0
15	15	16	550	2.0
16	16	17	2730	2.0
17	17	18	1750	2.0
18	18	19	800	3.5
19	19	3	400	3.5
20	3	20	2200	3.0
21	20	21	1500	2.0
22	21	22	500	2.0
23	20	23	2650	2.0
24	23	24	1230	3.0
25	24	25	1300	2.0
26	25	26	850	2.0
27	26	27	300	2.0
28	27	16	750	2.0
29	23	28	1500	2.0
30	28	29	2000	2.0
31	29	30	1600	2.0
32	30	31	150	2.0
33	31	32	860	2.0
34	32	25	950	2.0

junction besides the reservoir, the network topology, the upper bound on the speed of the water within the pipes, and the length of the pipes are reported in Table 9.5.

Concerning the available diameters for each pipe, we have 6 different possibilities. Table 9.4 reports for each $e \in E$ the value of $(\mathcal{D}(e,r), \mathcal{C}(e,r))$ for all $r = 1, \ldots, r_e$.

References

1. Beale, E. M. L., & Tomlin, J. A. (1970). Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. In J. Lawrence, editor, *Proc. of the 5th Int. Conf. on Operations Research*, pages 447–454
2. Bonami, P., Biegler, L. T., Conn, A. R., Cornuéjols, G., Grossmann, I. E., Laird, C. D., Lee, J., Lodi, A., Margot, F., Sawaya, N., & Wächter, A (2008). An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5:186–204
3. Bonmin. https://projects.coin-or.org/Bonmin, v. 1.0.
4. Bragalli, C., D'Ambrosio, C., Lee, J., Lodi, A., & Toth, P (2012). On the optimal design of water distribution networks: a practical minlp approach. *Optimization and Engineering*, 13:219–246
5. Fourer, R., Gay, D. M., & Kernighan., B. W. (2003). *A Modeling Language for Mathematical Programming*. Duxbury Press/Brooks/Cole Publishing Co., second edition
6. Fujiwara, O., & Khang, D. B. (1990). A two-phase decomposition method for optimal design of looped water distribution networks. Water Resources Research, 26: 539–549
7. Ipopt. https://projects.coin-or.org/Ipopt, v. 3.5.
8. Lee, J., & Leyffer, S. (2012). *Mixed Integer Nonlinear Programming*. The IMA Volumes in Mathematics and its Applications. Springer
9. Mathematica. www.wolfram.com/products/mathematica/index.html, v. 7.0.
10. Van Den Boomen, M., Van Mazijk, A., & Beuken, R. H. S. (2004). First evaluation of new design concepts for self-cleaning distribution networks. *Journal of Water Supply: Research and Technology AQUA, 53*(1), 43–50

Chapter 10
Wood Inventory Management in Paper Industry

G. S. R. Murthy, A. L. N. Murthy and Katta G. Murty

1 Brief History of Paper Making

Paper making is considered one of the four great inventions of China, and the paper making process is traced to Cai Lun, a court eunuch during the Hun Dynasty in 105 CE (Current Era). However, paper was being used for wrapping and padding in China since second century BCE (Before Current Era), as toilet paper since sixth century CE (an Arab traveler to China in 851 CE wrote about the strange practice of the Chinese use of toilet paper " ... the Chinese do not wash themselves with water when they have done their morning necessities, but only wipe themselves with paper!"), and for government issued paper currency by around 1000 CE.

Thick paper-like material produced from the pith of the *Cyperus Papyrus* plant by lamination, called papyrus, was used even much earlier in Ancient Egypt.

The Chinese process for making paper in those days consisted of soaking and pounding rags (recycled fibres from used textiles (of hemp, linen, or cotton)) and some plant fibres into a watery pulp, then pouring this pulp onto a woven bamboo screen. The layer of pulp that formed on the screen was then lifted off carefully, and as it dried, it matted into a sheet of paper.

Use of paper spread slowly outside China to other parts of East Asia. In those days, it was thin and translucent, hence written only on one side. The paper making technology traveled from China to Japan with Buddhist priests around 610 CE, and the Japanese started making paper themselves using fibres of mulberry trees called bast.

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_10) contains supplementary material, which is available to authorized users.

G. S. R. Murthy (✉) · A. L. N. Murthy · K. G. Murty
SQC & OR Unit, Indian Statistical Institute, Street No.8, Habsiguda,
500 007 Hyderabad, India
e-mail: murthygsr@gmail.com

A. L. N. Murthy
e-mail: simhaaln@rediffmail.com

K. G. Murty
e-mail: murty@umich.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_10, © Springer Science+Business Media New York 2015

With the defeat of China in the battle of Talas in Kyrgyzstan in 751 CE by the Arabs, the secret of paper making was obtained from Chinese prisoners of war by the Islamic world, and they set up a paper mill in Samarkand. Also they refined the manufacturing process, and made paper making into a major industry by setting up plants in Baghdad, Damascus and other places; and introduced books, bookbinding, and bookshops. They started wrapping goods sold to customers in markets with paper by twelfth century CE.

Soon paper making was diffused across all of the Islamic world and Europe. Paper manufacturing reached India in thirteenth century CE through Arab merchants, and it quickly replaced traditional writing materials there.

Archeological evidence indicates that in the USA, similar bark-paper writing material (parchment made by boiling and pounding the inner bark of trees until the material becomes suitable for art and writing) was in use among Mayas and other Mesoamerican cultures earlier than fifth century CE.

In Europe, paper making started in the then Islamic part of the Iberian peninsula in tenth century CE, and slowly spread to Italy, France, and Germany by 1400 CE, and was mechanized by the use of waterpower. Its rapid expansion was truly enhanced by the invention of the printing press and the beginning of the printing revolution in fifteenth century CE. However, due to their noise and smell, paper mills were allowed to be erected only outside the city boundaries in those days.

Until mid-nineteenth century CE, the most common fibre source for paper production was recycled fibre from rags. A process of removing printing inks from recycled paper (called de-inking now-a-days) was developed at the end of the eighteenth century. Also around that time, continuous paper making machines were introduced. Observing American wasps chewing up wood fiber and making their nests from it, people got the idea that pulp from wood may make an ideal raw material to make paper. Soon experiments began for using wood as a fibre source for paper making.

By mid-nineteenth century CE, a machine for extracting fibres from wood by pulping it (instead of rags) was invented; and the technique for bleaching the pulp (so that the paper produced will be white) was developed. This started a new era in paper making. By the end of the nineteenth century, all paper making machines were using wood instead of rags to make paper.

By 1900 CE, with the invention of the practical fountain pen, and mass-produced pencil, and the introduction of the rotary printing press, wood-based paper and the newspapers, schoolbooks, fiction books, magazines, etc. printed on paper caused a major transformation of the human society.

For more details see articles in references [1, 2, 3, 4].

2 Brief Description of the Paper Manufacturing Process

We will describe the paper manufacturing process using wood as the basic raw material (see Fig. 10.1). Wood of conifers is generally preferred because it consists of longer fibres than the wood of deciduous trees (like eucalyptus, subabul, beech,

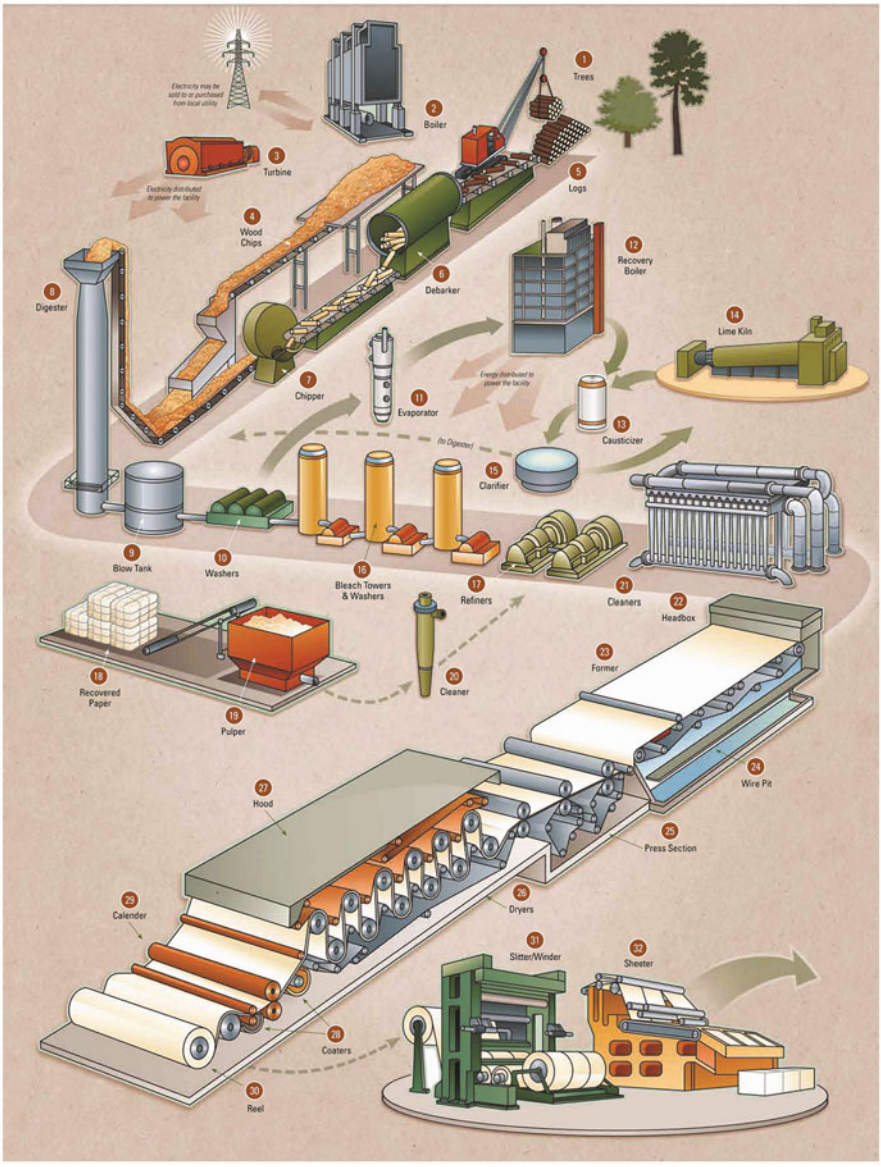


Fig. 10.1 From Forest to Finished Products: The Paper Making Process (Details at: http://www.internationalpaper.com/documents/EN/Sustainability/PaperProcess.pdf.)

etc.), as longer fibres form a firmer fibrous web leading to firmer paper on the paper machine. Most paper manufacturing companies in India, including the large ones, use debarked logs of eucalyptus, subabul, casurina trees as these are locally grown abundantly. The company where this project was carried out uses 1,300,000 t of wood

logs of eucalyptus, subabul, casurina (the percentage of these species in the mix is approximately 60, 30, 10 %, respectively) annually; and some imported wood pulp, to manufacture *writing and printing paper*, and *paperboard* (used for manufacturing packing boxes for packing a variety of consumer goods like toothpaste tubes etc.). About 70 % of their wood pulp requirement is met by pulp made at the company with the wood supply mentioned above, and 30 % met by imported wood pulp in the form of dry sheets. Paperboard is manufactured on the paper machine by gluing together three sheets of paper made simultaneously.

The first major operation in paper making is the preparation of pulp from which paper is made. Paper mills can make the pulp from wood themselves (these mills are called *integrated mills*), or they can buy pulp in the form of dry pressed sheets (these sheets are broken down into pulp with water as they enter the manufacturing process).

Wood consists of mostly cellulose, lignin (dark colored chemical compounds that hold the cellulose fibres in wood together), and minor amounts of resins, tanins, and mineral material. There are two basic ways of pulping wood logs. In *mechanical pulping*, wood logs are processed into pulp by grinding them against a quickly rotating stone disc with water, resulting in mechanical pulp (MP); this process yields 95 % pulp, but it leaves all the lignin in the pulp. The mechanical pulping process damages the fibres in wood, with the result that paper made from it is not strong; and since all the lignin is left in MP, paper made from it becomes yellow very quickly.

In *chemical pulping*, logs are processed into wood chips of about 3" size in the *chipping process* at the beginning of paper making. Wood chips of various species of wood in the raw material mix are mixed together, and this mixture then enters the *chemical cooking process*. Here, the wood chips are cooked in a chemical solution which converts lignin into a water-soluble substance that is then washed out, leaving a pulp that consists of mostly cellulose fibre. So the yield of chemical pulping is approximately 50 % of the input wood, but the fibres in it are clean and undamaged. Spent fluids from chemical cooking are processed to recover the chemicals for reuse.

The pulp at this stage is brown in color, it goes to the *bleaching process* next. Bleaching is a continuation of the chemical cooking process, it also removes the remaining lignin in the pulp. Bleaching is a complex process consisting of several chemical process steps with washing of the pulp between the various treatments.

Then, the pulp enters the *refining process*, where it goes through a fine blending of the fibre ends for a close-knit connection between individual fibres (for increasing the strength of paper produced), addition of water, fillers, sizing materials, dyes, and additives (like kaolin, china clay, chalk, titanium dioxide etc. to make paper more opaque, softer, and flexible). Then, the watery pulp enters the paper machine.

Once pulp has been prepared, the next operation in making paper for writing and printing, is forming it into a sheet. This transforms the diluted pulp into a fine uniform laminate on the *paper machine* consisting of a *head box*, and a *forming machine*. Pulp arrives at the head box through round tubing, but it transfers it onto the paper forming wire mesh in the form of a thin wide and uniform wet sheet. Pulp exits the head box through a series of equispaced nozzles spread across the width-line of the paper sheet to be formed; pulp comes out of each nozzle as a jet

spray onto a continuously moving paper forming wire mesh (normally made of either polyethylene or polyurethane). The paper sheet is consolidated during this journey; water passing freely through the wire mesh drops down by gravity from it first, and is sucked from it by vacuum suction towards the end of the forming section.

On leaving the forming section, the paper sheet has 20 % consistency (i.e., it is 80 % water), and enters the further drying process called *wet pressing*. In this process, the paper sheet is passed between rolls in contact with felt which turns around the rolls of the presses absorbing water from the sheet.

When the paper sheet comes out of wet pressing, its water content is about 60 %. Next it enters the *drying operation*, in which more water is eliminated from it through the application of heat by passing through heated air, through contact over huge steam-heated cylinders, and through infrared drying. At the end of this operation, the water content of the paper sheet is reduced to its target of approximately 5 %. This drying is one of the costliest operations in paper manufacturing.

Then, the paper sheet enters the *coating operation*, where a starch based emulsion consisting of special coatings is applied to the paper sheet in two stages called as pre-coat and a final-coat, to give it smoothness and shine necessary so that inks can easily be applied on it for writing, and printing. The composition of the coating emulsion is tailored to the specific properties required for the final sheet of paper, such as water resistance, gloss, opacity, smoothness, and whiteness.

Some papers that require high surface finish, now pass through an operation known as *calendering* whose main purpose is to improve the gloss and printing properties of paper.

After the calendering operation, the paper sheet is rolled up in the form of reels known as *jumbo rolls*. Customers require paper either in the form of reels of specified sizes, or in the form of sheets of standard sizes. Depending on these requirements, in the *finishing section*, jumbo rolls are either rewound into reels as required by the customer, or cut into sheets and packed into packets for delivery to customers.

2.1 Process for Making Paper Board

Above we discussed the process of making writing and printing paper on a paper machine in which paper is manufactured as a single sheet. The process for making paper board is the same as above until the paper machine stage. The paper machine for making paper board consists of three head boxes arranged vertically one below the other, each with its own separate forming machine and forming wire mesh, forming simultaneously three sheets of paper one below the other vertically. Each of these paper sheets goes through the same operations as above simultaneously until the calendering operation, after which the three sheets are glued together to form a single sheet of paper board, which is rolled up at the end to form a jumbo roll of paperboard.

3 Wood Inventory Management in a Paper Mill

In this chapter, we will discuss wood inventory management and optimization, based on a consulting project carried out by the Indian Statistical Institute at a leading paper manufacturing company in India. The first process at the company is the chipping process, and the mixture of wood of different species fed into the chippers is called *wood mixture*, and the proportions of different species in the wood mixture is referred to as *furnish*. The Management develops furnish specifications in each planning horizon based upon the yield forecasts of the various species in that year, cost/ton of each species, and other considerations.

When furnish is specified, we will refer to the specified proportions of various wood species in the wood mixture as the *furnish constraints*. Out of different species, subabul and eucalyptus account for about 90 % of the wood mixture. At present rates, the cost of these two woods in the annual consumption approximately accounts for about 10 % of the total annual turnover of the company.

Wood quantities are measured in tons. The quantity of wood that the company can procure daily depends on the availabilities of wood, and trucks to haul it; both of which tend to be low during the summer months of March to August. This seasonal variation in the supply of wood over the year is the reason for the company maintaining an inventory of wood in a storage yard to avoid the possibilities of shortages and stock outs.

Wood is stored in stacks which are maintained species-wise in the storage yard. Stored wood loses moisture over time, and is also subject to degradation by insect damage, hence its weight decreases. For these reasons, the company forms stacks to facilitate withdrawal by the first in first out (FIFO) principle, which it follows as far as possible; and the company imposes an upper bound on the shelf-life for wood in storage, that varies by species between 3 and 6 months.

Each day procured wood logs arrive at the company in trucks. Some of these trucks are sent to the production facility as they arrive, to unload the wood in them directly into the chippers. Wood fed into the chippers like this is called *direct feed*. The remaining incoming trucks are sent to the wood storage yard for storing the wood in them. Each day, besides the direct feed, some wood is also drawn from the storage yard to feed the chippers, stored wood fed like this is referred to as *indirect feed*. Thus, each day the feed into the chippers consists of direct and indirect feeds.

The *planning horizon* is a financial year, April 1st to March 31st. The production manager of the pulp mill specifies the daily requirements of wood for the entire year. From this and the specified furnish, the daily requirements are aggregated into monthly requirements of wood for each wood species. The company uses month as the *planning unit*. The raw materials manager uses these monthly requirements for wood, and the proportion of direct and indirect feed patterns, to work out a month-wise procurement plan for the year, and accordingly fixes deals with suppliers for wood.

3.1 Decision Making Problem Considered

The decision making problem that the management faces is that of deciding on (1) how much wood (to be specified species wise) to procure each month, and (2) how much of it to be used as direct feed (again to be specified species wise) so that various constraints are satisfied and the overall cost is minimized.

Since subabul and eucalyptus constitute 90 % of input wood mixture, we will confine the discussion in this chapter to just these two species only for simplicity. Other species in the wood mixture can be handled in a similar way. The focus of this chapter is to determine an optimum plan for procuring wood monthly, feeding it into the chippers, and storing it in the storage yard. The mathematical model for this decision making problem is given in detail in the next section. Section 5 presents the results of application of the model to sample data, and Sect. 6 will summarize the chapter.

4 Optimization Model for Wood Inventory Management

For the decision making problem in question, we shall consider two years—the current year and the previous year. Thus, we have 24 months in all. These are denoted by the set $M = \{1,\ 2,\ \ldots,\ 24\}$. Here, 1 represents April of previous year, 2 represents May of the previous year, and so on.

The planning can be done (revised) at the beginning of any month of the current year comprising months represented by 13 to 24. Let j_0 denote the first month of the planning period. That is, we are planning for the period month j_0 of the current year to rest of the months in the current year.

The decisions involved are how much of wood from each species to be purchased in each month and how much of it to be fed directly into the production as direct feed. Since the company follows FIFO policy, the decision on the indirect feed from the yard depends only on the furnish. If FIFO is to be built into the model, then the model becomes nonlinear and complex. Instead we can relax this constraint in the model but the solution obtained will make it feasible to implement FIFO in the practical execution. This aspect will be clear from the model we are building.

In our model, we define our decision variables as the quantities to be distributed from each month's stock/purchases to the production months without violating the shelf life constraints. Let L_s and L_e denote the shelf lives (in months) of subabul and eucalyptus, respectively. This means that subabul procured in month i can be used up to month $i + L_s$ but not in month $i + L_s + 1$ or later. Similarly eucalyptus procured in month i can be used up to month $i + L_e$ but not in month $i + L_e + 1$ or later. Let $u = \max(1, j_0 - L_s)$ and $v = \max(1, j_0 - L_e)$. Let $t = \min(u,\ v)$. Thus, the months to be considered for the model are $t, t+1, \ldots, 24$. Define the sets $P = \{j_0,\ j_0 + 1,\ \ldots,\ 24\}$, $S = \{(i, j) : j \geq i \geq u,\ j \in P \text{ and } j - i \leq L_s\}$ and $E = \{(i, j) : j \geq i \geq v,\ j \in P \text{ and } j - i \leq L_e\}$.

The decision variables are defined in the following subsection.

4.1 Decision Variables

The decision variables are the quantities of wood distributed from each month's stock/availabilities to future months.

1. For each $(i, j) \in S$ define s_{ij} = quantity of subabul taken from month i and to be used for production of month j.
2. For each $(i, j) \in E$, e_{ij} = quantity of eucalyptus taken from month i and to be used for production of month j.
3. It is possible that we may end up in an infeasible situation. The infeasibility occurs when there is a shortage of wood (subabul and eucalyptus). For this reason, we need to define the artificial variables only for the production months, of course, two sets of variables—one for subabul and the other for eucalyptus. Define
   (a) x_j = quantity of subabul that will be made available at the beginning of month $j \in P$ for month j's production/opening inventory requirement,
   (b) y_j = quantity of eucalyptus that will be made available at the beginning of month $j \in P$ for month j's production/opening inventory requirement,

4.2 Symbols Used for Model Parameters (Data Elements)

We shall now describe the model parameters.

1. *Availabilities*.
   (a) a_{si} = quantity of subabul available from month i, $u \leq i \leq 24$. For $i < j_0$, a_{si} is a known quantity; and for $i \geq j_0$, a_{si} is a forecasted availability.
   (b) a_{ei} = quantity of eucalyptus available from month i, $v \leq i \leq 24$. For $i < j_0$, a_{ei} is a known quantity; and for $i \geq j_0$, a_{ei} is a forecasted availability.
2. *Material Costs*.
   (a) c_{si} = cost of subabul/ton purchased in month i, $i \in M$.
   (b) c_{ei} = cost of eucalyptus/ton purchased in month i, $i \in M$.
3. *Handling Costs*. There are two types of handling—direct and indirect. Indirect handling cost is much higher than the direct handling cost. Since the wood loses its weight over time, the weight of wood also varies over time and hence the calculation of cost of wood handled indirectly (that is first storing it in yard and then feeding it to production later) also varies. For simplicity, it has been agreed to take a reasonable rate, cost per ton, as indirect cost that will be proportional to the weight at the time of storage. The present rates per ton of wood are Rs. 10 for direct handling and Rs. 110 for indirect handling. Let
   (a) c_d = cost per ton for direct handling.
   (b) c_h = cost per ton for indirect handling.
4. *Moisture Losses*. Moisture losses are specified separately for subabul and eucalyptus. It is assumed that one ton of subabul (eucalyptus) purchased in month i would weigh l^s_{ij} (l^e_{ij}) tons in month j.

5. *Opening Inventories*. For each production month j, there is an opening inventory requirement. This is specified for subabul and eucalyptus separately. Accordingly, there must be at least $q_{sj}(q_{ej})$ tons of subabul (eucalyptus) at the beginning of month j, $j \in P$.
6. *Direct Feed Percentages*. The direct feed refers to wood that is used directly from the purchases, and the indirect feed is the wood fed from the stocks. One of the requirements is that a certain percentage of the wood (by current weight in tons) fed to production should be the direct feed. For production month j, the direct feed percentage should be at least d_j.
7. *Production Requirement*. At the time of planning, the management will specify the total quantity of wood (in our case, the total quantity of subabul and eucalyptus put together) required for each of the production months in the planning horizon. We shall denote this by r_j tons for the jth month, $j \in P$.
8. *Furnish Limits*. If r_j is the production requirement for the month j, then the percentage of subabul in that is called the furnish percentage. Two limits are specified for each month—f_j and F_j. The subabul percentage in r_j should lie between f_j and F_j.

4.3 Constraints

There are a number of constraints in this optimization model. These are listed below.

1. *Availability Constraints*. As noted earlier a_{si}s and a_{ei}s denote stocks/forecasts of subabul and eucalyptus available from month i. Consider month i; the total quantity of subabul drawn from this month is given by $\sum_{j:(i,j)\in S} s_{ij}$, $i \geq u$, and this should be at most a_{si}. A similar expression holds for eucalyptus. Therefore, the availability constraints are:

$$\sum_{j:(i,j)\in S} s_{ij} \leq a_{si}, \text{ for } i \geq u, \text{ and } \sum_{j:(i,j)\in E} e_{ij} \leq a_{ei}, \text{ for } i \geq v. \tag{10.1}$$

2. *Shelf Life Constraints*. The shelf life constraints have been built in the model by defining the decision variables (s_{ij}s and e_{ij}s) appropriately.
3. *Production Requirement Constraints*. Consider month $j \in P$ and let r_j be the quantity of wood required for this month. The quantities of subabul and eucalyptus supplied to month j from month i are s_{ij} and e_{ij}. Due to moisture losses, s_{ij} would weight $s_{ij}l^s_{ij}$ and e_{ij} would weight $e_{ij}l^e_{ij}$. Taking the artificial variables into consideration, the production constraints are

$$x_j + y_j + s_{ij}l^s_{ij} + e_{ij}l^e_{ij} \geq r_j, \text{for all } j \in P. \tag{10.2}$$

4. *Opening Inventory Constraints*. Consider a month $j \in P$. Let us compute the opening inventory of subabul for this month. Let i be such that $i < j_0$ and $(i, j) \in S$. Note that $\sum_{k=j_0}^{j-1} s_{ik}$ is the quantity of subabul drawn for the months

$j_0,\ j_0+1,\ \ldots,\ j-1$. Therefore, the balance $a_{si}-\sum_{k=j_0}^{j-1} s_{ik}$ will remain at the beginning of month j from the stocks of month i, and its weight at the beginning of month j is $(a_{si}-\sum_{k=j_0}^{j-1} s_{ik})\times l_{ij}^{s}$. Note that for (i,j) to be in S, i must be at least $j-L_s$. Thus, $\sum_{i=j-L_s}^{j_0-1}((a_{si}-\sum_{k=j_0}^{j-1} s_{ik})\times l_{ij}^{s})$ is the amount of inventory of subabul from months $u, u+1, \ldots, j_0-1$. Now consider a month i such that $i \geq j_0$ and $(i,j)\in S$. Note that $l_{ij}^{s}\sum_{k=j}^{i+L_s} s_{ik}$ is the weight of subabul purchased in month i and available at the beginning of month j. Thus, $\sum_{i=j_0}^{j-1} l_{ij}^{s}\sum_{k=j}^{i+L_s} s_{ik}$ is the total quantity of subabul that is available as the opening inventory from the purchases planned for the months $j_0,\ j_0+1,\ \ldots,\ j-1$. Therefore, the total inventory of subabul at the beginning of month $j\in P$ is given by

$$I_s(j)=x_j+\sum_{i=j-L_s}^{j_0-1} l_{ij}^{s}\left(a_{si}-\sum_{k=j_0}^{j-1} s_{ik}\right)+\sum_{i=j_0}^{j-1} l_{ij}^{s}\sum_{k=j}^{i+L_s} s_{ik}.$$

Similarly, the opening inventory of eucalyptus for month j is given by

$$I_e(j)=y_j+\sum_{i=j-L_e}^{j_0-1} l_{ij}^{e}\left(a_{ei}-\sum_{k=j_0}^{j-1} e_{ik}\right)+\sum_{i=j_0}^{j-1} l_{ij}^{e}\sum_{k=j}^{i+L_e} e_{ik}.$$

Recall that q_{sj} and q_{ej} are the opening inventory requirements of subabul and eucalyptus, respectively for the month $j\in P$. Therefore, we have the following opening inventory constraints:

$$x_j+\sum_{i=j-L_s}^{j_0-1} l_{ij}^{s}\left(a_{si}-\sum_{k=j_0}^{j-1} s_{ik}\right)+\sum_{i=j_0}^{j-1} l_{ij}^{s}\sum_{k=j}^{i+L_s} s_{ik}\geq q_{sj}\quad \text{for } j\in P, \tag{10.3}$$

$$y_j+\sum_{i=j-L_e}^{j_0-1} l_{ij}^{e}\left(a_{ei}-\sum_{k=j_0}^{j-1} e_{ik}\right)+\sum_{i=j_0}^{j-1} l_{ij}^{e}\sum_{k=j}^{i+L_e} e_{ik}\geq q_{ej}\quad \text{for } j\in P. \tag{10.4}$$

5. *Direct Feed Constraints*. For $j\in P$, recall that d_j denotes the percentage of direct feed. As s_{jj} and e_{jj} denote the direct feeds of subabul and eucalyptus, respectively for month j, we must have

$$s_{jj}+e_{jj}\geq d_j r_j/100,\quad j\in P. \tag{10.5}$$

6. *Furnish Constraints*. For $j\in P$, recall that f_j and F_j denote the furnish limits of subabul in month j with production requirement r_j. Then, the furnish constraint states that the subabul content in r_j must be at least $f_j r_j/100$ and at most $F_j r_j/100$. Therefore, we have

$$\sum_{i:(i,j)\in S} l_{ij}^{s}s_{ij}\geq f_j r_j/100 \text{ and } \sum_{i:(i,j)\in S} l_{ij}^{s}s_{ij}\leq F_j r_j/100. \tag{10.6}$$

4.4 Objective Function

The objective function comprises three components, namely, the material cost, the cost of handling and the cost of inventory. Since we are solving the problem with artificial variables, the artificial variables (the shortages) are priced very high. For this reason, we can omit the inventory and handling costs on these variables (in a sense, these are included in the cost itself).

1. *Material Cost*. The cost of the material is associated with the old stocks (a_{si} and a_{ei} for $i < j_0$) and new purchases (s_{ij}s, e_{ij}s for $i \geq j_0$). Note that if a part of the old stock is not used, then it is rational to add the cost of this unused material to the objective function. Therefore, the cost of material can be taken as the cost of used material plus the cost of unused material. From the definition of the decision variables, the unused material arises only from the old stocks. Cost of used material is given by $\sum c_{si}s_{ij} + \sum c_{ei}e_{ij}$. Cost of unused material is given by $\sum_{i<j_0}(a_{si} - s_{ij})c_{si}$ for subabul and $\sum_{i<j_0}(a_{ei} - e_{ij})c_{ei}$ for eucalyptus. But $\sum c_{si}s_{ij} + \sum_{i<j_0}(a_{si} - s_{ij})c_{si} = \sum_{i<j_0} a_{si}c_{si} + \sum_{(i\geq j_0} c_{si}s_{ij}$ and $\sum c_{ei}e_{ij} + \sum_{i<j_0}(a_{ei} - e_{ij})c = \sum_{i<j_0} a_{ei}c_{ei} + \sum_{i\geq j_0} c_{ei}e_{ij}$. As $\sum_{i<j_0} a_{si}c_{si}$ and $\sum_{i<j_0} a_{ei}c_{ei}$ are constants, these two terms can be dropped from the objective function. Take θ, a large positive number, as the cost of artificial variables x_j and y_j. Then, the material cost component of the objective function is given by

$$C_M = \sum_{i\geq j_0} c_{si}s_{ij} + \sum_{i\geq j_0} c_{ei}e_{ij} + \theta \sum_{j\geq j_0}(x_j + y_j). \tag{10.7}$$

2. *Cost of Handling*. The total quantity of wood handled is given by $\sum(s_{ij} + e_{ij})$ of which $\sum(s_{jj}+e_{jj})$ is the quantity of direct handling. Again we are not considering the artificial variables which is unnecessary. Therefore, handling cost is given by

$$C_H = c_h \sum(s_{ij} + e_{ij}) + (c_d - c_h)\sum(s_{jj} + e_{jj}). \tag{10.8}$$

3. *Cost of Inventory.* We take the cost of holding as the interest on cost of wood held in inventory. Let r denote the rate of interest per annum. Subabul purchased at a cost c_{si} per ton in month i and used in month j is held in inventory for $j - i$ months. Therefore, the interest on this is given by $s_{ij} \times c_{si} \times (j - i) \times \frac{r}{1200}$. But the quantity of ith month, $i < j_0$, not used in any of the production months is held in stock for $(24 - i)$ months. The holding cost of this is equal to $c_{si}\left(a_{si} - \sum_{k=j_0}^{24} s_{ij}\right)(24 - i)r/1200$. Therefore, the total holding cost is given by

$$H_s = \frac{r}{1200}\left\{\sum_{(i,j)\in S} s_{ij} \times c_{si} \times (j - i) \times + \sum_{i=u}^{j_0-1} c_{si}\left(a_{si} - \sum_{k=j_0}^{24} s_{ij}\right)(24 - i)\right\}. \tag{10.9}$$

Similarly, the holding cost of eucalyptus is given by

$$H_e = \frac{r}{1200}\left\{\sum_{(i,j)\in E} e_{ij} \times c_{ei} \times (j-i) \times + \sum_{i=v}^{j_0-1} c_{ei}\left(a_{ei} - \sum_{k=j_0}^{24} e_{ij}\right)(24-i)\right\}. \tag{10.10}$$

Therefore, the total cost of inventory is given by $C_I = H_s + H_e$.

4.5 Complete Mathematical Formulation

The complete mathematical formulation for the decision making problem is given by: Minimize $C_M + C_H + H_s + H_e$ subject to constraints (1) to (6) mentioned above.

5 Application to Sample Data

We shall now apply the optimization model to a sample data from the company. A decision support system (DSS), a software package, was developed for the company which takes all the inputs for solving the problem and produces the required outputs. With the help of this software, optimal solutions can be obtained for the dynamic situations, that is, at the beginning of any month of the planning horizon. With the help of this DSS, it was possible to study the effectiveness of using the optimization model, by applying it to the past data at the end of a planning year (that is, end of month 24) so that the actual decisions made (based on the subjective decisions made by the management with their experience and expertise) as well as the actual availabilities and the forecasts were known at the beginning of each month in the past.

In this section, we shall present the typical scenarios of analysis that need to be performed on the data. One of them is when you are planning at the beginning of the planning horizon (Sect. 5.2) and the other when you are at the beginning of the second month of the planning horizon (Sect. 5.3). The second scenario talks about dynamic optimization. At the beginning of the second month, we know what has been planned and what has actually happened. This will give us an opportunity to examine the difference between the plan and the actual scope for revising the plan using the model.

5.1 Inputs

The inputs for the decision making problem are: (1) moisture losses, (2) old stocks and forecasted availabilities, (3) cost elements (material costs, handling costs, rate of interest), (4) shelf lives, (5) production requirements, and (6) opening inventory requirements.

1. *Moisture Losses*. Recall that $l^s_{ij}(l^e_{ij})$ is the residual weight of subabul (eucalyptus) (that is one ton of subabul (eucalyptus) purchased in month i will weigh l^s_{ij} tons in month j). Based on the past data the values of l^s_{ij}s and l^e_{ij}s are estimated and they are given below.

$$l^s_{ij} = \begin{cases} 0.68 & \text{if } j \geq i+3 \\ 0.75 & \text{if } j = i+2 \\ 0.79 & \text{if } j = i+1 \\ 1.00 & \text{if } j = i \end{cases} \tag{10.11}$$

and

$$l^e_{ij} = \begin{cases} 0.63 & \text{if } j \geq i+2 \\ 0.68 & \text{if } j = i+1 \\ 1.00 & \text{if } j = i \end{cases} \tag{10.12}$$

2. *Handling Costs, Shelf Lives and Rate of Interest*. The cost of direct handling per ton is Rs. 10 and that of indirect handling is Rs. 105. Shelf life of subabul is 4 months and that of eucalyptus is 6 months. The holding cost is taken as the interest on inventory, and the rate of interest is taken as $r = 15\,\%$ per annum.
3. *Direct Feed and Furnish Percentages*. Direct feed percentage is taken as 25 % for all months, that is, $d_j = 25$ for $j = 13, 14, \ldots, 24$. The furnish limits are also taken uniformly throughout the year, that is, $f_j = 40$ and $F_j = 65$ for $j = 13, 14, \ldots, 24$.
4. *Rest of the Inputs*. The other inputs, the availabilities and forecasts, the production requirements, the opening inventory requirement are given in Table 10.1.

In Table 10.1, data on a_{si}, a_{ei}, c_{si}, c_{ei} for $i \leq 12$ are the actual and known figures, where as the same for $i \geq 13$ are forecasts. The production requirements r_i are planned figures and the actual realizations may be at variance due to various factors. The opening inventories are also planned but usually less prone to changes.

5.2 Planning at the Start

In this section, we shall find the optimal decisions when we are at the beginning of the planning horizon, that is, at the beginning of month 13. In this case, our $j_0 = 13$ (see first paragraph of Sect. 4). It may be noted that the old stocks from yard are given in Table 10.1 from the months October and November of the old year (months 7 and 8) for subabul; as the shelf life is only 4 months, the availabilities of subabul of these two months will be ignored in the problem. The optimal plan is obtained using the model of Sect. 4 and the input data as given in the previous section with $j_0 = 13$. The solution is presented in Table 10.2. The overall cost, the objective value, is Rs. 1,208,985,562. This solution above is obtained with the constraint that shelf lives of subabul and eucalyptus as 4 and 6 months, respectively. It may be noted that the solution does not make use of the wood whose shelf life has expired.

Table 10.1 Inputs for a_{si}, a_{ei}, c_{si}, c_{ei}, q_{si}, q_{ei}, and r_i

	Subabul			Eucalyptus			
Month (i)	a_{si}	c_{si}	q_{si}	a_{ei}	c_{ei}	q_{ei}	r_i
Oct (7)	22,000	2222	–	6000	2300	–	–
Nov (8)	23,000	2122	–	17,000	2100	–	–
Dec (9)	16,700	2322	–	14,500	1900	–	–
Jan (10)	26,745	2000	–	26,254	2300	–	–
Feb (11)	15,466	2000	–	14,620	2300	–	–
Mar (12)	16,964	2000	–	10,456	2300	–	–
Apr (13)	17,283	2100	18,000	11,738	2478	18,000	33,100
May (14)	16,260	2200	18,000	12,478	2455	18,000	33,201
Jun (15)	15,772	2150	18,000	11,249	2485	18,000	33,071
Jul (16)	14,386	2300	18,000	11,784	2475	18,000	30,350
Aug (17)	11,793	2300	18,000	7086	2488	18,000	29,663
Sep (18)	9549	2300	18,000	8374	2389	18,000	29,868
Oct (19)	15,720	2300	18,000	15,614	2423	18,000	30,794
Nov (20)	18,615	2300	18,000	23,985	2395	18,000	27,447
Dec (21)	32,338	2100	18,000	30,515	2475	18,000	38,257
Jan (22)	20,473	2200	18,000	22,023	2497	18,000	39,881
Feb (23)	26,221	2150	18,000	18,486	2507	18,000	31,725
Mar (24)	22,643	2150	18,000	15,575	2519	18,000	31,264

All weights are in tons and costs are in Indian rupees per ton

Supposing the manager wishes to know the answer to the following question: If the subabul shelf life is restricted to 3 months, is it still possible to get a feasible solution, and if so, at what cost? Running the DSS software developed for this, the answer to this question can be obtained at the click of a button. The answer is that there will be shortages and that there is no feasible solution. The solution obtained using the artificial variables (x_js and y_js) exhibits that the shortages are as follows: subabul shortages are 1473 t in August, 5043 t in September and 1901 t in November; and the eucalyptus shortages are 2051 t in the month of September. This information would help the manager in revising his plans.

5.3 Planning After the First Month

We shall now look at a scenario at the end of April (month number 13). Now the planning is required for the rest of the months. Recall that the forecasts of availabilities for subabul and eucalyptus for the month of April were 17,283 and 11,738 t, respectively. The optimal solution at the beginning of April suggested to procure these amounts (see Table 10.2). Imagine, due to lack of sufficient quantity of eucalyptus, only 10,550 t of eucalyptus could be procured. Also, the actual production quantity consumed is 27,226 t of total wood. Based on the actual consumptions and procurements, the position of material availabilities and forecasts are given in Table 10.3. The solution for this scenario is shown in Table 10.4. The problem has a feasible solution and the total cost, the objective value (for the rest of the period), is Rs. 1,108,098,387.

Table 10.2 Solution for the decision making problem at the beginning of the planning horizon (April)

From (i)	To (j)	(l^s_{ij})	s_{ij}	From (i)	To (j)	(l^e_{ij})	e_{ij}
9	13	0.68	16,700	7	13	0.63	6000
10	13	0.68	6152	8	14	0.63	17,000
10	14	0.68	20,593	9	15	0.63	14,500
11	15	0.68	15,466	10	14	0.63	278
12	16	0.68	16,964	10	15	0.63	8159
13	13	1.00	5922	10	16	0.63	17,816
13	17	0.68	11,361	11	17	0.63	14,620
14	14	1.00	7529	12	18	0.63	10,456
14	18	0.68	8730	13	13	1.00	7818
15	15	1.00	8290	13	19	0.63	321
15	19	0.68	7482	14	14	1.00	771
16	16	1.00	565	14	20	0.63	11,706
16	20	0.68	13,822	15	20	0.63	6057
17	17	1.00	5644	15	21	0.63	5191
17	21	0.68	6148	16	16	1.00	7023
18	18	1.00	9549	16	21	0.63	596
19	19	1.00	10,215	16	22	0.63	4165
19	23	0.68	5505	17	17	1.00	7086
20	20	1.00	6459	18	18	1.00	7796
20	24	0.68	12,156	18	24	0.63	578
21	21	1.00	20,672	19	19	1.00	15,272
21	24	0.68	7898	19	24	0.63	342
22	22	1.00	20,473	20	20	1.00	403
23	23	1.00	16,865	20	24	0.63	23,582
23	24	0.79	5433	21	21	1.00	9756
				22	22	1.00	16,793
				22	24	0.63	3826
				23	23	1.00	11,104
				23	24	0.68	311
				24	24	1.00	7816

Note: Only non-zero s_{ij}s and e_{ij}s in the solution are presented

6 Summary

Wood inventory management is a crucial problem for pulp mills and paper industry. The problem involves making decisions on procurement of wood of different species, quantities to be stored and to be fed directly into production. The constraints include maintaining direct feed proportions, ensuring shelf lives, furnish percentage, monthly opening inventory requirements, production requirements and wood availabilities. Experience-based decisions are usually far from optimality, and optimization models are indispensable for getting optimal solutions. This chapter presented the problem as a linear programming problem. Examples and solutions of sample scenarios are presented. The company where this problem was solved was provided with a DSS, a software tool, to solve the problem as and when required. This will facilitate smooth planning and analysis of different scenarios.

Table 10.3 Inputs for a_{si}, a_{ei}, c_{si}, c_{ei}, q_{si}, q_{ei}, and r_i

	Subabul			Eucalyptus			
Month (i)	a_{si}	c_{si}	q_{si}	a_{ei}	c_{ei}	q_{ei}	r_i
Oct. (7)	22,000	2222	–	0	2300	–	–
Nov. (8)	23,000	2122	–	14,000	2100	–	–
Dec. (9)	0	2322	–	14,500	1900	–	–
Jan. (10)	23,904	2000	–	26,254	2300	–	–
Feb. (11)	15,466	2000	–	14,620	2300	–	–
Mar. (12)	16,964	2000	–	10,456	2300	–	–
Apr. (13)	12,920	2100	18,000	7850	2478	18,000	27,226
May (14)	16,260	2200	18,000	12,478	2455	18,000	33,201
Jun. (15)	15,772	2150	18,000	11,249	2485	18,000	33,071
Jul. (16)	14,386	2300	18,000	11,784	2475	18,000	30,350
Aug. (17)	11,793	2300	18,000	7086	2488	18,000	29,663
Sep. (18)	9549	2300	18,000	8374	2389	18,000	29,868
Oct. (19)	15,720	2300	18,000	15,614	2423	18,000	30,794
Nov. (20)	18,615	2300	18,000	23,985	2395	18,000	27,447
Dec. (21)	32,338	2100	18,000	30,515	2475	18,000	38,257
Jan. (22)	20,473	2200	18,000	22,023	2497	18,000	39,881
Feb. (23)	26,221	2150	18,000	18,486	2507	18,000	31,725
Mar. (24)	22,643	2150	18,000	15,575	2519	18,000	31,264

All weights are in tons and costs are in Indian rupees per ton

Exercises

1. For more versatile values of l_{ij}s, design a two-way table for capturing the inputs.
2. Explain why if first in first out (FIFO) is to be considered in the model, it will become a nonlinear programming problem. Argue that the model proposed in this chapter, the linear programming model, does not necessarily give an optimal solution to the problem with FIFO. Also, explain how the solution obtained by the model proposed in this chapter can still be adjusted to implement FIFO.
3. Consider the solution presented in Table 10.2. Note that the solution presents the original values. Verify that the production requirements are actually met by this solution. Also compute the direct and indirect feeds for each month from the solution presented in Table 10.2. Compute the furnish met in each month for the solution provided in Table 10.2.
4. Consider the following requirement. The management wants to ensure that at the end of the planning horizon, there should be 15,000 t of subabul that is 3 months old, 22,000 t of subabul that is 2 months old and 17,000 t of subabul that is 1 month old. Explain how you can ensure this using the proposed model.
5. *A Rolling Horizon approach for modeling this problem*: This is a simpler and more direct approach for this problem. In this approach, for constructing the model for the problem, the "year" does not play any role. The model depends only on the current month, for which we need to determine how much subabul, eucalyptus wood to buy in the current month for direct feed this month into the

Table 10.4 Solution for the decision making problem at the beginning of the planning horizon (May)

From (i)	To (j)	(l^s_{ij})	s_{ij}	From (i)	To (j)	(l^e_{ij})	e_{ij}
10	14	0.68	23,904	8	14	0.63	13,680
11	15	0.68	15,466	9	15	0.63	14,500
12	16	0.68	16,964	10	15	0.63	8194
13	17	0.68	12,920	10	16	0.63	17,816
14	14	1.00	5270	11	17	0.63	14,620
14	18	0.68	10,990	12	18	0.63	10,456
15	15	1.00	8268	13	19	0.63	7026
15	19	0.68	7504	14	14	1.00	3030
16	16	1.00	565	14	20	0.63	9447
16	20	0.68	13,822	15	20	0.63	4235
17	17	1.00	4581	15	21	0.63	2138
17	21	0.68	7212	16	16	1.00	7023
18	18	1.00	9549	16	21	0.63	4761
19	19	1.00	11,129	17	17	1.00	7086
19	23	0.68	4591	18	18	1.00	6255
20	20	1.00	6507	18	24	0.63	2120
20	24	0.68	12,109	19	19	1.00	10,133
21	21	1.00	19,946	19	24	0.63	5481
21	24	0.68	8768	20	20	1.00	2918
22	22	1.00	20,473	20	24	0.63	21,066
23	23	1.00	17,488	21	21	1.00	9058
23	24	0.79	4726	22	22	1.00	19,408
				23	23	1.00	11,104
				24	24	1.00	7816

Note: Only non-zero s_{ij}s and e_{ij}s in the solution are presented

paper making process, and for storing in the inventory and using to feed in each of future months.

Let L = maximum$\{L_s, L_e\}$, the maximum of the shelf lives of subabul and eucalyptus woods. Consider the current month to be month 0. Then, the planning horizon $H = \{j : 0 \leq j \leq L\}$. The decision variables to be determined for this current month are: s_{0j}, e_{0j} = tons of subabul, eucalyptus wood to purchase in the market for feeding into the paper making process in month j for each $j \in H$.

As time moves forward from month 0 to the next, 0 drops out and $L + 1$ gets added to the planning horizon. and a corresponding model has to be solved in that month 1 in this approach. That is why this type of approach is known as the rolling horizon approach. The model for the problem to be solved in month 1 is similar to that in month 0 with appropriate changes for the passage of time; the model with these changes has to be solved each month. Consider the current month 0. In this month, we already have the solutions obtained for the models in earlier months, from which we can easily compute the stock levels of both subabul and eucalyptus purchased in earlier months for feeding into the paper making process in month 0. Assuming that all other data are as given above, construct the model to be solved in the current month 0; and describe how this model can be used by the company for its decision making over time.

References

1. Paper. http://en.wikipedia.org/wiki/paper. Accessed 30 June 2014.
2. History of paper. http://en.wikipedia.org/wiki/History_of_paper. Accessed 30 June 2014.
3. Sappi: Inspired by life. www.sappi.com. Accessed 30 June 2014.
4. Toraspapel. www.torraspapel.com. Accessed 30 June 2014.

Chapter 11
Optimizing the Design of Heat Exchanger Networks in Crude Oil Refineries

Majid M. Al-Gwaiz and Katta G. Murty

1 Introduction

The scarcity of energy resources and environmental impacts from their use have made the pursuit of energy efficiency a pressing matter in several industries, such as the automobile, consumer appliances, and lighting industries. Factories and industrial plants also face rising energy costs, tightened energy regulations, and are subject to the cap and trade regulations; and as such, energy conservation efforts often yield significant returns. In chemical plants, energy is used in several forms such as electricity, steam, oil and gas, and some of its common uses include pumping fluid, running equipment, and heating and cooling. This chapter will primarily focus on designing energy efficient heating and cooling networks used in chemical processes, beginning with crude distillation in an oil refinery.

Heat exchangers are pieces of equipment used in industries in which both inputs and outputs are fluids (liquid or gases). A heat exchanger facilitates the transfer of heat from a hot stream of fluid that needs to be cooled, to a cold stream of fluid whose temperature needs to be elevated. One industry in which heat exchangers are used extensively is crude oil refining. The configuration of various heat exchangers in a refinery, called its **heat exchanger network (HEN)**, determines to a great extent the refinery's expensive thermal energy consumption. In this chapter, we will consider the problem of optimizing the HEN design in a crude oil refinery.

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_11) contains supplementary material, which is available to authorized users.

M. M. Al-Gwaiz (✉) · K. G. Murty
Industrial and Operations Engineering,
The University of Michigan, Ann Arbor, MI, USA
e-mail: mmgwaiz@umich.edu

K. G. Murty
e-mail: murty@umich.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_11, © Springer Science+Business Media New York 2015

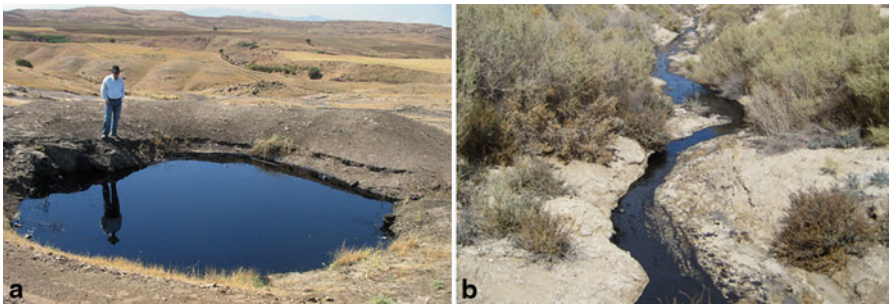


Fig. 11.1 Crude oil seeping up from the ground. **a** Oil seeping from the ground and accumulating in Northern Iraq. (For details see: King, Neil. "An Oasis of Oil in Kurdistan." *The Wall Street Journal*. 9 Jul. 2008). **b** Traces of an oil stream formed by an oil seep in McKittrick, California. (For details see: Elliott, Curt, "Natural Oil Seeps in California." 29 Sep. 2009. http://www.panoramio.com/photo/28085771)

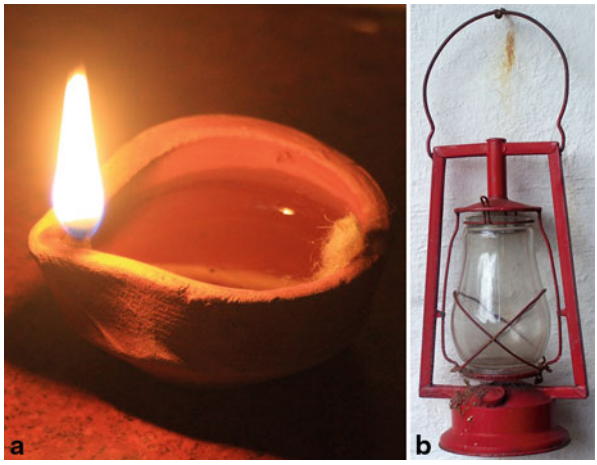


Fig. 11.2 Oil and kerosene lighting. **a** Oil lamp used for lighting in old times. (For details refer to http://en.wikipedia.org/wiki/Oil_lamp). **b** A lantern that works on kerosene. (For details refer to http://en.wikipedia.org/wiki/Kerosene_lamp)

1.1 Brief History of Crude Oil

1.1.1 Crude Oil Origins

Crude oil or petroleum is one of the most valuable natural resources in the world, that's why some people refer to it as **black gold**. It is the lifeblood of our modern industrial civilization. Fuels made from crude oil power our automobiles, airplanes, trucks and busses, trains, ships, some electric power generation units; and literally thousands of other products ranging from paving materials, fabrics, cosmetics, carpets, detergents, plastics, etc. are made from crude oil. The name *Petroleum* comes from Latin words, *petra* (means rock) and *oleum* (means oil), as people have been observing it seeping up from the earth through cracks in the soil and surface rocks since ancient times (Figs. 11.1 and 11.2).

Fig. 11.3 Whales, the most magnificent mammals on earth, being manhandled by humanity. Even now, with all the pollution we dump in the ocean (their habitat), and with the use of communications gear in war ships and submarines of superpowers; we are making life miserable for the whales. This may partly explain why several whales try to commit suicide every year by berthing themselves on beaches. (For details see the source http://costadevelopers.net/blog/dolphins-and-whales-wild-show-in-the-ocean-costa-developers/)

1.1.2 Before Crude Oil

In older times (before the nineteenth century), oil lamps burning some available oil were the most commonly used sources of light. People have been hunting and killing whales from the oceans for their meat and for whale oil to use as fuel for lamps in those days (see Fig. 11.3).

1.1.3 Whale Hunting and Our Shameful Treatment of Them

Whales are very gentle creatures and are a marvel of evolution, and it is a matter of great shame for humanity that by the mid-nineteenth century we made some species of whales extinct and we decimated the populations of other species bringing them to the brink of extinction (see Figs. 11.3 and 11.4). As a result, by mid-nineteenth century the price of whale oil started climbing rapidly. So the hunt was on for a replacement for whale oil for use in lamps. This led to a major breakthrough in the use of petroleum in the 1840s when Abraham Gesner (a Canadian geologist) discovered a distillation process for petroleum that yields kerosene that can be used for lighting lamps. Soon kerosene was being used widely as lamp fuel and the demand for crude oil started climbing (see Fig. 11.2).

Fig. 11.4 Whales committing suicide by berthing themselves on a beach has become an annual phenomenon. Here we show a giant whale shark found by fisherman in a beach on the Arabian sea. It was brought into a harbor in Karachi, Pakistan, using several cranes on Tuesday, February 7, 2012. This whale shark was reportedly unconscious for about 10 days before dying. Residents gather as it is pulled from the water by cranes at Karachi's fish harbor, and sold for 1.7 million Pakistani Rupees (US$ 18,758)

1.1.4 Kerosene as a Replacement to Whale Oil

This started the crude oil refining industry with kerosene as its chief product. Refiners considered other by-products of oil refining as useless in those days and often dumped them in creeks and rivers causing serious pollution.

1.1.5 Uses for Products Other Than Kerosene

The widespread use of kerosene also lead to prospecting for reliable sources of larger quantities of crude oil, and in 1859 Edwin Drake used an old steam engine to drill the first producing oil well near Titusville, Pensylvania. Soon commercial crude oil production and refining spread rapidly throughout the world (see Fig. 11.5). Then by about 1900 electric lights began to replace kerosene lamps, and the automobile rolled on to the world scene. Uses were found quickly for all the products from oil refining and demand for various products from oil refining has been climbing rapidly ever since.

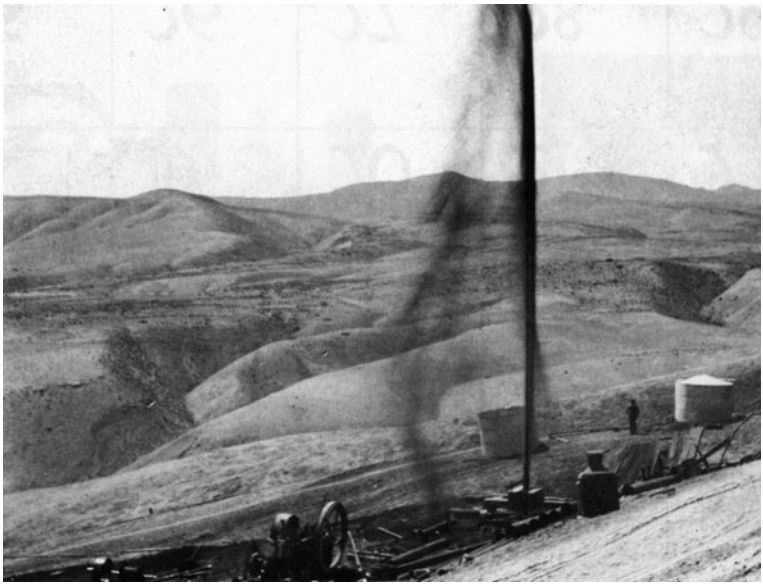


Fig. 11.5 An oil well gushing up vigorously in San Joaquin Valley, California, in 1898. (For details see the source: San Joaquin Valley Geology. http://www.sjvgeology.org/history/gushers.html)

1.2 Crude Oil Refining

1.2.1 Crude Oil Composition

Crude oil is a mixture of a variety of hydrocarbon groups and traces of other components such as sulfur and salts. The first stage of a refinery separates the oil into hydrocarbon fractions (these hydrocarbon fractions are themselves mixtures of a variety of hydrocarbon groups) using a process known as **fractional distillation** which is based on the principle that different hydrocarbon groups boil and vaporize at different temperatures and have different densities. For example, naphtha starts to vaporize at 24°C and some heavy fuel oils start to vaporize at 316°C, and their vapors also condense at different temperatures. A crude oil temperature boiling point curve can be used to determine the product portions a refinery can extract from a crude oil barrel. An example of a crude oil boiling point curve is shown in Fig. 11.6 and its product temperatures and quantities are shown in Table 11.1.

1.2.2 Fractional Distillation in the Crude Distillation Unit

The fractional distillation process takes place in the **crude distillation unit** (CDU) of a refinery, in which crude oil is pumped through pipes then heated to temperatures as high as 400°C converting it into a mixture of hot gasses and liquids. This mixture

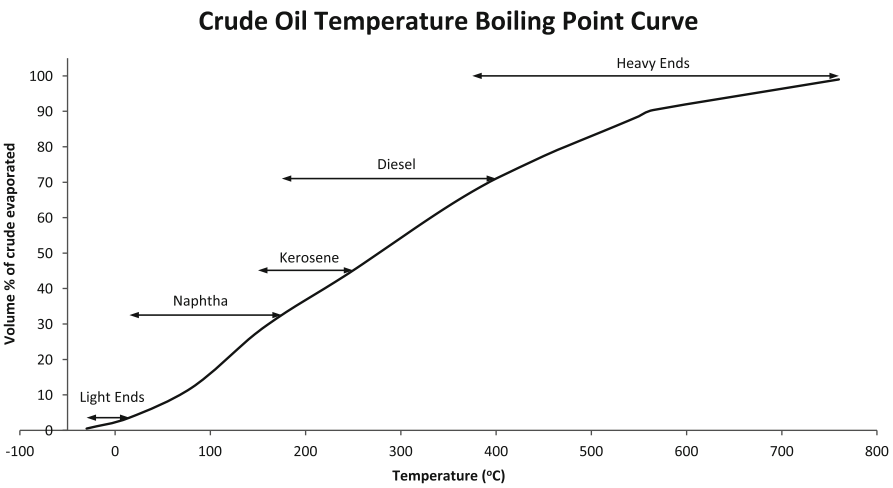


Fig. 11.6 The temperature boiling point curve for the BRENT crude. The five products shown are themselves heterogeneous mixes of hydrocarbons that fall within a boiling temperature range. Notice that product boiling ranges can overlap, and the refinery operators have some flexibility in determining the cutoff point of each product, which naturally has quantity and quality tradeoffs

Table 11.1 The BRENT crude oil products. When the crude oil is gradually heated, the products evaporate. The temperature range at which each product evaporates is shown in the second column. Notice that the temperature ranges for two consecutive products can overlap. In practice, a refinery has to set fine temperature ranges that do not intersect in order to define clear product *cuts*. The *yield* of a product is the quantity of the product as a percentage of the crude oil quantity. The yield can either be measured as the weight percentage or as the volume percent of the crude. The third and fourth columns show the range of weight and volume yield percentages respectively

Product	Temperature (°C)	Weight %	Volume %
Light ends	< 15	< 2.7	< 3.6
Naphtha	15–175	21–26	24–29
Kerosene	150–250	9–17	9–17
Diesel	175–400	26–39	26–39
Heavy ends	> 375	32–37	29–33

is fed into a tall vertical steel cylinder called a *fractionation tower*. As the vaporized fractions rise in this tower they condense at different levels. Vapours of heavy fuel oils condense in the lower section of the tower. Such light fractions as kerosene and diesel condense in the middle section, and naphtha condenses in the upper section. The condensing liquids collect in trays and are drawn off by pipes along the sides of the tower. Some fractions like petroleum gas do not cool enough to condense, these pass out of the top of the fractionating tower into a *vapor recovery unit*. Other fractions which vaporize at temperatures higher than 385°C remain as liquids or semisolids. These residues are recovered from the bottom of the tower and refined into such products as asphalt, lubricating oils and many other products. The fractions produced in fractional distillation are called *straight-run products*. Most of

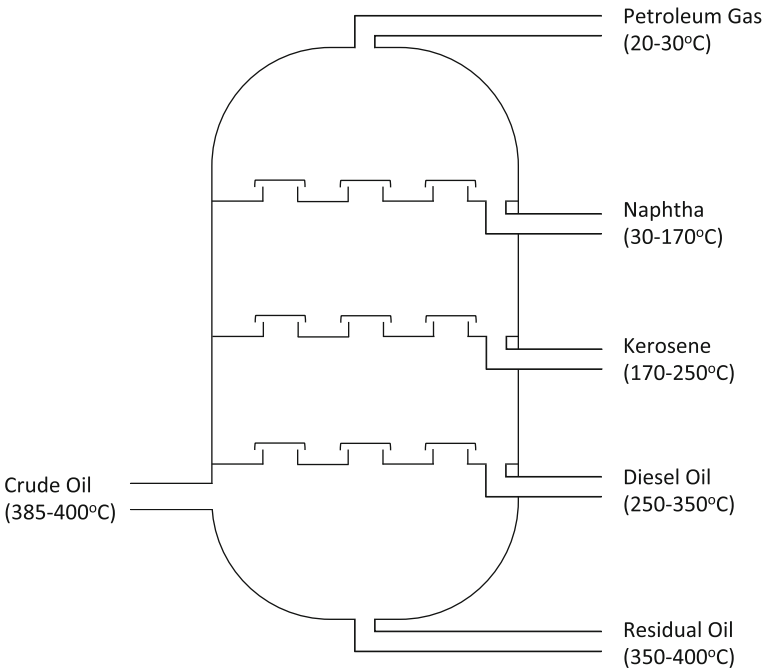


Fig. 11.7 The fractional distillation process

these undergo conversion and chemical treatment in later stages of the oil refinery. Figure 11.7 shows a simplified schematic of a fractional distillation process.

The fractional distillation process consists of several **streams**, where a stream refers to a fluid flowing with the same composition throughout the process but possibly with different temperatures within the process.

The input stream into the fractional distillation process is the crude oil stream entering it, and outputs from it in liquid form are the naphtha, kerosene, light diesel oil (LDO), heavy diesel oil (HDO), and residual oil (RO) streams. The crude oil stream coming into this process through a pipe is typically at the ambient temperature of 30°C (known as its **starting temperature**), while the process requires it to enter at about 400°C (known as its **target temperature**). Similarly the naphtha, kerosene, LDO, and HDO streams coming out of the crude distillation column are at their starting temperatures of 150, 200, 275, and 325°C respectively; these streams are to be cooled to temperatures of 45, 50, 60, and 60°C (their target temperatures) respectively for their next processing after which they are stored in their respective storage tanks. As shown in Fig. 11.6 and Table 11.1 the cutoff temperature between consecutive product groups in fractional distillation processes are not very precise, and what we discussed here is a specific example with specific starting and target temperatures.

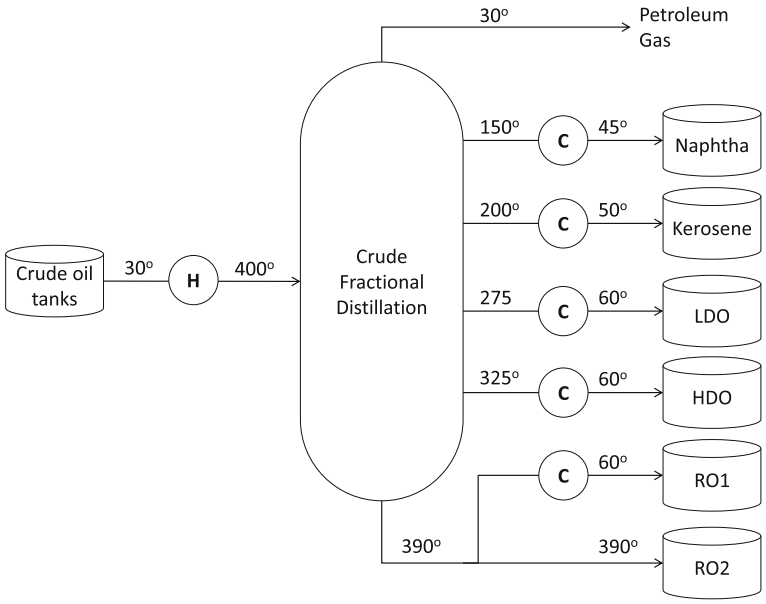


Fig. 11.8 Heating and cooling overview in a crude fractional distillation unit. The *circle* with the letter *C* (H) inside represent coolers (heaters) that reduce (increase) the temperature of a stream to its target temperature

The RO leaving the crude distillation column at its bottom with starting temperature of 400°C has multiple uses. A portion of it is sold to shipping lines under the name **bunker fuel** which they use as fuel for operating their ocean-going vessels. The remaining portion of RO undergoes further processing in subsequent stages in cracking units to convert in into lighter fuels which along with naptha are blended into gasoline (used as fuel in road vehicles like automobiles). So the RO stream coming out of of the crude distillation process is **split** into two separate streams: RO1 (bunker fuel stream), and RO2. The RO2 stream with a starting temperature of 400°C has to be heated to a higher target temperature level before it enters the next stage of processing. Figure 11.8 shows an overview of heating and cooling requirements in a crude fractional distillation process[1].

[1] In reality, fractional distillation processes have several treatment units and feedback flows that are not considered in this model. We are considering a simplistic model in this chapter to better illustrate the heat exchanger network designing method, which can be used for a more complete model as well.

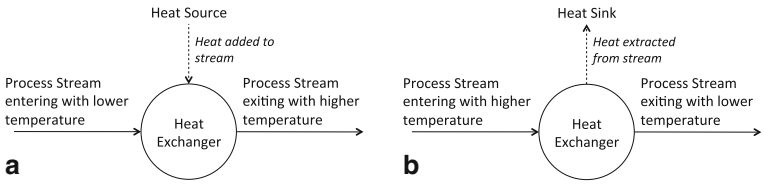


Fig. 11.9 A stream passing through a heat exchanger can either gain heat from a heat source or lose heat in a heat sink. **a** A heat source is used to raise a stream's temperature in a heat exchanger. **b** A heat sink is used to reduce a stream's temperature in a heat exchanger

1.2.3 Fractional Distillation Heating and Cooling Requirements

Heating and cooling streams to their target temperatures is central to the fractional distillation process, and the amount of energy consumed in this process can be substantial. Refiners seek efficient energy consumption in order to reduce the energy bill (electricity and fuel) and to reduce harmful emissions. We will go over some of the heat transfer fundamentals next, then we will discuss the process of optimizing the heating and cooling of streams in the fractional distillation process.

2 Heat Transfer and Heat Exchangers

As explained above, the streams in a CDU are required to be in specified temperature ranges which are typically different from their starting temperatures, so some streams need heating while others may need cooling. Changing the stream temperatures requires a heating source for heating or a heat sink for cooling, and a medium for transferring thermal energy to or from streams, known as a **heat exchanger**, as illustrated in Fig. 11.9. The heating or cooling of a stream is often carried out over several stages with multiple heat exchangers, resulting in a HEN. In this section, we will go over some heat transfer and heat exchanger fundamentals that will lead us to the modeling of HENs.

2.1 Heat Transfer

Heat sources and sinks can either be other streams in the process (for example using a hot diesel stream to heat up crude oil, while simultaneously cooling down the diesel stream) or energy **utilities** (such as steam, water, and fuel). A heat exchanger that transfers heat between a process stream and a utility is called a **utility heat exchanger** while an exchanger that transfers heat between two process streams is called an inter-process or **Process-To-Process** (P2P) heat exchanger. Heating and cooling using utilities requires fuel while heat exchange between process streams does not require energy consumption nor emits pollutants because P2P exchangers

essentially *recycle* thermal energy between process streams. Therefore, maximizing the heat transfer between process streams leads to minimizing the refinery's energy bill.

2.1.1 Energy Utilities and Process Streams

Utilities used for heating (heat sources) are called **hot utilities**. Common examples of hot utilities include steam, hot oil, oil and gas furnaces, and electric heating. Utilities used for cooling (heat sinks) are called **cold utilities**, and some examples of cold utilities are cooling water, fanned air, and refrigerants. Utilities vary in their costs and capabilities; a utility may only be used to raise or lower a stream's temperature to a specific temperature limit. Typically, utilities costs increase as their heating and cooling temperature limits become extreme. For example, a gas fired heater that can heat to over 450°C costs more per unit energy than a steam heater that can only reach up to 140°C. A process stream with a higher target temperature than its starting temperature needs to be heated, and is called a **cold stream**. A stream with a lower target temperature than its starting temperature needs to be cooled, and is called a **hot stream**. In utility heat exchangers hot streams are matched with cold utilities and cold streams are matched with hot utilities, and in P2P heat exchangers hot streams are matched with cold streams. Because energy transferred to a stream is the the same regardless of its source (whether it be a process stream or a utility), hot and cold utilities are often treated as hot and cold streams in the literature, but with positive usage costs.

2.1.2 Heating and Cooling Requirements

The heat (the thermal energy) required to change a material's temperature is measured in Joules (J), and the amount of heat required depends on the material's mass, **specific heat capacity** (denoted c_P), and the required temperature change. The specific heat capacity is a measure of the material's resistance to temperature change, and is measured in Joules per gram-Celcius (J/g°C). Materials with higher c_P values require more energy to raise their temperature. For example, pure water in room temperature has a specific heat of approximately 4.18 J/g°C while crude oil has a c_P in the range of 2 to 2.5 J/g°C (depending on the crude oil type). Consequently, heating a gram of water in room temperature by 1°C requires more energy than heating a gram of crude oil by 1°C. In general, the c_P value of a fluid is a function of its temperature and phase (i.e., liquid or gas). However, it is common to linearize the temperature-c_P curve of a stream into segments with constant c_P values, then consider each segment with a constant c_p value as a separate stream. We will assume in this chapter that the c_P values are constant for each stream.

When dealing with fluid flows, one needs to consider the heat *rate* (with respect to time) because fluid flow, denoted f, is measured in grams per hour (g/h). The heat rate of a fluid is called the **heat duty** and is measured in Watts (W); defined

Table 11.2 Summary of symbols used for fluid properties and heat transfer

Symbol	Unit of measure	Definition
c_P	J/g°C	Specific heat capacity of the fluid
CP	kW/°C	Heat capacity flow rate of the fluid
f	Mg/h	Stream's mass flow rate
Q	kW	Heat duty added to or extracted from a stream in a heat exchanger
T_1	°C	The fluid's temperature at the entry point of the heat exchanger
T_2	°C	The fluid's temperature at the exit point of the heat exchanger

as 1 Joule per second (W = J/s). Because of mass and energy magnitudes, kW (1 kilowatt = 1000 W), MW (1 megawatt = 1,000,000 W), and Mg/h (1,000,000 g/h) are often used for the duty and mass flow units. The heat duty added to or extracted from a stream through heat exchangers to reach its target temperature is denoted Q (in kW). The heat duty for a fluid with a mass flow rate f (in Mg/h) that enters a heat exchanger with a temperature T_1 and exits with a temperature T_2 (both temperatures in °C) and a specific heat capacity c_P (in J/g°C) is $Q = |T_1 - T_2| f c_P/3.6$ kW, where the 3.6 is a factor obtained by converting from hours to seconds (3,600 s/h) and from Mg to kg (1,000 k g/ Mg). We can express this formula as

$$Q = \Delta T \cdot CP \tag{11.1}$$

where $\Delta T = |T_1 - T_2|$ and $CP = f c_P/3.6$ is defined as the **heat capacity flow rate** and is measured in kW/°C. The symbols introduced in this section are summarized in Table 11.2 for quick reference.

It is common to represent Eq. (11.1) graphically as a line with slope $1/CP$ in a temperature versus heat duty plot, known as a **temperature–enthalpy diagram** (shown in Fig. 11.10 for a cold stream). This diagram is often used to visually represent streams because it contains information on the type of stream (hot or cold), input and output temperatures, and the stream's CP value (1/slope). The length of the horizontal line segment Q in Fig. 11.10 is the number of heat units gained or lost by the stream as its temperature changes from T_1 to T_2.

2.2 Heat Exchangers

A heat exchanger (HE) is a device that takes in two fluids at different temperatures, transfers heat from the hotter to the cooler stream, then sends them back to the process. In this section, we will explain how a heat exchanger is used to transfer heat between streams, then we will go over some of its relevant design issues. We will briefly show some of the heat exchanger types in Sect. 2.2.3 and end this section with some common assumptions and considerations when modeling heat exchangers.

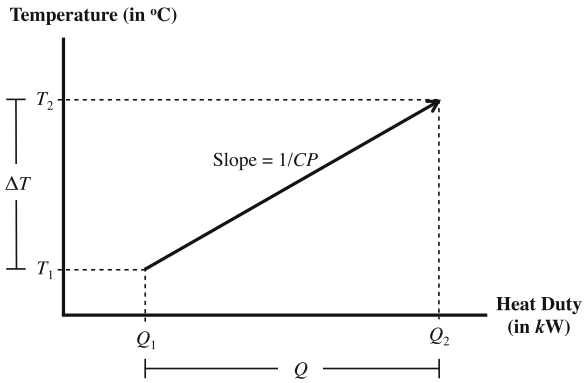


Fig. 11.10 Temperature–enthalpy diagram for a cold stream. In this diagram, the horizontal-axis represents the heat duty and the vertical-axis represents the temperature. The slope of the line is $\Delta T/Q = 1/CP$. Notice that a *directed* line is used in the plot to distinguish between the starting and ending temperatures T_1 and T_2 respectively. Q_1 and Q_2 are the heat content of the stream (in kW) before and after Q kW of heat are added to the stream

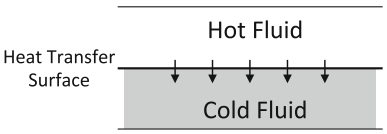


Fig. 11.11 Heat transfer between hot and cold fluids in a heat exchanger. The *line* separating the two fluids represents the separating surface, and heat flows through the surface in the direction of the *arrows*

2.2.1 How a Heat Exchanger Works

Heat is transferred between two fluids in a heat exchanger through a separating surface as illustrated in Fig. 11.11. For convenience, we will use the following notation throughout this section:

- Subscripts "H" and "C" denote hot and cold streams respectively.
- Subscript 1 denotes a stream's entry point to a heat exchanger and subscript 2 denotes its exit point.

Figure 11.12 shows a hot stream with temperature T_{H1} and heat capacity flow rate CP_H that is used to heat a cold stream with temperature T_{C1} and heat capacity flow rate CP_C in a heat exchanger. Given T_{H1}, T_{C1}, CP_H, and CP_C, we would like to determine the hot and cold stream exiting temperatures T_{H2} and T_{C2} respectively. By letting Q denote the quantity of heat transferred between the two streams, Eq. (11.1) can be used to formulate the following energy balance equation

$$Q = (T_{H1} - T_{H2})CP_H = (T_{C2} - T_{C1})C\,P_C. \tag{11.2}$$

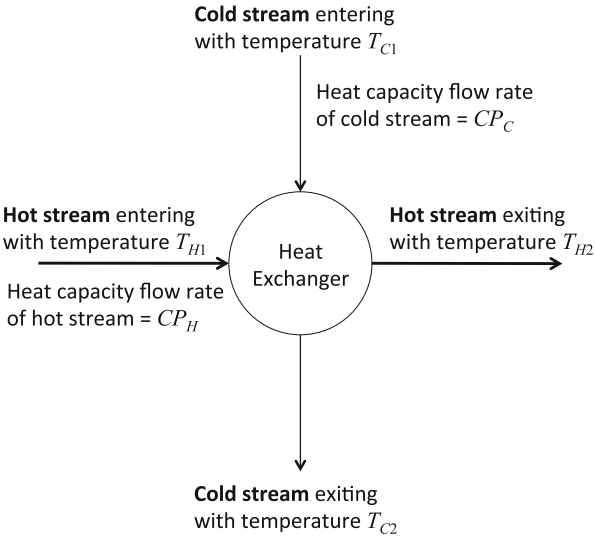


Fig. 11.12 Illustrative heat exchanger schematic. The *bold lines* entering the heat exchanger from the left and leaving from the right make up the hot stream, and the *thin lines* entering the heat exchanger from the top and leaving from the bottom make make up the cold stream

An implicit assumption in (11.2) is that the heat lost from the hot stream exactly equals the heat gained in the cold stream, which is not true in reality due to minor heat dissipation, but is often overlooked in practice for simplicity. Another important thing to remember is that heat can only be transferred from a fluid to a *cooler* fluid (with a smaller temperature); so in a heat exchanger, the temperature of the hot fluid must be higher than its counterpart cold fluid temperature across the heat transfer surface. Therefore, we must have $T_{H1} > T_{C1}$ and we can assume that the inequalities $T_{H1} > T_{H_2} > T_{C1}$ and $T_{H1} > T_{C2} > T_{C1}$ hold.

Equation (11.2) only considers stream attributes, and cannot be used alone to calculate the two variables T_{H2} and T_{C2}. By considering the heat exchanger design and attributes, we can calculate the amount of heat transferred between streams in the heat exchanger as

$$Q = A \cdot U \cdot \Delta T_{LM}, \tag{11.3}$$

where A is the **heat exchange surface area** in square meters (m^2), U is a coefficient known as the overall heat transfer coefficient in $kW/m^{2\circ}C$, and ΔT_{LM} is the logarithmic mean of the temperature difference (LMTD) between the two streams measured in °C, which depends on the temperatures T_{H1}, T_{C1}, T_{H2}, and T_{C2}, as well as the *fluid flow pattern* inside the heat exchanger as we will see in Sect. 2.2.2.

The heat transfer surface area of a heat exchanger is an important design parameter because it sets a tradeoff between the amount of heat transferred between streams and the capital cost of a heat exchanger. As A increases, so does the capital cost of the heat exchanger and the heat transferred between the two streams. If reaching the target temperatures for the hot and cold streams is achievable in a single heat exchanger, where A_H is the smallest heat transfer area at which T_{H2} reaches the hot stream's target temperature and A_C is the smallest heat transfer area at which T_{C2}

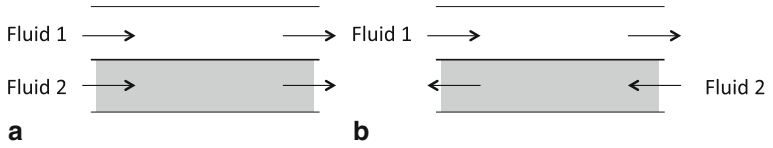


Fig. 11.13 Parallel flow and counterflow heat transfer inside a heat exchanger. **a** Parallel flow configuration: Both fluids flow in the same direction. **b** Counterflow configuration: The two streams flow in opposite directions

reaches the cold stream's target temperature, then obviously a designer would choose $A \leq \min\{A_H, A_C\}$. However, reaching the target temperatures for either stream may not be achievable because the inequalities $T_{H2} > T_{C1}$ and $T_{C2} < T_{H1}$ set a limit on the amount of heat that can be transferred between the two streams (otherwise a fluid would be heating a hotter fluid across some portion of the heat transfer surface, which is impossible). This limit is guaranteed by the ΔT_{LM} variable in (11.3), which depends on the design of the heat exchanger, as we will see next.

2.2.2 Fluid Flow Patterns

In addition to the heat transfer surface area, there are several other HE design characteristics that affect its performance. Notably, The path the two fluids take inside the HE plays a significant role in determining its heat transfer capabilities as it matches the heat transfer temperatures between the two fluids, which is a major factor in determining ΔT_{LM}. Two such path configurations are the **parallel flow** and the **counterflow** patterns illustrated in Fig. 11.13. In the parallel flow configuration hot and cold streams flow in the same direction, so heat is transferred between the hottest and coolest ends of each stream. The two fluids flow in the opposite direction in the counterflow configuration, so heat is transferred between the hot ends of both streams and between the cold ends of both streams. The heat transfer in the parallel flow and counterflow heat exchangers is illustrated graphically in a temperature–enthalpy diagram in Fig. 11.14. In this diagram, the two vertical lines in the plot represent the two ends or terminals of the heat exchangers, and any point on the hot stream's line supplies heat to the point on the cold stream below it. The log mean temperature difference for the parallel flow and counter flow exchangers are

$$\Delta T_{LM}^{Par} = \frac{(T_{H1} - T_{C1}) - (T_{H2} - T_{C2})}{\ln \frac{T_{H1} - T_{C1}}{T_{H2} - T_{C2}}} \text{ and } \Delta T_{LM}^{Cnt} = \frac{(T_{H1} - T_{C2}) - (T_{H2} - T_{C1})}{\ln \frac{T_{H1} - T_{C2}}{T_{H2} - T_{C1}}}$$

respectively.

At this stage, we can solve the problem in Sect. 2.2.1. Given the stream data CP_H, CP_C, T_{H1}, and T_{C1}, the heat exchanger parameters A and U, and the heat exchanger fluid flow pattern, we can use Eqs. (11.2) and (11.3) to solve for T_{H2} and T_{C2}, where ΔT_{LM} is determined by ΔT_{LM}^{Par}'s formula above for a parallel fluid flow pattern heat exchanger design, and by ΔT_{LM}^{Cnt}'s formula above for a counterflow heat exchanger

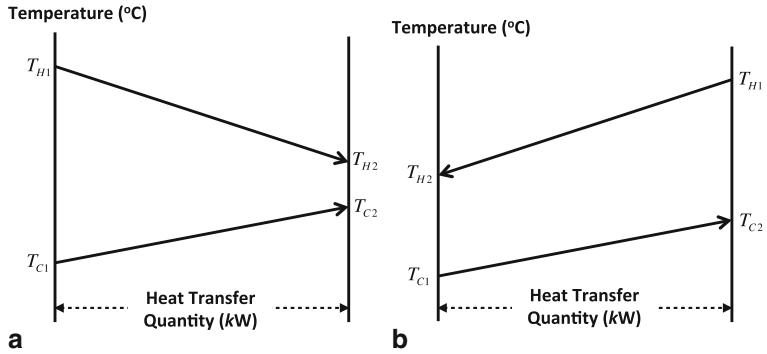


Fig. 11.14 The temperature–enthalpy diagram for the parallel flow and counter flow configurations inside a heat exchanger. The *horizontal line* represents Heat duty in kW and the two *vertical lines* are the temperatures of the steams in °C at the entry and exit points of the heat exchanger. **a** Parallel flow configuration: The two streams start from the same end and have the highest (*lowest*) temperature differential at the entry (*exit*) terminal. **b** Counter flow configuration: The two streams start from opposite ends, where the entry terminal of a stream is the exit terminal of the other stream

design. Equation (11.2) can be used to reduce a variable so that Eq. (11.3) becomes a single variable nonlinear equation, which can be solved using Newton's method (see for example [1, 13]).

Exercise 1 If two fluids with heat capacity flow rates $40k$W/°C and $20k$W/°C respectively and with temperatures 120°C and 50°C respectively enter a heat exchanger with a heat transfer area 10 m^2, an overall heat transfer coefficient $U = 0.8$ kW/m^2°C, and a parallel flow configuration, what would be the temperatures of these streams at their exit points from the heat exchanger? What would these temperatures be if the heat exchanger had a counterflow configuration?

Exercise 2 Given hot and cold streams with $T_{H1} > T_{H2} > T_{C2} > T_{C1}$, show that $\Delta T_{LM}^{Par} > \Delta T_{LM}^{Cnt}$. If your objective were to minimize the heat exchanger surface area A, would you use a parallel flow or a counterflow HE configuration for these fluids (assuming identical overall heat transfer coefficient U for the two exchanger types)?

Exercise 3 For hot and cold streams with $T_{H1} > T_{C1}$, show that the output temperatures must satisfy $T_{H1} > T_{H2} > T_{C2} > T_{C1}$ in a parallel flow heat exchanger (*Hint: an upper bound on heat transfer can be set as the surface area is increased indefinitely*).

Exercise 4 For hot and cold streams with $T_{H1} > T_{C1}$, show that the output temperatures T_{H2} and T_{C2} must be in the range (T_{C1}, T_{H1}), but T_{H2} can be equal to, larger than, or smaller than T_{C2} in a counterflow heat exchanger (*Hint: alter the ranges of the heat transfer surface area to find output temperature ranges*).

Exercises 3 and 4 show that the counterflow HE configuration can achieve a larger range of exit temperatures than the parallel flow HE configuration because the parallel

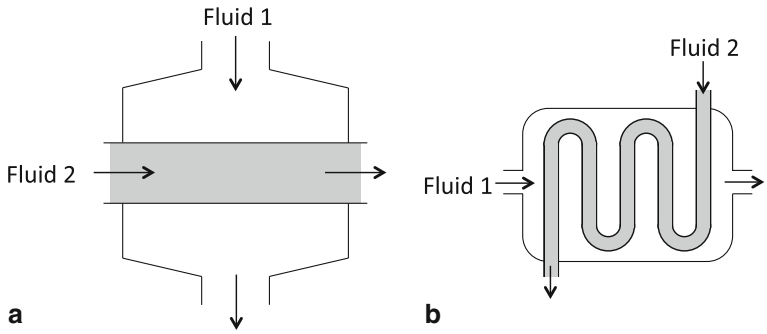


Fig. 11.15 Crossflow and mixed flow heat exchangers. **a** Crossflow heat exchanger configuration: the two streams cross each others paths. In this diagram, *Fluid 2* runs inside a tube enclosed in a vessel with *Fluid 1*. **b** Mixed flow heat exchanger configuration: the two streams can have different configurations in different segments in the heat exchanger. In this illustration, the the heat exchange is crossflow in the segments when *Fluid 2* runs vertically and counterflow when it runs horizontally

flow configuration is limited to solutions with $T_{H2} > T_{C2}$ whereas counterflow heat exchanger solutions can violate this condition. On the other hand, for the same (feasible) output temperatures, Exercise 2 shows that a parallel flow heat exchanger configuration uses less heat transfer surface area, which generally corresponds to a cheaper heat exchanger device.

We have thus far seen the parallel flow and the counterflow heat exchanger fluid flow patterns, but other flow patterns exist such as the *crossflow*, where the two streams cross each others paths (illustrated in Fig. 11.15a), and *mixed flow* (Fig. 11.15b), where the streams can be simultaneously in parallel flow, counterflow, and/or crossflow in different segments of the heat exchanger device. Because of its importance in determining ΔT_{LM}, knowledge of the fluid flow pattern before the initial HEN design is essential. It is typical to assume a counterflow pattern shown in Fig. 11.13b because it gives the largest feasible heat transfer range between streams. Unless specified otherwise, **we will assume counterflow heat exchanger devices in our analysis throughout the remainder of this chapter**. Therefore, the ΔT_{LM} formula is given by

$$\Delta T_{LM} = \frac{(T_{H1} - T_{C2}) - (T_{H2} - T_{C1})}{\ln \frac{T_{H1} - T_{C2}}{T_{H2} - T_{C1}}}. \tag{11.4}$$

2.2.3 Heat Exchanger Types

We have seen in Sect. 2.2.1 that a heat exchanger works by transferring heat between two flowing fluids across a heat exchange surface. There are many types of heat exchangers in industry that fundamentally work as explained in Sect. 2.2.1 but vary in the fluid flow structure and the design of the surface separating the two fluids. The most common heat exchanger type in refineries is the **shell and tube heat**

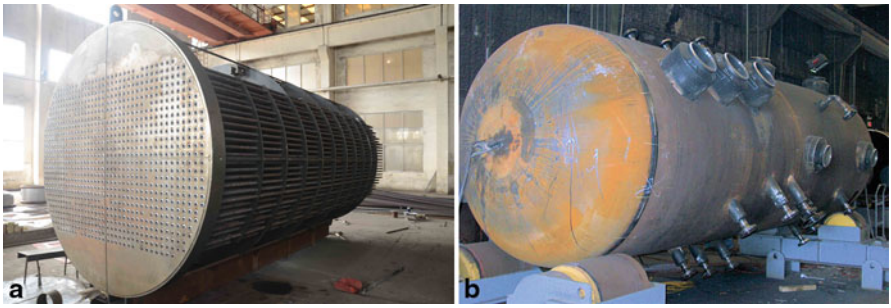


Fig. 11.16 Shell and tube heat exchanger. A fluid goes through a vessel called the shell, and another fluid goes through tubes that run through the shell. The two fluids exchange heat along the tube surface. **a** The tube side of a shell and tube heat exchanger. (For details see the source http://chrisexeter.en.made-in-china.com/product/JeqEZXLPgaHI/China-Shell-and-Tube-Type-Heat-Exchanger.html). **b** The shell side of a shell and tube heat exchanger. (For details see the source http://www.fabscollc.com/aboutS-T.htm)

exchanger depicted in Fig. 11.16. This exchanger has an array of *tubes* (Fig. 11.16a) encapsulated inside a vessel called the *shell* (Fig. 11.16b). As one fluid flows inside the tubes and the other fluid flows through the shell, heat is transferred across the surface of the tubes. Both configurations with the hot fluid flowing in the tubes or with the cold fluid in the tubes (and the other fluid flowing through the shell) are common in practice. Another popular type of heat exchangers is the *plate heat exchanger* (shown in Fig. 11.17), in which hot and cold fluids flow through adjacent clearances between thin plates and exchange heat through the surface of these plates. Some of the features of the plate heat exchangers is that they have a compact design and can achieve pure counterflow heat exchange as depicted in Fig. 11.13b. However, plate heat exchangers aren't as common as shell and tube exchangers in refineries because they cause a higher drop in fluid pressure, clog more frequently, and require more maintenance.

There are several other heat exchanger types used in industry with different attributes, but all types are subject to the same thermodynamic principles. The interested reader can refer to specialized heat exchanger references such as [8]. The choice of heat exchanger type, design, and other attributes is normally carried out in advanced stages of the refinery design, but heat exchangers in the initial design process are assumed to have a counterflow configuration because of its generality.

2.3 Heat Exchanger Modeling

Up to this point, we have not distinguished between P2P and utility heat exchanger models. In reality, the two types of heat exchangers obey the same set of physical laws and can have very similar mathematical models to describe their operation. However, we use a simpler model for the utility heat exchangers because accurately

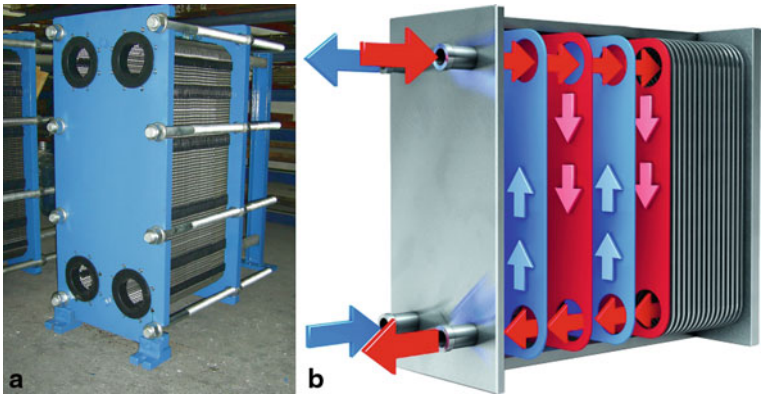


Fig. 11.17 Plate heat exchanger. The heat exchanger is made of thin plates with small clearances in between. Hot and cold fluids flow in these clearances in alternating order to maximize the heat exchange surface between these fluids. **a** Plate heat exchanger. (For details see the source http://yzpanstar.en.made-in-china.com/product/poAJMRrYuiVu/China-Plate-Heat-Exchanger-M15M-M15M-.html). **b** Fluid flow in a plate heat exchanger. (For details see the source http://www.engineerlive.com/content/22778)

calculating the utility output temperatures for utility heat exchangers is not important. For example, the air temperature leaving a fin-fan air cooler to the atmosphere and the temperature of the water leaving a cooling water heat exchanger to be dumped into the sea are insignificant to the design problem. On the other hand, calculating the output temperatures for process streams is essential to the design process. We proceed in this section by first modeling utility heat exchangers then we will describe a more general model that will be used for the P2P heat exchangers.

2.3.1 Utility Heat Exchangers

A utility can be described by two attributes: a utility *cost* and a utility *temperature*. The utility cost is the unit cost of energy (in $\$/k\text{Wh}$)[2] from using the utility. The utility temperature (in °C) for a hot utility is the maximum temperature to which a cold stream can be raised using this utility, and the utility temperature (in °C) for a cold utility is the minimum temperature to which a hot stream can be lowered using this utility. In other words, a cold stream with temperature T_{C1} and a heat capacity flowrate CP_C can be heated by a hot utility with a utility temperature T_{HU} if and only if $T_{C1} < T_{HU}$. Furthermore, the maximum temperature that can be attained using this utility is T_{HU}. If we let Q (in kW) be the amount of heat absorbed by the cold stream and T_{C2} be the cold stream's outlet temperature, then Eq. (11.2) can be used to calculate Q. The **hot utility heat exchanger model** for a hot utility to raise the

[2] Remember that kW is a unit for the energy *rate* (i.e., energy per unit time), so kWh (kilowatt-hour) is an energy unit.

temperature of a cold stream is given by

$$T_{C1} < T_{C2} \leq T_{HU}, \text{and} \tag{11.5}$$

$$Q = (T_{C2} - T_{C1})CP_C. \tag{11.6}$$

If we let c_{HU} (in \$/kWh) be the hot utility's unit cost, then the cost of using this utility is calculated as $c_{HU}Q$ (in \$ per hour of stream flow).

Similarly, for a cold utility with a temperature T_{CU}, the minimum temperature to which a hot stream can be cooled is T_{CU}. Therefore, a cold utility in a cold utility exchanger with a utility temperature T_{CU} that extracts Q kW of heat from a hot process stream with temperature T_{H1} and CP value CP_H to cool it down to T_{H2} can be described by the following **cold utility heat exchanger model** for a cold utility to lower the temperature of a hot stream

$$T_{CU} \leq T_{H2} < T_{H1}, \text{and} \tag{11.7}$$

$$Q = (T_{H1} - T_{H2})CP_H. \tag{11.8}$$

The utility cost is given by $c_{CU}Q$ (in \$/h), where c_{CU} (in \$/kWh) is the cold utility's per unit cost. Although the utility heat exchanger models described in this section are sufficient to carry out initial HEN designs, it is important to note that more advanced utility heat exchanger models are used in later plant design stages. In the remainder of this section, we will go over more complex models for P2P heat exchangers.

2.3.2 The Minimum Approach Temperature $\Delta T_{\min}$

Determining the fluid temperatures around a heat exchanger requires simultaneously solving Eqs. (11.2) and (11.3), which are coupled by ΔT_{LM} given by Eq. (11.4). Although solving for the fluid temperatures might be manageable for an existing heat exchanger, this problem is not trivial when designing a new heat exchanger because both the heat exchange surface area A and the output temperatures need to be determined while considering economic tradeoffs between the capital cost of a heat exchanger (heat exchange surface area, material, type, etc.) and the cost of energy utilities over the lifespan of the equipment. In addition, the rate of heat transfer from a hot stream to a cold stream becomes very slow when the difference between their temperatures becomes small. So in practice, we select a parameter $\Delta T_{\min} > 0$ (known as the **minimum approach temperature** for the temperature differential), and impose the following inequality constraints for the *counterflow* heat exchangers we are considering:

$$T_{H1} - T_{C2} \geq \Delta T_{\min} \tag{11.9}$$

$$T_{H2} - T_{C1} \geq \Delta T_{\min}. \tag{11.10}$$

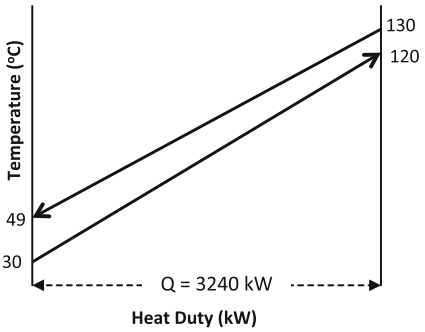


Fig. 11.18 The temperature–enthalpy diagram for a counter flow heat exchangers for a hot fluid with an entering temperature $T_{H1} = 130\,^{\circ}\text{C}$ and $CP_H = 40\,k\text{W}/^{\circ}\text{C}$ and a cold fluid with an entering temperature $T_{C1} = 30\,^{\circ}\text{C}$ and $CP_C = 36\,k\text{W}/^{\circ}\text{C}$. Heat transfer is limited by the temperature difference between the entering hot fluid and the exiting cold fluid

The heat exchanger surface area A is typically calculated at a later stage after determining the temperatures around the heat exchanger and the quantity of heat exchanged. This simplification enables the decomposition of the heat exchanger design problem and the energy cost minimization problem as we will see in Sect. 3.

Equation (11.2) along with Inequalities (11.9) and (11.10) characterize the feasible set of temperatures around a heat exchanger, which will be used as the P2P heat exchanger model in the remainder of this chapter. Example 2.1 below demonstrates how $\Delta T_{\min}$ is used to set heat transfer limits in a counterflow heat exchanger.

Example 2.1 (**Minimum approach temperature constraints**)

Consider a hot stream with $T_{H1} = 130\,^{\circ}\text{C}$ *and* $CP_H = 40\,k\text{W}/^{\circ}\text{C}$ *and a cold stream with* $T_{C1} = 30\,^{\circ}\text{C}$ *and* $CP_C = 36\,k\text{W}/^{\circ}\text{C}$. *For a* $\Delta T_{\min} = 10\,^{\circ}\text{C}$, *a single counterflow HE can transfer* $3,240k\text{W}$ *between the streams and achieve* $T_{H2} = 49\,^{\circ}\text{C}$ *and* $T_{C2} = 120\,^{\circ}\text{C}$ as shown in Fig. 11.18, *where no additional heat transfer is possible because* $\Delta T_{\min} = T_{H1} - T_{C2}$, *and any additional heat transfer would violate the minimum approach temperature constraint.*

The choice of $\Delta T_{\min}$ sets a tradeoff level between the heat exchange surface area and the fluid temperature differentials. A small $\Delta T_{\min}$ allows greater heat transfer between streams[3] at the expense of larger heat exchanger surface areas[4]. Figure 11.19 illustrates this tradeoff. $\Delta T_{\min}$ is considered a design parameter typically set by experienced HEN designers *before* the design process, which usually depends on the HEN application and the plant's economics. In fact, HEN designers usually evaluate the optimal HEN under multiple $\Delta T_{\min}$ values to choose an appropriate $\Delta T_{\min}$ for the HEN[5]. Different $\Delta T_{\min}$ values may also be used for different streams or heat exchangers within the same HEN, but we will only consider the case with a single

[3] Larger heat transfer between streams leads to less consumption of energy utilities.
[4] The capital cost of a heat exchanger device typically increases with its surface area.
[5] $\Delta T_{\min}$ values typically range from 5 to 35°C.

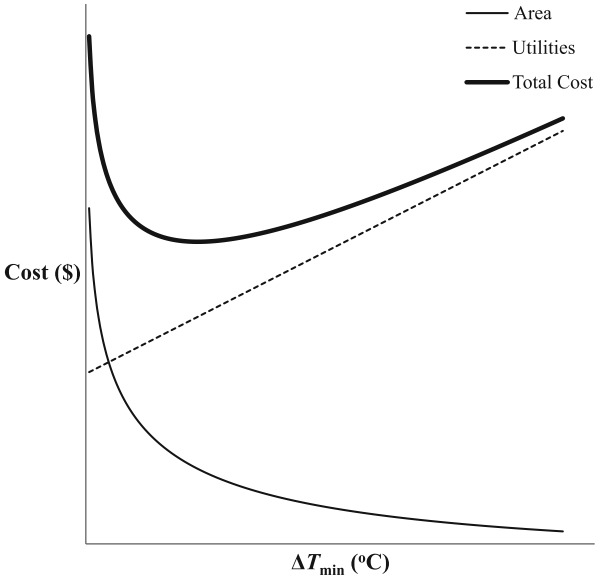


Fig. 11.19 This figure illustrates three cost plots: (1) the cost of the heat exchanger surface area needed annualized over the equipment life cycles, (2) the annual utilities cost, and (3) the total of these costs, all plotted in the vertical axis as a function of the $\Delta T_{\min}$ value used for the design, which is plotted on the horizontal axis. Lower $\Delta T_{\min}$ values can achieve higher interprocess heat transfer, which in turn reduces the utility costs, but requires larger heat exchanger surface areas that make the heat exchangers more expensive. As $\Delta T_{\min}$ increases the utility consumption generally increases linearly while the area reduces in a nonlinear manner. HEN designers seek a $\Delta T_{\min}$ that achieves the overall minimum cost

global $\Delta T_{\min}$ that is used for all heat exchangers. Because our aim is to illustrate the HEN design concept, we assume the following throughout this chapter:
A single (global) $\Delta T_{\min}$ value that is used for all heat exchangers is given as part of the problem data.

2.3.3 Heat Exchanger Limits

Heat transfer between any two fluids is physically feasible as long as they have different temperatures. However, from Eq. (11.3), the closer two fluid temperatures are to one another the more difficult heat transfer becomes, in the sense that a larger incremental heat transfer surface area is required. Although heat transfer may be feasible, there comes a point where it may not be practical or economical because of the large surface area requirement. We have set the limiting temperature differential $\Delta T_{\min}$ between the two fluids in Sect. 2.3.2 to draw a line on what is considered practical and impractical heat transfer. Indeed, according to Inequalities (11.9) and (11.10) (and the fact that that $T_{H1} > T_{H2}$ for the hot stream and $T_{C1} < T_{C2}$ for the cold stream) a hot stream and a cold stream can be *matched* in a heat exchanger if

and only if

$$T_{H1} - T_{C1} > \Delta T_{\min}. \tag{11.11}$$

This inequality can be considered a **feasibility rule** to determine potential hot and cold stream pairings when designing HENs for handling a given set of hot and cold streams.

In addition to stream matching feasibility, the minimum approach temperature $\Delta T_{\min}$ sets the range of feasible heat transfer rates Q and the range of feasible output temperatures T_{H2} and T_{C2}. To see this, restrict T_{H2} and T_{C2} in Eq. (11.2) by the two Inequalities (11.9) and (11.10) to get $Q \leq (T_{H1} - (T_{C1} + \Delta T_{\min}))CP_H$ and $Q \leq ((T_{H_1} - \Delta T_{\min}) - T_{C1})CP_C$. Therefore,

$$0 < Q \leq (T_{H1} - T_{C1} - \Delta T_{\min})CP_{\min} \equiv Q_{\max}, \tag{11.12}$$

where $CP_{\min} = \min\{CP_H, CP_C\}$ and $Q_{\max}$ is defined as the maximum attainable heat transfer rate between the two streams. It is left as an exercise to show that the output temperature ranges are given by

$$T_{H1} > T_{H2} \geq \frac{CP_{\min}}{CP_H}(T_{C1} + \Delta T_{\min}) + (1 - \frac{CP_{\min}}{CP_H})T_{H1} \text{and} \tag{11.13}$$

$$T_{C1} < T_{C2} \leq \frac{CP_{\min}}{CP_C}(T_{H1} - \Delta T_{\min}) + (1 - \frac{CP_{\min}}{CP_C})T_{C1}. \tag{11.14}$$

It is important at this stage to distinguish between a stream's *output temperature* from a heat exchanger (T_{H2} and T_{C2}) and its *target temperature* (which we refer to as T_H^t and T_C^t for the hot and cold streams respectively) introduced in Sect. 1.2.2. The stream target temperatures are input data that a HEN designer attempts to achieve, while the heat exchanger output temperatures are calculated values from a heat transfer process. A stream typically goes through a series of heat exchangers before it reaches its target temperature, so the heat exchanger output temperature only matches the stream's target temperature in the last heat exchanger that it flows through. To ensure that the target temperature is not exceeded, we restrict $T_{H2} \geq T_H^t$ and $T_{C2} \leq T_C^t$.

Exercise 5 (**Heat transfer temperature limit**) Show that the minimum outlet temperature of the hot stream in a heat exchanger is $\frac{CP_{\min}}{CP_H}(T_{C1} + \Delta T_{\min}) + \left(1 - \frac{CP_{\min}}{CP_H}\right) T_{H1}$, and that the maximum outlet temperature of the heat exchanger for the cold stream is $\frac{CP_{\min}}{CP_C}(T_{H1} - \Delta T_{\min}) + \left(1 - \frac{CP_{\min}}{CP_C}\right) T_{C1}$.

2.3.4 Pinch Temperatures

The maximum heat that can be transferred between streams in a heat exchanger, $Q_{\max}$, is attained when one of the Inequalities (11.9) or (11.10) becomes binding (i.e., the

LHS of the inequality and its RHS are equal). This happens when the temperature difference between the two fluids reaches the critical value $\Delta T_{\min}$. The point at which the two fluids have a $\Delta T_{\min}$ temperature differential is significant because it acts as a heat transfer "bottleneck." This bottleneck point is called the **pinch** point, and is defined by a **hot pinch** temperature and a **cold pinch** temperature (on the hot and cold fluids respectively) that are $\Delta T_{\min}$ apart. Example 2.1 has a hot pinch temperature of 130 °C and a cold pinch temperature of 120 °C as illustrated in Fig. 11.18.

When two streams exchange heat at their maximum duty $Q_{\max}$ given by (11.12) a pinch point occurs at the binding inequality. In other words, if Inequality (11.9) is binding then hot and cold pinch temperatures are T_{H1} and T_{C2} respectively, and if (11.10) is binding then pinch temperatures are T_{H2} and T_{C1} respectively. If both inequalities are binding, then the whole temperature intervals $[T_{H2}, T_{H1}]$ and $[T_{C1}, T_{C2}]$ make up the hot and cold pinch temperatures respectively. The binding inequality can be determined by the relative *CP* values between the hot and the cold streams as shown in Fig. 11.20. When both inequalities (11.9) and (11.10) are binding then the largest temperature change can be attained, which is $\Delta T_{\min}$ away from the other stream's inlet temperature. Inequalities (11.9) and (11.10) are simultaneously binding if the hot and cold streams have the same *CP* value, which is unlikely to occur in practice. However, HEN designers can produce streams with equal *CP* values by *stream splitting*, which is discussed in Sect. 3.2.2. This graphical representation of pinch points in a temperature–enthalpy diagram can be used to get design insights as we will see in Sect. 2.4 next.

2.4 Graphical HEN Design Method

We discuss in this section a method for designing HENs by analyzing the temperature–enthalpy diagram. We will go through two methods for manipulating the temperature–enthalpy lines. First, we will explain how shifting these lines may change the problem in Sect. 2.4.1, then we will show how splitting these lines can be useful in designing the HEN in Sect. 2.4.2. We then get the HEN design by successively shifting and splitting the temperature–enthalpy lines in a way that avoids violating the pinch bottlenecks in order to attain a feasible design.

2.4.1 Shifting the Temperature–Enthalpy Line

The temperature–enthalpy diagram first introduced in Sect. 2.1.2 gives a graphical representation of a process stream. A process stream in a HEN problem is defined by its starting temperature, its target temperature, and its *CP* value. The heat duty for this stream, Q, can be calculated by multiplying the *CP* value by the absolute difference between the starting and target temperatures using Eq. (11.1). This heat duty Q is shown graphically as the horizontal length of the temperature–enthalpy line. In Fig. 11.10, for example, the line starts at the point (Q_1, T_1) and ends at (Q_2, T_2),

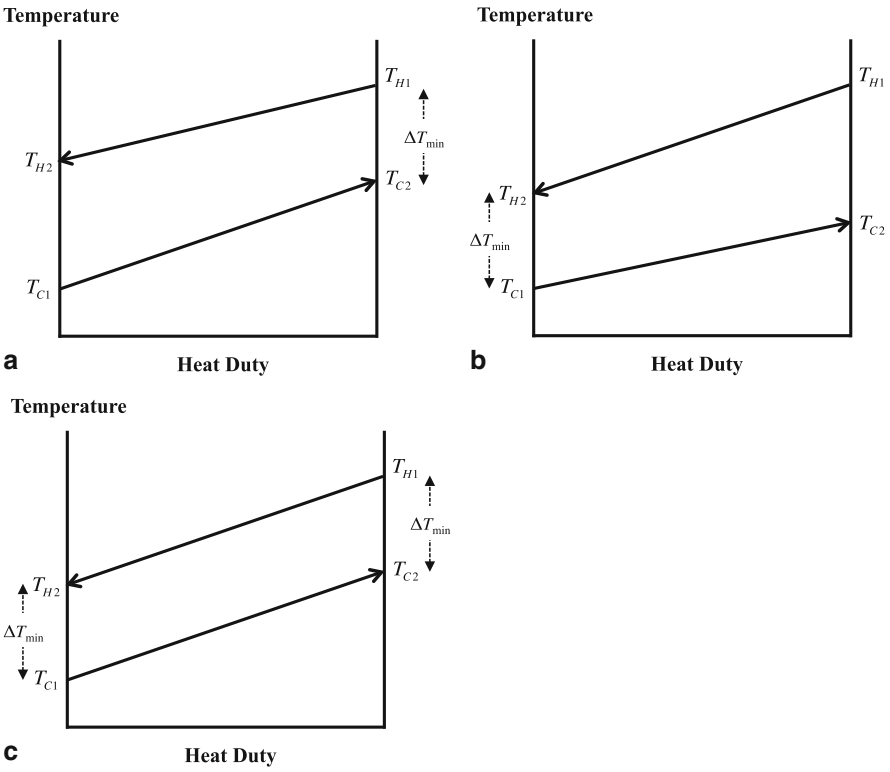


Fig. 11.20 The maximum heat transfer between a hot and a cold stream are shown on temperature–enthalpy diagrams for different CP values. The diagrams show that as the relative values of CP_H and CP_C change so do the binding constraint and the pinch temperatures. **a** $CP_H > CP_C$: inequality (11.9) is binding and the hot and cold pinch temperatures are T_{H1} and T_{C2} respectively. **b** $CP_H < CP_C$: inequality (11.10) is binding and the hot and cold pinch temperatures are T_{H2} and T_{C1} respectively. (c) $CP_H = CP_C$: both inequalities (11.9) and (11.10) are binding and the hot and cold pinch temperatures are given by their entire respective temperature ranges

so $Q = CP\Delta T = Q_2 - Q_1$. Notice that the Q_1 and Q_2 values are arbitrary, and using any two duties Q_1' and Q_2' that have a difference of Q would suffice as starting and ending duty values. Therefore, using $Q_1 + \varepsilon$ instead of Q_1 and $Q_2 + \varepsilon$ instead of Q_2 (for any finite ε value) in Fig. 11.10 is an alternative way to represent the same stream. This means that shifting the temperature–enthalpy line horizontally is permissible because it does not change the stream's data. Figure 11.21 shows three temperature–enthalpy lines with the same slope and vertical (temperature) start and end points, but with different horizontal (heat duty) start and end points. These lines are horizontally shifted images of each other, so they all represent the same hot stream with a starting temperature of 300°C, a target temperature of 100°C, and a CP value of 30 kW/°C.

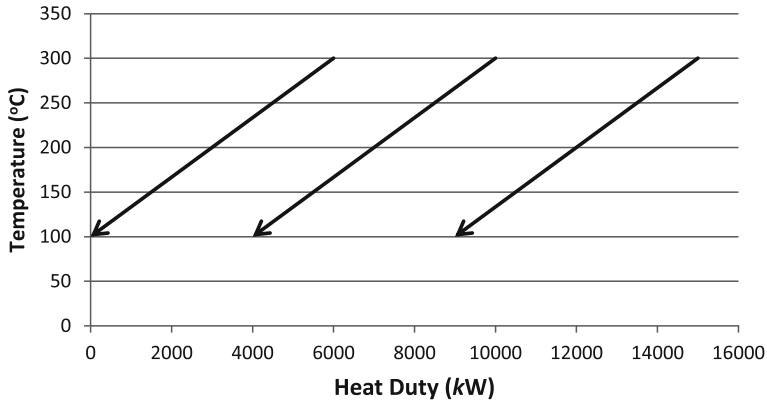


Fig. 11.21 This figure shows three temperature—enthalpy *lines* for the same hot stream with starting temperature 300°C, target temperature 100°C, and *CP* value 30 kW/°C. The three *lines* have a horizontal difference (*heat duty*) of 6000 kW, where the *first line* runs from 15,000 to 9000 kW, the *second line* from 10,000 to 4000 kW, and the *third line* from 6000 to 0 kW

A vertical shift of a temperature–enthalpy line preserves the stream's heat duty and the temperature differential between the starting and target temperatures because the shifted line would maintain the horizontal and vertical length of the original line. However, lines with the same vertical length (temperature differential between the starting and target temperatures) may have different starting and target temperatures, which means that a vertical displacement of a temperature–enthalpy line does change the stream's input and output temperatures, which essentially changes the stream. Figure 11.22 shows the two lines with a 100°C vertical shift. The two lines represent two different streams in this case. The following example illustrates how shifting the temperature–enthalpy curves can be used in designing a two stream HEN.

Example 2.2 (**Two streams with a single pinch point**)

Consider the hot stream H and the cold stream C with the data given in Table 11.3, *where* T^s *refers to each stream's starting temperature and* T^t *refers to its target temperature. If we draw the two streams' lines in the same temperature–enthalpy diagram the lines would intersect and their right end points would not be vertically aligned (see Fig.* 11.23). *Because the starting temperature of the cold stream is* 100 °C, *the cold stream cannot exchange heat with segments of the hot stream's line that have lower temperatures than* $100\,^{\circ}\text{C} + \Delta T_{\min}$. *If we set* $\Delta T_{\min}$ to 10 °C *in this example, then the cold stream can only exchange heat with segments of the hot stream with temperatures* 110 °C *and above, so another heat sink is needed for H. Furthermore, the cold stream's heat demand is* 9000 kW, *which is greater than the hot stream's excess heat of* 7500 kW, *so another heat source is also needed for C. We will assume that we have a cold utility with temperature* 50 °C *(so it can lower H's temperature to its target) and a hot utility with temperature* 250 °C *(so it can increase C's temperature to its target). Assume that the unit cost of both utilities is* \$1/kWh. *Our objective is to get each stream from its starting temperature to its*

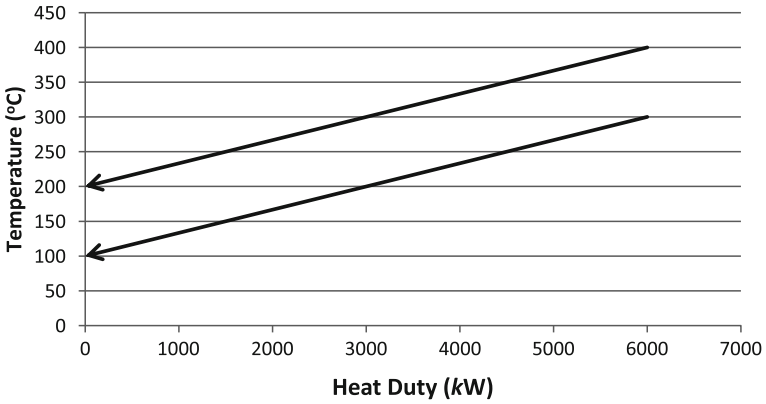
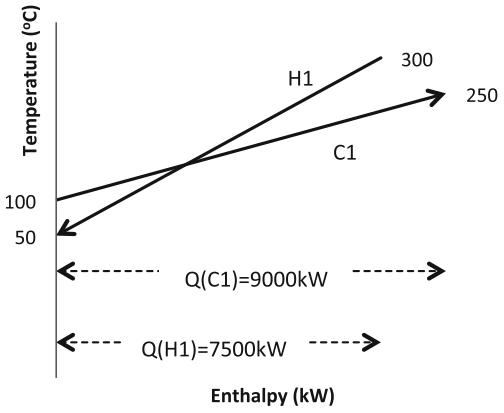


Fig. 11.22 This figure shows two temperature–enthalpy *lines* shifted by 100°C. The two *lines* represent two different hot streams, both with *CP* values 30 *k*W/°C and heat surpluses of 6000 *k*W. However, the starting and target temperatures for the stream represented by the higher temperature–enthalpy line are 400 and 200°C, while the starting and target temperatures for the stream represented by the *lower lines* are 300 and 100°C

Table 11.3 Stream data for Example 2.2

Stream	T^s (°C)	T^t (°C)	*CP* (*k*W/°C)	*Q* (*k*W)
H	300	50	30	7500
C	100	250	60	9000

Fig. 11.23 Temperature–enthalpy diagram for hot and cold streams in Table 11.3. The minimum approach temperature constraint is violated, so lines cannot exchange all their heat and other heat sources and/or sinks are required



target temperature with the minimal utility cost, which corresponds to the solution with the maximum heat transfer between the two streams.

One way to get a feasible heat transfer configuration is by shifting the cold stream's temperature–enthalpy line to the right to the first feasible point where the vertical difference between the two lines is at least 10 °C. *The resulting temperature–enthalpy plot is shown in Fig.* 11.24a. *Notice that the leftmost feasible point is a pinch point.*

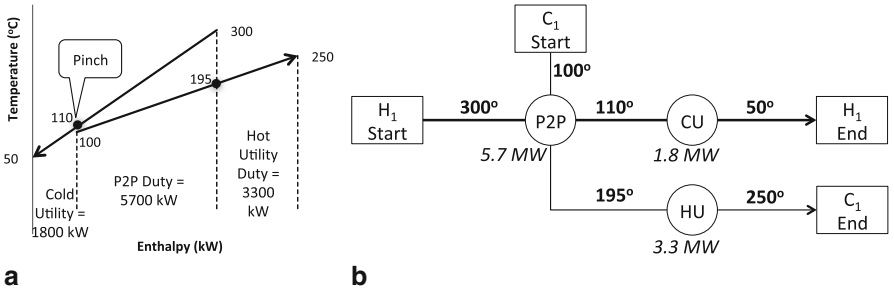


Fig. 11.24 A feasible design configuration for the two stream example in Table 11.3. **a** Temperature–enthalpy diagram with shifted cold temperature–enthalpy line. **b** Heat exchanger network design

After shifting the cold stream the feasible heat exchange range is represented by the horizontal overlap of the hot and cold lines in Fig. 11.24a. The heating and cooling duty of the nonoverlapping regions in the diagram can only be achieved by utilities or by combining some other appropriate streams. However, we will assume in this example that the only other heat sources and sinks are hot and cold utilities. The three regions illustrated in Fig. 11.24a *correspond to three heat exchangers; a hot utility, a cold utility, and a P2P heat exchanger. The temperature–enthalpy diagram shows that the hot stream starts at* 300 °C *then looses* 5.7 *MW of heat to to the cold stream in a P2P exchanger to reach* 110 °C *and finally looses* 1.8 *MW in a cold utility exchanger to end up at* 50 °C. *Figure* 11.24*a also shows that the cold stream starts at* 100 °C, *gains* 5.7 *MW from the hot stream in a P2P exchanger to reach* 195 °C *then gains another* 3.3 *MW in a hot utility exchanger to reach its target temperature of* 250 °C. *The resulting matching configuration of streams (in P2P heat exchangers) and utilities (in utilities heat exchangers) is called a HEN. The HEN corresponding to Fig.* 11.24a *is shown in Fig.* 11.24b.

It is worth noting that the HEN we get from shifting the cold stream beyond the point where the first $\Delta T_{\min}$ *is achieved (i.e., further to the right of Fig.* 11.24a) *would continue to be feasible since the hot and cold lines lines would have a vertical distance that exceeds* $\Delta T_{\min}$. *However, shifting the cold stream further to the right by some value, say* ε *kW, reduces the vertical overlap interval with the hot stream by* ε. *So this* ε *kW must be made up by* 2ε *kW in utilities (one* ε *of hot utilities to heat up the cold stream and one* ε *of cold utilities to cool down the hot stream). Therefore, it is desirable that the curve shift be "tight" so that the HEN would have at least one pinch point.*

2.4.2 Splitting the Temperature–Enthalpy Line

A process stream with starting and target temperatures T_1 and T_2 and with a heat capacity flow rate CP can be represented by two process streams in *series* configuration, where the first stream runs from T_1 to T_3 and the second stream runs from T_3 to T_2 for some temperature T_3 between T_1 and T_2. Both streams in this case would have

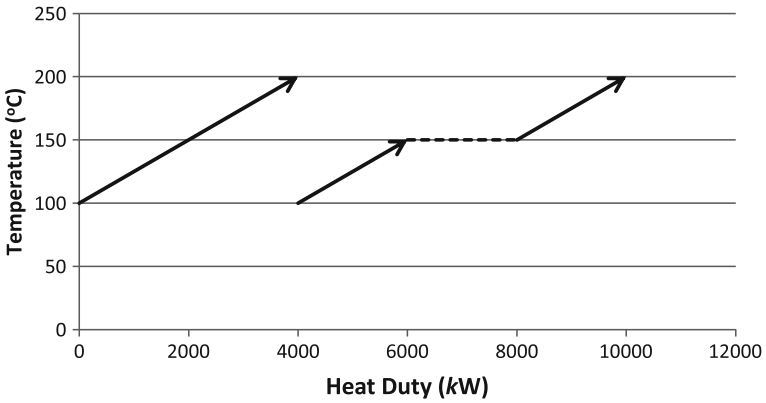


Fig. 11.25 This figure shows the temperature–enthalpy lines of a cold stream with starting temperature 100°C, target temperature 200°C, and *CP* value 40 *k*W/°C on the *left* side of the figure. This streams is split into two streams in series shown on the *right* side of the figure. The two streams in series have the same *CP* value as the original stream, but with temperatures running from 100 to 150°C and from 150 to 200°C respectively

the same heat capacity flow rate *CP*, and their heat duties would sum to the original stream's heat duty ($CP|T_1 - T_3| + CP|T_3 - T_2| = CP|T_1 - T_2|$). Consequently, the temperature–enthalpy line of a process streams can be split into two lines without changing the problem. For example, Fig. 11.25 shows a cold process stream that has starting and target temperatures 100 and 200°C and a *CP* value of 40 *k*W. This stream is split into two cold streams each with a 40 *k*W *CP* value, but with different starting and target temperatures. The first stream has starting and target temperatures 100 and 150°C and the second stream has 150 and 200°C starting and target temperatures as illustrated in Fig. 11.25. Notice that the horizontal distance between the two streams in Fig. 11.25 can be set arbitrarily as explained in Sect. 2.4.1.

In addition to serial configuration, a process stream can be split into two *parallel* streams. In such a configuration, the two resulting streams have the same starting and target temperatures, but their *CP* values sum to the original stream's *CP* value. We will further discuss parallel stream splitting in Sect. 3.2.2. In Example 2.2, we substituted the problem of *maximizing interprocess heat exchange* for the problem of *minimizing utility consumption*. Although increasing interprocess heat exchange usually reduces utility consumption, these two problems are not identical in general except when all the utilities have the same cost. We will show next how stream splitting can be used to get the optimal solution when the two utilities have different costs.

Example 2.3 (**Two streams with two pinch points**) *Consider the problem in Example* 2.2*, but with a different set of utilities. The set of utilities for this problem is given in Table* 11.4*, where* T^u *refers to the utility temperatures and c refers to the utility cost. To minimize the consumption of the hot utility* $HU1$*, we can split the hot*

Table 11.4 Utility data for Example 2.3

Utility	T^u (°C)	c (\$/kWh)
$HU1$	250	1
$HU2$	220	0.5
CU	50	1

stream's line in the temperature–enthalpy diagram to exchange heat at different temperature ranges as shown in Fig. 11.26. Even though both designs in Figs. 11.24 and 11.26 use the same amount of energy utilities, the design in Fig. 11.26 uses cheaper energy. The disadvantage of the cheaper energy design option is the additional heat exchanger units required, which is likely to be more expensive to construct because heat exchanger devices tend to have concave cost functions with fixed installation costs. This leads to investment and energy cost tradeoffs in designs.

We have used in this section an ad hoc approach to design optimal HENs for the two stream problem in Examples 2.2 and 2.3. These examples are solved graphically using intuition by choosing the temperatures at which to exchange heat between streams and the heat exchange quantity. The heat transfer temperatures were essentially chosen by shifting the lines in the temperature–enthalpy diagram to obtain pinch points. The pinch points that resulted from these line shifts are important to the HEN design problem because they set limits to how much heat can be transferred between streams. However, because of the combinatorial nature of HEN problems, intuition often fails when designing HENs with more than two streams. Example 2.4 illustrates this using a 4 stream and 4 utility HEN.

Example 2.4 (**A four stream and four utility example**) *A plant processes a 100 M*g/h *of raw material to produce a single product. The plant schematic is shown in Fig. 11.27. The raw material is fed into a reactor (process 1) and the output goes into a separator (process 2). The separator has two outputs, one goes back as an input to the reactor and the other leaves the plant as a final product. The original feed is supplied at* 30 °C *and the target temperature for the final product is* 40 °C. *Processes between the original feed and the final product have different*

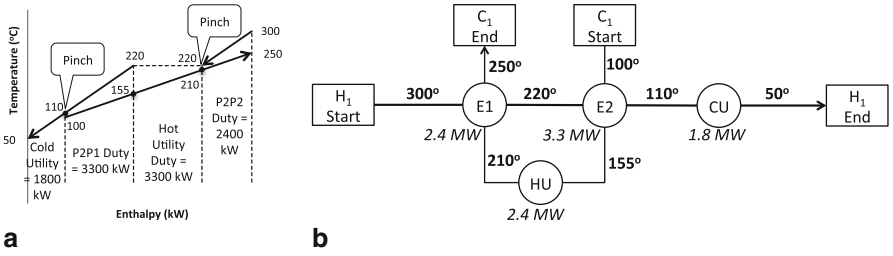


Fig. 11.26 Another feasible design configuration for the two stream example in Table 11.3. Given the energy utility data in Table 11.4, this design has a lower energy utilities cost than the design shown in Fig. 11.24. **a** Temperature–enthalpy diagram with shifted cold curve and divided hot curve. **b** Heat exchanger network design

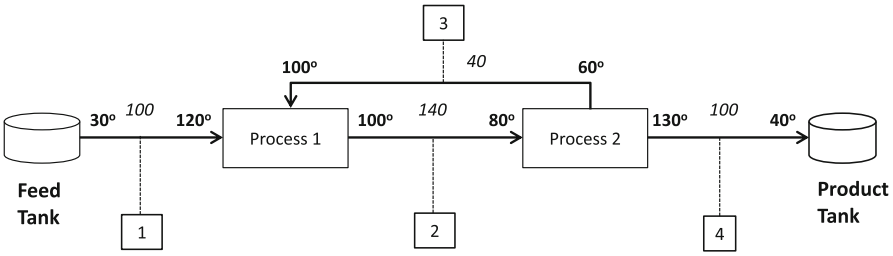


Fig. 11.27 Process schematic for Example 2.4. *Boxed numbers* are the stream indices, *bold numbers* represent temperatures in °C, and *italic numbers* represent material flows in Mg/h

Table 11.5 Stream data for Example 2.4.1

Stream number	Start temp. T^s (°C)	Target temp. T^t (°C)	Mass flow f (Mg/h)	Specific heat capacity c_P (J/gK)
1	30	120	100	1.296
2	100	80	140	2.571
3	60	100	40	7.2
4	130	40	100	1.44

Table 11.6 Utility data for Example 2.4

Utility name	Utility type	Temperature T^u (°C)	Cost c ($/$M$Wh)
$HU1$	Hot	80	0.5
$CU1$	Cold	20	1
$HU2$	Hot	140	2
$CU2$	Cold	50	0.8

temperature requirement. Table 11.5 summarizes the given stream data and the process temperature requirements. Four types of utilities with different temperatures and costs can be used for heating and cooling. The utilities data are shown in Table 11.6. We would like to design a HEN with $\Delta T_{\min} = 10$°C *that satisfies the process heat requirements with the minimum utilities cost.*

Notice that streams 1 and 3 are cold streams while streams 2 and 4 are hot streams. We will name the cold streams $C1$ *and* $C2$ *and the hot streams* $H1$ and $H2$ *respectively. Recall from Sect. 2.1.2 that the heat capacity flowrate of a fluid in* kW/°C *can be calculated by the equation* $CP = f \cdot c_p/3.6$ *and that the heat duty for a stream can be calculated using Eq. (11.1). Table 11.7 shows the stream data with the heat capacity flowrate CP and the heat duty* Q.

The graphical method used in Examples 2.2 and 2.3 cannot be used to determine the optimal heat transfer configuration for four process streams. In order to design the network, we need to determine which pairs of streams to combine in P2P heat exchangers, how much heat to transfer in each heat exchanger, and the order in which streams are matched. Therefore, this problem cannot be solved by enumerating the different hot and cold stream pairs because the amount of heat transferred in each heat exchanger is a continuous variable. Another complicating factor is that once two

Table 11.7 Stream data for Example 2.4 showing the heat capacity flow rate CP and the heat duty Q

Stream name	Start temp. T^s (°C)	Target temp. T^t (°C)	Heat capacity flow CP (kW/°C)	Heat duty Q (kW)
$C1$	30	120	36	3,240
$H1$	100	80	100	2,000
$C2$	60	100	80	3,200
$H2$	130	40	40	3,600

streams exchange heat their output temperatures change, and with it the remaining set of feasible matches and available heat surplus and deficit, so the HEN design problem cannot be solved as a simple matching problem (where hot and cold streams are matched in P2P heat exchangers). Furthermore, it may be beneficial to match two streams more than once at different temperature ranges (as we have seen in Example 2.3)*, so the matching options are not well defined. The tools we have considered so far are not sufficient to get an optimal HEN design for this four stream and four utilities problem. We will consider a more structured method for HEN design in Sect.* 3 *then revisit this example*.

One way of simplifying the HEN design problem is to combine all hot streams into a single "composite hot stream" and all cold streams into a single "composite cold stream." This method of designing HENs is known as the **pinch design method** or **pinch analysis** (see for example [9]) which was first developed by Bodo Linnhoff in the late 1970's[6]. The pinch design method approach is used to maximize the heat transfer (in kW) between all the hot and cold process streams. However, we seek here a method for minimizing the overall utility cost (in $\$/h$). These two problems are not necessarily the same. In the next section, we will go over a systematic method for optimizing the HEN design, where pinch points continue to be central to the design process.

3 Heat Exchanger Networks

Every stream in the CDU has a given starting temperature and a given target temperature as shown in Fig. 11.8. Heat is transferred to or from a stream via a series of heat exchangers to heat it up or cool it down from its starting to its target temperature. The simplest feasible way to achieve this is by using a single utility heat exchanger for each stream to transfer heat between a *costly* utility and the stream. However, we saw in Sect. 2.1 that heat can be recycled within process streams through P2P heat exchangers to reduce the utility heating and cooling load. A P2P heat exchanger is essentially a *matching* or *pairing* between two process streams while a utility heat exchanger is a matching between a utility and a process stream. A *feasible* matching configuration is one that satisfies the thermodynamic rules between streams and

[6] For example, see [11] and [12].

brings process streams closer to their target temperatures. There are numerous feasible matching configurations between streams, each defining a unique HEN. HEN configurations generally have different performance characteristics such as construction cost, flexibility of operations, and energy utilities consumption cost. We will seek in this chapter HEN configurations with the least energy utilities cost. We follow in this section the model and solution method of [14].

3.1 HEN Problem Description

3.1.1 HEN Problem Types

The HEN problems can be classified into *operational* and *design* problems. In a HEN operations problems the network configuration is given and the decision maker controls the fluid flows through the network in order to optimize some performance measure. In the design problem, the decision maker seeks a HEN design that optimizes a performance measure. HEN design problems can be of two types: **grassroots** problem where the network is designed from scratch, and the **retrofit** problem where modifications are made to an existing HEN. We will only consider the grassroots design problem in this section.

3.1.2 Grassroots HEN Design Problem Statement

We are given a set of hot process streams H, a set of cold process streams C, a set of hot utilities HU, a set of cold utilities CU, and a global minimum approach temperature $\Delta T_{\min}$ (used for all stream matches). Denote by N_H, N_C, N_{HU}, and N_{CU} the number of hot steams, cold streams, hot utilities, and cold utilities respectively. For each process stream $i \in H \cup C$ we know its starting temperature T_i^s, its target temperature T_i^t, and its heat capacity flowrate CP_i. For each utility $i \in HU \cup CU$ we are given the utility unit cost c_i (in \$ per kWh) and the utility temperature T_i^u (look at Sect. 2.3.1). We seek to match streams together in P2P exchangers and to match streams with utilities in utilities heat exchangers so that each process stream reaches its target temperature. For each stream, we should determine the *order* of process stream matches and utility matches as well as the *quantity* of heat transfer in each heat exchanger while achieving thermodynamic feasibility for each match (i.e., Equation (11.2) and inequalities (11.9) and (11.10) for P2P exchangers, and (11.5)–(11.8) for utility exchangers). We assume the sufficient availability of utilities of all types. Furthermore, we assume for feasibility that there exists a hot utility $j \in HU$ such that $T_j^u \geq T_i^t$ for all cold streams $i \in C$, and that there exists a cold utility $j \in CU$ such that $T_j^u \leq T_i^t$ for all hot streams $i \in H$. Our objective is to find a design with the **cheapest total utilities cost**. Table 11.8 summarizes the nomenclature used in this problem.

Table 11.8 Summary of the sets and input data that define the HEN problem

Set	*Definition*
H	Set of hot streams
C	Set of cold streams
HU	Set of hot utilities
CU	Set of cold utilities
Data	*Definition*
N_H	Number of hot streams
N_C	Number of cold streams
N_{HU}	Number of hot utilities
N_{CU}	Number of cold utilities
$\Delta T_{\min}$	The minimum approach temperature (in °C)
T_i^s	Starting temperature (in °C) of a process stream $i \in H \cup C$
T_i^t	Target temperature (in °C) of a process stream $i \in H \cup C$
T_i^u	Utility temperature (in °C) of a utility $i \in HU \cup CU$
CP_i	Heat capacity flowrate (in kW/°C) for a process stream $i \in H \cup C$
c_i	Unit cost (in \$/kWh) of a utility $i \in HU \cup CU$

3.2 Solution Methodology

If we model process streams as nodes with supply and demand for heat and model utilities as excess costly supply and demand nodes, then the HEN matching problem becomes a single commodity network-flow problem. However, in order to transfer heat between streams it is not enough to have sufficient supply of heat, one must also have this excess heat at the right temperature. Our approach to solving this problem is to decompose the problem into several subproblems where the temperatures are in the *right* range in every subproblem. This will be done by defining each subproblem over a unique **temperature interval** over which heat transfer is feasible. Afterwards, we will combine these solutions to get a HEN configuration.

In this section, we will seek a temperature interval partitioning that can be used as a basis for the subproblems. To achieve heat transfer feasibility, inequalities (11.9) and (11.10) for P2P heat exchangers and (11.5) and (11.7) for utility heat exchangers must hold in each temperature interval. The difficulty in determining such a partitioning is that the heat flow values depend on the final network configuration. Consequently, we cannot specify the temperatures where these constraints become binding before solving the problem. Fortunately, we can determine potential temperatures where inequalities (11.9), (11.10), (11.5), and (11.7) may be violated and use these temperatures to partition the problem. Before discussing temperature partitioning, we will go over a temperature shifting procedure that simplifies the formulation and the temperature partitioning.

3.2.1 Temperature Shifting

The temperature feasibility constraints in a P2P heat exchanger in which a hot stream i enters with temperature T_{i1} and exits with temperature T_{i2} and a cold stream j enters

Table 11.9 Shifted stream data for Example 2.4

Stream name	Start temp. T^s (°C)	Target temp. T^t (°C)	Heat capacity flow CP (kW/°C)	Heat duty Q (kW)
$C1$	30	120	36	3240
$H1$	90	70	100	2000
$C2$	60	100	80	3200
$H2$	120	30	40	3600

with a temperature T_{j1} and leaves with the temperature T_{j2} is given by

$$T_{i1} - T_{j2} \geq \Delta T_{\min}, \text{and}$$
$$T_{i2} - T_{j1} \geq \Delta T_{\min}.$$

It is a matter of convenience to subtract $\Delta T_{\min}$ from all hot stream temperatures (both starting and target temperatures) in the initial stage of the design in order to simplify this formula to

$$T_{i1} - T_{j2} \geq 0, \text{and} \tag{11.15}$$
$$T_{i2} - T_{j1} \geq 0. \tag{11.16}$$

This shift makes the transfer of heat from a hot stream to a cold stream legal as long as the hot stream has higher terminal temperatures, so $\Delta T_{\min}$ need not be considered. Note that the subtraction of $\Delta T_{\min}$ is carried out for the hot streams, and not the cold streams, and that we are using the same symbols to denote the hot stream temperatures after this subtraction. This transformation does not affect the final result since all hot stream temperatures can be increased by $\Delta T_{\min}$ after attaining the final HEN design.

Inequality (11.7) tells us that if an *unshifted* hot process steam i were to dump its heat to a cold utility j with temperature T_j^u over a cold utility heat exchanger (where stream i enters with temperature T_{i1} and leaves with temperature T_{i2}) then these temperatures must satisfy $T_{i1} > T_{i2} \geq T_j^u$. However, if the hot process stream temperatures were *shifted* (i.e., subtracted by $\Delta T_{\min}$), then Inequality (11.7) would be

$$T_j^u - \Delta T_{\min} \leq T_{i2} < T_{i1}.$$

To rid this inequality of $\Delta T_{\min}$ we will subtract $\Delta T_{\min}$ from all cold utility temperatures. **We will assume that this transformation of hot process stream temperatures and cold utility temperatures holds in the remainder of this chapter.** Therefore, Inequalities (11.15) and (11.16) will be used as the temperature constraints instead of (11.9) and (11.10).

Example 3.1 *The shifted stream and utility temperatures for the four stream and four utility problem in Example* 2.4 *are shown in Tables* 11.9 *and* 11.10. *Recall that* $\Delta T_{\min} = 10$°C *for this problem.*

Table 11.10 Shifted utility data for Example 2.4

Utility name	Utility type	Temperature T^u (°C)	Cost c ($/$MWh$)
$HU1$	Hot	80	0.5
$CU1$	Cold	10	1
$HU2$	Hot	140	2
$CU2$	Cold	40	0.8

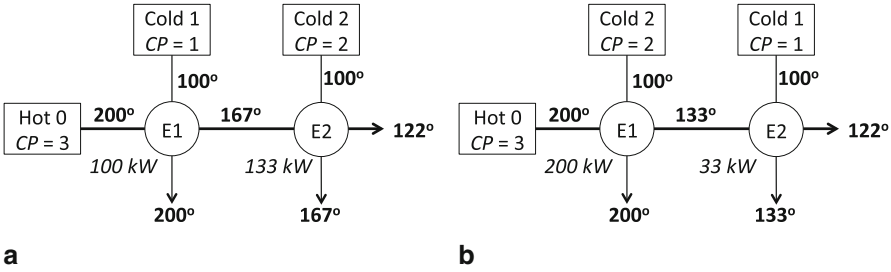


Fig. 11.28 Serial heat exchanger configuration that requires utility consumption. **a** Series heat exchanger configuration with cold stream 1 first. **b** Series heat exchanger configuration with cold stream 2 first

3.2.2 Stream Splitting

There are situations in HEN problems where **splitting** a stream into several parallel branches then combining the branches back into a single stream can increase heat transfer. As an example, consider a singe hot stream indexed 0 with $CP_0 = 3\,k$W/°C and two cold streams indexed 1 and 2 with $CP_1 = 1\,k$W/°C and $CP_2 = 2\,k$W/°C. For simplicity, assume that all streams have common starting and target temperatures 100 and 200 °C (i.e., $T_0^s = T_1^t = T_2^t = 200$ °C and that $T_0^t = T_1^s = T_2^s = 100$ °C). There are only two possible serial HEN configurations for this problem: (a) the hot stream is matched with cold stream 1 then with cold stream 2, or (b) the hot stream is matched with cold stream 2 then cold stream 1. In configuration (a), if the cold stream 1 reaches its target temperature, then the hot stream will start at a lower temperature, so hot and cold utilities must be used to get streams 0 and 2 to their target temperatures. Similarly, in configuration (b), if the cold stream 2 reaches its target temperature, then hot and cold utilities will be needed to get streams 0 and 2 to their target temperatures. Figure 11.28 summarize these solutions.

At a first glance, one may think that utilities must be used in this problem if we were to get all streams to their target temperatures. Recall from Sect. 2.1.2 that $CP_0 = c_{p0} f_0$, where c_{p0} is the hot stream's specific heat capacity in J/g°C and that f_0 is the hot stream's mass flow rate in Mg/h. If the hot stream is **split** into two parallel branches, with flow $\frac{1}{3} f_0$ in the first branch and flow $\frac{2}{3} f_0$ in the second branch, then the CP values of the first and second branches are 1 and 2 kW/°C respectively. Finally, we can match stream 1 with the first hot branch and stream 2 with the second hot branch to get all streams to their target temperatures without using utilities as

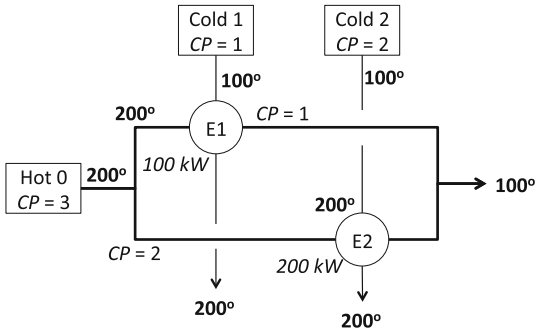


Fig. 11.29 We have one hot stream Hot 0 with $CP = 3$ and cold streams Cold1, Cold 2. By splitting Hot 0 into two streams Hot 01, 02 with CPs 1, 2 respectively; and matching Hot 01, 02 with Cold 1, 2 respectively in heat exchangers, all the streams reach their target temperatures with no utility consumption

shown in Fig. 11.29. Stream splitting is very common in HEN configurations as we will see in the refinery problem later in this chapter.

3.2.3 Uniform Starting and Target Temperatures

Consider the special HEN design problem with the property that the starting and target temperatures for all streams are either T_1 or T_2, and that no utility temperature is in the open interval (T_1, T_2) where $T_1 < T_2$. So $T_i^s = T_j^t = T_2$ and $T_i^t = T_j^s = T_1$ for all streams $i \in H$ and $j \in C$. Because we can neglect all hot utilities except for the one with the cheapest cost, we can assume that the problem has a single hot utility with a temperature $\geq T_2$. Similarly, we can assume without loss of generality that the problem has a single cold utility with a temperature $\leq T_1$. The supply of heat from a hot stream $i \in H$ is $S_i = CP_i(T_2 - T_1)$ and the demand for heat of a cold stream $i \in C$ is $D_i = CP_i(T_2 - T_1)$. We can show that **in this special problem no utility is needed if** $\sum_{i \in H} CP_i = \sum_{i \in C} CP_i$. Consequently, this special HEN has no pinch points and can be designed using only stream splitting.

Proof If the problem only has a single hot stream and a single cold stream then both streams have the same CP value and thus can exchange all their heat without violating (11.15) and (11.16). Therefore, no utilities are needed in this case. Consider the case with one hot stream indexed 0 and m cold streams indexed $1, \cdots, m$. We have $CP_0 = \sum_{i=1}^{m} CP_i$. Let the flow rate of the hot stream be f and its specific heat capacity be c_P, so $CP_0 = f c_P$. We can split the hot stream into m parallel streams with flow $f \frac{CP_i}{CP_0}$ for parallel branch $i \in \{1, \cdots, m\}$. After this split, parallel branch i of the hot stream has supply equal to the demand in cold stream i, so we can completely match the hot parallel branches with their respective cold streams, and thus no utilities are needed in this case. The same can be said about the case with a single cold stream and multiple hot streams.

Now consider the case with n hot streams and m cold streams. Assume that hot stream k has the smallest CP value among all hot and cold streams. If there exists a cold stream with the same CP value then match the two streams to obtain a similar problem but with one less hot and cold stream. If there is no cold stream with heat

capacity flowrate CP_k, then choose any cold stream and split it into two branches with a heat capacity flowrate CP_k in one branch and match it with the hot stream k to eliminate both streams. This transforms the problem into a similar problem with $n-1$ hot streams and m cold streams. Similarly, if the the stream with the smallest CP value happens to be a cold stream, then this problem can be transformed into a similar problem with $m-1$ cold streams and either n or $n-1$ hot streams. In either case, the number of streams is reduced by at least 1 in every iteration, and eventually either the number of the hot or the cold streams will reach 1, which can be solved without utilities as we saw earlier. Therefore, no utility is needed if the hot and cold streams have the same total heat supply and demand.

■

We can extend this result to show that if $\sum_{i\in H} CP_i > \sum_{i\in C} CP_i$ then no hot utility is needed, and if $\sum_{i\in H} CP_i < \sum_{i\in C} CP_i$ then no cold utility is needed. The proofs for these results are left as exercises.

Exercise 6 Show that if $\sum_{i\in H} CP_i > \sum_{i\in C} CP_i$ then the HEN problem with uniform starting and target stream temperatures can be solved with no cold utility consumption.

Exercise 7 Show that if $\sum_{i\in H} CP_i < \sum_{i\in C} CP_i$ then the HEN problem with uniform starting and target stream temperatures can be solved with no hot utility consumption.

3.2.4 Temperature Partitioning

Consider a closed temperature interval $I = [T_1, T_2]$, where $T_1 < T_2$. We will use this interval to define a **subproblem** that we refer to as HEN(I). The subproblem only considers the set of streams whose temperatures ranges intersect the interior of interval I (i.e., excluding I's endpoints). So the set of hot and cold streams considered are

$$H_I = \{i \in H : T_i^s > T_1 \text{ and} T_i^t < T_2\} \text{ and}$$
$$C_I = \{i \in C : T_i^t > T_1 \text{ and} T_i^s < T_2\}$$

respectively.

For each stream, we will only consider the temperature range that results from the intersection of its original temperature range and I. The starting and target temperatures for a hot stream $i \in H_I$ in this subproblem are $T_{iI}^s = \min\{T_i^s, T_2\}$ and $T_{iI}^t = \max\{T_i^t, T_1\}$ respectively. Similarly, the starting and target temperatures for a cold stream $i \in C_I$ in this subproblem are $T_{iI}^s = \max\{T_i^s, T_1\}$ and $T_{iI}^t = \min\{T_i^t, T_2\}$ respectively. Notice that the temperature range of a stream in the subproblem is a subset of the original stream's temperature range. We refer to a stream i in the subproblem as a **stream segment** of the original stream i. The hot utilities that can be used to heat streams in this subproblem are given by $HU_I = \{i \in HU : T_i^u > T_1\}$, and the relevant cold utilities in this subproblem are $CU_I = \{i \in CU : T_i^u < T_2\}$.

If the interior of I does not contain any starting or target process stream temperatures nor utility temperatures then the subproblem defined by I has uniform starting and target temperatures as defined in Sect. 3.2.3. The proof of this result is given next.

Proof If the interior of I does not intersect any starting or target process stream temperature or utility temperature then for $i \in H_I$ $T_i^s \geq T_2$ and $T_i^t \leq T_1$ so $T_{iI}^s = \min\{T_i^s, T_2\} = T_2$ and $T_{iI}^t = \max\{T_i^t, T_1\} = T_1$. Using a similar argument we can also show that $T_{iI}^s = T_1$ and $T_{iI}^t = T_2$ for all cold streams $i \in C_I$, so every stream segment in the subproblem has the uniform temperature endpoints T_1 and T_2. Similarly, $T_i^u \geq T_2$ for all hot utilities $i \in HU_I$, and $T_i^u \leq T_1$ for all cold utilities $i \in CU_I$.

■

Let the total heat supply from all process stream segments in the subproblem defined by I be $S_I = (T_2 - T_1) \sum_{i \in H_I} CP_i$, and the total demand for heat from all process stream segments in this interval be $D_I = (T_2 - T_1) \sum_{i \in C_I} CP_i$. Because the subproblem has uniform starting and target temperatures, then from the results of Sect. 3.2.3 we can conclude that

- If there is excess supply of heat ($S_I > D_I$), then the maximum P2P heat transfer in I is D_I and $S_I - D_I$ kW must be either dissipated in cold utilities in CU_I or supplied to cold streams with temperatures below T_1 (or a combination of both).
- If there is excess demand for heat ($S_I < D_I$), then the maximum P2P heat transfer in I is S_I and $D_I - S_I$ kW must be supplied by either hot utilities in HU_I or by hot process streams with temperatures above T_2 (or a combination of both).
- If $S_I = D_I$ then all process heat can be recycled and no heat sources outside H_I or heat sinks outside C_I are needed.

Temperature Partitioning Procedure We will now use the results from the subproblem I to describe a procedure that can be carried out to partition the problem into closed temperature intervals with interiors in which no starting or target process stream temperature and no utility temperature resides. Combine all process stream starting and target temperatures and all utilities temperatures in one list, and delete duplicate temperatures. Notice that the number of temperatures in this list is at most $2N_H + 2N_C + N_{HU} + N_{CU}$. Let $N + 1$ be the number of temperatures in this list, so $N \leq 2N_H + 2N_C + N_{HU} + N_{CU} - 1$. Recall that the hot process stream temperatures and the cold utility temperatures are assumed to be shifted by $\Delta T_{\min}$ as discussed in Sect. 3.2.1. Now sort all these temperatures in ascending order. Let the sorted list be $T = (T_0, T_1, \cdots, T_N)$, where $T_i < T_j$ for $i < j$. We now have N temperature intervals where each temperature interval $I_i = [T_{i-1}, T_i]$ has no starting or target process stream temperature and no utilities temperature in its interior, so the subproblem defined by I_i must have uniform starting and target temperatures for $i = 1, \cdots, N$. Therefore, $I_1, \cdots, I_N$ can be used as a temperature partitioning for this problem.

It is important to note that there may be many valid temperature interval partitions for the same problem. In fact, there are other algorithms in the literature for temperature partitioning that can result in a smaller number of temperature intervals (see

Table 11.11 Utility data for Example 2.4 after temperature shifting and initial screening

Utility name	Utility type	Temperature T^u (°C)	Cost c (\$/MWh)
$HU1$	Hot	80	0.5
$CU1$	Cold	30	1
$HU2$	Hot	120	2
$CU2$	Cold	40	0.8

for example [2]). Although it would not change the solution, having less temperature intervals would reduce the problem size, so it would require less computation. To reduce the number of temperature partitions in our problem, we do an initial screening of utility temperatures, where we ensure they do no go beyond the highest cold stream target temperature and lowest hot stream target temperature. In other words, let $T_{\max} = \max_{i \in C}\{T_i^t\}$ and $T_{\min} = \min_{i \in H}\{T_i^t\}$. Then, for any hot utility i with temperature $T_i^u > T_{\max}$ we change its temperature to $T_{\max}$, and for any cold utility i with temperature $T_i^u < T_{\min}$ we change its temperature to $T_{\min}$. This screening will not have an effect on the solution, but it will reduce the number of partitions and hence the problem size.

Example 3.2 (**Temperature partitioning**) *We will illustrate the temperature partitioning procedure for the 4 stream and 4 utility problem of Example* 2.4. *Notice that we are using the shifted streams and utilities shown in Tables* 11.9 *and* 11.10. *To conduct the initial screening process, we find the largest stream temperature in the network, which is* 120°C *(target of stream* $C1$ *and starting temperature of stream* $H2$*). We therefore modify* $HU2$*'s temperature to* 120°C. *Similarly,* $C1$*'s starting temperature and* $H2$*'s target temperature* 30°C *is the lowest stream temperature, so we set* $CU1$*'s temperature to* 30°C. *The modifications to the utilities temperatures are shown in Table* 11.11.

The temperature partitioning procedure is illustrated in Fig. 11.30, where each stream is drawn as a directed line with its head at its target temperature and its tail at its starting temperature. Because utilities only have one temperature, they are plotted as single points in Fig. 11.30. *If we superimpose the stream starting and target temperatures along with the utility temperatures on one line we get the eight temperature partitions shown in Fig.* 11.30. *Therefore, the sorted temperature partition list is* (30, 40, 60, 70, 80, 90, 100, 120) *in* °C.

3.2.5 Problem Decomposition

The significance of the temperature partitioning approach is that it enables us to treat streams as heat sources and sinks with fixed supply and demand. Utilities, on the other hand, do not have target heat supply and demand levels, instead, their consumption levels are decision variables set by the HEN designer. In fact, were it not for the utilities, this problem could be modelled as a matching problem. Fortunately, it turns out that we can find the optimal utilities consumption levels efficiently before finding a feasible matching configuration. The problem of finding utility consumption levels

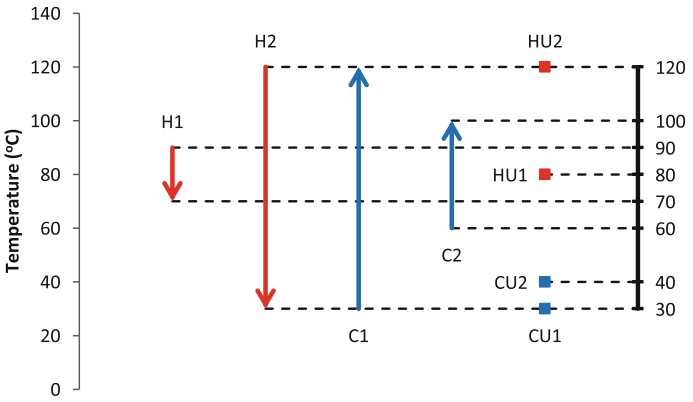


Fig. 11.30 Temperature partitions for problem of Example 2.4

that would result in a feasible HEN design with the minimum utility cost is known as the **energy targeting** problem. After determining utility targets, we can set these targets as fixed heat supply and demand levels for utilities, and essentially treat utilities as if they were regular process streams. Because the modified stream list has fixed heat supply and demand levels, a **stream matching** problem can be used to produce the optimal HEN. This two stage sequential approach will be the subject of the following two sections.

3.3 Energy Targeting

We will use the transshipment model of [14] to solve the energy targeting problem.

Starting from a temperature partitioning $I_1, \cdots, I_N$ sorted in ascending temperatures (where $I_k = [T_{k-1}, T_k]$), our objective is to find the cheapest utility consumption levels that can lead to a feasible HEN design. Let

$q_{ik} =$ the consumption level (in kW) of utility $i \in HU \cup CU$ by stream segments in temperature interval I_k for $k = 1, \cdots, N$.

For a temperature interval I_k, where $k \in \{1, \cdots, N\}$, let H_k, C_k, HU_k, and CU_k be the sets of hot process streams, cold process streams, hot utilities, and cold utilities present in the kth temperature interval respectively. The definitions of these sets is given in Table 11.14. Furthermore, let

$S_{ik} =$ the heat supply for hot stream segment $i \in H_k$, and let
$D_{ik} =$ the heat demand for cold stream segment $i \in C_k$.

The definitions of S_{ik} and D_{ik} are also provided in Table 11.14. Example 3.3 illustrates these sets and parameters for the problem of Example 2.4.

Table 11.12 Stream and utility availability in each temperature interval for Example 2.4's problem

Temp. Int. index, k	Start temp. T_{k-1} (°C)	End temp. T_k (°C)	Hot stream H_k	Cold stream C_k	Hot utility HU_k	Cold utility CU_k
1	30	40	$H2$	$C1$	$HU1, HU2$	$CU1$
2	40	60	$H2$	$C1$	$HU1, HU2$	$CU1, CU2$
3	60	70	$H2$	$C1, C2$	$HU1, HU2$	$CU1, CU2$
4	70	80	$H1, H2$	$C1, C2$	$HU1, HU2$	$CU1, CU2$
5	80	90	$H1, H2$	$C1, C2$	$HU2$	$CU1, CU2$
6	90	100	$H2$	$C1, C2$	$HU2$	$CU1, CU2$
7	100	120	$H2$	$C1$	$HU2$	$CU1, CU2$

Table 11.13 Heat supply and demand for hot and cold streams in each temperature interval for Example 2.4's problem

Temp. Int. index, k	Hot stream supply S_{ik} (MW)			Cold stream demand D_{ik} (MW)		
	$H1$	$H2$	Total supply	$C1$	$C2$	Total demand
1	0	400	400	360	0	360
2	0	800	800	720	0	720
3	0	400	400	360	800	1160
4	1000	400	1400	360	800	1160
5	1000	400	1400	360	800	1160
6	0	400	400	360	800	1160
7	0	800	800	720	0	720
Total	2000	3600	5600	3240	3200	6440

Example 3.3 (**Process stream and utilities in the temperature intervals**) *The temperature intervals and the four sets H_k, C_k, HU_k, and CU_k for Example* 2.4*'s problem are shown in Table* 11.12*. The hot stream supply and cold stream demand for each temperature interval is shown in Table* 11.13*. Notice that we substitute 0 values for S_{ik} if $i \notin H_k$ and 0 values for D_{ik} if $i \notin C_k$ in Table* 11.13*.*

Energy Balance From the discussion in Sect. 3.2.4, a hot stream segment in temperature interval I_k can dissipate heat in one of three sinks:

1. Cold streams present in the temperature interval I_k.
2. Cold utilities in the temperature interval I_k.
3. Cold streams and cold utilities present in cooler temperature intervals ($I_1 \cup \cdots \cup I_{k-1}$).

Equivalently, a cold stream segment in temperature interval I_k can be heated by hot streams in I_k, hot utilities in I_k, or hot streams and hot utilities present in hotter temperature intervals ($I_{k+1} \cup \cdots \cup I_N$). Consequently, the overall energy balance for the subproblem defined by I_k is given by:

$$\begin{aligned}&(\text{Heat supplied by streams } i \in H_k) + (\text{Heat supplied by hot utilities } i \in HU_k)\\ &+ (\text{Heat coming from hot streams and hot utilities in } I_{k+1} \cup \cdots \cup I_N)\\ &= (\text{Heat consumed by streams } i \in C_k) + (\text{Heat dissipated in cold utilities } i \in CU_k)\\ &+ (\text{Heat sent to cold streams and cold utilities in } I_1 \cup \cdots \cup I_{k-1}).\end{aligned}$$

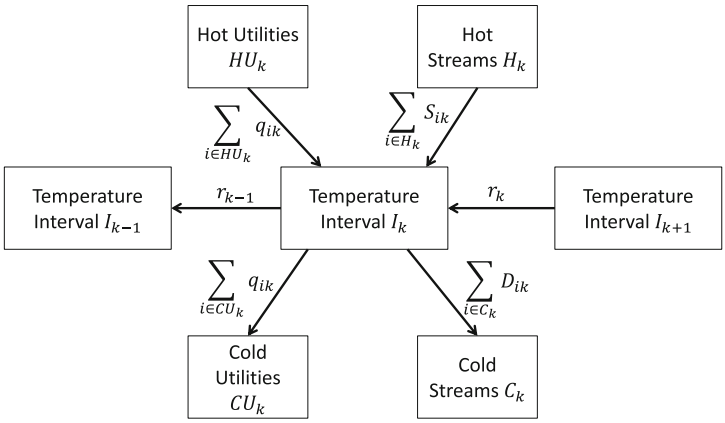


Fig. 11.31 Heat flows for temperature interval I_k

Define the last term in the LHS of this equation (the heat coming into interval I_k from hot streams and hot utilities in intervals $I_{k+1}, \cdots, I_N$) as the the **residual heat** to interval I_k, denoted r_k (in kW). Notice that the last term in the RHS is the heat from intervals $I_k, \cdots, I_N$ going into interval I_{k-1} (that eventually ends up in cold streams and utilities in this interval and intervals $I_1, \cdots, I_{k-2}$), so it is precisely r_{k-1}. Then this balance constraint can be written as

$$\sum_{i\in H_k} S_{ik} + \sum_{i\in HU_k} q_{ik} + r_k = \sum_{i\in C_k} D_{ik} + \sum_{i\in CU_k} q_{ik} + r_{k-1}.$$

The heat flows for temperature interval I_k are shown graphically in Fig. 11.31. Clearly $r_N = 0$ because I_N is the largest temperature interval, and defined $r_0 = 0$ by convention. Table 11.14 summarizes the nomenclature introduced in this section.

Energy Targeting Linear Programming Formulation The energy targeting problem is given by the following linear program

$$\min_{q,r} \sum_{k=1}^{N} \sum_{i\in HU_k\cup CU_k} c_i q_{ik} \tag{11.17}$$

$$\text{s. to } r_k - r_{k-1} + \sum_{i\in HU_k} q_{ik} - \sum_{i\in CU_k} q_{ik} = \sum_{j\in C_k} D_{jk} - \sum_{i\in H_k} S_{ik} \text{ for } k = 1, \cdots, N \tag{11.18}$$

$$q_{ik} \geq 0 \text{ for } k = 1, \cdots, N, i \in HU_k \cup CU_k \tag{11.19}$$

$$r_k \geq 0 \text{ for } k = 1, \cdots, N \tag{11.20}$$

$$r_0 = r_N = 0. \tag{11.21}$$

This linear program is equivalent to a single commodity transshipment network model, where the commodity is heat flow in kW. This model is illustrated in Fig. 11.32

Table 11.14 Summary of the sets, parameters, and variables used in the energy targeting problem

Set	*Definition*
H_k	$= \{i \in H : T_i^t \leq T_{k-1} \text{ and } T_i^s \geq T_k\}$. Set of hot streams present in the kth temperature interval
C_k	$= \{i \in C : T_i^s \leq T_{k-1} \text{ and } T_i^t \geq T_k\}$. Set of cold streams present in the kth temperature interval
HU_k	$= \{i \in HU : T_i^u \geq T_k\}$. Set of hot utilities present in the kth temperature interval
CU_k	$= \{i \in CU : T_i^u \leq T_{k-1}\}$. Set of cold utilities present in the kth temperature interval
Parameter	*Definition*
N	Number of temperature partitioning intervals
T_k	End point of a temperature interval (in °C) for $k = 0, \cdots N$ T_k is assumed to be sorted, so $T_i < T_j$ for $i < j$
I_k	$= [T_{k-1}, T_k]$ for $k = 1, \cdots, N$. kth temperature interval
S_{ik}	$= CP_i(T_k - T_{k-1})$ for $k = 1, \cdots, N$ and $i \in H_k$ Supply of heat by hot stream i in the kth temperature interval
D_{ik}	$= CP_i(T_k - T_{k-1})$ for $k = 1, \cdots, N$ and $i \in C_k$ Demand for heat by cold stream i in the kth temperature interval
r_0	$= 0$. Starting residual heat
r_N	$= 0$. Ending residual heat
Variable	*Definition*
q_{ik}	≥ 0. Consumption level of utility $i \in HU_k \cup CU_k$ in temperature interval I_k for $k = 1, \cdots, N$ (in kW)
r_k	≥ 0. Residual heat to interval I_k (in kW)

for the network in Example 2.4. Nodes $1, \cdots, 7$ in the middle row of Fig. 11.32 represent the seven temperature intervals. The nodes in the top row in the figure are hot utility and hot stream nodes, which represent heat sources[7]. The bottom row of nodes are the cold utilities and cold streams, which represent heat sinks[8]. Flows from hot utilities to temperature intervals and flows from temperature intervals to cold utilities are the utility consumption level variables q_{ik}, while the flows to and from stream nodes are the heat demand $\sum_{i \in C_k} D_{ik}$ and supply $\sum_{i \in H_k} S_{ik}$. The residual heat variables r_k are heat flows between temperature intervals shown in Fig. 11.32. Although not shown in the network, arcs q_{ik} from hot utilities to temperature intervals and from temperature intervals to cold utilities have costs c_i \$/MWh, while all other arcs have 0 costs. Example 3.4 shows the solution to this problem.

[7] A node H_i represents the total supply of heat from both $H1$ and $H2$ into the ith temperature interval.

[8] A node C_i represents the total heat demand for cold streams $C1$ and $C2$ in the ith temperature interval.

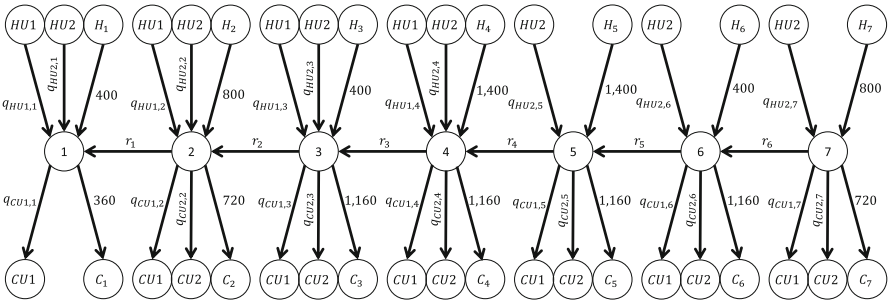


Fig. 11.32 Transshipment network model for the four stream and four utility problem of Example 2.4

Table 11.15 Targeting solution to the problem of Example 2.4

Temp Int. index, k	Residual heat flow r_k (kW)	Utility flow, q_{ik} (kW) $HU1$	$CU1$	$HU2$	$CU2$
1	0	0	40	0	0
2	0	0	0	0	80
3	760	0	0	0	0
4	240	280	0	0	0
5	0	0	0	0	0
6	760	0	0	0	0
7	0	0	0	680	0

Example 3.4 (**Energy targeting**) *The energy targeting solution to the problem of Example* 2.4. *is attained by solving the system* (11.17)–(11.21) *for the data given in Tables* 11.12 *and* 11.13 *and the utility costs given in Table* 11.6. *After solving this system, we get a minimum energy cost (problem objective) of $ 1.604/h, which can be achieved by choosing the decision variables according to Table* 11.15. *Note that by convention we set* q_{ik} to 0 *if* $i \notin HU_k \cup CU_k$ *in Table* 11.15. *Also, Table* 11.15 *does not show the value of* r_0, *which we set to 0 along with* r_7 *because they are parameters as indicated by Table* 11.14.

A solution to the system (11.17)–(11.21) gives utility consumption levels q_{ik} in kW that can result in a feasible HEN design with the cheapest energy cost. It is somewhat counterintuitive to attain the total energy cost and utility consumption values of an optimal design prior to designing the HEN. Now that we know what solution to look for, we will seek in Sect. 3.4 a HEN design that satisfies the energy targets.

3.4 Stream Matching

After getting utility consumption targets, we are ready to start the HEN design. A complete HEN design entails determining the set of heat exchangers in the network, the quantity of heat (in kW) transferred in the heat exchangers, the pairs of streams

(process or utility streams) between which heat is transferred in the heat exchangers, the temperatures (in °C) at which these streams enter and exit the heat exchangers, and the fluid flows (in Mg/h) of the streams in the heat exchangers. We will proceed with the HEN design in this section by deriving a mixed integer programming formulation for matching streams in heat exchangers and determining their heat transfer quantities and temperatures. The HEN design we are looking for must consume the utility values given by the targeting problem's solution. Naturally, there may be many matching configurations that can represent the solution given by the energy targeting problem, so we will choose a HEN configuration that is *likely* to be the cheapest to construct. Because of economies of scale and fixed costs of heat exchanger devices, it is often preferable to **minimize the number of heat exchanger devices**, which will be our objective in the matching problem. It is important to note that this objective may not guarantee the cheapest HEN cost, but it gives good results in practice.

3.4.1 Combining Utilities and Process Streams

The utility consumption values in the stream matching problem are given by the q_{ik} variables we get from solving the targeting problem (11.17)–(11.21). Consequently, we can now set these utility consumption levels as *fixed targets* to be met by a HEN design so that cold utility consumption can be treated as a fixed demand (D) and hot utility consumption can be considered fixed supply (S). Equivalently, **utilities can be regarded in the stream matching problem as regular process streams**, so that the set of hot and cold streams can be modified to

$$\overline{H} = H \cup HU \text{ and}$$
$$\overline{C} = C \cup CU$$

respectively, where the heat supply and demand for the utilities is given by the q_{ik} variables solution from the targeting problem ($S_{ik} = q_{ik}$ for $i \in HU$ and $D_{ik} = q_{ik}$ for $i \in CU$). The set of hot and cold streams present in a temperature interval can be modified to

$$\overline{H}_k = \left\{ i \in \overline{H} : S_{ik} > 0 \right\}$$
$$\overline{C}_k = \left\{ i \in \overline{C} : D_{ik} > 0 \right\}.$$

Example 3.5 shows the modified hot and cold stream sets for the problem of Example 2.4.

Example 3.5 (**Modified sets of hot and cold streams**) *The modified sets of hot and cold streams for the four stream and four utility problem of Example* 2.4 *is*

$$\overline{H} = \{H1, H2, HU1, HU2\} \text{ and}$$
$$\overline{C} = \{C1, C2, CU1, CU2\}.$$

The hot and cold streams present in each temperature interval is shown in Table 11.16.

Table 11.16 Hot and cold streams availability in each temperature interval for Example 2.4's problem

Temp. Int. index, k	Start temp. T_{k-1} (°C)	End temp. T_k (°C)	Hot streams $\overline{H}_k$	Cold streams $\overline{C}_k$
1	30	40	$H2, HU1, HU2$	$C1, CU1$
2	40	60	$H2, HU1, HU2$	$C1, CU1, CU2$
3	60	70	$H2, HU1, HU2$	$C1, C2, CU1, CU2$
4	70	80	$H1, H2, HU1, HU2$	$C1, C2, CU1, CU2$
5	80	90	$H1, H2, HU2$	$C1, C2, CU1, CU2$
6	90	100	$H2, HU2$	$C1, C2, CU1, CU2$
7	100	120	$H2, HU2$	$C1, CU1, CU2$

3.4.2 Pinch Temperatures and Subnetworks

We have so far used the utility flow results q_{ik} from the targeting problem to set utility targets and treat utilities as process streams in the stream matching problem. As it turns out, the residual heat flow results r_k from the targeting problem also play an important role in the stream matching problem as they can be used to decompose the problem. To see this, consider a residual flow value of $r_k = 0$ for some temperature interval index $k \in \{1, \cdots, N-1\}$. The zero flow residual heat value indicates that no heat flows from hot stream segments or hot utilities with temperatures above T_k to cold stream segments or cold utilities with temperatures below T_k. The fact that no interprocess heat transfer occurs in the minimum utility cost solution between stream segments across T_k implies that T_k is a *pinch temperature* (i.e., at least one of the Inequalities (11.15) and (11.16) is binding) because otherwise the targeting solution could be improved by increasing interprocess heat transfer and reducing utilities consumption. Another implication of the lack of heat transfer across the temperature T_k is that we can design the part of the HEN with the stream segments and utilities in temperature intervals I_{k+1} to I_N separately from the part with the stream segments in I_1 to I_k. In general, designing the part of the HEN for the stream segments and utilities with temperature ranges that lie between two pinch temperatures can be completed independently from the rest of the network. We will refer to the network problem between two consecutive pinch points as a **subnetwork**.

Determining the Pinch Temperatures The set of all pinch temperatures for a HEN design problem is precisely the set of temperatures T_k for which $r_k = 0$, where the r_k values are obtained from the solution of the system (11.17)–(11.21). We will denote the list of pinch interval indices as

$$P = (k \in \{1, \cdots, N-1\} : r_k = 0)$$

where the r_k values are the residual heat flows from the solution to the targeting problem. Let the number of pinch points be $N_P = |P|$ and assume that P is an ordered list in increasing order. Let P_i denote the ith entry in P for $i = 1$ to $|P|$. So, $P_i < P_j$ for $i < j$. Notice that each pinch interval corresponds to a pinch

temperature T_k from the list of temperature partitions. Let $\left(T_1^P, \cdots, T_{N_P}^P\right)$ be the ordered list of pinch temperatures where $T_i^P < T_j^P$ for $i < j$. For example, there are 3 pinch points from Table 11.15 that occur at temperature intervals 1, 2, and 5. Notice that we do not consider 0 and 7 as pinch intervals even though $r_0 = r_7 = 0$ by definition. The list of pinch temperatures for this example is (40, 60, 90) which are found by mapping the pinch temperature interval indices to the interval's right end point temperature shown in Table 11.12.

Subnetwork Decomposition of the HEN A subnetwork is the set of stream segments and utilities in the temperature intervals between two consecutive pinch temperatures, so knowing the temperature intervals that belong to a subnetwork is sufficient to determine the set of stream segments and utilities that belong to the subnetwork. If $N_P \geq 1$, the HEN can be decomposed into $N_P + 1$ subnetworks defined by their temperature intervals as follows:

- Subnetwork 1: Contains temperature intervals $1, \cdots, P_1$.
- Subnetwork i: Contains temperature intervals $P_{i-1}+1, \cdots, P_i$ for $i = 2, \cdots, N_P$.
- Subnetwork $N_P + 1$: Contains temperature intervals $P_{N_P}, \cdots, N$.

When the problem has no pinch points ($N_P = 0$) then it cannot be decomposed into smaller subnetworks, so the problem has a single subnetwork that contains all temperature intervals when $N_P = 0$. The problem decomposition into subnetworks is illustrated in Fig. 11.33. Even though temperature intervals 1 and N are not pinch intervals and P_0 and $P_{N_P+1} \notin P$, we will define $P_0 \equiv 1$ and $P_{N_P+1} \equiv N$ to simplify the subnetwork temperature intervals notation. Using this notation, we can write the temperature intervals for subnetwork i as $P_{i-1}+1, \cdots, P_i$ for all $i = 1, \cdots, N_P+1$.

Let the set of subnetworks be $SN = \{1, \cdots, N_P + 1\}$ and let the number of temperature intervals in subnetwork $s \in SN$ be N_s. We will use $\overline{H}^s$ and $\overline{C}^s$ for the set of hot and cold streams in subnetwork s. To reduce the set sizes, we will exclude utilities that have 0 flow in the subnetwork (as given by the q_{ik} variables in the targeting problem) even if they are usable in the subnetwork. Therefore, the sets of hot and cold subnetwork streams are given by

$$\overline{H}^s = \left\{ i \in \overline{H} : \sum_{k=1+P_{s-1}}^{P_s} S_{ik} > 0 \right\}$$

$$\overline{C}^s = \left\{ i \in \overline{C} : \sum_{k=1+P_{s-1}}^{P_s} D_{ik} > 0 \right\}.$$

Example 3.6 shows the different subnetworks for Example 2.4's problem.

Example 3.6 (**Finding pinch temperatures and subnetworks**) *The pinch interval indices for the problem in Example 2.4 are the indices k for which* $r_k = 0$ *in Table 11.15, which are the 3 indices* $P = (1, 2, 5)$, *so* $N_P = 3$. *These temperature interval indices correspond to the three pinch temperatures 40, 60, and 90* °C. *These*

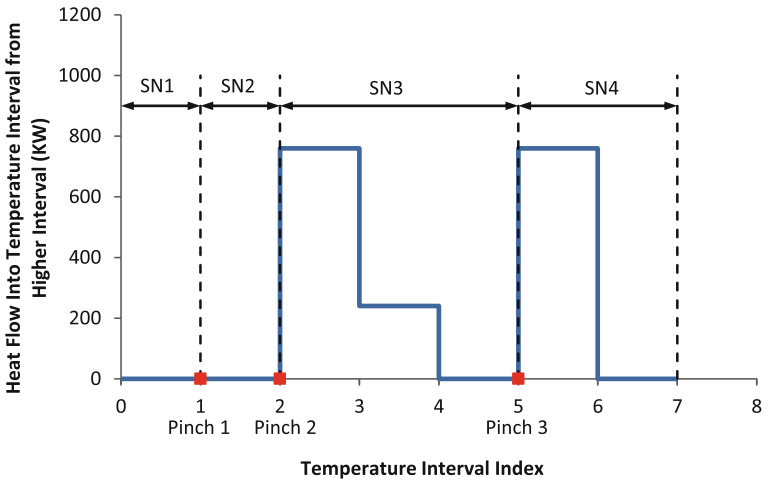


Fig. 11.33 Pinch points and subnetwork intervals are shown for the problem of Example 2.4 with the targeting solution of Table 11.15. Pinch points are indicated by the *squares* on the horizontal axis and the subnetworks (*SN1* through *SN4*) contain the temperature intervals between the pinch points

Table 11.17 Subnetworks, temperature intervals, and stream availability in each subnetwork for the problem discussed in Example 2.4

Subnetwork s	Temp. Int. index, k	Start temp. T_{k-1} (°C)	End temp. T_k (°C)	Hot streams $\overline{H}^s$	Cold streams $\overline{C}^s$
1	1	30	40	$H2$	$C1, CU1$
2	2	40	60	$H2$	$C1, CU2$
3	3	60	70	$H1, H2, HU1$	$C1, C2$
	4	70	80		
	5	80	90		
4	6	90	100	$H2, HU2$	$C1, C2$
	7	100	120		

pinch points give 4 subnetworks. The subnetworks along with their hot and cold stream sets are shown in Table 11.17.

The heat supply and demand in each subnetwork and temperature interval is shown in Table 11.18. *In this table, we set the supply and demand for stream i to 0 if $i \notin \overline{H}^s_k \cup \overline{C}^s_k$.*

3.4.3 Subnetwork Matching Model

Our HEN design method is based on the network decomposition approach explained in Sect. 3.4.2. In this approach, we will solve each of the HEN's subnetworks individually then combine the results to come up with a complete HEN design. Our goal in this section is to model the stream matching problem for a generic subnetwork

Table 11.18 Supply and demand for each stream of every temperature interval in the subnetworks for the problem discussed in Example 2.4

Subnetwork s	Temp. Int. index, k	Hot stream supply, S_{ik} (kW)				Cold stream demand, D_{ik} (kW)			
		$H1$	$H2$	$HU1$	$HU2$	$C1$	$C2$	$CU1$	$CU2$
1	1	0	400	0	0	360	0	40	0
2	2	0	800	0	0	720	0	0	80
3	3	0	400	0	0	360	800	0	0
	4	1000	400	280	0	360	800	0	0
	5	1000	400	0	0	360	800	0	0
4	6	0	400	0	0	360	800	0	0
	7	0	800	0	680	720	0	0	0

$s \in SN$, then use this model to design all the subnetworks. Our design objective is to **minimize the number of matches between hot and cold streams**. We will model this by considering all possible matches in the subnetwork then selecting the minimum number of matches that gives a feasible solution. Let y_{ij} be a binary decision variable for matching a hot stream i to a cold stream j (remember that hot utilities are considered hot streams and cold utilities are considered cold streams). Then

$$y_{ij} = \begin{cases} 1 & \text{, if streams } i \text{ and } j \text{ are matched} \\ 0 & \text{, otherwise} \end{cases}$$

for streams $i \in \overline{H}^s$ and $j \in \overline{C}^s$. The objective is to minimize the number of heat exchangers given by

$$\sum_{i \in \overline{H}^s} \sum_{j \in \overline{C}^s} y_{ij}.$$

Notice that the energy balance constraints (11.18) guarantees that the total supply of heat equal to the total heat demand $\left(\sum_{i \in \overline{H}} \sum_{k=1}^{N} S_{ik} = \sum_{j \in \overline{C}} \sum_{k=1}^{N} D_{jk}\right)$, which is a required condition for the matching problem.

Hot Stream Heat Flow Balance Constraints The energy targeting problem gives the aggregate heat flow between sets of streams within temperature intervals, but does not give specific stream matches, which is what we seek in the stream matching model. For example, q_{ik} in the energy targeting problem tells us how much utility of type i is used in temperature interval k, but gives no information on the set of streams for which it is used. Similarly, r_k gives the amount heat flowing from hot streams and utilities with temperatures $\geq T_k$ to cold streams and utilities with temperatures $\leq T_k$ without specifying the stream matches. In order to get specific stream matches, we need to keep track of the heat sources and sinks. Tracking heat from a source (hot stream) to all the destinations (cold streams) requires having an energy balance constraint for every hot stream. For example, Fig. 11.34 shows the heat flows from hot stream $H1$ to the temperature intervals and cold streams in subnetwork 3 for the problem of Example 2.4 using the solution given in Table 11.18.

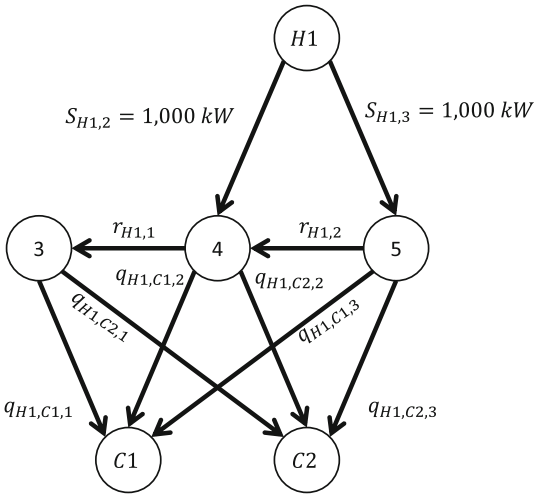


Fig. 11.34 Heat flow from source ($H1$) to sinks ($C1$ and $C2$) in subnetwork 3 of the problem of Example 2.4, for the solution given in Table 11.18. The heat flows leaving node $H1$ are the supply for heat in temperature intervals 2 and 3 given in Table 11.18. The three temperature interval nodes are shown in the middle row of nodes. The flows between the temperature intervals are residual heat flows for stream $H1$. The two demand nodes (cold stream nodes $C1$ and $C2$) are shown in the *bottom* of the figure. The flow from a temperature interval to a cold streams is the heat provided by segments of hot stream $H1$ in the given or higher temperature intervals to the cold stream

For a given temperature interval k and a hot stream i with available heat in interval k, the heat flow balance equation is

$$\begin{aligned} &S_{ik} + (\text{Heat from stream } i \text{ in temperature intervals} > k) \\ &= (\text{Heat used by cold streams in interval } k \text{ from stream } i) \\ &+ (\text{Heat sent to temperature intervals} < k \text{ from stream } i). \end{aligned}$$

The major difference between this heat balance equation and the one in Sect. 3.3 is that we are tracing the heat flow from stream i alone here, whereas all heat flows were lumped into aggregate heat flow variables in the energy targeting problem. To account for this in the formulation, we will add an extra index to the heat flow decision variables q and the residual heat flow decision variables r from the energy targeting problem. The two heat flow decision variables for the stream matching problem are

$q_{ijk} =$ the amount of heat flowing from all segments of hot stream i in temperature intervals $\geq k$ to the segment of cold stream j in the temperature interval k.

$r_{ik} =$ the heat supplied by hot stream or hot utility i with temperature $\geq T_k$ to cold streams or utilities with temperatures $\leq T_k$.

We can now express the second term in the LHS of the heat flow balance equation as r_{ik} and the last term in the equation as $r_{i,k-1}$. In order to express the total heat used

Table 11.19 Hot streams $\overline{H}_k^s$ and cold streams $\overline{C}_k^s$ in each subnetwork and temperature interval for Example 2.4's problem using the targeting results in Table 11.18

Subnetwork s	Temp. Int. index, k	Hot streams $\overline{H}_k^s$	Cold streams $\overline{C}_k^s$
1	1	$H2$	$C1, CU1$
2	2	$H2$	$C1, CU2$
3	3	$H1, H2, HU1$	$C1, C2$
	4	$H1, H2, HU1$	$C1, C2$
	5	$H1, H2$	$C1, C2$
4	6	$H2, HU2$	$C1, C2$
	7	$H2, HU2$	$C1$

by cold streams in interval k we first define the set of cold streams in the subnetwork that should be considered in intervals k as

$$\overline{C}_k^s = \left\{ j \in \overline{C}^s : D_{jk} > 0 \right\}.$$

The first term on the RHS of the balance equation is therefore $\sum_{j \in \overline{C}_k^s} q_{ijk}$. By combining the terms, we get the following heat flow balance constraint for hot stream i in temperature interval k (provided that stream i can be a source of heat in temperature interval k)

$$r_{i,k-1} - r_{ik} + \sum_{j \in \overline{C}_k^s} q_{ijk} = S_{ik}.$$

As before, we set $r_{i0} = r_{iN_s} = 0 \; \forall \; s \in SN$ and $i \in \overline{H}^s$. Figure 11.35 shows a graphical representation of the hot stream heat flow balance. Notice that a hot stream i can provide heat to temperature interval k if it has available heat in any of the intervals $k, k+1, \cdots, P_s$. So the hot stream balance constraint must hold for all hot streams with available heat in intervals $k, k+1, \cdots, P_s$. Therefore, the set of hot streams we need to consider for temperature interval k is

$$\overline{H}_k^s = \left\{ i \in \overline{H}^s : \sum_{l=k}^{P_s} S_{il} > 0 \right\}.$$

The sets $\overline{C}_k^s$ and $\overline{H}_k^s$ for all subnetworks and all temperature intervals are shown in Table 11.19 for the problem of Example 2.4 using the targeting results in Table 11.18.

Cold Stream Heat Flow Balance Constraints To ensure that the heat flow to cold streams exactly equals their heat demand, we need to enforce that the total heat flow from all hot streams amounts to the cold stream's demand, which is formulated as

$$\sum_{i \in \overline{H}_k^s} q_{ijk} = D_{jk} \;\; \text{for all} \;\; k = P_{s-1} + 1, \cdots, P_s, \;\; j \in \overline{C}_k^s.$$

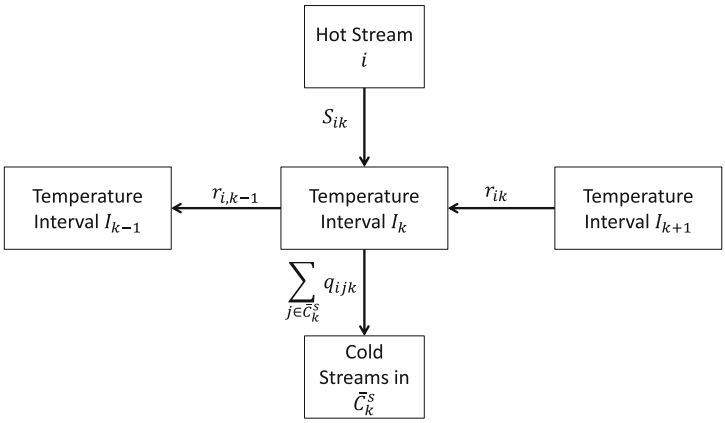


Fig. 11.35 Heat flow balance from a stream segment of hot stream i in temperature interval k. Heat from stream i in this temperature interval, S_{ik}, is combined with heat from stream i in higher temperature intervals, r_{ik}, and used to heat cold streams in this temperature intervals. The flow to cold stream j from segments of hot stream i in temperature intervals k and higher is given by q_{ijk}, and the total heat sent to cold streams in this temperature interval is $\sum_{j\in\overline{C}_k^s} q_{ijk}$. The remaining unused heat $r_{i,k-1}$ is sent to cold stream segments in lower temperature intervals through

Heat Transfer Bound Constraints If hot stream i were to be matched with cold stream j, then the amount of heat that can be transferred between the two streams cannot exceed the hot stream's total heat supply or the cold stream's total heat demand. Therefore, the maximum possible heat transfer between the two streams is

$$U_{ij} = \min \left\{ \sum_{k=1+P_{s-1}}^{P_s} S_{ik}, \sum_{k=1+P_{s-1}}^{P_s} D_{jk} \right\}.$$

In addition to the upper heat transfer bound U_{ij}, we need to ensure that no heat is transferred between the two streams if they are not matched in a heat exchanger, which holds when $y_{ij} = 0$. This can be done by setting $U_{ij}y_{ij}$ as the heat transfer bound between a hot stream i and a cold stream j. Accordingly, the heat transfer bound constraints are

$$\sum_{k=1+P_{s-1}}^{P_s} q_{ijk} \le U_{ij}y_{ij} \quad \text{for all } i \in \overline{H}^s, j \in \overline{C}^s.$$

The notation introduced in this section are listed in Table 11.20.

The Stream Matching Mixed Integer Programming Formulation Now that we have introduced the required notation and derived the problem's model, we can formulate the matching problem for subnetwork $s \in SN$ as the following mixed

Table 11.20 Summary of the sets, input data, and variables used in the stream matching problem

Set	*Definition*
$\overline{H}$	$= H \cup HU$ Set of hot streams and hot utilities
$\overline{C}$	$= C \cup CU$ Set of cold streams and cold utilities
$\overline{H}^s$	$= \left\{ i \in \overline{H} : \sum_{k=1}^{N_s} S_{ik} > 0 \right\}$ Set of hot streams and utilities present in subnetwork $s \in SN$
$\overline{C}^s$	$= \left\{ i \in \overline{C} : \sum_{k=1}^{N_s} D_{ik} > 0 \right\}$ Set of cold streams and utilities present in subnetwork $s \in SN$
$\overline{H}_k^s$	$= \left\{ i \in \overline{H}^s : \sum_{l=k}^{N_s} S_{il} > 0 \right\}$ Hot streams present in the intervals $k, \cdots, N_s$ in subnetwork s
$\overline{C}_k^s$	$= \left\{ i \in \overline{C}^s : D_{ik} > 0 \right\}$ Cold streams present in the interval k in subnetwork s
P	$= (k \in \{1, \cdots, N-1\} : r_k = 0)$, where r_k is the solution from the targeting problem Ordered set of pinch interval indices
SN	$= \{1, \cdots, N_P + 1\}$ Set of subnetworks
Parameter	*Definition*
N_P	$= \|P\|$. Number of of pinch temperatures
N_s	Number of temperature intervals in subnetwork $s \in SN$
P_0	$= 0$. Constant used to represent the initial temperature index in subnetwork 1
P_{N_P+1}	$= N$. Constant used to represent the terminal temperature index in subnetwork $N_P + 1$
T_k^P	kth pinch temperature, where $k = 1, \cdots, N_P$ T_k^P is ordered such that $T_i^P < T_j^P$ for $i < j$
U_{ij}	$= \min \left\{ \sum_{k=1}^{N_s} S_{ik}, \sum_{k=1}^{N_s} D_{jk} \right\}$ Maximum heat that can be transferred from hot sink i to cold sink j
r_{i0}	$=0$. Starting residual heat for hot stream or utility i
r_{iN}	$=0$. Ending residual heat for hot stream or utility i
a_{ij}	Cost of matching streams i and j (in \$ per match)
Variable	*Definition*
y_{ij}	$\in \{0, 1\}$. Stream matching decision variable 1 if streams i and j are matched and 0 if not
q_{ijk}	≥ 0. Heat flowing from all segments of hot stream or hot utility i in temperature intervals $\geq k$ to the segment of cold stream or cold utility j in temperature interval k
r_{ik}	≥ 0. Heat supplied by hot stream or hot utility i with temperature $\geq T_k$ to cold streams or cold utilities with temperatures $\leq T_k$

integer program.

$$\min_{y,q,r} \sum_{i \in \overline{H}} \sum_{j \in \overline{C}} a_{ij} y_{ij} \tag{11.22}$$

$$\text{s. to } r_{i(k-1)} - r_{ik} + \sum_{j \in \overline{C}_k^s} q_{ijk} = S_{ik} \ \forall \ k \in \{P_{s-1}+1, \cdots, P_s\}, i \in \overline{H}_k^s \tag{11.23}$$

$$\sum_{i \in \overline{H}_k^s} q_{ijk} = D_{jk} \ \forall \ k \in \{P_{s-1}+1, \cdots, P_s\}, j \in \overline{C}_k^s \tag{11.24}$$

$$\sum_{k=1+P_{s-1}}^{P_s} q_{ijk} - U_{ij} y_{ij} \leq 0 \ \forall \ i \in \overline{H}^s, j \in \overline{C}^s \tag{11.25}$$

$$r_{ik} \geq 0 \ \forall \ k \in \{P_{s-1}+1, \cdots, P_s\}, i \in \overline{H}^s \tag{11.26}$$

$$r_{i0} = r_{iN_s} = 0 \ \forall \ i \in \overline{H}^s \tag{11.27}$$

$$q_{ijk} \geq 0 \ \forall \ k \in \{P_{s-1}+1, \cdots, P_s\}, i \in \overline{H}_k^s, j \in \overline{C}_k^s \tag{11.28}$$

$$y_{ij} \in \{0, 1\} \ \forall \ i \in \overline{H}^s, j \in \overline{C}^s. \tag{11.29}$$

HEN designers sometimes have stream matching preferences. For example, some matches may not be desirable because of the long distance between streams, expensive material of construction that is needed for a heat exchanger for some of the matches, or because of some other considerations that may influence the decision. We have included the cost coefficient a_{ij} (in \$ per match) in the objective Eq. (11.22) to account for such matching preferences. The HEN final design is obtained by solving the mixed integer program (11.22)–(11.29) for each subnetwork $s \in SN$ then combining their solutions to form the overall HEN. A solved matching HEN problem is shown in Example 3.4.3. Although the solution to the matching problem gives the set of heat exchangers in the network, some dots need to be connected before the final HEN design is attained. We will show how to get a complete HEN design from the matching solution in Sect. 3.4.4 next.

Example 3.7 (**Stream matching**) *To find the HEN design for the problem of Example* 2.4 *we first solve the matching problem* (11.22)–(11.29) *for the four subnetworks given in Table* 11.18. *The solution to each of the subnetwork problems is given in Table* 11.21.

A heat exchanger device exists in the final design between two streams whenever the y variable is 1. In this problem, the solution gives 2 heat exchangers in subnetwork 1, 2 in subnetwork 2, 4 in subnetwork 3, and 3 in subnetwork 4, for a total of 11 heat exchangers. It is worth noting that the information in Table 11.21 *is sufficient to produce a HEN design, which will be shown in Sect.* 3.4.4.

3.4.4 HEN Design

A complete HEN design description requires determining the set of heat exchangers in the network, the quantity of heat (in kW) transferred in the heat exchangers, the pairs of streams (process or utility streams) between which heat is transferred in the heat exchangers, the temperatures (in °C) at which these streams enter and exit the heat

Table 11.21 Matching problem solution to each of the subnetworks given in Table 11.18

Subnetwork 1		Subnetwork 2		Subnetwork 3				Subnetwork 4	
Variable	Value	Variable	Value	Variable	Value	Variable	Value	Variable	Value
$y_{H2,C1}$	1	$y_{H2,C1}$	1	$y_{H1,C1}$	0	$q_{H2,C1,4}$	360	$y_{H2,C1}$	1
$y_{H2,CU1}$	1	$y_{H2,CU2}$	1	$y_{H1,C2}$	1	$q_{H2,C2,4}$	80	$y_{H2,C2}$	1
$q_{H2,C1,1}$	360	$q_{H2,C1,2}$	720	$y_{H2,C1}$	1	$q_{HU1,C1,4}$	0	$y_{HU2,C1}$	0
$q_{H2,CU1,1}$	40	$q_{H2,CU2,2}$	80	$y_{H2,C2}$	1	$q_{HU1,C2,4}$	0	$y_{HU2,C2}$	1
				$y_{HU1,C1}$	1	$q_{H1,C1,5}$	0	$q_{H2,C1,6}$	360
				$y_{HU1,C2}$	0	$q_{H1,C2,5}$	800	$q_{H2,C2,6}$	120
				$q_{H1,C1,3}$	0	$q_{H2,C1,5}$	360	$q_{HU2,C1,6}$	0
				$q_{H1,C2,3}$	480	$q_{H2,C2,5}$	0	$q_{HU2,C2,6}$	680
				$q_{H2,C1,3}$	80	$r_{H1,3}$	0	$q_{H2,C1,7}$	720
				$q_{H2,C2,3}$	320	$r_{H1,4}$	480	$q_{HU2,C1,7}$	0
				$q_{HU1,C1,3}$	280	$r_{H2,3}$	80	$r_{H2,6}$	80
				$q_{HU1,C2,3}$	0	$r_{H2,4}$	320	$r_{HU2,6}$	680
				$q_{H1,C1,4}$	0	$r_{HU1,3}$	280		
				$q_{H1,C2,4}$	720				

exchangers, and the fluid flows (in Mg/h) of the streams in the heat exchangers. A solution to the matching problem (11.22)–(11.29) gives the stream pairs to be matched in the HEN design (y variables) as well as the heat transfer quantities (by summing the q over all the temperature intervals for the two streams). If we know the inlet and outlet temperatures to heat exchangers, then we can use Eq. (11.1) to determine the heat capacity flowrate CP in kW/°C for each stream, which can be used to determine the fluid flowrate f in Mg/h by using the formula $f = 3.6 \cdot CP/c_P$, where c_P is the stream's specific heat capacity in J/g°C as discussed in Sect. 2.1.2. Depending on the stream flows, the matching configuration may result in series or parallel (using stream splitting as in Sect. 3.2.2) heat exchanger configurations.

After getting the stream matching solution, the remaining challenge to getting a complete HEN design is to determine the heat exchanger inlet and outlet temperatures for the two streams. Fortunately, the temperature ranges can be inferred from the heat flow and residual variables q and r. For example, if hot stream i were matched with cold stream j in subnetwork s, then $q_{ijk} > 0$ for some temperature interval k in subnetwork s. If we find the largest k in subnetwork s for which $q_{ijk} > 0$, then we can deduce that the hot stream's inlet temperature to the heat exchanger must be between the two end point temperatures of interval k, and the cold stream's outlet temperature from the heat exchanger is bounded by the hot stream's inlet temperature. If k is the smallest interval index in subnetwork s with $q_{ijk} > 0$ then the cold stream's inlet temperature to the heat exchanger must be between interval k's two end point temperatures, but the hot stream's outlet temperature range cannot be inferred without considering its residual heat flow from higher temperature intervals to k.

In some cases, the exact heat exchanger inlet and outlet temperatures can be determined from the solution to the stream matching problem which is sufficient to give a unique HEN design. However, there may be several valid heat exchanger inlet and outlet temperature assignments within the acceptable temperature ranges, which can result in multiple HEN designs for a given stream matching solution. There are many heuristics that have been developed to get final HEN designs, some starting from the stream matching solution while others start from the targeting solution, such as the pinch design method (see for example [9]). In this chapter, we will design the HEN starting from the matching solution and determining the heat exchanger inlet and outlet temperatures for the most constrained non-utility stream (the one with the least matches), then do it for next most constrained non-utility stream and so on until all heat exchanger inlet and outlet temperatures are determined. The following subroutine is carried out for every subnetwork to get the subnetwork's HEN design.

HEN design in subnetwork $s \in SN$:

1. **Determine the set of heat exchangers:** The set of heat exchangers in a subnetwork is given by the non-zero y_{ij} variables. If $y_{ij} = 1$, then the two streams i and j are matched in a heat exchanger.
2. **Determine the heat duty of each heat exchanger:** The duty of a HE h that matches streams i and j in subnetwork s is the total heat flow between the two streams in the matching solution, given by $Q_h = \sum_{k=1+P_{s-1}}^{P_s} q_{ijk}$.
3. **Choose a process stream:** Choose a process stream i with the least number of HE matches that has not been already chosen.

4. **Find the HE inlet and outlet temperature ranges:** For every heat exchanger h matching i, we seek to find the inlet temperature T_{i1}^h and outlet temperature T_{i2}^h of i into and out of h. We will consider the two cases when i is a hot stream and when i is a cold stream.
   a) If i is a hot stream: If the HE h matches i to j then i enters h in temperature interval $b = \max\{k \in \{P_{s-1}+1, \cdots, P_s\} : q_{ijk} > 0\}$. Stream i sends heat to stream j through the heat flow variables q_{ijk}, so i leaves h in an interval no less than $a = \min\{k \in \{P_{s-1}+1, \cdots, P_s\} : q_{ijk} > 0\}$. The inlet temperature $T_{i1}^h \in [T_{b-1}, T_b]$ and the outlet temperature $T_{i2}^h \in [T_{a-1}, T_{i1}^h]$. If stream j has already been processed then T_{j1}^h and T_{j2}^h would be known, so we also restrict $T_{i1}^h \geq T_{j2}^h$ and $T_{i2}^h \geq T_{j1}^h$.
   b) If i is a cold stream: If the HE h matches i to j then i enters h in temperature interval $a = \min\{k \in \{P_{s-1}+1, \cdots, P_s\} : q_{jik} > 0\}$ and leaves in a temperature interval no greater than $b = \max\{k \in \{P_{s-1}+1, \cdots, P_s\} : q_{jik} > 0\}$. Stream i's inlet temperature to h is $T_{i1}^h \in [T_{a-1}, T_a]$ and the outlet temperature is $T_{i2}^h \in [T_{i1}^h, T_b]$. If j is already handled then we must restrict $T_{i1}^h \leq T_{j2}^h$ and $T_{i2}^h \leq T_{j1}^h$ to ensure that heat flows from hot streams to cold streams.
5. **Find the flows and temperatures:** We find in this step the exact inlet and outlet temperature and flow of stream i through every HE h that matches i. The heat transferred to or from stream i through HE h is $Q_h = \Delta T_i^h CP_i^h$, where Q_h was found in step 2, ΔT_i^h is the absolute difference between the inlet and outlet temperature of stream i in HE h, and $CP_i^h \in [0, CP_i]$ is the CP value used for stream i in HE h. Notice that if i's entire flow goes through h then $CP_i^h = CP_i$, but if $CP_i^h < CP_i$ then only part of i's flow goes through h, and so i is split into multiple parallel streams where one branch with partial flow goes through h (refer to Sect. 3.2.2). We initially set $CP_i^h = CP_i$ and find T_{i1}^h and T_{i2}^h within the range given by step 4 that satisfies $|T_{i1}^h - T_{i2}^h| = Q_h/CP_i^h$. If we can find such a solution, then we are done with this HE. If such a solution does not exist, then using feasible T_{i1}^h and T_{i2}^h values, we set $CP_i^h = Q_h/|T_{i1}^h - T_{i2}^h|$ to attain a feasible solution. In this case, we split stream i (refer to Sect. 3.2.2) with the flow fraction CP_i^h/CP_i into HE h and $1 - CP/_i^h/CP_i$ into other branches (parallel heat exchanger). At this stage we have found the inlet and outlet temperatures T_{i1}^h and T_{i2}^h of stream i for HE h as well as i's flow through h. We can repeat this process for all heat exchangers matching i then remove the stream i from the set of streams to be processed in later iterations of this algorithm.
6. **Terminal condition:** If all streams have been considered then terminate, otherwise return to step 3.

One thing to notice about this algorithm is that it may not give a unique HEN design because there may be a range of feasible T_{i1}^h and T_{i2}^h to choose from in step 5. Also note that we need not solve for the utility stream temperatures because they are assumed to be constant throughout the exchanger. Example 3.8 illustrates this HEN design process.

Example 3.8 (**HEN design**) *In this example, we will find the final HEN design of the 4 stream and 4 utility problem presented in Example 2.4. We have found a stream matching solution for this problem in Example 3.7, which shows that the HEN has*

Table 11.22 Stream information for each subnetwork given by the targeting solution in Table 11.18. The table shows the stream name, starting temperature, target temperature, heat capacity flow rate, and the streams duty (heat supply for hot streams and heat demand for cold streams)

Subnetwork	Stream	T_s (°C)	T_t (°C)	CP (kW/°C)	Duty (kW)
1	$H2$	40	30	40	400
	$C1$	30	40	36	360
	$CU1$	30	30	NA	40
2	$H2$	60	40	40	800
	$C1$	40	60	36	720
	$CU2$	40	40	NA	80
3	$H1$	90	70	100	2000
	$H2$	90	60	40	1200
	$HU1$	80	80	NA	280
	$C1$	60	90	36	1080
	$C2$	60	90	80	2400
4	$H2$	120	90	40	1200
	$HU2$	120	120	NA	680
	$C1$	90	120	36	1080
	$C2$	90	100	80	800

11 heat exchangers. We will name these heat exchangers E1–E11, starting with E1 and E2 for the two heat exchangers in subnetwork 1, E3 and E4 for the two heat exchangers in subnetwork 2, E5-E8 for the four heat exchangers in subnetwork 3, and E9-E11 for the three heat exchangers in subnetwork 4. The stream information for each subnetwork is shown in Table 11.22. We will first find a HEN design for each of the subnetworks independently then combine these results to get the overall HEN design.

Subnetwork 1 *Subnetwork 1 has 2 heat exchangers: E1 and E2. $H2$ supplies 360 kW of heat to $C1$ in E1 and 40 kW of heat to $CU1$ in E2. Cold stream $C1$ is the most constrained non-utility stream since it is matched only once with $H2$. Therefore, its temperature is increased from 30°C to 40°C in E1. If $H2$ were to exchange heat in E2 before E1 then its starting temperature in E1 would be below 40°C, making it impossible to raise $C1$'s temperature to 40°C in E1. Therefore, $H2$ starts exchanging heat with $C1$ in E1 and the outlet goes into E2 to be cooled by $CU1$. Using Eq. (11.1), $H2$'s output temperature from E1 can be calculated to be $40 - 360/40 = 31$°C, which also happens to be $H2$'s inlet temperature to E2. Subnetwork 1's HEN design is shown in Fig. 11.36*[9].

Subnetwork 2 *Subnetwork 2 has 2 heat exchangers: E3 and E4. In E3, 720 kW of heat is transferred from $H2$ to $C1$, while 80 kW is transferred from $H2$ to $CU2$ in*

[9] For all the HEN design schematics (Figs. 11.36–11.46), the following apply: The thin lines in the drawings represent cold streams while the thick lines represent hot streams. Each stream segment has its temperature indicated (in °C) with a circle superscript, and each heat exchanger has its duty value (in kW) indicated on its bottom right. For utility heat exchangers, the utility is indicated on the heat exchanger's top right. The stream names are shown in boxes at the start and end points of each stream.

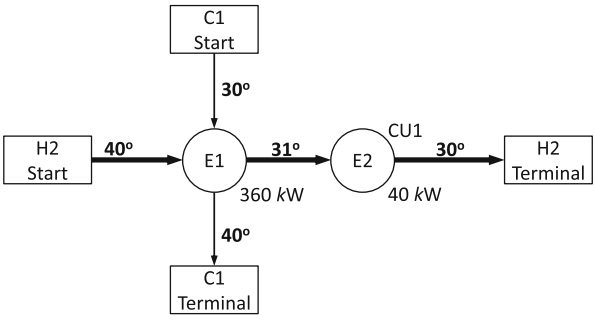


Fig. 11.36 HEN design schematic for subnetwork 1 in the four stream and four utility problem of Example 2.4. This subnetwork has the hot stream $H2$, the cold streams $C1$, the cold utility $CU1$, and the two heat exchangers *E1* and *E2*. (Refer to footnote 9)

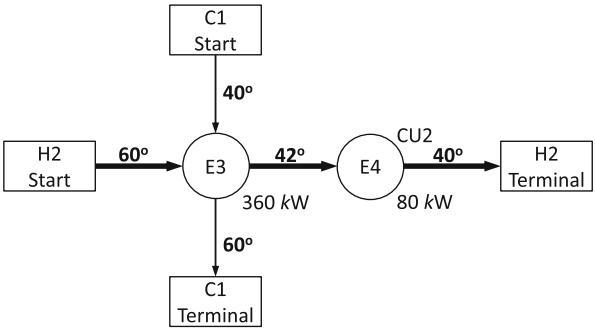


Fig. 11.37 HEN design schematic for subnetwork 2 in the four stream and four utility problem of Example 2.4. This subnetwork has the hot stream $H2$, the cold stream $C2$, the cold utility $CU2$, and the two heat exchangers *E3* and *E4*. (Refer to footnote 9)

E4. The most constrained non-utility stream is $C1$ *since it appears in a single heat exchanger. Therefore,* $C1$ *must raise its temperature from* 40 to 60°C *using heat from* $H2$ *in E3.* $H2$ *starts off with* 60°C *in subnetwork 2, and if it were to exchange heat with* $CU2$ *in E4 before* $C1$ *in E3 then it would have a starting temperature below* 60°C *in E3, making it impossible to raise* $C1$*'s temperature to* 60°C*. Therefore,* $H2$ *first goes through E3 then through E4. From Eq. (11.1),* $H2$*'s outlet temperature from E3 and inlet temperature to E4 is* $60 - 720/40 = 42$°C*. Figure 11.37 shows the HEN design for subnetwork 2.*

Subnetwork 3 *This subnetwork has four heat exchangers: E5, E6, E7, and E8. The heat transferred from* $H1$ *to* $C2$ *in E5 is* 2000 *k*W, 800 *k*W *from* $H2$ *to* $C1$ *in E6,* 400 *k*W *from* $H2$ *to* $C2$ *in E7, and* 280 *k*W *from* $HU1$ *to* $C1$ *in E8.* $H1$ *is matched only once with* $C2$ *in E5, so it enters E5 at* 90°C *and leaves at its target temperature of* 70°C*. At this stage, each non-utility stream has 2 degrees of freedom (i.e., can exchange heat in two heat exchangers), so there may be multiple solutions. However, notice that* $q_{H2,C2,5} = 0$ *and* $q_{H2,C2,4} > 0$*, which means that the inlet temperature of* $H2$ *in E7 is between* 70 *and* 80°C*, implying that* $C2$ *cannot be heated to its target* 90°C *in E7. Therefore,* $C2$ *must reach its target* 90°C *in E5, which makes its outlet temperature from E5* 90°C*. Therefore,* $C2$ *goes through E7 then E5.* $C2$*'s starting temperature in E5 is* $90 - 2000/80 = 65$°C *(using Eq. (11.1)), which is also its outlet temperature from E7, and its inlet temperature to E7 is* 60°C*. Now consider stream* $C1$*. Notice that* $q_{HU1,C1,4} = 0$ *and* $q_{HU1,C1,3} > 0$*, which means that* $HU1$

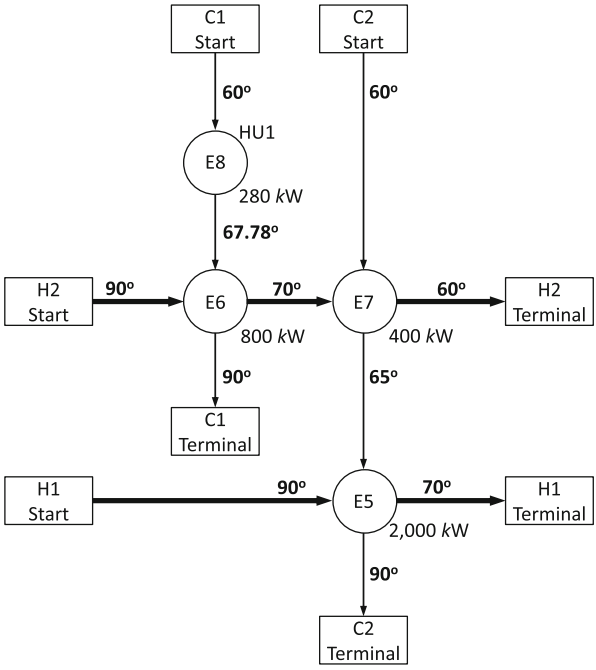


Fig. 11.38 HEN design schematic for subnetwork 3 in the four stream and four utility problem of Example 2.4. This subnetwork has the hot streams $H1$ and $H2$, the cold streams $C1$ and $C2$, the hot utility $HU1$, and the four heat exchangers *E5–E8*. (Refer to footnote 9)

exchanges heat at a temperature $\leq 70°C$. *Therefore,* $C1$ *cannot reach its target* $90°C$ *in E8, so it must reach its target in E6. This means that* $C1$ *first goes through E8 then through E6.* $C1$*'s outlet temperature from E6 is* $90°C$ *and its inlet temperature is calculated using Eq.* (11.1) *to be* $90 - 800/36 \approx 67.78°C$. $C1$*'s inlet temperature to E8 is* $60°C$ *and its outlet (using Eq.* (11.1)*) is* $67.78°C$. *Notice that* $C1$ *cannot cool* $H2$ *down to its target* $60°C$ *in E6, so* $H2$ *must reach its target* $60°C$ *in E7. This means that* $H2$ *goes first through E6 then through E7.* $H2$*'s inlet to E6 is* $90°C$ *and its outlet is* $90 - 800/40 = 70°C$. *Its inlet to E7 is* $70°C$ *and its outlet is* $60°C$. *Figure* 11.38 *shows an illustration of subnetwork 3's HEN.*

Subnetwork 4 *Subnetwork 4 has the three heat exchangers E9, E10, and E11.* $1,080$ *kW of heat is transferred from* $H2$ *to* $C1$ *in E9,* 120 *kW from* $H2$ to $C2$ *in E10, and* 680 *kW from* $HU2$ *to* $C2$ *in E11. Since* $C1$ *is only matched once, then it enters E9 with* $90°C$ *and leaves E9 with* $120°C$. *Also, since* $C1$ *leaves E9 with* $120°C$ *then* $H2$ *must enter E9 with* $120°C$. *Therefore,* $H2$ *goes through E9 first then E10. Using Eq.* (11.1), $H2$*'s outlet temperature from E9 is* $120 - 1,080/40 = 93°C$, *which is also its inlet temperature to E10.* $H2$ *leaves E10 with its target temperature* $90°C$. *Since* $H2$ *enters E10 with* $93°C$, *then* $C2$ *cannot be heated to its target* $100°C$ *in E10. Therefore,* $C2$ *must first pass through E10 then E11.* $C2$*'s inlet to E10 is* $90°C$ *and its outlet is calculated from Eq.* (11.1) *to be* $90 + 120/80 = 91.5°C$. $C2$ *then enters E11 with* $91.5°C$ *and leaves with its target temperature* $100°C$. *A schematic of subnetwork 4's HEN is shown in Fig.* 11.39.

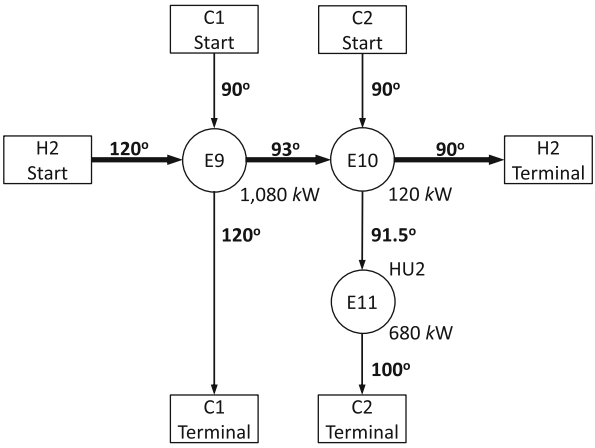


Fig. 11.39 HEN design schematic for subnetwork four in the four stream and four utility problem of Example 2.4. This subnetwork has the hot stream $H2$, the cold streams $C1$ and $C2$, the hot utility $HU2$, and the heat exchangers *E9–E11*. (Refer to footnote 9)

The list of all the heat exchangers and their attributes from the four subnetworks is shown in Table 11.23. To get the final HEN design, we first need to undo the temperature shifting to the hot stream and cold utility data (look at Sect. 3.2.1). This requires adding $\Delta T_{\min} = 10°\text{C}$ to all hot streams and cold utility temperatures. The resulting final HEN schematic for this problem (after re-shifting the temperatures) is shown in Fig. 11.40.

The solution in Example 3.4.4 was found without having to split streams as explained in Sect. 3.2.2. In practice however, stream splitting may be required to attain feasible matches. We will see how stream splitting is used in the solved refinery example.

3.5 HEN Design Summary

The HEN grassroots design problem statement is given in Sect. 3.1.2. We provide in this section a summary of the input data and the steps required to solve this problem.

3.5.1 Input Data

The input datasets for this problem are as follows:

- **Sets:** Hot process streams H, cold process streams C, hot utilities HU, and cold utilities CU.
- **Temperatures (in °C):** Starting temperatures T_i^s for $i \in H \cup C$, target temperatures T_i^t for $i \in H \cup C$, utility temperatures T_i^u for $i \in HU \cup CU$, and minimum approach temperature $\Delta T_{\min}$.
- **Heat capacity flowrates (in kW/°C):** CP_i for $i \in H \cup C$.
- **Utility costs (in $\$/MWh$):** c_i for $i \in HU \cup CU$.

Table 11.23 Heat exchangers for the problem of Example 2.4. The table shows the heat exchangers E1–E11 in each of the problem's 4 subnetworks. Each column in the table corresponds to a heat exchanger and each row corresponds to a HE attribute

HE Attribute	Subnetwork 1		Subnetwork 2		Subnetwork 3				Subnetwork 4		
Name	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11
Type	P2P	Util	P2P	Util	P2P	P2P	P2P	Util	P2P	P2P	Util
Hot stream	H2	H2	H2	H2	H1	H2	H2	HU1	H2	H2	HU2
Cold stream	C1	CU1	C1	CU2	C2	C1	C2	C1	C1	C2	C2
Duty (kW)	360	40	720	80	2000	800	400	280	1080	120	680
Hot T_{in} (°C)	40	31	60	42	90	90	70	70	120	93	100
Hot T_{out} (°C)	31	30	42	40	70	70	60	70	93	90	100
Cold T_{in} (°C)	30	30	40	40	65	67.78	60	60	90	90	91.5
Cold T_{out} (°C)	40	30	60	40	90	90	65	67.78	120	91.5	100
Hot CP (kW/°C)	40	40	40	40	100	40	40	NA	40	40	NA
Cold CP (kW/°C)	36	NA	36	NA	80	36	80	36	36	80	80
Hot flow (Mg/h)	100	100	100	100	140	100	100	NA	100	100	NA
Cold flow (Mg/h)	100	NA	100	NA	40	100	40	100	100	40	40

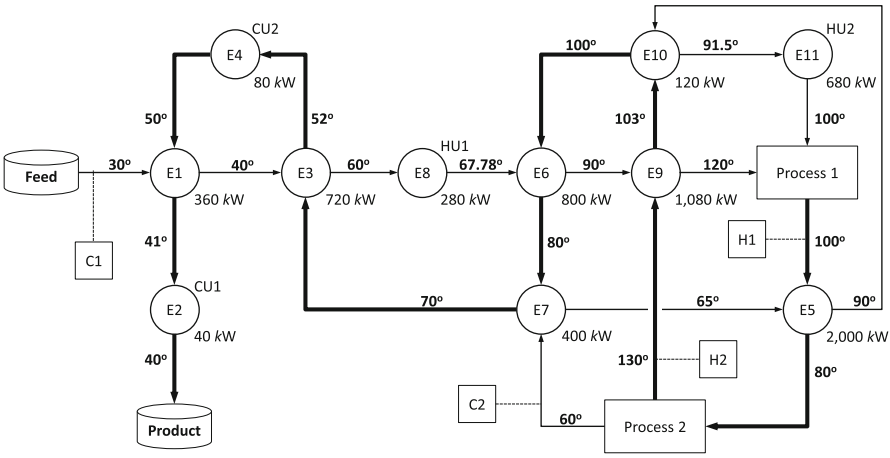


Fig. 11.40 Final HEN design schematic for the four stream and four utility problem of Example 2.4. The stream names ($H1$, $H2$, $C1$, and $C2$) are indicated in *boxes* at the starting point of each stream. (Refer to footnote 9)

3.5.2 Network Design Procedure

The HEN design procedure described throughout Sect. 3 to solve this problem is summarized in the following steps:

1. **Temperature shifting:** Subtract $\Delta T_{\min}$ from all hot stream stargin and target temperatures (T_i^s and T_i^t for $i \in H$) as well as cold utility temperatures (T_j^u for $j \in CU$). Refer to Sect. 3.2.1 for more information.
2. **Temperature partitioning:** Combine all process stream starting and target temperatures and all utility temperatures in one list $T = \{T_i^s : i \in H \cup C\} \cup \{T_i^t : i \in H \cup C\} \cup \{T_i^u : i \in HU \cup CU\}$. Remove duplicates from T and sort it in ascending order, so that $T = (T_0, T_1, \cdots, T_N)$ where $T_0 < T_1 < \cdots < T_N$. The list of temperature intervals is $(I_1, \cdots, I_N)$ where $I_k = [T_{k-1}, T_k]$, and the list of temperature interval indices is $(1, \cdots, N)$. Refer to Sect. 3.2.4 for more information.
3. **Energy targeting:**
   a) **Find heat supply and demand:** The supply of heat for each process stream $i \in H$ in temperature interval $k \in \{1, \cdots, N\}$ is $S_{ik} = CP_i(T_k - T_{k-1})$, and the heat demand for cold stream $j \in C$ in temperature interval $k \in \{1, \cdots, N\}$ is $D_{ik} = CP_j(T_k - T_{k-1})$.
   b) **Find the process streams in the temperature intervals:** The sets of hot and cold process streams in temperature interval $k \in \{1, \cdots, N\}$ are $H_k = \{i \in H : S_{ik} > 0\}$ and $C_k = \{i \in C : D_{ik} > 0\}$.
   c) **Find the set of utilities in the temperature intervals:** The sets of hot and cold utilities in the temperature interval $k \in \{1, \cdots, N\}$ are $HU_k = \{i \in HU : T_i^u \geq T_k\}$ and $CU_k = \{i \in CU : T_i^u \leq T_{k-1}\}$ respectively.

d) **Solve the energy targeting problem:** Solve the linear program given by (11.17)–(11.21). The solution to this problem gives the variables q_{ik} and r_k for $k \in \{1, \cdots, n\}$ and $i \in HU_k \cup CU_k$. The variable q_{ik} is the utility consumption targets for utility i in temperature interval k while the variable r_k represents the total heat flow from hot streams and hot utilities in temperature intervals $k, k+1, \cdots, N$ to cold stream sand cold utilities in temperature intervals $1, 2, \cdots, k-1$.

Refer to Sect. 3.3 for more information on energy targeting.

4. **Stream matching:**

a) **Combine process and utilities streams:** The combined set of hot streams and hot utilities is $\overline{H} = H \cup HU$ and the combined set of cold streams and cold utilities is $\overline{C} = C \cup CU$. Set the hot stream supply for hot utility $i \in HU$ and the cold stream demand for cold utility $j \in CU$ in temperature interval $k \in \{1, \cdots, N\}$ to the utility targets from the energy targeting solution in step 3d (so $S_{ik} = q_{ik}$ and $S_{jk} = q_{jk}$ respectively). Note that the hot process stream supply and cold process stream demand in each temperature interval is given in step 3a. Refer to Sect. 3.4.1 for more information.

b) **Find the pinch points:** The list of pinch temperature interval indices is $P = (k \in \{1, \cdots, N-1\} : r_k = 0)$, where r_k is the residual heat flow solution from the targeting problem in step 3d. Sort P in ascending order and let $N_P = |P|$ be the number of pinch points. Refer to Sect. 3.4.2 for more information.

c) **Find the subnetworks:** The set of subnetwork indices is $SN = \{1, \cdots, N_P + 1\}$. Let $P_0 \equiv 0$ and $P_{N_P+1} \equiv N$. The set of hot and cold steams in subnetwork $s \in SN$ is $\overline{H}^s = \left\{i \in \overline{H} : \sum_{k=1+P_{s-1}}^{P_s} S_{ik} > 0\right\}$ and $\overline{C}^s = \left\{i \in \overline{C} : \sum_{k=1+P_{s-1}}^{P_s} D_{ik} > 0\right\}$ respectively. The set of hot streams in subnetwork $s \in SN$ in intervals $k, \cdots, P_s$ for some $k \in \left[P_{s-1}+1, P_s\right]$ is $\overline{H}_k^s = \{i \in \overline{H}^s : \sum_{l=k}^{P_s} S_{il} > 0\}$ and the set of cold streams in subnetwork $s \in SN$ in interval k is $\overline{C}_k^s = \left\{i \in \overline{C}^s : D_{ik} > 0\right\}$. Refer to Sects. 3.4.2 and 3.4.3 for more information.

d) **Solve the stream matching problem:** For every subnetwork $s \in SN$, find the heat transfer bounds $U_{ij} = \min\left\{\sum_{k=1+P_{s-1}}^{P_s} S_{ik}, \sum_{k=1+P_{s-1}}^{P_s} D_{jk}\right\}$ for all streams $i \in \overline{H}^s$ and $j \in \overline{C}^s$ then solve the mixed integer program (11.22)–(11.29) to get the matching solutions for each subnetwork. The solution to this problem gives the matches y_{ij} between hot stream i and cold stream j, the heat flows q_{ijk} from segments of hot stream i in temperature intervals $\geq k$ to the segment of cold stream j in temperature interval k, and the residual heat flows r_{ik} from hot stream i in temperature intervals $\geq k$ to cold streams in temperature intervals $< k$. Refer to Sect. 3.4.3 for more information.

e) **Design the HEN:** Follow the algorithm given in Sect. 3.4.4 for every subnetwork to get a complete HEN design.

Refer to Sect. 3.4 for more information on the stream matching problem.

5. **Undo temperature shifting:** Undo the temperature shifting in the first step by adding $\Delta T_{\min}$ to all hot stream temperatures and cold utility temperatures throughout the design. This includes hot stream starting temperatures, target temperatures, inlet temperatures to all heat exchangers, and outlet temperatures from all heat exchangers.

4 Refinery HEN Problem

After going through some heat exchanger background in Sect. 2 and the HEN design procedure in Sect. 3, we return to the problem of designing the crude fractional distillation unit HEN presented in Sect. 1.2 and shown in Fig. 11.8. We will proceed with this problem according to the steps shown in Sect. 3.5 starting with the problem data next and concluding with a complete HEN design.

4.1 Problem Data

Although the original problem has 8 streams, only 6 of them require heating or cooling, so we will only consider these 6 streams. The stream list has a single cold stream, which is the Crude Oil stream, and five hot streams; Naphtha, Kerosene, LDO, HDO, and RO1. We will consider 4 hot utilities and 2 cold utilities in this problem, which are summarized below.

Hot Utilities

- *Fuel Gas (FG)*: Fuel gas is burned in a furnace to heat up process streams.
- *High Pressure Stream (HP)*: HP Steam is produced by heating water in a high pressure boiler. The primary energy used to produce HP steam is the high pressure boiler fuel.
- *Medium Pressure Stream (MP)*: MP steam is produced by heating water in a medium pressure water or by reducing the pressure of HP steam. The primary energy used to produce MP steam is the fuel in the HP and MP boilers.
- *Low Pressure Stream (LP)*: LP stream is produced in a lower pressure boiler, by reducing the HP steam pressure, or by reducing the MP stream pressure. The primary energy used is fuel in the HP, MP, and LP boilers.

Cold Utilities

- *Air*: Cold air is used to cool streams through fin-fan air coolers, where the cool air is blown using a fan on fins mounted on pipes in which a hot stream flows. The primary energy used is electricity to power the fans.

Table 11.24 Stream data for the CDU HEN design problem. The data show the starting temperature, target temperature, specific heat capacity, and fluid flow value for each stream

Stream number	Stream name	Start temp. T^s (°C)	Target temp. T^t (°C)	Specific heat capacity c_P (J/g°C)	Flow f (Mg/h)
1	Crude oil	30	400	2.20	1000
2	Naphtha	150	45	2.00	230
3	Kerosene	200	50	2.05	120
4	LDO	275	60	2.10	100
5	HDO	325	60	2.15	180
6	RO1	390	60	2.20	200

Table 11.25 Utility data for the CDU HEN design problem. The table shows the utility types, temperatures, and costs

Utility name	Utility type	Temperature T^u (°C)	Cost c ($/$M$Wh)
Fuel gas	Hot	1000	30
HP steam	Hot	250	20
MP stream	Hot	200	15
LP stream	Hot	150	6
Air	Cold	50	2
CW	Cold	30	3

- *Cooling Water (CW):* Cold water is pumped from a near by body of water and hot water is pumped back. The primary energy used is electricity for running the pumps.

The stream and utility data for this problem are given in Tables 11.24 and 11.25, and the $\Delta T_{\min}$ value used in this problem is 20°C. For simplicity, we will index the process streams according to Table 11.24, so the sets of hot and cold streams are

$$H = \{2, 3, 4, 5, 6\} \text{ and } C = \{1\}.$$

We will use the abbreviations given for each utility to get the hot and cold utility sets

$$HU = \{FG, HP, MP, LP\} \text{ and } CU = \{Air, CW\}.$$

4.2 Temperature Shifting

After collecting the data, we start the HEN design algorithm by shifting the hot process stream and cold utility temperatures down by $\Delta T_{\min} = 20$°C as described in Sect. 3.2.1. To reduce the problem size, we will perform the initial screening step for the utility temperatures shown in the temperature partitioning procedure in Sect. 3.2.4. The largest process stream temperature to which we need to heat is

Table 11.26 Temperature-shifted process stream data for the CDU HEN design problem. The data show the starting temperature, target temperature, heat capacity flow rate, and heat duty for each stream

Stream number	Stream name	Start temp. T^s (°C)	Target temp. T^t (°C)	Heat capacity flow CP (kW/°C)	Heat duty Q (kW)
1	Crude oil	30	400	611	226,070
2	Naphtha	130	25	128	13,440
3	Kerosene	180	30	68	10,200
4	LDO	255	40	58	12,470
5	HDO	305	40	108	28,620
6	RO1	370	40	122	40,260

Table 11.27 Temperature-shifted utility data for the CDU HEN design problem. The table shows the utility types, temperatures, and costs

Utility name	Utility type	Temperature T^u (°C)	Cost c ($/$M$Wh)
Fuel gas	Hot	400	30
HP steam	Hot	250	20
MP stream	Hot	200	15
LP stream	Hot	150	6
Air	Cold	30	2
CW	Cold	25	3

$T_{\max} = \max_{i \in C}\{T_i^t\} = 400$°C and the smallest temperature to which we need to cool is $T_{\min} = \min_{i \in C}\{T_i^t\} = 45$°C, which means that we do not need to heat streams beyond 400°C or cool them below 45°C. Therefore, we can reduce the fuel gas temperature from 1,000°C to 400°C and raise the cooling water temperature from 30°C to 45°C.

Table 11.26 shows the process streams with shifted hot stream temperature. We have used $CP = fc_P/3.6$ to get the heat capacity flowrate in kW/°C and Eq. (11.1) to get the heat duty Q in kW for each stream in Table 11.26. Table 11.27 shows the the shifted utility data after the initial screening step described in Sect. 3.2.4.

4.3 Temperature Partitioning

The list of starting and target temperatures from Table 11.26 and the utility temperatures from Table 11.27 (in °C) is {30, 400, 130, 25, 180, 30, 255, 40, 305, 40, 370, 40, 400, 250, 200, 150, 30, 25}. We get the following temperature partitions by removing the duplicates from this set then sorting it

$$T = (25, 30, 40, 130, 150, 180, 200, 250, 255, 305, 370, 400).$$

There are 12 temperature partitions, which means that the problem has 11 temperature intervals, so $N = 11$. A temperature interval indexed by k has a starting temperature T_{k-1}, an ending temperature T_k, a set of hot streams H_k, a set of cold streams C_k, a

Table 11.28 Temperature intervals for the CDU HEN design problem and the process stream and utility availability in each temperature interval. The symbol ϕ in this table refers to the empty set

Temp. Int. index, k	Start temp. T_{k-1} (°C)	End temp. T_k (°C)	Hot stream H_k	Cold stream C_k	Hot utility HU_k	Cold utility CU_k
1	25	30	2	ϕ	FG, HP, MP, LP	CW
2	30	40	2, 3	1	FG, HP, MP, LP	Air, CW
3	40	130	2, 3, 4, 5, 6	1	FG, HP, MP, LP	Air, CW
4	130	150	3, 4, 5, 6	1	FG, HP, MP, LP	Air, CW
5	150	180	3, 4, 5, 6	1	FG, HP, MP	Air, CW
6	180	200	4, 5, 6	1	FG, HP, MP	Air, CW
7	200	250	4, 5, 6	1	FG, HP	Air, CW
8	250	255	4, 5, 6	1	FG	Air, CW
9	255	305	5, 6	1	FG	Air, CW
10	305	370	5, 6	1	FG	Air, CW
11	370	400	ϕ	1	FG	Air, CW

set of hot utilities HU_k, and a set of cold utilities CU_k. Sect. 3.2.4 and 3.3 describe the procedure for attaining these sets. Table 11.28 shows the 11 temperature intervals and the corresponding data for each interval.

4.4 Energy Targeting

For a temperature interval k, the supply of heat of a hot process stream $i \in H_k$ is $S_{ik} = CP_i(T_k - T_{k-1})$ and the demand for heat of a cold process stream $j \in C_k$ is $D_{jk} = CP_j(T_k - T_{k-1})$. The total process stream heat supply in temperature interval k is $\sum_{i \in H_k} S_{ik} = (T_k - T_{k-1}) \sum_{i \in H_k} CP_i$ and the total demand for heat by process streams in k is $\sum_{j \in C_k} D_{jk} = (T_k - T_{k-1}) \sum_{j \in C_k} CP_j$. The process heat supply and demand for this problem is shown in Table 11.29.

We can now use this data to solve the energy targeting problem (11.17)–(11.21). The solution to the problem is shown in Table 11.30.

4.5 Stream Matching

In the stream matching problem, we extend the set of streams to include both process streams and utilities. So the extended sets of hot and cold streams are

$$\overline{H} = \{2, 3, 4, 5, 6, FG, HP, MP, LP\}$$

Table 11.29 Heat supply and demand for hot and cold streams in each temperature interval for the CDU HEN design problem. Because there is only one cold stream, having a total demand column would be redundant because the total demand equals stream 1's demand in every temperature interval

Temp. Int. index, k	Hot stream supply S_{ik} (MW) 2 (Naph)	3 (Kero)	4 (LDO)	5 (HDO)	6 (RO1)	Total supply	Demand D_{ik} (MW) 1 (Crude oil)
1	640	0	0	0	0	640	0
2	1280	680	0	0	0	1960	6110
3	11,520	6120	5220	9720	10,980	43,560	54,990
4	0	1360	1160	2160	2440	7120	12,220
5	0	2040	1740	3240	3660	10680	18,330
6	0	0	1160	2160	2440	5760	12,220
7	0	0	2900	5400	6100	14,440	30,550
8	0	0	290	540	610	1440	3055
9	0	0	0	5400	6100	11,500	30,550
10	0	0	0	0	7930	7930	39,715
11	0	0	0	0	0	0	18,330
Total	13,440	10,200	12,470	28,620	40,260	104,990	226,070

Table 11.30 Targeting solution to the CDU HEN design problem

Temp. Int. index, k	Residual heat flow r_k (kW)	Utility flow, q_{ik} (kW) FG	HP	MP	LP	Air	CW
1	0	0	0	0	0	0	640
2	4150	0	0	0	0	0	0
3	15,580	0	0	0	0	0	0
4	0	0	0	0	20,680	0	0
5	7650	0	0	0	0	0	0
6	0	0	0	14,110	0	0	0
7	0	0	16,150	0	0	0	0
8	1615	0	0	0	0	0	0
9	20,665	0	0	0	0	0	0
10	52,450	0	0	0	0	0	0
11	0	70,780	0	0	0	0	0

$$\overline{C} = \{1, Air, CW\}.$$

Not counting the last interval, the energy targeting solution in Table 11.30 has four intervals with 0 residual flow, which means that the problem has 4 pinch temperatures ($N_P = 4$) and 5 subnetworks. The pinch intervals are $P = (1, 4, 6, 7)$, which correspond to the temperatures 30, 150, 200, and 250°C. The subnetworks, their corresponding temperature intervals, hot streams $\overline{H}^s$, and cold streams $\overline{C}^s$ are shown in Table 11.31. Using the data in Tables 11.29 and 11.30, we can get the heat supply and demand for each stream and temperature interval in the different subnetworks as shown in Tables 11.32 and 11.33.

The matching problem given by (11.22)–(11.29) is solved for each of the five subnetworks. The matching solutions to the subnetworks are shown in Sects. 4.5.1–4.5.5 next.

Table 11.31 Subnetworks, temperature intervals, and stream availability in each subnetwork for the CDU HEN problem

Subnetwork s	Temp. Int. index, k	Start temp. T_{k-1} (°C)	End temp. T_k (°C)	Hot streams $\overline{H}^s$	Cold streams $\overline{C}^s$
1	1	25	30	2	CW
2	2	30	40	$2, 3, 4, 5, 6, LP$	1
	3	40	130		
	4	130	150		
3	5	150	180	$3, 4, 5, 6, MP$	1
	6	180	200		
4	7	200	250	$4, 5, 6, HP$	1
5	8	250	255	$4, 5, 6, FG$	1
	9	255	305		
	10	305	370		
	11	370	400		

4.5.1 Subnetwork 1

This subnetwork only has two streams and one temperature interval, so the matching configuration is trivial. The subnetwork design has a single heat exchanger, E1, whose information is given in Table 11.34. The subnetwork configuration is shown in Fig. 11.41.

4.5.2 Subnetwork 2

This subnetwork has 7 streams and three temperature intervals. The matching solution indicates that the single cold stream, Crude Oil, be matched with every cold stream, so the subnetwork problem has 6 heat exchangers which we name E2–E7. The solutions for the heat flows between hot and cold streams in every temperature interval q_{ijk} and the residual heat flows from hot streams to temperature intervals r_{ik} are shown in Tables 11.35 and 11.36. The heat exchanger information is given in Table 11.37. Notice that the cold stream branches into five parallel streams. The last row in Table 11.37 shows the percentage of the cold stream's flow in each parallel heat exchanger. Figure 11.42 shows a schematic of the subnetwork's HEN.

4.5.3 Subnetwork 3

Subnetwork 3 has 4 hot streams, 1 cold stream, and 2 temperature intervals. The matching solution indicates that every hot stream is matched with the cold stream, which gives 5 heat exchangers that we name E8–E12. The heat flow and residual flow data are shown in Tables 11.38 and 11.39 and the heat exchanger data for this subnetwork is given in Table 11.40. The HEN configuration is shown in Fig. 11.43.

Table 11.32 Supply of each hot stream in every temperature interval in the subnetworks of the CDU HEN problem

Subnetwork	Temp. Int.	Hot stream supply, S_{ik} (kW)								
s	index, k	2 (Naph)	3 (Kero)	4 (LDO)	5 (HDO)	6 (RO1)	FG	HP	MP	LP
1	1	640	0	0	0	0	0	0	0	0
2	2	1280	680	0	0	0	0	0	0	0
	3	11,520	6120	5220	9720	10,980	0	0	0	0
	4	0	1360	1160	2160	2440	0	0	0	20,680
3	5	0	2040	1740	3240	3660	0	0	0	0
	6	0	0	1160	2160	2440	0	0	14,110	0
4	7	0	0	2900	5400	6100	0	16,150	0	0
5	8	0	0	290	540	610	0	0	0	0
	9	0	0	0	5400	6100	0	0	0	0
	10	0	0	0	0	7930	0	0	0	0
	11	0	0	0	0	0	70,780	0	0	0

Table 11.33 Demand for each cold stream in every temperature interval in the subnetworks of the CDU HEN problem

Subnetwork s	Temp. Int. index, k	Cold stream demand, D_{ik} (kW) 1 (Crude oil)	Air	CW
1	1	0	0	640
2	2	6110	0	0
	3	54,990	0	0
	4	12,220	0	0
3	5	18,330	0	0
	6	12,220	0	0
4	7	30,550	0	0
5	8	3055	0	0
	9	30,550	0	0
	10	39,715	0	0
	11	18,330	0	0

Table 11.34 Heat exchanger information for subnetwork 1 of the CDU HEN design problem. This subnetwork has the single heat exchanger E1

HE Attribute	E1
Type	Utility
Hot stream	Naphtha
Cold stream	CW
Duty (kW)	640
Hot T_{in} (°C)	30
Hot T_{out} (°C)	25
Cold T_{in} (°C)	25
Cold T_{out} (°C)	25
Hot CP (kW/°C)	128
Cold CP (kW/°C)	NA

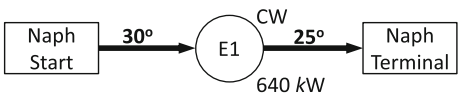


Fig. 11.41 HEN design schematic for subnetwork 1 in the CDU HEN design problem. (Refer to footnote 9 in Sect. 3.4.4)

4.5.4 Subnetwork 4

This subnetwork has 4 hot streams, 1 cold stream, and 1 temperature interval, and the matching solution indicates that every hot stream is matched with the Crude Oil cold stream. Therefore, the subnetwork has four heat exchangers, which we name E13–E16. The heat flow solution q_{ijk} for this subnetwork is shown in Table 11.41. The subnetwork has no residual heat values because it only has 1 temperature interval. The heat exchanger information is given in Table 11.42 and the HEN design for this subnetwork is shown in Fig. 11.44.

Table 11.35 The heat flow variables q_{ijk} in kW from the matching solution for subnetwork 2 of the CDU HEN design problem. Each entry shows the heat flow from hot streams (rows) to the crude oil cold stream in every temperature interval

Stream	Crude oil			
Interval (k)	2	3	4	*Total*
Naphtha	1280	11520	0	12800
Kerosene	680	7480	0	8160
LDO	0	6380	0	6380
HDO	0	11880	0	11880
RO1	0	13420	0	13420
LP	4150	4310	12220	20680
Total	6110	54990	12220	73320

Table 11.36 Residual heat flow r_{ik} in kW from hot stream i to temperature interval k in subnetwork 2 of the CDU HEN design problem

Interval k	2	3
Naphtha	0	0
Kerosene	0	1360
LDO	0	1160
HDO	0	2160
RO1	0	2440
LP	4150	8460

4.5.5 Subnetwork 5

Subnetwork 5 has 4 hot streams, a single cold stream, and 4 temperature intervals. The matching solution shows that every hot streams is matched with the cold Crude Oil stream, and so the subnetwork has the 4 heat exchangers E17–E20. The heat flow results q_{ijk} are shown in Table 11.43 and the residual flow results r_{ik} are shown in Table 11.44. The heat exchanger information is shown in Table 11.45 and the network design schematic for this subnetwork is shown in Fig. 11.45.

Table 11.37 Heat exchanger information for subnetwork 2 of the CDU HEN design problem. This subnetwork has the six heat exchangers E2–E7

HE Attribute	E2	E3	E4	E5	E6	E7
Type	P2P	P2P	P2P	P2P	P2P	Utility
Hot stream	Naptha	Kerosene	LDO	HDO	RO1	*LP*
Cold stream	Crude oil	Crude oil	Crude oil	Crude oil	Crude oil	Crude oil
Duty (kW)	12,800	8160	6380	11,880	13,420	20,680
Hot T_{in} ($^\circ C$)	130	150	150	150	150	150
Hot T_{out} ($^\circ C$)	30	30	40	40	40	150
Cold T_{in} ($^\circ C$)	30	30	30	30	30	116.1538
Cold T_{out} ($^\circ C$)	116.15	116.15	116.15	116.15	116.15	150
Hot *CP* (kW/°C)	128	68	58	108	122	NA
Cold *CP* (kW/°C)	148.57	94.71	74.05	137.89	155.77	611
% of flow	24.3 %	15.5 %	12.1 %	22.6 %	25.5 %	100.0 %

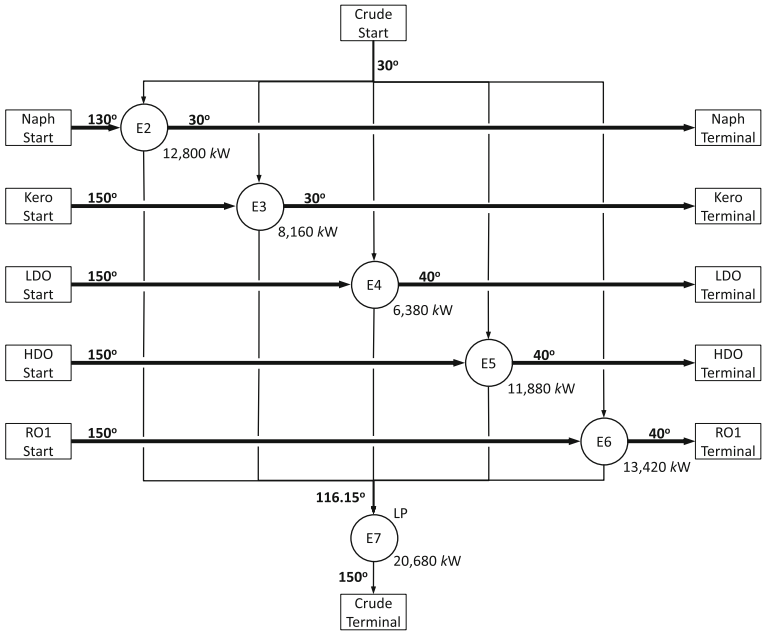


Fig. 11.42 HEN design schematic for subnetwork 2 in the CDU HEN design problem. (Refer to footnote 9 in Sect. 3.4.4)

Table 11.38 The heat flow variables q_{ijk} in kW from the matching solution for subnetwork 3 of the CDU HEN design problem. Each entry shows the heat flow from hot streams (*rows*) to the crude oil cold stream in every temperature interval

Stream	Crude oil		
Interval (k)	5	6	*Total*
Kerosene	2040	0	2040
LDO	1740	1160	2900
HDO	3240	2160	5400
RO1	3660	2440	6100
MP	7650	6460	14,110
Total	18,330	12,220	30,550

Table 11.39 Residual heat flow r_{ik} in kW from hot stream i to temperature interval k in subnetwork 3 of the CDU HEN design problem

Interval k	5
Kerosene	0
LDO	0
HDO	0
RO1	0
MP	7650

4.6 Final HEN Design

After getting the design for every subnetwork, all that remains is to undo the temperature shifting made in Sect. 4.2 by adding $\Delta T_{\min} = 20°\text{C}$ to all hot stream

Table 11.40 Heat exchanger information for subnetwork 3 of the CDU HEN design problem. This subnetwork has the five heat exchangers E8–E12

HE Attribute	E8	E9	E10	E11	E12
Type	P2P	P2P	P2P	P2P	Util
Hot stream	Kerosene	LDO	HDO	RO1	*MP*
Cold stream	Crude oil	Crude oil	Crude oil	Crude oil	Crude oil
Duty (kW)	2040	2900	5400	6100	14110
Hot T_{in} ($^{\circ}C$)	180	200	200	200	200
Hot T_{out} ($^{\circ}C$)	150	150	150	150	200
Cold T_{in} ($^{\circ}C$)	150	150	150	150	176.91
Cold T_{out} ($^{\circ}C$)	176.91	176.91	176.91	176.91	200
Hot CP (kW/°C)	68	58	108	122	NA
Cold CP (kW/°C)	75.82	107.78	200.69	226.71	611
% of flow	12.4 %	17.6 %	32.8 %	37.1 %	100 %

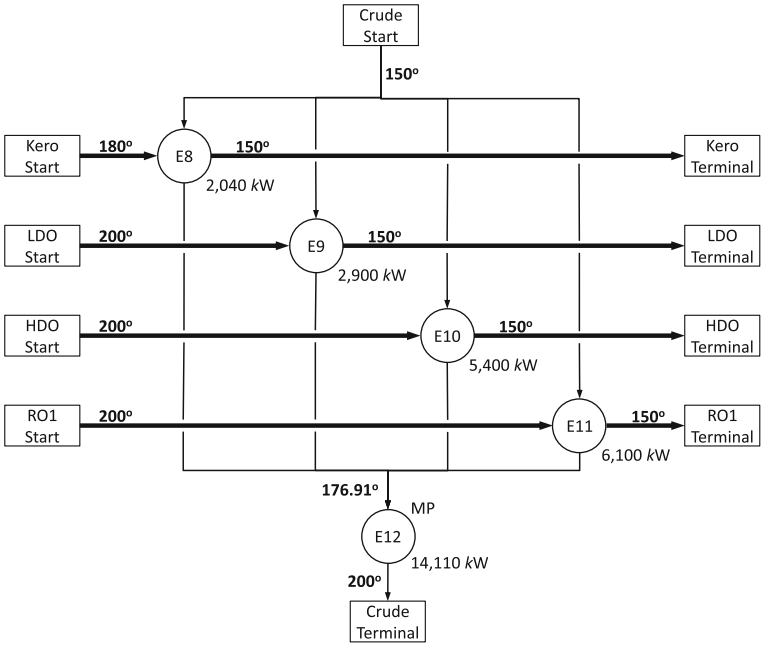


Fig. 11.43 HEN design schematic for subnetwork 3 in the CDU HEN design problem. (Refer to footnote 9 in Sect. 3.4.4)

temperatures, then connect all the subnetwork solutions into a single HEN. The resulting network is shown in Fig. 11.46.

Table 11.41 The heat flow variables q_{ijk} in kW from the matching solution for subnetwork 4 of the CDU HEN design problem. Each entry shows the heat flow from hot streams (rows) to the crude oil cold stream in every temperature interval

Stream	Crude oil
Interval (k)	7
LDO	2900
HDO	5400
RO1	6100
HP	16,150
Total	30,550

Table 11.42 Heat exchanger information for subnetwork 4 of the CDU HEN design problem. This subnetwork has the four heat exchangers E13–E16

HE Attribute	E13	E14	E15	E16
Type	P2P	P2P	P2P	Util
Hot stream	LDO	HDO	RO1	HP
Cold stream	Crude oil	Crude oil	Crude oil	Crude oil
Duty (kW)	2900	5400	6100	16,150
Hot T_{in} ($^\circ C$)	250	250	250	250
Hot T_{out} ($^\circ C$)	200	200	200	250
Cold T_{in} ($^\circ C$)	200	200	200	223.57
Cold T_{out} ($^\circ C$)	223.57	223.57	223.57	250
Hot CP (kW/°C)	58	108	122	NA
Cold CP (kW/°C)	123.05	229.13	258.83	611
% of flow	20.1 %	37.5 %	42.4 %	100 %

Table 11.43 The heat flow variables q_{ijk} in kW from the matching solution for subnetwork 5 of the CDU HEN design problem. Each entry shows the heat flow from hot streams (rows) to the crude oil cold stream in every temperature interval

Stream	Crude oil				
Interval (k)	8	9	10	11	*Total*
LDO	290	0	0	0	290
HDO	540	5400	0	0	5940
RO1	2225	4485	7930	0	14,640
FG	0	20,665	31,785	18,330	70780
Total	3055	30,550	39,715	18,330	91,650

Table 11.44 Residual heat flow r_{ik} in kW from hot stream i to temperature interval k in subnetwork 5 of the CDU HEN design problem

Interval k	8	9	10
LDO	0	0	0
HDO	0	0	0
RO1	1615	0	0
FG	0	20,665	52, 450

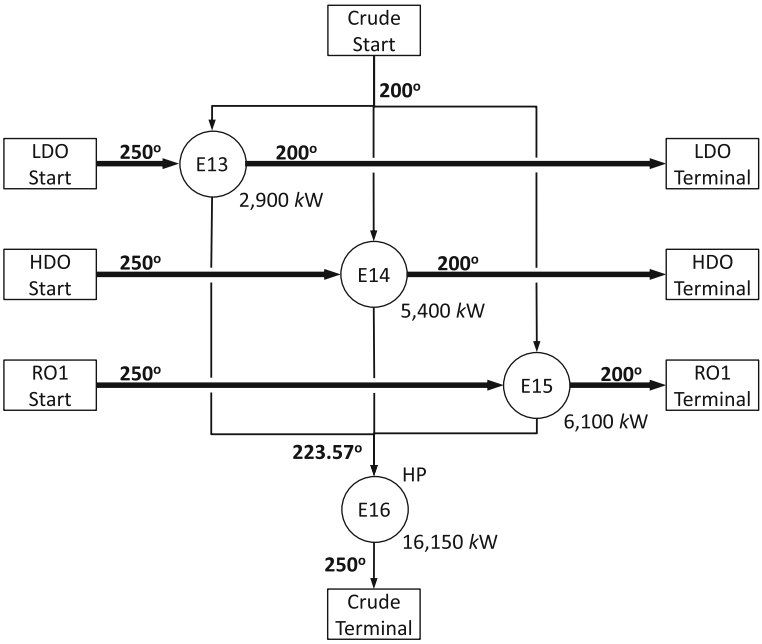


Fig. 11.44 HEN design schematic for subnetwork 4 in the CDU HEN design problem. (Refer to footnote 9 in Sect. 3.4.4)

Table 11.45 Heat exchanger information for subnetwork 5 of the CDU HEN design problem. This subnetwork has the four heat exchangers E17–E20

HE Attribute	E17	E18	E19	E20
Type	P2P	P2P	P2P	Utility
Hot stream	LDO	HDO	RO1	Fuel gas
Cold stream	Crude oil	Crude oil	Crude oil	Crude oil
Duty (kW)	290	5940	14,640	70,780
Hot T_{in} ($^{\circ}C$)	255	305	370	400
Hot T_{out} ($^{\circ}C$)	250	250	250	400
Cold T_{in} ($^{\circ}C$)	250	250	250	284.16
Cold T_{out} ($^{\circ}C$)	284.16	284.16	284.16	400
Hot CP (kW/°C)	58	108	122	NA
Cold CP (kW/°C)	8.49	173.90	428.61	611
% of flow	1.39 %	28.46 %	70.15 %	100 %

5 Conclusion

We considered in this chapter the HEN design problem with the primary objective of minimizing the total utilities cost, and with the secondary objective to minimize the number of heat exchanger units. The problem is solved over two stages. First, we find the minimum utilities consumption in the *energy targeting problem*, then we find the set of heat exchangers in the *stream matching problem*. The energy matching problem

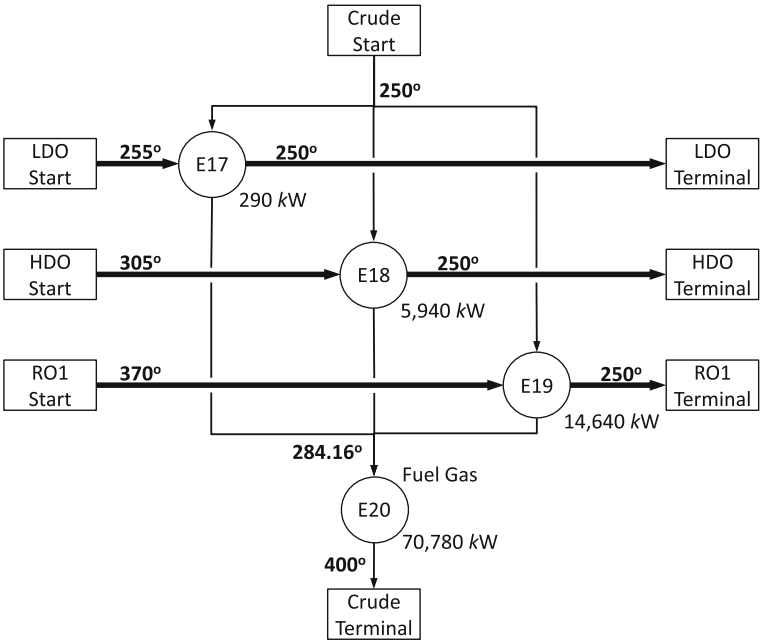


Fig. 11.45 HEN design schematic for subnetwork 5 in the CDU HEN design problem. (Refer to footnote in Sect. 3.4.4)

can be formulated as a transshipment problem, while the stream matching problem is formulated as a mixed integer programming problem, which is a more difficult problem to solve. In fact, the stream matching problem is NP-hard as shown by [7]. Another common approach to solve this problem is by constructing a "superstructure" model that can represent all possible HEN configurations (see for example [3, 5, 6]). A common extension to the HEN design problem is to simultaneously solve for the heat exchanger areas while designing the HEN (e.g., [15] and [4]). This problem is typically formulated as a mixed integer nonlinear program that is difficult to solve in general, which is why many heuristic methods have been developed for this problem (e.g., [10]).

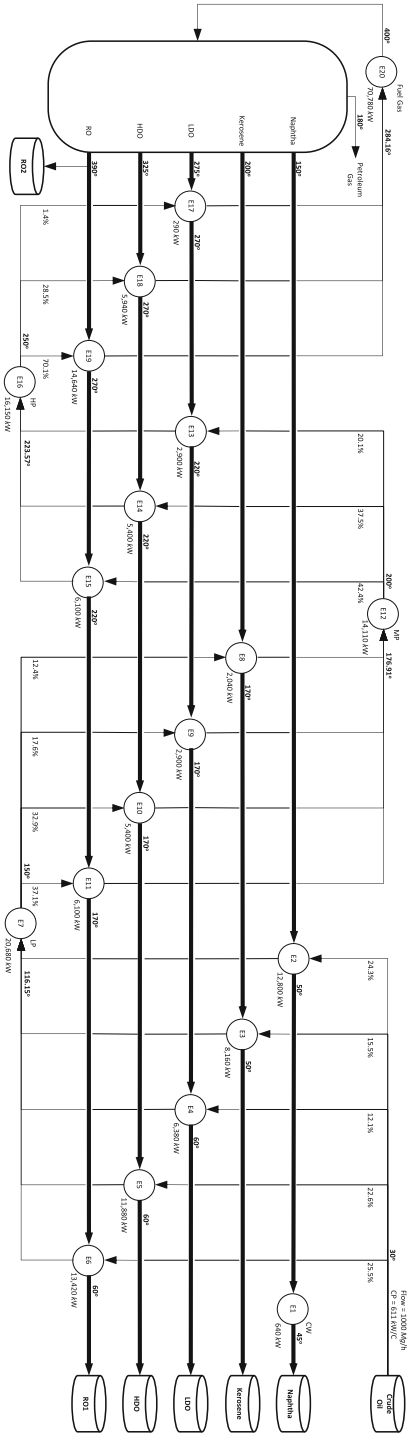


Fig. 11.46 Final CDU HEN design. This HEN has many stream splits, shown as *vertical lines* branching from the horizontal crude oil stream flow lines in the *top-most* and *bottom-most lines* in the schematic. The stream split ratios are indicated on the beginning of each *branching lines* as a percentage of the original stream's flow. (Refer to footnote 9 in Sect. 3.4.4)

References

1. Bazara, M. S., Sherali, H. D., & Shetty, C. M. (2006). *Nonlinear programming: Theory and Algorithms*. Wiley
2. Cerda, J., Westerberg, A. W., Mason, D., & Linnhoff, B. (1983). Minimum utility usage in heat exchanger network synthesis. *Chemical Engineering Science, 38*(3), 373–387.
3. Ciric, A., & Floudas, C. (1989). A retrofit approach for heat exchanger networks. *Computers & chemical engineering, 13*(6), 703–715.
4. Ciric, A., & Floudas, C. (1991). Heat exchanger network synthesis without decomposition. *Computers & chemical engineering, 15*(6), 385–396.
5. Floudas, C. A., & Grossmann, I. E. (1987). Synthesis of flexible heat exchanger networks with uncertain flowrates and temperatures. *Computers & chemical engineering, 11*(4), 319–336.
6. Floudas, C. A., Ciric, A. R., & Grossmann, I. E. (1986). Automatic synthesis of optimum heat exchanger network configurations. *AIChE Journal, 32*(2), 276–290.
7. Furman, K. C., & Sahinidis, N. V. (2001). Computational complexity of heat exchanger network synthesis. *Computers & Chemical Engineering, 25*(9), 1371–1390.
8. Kakac, S., Pramuanjaroenkij, A., & Liu, H. (2012). *Heat exchangers: Selection, rating, and thermal design*. CRC press.
9. Kemp, I. C. (2007). *Pinch analysis and process integration: A user guide on process integration for the efficient use of energy*. Butterworth Heinemann, Waltham, MA, USA; and Oxford, UK.
10. Lewin, D. R. (1998). A generalized method for hen synthesis using stochastic optimizationii.: The synthesis of cost-optimal networks. *Computers and chemical engineering, 22*(10), 1387–1405.
11. Linnhoff, B., & Flower, J. R. (1979). Synthesis of heat exchanger networks: I. systematic generation of energy optimal networks. *AIChE Journal, 24*(4), 633–642.
12. Linnhoff, B., & Hindmarsh, E. (1983). The pinch design method for heat exchanger networks. *Chemical Engineering Science, 38*(5), 745–763.
13. Murty, K. G. (1988). *Linear complementarity, linear and nonlinear programming*. Can be downloaded from the authors website at: http://www-personal.umich.edu/∼murty/
14. Papouliast, S. A., & Grossman, I. E. (1983). A structural optimization approach in process synthesis-II. *Computers and Chemical Engineering, 7*(6), 707–721.
15. Yee, T. F., & Grossmann, I. E. (1990). Simultaneous optimization models for heat integration-II. Heat exchanger network synthesis. *Computers and Chemical Engineering, 14*(10), 1165–1184.

Chapter 12
Optimizing the Allocation of Cuboidal Boxes to Cuboidal Compartments for Storage in a Warehouse

G. S. R. Murthy, A. L. N. Murthy and Katta G. Murty

1 Introduction

In this chapter we consider the problem of managing the storage space optimally at a warehouse for storing cuboidal boxes in cuboidal compartments. Footwear manufacturers face this problem for storing shoe boxes; drug companies manufacturing medicines packed in cartons cuboidal in shape face the same problem, etc. Typically, warehouse management problems involve continual storage and retrieval (issues) of goods from the warehouse (Fig. 12.1). Therefore, a scheme is required to handle the dynamic storage and retrieval of goods optimally from the warehouse. In this chapter we will discuss an efficient procedure for developing a decision support system for the dynamic warehouse management problem. The source for this chapter is based on the work done by Das [5] and a subsequent paper by Murthy [11].

We shall first describe the problem briefly here before proceeding with the mathematical formulation and providing a solution to the problem. As mentioned earlier, the problem in this chapter deals with storing and retrieving cuboidal boxes (or cartons) in cuboidal compartments (or racks). The boxes and compartments vary in sizes (lengths, breadths and heights) and information on them are provided as input data to the problem. Depending upon the flow of goods (for sales and/or consumption),

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_12) contains supplementary material, which is available to authorized users.

G. S. R. Murthy (✉) · A. L. N. Murthy
SQC & OR Unit, Indian Statistical Institute, Street No. 8, Habsiguda,
Hyderabad, 500 007, India
e-mail: murthygsr@gmail.com,

A. L. N. Murthy
e-mail: simhaaln@rediffmail.com

K. G. Murty
Department of Industrial and Operations Engineering, University of Michigan,
Ann Arbor, MI 48109-2117, USA
e-mail: murty@umich.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_12, © Springer Science+Business Media New York 2015

Fig. 12.1 Warehouse of a shoe manufacturing company. Each rack segment is a compartment here. For details see "Images of warehouse of shoes" at https://www.google.co.in/search?q=warehouse+of+shoes

decisions are to be made each time a consignment is received and/or goods are withdrawn from the warehouse. In addition to the knowledge of sizes, the quantities (the number of boxes and the number of compartments available) are also given in the data. We shall present the methodology with the help of a problem that was solved for a footwear industry in India [see 5]. We shall refer to this problem as *The Footwear Inventory Management Problem* (FIMP). Apart from space constraints that arise naturally in warehousing problems, one faces constraints from the angle of operational conveniences. Two of the constraints faced in FIMP are stated below.

Constraint 1. Each compartment can be used only for one type of box. That is, two different types of boxes cannot be stored in a single compartment.

Constraint 2. The placement of any box in any compartment should be such that the side representing the height of the box should be parallel to the side of the compartment representing the height of the compartment, and the base sides of the boxes should be aligned with the base sides of the compartment.

The following notation and assumptions are used. We shall use the upper case letters L, B, and H for the length, breadth and height of any cuboidal compartment respectively; and use the lower case letters l, b, and h for the length, breadth and height of cuboidal boxes respectively. Finally, the dimensions of a cuboid will be mentioned as $Length \times Breadth \times Height$, that is, in the order of length first, breadth next and height last.

With the above notation and understanding, Constraint 2 above insists that boxes should be placed in compartments in such a way that the edge of any box representing its height should be parallel only to the edges of the compartments representing their heights.

There are two decision making problems here. The first of these arises when setting up the warehouse. Based on the movement of goods (sales/consumption), the initial volume of goods to be stored at the time of setting up the warehouse will be worked based on the sales/consumption patterns. Given the initial volumes, the number of boxes of each type to be stored in FIMP, the problem is to determine

Table 12.1 Compartments in the warehouse

Compartment type	Dimensions L	B	H	Number of compartments
C1	48	24	27	390
C2	36	24	27	534
C3	48	24	18	130
C4	36	24	18	178
C5	36	16	28	52
C6	36	16	18	52
Total number of compartments				1336

Dimensions are in inches

the number of compartments, type wise, and determine which boxes to store in which compartments. This problem will be referred to as the *Set-up Version*. The second decision making problem is a repetitive one. Each time a consignment is received, a decision is to be made on where to store the boxes of the consignment in the compartments. It must be noted that when a consignment is received, the compartments may be empty or partially occupied due to the presence of inventory in them. Therefore, to make the decision when a consignment is received, one must know the position of each compartment at that time. This information can be made available at any time if the inventory database is maintained properly. We shall refer to the second decision making problem, the repetitive one, as the *Dynamic Version*. Essentially, the two versions of the problem only differ in their inputs. But for this, we can use the same methodology for solving them.

2 Set-up Version

In this section we shall consider the Set-up Version of the problem. Here again, we can consider two situations: (i) a warehouse is already in place and (ii) a warehouse is to be set up. For FIMP, a warehouse was already in place. Therefore, we shall confine ourselves to the first situation.

The warehouse can be simply described by specifying the number of different compartments in it along with their dimensions. FIMP has 1336 compartments comprising of six types. The type of a compartment is determined by its dimensions. The dimensions of the six types of compartments and the number of compartments of each type are presented in Table 12.1. Unless stated otherwise, all the linear measurements in this chapter will be in inches.

In the set-up version, it is assumed that all the compartments are empty, and that the requirement of boxes to be stored is prescribed by the management. Table 12.2 presents the data on boxes for FIMP. In these data, we have 28 types of boxes and their dimensions are given in second, third, and fourth columns. The data in the column titled "Consignment" are the numbers of boxes of respective types to be stored initially.

Table 12.2 Box types and consignment

Type	l	b	h	Consignment
B1	3.5	2.0	8.5	1300
B2	3.0	4.5	7.0	3800
B3	3.0	3.5	9.5	800
B4	3.0	4.5	7.5	1900
B5	3.5	4.0	9.0	300
B6	3.5	3.5	10.0	500
B7	3.5	5.0	7.5	5900
B8	4.0	3.0	11.0	1500
B9	4.0	3.0	11.5	1400
B10	3.5	3.5	11.5	500
B11	3.5	5.0	9.0	500
B12	4.0	4.5	10.5	1900
B13	4.0	5.0	10.5	11900
B14	4.0	6.0	10.0	1900
B15	4.0	5.5	11.0	7500
B16	4.0	6.0	10.5	4500
B17	4.5	5.0	11.5	16700
B18	5.5	4.0	12.0	3800
B19	4.5	5.0	12.0	800
B20	4.0	6.0	11.5	5800
B21	4.5	5.5	11.5	800
B22	4.0	7.5	11.0	500
B23	4.0	7.0	12.0	1200
B24	4.0	8.0	11.0	300
B25	4.5	7.0	12.5	500
B26	5.0	7.5	11.5	1300
B27	4.0	10.0	12.5	400
B28	5.0	8.0	13.0	300

3 Storing Boxes in Compartments

As pointed out earlier, both versions of the problem, set-up and dynamic, are essentially same but for their inputs. In order to solve the common problem, we must first decide upon how the boxes will be stored in the compartments. Though theory might suggest that the storing should be such that the number of boxes to be stored in a compartment be as large as possible, such suggestions may not be practical because of operational inconveniences. For this reason, restrictions such as Constraint 1 and Constraint 2 mentioned earlier are imposed in practice. Therefore, the first decision making problem in warehouse management is to decide upon the storage pattern of boxes into the compartments. Once this is decided, the routine inventory management problem seeks a solution to usage of compartments to store the boxes. We shall consider both these problems but first take up the problem of deciding upon the storage pattern of boxes in compartments.

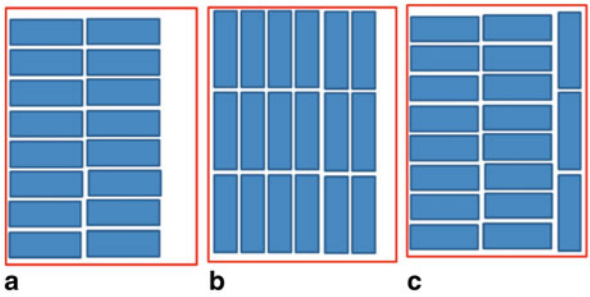


Fig. 12.2 Different patterns of storing

3.1 Determination of Storage Pattern

Consider one type of compartment (say $C1$) and one type of box (say $B1$). Let their dimensions be $L \times B \times H$ and $l \times b \times h$ respectively. The question is: How many boxes of type $B1$ can be stored in a type $C1$ compartment? Due to restriction of Constraint 2, we can store k layers of boxes where $k = \lfloor \frac{H}{h} \rfloor$, the largest integer less than or equal to H/h. The number of boxes that can be stored in each layer depends on arrangement or pattern of boxes in each layer. Assuming that boxes are stored in the same pattern in each layer, the total number of boxes stored in the compartment is equal to kq, where q is the number of boxes in each layer. But q depends upon the pattern. Consider the three patterns shown in Fig. 12.2. The pattern in (a) accommodates 16 boxes in a single layer while patterns in (b) and (c) accommodate 18 and 19 boxes respectively. The difference between the first two patterns and the last pattern (c) is that all the boxes in (a) and (b) have their lengths in the same direction.

Obviously maximizing the number of boxes per layer will lead to optimal solutions in terms of utilizing the storage space optimally. But insisting on putting maximum number of boxes per layer may cause operational inconveniences. In addition, finding the maximum number of boxes per layer involves solving Bin Packing problems or the two dimensional cutting stock problem. The Bin packing problems are well known and well studied problems in the literature of Operations Research [see 6, 7, 9, 10, 13]. Bin packing problems are NP-hard problems. Numerous articles in the literature have dealt with algorithms, heuristics and metaheuristics for solving the 2D bin packing problems. For details see [1, 2, 4, 6, 8 13].

To avoid the complexity of implementing mixed type of storage patterns (like in Fig. 12.2c) in practice for the warehouse operators, the following constraint is imposed.

Constraint 3. All the boxes stored within a compartment should have their lengths aligned in the same direction.

Note that Constraint 3 means that only patterns of the type (a) or (b) are allowed. Under this constraint, we have a simple formula for w_{ij}, the maximum number of boxes per layer for storing the i^{th} type boxes (with dimensions $l_i \times b_i \times h_i$) in j^{th} type

Table 12.3 The **W**-matrix

Box Type	C1	C2	C3	C4	C5	C6
B1	468	360	312	240	240	160
B2	240	192	160	128	160	80
B3	208	160	104	80	100	50
B4	240	192	160	128	120	80
B5	234	180	156	120	120	80
B6	156	120	78	60	80	40
B7	162	126	108	84	90	60
B8	192	144	96	72	96	48
B9	192	144	96	72	96	48
B10	156	120	78	60	80	40
B11	162	126	108	84	90	60
B12	120	96	60	48	64	32
B13	108	84	54	42	56	28
B14	96	72	48	36	48	24
B15	96	72	48	36	48	24
B16	96	72	48	36	48	24
B17	90	70	45	35	48	24
B18	96	72	48	36	48	24
B19	90	70	45	35	48	24
B20	96	72	48	36	48	24
B21	80	64	40	32	36	18
B22	72	54	36	27	36	18
B23	72	60	36	30	40	20
B24	72	54	36	27	36	18
B25	60	50	30	25	32	16
B26	54	42	27	21	28	14
B27	48	36	24	18	24	12
B28	54	42	27	21	28	14

The ijth number in the table is w_{ij}

compartment (with dimensions $L_j \times B_j \times H_j$). The formula is given by $w_{ij} = k\lambda$, where $k = \lfloor \frac{H_j}{h_i} \rfloor$ and $\lambda = \max\{\lfloor \frac{L_j}{l_i} \rfloor \lfloor \frac{B_j}{b_i} \rfloor, \lfloor \frac{L_j}{b_i} \rfloor \lfloor \frac{B_j}{l_i} \rfloor\}$. If $\lambda = \lfloor \frac{L_j}{l_i} \rfloor \lfloor \frac{B_j}{b_i} \rfloor$, then the boxes should be arranged in such a way that the lengths of the boxes are aligned with the length of the compartment, otherwise, the box lengths should be aligned with the width of the compartment. In the example of Fig. 12.2, $w_{ij} = 18$ and the length of the boxes are aligned with the length of the compartment (pattern (b)). Table 12.3 presents the w_{ij}s for FIMP with data given in Tables 12.1 and 12.2.

4 Mathematical Formulation

We now return to our main problem of modeling the inventory management. We first present the mathematical model for the problem. Consider m types of cuboidal boxes with dimensions of i^{th} type being $l_i \times b_i \times h_i$, where l_i, b_i, and h_i are length, breadth and height of the i^{th} type box respectively, $i = 1, 2, \ldots, m$. Consider n types of cuboidal compartments with dimensions of jth type being $L_j \times B_j \times H_j$,

where L_j, B_j, and H_j are length, breadth, and height of the jth type compartment respectively, $j = 1, 2, \ldots, n$. For FIMP, $m = 28$ and $n = 6$. Let v_j be the volume of the jth compartment (in cubic inches). There are a_i boxes of type i, $i = 1, 2, \ldots, m$, to be stored in the warehouse, and compartments c_j of type j, $j = 1, 2, \ldots, n$ are available in the warehouse to store them. It is assumed that the boxes in compartments are stored in such a way that constraints 1, 2, and 3 are satisfied. Further, we assume that w_{ij} is the capacity of storing i^{th} type boxes in j^{th} type compartment. We shall call the matrix of w_{ij}s as the *weights matrix* or the *W-matrix*.

The objective in this problem is to use minimum number of compartments to accommodate the given $\sum a_i$ boxes. An alternative objective function could be to minimize the total volume of the compartments utilized for storing all the boxes. Define

$$x_{ij} = \text{number of } j^{th} \text{ type compartments to be used for storing the } i^{th} \text{ type boxes,}$$

$j = 1, 2, \ldots, n$, and $i = 1, 2, \ldots, m$. As $v_j = L_j B_j H_j$ is the volume of j^{th} type compartment, $\sum_j v_j \sum_i x_{ij}$ is the volume of compartments utilized for storing all the boxes. On the other hand, $\sum_j \sum_i x_{ij}$ will be the number of compartments used for storing all the boxes.

The optimization problem of the set-up version is to minimize either $\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}$ or $\sum_{j=1}^{n} v_j \sum_{i=1}^{m} x_{ij}$ subject to the constraint that all the $\sum a_i$ boxes are stored in the compartments. Both the problems are special cases of the following problem.

$$\text{Minimize} \sum_j \sum_i \mu_{ij} x_{ij}$$

$$\text{subject to} \sum_j w_{ij} x_{ij} \geq a_i \tag{12.1}$$

$$\sum_i x_{ij} \leq c_j, \; x_{ij}\text{s are nonnegative integers.} \tag{12.2}$$

The LP relaxation of this problem is known as the generalized transportation problem (GTP). If we set $\mu_{ij} = v_j$ for all i, then the above problem is FIMP with volume as the objective function. On the other hand, if we set $\mu_{ij} = 1$ for all i, j, then the problem is FIMP with number of compartments as the objective function.

GTP is a special case of generalized network flow problem—a well studied problem in the literature of Operations Research. The generalized network flow problems can be solved efficiently. An implementation of the primal simplex algorithm without the need for computing the inverses can be used to compute the solutions of GTP. Murty (1992) presents a detailed discussion on the treatment of generalized network flow problems.

Note that the inputs to this problem are a_is, c_js, v_js, and w_{ij}s. For ease of reference, we shall use the vector notation for these inputs. That is, we shall write $\mathbf{a} = (a_1, a_2, \ldots, a_m)$, $\mathbf{c} = (c_1, c_2, \ldots, c_n)$, $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ and $\mathbf{W}$, the weights matrix of w_{ij}s.

5 Solution to FIMP

In this section we shall illustrate our methodology with the help of FIMP and provide the solutions for both versions, the set-up version and the dynamic version. Before doing this, let us have a clear understanding of FIMP. For quick reference, the list of various symbols used are presented in Exhibit.

Based on the assessment of movement of material, consider the situation where the warehouse receives the same consignment, $\mathbf{a} = (a_1, a_2, \ldots, a_m)$, where a_i is the number of footwear boxes of type $i = 1, 2, \ldots, m$, at the end of every week to store. The vector **a** is called the *consignment vector.* We can also consider the more general case where consignment vector changes in every shipment (see Exercise 1). Thus, to start with, the initial inventory is given by the vector **a**. For FIMP, the vector **a** is given in Table 12.4.

Exhibit of notation Used

Symbol	Description
n	No. of types of compartments
m	No. of types of boxes
i	The index used for box type, $1 \leq i \leq m$
j	The index used for compartment type, $1 \leq j \leq n$
Bi	Box type i, B1 is box type 1, B2 box type 2, and so on
Cj	Compartment type j, C1 is compartment type 1, and so on
L_j, B_j, H_j	Length, breadth, and height of Cjs
l_i, b_i, h_i	Length, breadth and height of Bis
v_j	Volume (in cubic inches) of Cjs
a_i	No. of Bis in a consignment vector $\mathbf{a} = (a_i)$ received at warehouse
$\bar{a}_i$	No. of Bis in a consignment vector received after filling up the partially filled compartment with Bis to the extent possible
c_j	No. of Cjs in the warehouse vector $\mathbf{c} = (c_j)$
$\bar{c}_j$	No. of empty Cjs when a new consignment is received, $\bar{\mathbf{c}} = (\bar{c}_j)$
rs_i	No. of Bis remaining in stock when a new consignment received, $\mathbf{rs} = (rs_i)$
q_{ij}	No. of Cjs containing Bi from remaining stock when a new consignment is received
r_i	Maximum no. of additional Bi required to fill the partially filled compartment completely with Bis when a new consignment is received
w_{ij}	The maximum no. of Bis that can be stored in a Cj; $W = (w_{ij})$, the matrix of w_{ij}s
x_{ij}	No. of Cjs to be used for storing the Bis

Let us assume that s_i boxes of type i are withdrawn from the inventory during the week. We shall call the vector of s_i as *the sales vector* and denote it by **s**. Therefore, at the end of the week, the inventory will be given by the vector $\mathbf{rs} = \mathbf{a} - \mathbf{s}$, where $rs_i = a_i - s_i$. We shall call this vector **rs** as *the remaining-stock vector*. Since a fresh consignment **a** will be received at the end of the first week, the opening inventory at the beginning of the second week will be given by $\mathbf{a} + \mathbf{rs}$. Let c_j denote the number

Table 12.4 Solution to FIMP set-up version with $\sum_{ij} x_{ij}$ as objective

Box type	**v**	31104	23328	20736	15552	16128	10368
	c	390	534	130	178	52	52
	a	C1	C2	C3	C4	C5	C6
B1	1200	2	1	0	0	0	0
B2	3700	2	17	0	0	0	0
B3	700	2	2	0	0	0	0
B4	1800	3	6	0	0	0	0
B5	200	1	0	0	0	0	0
B6	400	2	1	0	0	0	0
B7	5800	4	41	0	0	0	0
B8	1400	6	2	0	0	0	0
B9	1300	7	0	0	0	0	0
B10	400	2	1	0	0	0	0
B11	400	1	2	0	0	0	0
B12	1800	3	15	0	0	0	0
B13	11800	2	138	0	0	0	0
B14	1800	18	1	0	0	0	0
B15	7400	75	3	0	0	0	0
B16	4400	46	0	0	0	0	0
B17	16600	92	119	0	0	0	0
B18	3700	38	1	0	0	0	0
B19	700	7	1	0	0	0	0
B20	5700	58	2	0	0	0	0
B21	700	0	11	0	0	0	0
B22	400	5	1	0	0	0	0
B23	1100	2	16	0	0	0	0
B24	200	3	0	0	0	0	0
B25	400	0	8	0	0	0	0
B26	1200	2	26	0	0	0	0
B27	300	4	3	0	0	0	0
B28	200	3	1	0	0	0	0

The entries under columns C1, ..., C6 are x_{ij}s

of compartments of type j, $j = 1, 2, \ldots, n$, available in the warehouse to store them, and let **c** denote the vector of c_js. We shall call **c** as *the compartment vector*. Finally, let **v** denote the vector of compartment volumes, where v_j is the volume of the j^{th} compartment in cubic inches, j, $j = 1, 2, \ldots, n$. We shall now provide the solutions to the two versions of FIMP.

5.1 Solution to Set-up Version of FIMP

Since it is the initial stage, all the compartments are assumed to be empty. In this case, we have to store $\sum_i a_i$ boxes in the compartments. The inputs to this problem, **a**, **c**, and **v** are given in Table 12.4. This problem is solved by LINGO (see http://www.lindo.com/), a professional OR Solver. The problem was solved separately, firstly by considering the number of occupied compartments for the objective

Table 12.5 Solution to FIMP set-up version with volume as objective

Box type	**v**	31104	23328	20736	15552	16128	10368
	c	390	534	130	178	52	52
	a	C1	C2	C3	C4	C5	C6
B1	1200	0	2	0	2	0	0
B2	3700	0	1	0	0	22	0
B3	700	3	0	0	1	0	0
B4	1800	0	7	0	3	0	1
B5	200	0	0	0	1	0	1
B6	400	0	2	0	0	2	0
B7	5800	0	3	0	6	22	49
B8	1400	7	0	0	1	0	0
B9	1300	7	0	0	0	0	0
B10	400	0	2	0	0	2	0
B11	400	0	2	0	0	1	1
B12	1800	3	15	0	0	0	0
B13	11800	0	140	0	1	0	0
B14	1800	18	1	0	0	0	0
B15	7400	75	3	0	0	0	0
B16	4400	43	4	0	0	0	0
B17	16600	1	236	0	0	0	0
B18	3700	38	1	0	0	0	0
B19	700	0	10	0	0	0	0
B20	5700	59	0	0	1	0	0
B21	700	0	11	0	0	0	0
B22	400	5	1	0	0	0	0
B23	1100	0	17	0	0	2	0
B24	200	3	0	0	0	0	0
B25	400	0	8	0	0	0	0
B26	1200	2	26	0	0	0	0
B27	300	4	3	0	0	0	0
B28	200	1	3	0	1	0	0

The entries under columns C1, ... ,C6 are x_{ij}s.

function, and secondly by considering the volume of the occupied compartments for the objective function.

The solution with respect to number of occupied compartments as the objective function is given in Table 12.4 (see x_{ij}s under columns labeled C1, C2, ... , C6). The solver took less than a second to produce the solution. Note that the problem is an integer programming problem. The solution obtained is *a global optimal* solution. With respect to this solution, the total number of compartments utilized is 809 (out of 1336) and their total volume is 21,904,992 cubic inches which is 70 % of the total volume of all compartments which is 31,429,440 cubic inches. In terms of number of compartments occupied, 61 % of the compartments are occupied.

When the problem was solved with occupied volume as the objective function, LINGO produced a feasible solution within 2 s, but could not find an optimal solution even after 5 min of CPU running time. The solver was interrupted at the end of 5 min to get the best solution thus far. The solution is produced in Table 12.5. LINGO produces

the lower bound for the objective function and the objective value of the best solution found at each iteration after finding a feasible solution (the algorithm used is Branch and Bound). When the solver was terminated (at the end of the 5th minute), the best solution was 99.99 % close to the lower bound on the minimum objective value. In terms of the volume, the *near optimal solution* produced in Table 12.5 occupied 69 % of the total volume; and in terms of the number of compartments, 66 % of the compartments are occupied (887 out of 1336 compartments). Furthermore, in the first solution with objective function as the number of compartments, only two types of compartments (C1 and C2) were used; whereas in the case of the later solution, five types of compartments are used (C3 type is not used).

5.2 Solution to Dynamic Version of FIMP

From the discussion of the two solutions with two different objective functions above, it is easier to handle the number of compartments objective. This is not only from the view point of computation but also from a practical angle. If more compartments are empty, then the empty compartments can be used for storing any type of boxes. For this reason, we shall take the number of occupied compartments $= \sum_i \sum_j x_{ij}$ as the objective function to be minimized for implementation at the warehouse.

Now, let us consider the dynamic version. Upon receiving the initial consignment **a**, the boxes are stored according to the decision shown in Table 12.4. Note that in this decision, none of the compartments of type C3–C6 is used. After receiving the initial consignment, some of the boxes are retrieved from storage for sales during the week following the consignment. The consignment vector, the sales vector and the remaining-stock vector for these inputs are shown in Table 12.6. Since compartment types C3–C6 were not used for storing, the data and decisions corresponding to these types have been omitted in Table 12.6 to save space.

Consider B1 type boxes. Twelve hundred of these were received in the consignment for storing. From the solution, the 1200 boxes were stored in two C1 type compartments and one C2 type compartment. Note that two C1 compartments and one C2 compartment can accommodate 1296 boxes ($= 2w_{11} + w_{12}$, see Table 12.6 for the values). Since there are only 1200 boxes, there is space for 96 boxes more in those three compartments put together. The question is: Where is this space—in C1 type compartment or in C2 type compartment? The answer to this question depends on the procedure for storing and retrieval. As pointed out in the above paragraph, it will be useful to store and retrieve in such a way that we make room for more empty compartments so that they will be available for storing any type of boxes. Therefore, we lay down the following procedure for retrieval of stored boxes, and for storage of newly arriving consignment of boxes.

Procedure for storing boxes of new consignment arrivals: When a new consignment of boxes arrives at the warehouse for storage, for boxes of each type in it, first start storing them in a compartment partially filled according to some pattern

Table 12.6 FIMP input data and the solution for dynamic version

Box type	**a**	**s**	**rs**	w_{i1}/C1	**v** / **c** / w_{i2}/C2	31104 / 390 / C1	23328 / 534 / C2
B1	1200	852	348	468	360	2	1
B2	3700	3361	339	240	192	2	17
B3	700	616	84	208	160	2	2
B4	1800	1475	325	240	192	3	6
B5	200	181	19	234	180	1	0
B6	400	318	82	156	120	2	1
B7	5800	4095	1705	162	126	4	41
B8	1400	998	402	192	144	6	2
B9	1300	1192	108	192	144	7	0
B10	400	360	40	156	120	2	1
B11	400	383	17	162	126	1	2
B12	1800	1421	379	120	96	3	15
B13	11800	10207	1593	108	84	2	138
B14	1800	1698	102	96	72	18	1
B15	7400	6651	749	96	72	75	3
B16	4400	3874	526	96	72	46	0
B17	16600	14442	2158	90	70	92	119
B18	3700	3580	120	96	72	38	1
B19	700	680	20	90	70	7	1
B20	5700	4220	1480	96	72	58	2
B21	700	667	33	80	64	0	11
B22	400	385	15	72	54	5	1
B23	1100	1079	21	72	60	2	16
B24	200	124	76	72	54	3	0
B25	400	257	143	60	50	0	8
B26	1200	1178	22	54	42	2	26
B27	300	282	18	48	36	4	3
B28	200	135	65	54	42	3	1

with boxes of this type (if one such compartment exists in the warehouse at that time) in the same pattern; until either that compartment is full, or there are no more boxes of this type left to store in this consignment. Repeat the same for boxes of each type in that consignment.

If there are still some more boxes left to store in this consignment, apply the algorithm described in Sect. 4 to find the optimum way of storing the remaining portion of this consignment in the set of empty compartments at this stage.

Procedure for retrieving boxes from storage: For retrieving boxes from storage in the warehouse to fill an order, for each type of box, start retrieving from a compartment partially filled with boxes of this type (if there is one at that time) until it is emptied; and if more boxes of this type are needed, continue retrieving from another compartment in increasing order of compartment capacity. Each time you start retrieving from a compartment, continue until it becomes empty before you retrieve from another compartment.

Note that the above procedures ensure that at any given time, you can have at most one partially filled compartment for each box type (see Exercise 2). Let us now apply this procedure for B1 type boxes and B7 type boxes. As there are 1200 boxes to be stored in two C1 and one C2 compartments and because there are no partially filled compartments at the beginning, fill two C1 compartments (because $w_{11} > w_{12}$) with 936 ($= 2w_{11}$) boxes and then fill C2 with the remaining 264 boxes. This way C2 will be partially occupied ($w_{12} = 360$) with space for 96 more boxes. Now consider B7 boxes. There are 5800 boxes to be stored in 4 C1 and 41 C2 compartments. Following the procedure, store 648 of them in 4 C1 compartments (because $w_{71} > w_{72}$) and the balance 5152 in 41 C2 compartments. With this, there will be 1 C2 compartment partially filled with 112 boxes and having space for 14 more boxes ($w_{72} = 126$).

Consider the retrieval part. During the week, withdrawals occurred according to the sales vector **s** and we are left with the inventory represented by the vector **rs** at the end of the week. The vectors **s** and **rs** are given Table 12.6. Again consider boxes of type B1. Out of 1200 stored initially, 852 of them have been taken out during the week and we are left with 348 boxes at the end of the week. Following the retrieval procedure, the 264 boxes stored in the partially filled C2 compartment are first taken out and then the remaining 588 ($=852-264$) are removed from the two C1 type compartments. Since each C1 type compartment can accommodate 468 B1 boxes, this will leave us with one of the two C1 compartments empty and the other with 348 ($=rs_1$) B1 boxes. Thus, when the next consignment is received one of the C1 compartments is partially full with B1 boxes.

Now, consider the decision making when the new consignment is received for storage at the end of the 1st week. Recall that the consignment vector is same at the end of every week. At this juncture, we need to compute the input data for running the optimization model. After the sales during the first week, each compartment is either empty or full or partially full. If a compartment is partially full, that is, it has at least one box in it and it has space for at least one more box, then we try to *square up (by square up we mean fill completely)* all the partially filled compartments, if possible. For example, in the above paragraph, we observed that one of the C1 compartments is filled with 348 boxes and has room for 120 more boxes (see Table 12.6). Therefore, to square up this partially filled compartment, we take out 120 boxes from the new consignment and put them in this partially filled compartment so that it becomes full. The squaring up might not always be possible. As an example, consider Box type B5. In the initial consignment we received 200 boxes of this type, and of these, 181 were taken out leaving behind 19 boxes in the inventory at the end of the first week. From Table 12.6, we see that the 200 boxes were stored in a C1 compartment which has a capacity of storing 234 B5 boxes (see Table 12.6). After withdrawing 181 boxes from this compartment, we are left with 19 boxes, and when we receive 200 B5 boxes in the consignment at the end of the first week, we will place all of them in the C1 compartment having the 19 B5 boxes. This will leave the C1 compartment with 219 B5 boxes still making the compartment partially filled as $w_{51} = 234$. Therefore, it may not always be possible to square up all the partially filled compartments. See Exercise 3.

After squaring up, let the vector of remaining boxes in the consignment vector be $\bar{\mathbf{a}}$, and let $\bar{\mathbf{c}}$ denote the vector of remaining empty compartments of various types at that time.

Let us examine how to compute $\bar{\mathbf{a}}$ and $\bar{\mathbf{c}}$ vectors. Notice that the compartments that are occupied at the end of the first week are occupied by the boxes given by the remaining-stock vector **rs**. Consider a box type, say B1. We are left with $rs_1 = 348$ boxes of this type at the end of the first week. The compartments that are occupied by these boxes will not be available for storing boxes from $\bar{\mathbf{a}}$ (see Exercise 4). We need to identify the compartment types along with their frequencies that are occupied by these 348 boxes. Let q_{ij} denote the number of j^{th} type compartments that hold the i^{th} type boxes from the remaining-stock vector **rs** at the end of the first week. For $i = 1$, we have $q_{11} = 1$ and $q_{1j} = 0$ for $j = 2, 3, 4, 5, 6$; and for $i = 7$, $q_{71} = 4$, $q_{72} = 9$ and $q_{7j} = 0$ for $j = 3, 4, 5, 6$. Note that for a fixed i, among the compartments filled with boxes from the remaining-stock vector **rs**, there will be at most one compartment that is partially filled. The type of this compartment has to be the one with smallest w_{ij} among the w_{ij}s for which q_{ij}s are positive (see Exercise 5). Let u_i denote the type of this partially filled compartment, if there is one. If there is no partially filled compartment, then set $u_i = 0$. Observe that $u_1 = 1$ and $u_7 = 2$. Let $r_i = \sum_j w_{ij} q_{ij} - rs_i$. Note that r_i is the space left in the partially filled compartment with i^{th} type boxes, if any. That is, the partially filled compartment can take r_i boxes more.

Define the modified consignment vector $\bar{\mathbf{a}}$ as $\bar{a}_i = \max(0, a_i - r_i)$. Let $p_j = \sum_i q_{ij}$. Note that p_j compartments of type j will not be available for storing the boxes from $\bar{\mathbf{a}}$ after squaring up the partially filled compartments at the end of the first week. Define the modified compartment vector $\bar{\mathbf{c}}$ as $\bar{c}_j = c_j - p_j$. We are now ready to act on the new consignment. As soon as we receive the new consignment, we first fill the partially filled compartment containing i^{th} type box with $\min(a_i, r_i)$ boxes of type i, $i = 1, 2, \ldots, 28$ (note that the type of this partially filled compartment is given by u_i provided $r_i > 0$; and this action will be preempted if $r_i = 0$). In this process, the partially filled compartments will either become full or will still remain partially filled. If a compartment containing i^{th} type boxes is still not full, it implies, that $\bar{a}_i = 0$. After completing these filling actions, we now have a problem of deciding on how to fill the empty compartments described by the compartment vector $\bar{\mathbf{c}}$ with the boxes given by the consignment vector $\bar{\mathbf{a}}$. We once again solve the optimization problem described in Sect. 4 with the inputs $\bar{\mathbf{a}}$, $\bar{\mathbf{c}}$, and the **W**-matrix with the objective of minimizing the number of occupied compartments (among $\bar{\mathbf{c}}$).

The above procedure can be repeated each time we receive a new consignment. From the discussion it might appear that the procedure for computing the r_is, u_is and the q_{ij}s is somewhat tedious. But in practice, it will be quite simple as these numbers will be obtained simply from the database. However, for the benefit of the reader, we have created a Microsoft Excel file to compute the r_is, p_js, and q_{ij}s. The reader can access this file on the web site for this book at Springer.com[1]. This file also has a provision to solve the optimization problem posed in Sect. 4 using Microsoft Excel solver.

[1] http://www.springer.com/business+%26+management/operations+research/book/978-1-4939-1006-9

Table 12.7 FIMP input data for dynamic version

Box type	**a**	**s**	**rs**	**r**	**u**	$\mathbf{q}_{.1}$	$\mathbf{q}_{.2}$	$\bar{\mathbf{a}}$
B1	1200	852	348	120	1	1	0	1080
B2	3700	3361	339	141	1	2	0	3559
B3	700	616	84	124	1	1	0	576
B4	1800	1475	325	155	1	2	0	1645
B5	200	181	19	215	1	1	0	0
B6	400	318	82	74	1	1	0	326
B7	5800	4095	1705	77	2	4	9	5723
B8	1400	998	402	174	1	3	0	1226
B9	1300	1192	108	84	1	1	0	1216
B10	400	360	40	116	1	1	0	284
B11	400	383	17	145	1	1	0	255
B12	1800	1421	379	77	2	3	1	1723
B13	11800	10207	1593	51	2	2	17	11749
B14	1800	1698	102	90	1	2	0	1710
B15	7400	6651	749	19	1	8	0	7381
B16	4400	3874	526	50	1	6	0	4350
B17	16600	14442	2158	2	1	24	0	16598
B18	3700	3580	120	72	1	2	0	3628
B19	700	680	20	70	1	1	0	630
B20	5700	4220	1480	56	1	16	0	5644
B21	700	667	33	39	2	0	1	661
B22	400	385	15	57	1	1	0	343
B23	1100	1079	21	51	1	1	0	1049
B24	200	124	76	68	1	2	0	132
B25	400	257	143	37	2	0	3	363
B26	1200	1178	22	32	1	1	0	1168
B27	300	282	18	30	1	1	0	270
B28	200	135	65	43	1	2	0	157

$r_1 = 23, r_2 = 5, \bar{\mathbf{c}} = (300, 503, 130, 178, 52, 52)$; $\mathbf{q}_{.a}$ and $\mathbf{q}_{.2}$ are the first two columns of the matrix (q_{ij}), others being zero

We will now present the solution to the dynamic version of FIMP with input data for the end of first week (equivalently beginning of the second week) as given in Table 12.6. Table 12.7 presents the values of r_is, u_is, p_js, q_{ij}s, $\bar{\mathbf{a}}$ and $\bar{\mathbf{c}}$. The company has to keep track of the partially filled compartments at the end of the first week so that they are first filled as soon as the new consignment arrives. Observe that $\bar{a}_5 = 0$. This is because 19 boxes were left in a C1 compartment at the end of the first week and we receive 200 in the second consignment. Thus, we have a total of 219 B5 boxes at the beginning of the second week stored in a C1 compartment partially filled (C1 can hold 234 B5 boxes). Table 12.8 presents the solution to the dynamic version of the instance under columns C1, C2, and C3. The entry under column "Old Occ" corresponding to Bi is the total number of compartments of all types that filled, partially and fully, with Bis from old stock (rs) when the new consignment is received. The entry under column "New Occ" corresponding to Bi is the total number of compartments assigned by the solution to accommodate the $\bar{a}_i$ boxes of type Bi of the new consignment.

Table 12.8 Solution to FIMP dynamic version

Box type	$\bar{\mathbf{a}}$	C1	C2	C3	New Occ	Old Occ	Tot Occ
B1	1080	1	0	2	3	1	4
B2	3607	0	17	2	19	2	21
B3	576	2	1	0	3	1	4
B4	1693	0	7	2	9	2	11
B5	0	0	0	0	0	1	1
B6	326	0	3	0	3	1	4
B7	5737	0	42	4	46	13	59
B8	1322	5	2	0	7	3	10
B9	1216	5	2	0	7	1	8
B10	284	2	0	0	2	1	3
B11	255	1	0	1	2	1	3
B12	1781	0	18	0	18	4	22
B13	11759	0	140	0	140	19	159
B14	1734	18	0	0	18	2	20
B15	7333	77	0	0	77	8	85
B16	4326	44	2	0	46	6	52
B17	16568	8	227	0	235	24	259
B18	3652	38	0	0	38	2	40
B19	630	7	0	0	7	1	8
B20	5644	59	0	0	59	16	75
B21	663	2	8	0	10	1	11
B22	343	5	0	0	5	1	6
B23	1049	0	17	0	17	1	18
B24	150	2	0	0	2	2	4
B25	357	2	5	0	7	3	10
B26	1168	17	6	0	23	1	24
B27	270	5	1	0	6	1	7
B28	169	0	4	0	4	2	6
				Total	813	121	934

The entries under columns C1, C2, and C3 are new x_{ij}s
Occ = occupancy

6 Summary

In this chapter we have looked an important problem of inventory management of footwear boxes. Two situations were considered—the initial set-up version and the dynamic version. Warehouse keeps receiving material on a weekly consignment basis. Boxes are withdrawn during the week for sales. The optimization problem is that of storing the boxes optimally in the compartments of the warehouse. The problem is formulated as an optimization problem by considering some meaningful practical constraints. The methodology is illustrated with specific situations. Though the problem is discussed for cuboidal boxes to be stored in cuboidal compartments, it can be applied to more general situations provided the weight matrix is given or determined. A Microsoft excel file is provided for the readers to workout exercises. It can also be used as a decision support system if the number of variables and constraints are not exceeding the limitations of the program.

7 Problems

Exercise 1. Describe the warehouse problem where the consignment vector changes in every shipment.

Exercise 2. Prove that the stated procedures will maximize the number of empty compartments. Also, prove that, under the stated procedures for storing and withdrawal, there can be at most one partially filled compartment for any type of boxes.

Exercise 3. Take the compartment capacities as given in Table 12.5 (the **c** vector). Give your own values for a_i and s_i for $i = 7$ and $i = 19$ so that the following happen:

(a) There is no partially filled compartment containing $B7$ boxes,
(b) There is one compartment containing four B19 boxes,
(c) There is one compartment containing B5 boxes that is full,
(d) Both B7 and B19 have only one compartment each that contain 5 and 7 boxes respectively.

Exercise 4. Explain why a partially filled compartment with boxes from the remaining-stock vector **rs** cannot be used for storing anymore boxes from the boxes of $\bar{\mathbf{a}}$.

Exercise 5. Read Sect. 5.2 and do the following exercises.

(a) Show that the q_{ij} defined Sect. 5.2 is positive only if the corresponding $x_{ij} > 0$.
(b) Prove that, under the stated storing and withdrawal procedures, the type j_0 of the compartment partially filled by the i^{th} type of boxes of the remaining-stock vector **rs** at the end of the first week satisfies $w_{ij_0} = \min_j\{w_{ij} : q_{ij} > 0\}$.
(c) Explain how you will compute the vector $q_i = (q_{i1}, q_{i2}, \ldots, q_{i6})$. Compute the vectors q_i for $i = 4, 9$ using data from Table 12.6.
(d) Show that $r_i = \sum_j w_{ij}q_{ij} - rs_i$.

Exercise 6. Simulate a warehouse problem with 35 box types, sales data for 4 weeks and solve the problem for each week.

Exercise 7. Currently for solving the set-up version of the problem, we are using the model: minimize $\sum_i \sum_j x_{ij}$ subject to $\sum_i w_{ij}x_{ij} \geq \bar{a}_i$, $\sum_i x_{ij} \leq \bar{c}_j$, $x_{ij} \geq 0$ and integer for all i, j; where $\bar{a}_i$ and $\bar{c}_j$ are the current updated values of these parameters. In the continuing dynamic version of the algorithm, we are anyway taking full advantage of any partially filled compartments at the time of storing new consignments of boxes that arrive for storage. For this reason, instead of using the above integer programming model, what happens if we just use its LP relaxation obtained by relaxing the integer requirements on all the variables x_{ij} for solving the set-up version of the problem? Also check by obtaining results on the computational performance of this approach.

References

1. Bengtsson, B. E. (1982). Packing rectangular pieces—a heuristic approach. *The Computer Journal, 25,* 353–357.
2. Chung, F. K .R., Garey, M. R., & Johnson, D. S. (1982). On packing two-dimensional bins. *SIAM Journal of Algebraic and Discrete Methods, 3,* 66–76.
3. Coffman, E. G. Jr., Garey, M. R., & Johnson, D. S. (1996). Approximation algorithms for bin packing: A survey. In D. S. Hochhaum (Ed.), *Approximation algorithms for NP-Hard problems.* (pp. 46–93). Boston: PWS Publishing Company.
4. Coffman, E. G. Jr., Galambos, G., Martello, S., & Vigo, D. (1998). Bin packing approximation algorithms: combinatorial analysis. In D. Z. Du & P. M. Paradalos (Eds.), *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers.
5. Das, P. (2005). A heuristic approach for arrangement of footwear boxes to maximize space utilization and related business issues. *International Journal of Management Science, 11*(2), 61–84.
6. Dyckhoff, H. (1990). A typology of cutting and packing problem. *European Journal of Operational Research, 44,* 145–159.
7. Dyckhoff, H., & Finke, U. (1992). Cutting and packing in production and distribution: A typology and bibliography. Heidelberg: Physica-Verlag.
8. El-Bouri, A., Popplewell, N., Balakrishnan, S. & Alfa, A. (1994). A search based heuristic for the two-dimensional bin-packing problem. *INFOR, 32,* 265–274.
9. Haessler, R. W., & Sweeney, P. E. (1991). Cutting stock problems and solution procedures. *European Journal of Operational Research, 54,* 141–150.
10. Hinxman, A. I. (1980). The trim-loss and assortment problems: A survey. *European Journal of Operational Research, 5,* 8–18.
11. Murthy, A. L. N. (2012). Space optimization for warehousing problem: A Methodology for Decision Support System. *Management Science and Financial Engineering, 18*(1), 39–48.
12. Murty, K. G. (1992). *Network programming*. Angelwood Cliffs: Prentice-Hall.
13. Sweeney, P. E., &Paternoster, E. R. (1992). Cutting and packing problems: A categorized, application-orientated research bibliography. *Journal of the Operational Research Society, 43*(7), 691–706.

Chapter 13
Optimal Intake and Routing of Floating Oil Rigs in the North Sea

Dag Haugland and Bjørn Peter Tjøstheim

1 Introduction

1.1 Offshore Oil and Gas Fields Developed by Sub-Sea Solutions and Floating Rigs

For oil and gas fields in production in the North Sea, a key task is to maximize the profit made by recovery and processing of oil/gas reserves. A main source of increased hydrocarbon off-take is through wells, which can be either producers or injectors (gas or water). The choice of technology for field development depends upon key parameters like the size of the field and water depth, and several different layouts are possible. The layouts can coarsely be divided into two categories:

Platform development:	Drilling and processing are carried out from one or more platforms positioned on the seabed.
Sub-sea development:	Sub-sea templates are placed on the seabed with flow lines to a processing platform.

Each sub-sea template has slots from which a well can be drilled. Typically, there are four or six slots per template. As sub-sea development is particularly favorable at large water depths, such as at the Heidrun oil and gas field in the Norwegian Sea, the processing platforms are often floating units. Some fields, such as the Oseberg South oil field in the North Sea, are developed by applying a combination of a platform and a sub-sea solution.

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_13) contains supplementary material, which is available to authorized users.

D. Haugland (✉)
Department of Informatics, University of Bergen, Norway
e-mail: dag@ii.uib.no

B. P. Tjøstheim
Statoil ASA, Postboks 7200, NO-5020 Bergen, Norway
e-mail: bpt@statoil.com

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_13, © Springer Science+Business Media New York 2015

Fig. 13.1 The semi-submersible rig *Songa Delta* operating in the North Sea

Fig. 13.2 The jack-up rig *West Elara* (©Seadrill)

In this text, we focus on sub-sea development, where primarily *semi-submersible* rigs are used for drilling. A semi-submersible rig, such as the *Songa Delta* operating in the North Sea (see Fig. 13.1), is carried by pontoons located below the sea level. Ballast in the lower parts of the pontoons ensures good stability of the rig. The operating deck, connected to the pontoons by structural columns, can thus be located high above the sea level. If not equipped with its own propulsive device, the semi-submersible rig can be tugged to and anchored in the position of the well to be drilled.

In relatively shallow waters (depths up to 200 m), *jack-up* rigs (Fig. 13.2) can be an alternative to the semi-submersible rigs. They are made buoyant by a steel hull to which three or four legs are connected. When the rig is tugged to its target position, its legs are jacked downwards until they reach the seabed. Thus, the rig is anchored and the hull is raised above the sea level.

1.2 Operations in Offshore Oil and Gas Field Exploitation

The main stages of offshore well drilling are:

Plug and abandonment: This stage is needed only if the well slot on the template has been used previously. It involves killing of the old well and setting cement barriers towards the reservoir.

Drilling: A drilling operation is a section based operation, typically starting with a 26" diameter hole section, reducing in size down to 8.5" or 6" in the reservoir where the oil/gas is found. After each section, a steel tube is inserted into the drilling hole in order to avoid collapse of the formation.

Completion: After the reservoir section is drilled, an inner tube, along with valves and packers, are inserted for production/injection. The well is also finally set up for production/injection from the platform. After completion, the well is cleaned up by streaming it toward the platform to remove completion fluids.

The first step in locating a new oil or gas field is through seismic acquisition. If prospects (potential closures) are located, a geological model is established. The final modeling step is to establish a reservoir model where fluid properties are added, together with wells. By means of reservoir simulations, the operator thus gets estimates of the total reserves and production as a function of time for each possible well. A reservoir simulator is a computer model of rock and fluid conditions, which predicts oil and gas recovery per day from each of the wells. Estimates of the total recoverable reserves are thus found by integrating and summing the predicted daily recoveries.

To mention only the few most significant parameters, predictions of well production rates depend upon inner well diameter, reservoir formation, flow properties, reservoir pressure near the well, reservoir size, and the number of wells already producing from the reservoir. In early evaluation studies, the forthcoming well production can only be estimated rather crudely. The production profile of a well is typically divided into two stages:

Plateau stage: The well produces at a stable rate.
Tail production: The production rates are declining.

Both plateau rate and length will vary greatly between wells and reservoirs, with oil production rates ranging from 200 Sm3/day to as much as 5000 Sm3/day, where Sm3 means cube meters at standardized pressure and temperature conditions. The wells will produce either until the pressure is too low for production, or the well is re-drilled for a new reservoir target. The tail production phase will typically last for 5–10 years, often with on/off periods to allow for separation of different flow streams (oil, water, and gas) in the reservoir.

It is important to keep in mind that collected seismic data can be of varying quality and have a resolution limit. This contributes to a degree of uncertainty in geological and simulation models, which in their turns introduce uncertainties in producible volumes.

To prove hydrocarbons and allow for closer estimates of available reserves, a pilot well may be drilled first. Thereafter, the pilot is plugged, and drilling of producers and possible injectors commences.

The amount of time spent on the drilling operation depends upon depth of the reservoir, length and width of the reservoir, geological formations, and many other reservoir characteristics. Thus, the drilling time per well can vary from 20 days up to 150 days, or even more. Plug-and-abandonment and completion typically take in the range of 30–40 days each.

Rig rates correlate with oil prices, as rig owners enjoy profit opportunities through adjustments to the market. With an oil price around 100 USD/bbl, the total cost of renting and operating a semi-submersible rig will typically be around a million USD per day in the Norwegian Shelf market. Consequently, the cost of drilling a well is in tens or even hundreds of million USD (around 100 million USD if drilling takes 100 days).

Maintaining stability during drilling operations requires that the semi-submersible rigs are anchored while drilling. This implies that simultaneous drilling operations on nearby templates are not possible. Drilling operations are also sensitive to ocean heave, which especially during fall and winter causes delays in the drilling operations.

1.3 Rig Hiring

Operators of offshore oil and gas fields often practice a policy of hiring rather than owning the floating rigs from which wells are drilled. For a best possible exploitation of the total portfolio of fields, it is crucial for the operators to determine how many and what rigs to hire at what time. To this end, they have to assess their own future needs as well as the supply at the rig market, With this assessment as input, a schedule for rig hiring can be developed.

Future needs for floating rigs depend on the operator's *covered* and *uncovered* wells. The former category consists of wells that already are assigned to contracted floating rigs. They are contrasted by the uncovered wells, which also are decided to be drilled, but the selection of what rig to drill them is left to be made. All floating rigs need a certification. About 20 certified rigs are currently operative on the Norwegian Continental Shelf in the North Sea, and this number is expected to increase in the near future.

Rigs can either be dedicated to a unique field, or be shared between several fields with one common responsible operator. In some instances, a rig can also be shared between fields operated by different companies. Large fields may very well be visited by more than one rig simultaneously. Decisions on what rigs to hire at what time must be underpinned by drilling schedules for the rigs. At least rough estimates of the floating rigs' geographical positions over time must accompany the decisions on intake, i.e., the schedule for hiring rigs. Decision support tools therefore need to integrate rig hiring and routing in the same model, and suggest solutions for both operations simultaneously.

Assuming that the oil company's policy is to hire rigs, rig ownership is not evaluated. However, much of the analysis in this chapter also applies to rig ownership. It is assumed that rig contracts are time dependent, and that they may contain rig optional periods. Before each rig optional period, the operator can choose to cease the contract at no added cost. Historically, occasional contracts have been based on the well targets to be drilled, but this has not been an option in the later years economic climate. We assume there is no bound on how many rigs an operator can hire at any given time.

1.4 Previous Work

Models for optimizing investments in and operation of offshore oil and gas fields, appear quite frequently in the literature. Recently, Gupta and Grossmann [2] suggested a model for multiple field locations, supporting decisions including drilling, well location, and production. Like the models in many earlier and closely related works [1, 3, 9, 5, 6, 7], their model is defined over multiple periods, and contains both continuous and discrete decision variables. Detailed modeling of the oil and gas reservoirs' response to production schedules lead to complex nonlinear constraints, which in [2] are approached by piecewise linearizations, turning the model into a mixed-integer linear programming problem.

Only a few attempts (e.g., [4]) seem to be made in order to extend existing models for offshore petroleum field planning to the case of floating rigs. In his master thesis [8], Tjøstheim suggests a model supporting decisions related to, e.g., rig intake, routing, and drilling. He also presents an experimental evaluation of the model. Building on Tjøstheim's work, we develop in the current text two models for selecting what floating rigs to be used at what site and at what time. In order to keep focus on rig intake and routing, we prefer to model the production and the marketing of the products with as few details as possible. Hence, we assume each possible well to be characterized by a production profile, and that estimates of the market price are available for the entire life time of the fields. All production profiles are given as input, and specify the daily recovery from each of the wells in each of a set of periods. To capture the life-time of all wells, which may be as long as 20 years, the planning horizon of the models will in typical applications also be a few decades long. In principle, the models suggested in this chapter can be integrated with most of the models in the papers cited above.

The remainder of the chapter is organized as follows. In Sect. 2, we introduce some mathematical notation, and state some basic conditions that any model for the problem in question should fulfill. Integer programming models are suggested in Sects. 3 and 4, and computational experiments illustrating their properties are reported in Sect. 5.

2 Modeling the Rig Intake Problem

We consider the rig intake problem over a fixed time horizon with sufficient length (typically in the magnitude of decades) to cover the life-time of all wells in the actual region. For simplicity, we assume all time periods into which it is subdivided to have equal lengths (typically in tens or hundreds of days).

2.1 Notation and Definitions

Let T denote the number of time periods, and let the periods be associated with the integers $1, \ldots, T$. For convenience, we also define time periods 0 and $T + 1$ as the initial and final periods, respectively, when no drilling or production occurs. The actual value of T is given as the ratio between the length of the planning horizon and the length Δ (measured in number of days) of each period. It is natural to let Δ be in the range of the time it takes to drill a well. If, for instance, the length of the periods is set to 100 days and the horizon is 20 years (see Sect. 1.2), we set the period length to $\Delta = 100$ days, and the number of periods becomes $T = 73$.

We denote the set of possible wells to be drilled by W, and part of our problem is to determine which of the wells in W that should be drilled. The set of rigs that can be hired to drill a well is denoted R, and let $R_w \subseteq R$ denote the rigs that can be used to drill well $w \in W$. Likewise, we define $W_r = \{w \in W : r \in R_w\}$ as the set of wells that rig $r \in R$ possibly can drill. The largest oil fields already in production in the North Sea, such as *Statfjord*, have more than 100 wells, but for most applications in question, it is unrealistic that $|W|$ exceeds 50. Most often, we have $W_r = W$ for all $r \in R$, although it may be possible to reduce W_r, e.g., if some wells are in area with too deep water for rig r to drill there. Since an extension to covered wells is straightforward, we assume for the sake of simplicity that all wells in W are uncovered.

The set R contains all rigs available on the market, which typically count 10–20. Any rig $r \in R$ is, in any time period, either located at some well $w \in W_r$, located at the depot, or on the move between any two locations. The depot, denoted w_0, is an imaginary location in the sense that if a rig is located there, then the rig is either not hired yet, or it has been returned to the owner. A rig arrives at a well only if it is going to drill it, and once the rig has left the well, it never returns. We also assume that once a rig has started drilling a well, it does not stop until drilling is completed. This means that we, in order to limit the scope of the model, disregard the fact that harsh weather conditions can interrupt drilling. Hence, the set of periods in which the rig is located at the well is contiguous. Analogously, the rig leaves and returns to the depot at most once, such that the set of periods that the rig is absent from the depot constitute a contiguous block. We define the total set of locations as $L = W \cup \{w_0\}$, and for any rig $r \in R$, we define the set $L_r = W_r \cup \{w_0\}$ of possible locations.

Concerning drilling and production, we make the following assumptions:

- All wells produce only oil. This assumption is made for the purpose of simplicity, and an extension to a multi-product formulation is straightforward.
- The production from a well, given that it is drilled, is assumed to be known in advance, and represented as a function of the number of time periods elapsed since it was drilled.
- Production, measured in Sm3/day (see Sect. 1.2), is constant within each period.
- Drilling of one well does not affect the production from other wells.
- The operator can exclude the possibility that the well is dry.
- The drilling time of a well is an integer number of periods.

Let p_w^t denote the production from well $w \in W$, t time periods after the well was drilled, and let b_{rw} denote the number of time periods rig $r \in R_w$ needs to drill the well. For any well $w \in W$, let g_{rw}^t denote the cost in USD/day of drilling it from rig $r \in R_w$ in period $t = 1, \ldots, T$. Note that the drilling cost is time dependent, reflecting the fact that the cost includes the rig hire, which correlates strongly with the oil price (see Sect. 1.2). We let $b_{rw} > 0$ denote the number of periods it takes rig r to drill well $w \in W_r$. Consequently, the total drilling cost, before discounting, incurred for well w becomes $\Delta \sum_{k=t}^{t+b_{rw}-1} g_{rw}^k$ if rig r starts drilling the well in period t. Under the same assumption, the production from the well is p_w^k in period $t + b_{rw} - 1 + k$ for $k = 1, 2, \ldots, T$, whereas the production is 0 in periods $1, 2, \ldots, t, \ldots, t + b_{rw} - 1$.

We let the operator's forecast of the oil price, measured in USD/Sm3, in period $t = 1, \ldots, T$ be denoted c^t. By letting α denote the discount factor per time period, we can express the present value of total drilling cost at well w if rig r starts drilling it in period t as $G_{rw}^t = \Delta \sum_{k=t}^{t+b_{rw}-1} (1+\alpha)^{-k} g_{rw}^k$. The corresponding present value of total revenues from the well is $C_{rw}^t = \Delta \sum_{k=t+b_{rw}}^{T} (1+\alpha)^{-k} c^k p_w^{k-t-b_{rw}+1}$. Discounting of costs and revenues is made relatively to time period 0.

For any two possible locations $u, w \in L_r$ of rig r, we define the number of time periods τ_{ruw} that the rig needs to move from u to w. That is, if the rig is at location w in period t, it is not located at $u \neq w$, neither in period t nor in any of the last τ_{ruw} periods $t - \tau_{ruw}, \ldots, t - 1$ before t. Without loss of realism, we assume that the moving times satisfy the triangle inequality, i.e., $\tau_{ruw} \leq \tau_{ruv} + \tau_{rvw}$ for all $r \in R$ and all $u, v, w \in L_r$. Finally, we define m_r^t as the moving cost in USD/day for rig r in time period $t = 1, \ldots, T$, which for the same reason as the drilling cost, is time dependent. It is assumed that the costs depend only indirectly, that is via the time the move takes, on what locations the rig is moving between. The present value of the cost of moving rig r in period t thus equals $M_r^t = \Delta (1+\alpha)^{-t} m_r^t$.

2.2 *Basic Model Properties*

Models for optimizing routing and drilling have to support the following decisions:

- Whether or not to let rig r be located at location u in period t.
- Whether or not to let rig r drill well w in period t.

Decisions concerning location pertain to all time periods $t = 0, 1, \ldots, T, T+1$ and all combinations of r and u such that $u \in L_r$. For the drilling decisions, the corresponding range is $t = 1, \ldots, T$ and $w \in W_r$.

Our models also have to comply with the following logistics conditions:

1. All rigs are located at the depot in periods 0 and $T + 1$.
2. A rig can reside at at most one location at the time.
3. A well can be visited by at most one rig.
4. Between two visits by the same rig to two different locations, at least the number of time periods the rig needs for the move must elapse.
   To make drilling compatible with routing, we impose the conditions:
5. Each well is drilled at most once.
6. A rig visits a well only if it is going to drill it.
7. If a rig started drilling a well less than the number of required drilling periods ago, it must still be located at that well.
8. A rig does not return to the depot until it has drilled all assigned wells.

Different optimization models building on different types of decision variables can be developed in order to accommodate these requirements. Two suggestions are given in Sects. 3 and 4 below.

2.3 Input Data

The following data must be input to the models:

- R = set of rigs,
- W = set of wells,
- for all $r \in R$: W_r = set of wells that can be drilled by rig r,
- T = number of time periods,
- Δ = length of the time periods (days),
- α = discount factor per time period,
- for all $t = 1, \ldots, T$: c^t = oil price in time period t (USD/Sm3),
- for all $r \in R, w \in W_r, t = 1, \ldots, T$: g^t_{rw} = cost of drilling well w from rig $r \in R_w$ in period t (USD/day),
- for all $r \in R, w \in W_r$: b_{rw} = number of time periods rig r needs to drill well w,
- for all $w \in W, t = 1, \ldots, T$: p^t_w = production from well w, t time periods after drilling (Sm3/day),
- for all $r \in R$, $u, w \in W_r \cup \{w_0\}$: τ_{ruw} = number of time periods rig r needs to move from location u to location w,
- for all $r \in R, t = 1, \ldots, T$: m^t_r = cost of moving rig r in period t (USD/day).

3 A Model Formulation Using Drilling and Location Variables

To model the itineraries of the rigs, we introduce in our first integer programming formulation, variables representing the possible arrival of a rig at a location, its presence there, and its departure from the location.

3.1 Definition of Variables and Constraints

For all $r \in R$, $w \in L_r$, and $t = 1, \ldots, T$, we let

$$a_{rw}^t = \begin{cases} 1, & \text{rig } r \text{ arrives at } w \text{ in period } t, \\ 0, & \text{otherwise,} \end{cases}$$

$$y_{rw}^t = \begin{cases} 1, & \text{rig } r \text{ is located at } w \text{ in period } t, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$d_{rw}^t = \begin{cases} 1, & \text{rig } r \text{ leaves } w \text{ in period } t, \\ 0, & \text{otherwise.} \end{cases}$$

For all wells $w \in W_r$ and periods $t = 1, \ldots, T$, we also define the drilling variables

$$x_{rw}^t = \begin{cases} 1, & \text{rig } r \text{ starts drilling well } w \text{ in period } t, \\ 0, & \text{otherwise.} \end{cases}$$

The initial and final rig locations are at the depot, and therefore we define the constants $y_{rw_0}^0 = y_{rw_0}^{T+1} = 1$ and $y_{rw}^0 = y_{rw}^{T+1} = a_{rw}^{T+1} = d_{rw}^{T+1} = 0$ for all $r \in R$ and $w \in W_r$.

We define the arrival period at a well to be the first period that the rig is located there, and the departure period to be the first period the rig no longer resides at the well. For simplicity, we assume arrival and departure to occur in the very start of a period. Consistency between arrival, presence and departure variables dictates that $y_{rw}^t = \sum_{k=1}^t \left(a_{rw}^k - d_{rw}^k\right)$. Hence, $y_{rw}^t = 1$ becomes equivalent to the event that during periods $1, \ldots, t$, rig r has arrived, but not left, well w. Consequently, all location variables y_{rw}^t can be eliminated through substitution by the above consistency relations, but they will be kept in our model formulations in order to simplify explanations.

Since a rig arrives at a well exclusively for the purpose of drilling it, and each well is drilled at most once, it follows that at most one arrival occurs at each well, and thus, we impose $\sum_{r \in R_w} \sum_{t=1}^T a_{rw}^t \leq 1$ for all $w \in W$. Drilling is not possible before arrival, yielding $\sum_{k=1}^t x_{rw}^k \leq \sum_{k=1}^t a_{rw}^k$, and departure is allowed only when drilling is done, suggesting $\sum_{k=1}^t d_{rw}^k \leq \sum_{k=1}^{t-b_{rw}} x_{rw}^k$.

Constraints introduced so far, model necessary relations between a-, x-, d-, and y-variables for the wells. At the depot, departure occurs prior to arrival (return),

and rig r is absent in period t if it has already left but not returned. This yields $1-y^t_{rw_0} = \sum_{k=1}^{t}\left(d^k_{rw_0} - a^k_{rw_0}\right)$. To ensure that the rig completes all drilling activities before its return, we impose the constraint $\sum_{k=1}^{T} d^k_{rw_0} \le 1$, stating that at most one tour is allowed per rig.

Finally, because rig r needs τ_{ruw} periods to move from location $u \in L_r$ to location $w \in L_r$, we cannot have $y^k_{ru} = y^t_{rw} = 1$ if $t - \tau_{ruw} \le k \le t$, and for all such k and t, we therefore impose $y^k_{ru} + y^t_{rw} \le 1$.

We arrive at the following model:

$$\max_{x,y} \sum_{r\in R}\sum_{w\in W_r}\sum_{t=1}^{T}\left(C^t_{rw} - G^t_{rw}\right)x^t_{rw} - \sum_{r\in R}\sum_{t=1}^{T} M^t_r\left(1 - \sum_{w\in L_r} y^t_{rw}\right) \tag{13.1}$$

$$\text{s.t.} \quad y^k_{ru} + y^t_{rw} \le 1 \quad r \in R; u, w \in W_r; u \ne w; \quad t = 1,\dots,T+1; k = t-\tau_{ruw},\dots,t \tag{13.2}$$

$$y^t_{rw} = \sum_{k=1}^{t}\left(a^k_{rw} - d^k_{rw}\right) \quad r \in R; w \in W_r; t = 1,\dots,T+1 \tag{13.3}$$

$$1 - y^t_{rw_0} = \sum_{k=1}^{t}\left(d^k_{rw_0} - a^k_{rw_0}\right) \quad r \in R; t = 1,\dots,T+1 \tag{13.4}$$

$$\sum_{k=1}^{t} d^k_{rw} \le \sum_{k=1}^{t-b_{rw}} x^k_{rw} \quad r \in R; w \in W_r; t = 1,\dots,T \tag{13.5}$$

$$\sum_{k=1}^{t} x^k_{rw} \le \sum_{k=1}^{t} a^k_{rw} \quad r \in R; w \in W_r; t = 1,\dots,T \tag{13.6}$$

$$\sum_{r\in R_w}\sum_{t=1}^{T} a^t_{rw} \le 1 \quad w \in W \tag{13.7}$$

$$\sum_{t=1}^{T} d^t_{rw_0} \le 1 \quad r \in R; \tag{13.8}$$

$$a^t_{rw}, d^t_{rw} \in \{0,1\} \quad r \in R; w \in L_r; t = 1,\dots,T+1 \tag{13.9}$$

$$y^t_{rw} \in \{0,1\} \quad r \in R; w \in L_r; t = 0,\dots,T+1 \tag{13.10}$$

$$x^t_{rw} \in \{0,1\} \quad r \in R; w \in W_r; t = 1,\dots,T \tag{13.11}$$

It is easily verified that each of the conditions 1–8 in Sect. 2.2 is satisfied by model (13.1)–(13.11). This is accomplished either by a dedicated constraint or by combination of several of the suggested constraints. Condition 1 follows directly from $y^0_{rw} = y^{T+1}_{rw} = 0$, condition 4 follows from (2), and condition 5 follows by combining (13.6) and (13.7). The constraint $y^t_{rw} \ge \sum_{k=t-b_{rw}+1}^{t} x^k_{rw}$, stating that rig r must be located at well w in period t if it started drilling it in any of the periods

$\{t - b_{rw} + 1, \ldots, t\}$ (condition 7), follows immediately by combining (13.3) with the difference between (13.6) and the reverse of (13.5).

By summing (13.3) over all $r \in R_w$, and utilizing non-negativity of all a- and d variables, we get from (13.7) that $\sum_{r \in R_w} y_{rw}^t = \sum_{r \in R_w} \sum_{k=1}^{t} \left(a_{rw}^k - d_{rw}^k\right) \leq \sum_{r \in R_w} \sum_{k=1}^{T} a_{rw}^k \leq 1$. Hence, the constraint $\sum_{r \in R_w} y_{rw}^t \leq 1$, stating that at most one rig can visit well w in period t (condition 3), is implied by a linear combination of other constraints.

From (13.3), we get $y_{rw}^t = \sum_{k=1}^{t} \left(a_{rw}^k - d_{rw}^k\right) \leq \sum_{k=1}^{T+1} a_{rw}^k - \sum_{k=1}^{t} d_{rw}^k = y_{rw}^{T+1} + \sum_{k=1}^{T+1} d_{rw}^k - \sum_{k=1}^{t} d_{rw}^k = 0 + \sum_{k=t+1}^{T+1} d_{rw}^k \leq \sum_{k=1}^{T} d_{rw}^k \leq \sum_{k=1}^{T-b_{rw}} x_{rw}^k$, where (13.5) is used to obtain the last inequality. Hence, rig r visits well w only if it drills it (condition 6).

Let $1 \leq t' < t \leq T$. Combining condition 7 with $a_{rw_0}^k \geq 0$ and (13.4), yields $x_{rw}^t + \sum_{k=1}^{t'} a_{rw_0}^k \leq y_{rw}^t + \sum_{k=1}^{t} a_{rw_0}^k = y_{rw}^t + \sum_{k=1}^{t} d_{rw_0}^k - 1 + y_{rw_0}^t \leq \sum_{k=1}^{t} d_{rw_0}^k \leq 1$, where (13.2) and (13.8), respectively, are used to obtain the last two inequalities. Consequently, rig r cannot arrive at the depot in one of the first t' periods and also drill some well w in a later period $t > t'$ (condition 8).

In combination with the integrality constraints $y_{rw}^t \in \{0, 1\}$, constraints (13.2) imply $\sum_{w \in L_r} y_{rw}^t \leq 1$, i.e., an inequality stating that a rig visits at most one location at the time (condition 2). However, this inequality is not implied when integrality is relaxed, and therefore the continuous relaxation of the model may be strengthened if the inequality is added. We will revert to this issue in a discussion of strong valid inequalities in Sect. 3.3.

3.2 A Constraint Inclusion Scheme

In instances with fine time discretization, particularly if most well sets W_r contain nearly all wells in W, the number of constraints (13.2) grows large. Since it is unlikely that a large proportion of the constraints will be binding at the optimal solution, neither in the IP-model nor in its continuous relaxation, we suggest a procedure for introducing them gradually. First, we solve the LP-problem (13.1)–(13.8) with all constraints (13.2) excluded. All excluded constraints that are violated in the optimal solution are then added to the model, and will not be removed later. The procedure is repeated until none of the excluded constraints are violated, and the optimal solution to the LP-relaxation of (13.1)–(13.11), hence is found.

Next, the integrality constraints (13.9)–(13.11) are introduced, and the resulting integer program is solved by a branch-and-bound algorithm. Again, constraints of type (13.2) that are violated in the IP-solution are added, and analogously to the procedure for solving the LP-relaxation, we solve a sequence of IP-problems until no violation of (13.2) is observed. The constraint inclusion scheme is summarized in Algorithm 1.

Algorithm 1 A constraint inclusion scheme

Let LP be the linear program defined by (1), (3)–(8), and (9)–(11) with $\{0,1\}$ replaced by $[0,1]$.
repeat
  Solve LP
  Extend LP by all constraints (2) violated by the current solution
until all constraints in the LP-relaxation of the original model are satisfied by the current solution
Let IP be the integer program obtained by adding the integrality constraints (9)–(11) to LP
repeat
  Solve IP
  Extend IP by all constraints (2) violated by the current solution
until all constraints in the original model are satisfied by the current solution

3.3 Strengthening the Formulation: Clique Constraints

In the sense that constraint (13.2) says that at most one of them can take the value 1, a relation of mutual exclusion applies between the variables y^k_{ru} and y^t_{rw}. If the same relation applies between some other variable z and both y^k_{ru} and y^t_{rw}, it follows that $y^k_{ru} + y^t_{rw} + z \leq 1$ is a valid inequality, since at most one of the three variables can take the value 1.

More generally, assume that $\Pi_r \subseteq L \times \{0, \ldots, T+1\}$ is a set of pairs consisting of a location and a period, such that $\left\{y^t_{rw}\right\}_{(w,t)\in\Pi_r}$ is a set of mutually exclusive variables. Then the constraint $\sum_{(w,t)\in\Pi_r} y^t_{rw} \leq 1$ is valid. Such constraints are commonly referred to as *clique* constraints, since in a graph with the vertices corresponding to variables and edges corresponding to the exclusion relation, the set $\left\{y^t_{rw}\right\}_{(w,t)\in\Pi_r}$ corresponds to a clique. Ideally, the cliques are *maximal*, meaning that there exists no variable outside the clique that is mutually exclusive with all variables in the clique.

If we identify all the maximal cliques, and add the corresponding constraint, we arrive at a stronger formulation. However, finding all maximal cliques is in itself an NP-hard problem, and we therefore consider this approach computationally infeasible. As a compromise, we rather maintain a set Q_r of (not necessarily maximal) cliques for each $r \in R$. In the course of the algorithm, the initially empty sets of cliques are updated as follows:

Once a violated constraint $y^k_{ru} + y^t_{rw} \leq 1$ is observed, we add two new cliques K^k and K^t to Q_r. Both of these contain the two variables $\left\{y^k_{ru}, y^t_{rw}\right\}$, but K^k also contains all variables y^k_{rv} that are mutually exclusive with y^t_{rw}, where $v \in L_r \setminus \{u, w\}$. That is, $K^k = \left\{y^t_{rw}\right\} \cup \left\{y^k_{rv} : \tau_{rvw} \geq t-k, v \in L_r \setminus \{w\}\right\}$, and analogously, we let $K^t = \left\{y^k_{ru}\right\} \cup \left\{y^t_{rv} : \tau_{ruv} \geq t-k, v \in L_r \setminus \{u\}\right\}$. It follows that if $k = t$, in which case the two cliques coincide, the corresponding clique inequality becomes $\sum_{v\in L_r} y^k_{rv} \leq 1$, stating that at most one well is visited by rig r in period k.

We also extend all cliques in Q_r by y^k_{ru} if y^k_{ru} is mutually exclusive with all variables currently in the clique. The same procedure is applied to y^t_{rw}. To the problem defined by (13.1) and (13.3)–(13.11), we then add the clique constraints for all cliques in Q_r, except those that are subsets of other cliques in Q_r. The resulting clique constraint generation scheme is given by Algorithm 2.

Note that the algorithm is in either of its two possible modes: LP and IP. Initially, the mode is LP, indicating that Problem P is a linear program. Once the optimal solution to P satisfies all clique constraints, the integrality constraints are introduced, and the mode is set to IP.

Practical implementations of Algorithm 2 should avoid generation of clique constraints that obviously are implied by others. That is, if $K' \subseteq K$ and $K', K \in Q_r$, we do not add the constraint corresponding to K'. It is still worthwhile to keep K' in Q_r in case K' is extended later in the process.

Algorithm 2 A clique constraint generation scheme

for all $r \in \mathrm{R}$ **do**
  $Q_{\mathrm{r}} \leftarrow \emptyset$
Let P be the problem defined by (1), (3)–(8), and (9)–(11) with $\{0,1\}$ replaced by $[0,1]$.
done $\leftarrow$ **false**, mode $\leftarrow$ LP
repeat
  Add the clique constraints corresponding to $\bigcup_{\mathrm{r}\in\mathrm{R}} Q_r$ to Problem P and solve the problem
  for all $r \in R$ **do**
    for all $u, w \in W_r$ $(u \neq w)$ and $t, k \in \{0, \ldots, T+1\}$ where $t - \tau_{ruw} \leq k \leq t$ **do**
      if $y^k_{ru} + y^t_{rw} > 1$ **then**
        for all $K \in \mathrm{Q_r}$ **do**
          if y^k_{ru} is mutually exclusive with all $y \in K$ **then**
            $K \leftarrow K \cup \{y^k_{ru}\}$
          if y^t_{rw} is mutually exclusive with all y ∈ K **then**
            $K \leftarrow K \cup \{y^t_{rw}\}$
        $K^k \leftarrow \{y^t_{rw}\} \cup \{y^k_{rv} : \tau_{\mathrm{rvw}} \geq t - k, v \in L_r \setminus \{w\}\}$
        $K^t \leftarrow \{y^k_{ru}\} \cup \{y^t_{rv} : \tau_{\mathrm{ruv}} \geq t - k, v \in L_r \setminus \{u\}\}$
        $Q_r \leftarrow Q_r \cup \{K^k, K^t\}$
  if all constraints (2) are satisfied **then**
    if mode=LP **then**
      add the integrality constraints (9)–(11) to P
      mode $\leftarrow$ IP
    else
      done $\leftarrow$ **true**
until done

3.4 Valid Inequalities

Special cases of the clique constraints discussed in the previous section are generated by the cliques $K^t_r = \{y^t_{rw} : w \in L_r\}$, reflecting the fact that no rig can visit more than one location at the time. These inequalities can be strengthened in two different ways.

Let $\tau^{\mathrm{out}}_{ru} = \min\{\tau_{ruw} : w \in L_r \setminus \{u\}\}$ be the minimum number of time periods rig r needs to move from location u to any other location. If rig r visits some location in period t, it cannot have left location u during the last τ^{out}_{ru} periods. Further, the rig cannot leave both location u_1 in period k_1 and location u_2 in period k_2 if $t - \tau^{\mathrm{out}}_{ru_1} < k_1 \leq t$ and $t - \tau^{\mathrm{out}}_{ru_2} < k_2 \leq t$. It follows that all variables

in $K_r^t \cup \left\{ d_{ru}^k : u \in L_r, k > t - \tau_{ru}^{\text{out}}, k \leq t \right\}$ are mutually exclusive, suggesting the constraint

$$\sum_{u \in L_r} \left(y_{ru}^t + \sum_{k=t-\tau_{ru}^{\text{out}}+1}^{t} d_{ru}^k \right) \leq 1. \tag{13.12}$$

In an analogous way, we get that

$$\sum_{u \in L_r} \left(y_{ru}^t + \sum_{k=t}^{t+\tau_{ru}^{\text{in}}-1} a_{ru}^k \right) \leq 1 \tag{13.13}$$

is a valid inequality, where $\tau_{ru}^{\text{in}} = \min \{\tau_{rwu} : w \in L_r \setminus \{u\}\}$.

4 A Dynamic Network Flow Model with Variables Representing Moving and Drilling

As an alternative to decisions representing the actual location of a rig, we introduce variables stating whether or not a specific move is initiated in a given period. The new variables can thus be considered as flow in a network where each node corresponds to a unique pair of a location and a time period. Any two pairs of nodes are directly connected if some rig can be moved between the corresponding locations in time equal to the difference between the two periods. Following a multi-commodity network flow approach, each rig plays the role of a flow commodity.

4.1 Definition of Variables and Constraints

We let the drilling variables x_{rw}^t be defined as in Sect. 3.1, and introduce the new flow variables

$$z_{ruw}^t = \begin{cases} 1, & \text{rig } r \text{ starts moving from location } u \text{ to location } w \text{ in period } t, \\ 0, & \text{otherwise.} \end{cases}$$

This decision variable is also defined for $u = w$, with the interpretation that z_{rww}^t indicates whether or not rig r resides at location w in period t. Let $\tau_{rww} = 0$ for all $r \in R$ and $u \in L_r$. Hence, if $z_{rww}^t = 1$, we also have to ensure that $z_{rwu}^{t+1} = 1$ for some $u \in L_r$, including the possibility that $u = w$.

To comply with the initial and final conditions that the rigs are located at the depot, we define as constants $z_{rw_0w_0}^0 = z_{rw_0w_0}^{T+1} = 1$, $z_{rw_0w}^0 = z_{rw_0w}^{T+1} = 0$ and $z_{ruw}^t = 0$ for all $u \in L_r$, $w \in W_r$, $t > T + 1 - \tau_{ruw}$.

The model becomes:

$$\max_{x,y} \sum_{r\in R}\sum_{w\in W_r}\sum_{t=1}^{T}\left(C^t_{rw}-G^t_{rw}\right)x^t_{rw} - \sum_{r\in R}\sum_{t=1}^{T} m^t_r\left(1-\sum_{w\in L_r} z^t_{rww}\right) \tag{13.14}$$

$$\text{s.t.} \quad \sum_{r\in R_w}\sum_{t=1}^{T} x^t_{rw} \le 1 \qquad w\in W \tag{13.15}$$

$$\sum_{r\in R_w}\sum_{u\in L_r}\sum_{t=1}^{T} z^t_{ruw} \le 1 \qquad w\in W \tag{13.16}$$

$$\sum_{u\in L_r} z^t_{rwu} = \sum_{u\in L_r} z^{t-\tau_{ruw}-1}_{ruw} \qquad r\in R; w\in L_r; t=1,\ldots,T+1 \tag{13.17}$$

$$z^t_{rww} \ge \sum_{k=t-b_{rw}+1}^{t} x^k_{rw} \qquad r\in R; w\in W_r; t=1,\ldots,T \tag{13.18}$$

$$x^t_{rw}\in\{0,1\} \qquad r\in R; w\in W_r; t=1,\ldots,T \tag{13.19}$$

$$z^t_{ruw}\in\{0,1\} \qquad r\in R; u,w\in L_r; t=0,\ldots,T+1 \tag{13.20}$$

Similarly to what we did in the model in Sect. 3.1, we use $1-\sum_{u\in L_r} z^t_{ruu}$ to indicate whether or not rig r is moving in period t. Constraints (13.15) and (13.16) state that each well is drilled and arrived at, respectively, at most once. The flow conservation constraints (13.17) ensure that in period t, rig r leaves location w or remains there for yet another period, if and only if it started a move from some other location u toward location w $\tau_{ruw}+1$ time periods earlier. The concept of a move is here generalized to include "moves" where the current position is the target, i.e., that the rig remains in its current position. Constraints (13.18) reflect that rig r must be located at well $w\in W_r$ in period t if it starts drilling it in one of the periods $t-b_{wr}+1,\ldots,t$.

By $z^0_{rw_0w_0} = z^{T+1}_{rw_0w_0} = 1$, condition 1 in Sect. 2.2 is satisfied. For all $t>0$, the conservation of flow constraints (13.17) imply, when combined with the initial conditions $z^0_{rw_0w_0}=1$, that $z^t_{rww}=1$ for at most one well w (condition 2). The same constraints also imply that a rig arrives at location w, $\tau_{ruw}+1$ periods after the start of the move from location u (condition 4). Constraints (13.15) ensure that each well is drilled at most once, and thereby by at most one rig (condition 5). Since the moving times τ satisfy the triangle inequality, no rig will visit a well unless it is going to drill it (condition 6), and consequently, all wells are visited by at most one rig (condition 3). The triangle inequality also implies that a rig does not return to w_0 until it has drilled all its wells (condition 8). Finally, condition 7 follows directly from (13.18).

4.2 Valid Inequalities

The fact that arrival, drilling and departure occur in the said order, can be exploited to derive valid inequalities also for the network flow model. Rig r arrives at well

$w \in W_r$ as early as period t if and only if $\sum_{u \in L_r \setminus \{w\}} \sum_{k=1}^{t-\tau_{ruw}-1} z_{ruw}^k = 1$. Likewise, it has finished drilling the well as early as period t if and only if $\sum_{k=1}^{t-b_{rw}} x_{rw}^k = 1$. We thus get the valid inequalities

$$\sum_{k=1}^{t} x_{rw}^k \leq \sum_{u \in L_r \setminus \{w\}} \sum_{k=1}^{t-\tau_{ruw}-1} z_{ruw}^k \qquad r \in R, w \in W_r, t = 1, \ldots, T, \tag{13.21}$$

$$\sum_{u \in L_r \setminus \{w\}} \sum_{k=1}^{t} z_{rwu}^k \leq \sum_{k=1}^{t-b_{rw}} x_{rw}^k \qquad r \in R, w \in W_r, t = 1, \ldots, T. \tag{13.22}$$

These inequalities are satisfied by any feasible solution to (13.14)–(13.20), but not necessarily by feasible solutions to the corresponding LP-relaxation. At the computational expense of larger LP-problems to be solved, sharper upper bounds on the optimal objective function value can be obtained by adding (13.21)–(13.22) to our network flow model.

5 Computational Experiments

In order to illustrate the performance of the two integer programming models we have developed for the rig intake and routing process, we present in this section results from computational experiments. Both models are implemented in the modeling language AMPL, and CPLEX 10.1 is used to solve the problems. All experiments are run on a 2.4 GHz Intel(R) processor with 2 GB RAM.

We present results from experiments exploring the following alternatives:

- For the location model in Sect. 3: Either solve it directly by submitting the complete model (13.1)–(13.11) to CPLEX, or apply the clique constraint generation scheme of Sect. 3.2. Both these alternatives are tried with and without addition of the valid inequalities (13.12)–(13.13). If the strategy is to add the valid inequalities, all of them are added in the very beginning of the process. That is, we add all of (13.12)–(13.13) before CPLEX is invoked (direct approach) or before Algorithm 2 is invoked (clique constraint generation approach).
- For the flow model in Sect. 4: Solve it directly by submitting (13.14)–(13.20) to CPLEX, with and without addition of the valid inequalities (13.21)–(13.22).

We do not report results from Algorithm 1, as preliminary experiments indicated that this approach performs poorly in comparison to the alternatives above.

Hence, we are left with four alternative ways to apply the model with location variables, and two alternative ways to apply the dynamic network flow model. We do not claim to make an exhaustive comparison of the various alternatives, and our observations do not necessarily carry over to any instance of the problem. The purpose of the experiments is rather, by use of a few selected numerical examples, to illustrate the effect of applying a clique constraint generation scheme, adding valid

Table 13.1 Size of test instances

Instance	$\|W\|$	$\|R\|$	T
A	3	2	5
B	8	4	10
C	8	4	20
D	20	8	40

inequalities, and switching to another model formulation. To this end, we report for each alternative the time it takes for the solver to compute the optimal solution, and the total number of branch-and-bound nodes generated.

Since all our real rig data from the North Sea are confidential, we have generated test instances on the basis of simulated data. The size of each generated instance, reported in Table 13.1, is within the range of what is realistic in practical applications in the geographical area in question. In all runs, the discount factor is 0.05 per time period.

In Table 13.2, we report the running time needed to complete each of the runs. If the solution is not found within 30 CPU minutes, we interrupt execution. In such instances, the entry in Table 13.2 is given in parentheses, and reports the upper bound on the optimal objective function value relative to its true value. For the location model developed in Sect. 3, columns 2–3 show the results obtained by submitting (13.1)–(13.11) to CPLEX, and results obtained by applying Algorithm 2 are displayed in columns 4–5. Columns 2 and 4 (3 and 5) show results without (with) the strategy of adding (13.12)–(13.13) to the model. Results from the dynamic network flow model are given in columns 6 (without addition of (13.21)–(13.22)) and 7 (with addition of (13.21)–(13.22)), respectively.

In Table 13.3, organized analogously, we report the total number of nodes in the branch-and-bound trees as reported by CPLEX, where the entries corresponding to clique constraint generation are found as the total number of nodes accumulated over all CPLEX runs. For each of the two phases of the clique constraint generation scheme, Table 13.4 shows the number of iterations and the total number of clique constraints generated. The columns with header "mode = LP" give the number of iterations (the number of LPs solved) and the number of clique constraints generated in the first phase, i.e., before the integrality constraints are introduced. Correspondingly, the columns with header "mode = IP" give the number of iterations (the number of integer programs solved) and the number of clique constraints generated after introducing the integrality constraints.

We observe from Table 13.2 that without the redundant valid inequalities (13.12)–(13.13), the largest instance (D) cannot be solved within 30 min if the first model is used. This applies both to the direct approach where the complete model (13.1)–(13.11) is submitted to CPLEX, and to the clique constraint generation scheme. Further, the direct approach is the better, as the instances that can be solved (A–C) are solved faster, and in the unsolved instance (D), the bound on the optimal profit is sharper (5 % versus 11 % above the true value).

However, when (13.12)–(13.13) are added, the running time goes down considerably, and also instance D becomes solvable. The direct approach remains the faster

Table 13.2 Results from experiments—running time in CPU seconds with and without the strategy of adding valid inequalities (v.i.)

	Location model					
Inst-	Solve model (13.1)–(13.11) directly		Clique constraint generation		Flow model	
ance	Without v.i.	With v.i.	Without v.i.	With v.i.	Without v.i.	With v.i.
A	0.0	0.0	0.2	0.0	0.1	0.0
B	0.5	0.1	19.1	0.8	0.2	0.1
C	6.1	2.0	65.0	4.7	0.7	0.5
D	(1.05)	722.6	(1.11)	575.7	172.2	36.8

Table 13.3 Results from experiments—number of nodes in the branch-and-bound search tree with and without the strategy of adding valid inequalities (v.i.)

	Location model					
Inst-	Solve model (13.1)–(13.11) directly		Clique constraint generation		Flow model	
ance	Without v.i.	With v.i.	Without v.i.	With v.i.	Without v.i.	With v.i.
A	0	0	0	0	0	0
B	76	0	26	0	0	0
C	131	0	162	0	18	0
D	1962	848	0	5291	4543	13

Table 13.4 Computational effort of the clique constraint generation scheme with and without the strategy of valid inequalities (v.i.)

	Number of iterations				Number of clique constraints generated			
Inst-	Without v.i.		With v.i.		Without v.i.		With v.i.	
ance	mode = LP	mode = IP	mode = LP	mode = IP	mode = LP	mode = IP	mode = LP	mode = IP
A	2	1	1	1	58	0	0	0
B	4	1	5	2	663	0	54	53
C	2	2	4	4	1220	169	50	6
D	2	0	36	9	7932	0	186	137

for the three smaller instances, but in the more challenging instance D, the clique constraint generation scheme computes the optimal solution more quickly. The main reason seems to be in the size of the LP-problems to be solved, as the total number of branch-and-bound nodes is larger for the clique constraint generation scheme. Although as many as 36 linear programs and 9 integer programs (see Table 13.4) have to be solved before all necessary constraints are generated, this approach is faster than solving a complete instance of (13.1)–(13.11).

Table 13.2 shows that by using the location formulation without adding the suggested valid inequalities, the optimal solution of case D was not found within the time limit. However, the solver did manage to identify a feasible solution, and its objective function value turned out to be no more than 0.8 % below the optimal one. To compare the two solutions to case D, we have given their drilling schedules in Tables 13.5 (for the optimal solution) and 13.6 (for the best solution found with the location model). The wells are denoted a–t, and non-blank entries in Tables 13.5–13.6 indicate that in the period corresponding to the column, the rig corresponding to the row is busy drilling the given well. For instance, the two solutions agree that rig 2 should drill well c in periods 11–12. All drilling is completed during the first 28 periods in both solutions.

Table 13.5 Optimal drilling schedules for case D. *Non-blank entries* are well identifiers, signifying that the well is to be drilled by the rig corresponding to the row in the time period corresponding to the column

	Time period																									
Rig	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1					j																							
2											c	c																
3			h				a				d	d				s	s		l	l		e	e				g	g
4			r	r			i				b					f	f											
5					o	o			p																			
6				t	t				m																			
7										k	k						n	n	n	n								
8					q	q																						

Table 13.6 Best drilling schedules for case D found by means of the location formulation within 30 CPU minutes. *Non-blank entries* are well identifiers, signifying that the well is to be drilled by the rig corresponding to the row in the time period corresponding to the column

	Time period																									
Rig	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1							j																					
2											c	c																
3			h		s	s		l	l		e	e			a						d	d					g	g
4						f	f				b				i									r	r			
5					o	o			p																			
6				t	t				m																			
7										k	k						n	n	n	n								
8					q	q																						

We observe from the tables that there is no difference in the drilling schedules for rigs 2 and 5–8, and for rig 1, the only difference is that the drilling suggested by the location model starts two periods later than what is optimal. For the two busiest rigs, 3 and 4, there are larger differences, but also for these rigs, the two solutions agree in the allocation of wells to rigs. The only failure of the nonoptimal solution thus lies in the sequencing and timing of the drilling operations.

Further improvement of the performance is achieved by moving to the dynamic network flow formulation. Even without the redundant constraints (13.21)–(13.22), the solution to all instances can be found within the time limit. When they are added, all instances are solved in less than one CPU minute. For the largest instance, the speed-up relative to the fastest approach based on the location model is by more than a factor 15. A branch-and-bound search tree of very modest size, containing only 13 nodes, is sufficient to solve the instance.

6 Conclusions

Realistic instances of the rig intake and routing problem considered in this chapter can be solved by means of either a location model or a dynamic network flow model. However, submitting plain variants of either of the models to an IP-solver does not

seem to be a viable approach. Large instances of the first model, particularly instances with a fine time resolution, are likely to be solved faster by application of a clique constraint generation scheme. Also, the continuous relaxations of both models can be strengthened by adding redundant valid inequalities, and significant speed-up can be expected.

The purpose of this chapter has been to demonstrate the basic principles of modeling floating oil rigs. In practical use, the models would be extended by several features. We left them unmentioned for the reason of a focused study, but most extensions in question are straightforward to accomplish, regardless of which of our two models is chosen. Possible extensions include:

- Drilling operations can in some cases be split between two (in rare cases, more than two) rigs.
- Precedence relations, possibly with a time lag, frequently appear between drilling operations. This applies in particular when an injection well is drilled in order to support the production from an oil well. Drilling the injection well becomes the predecessor, and drilling the oil producing well becomes the successor in the given relation.
- Time windows may apply, especially to wells in area with seasonally rough weather conditions. Correspondingly, rigs may have to be routed differently in winter than in the summer season, suggesting that moving times (in addition to costs) be time dependent.

7 Practical Exercises

7.1 Heuristic Methods

Observe that in the location model given in Sect. 3, only constraints (13.7), stating that at most one rig arrival is allowed at any well, involve summation over a set of rigs. If we replace the constraint by $\sum_{t=1}^{T} a_{rw}^t \leq 1$ for all $w \in W$ and $r \in R_w$, i.e., by the requirement that *each rig* may arrive at a well at most once, the entire location model decouples by rigs. Thus, we get one integer program per rig.

Solving $|R|$ smaller problems is likely to be considerably faster than solving the original problem. The solution is not necessarily feasible, as wells may be drilled by more than one rig. However, we can use this solution in order to construct feasible solutions, and hopefully with good quality, as follows: First, we identify the rig that, in the solution to the decoupled problem, provides the largest profit. We fix the route and the schedule of this rig (i.e., all decision variables with this particular rig index) as suggested by the solution. All wells that the rig visits are allocated to it, and thereby excluded from further decision making. Next, we repeat the procedure with the remaining rigs and wells, and stop when all rigs have been scheduled or it becomes optimal to let all remaining rigs be idle in every period.

In this exercise, you should elaborate on this idea, and check whether good solutions can be obtained fast. One or more of the following approaches should be considered:

1. Apply the heuristic to the location model as indicated above by submitting the decoupled problems to an IP-solver.
2. Observe that each clique constraint discussed in Sect. 3.3 involves only one rig. Thus, an alternative approach is to apply the heuristic above by using Algorithm 2 as solver for the decoupled problems.
3. Apply the same idea to the flow model by replacing (13.15)–(13.16) by constraints applying to each rig individually, rather than the sum over rigs.
4. Like approach 3, but with the addition of the valid inequalities (13.21)–(13.22).

When applied to case D, do these approaches vary much in terms of computational speed, do they give very different solutions, and are the solutions close to the optimal one? Discuss what approach(es) give(s) best trade-off between running time and solution quality.

7.2 Extension to Oil Fields with Injection Wells

Injection wells are often drilled in order to keep up a high bottom-hole pressure of nearby oil production wells. Mobile rigs of the types discussed in Sect. 1.1 can also be applied to the injection wells, and a rig typically drills both production and injections wells. In this exercise, you are asked to extend the models suggested in Sects. 3–4, such that they also support decisions related to injector drilling.

Since the injectors, contrary to the oil producers, associate no production profile that generates profit, the benefit of drilling the injectors also has to be modeled differently. How this is done depends on what relation between presence of injection wells and production we assume. Consider each of the following assumptions, and extend either the location model or the flow model accordingly.

1. Assume that production from certain oil wells *requires* specific injectors to be drilled. To this end, let I denote the set of possible injectors, and let $WI \subseteq W \times I$ be the set of pairs (w, i) such that production from oil well w cannot start until injection well i is drilled. Note that any $w \in W$ and any $i \in I$ may occur in many such pairs.
2. Let the set of well pairs be defined as in the first assumption, but with a slightly different interpretation. For each $(w, i) \in WI$, define $\delta_{wi} \in [0, 1]$ such that, unless injector i is already drilled when production from producer w starts, the estimate of the production is downgraded from p_w^t to $\delta_{wi} p_w^t$ for all periods t. For simplicity, we assume that drilling the injection well has no effect on oil wells where production is already started.

Implement the model for each of the two assumptions independently. Test the model by extending the data set for either of the cases A–D, where you define some injection

wells and relations to selected oil wells. You can assume that for injection wells, typical values for drilling time and cost, as well as time to move between wells, resemble those of the production wells.

Acknowledgement Statoil ASA is gratefully acknowledged for letting the second author work on the problem under study for his master thesis. Methods developed and examples used in this chapter do not necessarily express Statoil's current disposal of rigs.

References

1. Carvalho, M. C. A., & Pinto, J. M. (2006). An MILP model and solution technique for the planning of infrastructure in offshore oilfields. *Journal of Petroleum Science and Engineering, 23*(1), 67–82.
2. Gupta, V., & Grossmann, I. E. (2012). An efficient multiperiod MINLP model for optimal planning of offshore oil and gas field infrastructure. *Industrial and Engineering Chemistry Research, 51*(19), 6823–6840.
3. Haugland, D., Hallefjord, Å., & Asheim, H. (1988). Models for petroleum field exploitation. *European Journal of Operational Research, 37*(1), 58–72.
4. Haugland, D., Jörnsten, K., & Shayan, E. (1991). Modeling petroleum fields with movable platforms. *Applied Mathematical Modelling, 15*(1), 33–39.
5. Iyer, R. R., Grossmann, I. E., Vasantharajan, S., et al. (1998). Optimal planning and scheduling of offshore oil field infrastructure investment and operations. *Industrial and Engineering Chemistry Research, 37*(4), 1380–1397.
6. Lin, X., & Floudas, C. A. (2003). A novel continuous-time modeling and optimization framework for well platform planning problems. *Optimization and Engineering, 4*(1–2), 65–95.
7. Ortiz-Gomez, A., Rico-Ramirez, V., & Hernandez-Castro, S. (2002). Mixed-integer multiperiod model for the planning of oilfield production. *Computers and Chemical Engineering, 26*(4–5), 703–714.
8. Tjøstheim, B. P. (2009). Optimization of the centralized rig intake process in Statoil. Master thesis in petroleum business engineering, Delft University of Technology, The Netherlands.
9. van den Heever, S. A., Grossmann, I. E., Vasantharajan, S., et al. (2000). Integrating complex economic objectives with the design and planning of offshore oilfield infrastructures. *Computers and Chemical Engineering, 24*(2–7), 1049–1055.

Chapter 14
Addressing the Peak Power Problem Through Thermal Energy Storage

Wesley Cole, JongSuk Kim, Kriti Kapoor and Thomas Edgar

1 Introduction

In the United States, the electrical power grid is divided into three primary regions: the Western Interconnection, the Eastern Interconnection, and the Texas Interconnection (see Fig. 14.1). Each of these regions struggles with peak power issues, but this case study will focus on the Texas Interconnection, which is operated by the Electricity Reliability Council of Texas (ERCOT).

Electricity demand fluctuates throughout the day. During the summer months in ERCOT, peak demand (sometimes just called "the peak") occurs during the late afternoon or early evening (see Fig. 14.2). One of the primary drivers of the peak is air conditioning, both from residential and commercial buildings.

The electrical grid has to be built to meet the peak demand. For ERCOT in 2012, that means that they needed more than 65 GW (gigawatts) of capacity. However, for most of the year, much of this capacity sits idle. For example, during a mild day in March, the peak may only reach 35 GW, meaning that more than 30 GW of generating capacity is sitting unused that day. Even on peak days (like in Fig. 14.2) there are several hours with significant untapped capacity. This leads to low capital utilization and increases the overall cost of electricity. Also, because peak power

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_14) contains supplementary material, which is available to authorized users.

W. Cole (✉) · J. Kim · K. Kapoor · T. Edgar
McKetta Department of Chemical Engineering,
University of Texas, Austin 78712, TX, USA
e-mail: wcole@che.utexas.edu

J. Kim
e-mail: jongkim@che.utexas.edu

K. Kapoor
e-mail: kriti@che.utexas.edu

T. Edgar
e-mail: edgar@che.utexas.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_14, © Springer Science+Business Media New York 2015

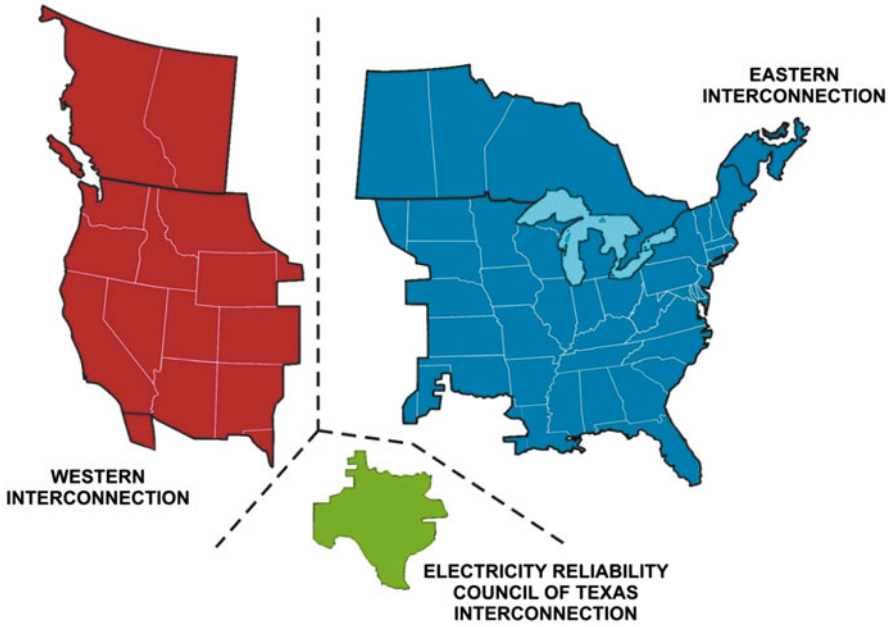


Fig. 14.1 Map of the North American electricity interconnections. (Image is from http://www.eia.gov/todayinenergy/detail.cfm?id=5070)

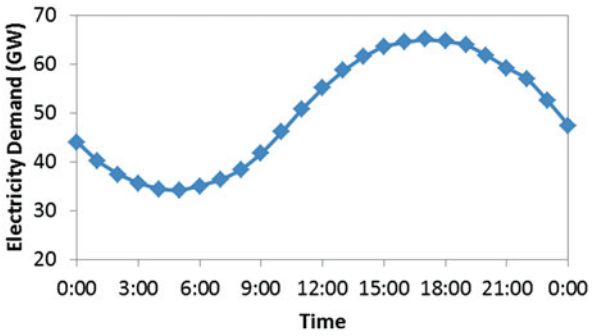


Fig. 14.2 Electricity power demand in the ERCOT system on June 25, 2012. The peak demand occurs during late afternoon, and its magnitude is nearly twice the demand at 5:00 am

plants are only used during peak times, they tend to be single-cycle gas turbines which have lower capital costs, but also lower efficiency and higher emissions than their combined-cycle counterparts. Combined-cycle power plants utilize the waste heat from the gas turbine to produce additional electricity.

Peak power creates disparity in the electricity prices throughout the day. In ERCOT, electricity is bought and sold in the wholesale market. For example, the utility that supplies electricity to homes to Austin may either produce that power themselves or purchase that power from another generator in the ERCOT market. At the same time, the local utility has the option of producing extra power and selling it in the ERCOT market. There are a variety of methods for buying and selling power in ERCOT, including real-time markets (electricity is purchased at the time of use), day-ahead

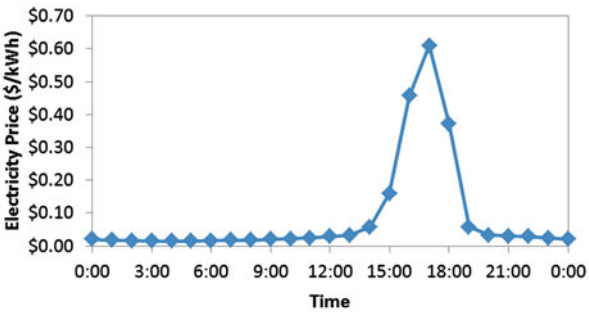


Fig. 14.3 Day-ahead wholesale market prices in the Austin load zone of ERCOT on June 25, 2012. Prices increase dramatically during the peak period

markets (electricity is purchased the day before it is used), and negotiated contracts (the parties involved set the individual terms of the contract). Prices are specified at 5-min intervals in the real-time market and at 1-h intervals in the day-ahead market. In both markets, however, the price of the electricity corresponds to the scarcity of electricity during that time.

For example, during the morning when demand is lowest, the most efficient, cheapest power plants are operating. As demand increases throughout the day, less-efficient and more expensive plants are brought online, increasing the market electricity prices. This is shown in Fig. 14.3, which corresponds to the same day as Fig. 14.2. Because of these fluctuations in prices during the day, the ability to have flexibility in when electricity is consumed can be very valuable. In this case study we consider adding that flexibility through the use of thermal energy storage.

2 Thermal Energy Storage

Thermal energy storage (TES) is the storage of thermal energy in some medium. The energy can be stored as latent heat, sensible heat, or chemical heat, though latent and sensible heat are the most common. Latent heat is the heat released or absorbed by a body during a constant temperature process, such as a phase change (e.g., boiling water). Sensible heat refers to heat transfer that results in a temperature change, but no phase change occurs (e.g., heating up a piece of metal). Chemical heat is heat stored in chemical bonds, such as H_2, which can be reacted to release heat.

In warmer climates such as in the ERCOT region, TES is often used for storing "cooling" in the form of chilled water or ice. For example, the University of Texas at Austin recently installed a four million gallon (15,140 m^3) chilled water TES tank. This insulated tank is used to store cold water when the cooling and power demands are low (e.g., at night time), and then the water can be discharged in the afternoon when the cooling and power demands are high. Without the TES, the campus buildings are cooled directly by the electric chillers, resulting in chilled water being used at the same time it is made. When using the water in the TES to provide cooling, the electric chillers can be turned down or off, thus reducing the electricity required by the campus while the TES is discharging. Though not

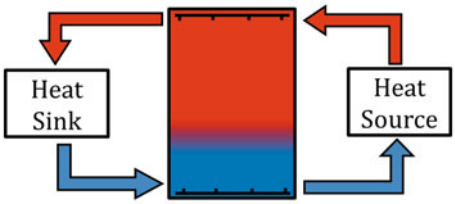


Fig. 14.4 Schematic of a stratified chilled water TES tank. The differences in density between the warmer and colder water in the tank ensure that the two do not mix. The heat source on the right could be, for example, a building that uses cold water to cool air for air conditioning, while the heat sink could be an electric chiller, which removes heat from the water. (Image from [8])

as versatile as battery storage (because TES can only meet thermal loads), TES is much cheaper with costs ranging from $ 6-43/kWh [1–4]. Battery costs range from $ 74–1484/kWh [5]. For more information on TES see [6, 7].

Thermal storage provides an opportunity to shift one of the largest electricity loads (air conditioning) from the expensive afternoon peak to the cheaper nighttime hours. It also provides an opportunity for the air conditioning equipment to operate more efficiently. The efficiency of large chillers, which make the chilled water for cooling the air in buildings, is a function of the amount of cooling energy (cooling load) provided by the chillers and the temperature of the outdoor air. TES gives the chillers more flexibility to operate in the regions where they are most efficient (cooler outdoor temperatures with moderate to high cooling loads).

The thermal storage considered in this problem is chilled water thermal storage using a single stratified water tank. This type of TES tank takes advantage of the fact that colder water is denser that warmer water. When cold water is needed it is removed from the bottom of the tank and sent to a heat source, such as a building, to provide cooling (see Fig. 14.4). The warmer water is returned to the top of the tank using carefully designed diffusers that ensure the flow is not turbulent. The difference in density keeps the warm water on top from mixing with the cold water on the bottom. This also causes the tank to only have two temperatures—the warm water temperature and the cold water temperature. To recharge the tank, warm water is taken from the top, cooled in the heat sink (i.e., the chillers), and returned to the bottom of the tank. The advantage of the stratified chilled water tank is that capital costs are reduced because only a single tank is required to store both the hot and the cold fluids.

3 Problem Description

Two variations of a single system are analyzed in this study. The first is a traditional chiller/cooling tower configuration (see Fig. 14.5a) that does not have TES. This provides a base case scenario for comparing the benefits of TES. The second is identical to the first except that it has a chilled-water TES unit running in parallel with

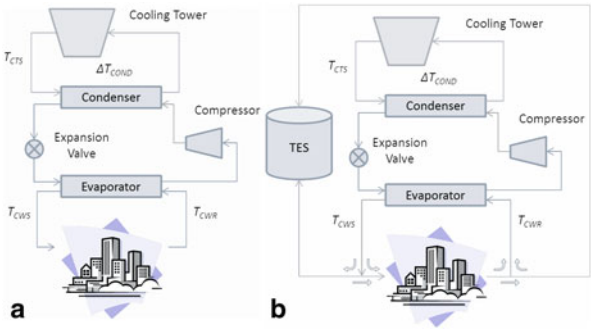


Fig. 14.5 Schematic of the system with (**b**) and without (**a**) thermal energy storage [9]

the chillers that supplies cold water to the buildings during peak hours (see Fig. 14.5b). The TES is recharged by the chillers during off-peak hours. Although Fig. 14.5 only shows one chiller, each chiller system has two identical chillers connected in parallel.

A section of the University of Texas at Austin campus was used to produce realistic cooling demands for the chiller-TES configurations. This section consists of five buildings: two mixed-use (classroom and office) buildings, an office administration building, a performing arts building, and a lab building with the total floor area of 65,500 m^2. Dynamic building cooling load profiles were calculated using DOE 2.2 building modeling software [10] and are provided with the attached data.

In both cases the chillers, cooling tower, and campus chilled water loop are identical. The cooling system is sized to meet a maximum cooling demand of 6050 kW by using two identical 3025 kW centrifugal chillers. Two chillers are selected because during times of low or medium load, a single, large chiller would be operating far below its design capacity. The minimum thermal operating capacity of a chiller is 30 kW. The temperatures of the chilled water supply (T_{CWS}, see Fig. 14.5) and chilled water return (T_{CWR}, see Fig. 14.5) are fixed at 40°F and 53°F, respectively.

4 Models

The chillers' power consumption ($P_{chiller}$) is a function of the chilled water supply temperature (T_{CWS}), the cooling tower supply temperature (T_{CTS}), and the ratio of the actual cooling load on the chiller to the nominal cooling load. The chilled water supply temperature is the temperature of the water leaving the chiller's evaporator and the cooling tower supply temperature is the temperature of the water leaving the cooling tower and entering the condenser (see Fig. 14.5). The actual load on the chiller is the cooling load supplied by the chiller at a given time and the nominal load is the rated capacity of the chiller (in this case, 3025 kW). The following set of equations predict the chillers' power consumption [11]:

$$\begin{aligned} CAPFT = a_1 + b_1 \cdot T_{CWS} + c_1 \cdot T_{CWS}^2 + d_1 \cdot T_{CTS} \\ + e_1 \cdot T_{CTS}^2 + f_1 \cdot T_{CWS} \cdot T_{CTS} \end{aligned} \tag{14.1}$$

Table 14.1 Constants used in (1), (2), and (4) for centrifugal chillers

	a	*b*	*c*	*d*	*e*	*f*
CAPFT	−0.299	0.0300	-8.00×10^{-4}	0.0174	-3.3×10^{-4}	6.31×10^{-4}
EIRFT	0.518	-4.00×10^{-3}	2.03×10^{-5}	6.99×10^{-3}	8.29×10^{-5}	-1.60×10^{-4}
EIRFPLR	2.54	−7.75	15.5	−15.5	7.69	−1.50

Table 14.2 Descriptions of *CAPFT*, *EIRFT*, *PLR*, and *EIRFPLR*

Acronym	Description
CAPFT	Available chiller capacity as a function of T_{CWS} and T_{CTS}. For example, a value of 0.9 for *CAPFT* indicates that 90 % of the chiller's capacity is available at that T_{CWS} and T_{CTS}.
EIRFT	Full load efficiency of the chiller as a function of T_{CWS} and T_{CTS}. Full loads indicates that the chiller is operating at maximum load (i.e., $L_{chiller} = Q_{nominal} \cdot CAPFT$).
PLR	The ratio of the chiller's current operating load to the maximum possible operating load.
EIRFPLR	Chiller efficiency as a function of the part load ratio (*PLR*).

$$EIRFT = a_2 + b_2 \cdot T_{CWS} + c_2 \cdot T_{CWS}^2 + d_2 \cdot T_{CTS} + e_2 \cdot T_{CTS}^2 + f_2 \cdot T_{CWS} \cdot T_{CTS} \tag{14.2}$$

$$PLR = L_{chiller}/(Q_{nominal} \cdot CAPFT) \tag{14.3}$$

$$EIRFPLR = a_3 PLR + b_3 PLR^2 + c_3 PLR^3 + d_3 PLR^4 + e_3 PLR^5 + f_3 PLR^6 \tag{14.4}$$

$$P_{chiller} = P_{nominal} \cdot CAPFT \cdot EIRFT \cdot EIRFPLR \tag{14.5}$$

where *CAPFT* is capacity as a function of temperature (defined by (14.1)), *EIRFT* is energy input ratio as a function of temperature (defined by (14.2)), *PLR* is part load ratio (defined by (14.3)), *EIRFPLR* is energy input ratio as a function of part load ratio (defined by (14.4)), $L_{chiller}$ is the cooling load (in kW) being provided by the chiller, $Q_{nominal}$ is the nominal rating of the chiller (in this case 3025 kW), $P_{nominal}$ is the nominal power usage of the chiller which in this case is 540 kW, and a_i through f_i are constants for centrifugal chillers as given in Table 14.1. Temperatures for all equations must be in degrees Fahrenheit. The power consumption of the chillers, given by (14.5), is a function of *CAPFT*, *EIRFT*, *PLR*, and *EIRFPLR* for the chillers. *CAPFT*, *EIRFT*, *PLR*, and *EIRFPLR* are all unitless and are further explained in Table 14.2.

The cooling tower supply temperature (T_{CTS}) is a function of the wet bulb temperature (T_w) and the temperature difference between the inlet and outlet of the chiller's condenser (ΔT_{COND}):

$$T_{CTS} = 9.14 + 0.771 T_w + 0.00149 {T_w}^2 + (2.431 - 0.0196 T_w - 3.51 \cdot 10^{-5} {T_w}^2)\Delta T_{COND} + (-0.0209 - 3.91 \cdot 10^{-5} T_w + 2.71 \cdot 10^{-6} {T_w}^2)(\Delta T_{COND})^2 \tag{14.6}$$

The value of ΔT_{COND} is controlled (i.e., held constant) at 10°F. A variable speed pump is included on the cooling tower loop to keep T_{CTS} above 53°F, which means that if (14.6) predicts T_{CTS} is lower than 53°F, then set T_{CTS} to 53°F.

The cooling load required by the buildings is met by some combination of the chiller and the TES:

$$L_{building,i} = L_{chiller,i} + TES_i \tag{14.7}$$

where $L_{building,i}$ is the cooling load required by the buildings at time i, $L_{chiller,i}$ is the cooling load provided by the chillers at time i, and TES_i is the amount of cooling provided by the TES (when it is negative it means that the TES is recharging, i.e., requiring rather than providing cooling). The power consumed by each chiller is given by Eq. (14.5). The TES is limited both by how quickly it can discharge and by how much energy it can hold:

$$C_{TES}, i = C_{TES,i-1} - TES_i \tag{14.8}$$

$$-ub_1 \leq TES_i \leq ub_1 \tag{14.9}$$

$$0 \leq C_{TES,i} \leq ub_2 \tag{14.10}$$

where $C_{TES,i}$ is the charge (amount of cooling energy) in the TES at time i, ub_1 and ub_2 are the upper bounds of the TES discharge rate (TES) and the TES storage capacity (C_{TES}), respectively.

5 Energy and Power

Power is the rate at which energy is used or consumed. The SI unit of energy is the Joule (J), and for power is the Watt (W), which is a Joule per second (J/s). A common unit of energy, especially when dealing with electricity, is the kilowatt-hour (kWh). One kWh is the amount of energy used by a system at a rate of 1 kW for 1 h. Because the time increment in this problem is 1 h, if the chiller operates at 200 kW for one time step, then it consumes 200 kWh.

Another important note to make is that a chiller essentially converts electrical energy to thermal energy. These two forms of energy are differentiated because electricity is a more valuable form of energy than thermal energy. So while the chillers operate at their thermal capacity of 3025 kW for a 1-h time step, they may only consume 1000 kWh of electricity during that time step to provide the 3025 kWh of cooling. The actual amount of electricity consumed by the chillers is a function of several variables and can be calculated using (14.1)–(14.5).

The Coefficient of Performance (COP) values for air conditioning systems are typically greater than one. Commercial chillers such as the ones modeled in this chapter have typical COPs of 3–6, meaning that they produce 3–6 times more cooling energy than the electrical energy they consume (see, for example http://en.wikipedia.org/wiki/Chiller).

The TES tank is sized according to volume (in m^3), but because the chilled water supply (T_{CWS}) and chilled water return (T_{CWR}) are fixed at 40 and 53°F, respectively, the volume of the TES tank has an energy equivalent (given in assumption 5 below). For this problem, it may be easier to consider the TES as an energy tank, having zero usable energy when the tank is full of 53°F water and having a full charge of energy (given by ub_2) when the tank is full of 40°F water. Thus the amount of energy in the TES tank (C_{TES}) can be measured in thermal kWh and range from 0 to ub_2.

6 Data and Assumptions

Use the following data and assumptions in this problem:

1. Hourly values for $L_{building}$, T_w, and electricity prices (p_{elec}) for a 1 year period are provided in the accompanying spreadsheet.
2. Use a 1 h time step in solving the problem.
3. The building cooling load ($L_{building}$) must be met exactly at each time step.
4. The cost of installing the TES is given by

$$Cost = 5265(V)^{-0.455} \tag{14.11}$$

   where *Cost* is the cost of the TES in \$/m^3 and V is the vol-ume of the TES given in m^3 (price adapted from [12]).
5. The thermal capacity of the water in the TES tank is 8.5 kWh/m^3.
6. The maximum hourly discharge rate (in kWh/h) of the TES (ub_1) is 25 % of the maximum storage capacity (in kWh) of the TES (ub_2), i.e., $ub_1 = 0.25ub_2$.
7. The annual interest rate (r) is 6 %.
8. The TES loses 0.1 % of its usable energy every hour.
9. The TES tank can be installed immediately and will last for 20 years.
10. There are no additional maintenance costs due to the TES.
11. The campus cooling load, weather, and electricity prices will be identical to the 2011 values for the life of the TES. For example, the cooling load, weather, and electricity prices for 2016 will be identical to those in 2011.
12. There is no cost for turning on or off a chiller.
13. The TES tank is initially full of water at 53°F. This means that the tank starts with no usable energy ($C_{TES} = 0$ kWh at the initial time).

To reduce the complexity of the problem, you can include a simple heuristic to divide the load between the two chillers. For example, if the desired cooling load is less than the maximum cooling load of one chiller, then only one chiller operates, but if the desired load is greater than the maximum of the cooling load of one chiller, then the chillers share the cooling load equally. This will eliminate the need to include binary variables for the chillers' operation.

7 Desired Outputs

Your task is to determine the optimal size of the TES tank using the net present value (NPV) and the payback period (PBP)—a description of each is given below. You will get a different optimal size based on each metric. Compare the optimal size of the TES tank in each case. Note that the optimal size of the TES is a function of the TES operation, so the annual TES operation will also have to be determined. The baseline for comparison is the system without the TES (Fig. 14.5a).

An optimal solution should include the tank volume in m^3 (V), the cooling load supplied by the TES (TES) in kWh for every hour of the year, and the value of the objective function (in dollars or years, depending on which metric is used).

7.1 Economic Metrics

Net Present Value (NPV)—NPV is the present value (in dollars) of a series of cash flows by a project. In this case the cash flows are the investment cost (a negative cash flow) and the savings in operating costs by using the TES system (positive cash flows). Because the operating cost savings are accrued over the life of the system, the cash flows must be discounted according to the interest rate. A project is acceptable only if the NPV is at least zero, and projects with a higher NPV are considered to be more profitable. In mathematical terms NPV is given as

$$NPV = E(V)\frac{1-(1+r)^{-n}}{r} - V \cdot Cost(V) \tag{14.12}$$

where $E(V)$ is the savings in operating costs (in dollars) from using the TES over 1 year for a given volume (V), r is the annual interest rate, n is the lifetime of the TES system (given in years), and $Cost(V)$ is the investment cost (in dollars) of the TES given by Eq. (14.11).

Payback Period (PBP)—PBP is the time (in years) in which the initial expenditure of an investment is expected to be recovered from the revenues generated by the investment. For this problem, the payback period is the time in which the initial costs of installing a thermal storage tank are expected to be recovered by the energy savings from using the storage:

$$PBP = \frac{V \cdot Cost(V)}{E(V)} \tag{14.13}$$

8 Problem Formulation

There are several ways to simplify this problem. The first is fairly obvious and straightforward. Because it is assumed that the weather, electricity prices, and building cooling loads will be the same for every year over the lifetime of the TES system,

only 1 year of TES operation needs to be solved. The savings will be identical over all the other years considered.

An additional way to simplify the problem is to consider that the factors affecting the TES are diurnal in natural. Weather, electricity prices, and building loads all have their peak during their day and their trough during the night. The TES adds value to the system by storing cooling energy during the night and discharging that cooling energy during the day. Because the TES includes a loss term, it will always be optimal to store energy the night before it will be used. Thus, the optimal TES operation problem can be solved as a series of 365, 24-h problems:

$$E\left(V, TES_i^d\right) = \sum_{d=1}^{365}\left[\min_{TES_i^d} \sum_{i=1}^{24} C_{elec,i}^d \left(P_{chiller,i}^{base,d} - P_{chiller,i}^{V,d}\right)\right] \tag{14.14}$$

where TES_i^d is the thermal storage discharge/recharge rate (in kWh/h) for day d and hour i. $C_{elec,i}^d$ is the electricity price (in dollars) for day d and hour i, $P_{chiller,i}^{base,d}$ is the electrical power consumed (in kW) by the chillers in the base case (where $V = 0$) for day d and hour i, $P_{chiller,i}^{V,d}$ is the electrical power consumed (in kW) by the chillers for a given volume (V) at day d and hour i. $P_{chiller}$ is defined in (14.5) for a single hour (i).

Eqs. (14.11)–(14.14) are subject to (14.7)–(14.10) as well as the following constraint:

$$L_{chiller,min} \leq L_{chiller,i}^d \leq L_{chiller,max} \tag{14.15}$$

where $L_{chiller,min}$ is the minimum chiller load for a single chiller (30 kW) and $L_{chiller,max}$ is the maximum chiller load for the combined chillers (6050 kW). This means that $L_{chiller,max}$ is twice the nominal rating of a single chiller in order to take into account that two identical chillers are used if the chiller cooling load exceeds the nominal rating. In other words, when $L_{chiller}$ is greater than $Q_{nominal}$, $L_{chiller}$ will be shared equally by two chillers, so (14.3) and (14.5) need to be modified as the followings:

$$PLR = L_{chiller}/(2 \cdot Q_{nominal} \cdot CAPFT) \tag{14.16}$$

$$P_{chiller} = 2 \cdot P_{nominal} \cdot CAPFT \cdot EIRFT \cdot EIRFPLR \tag{14.17}$$

This conditional relationship between the chiller load and power consumption can be handled by IF-ELSE statement in many programming languages such as MATLAB; however, the problem can also be formulated as a MINLP problem in order to avoid the IF-ELSE logic. Using the IF-ELSE logic makes the differential discontinuous, but because the problem is solved using a computer which discretizes the space, the differential becomes continuous. The differential calculation does, however, lose accuracy as it approaches the cutoff value in the conditional statement.

The TES storage capacity is specified by the upper bound on the energy in the tank:

$$ub_2 = \rho_{w,energy} \cdot V \tag{14.18}$$

where $\rho_{w,\,energy}$ is the thermal capacity of water in the TES tank (given as 8.5 kWh/m^3). The upper bound of the TES discharge rate (ub_1) is 25 % of ub_2:

$$ub_1 = 0.25 \cdot ub_2 \tag{14.19}$$

Eq. (14.20) takes into account that the TES tank loses 0.1 % of its energy every hour:

$$C^d_{TES,i} = (1 - 0.001) \cdot \left(C^d_{TES,i-1} - TES^d_i\right) \tag{14.20}$$

This equation also allows for continuity between days—when $i = 1$, then

$$C^d_{TES,1} = (1 - 0.001) \cdot \left(C^{d-1}_{TES,24} - TES^d_1\right) \tag{14.21}$$

In summary, the size of the TES tank is optimized depending on two different economic metrics:

$$NPV = \max_{V, TES^d_i} \left[\sum_{d=1}^{365} \left[\min_{TES^d_i} \sum_{i=1}^{24} C^d_{elec,i} \left(P^{base,d}_{chiller,i} - P^{V,d}_{chiller,i} \right) \right] \cdots \right. \\ \left. \cdot \frac{1 - (1+r)^{-n}}{r} - \left(5265 \cdot V^{-0.455}\right) V \right] \tag{14.22}$$

or

$$PBP = \min_{V,\, TES^d_i} \left[\frac{\left(5265 \cdot V^{-0.455}\right) V}{\sum_{d=1}^{365} \left[\min_{TES^d_i} \sum_{i=1}^{24} C^d_{elec,i} \left(P^{base,d}_{chiller,i} - P^{V,d}_{chiller,i} \right) \right]} \right] \tag{14.23}$$

subject to

$$L^d_{building,i} = L^d_{chiller,i} + TES^d_i \tag{14.24}$$

$$C^d_{TES,i} = (1 - 0.001) \cdot \left(C^d_{TES,i-1} - TES^d_i\right) \tag{14.25}$$

$$ub_2 = \rho_{w,energy} \cdot V \tag{14.26}$$

$$ub_1 = 0.25 \cdot ub_2 \tag{14.27}$$

$$\begin{aligned} T^d_{CTS,i} &= 9.14 + 0.771 T^d_{w,i} + 0.00149 (T^d_{w,i})^2 \\ &+ (2.431 - 0.0196 T^d_{w,i} - 3.51 \cdot 10^{-5} (T^d_{w,i})^2) \Delta T_{COND} \\ &+ (-0.0209 - 3.91 \cdot 10^{-5} T^d_{w,i} + 2.71 \cdot 10^{-6} (T^d_{w,i})^2) (\Delta T_{COND})^2 \end{aligned} \tag{14.28}$$

$$\begin{aligned} CAPFT^d_i &= a_1 + b_1 \cdot T^d_{CWS,i} + c_1 \cdot (T^d_{CWS,i})^2 \\ &+ d_1 \cdot T^d_{CTS,i} + e_1 \cdot (T^d_{CTS,i})^2 + f_1 \cdot T^d_{CWS,i} \cdot T^d_{CTS,i} \end{aligned} \tag{14.29}$$

$$EIRFT_i^d = a_2 + b_2 \cdot T_{CWS,i}^d + c_2 \cdot (T_{CWS,i}^d)^2 + d_2 \cdot T_{CTS,i}^d + e_2 \cdot (T_{CTS,i}^d)^2 + f_2 \cdot T_{CWS,i}^d \cdot T_{CTS,i}^d \tag{14.30}$$

$$PLR_i^d = \begin{cases} L_{chiller,i}^d / (2 \cdot Q_{nominal} \cdot CAPFT_i) & \text{if } L_{chiller,i}^d > Q_{nominal} \\ L_{chiller,i}^d / (Q_{nominal} \cdot CAPFT_i) & \text{otherwise} \end{cases} \tag{14.31}$$

$$EIRFPLR_i^d = a_3 PLR_i^d + b_3 \left(PLR_i^d\right)^2 + c_3 \left(PLR_i^d\right)^3 + d_3 \left(PLR_i^d\right)^4 + e_3 \left(PLR_i^d\right)^5 + f_3 \left(PLR_i^d\right)^6 \tag{14.32}$$

$$P_{chiller,i}^d = \begin{cases} 2 \cdot P_{nominal} \cdot CAPFT_i^d \\ \cdot EIRFT_i^d \cdot EIRFPLR_i^d & \text{if } L_{chiller,i}^d > Q_{nominal} \\ P_{nominal} \cdot CAPFT_i^d & \text{otherwise} \\ \cdot EIRFT_i^d \cdot EIRFPLR_i^d \end{cases} \tag{14.33}$$

$$-ub_1 \leq TES_i^d \leq ub_1 \tag{14.34}$$

$$0 \leq C_{TES,i}^d \leq ub_2 \tag{14.35}$$

$$L_{chiller,min} \leq L_{chiller,i}^d \leq L_{chiller,max} \tag{14.36}$$

9 Problem Solution, Analysis, and Discussion

The problem was solved using the following general set of steps:

1. Select TES tank size
2. Determine the optimal operation of the TES tank over its lifetime
3. Calculate the economic metric (NPV/PBP)
4. Check stopping criteria—if met, then end
5. Go to 1

This set of steps allows one to deal with the nested optimization problems in (14.22) and (14.23). By first specifying the tank volume, the inner minimization problem can be solved to find *TES* for every hour of the year. Once *V* and *TES* are known, the objective function value can be determined and the process continued. There are a variety of methods that can be used to implement each specific step. Some of them are discussed here, but this is far from a comprehensive overview.

In every case the initial TES tank size was chosen as 1000 m^3. This represents a moderately small tank size that is capable of meeting about 40 % of the peak load of the system and assumes that a TES system will be a favorable investment. If there is good reason to think that the optimal TES size would be zero (i.e., do not install a TES tank), then a smaller initial TES size might be a better choice.

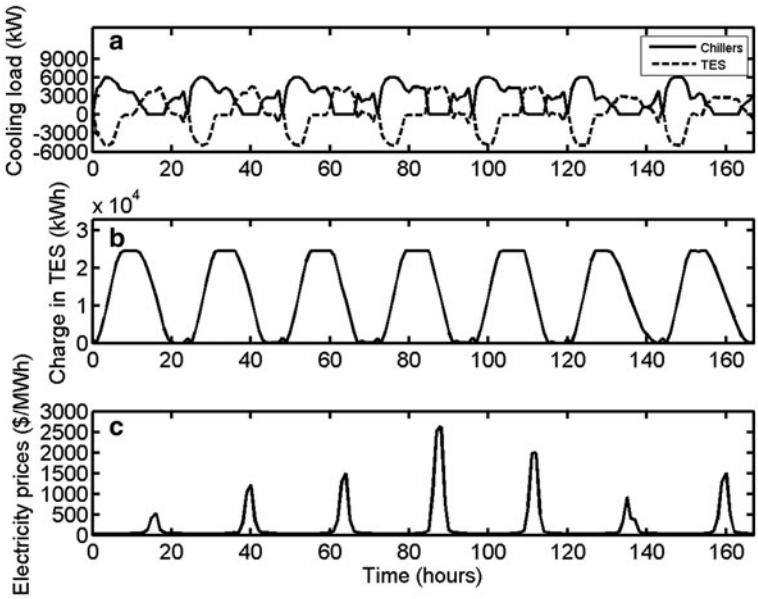


Fig. 14.6 Optimized schedule for one week in August using a tank size of 2896 m^3. The three plots are (a) Chiller load ($L_{chiller}$—*solid line*) and TES load (*TES—dashed line*), (b) charge in the TES (C_{TES}), and (c) electricity prices. The negative values for the TES cooling load in (a) show when and by how much the TES is being recharged (i.e., the chillers are putting energy into the tank)

Once a tank size is selected, then the optimal operation of the TES tank can be determined. Fig. 14.6 shows the optimal operation of the TES system for a week in August using a tank size of 2896 m^3. The TES system allows the chillers to turn off during the daytime when electricity prices are highest and then recharge the tank at night when prices are low. The TES tank is sized so that it can meet the entire cooling demand for a period of several hours (shown by the regions where the chiller load goes to zero). This results in tremendous operating savings because prices during peak afternoon times can be orders of magnitude higher than during off-peak, nighttime hours (see Fig. 14.6c). The chillers often operate at their maximum capacity during the night hours to recharge the TES tank with chilled water for the next day. The diurnal pattern that allows this problem to be solved as 365 separate problems is very apparent in Fig. 14.6. Changing the size of the TES results in the same general pattern, but more or less of the cooling load is shifted from daytime to nighttime.

Once the optimal operation is solved, the annual operating savings in dollars is compared to the base case to obtain E for the given volume. Eqs. (14.12) and (14.13) can then be used to determine the NPV or PBP.

The stopping criteria used in step 4 can be based on the objective function value, on the TES size, or both. In this case, the stopping criterion was set as the resolution of the TES size. Once the optimal TES size was determined to within 10 m^3, the optimal solution was said to have been found. This resolution was chosen for practical

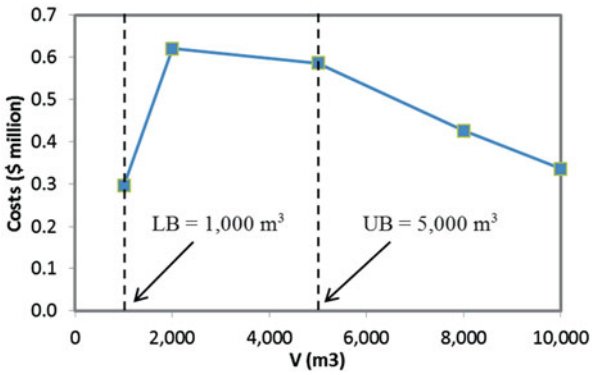


Fig. 14.7 NPV as a function of the volume of the TES. The optimal solution lies between the lower bound (LB) and the upper bound (UB) shown in the figure

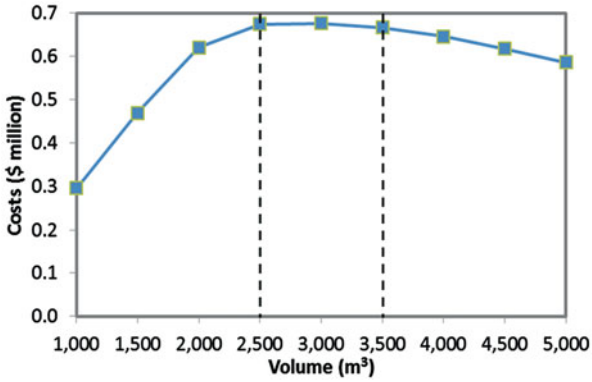


Fig. 14.8 NPV as a function of volume of the TES. The optimal solution lies between 2500 m^3 and 3500 m^3

reasons. Given a tank of 3000 m^3, 10 m^3 is less than 1 % of the total volume and at that level of resolution the actual volume will be subject to construction and engineering constraints.

Two methods were employed in selecting the next TES size to try. The first was a manual grid search method. This method represents a typical approach that might be applied by a design engineer. To find the optimal TES tank size using this method, a variety of tank sizes was chosen ranging from 1000 to 10,000 m^3. The optimal operation for each of these tank sizes was determined so that the NPV or PBP could be calculated. The results of this first pass are shown in Fig. 14.7 for the NPV and indicate that the optimal tank volume lies somewhere between 1000 and 5000 m^3. A new set of tank volumes was chosen that were within this range and the optimal operation was determined for each volume (see Fig. 14.8). Because this process was automated the TES tank volumes were simply set and allowed to run. This process was continued until the granularity in tank sizes was 10 m^3.

The second method used to determine the next TES volume was a simple line search method. In the line search method employed here, the TES size was increased by 1000 m^3 times the iteration number until a minimum or maximum was bracketed (similar to Fig. 14.7). A parabola was then fit to the three points that formed the

Table 14.3 Summary of the optimal solutions

Objective Function	Volume of the TES, V	Maximum TES Discharge Rate, ub_1	TES Capacity, ub_2 (or C_{TES})	Annual Savings in Operating Costs, E	Objective Function Value
Maximizing the NPV	2896 m^3	6154 kW	24,616 kWh	$ 94,700	$ 680,000
Minimizing the PBP	2152 m^3	4573 kW	18,292 kWh	$ 87,900	3.93 years

Table 14.4 Solution times for the inner minimization problem using the different fmincon algorithms

Tank Volume (m^3)	Interior-point	Active-set	Sequential Quadratic Programming (SQP)
2896	25.5 min	22.2 min	13.9 min

bracketed region, and the TES volume that maximized or minimized this parabola was chosen as the next TES volume to try. This method continued until the change in TES size from one iteration to the next was within 10 m^3. This method was entirely automated, so once the initial size and convergence tolerance were set, the algorithm would run until the final solution was reached.

Both methods effectively arrive at an optimal solution. The line search reduces the amount of user involvement and the time to arrive at an optimal solution, but the manual grid search method is easy to implement and may be more straightforward when explaining the optimal size to project management personnel.

The optimal TES size for the NPV and the PBP are different and are summarized in Table 14.3. The optimal TES volume given by the NPV objective function is 35 % larger than the one given by the PBP objective function. Thus more value can be obtained from a larger tank, but the capital can be recovered more quickly using a smaller tank. The choice of optimal size depends on the available capital and on the company goals. The smaller tank will expend less capital and recover the capital more quickly, allowing that capital to be used on other projects. The larger tank will be more profitable over the lifetime of the system, but incurs a greater initial capital expense, limiting the resources available for other projects. Also, the larger tank is able to shift a greater amount of energy from peak hours to off-peak hours, so if there is concern that prices will increase over the years due to a scarcity in the electricity markets, then the larger tank would be more advantageous.

This problem was formulated and solved in MATLAB using fmincon, which has three algorithms available for NLPs: interior-point, active-set, and sequential quadratic programming (SQP).The solution times for the different algorithms are shown in Table 14.4. The SQP algorithm significantly outperforms the other algorithms in terms of solution time due in part to the quadratic nature of the optimization problem.

10 Additional Exercises

There are a number of aspects not considered here that are nonetheless interesting and relevant. For example, the assumption that electricity prices, weather, and building loads will remain the same for the life of the system is obviously not correct. To analyze this assumption, try changing these three inputs (electricity prices, weather, and building loads) up and down by 10 % to see the effect they have on the optimal solution. To which of the inputs is the system most sensitive?

The two chillers in this system are assumed to be identical, but in practice that is not necessarily the case. Change the coefficients in Table 14.1 by a random ± 5 % for the second chiller and then formulate the problem as a mixed integer nonlinear program. See how the optimal solution changes when the two chillers are no longer identical. Also, compare the solution times from the MINLP formulation and the NLP formulation that uses the identical chillers.

11 Summary

Thermal energy storage is an effective technology for shifting power from peak demand hours to off-peak hours. This power shifting capability results in financial benefits because the TES system consumes more energy during cheap night hours and reduces use during expensive peak hours. The optimal operation can be determined using optimization methods, and the savings achieved can be quantified using metrics such as net present value (NPV) and payback period (PBP). Depending on the metric chosen, the optimal TES tank volume can vary substantially—in this case the NPV-based design resulted in a tank 35 % larger than the PBP-based design. Both metrics, however, are useful in making a final decision regarding TES system sizing.

References

1. Dorgan, C. E., & Elleson, J. S. (1993) *Design guide for cool thermal storage*. Atlanta: ASHRAE.
2. Hasnain, S. M. (1998) Review on sustainable thermal energy storage technologies, part II: Cool thermal storage. *Energy Conversion and Management, 39*(11), 1139–1153.
3. Hasnain, S. M., Alawaji, S. H., Al-Ibrahim, A. M., & Smiai, M. S. (2000) Prospects of cool thermal storage utilization in Saudi Arabia. *Energy Conversion and Management, 41*(17), 1829–1839.
4. Roth, K., Zogg, R., & Brodrick, J. (2006). Cool thermal energy storage. *ASHRAE Journal, 48,* 94–96.
5. Hou, Y., Vidu, R., & Stroeve, P. (2011). Solar energy storage methods. *Industrial & Engineering Chemistry Research, 50*(15), 8954–8964.
6. Cole, W. J., Powell, K. M., & Edgar, T. F. (2012) Optimization and advanced control of thermal energy storage systems. *Reviews in Chemical Engineering, 28*(2–3), 81–99.
7. Dinçer, İ., & Rosen, M. (2011). *Thermal energy storage: Systems and applications* (2nd ed.). Hoboken: Wiley.

8. Powell, K. M., & Edgar, T. F. (2013). An adaptive-grid model for dynamic simulation of thermocline thermal energy storage systems. *Energy Conversion and Management, 76,* 865–873.
9. Cole, W. J., Edgar, T. F., & Novoselac, A. (2012) *Use of model predictive control to enhance the flexibility of thermal energy storage cooling systems*. In Proceedings of the 2012 American Control Conference, Montreal, Canada, pp. 2788–2793.
10. DOE (2013) Building technologies office: EnergyPlus energy simulation software, 2013 (Online). http://apps1.eere.energy.gov/buildings/energyplus/. Accessed 13 Sept 2013.
11. Hydeman, M., Webb, N., Sreedharan, P., & Blanc, S. (2002) Development and testing of a reformulated regression-based electric chiller model. *ASHRAE Transactions, 108*(2), 1118–1127.
12. de Wit, J. (2007) *Heat storages for CHP optimisation*. In Proceedings of the Power Gen Europe 2007 Paper (ID-94).

Chapter 15
Optimal Flight Planning for a Jet Aircraft

Harris McClamroch and Taeyoung Lee

1 Overview

This chapter formulates a flight optimization problem to characterize the optimal flight of a jet aircraft subject to mission constraints and constraints that arise from the basic physics of flight of the aircraft. A hierarchical optimization approach is suggested: a flight plan is first developed, specified by way points and segments of helical paths between these waypoints, that meet the mission constraints; then an optimal steady flight problem is formulated and solved for each of the helical flight path segments. The required background in flight physics is summarized and optimal flight conditions for several categories of steady flight are described. Finally, a specific fuel optimal flight optimization problem for an example of a jet aircraft is introduced as a case study. The hierarchical optimization approach is followed and a fuel optimal flight plan is obtained and its flight performance is analyzed.

2 Background

The modern era of flight began with the Wright brothers and their demonstration of controlled flight in 1903. Major advances have been made in the design of flight vehicles since that time, building on new knowledge and technologies in aerodynamics, structures, propulsion and, most recently, advanced computer and information systems. Today, aircraft flight is a critical component of our technological infrastructure playing dominant roles in commercial transportation and military sectors.

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_15) contains supplementary material, which is available to authorized users.

H. McClamroch (✉)
University of Michigan, Ann Arbor, MI 48109, USA
e-mail: nhm@umich.edu

T. Lee
George Washington University, Washington, DC, USA
e-mail: tylee@gwu.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_15, © Springer Science+Business Media New York 2015

Current flight technology is mature in the sense that expected advances now are most likely to occur incrementally rather than through major new breakthroughs.

On the other hand, the continuing rapid increase in flight vehicle traffic, in the US, Europe and throughout the world, has led to increased air traffic congestion and associated expenses. This has been true for commercial air traffic, general aviation air traffic and military air traffic. Consequently, many of the major challenges today arise from the need to improve flight planning and operations for both individual aircraft and for coordination and scheduling of many aircraft in a fixed air space. There have been many advances in flight planning and operations during the last 25 years, but many challenges remain.

This chapter provides an introduction to the flight planning and operations problem, as an optimization problem, for a single aircraft. These results are important in themselves, but they also serve to inform studies on multiple aircraft flight.

Our approach here is to view flight planning for a single aircraft as an optimization problem. A mission performance objective, such as using the least possible amount of fuel, is specified. The mission constraints include specification that the aircraft take off and climb to altitude, cruising flight segments and turning flight segments, and finally descent and landing. In addition, aircraft flight is constrained by its flight physics and these constraints must also be taken into account in the optimal planning problem. This formulation of an optimal flight planning problem is introduced in Chap. 12 of [6]. The development here expands on the material given in [6].

This perspective is based on the assumption of specified jet aircraft characteristics; that is, the problem is one of operational planning rather than aircraft design. The obtained results inform a pilot or remote operator on the flight conditions that must be maintained to achieve mission optimality.

The following topics are addressed. We introduce the optimal flight planning problem formulated as a hierarchical optimization problem. The problem is decomposed into selection of a sequence of helical flight path segments and determination of optimal flight conditions for each flight path segment. To provide necessary background, we summarize the important flight physics and we summarize optimal flight conditions for several different categories of steady flight. We then analyze a hypothetical but typical jet aircraft optimal flight planning problem as a case study.

This material should be accessible to a broad audience with a basic background in engineering. For some readers, the flight physics may not be familiar; our summary of the results is at a conceptual level but more detailed treatments are available in the cited literature. Familiarity with the basics of optimization subject to constraints is helpful.

3 Optimal Flight Planning Problems

Flight optimization problems are typically stated in terms of determining a flight plan so that the aircraft flies from a specified take-off location to a specified landing location while achieving some optimal objective, such as using minimum fuel. Additional constraints on the flight mission, in the form of path equality and/or inequality

constraints in 3D, are often included. In addition, important constraints arise from the basic flight physics of the aircraft; these must also be satisfied to achieve a flight plan that the aircraft can physically fly. A part of this problem is the determination of pilot inputs, namely aircraft thrust, elevator deflection and bank angle, that must be applied to achieve the optimal flight plan.

In mathematical terms, a flight optimization problem can be formulated as an optimal control problem or as a nonlinear programming problem: optimize a specific performance measure subject to satisfaction of a number of mission and flight constraints. There is a large published literature on optimal control and nonlinear programming that treats both theoretical and numerical aspects [3]. Unfortunately, the most general formulation of the optimal flight planning problem as a nonlinear programming problem is not generally a convex programming problem, it is infinite dimensional, and it has no special mathematical structure that can easily be exploited. These features of optimal flight planning problems make them exceedingly challenging.

As can be seen from this informal description, there are two important features of flight optimization problems: (1) what is the optimal 3D path along which the aircraft should fly? (2) what are optimal pilot inputs that cause the aircraft to actually fly along this optimal path? These two aspects provide the motivation for viewing the flight optimization problem as a simplified two stage hierarchical optimization problem.

4 Hierarchical Decomposition of the Optimal Flight Planning Problem

The optimal flight planning problem is decomposed into two simpler problems at two hierarchical levels. In the first level, referred to as flight path planning, the flight plan is segmented into a concatenation of helical flight paths, using a sequence of way points, that meet constraints imposed by landing, take-off, and other mission constraints. Here we make use of helical flight path segments in 3D: straight line segments and circular arc segments in a horizontal plane.

The optimality features are introduced for each flight segment at the second level, where optimal pilot decisions, namely thrust level provided by the jet engines, elevator deflection, and bank angle, are selected so that the aircraft flies along the specified helical flight segment in an optimal way. This approach handles the mission constraints in the first level by selection of way points that lead to efficient flight paths. The constraints that arise from the flight physics are included at the second level where formal optimization methods are used for each of the several flight segments based on the assumption of steady flight on that flight segment.

The optimal flight planning problem case study provides a concrete illustration of the hierarchical optimization process. Details are given to demonstrate the use of both decision heuristics as well as formal optimization methods in constructing optimal or near-optimal solutions. The specific development illustrates how the results for each of the several flight segments are integrated to evaluate and assess the optimal flight plan and the mission performance.

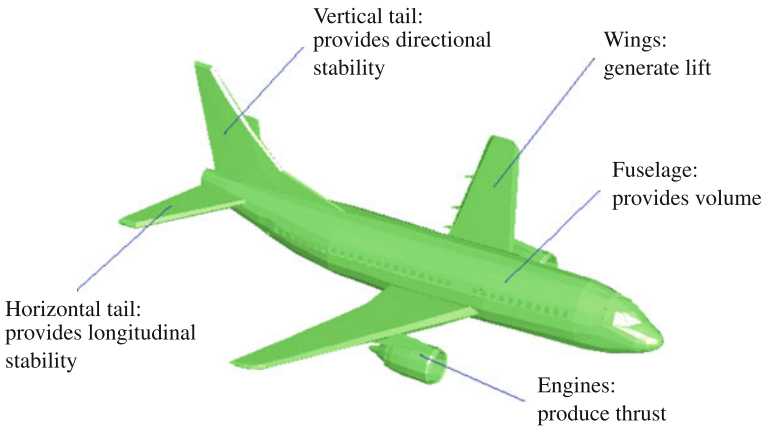


Fig. 15.1 Aircraft subsystems

This hierarchical decision approach is an important simplification, since it involves heuristic selection of a sequence of helical flight path segments in 3D. This sequence of helical flight path segments is not guaranteed to be optimal. However, this hierarchical approach has proved to be efficient and effective in providing good approximate solutions for many optimal flight planning problems.

5 Aircraft Fundamentals

We introduce the basic features of flight of fixed wing aircraft. We identify aircraft subsystems, giving special attention to aerodynamic control surfaces and jet aircraft propulsion systems. We consider only subsonic flight where the air speed, the velocity of the aircraft relative to the surrounding wind, is less than the local speed of sound.

5.1 Conventional Fixed-Wing Aircraft

Figure 15.1 illustrates a conventional fixed-wing aircraft. The key physical components, or subsystems, that define the aircraft are the fuselage, the wings, the horizontal tail, the vertical tail, and the propulsion system. The fuselage provides working volume for passengers, cargo, and aircraft subsystems that are internal to the aircraft. The fuselage is not important from a flight performance perspective. The two wings are crucial for flight, since their main purpose is the generation of lift. The aircraft illustrated in Fig. 15.1 is a fixed-wing aircraft, since the wings are rigidly attached to the fuselage. This is in contrast with helicopters or other rotary wing flight vehicles that generate lift using rotating blades.

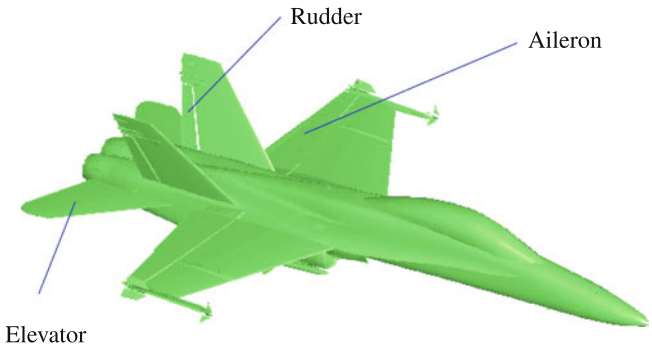


Fig. 15.2 Aerodynamic control surfaces

Other important flight subsystems, illustrated in Fig. 15.1, are the horizontal tail, the vertical tail, and the engines. The horizontal and vertical tails are rigidly attached to the fuselage as indicated. The horizontal tail provides longitudinal stability, while the vertical tail provides directional stability. The jet engines are crucial flight subsystems, since they generate the thrust force that acts on the aircraft.

The complete aircraft, consisting of the fuselage, the wings, the horizontal and vertical tails, and all other flight subsystems, is assumed to have a plane of mass symmetry that exactly bisects the aircraft. This assumption is a consequence of the design of conventional fixed-wing aircraft where, in particular, engines mounted on the fuselage or the wings are balanced to satisfy this mass symmetry assumption.

5.2 Aerodynamic Control Surfaces

Figure 15.2 illustrates three types of aerodynamic control surfaces. If a control surface is deflected, the flow of air past the surface is modified giving rise to an aerodynamic moment that the pilot uses to control the motion of the aircraft. The three types of aerodynamic control surfaces are the elevator, the ailerons, and the rudder. The elevator is one (or more than one) movable flap, located on the trailing edge of the horizontal tail. Deflection of the elevator changes the air flow over the horizontal tail in such a way that a pitch moment on the aircraft is generated. The ailerons consist of a pair of movable flaps, located on the trailing edge of each wing; ailerons usually operate in differential mode so that if one flap is deflected up the other flap is deflected down by the same amount, or vice-versa. Differential deflection of the ailerons changes the air flow over the wings in such a way that a roll moment on the aircraft is generated. The rudder is one (or more than one) movable flap, located on the trailing edge of the vertical tail. Deflection of the rudder changes the air flow over the vertical tail in such a way that a yaw moment on the aircraft is generated. The elevator deflection, rudder deflection, and the differential deflection of the ailerons are typically viewed as angles measured in degrees from some references.

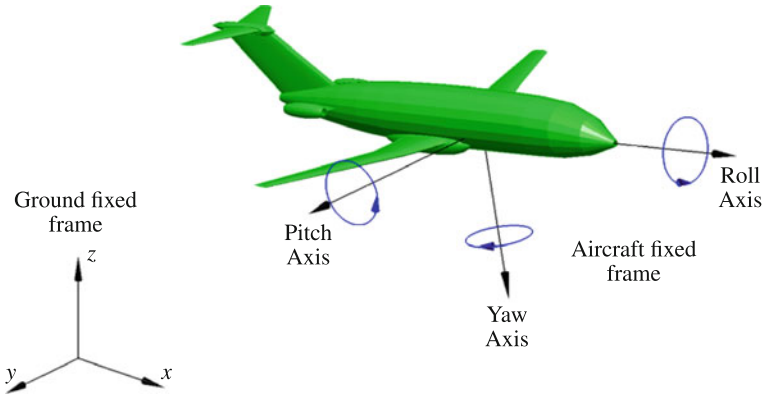


Fig. 15.3 Aircraft fixed frame and ground fixed frame

These movable flaps are referred to as aerodynamic control surfaces; they generate moments on the aircraft according to the principles of aerodynamics. These moments are used by the pilot to maneuver and control the flight of the aircraft. The rudder is primarily used to keep the direction of the velocity vector of the aircraft in the plane of mass symmetry of the aircraft. The ailerons are used to bank the aircraft as required in turning flight. The pitching moment is important in controlling the magnitude of the velocity vector and the flight path angle of the aircraft, which is the angle that the velocity vector makes with the local horizontal plane.

Figure 15.3 indicates a coordinate frame consisting of three mutually perpendicular axes *fixed* to the body of an aircraft in flight. The axis through the center line of the aircraft is the roll axis which defines the direction of the roll moment due to a deflection of the ailerons. The axis perpendicular to the roll axis through the wings is referred to as the pitch axis and it defines the direction of the pitch moment due to deflection of the elevator. Finally, the yaw axis is perpendicular to both the roll axis and the pitch axis as shown and it defines the direction of the yaw moment due to deflection of the rudder. The origin of the coordinate frame is usually taken as the center of mass of the aircraft and the roll axis and yaw axis define the plane of mass symmetry of the aircraft. In the subsequent development, we focus on the elevator deflection and the bank angle (determined by the deflection of the ailerons) and view them as primary pilot decision variables; the rudder deflection is a secondary pilot input and does not affect the flight optimization problem.

The body fixed frame is to be distinguished from an inertial frame, also shown in Fig. 15.3. The inertial frame is fixed to the ground (Earth). The inertial frame consists of three mutually perpendicular axes, denoted by x, y, z, whose origin is at a *fixed* location on the Earth. We ignore the curvature of the Earth, and we assume the x and y axes lie in a horizontal plane while the z axis is vertical. The inertial or ground fixed frame is used to describe the take off and landing locations and mission constraints, and it is important for navigational purposes.

5.3 Jet Aircraft Propulsion Systems

The aircraft jet engines, together with associated fuel tanks and related hardware, are referred to as a propulsion system. The purpose of the propulsion system is to generate a thrust force that propels the aircraft in flight. Although engines and propulsion systems are extremely complicated, detailed knowledge of the engine specifications is not required for the purpose of analyzing flight properties of an aircraft. Rather, the key features for flight optimization problems are the maximum thrust that the engines can produce and the rate at which fuel is consumed to produce a given thrust level.

The aircraft propulsion system produces a thrust force vector that has a fixed direction with respect to the aircraft. This property arises since the engines are typically fixed in the aircraft, either on the wings or on the fuselage. The direction of the thrust vector is assumed to lie along the aircraft roll axis in the plane of mass symmetry of the aircraft. Finally, it is assumed that the engine can be throttled or adjusted by the pilot so that any thrust level between zero thrust and maximum thrust can be achieved.

The scalar thrust produced by the propulsion system is a primary pilot decision variable. It is adjusted by the pilot to achieve a desired flight condition.

6 Selection of Helical Flight Path Segments

The first level in the proposed hierarchical decomposition approach is the selection of a flight path plan consisting of a concatenation of helical flight segments or paths in 3D that the aircraft should fly in order to meet the mission objectives and constraints. Each flight path segment is chosen to be the arc of a helix with a vertical central axis. This implies that the flight path angle (the angle the path makes with the horizontal) and the radius of curvature of the path are constant throughout the flight segment.

6.1 Background on Steady Flight

Steady aircraft flight occurs if the pilot inputs, thrust, elevator deflection and bank angle, are maintained constant. It can be shown that steady flight always leads to an aircraft flight path that is the arc of a helix with a vertical central axis [6]. This fact motivates our choice of helical flight path segments for the first level of the hierarchical decomposition. In addition, pilots and automated pilots almost always seek to fly along a concatenation of helical or steady flight segments, to the extent possible. This connection between helical flight and steady flight is made more explicit in the following summary of steady flight categories.

Over relatively short flight times, we can assume that the aircraft weight is constant and the air density is constant, that is the fuel consumed is negligible and the change

in altitude is negligible. In such a setting, steady flight has the property that all important flight variables, including the pilot inputs, are constant. It can be shown that the categories of steady flight are as follows [6]:

- *Steady cruise*: This is steady flight where the aircraft moves along a horizontal straight line. Such steady flight occurs when the bank angle is zero so that the aircraft does not turn. The pilot must select appropriate values of the thrust and elevator deflection to achieve steady straight line flight in a horizontal plane.
- *Steady level turning flight*: This is steady flight where the aircraft turns in a horizontal plane along the arc of a circle. The pilot must select appropriate values of the thrust, elevators and bank angle so that the flight path angle is zero and the turn radius is constant.
- *Steady longitudinal flight*: This is steady flight where the aircraft is climbing or descending along a straight line. Such steady flight occurs when the bank angle is zero so that the aircraft does not turn. The pilot must select appropriate values of the thrust and elevator deflection to achieve the specified flight path angle.
- *Steady helical flight*: It can be shown that this is the most general category of steady flight corresponding to motion of the aircraft along the arc of a helix with vertical axis. In fact, each of the above categories: steady cruise, steady level turning flight, and steady longitudinal flight can be viewed as special cases of steady helical flight. The pilot must select the bank angle, thrust and elevator deflection to achieve a particular steady helical flight condition.

6.2 Concatenation of Helical Flight Segments

Most aircraft are designed for normal operation along steady flight segments defined by a concatenation of helical flight segments in 3D, that is helical flight segments appropriately patched together. The concatenation of helical flight segments specify the aircraft path or route, e.g., take-off, climb, turn, cruise, turn, descent, and landing. Equivalently, most typical aircraft routes, planned by the pilot, a flight management computer, or air traffic control, are defined by a concatenation of helical flight paths that meet the mission constraints.

Consequently, the first phase of the optimal flight planning problem is to select a concatenation of helical flight segments or paths that meet the mission constraints and are consistent with the mission objective. There are typically many possible choices of flight paths that meet the mission constraints; this is primarily a 3D geometry problem to select a concatenation of segments of straight lines and segments of the arc of a circle (in a horizontal plane). Depending on the mission constraints, there are often natural choices for the flight paths. This phase of the flight optimization problem is based on simple 3D geometry of straight line and circular arc paths and heuristics.

6.3 Way points

Waypoints are the boundary points between helical flight path segments. Consequently, each helical flight path segment connects one way point with its succeeding way point. Each way point is described by its three coordinates in the inertial or ground fixed frame.

As the aircraft passes a way point, it transitions from one steady flight condition to the next steady flight condition; during these transition periods the flight is not steady and flight dynamics effects are important.

The flight path is assumed to be continuous at each way point; the slope of the helical path entering a way point may differ slightly from the slope of the helical path leaving the way point, but the change in slope should be small. The pilot inputs (thrust, elevator deflection, bank angle) and other flight variables such as flight path angle, air speed and turn radius can be discontinuous at a way point.

7 Descriptions of Steady Aircraft Flight

Knowledge of the physics of flight is essential as a basis for formulating and solving meaningful steady flight optimization problems. Flight physics involves the integration of several basic sciences, including aerodynamics, propulsion, and flight vehicle dynamics.

The fundamental physics associated with steady flight of a jet aircraft is summarized. We first provide a brief summary of aerodynamics. Important constraints on the flight variables that arise from aerodynamics, jet engine limitations, and aircraft structural limitations are identified. These are used to determine important relationships between the flight variables for an aircraft in steady flight. We then characterize optimal steady flight conditions for several categories of steady flight.

This material is covered in detail in many textbooks [1, 6]. The treatment here is brief and emphasis is given to those aspects of conventional jet aircraft that are most relevant to their flight performance characteristics.

7.1 Lists of Flight Parameters and Flight Variables

It is important to keep track of the notation and the meaning of the most important physical quantities associated with jet aircraft flight. This is facilitated by two lists that provide brief definitions and associated physical units of physical flight quantities that are subsequently introduced. The first list is of flight parameters that are characteristic of a specific aircraft; these parameters are constant throughout flight. The second list is of flight variables that can change throughout a flight.

Following are the important flight parameters:

C_{D_0} the zero-lift drag coefficient (dimensionless); an important aerodynamic parameter that determines the aircraft aerodynamic efficiency;
K the induced drag coefficient (dimensionless) that characterizes the part of the aerodynamic drag that is dependent on the lift generated by the wings;
C_{L_0} the lift coefficient (dimensionless) when the aircraft angle of attack is zero;
C_{L_α} the lift slope coefficient (1/degrees) that characterizes the dependence of the aerodynamic lift on the aircraft angle of attack (degrees);
S the total wing surface area (ft^2);
c_w the mean wing chord (ft);
C_{M_0} the pitching moment coefficient (dimensionless) when the angle of attack and elevator deflection are zero;
C_{M_α} the pitching moment coefficient (1/degrees) that characterizes the dependence of the aerodynamic pitching moment on the aircraft angle of attack (degrees);
$C_{M_{\delta_e}}$ the pitching moment coefficient (1/degrees) that characterizes the dependence of the aerodynamic pitching moment on the elevator deflection (degrees);
C_{Lmax} the aerodynamic stall coefficient that defines the maximum allowable value of the lift coefficient (dimensionless) for safe flight to avoid aircraft stall;
T^s_{max} the maximum thrust (lbs) that the jet engines and propulsion system can produce at sea level;
c the thrust specific fuel consumption rate; that is the rate at which fuel is consumed (lbs/s) per lb of thrust generated by the jet engines;
ρ^s the sea level value of the air density (slugs/ft^3) as determined from the Standard Atmospheric Model;
m a positive exponent (dimensionless) that is characteristic of a propulsion system;
n_{max} the maximum load factor (dimensionless); the maximum allowable value of the lift to aircraft weight ratio that the wing can support without exceeding its structural design specifications;
g the constant acceleration of gravity, namely 32.2 ft/s^2.

Following are the important flight variables:

δ_e the elevator deflection (degrees) or deflection of a moving flap on the horizontal tail that generates a pitching moment on the aircraft; a primary pilot decision variable;
μ the bank angle about the velocity vector (degrees) that causes the aircraft to turn; a primary pilot decision variable that is indirectly controlled using the ailerons;
r the turn radius (ft) of the aircraft; turning occurs when the aircraft has a nonzero bank angle whereas a zero bank angle makes the turn radius infinite;
T the thrust force (lbs) produced by the jet engines and propulsion system; this is a primary pilot decision variable;
V the air speed of the aircraft (ft/s) or, equivalently, the magnitude of the velocity vector of the aircraft assuming a stationary atmosphere;
W the net aircraft weight (lbs); over short time periods assumed constant while over longer time periods it decreases as fuel is consumed by the jet engines;
γ the flight path angle (radians) which is the angle between the velocity vector of the aircraft and the horizontal plane;
α the aircraft angle of attack (degrees) which is the angle between the velocity vector of the aircraft and the roll axis of the aircraft;
ρ the density of air (slugs/ft^3); depends on the flight altitude according to the Standard Atmospheric Model;

L the lift force (lbs) on the aircraft, primarily due to the air flow over the wings;
D the drag force (lbs) on the aircraft, due to friction effects and the lift generated by the wings;
C_L the lift coefficient (dimensionless);
C_D the drag coefficient (dimensionless);
C_M the pitching moment coefficient (dimensionless).

In the British units used here (lbs, ft, and s), the mass is measured in a unit known as a slug, where slug $= \frac{\text{lbs}}{\text{ft/s}^2}$.

7.2 Aerodynamics

The air density is an important atmospheric variable in flight studies since it influences the aerodynamic forces and moments that act on an aircraft in flight. The air density depends on the flight altitude and this can be an important effect: the air density decreases as the altitude increases. This dependence is tabulated in data tables referred to as the Standard Atmospheric Model; see [6] to obtain more complete data.

Altitude (ft)	Air density (slugs/cu ft)
sea level	0.0023769
1000	0.0023081
2000	0.0022409
3000	0.0022409
4000	0.0021109
5000	0.0020481
6000	0.0019867
10,000	0.0017553
15,000	0.0014956
20,000	0.0012664

We now assume that an aircraft is in steady flight at an altitude where the air density is ρ. We make a careful distinction between the aircraft velocity vector, which includes both the direction and magnitude of the aircraft velocity with respect to the ground frame, and the air speed of the aircraft, denoted by V, which is the aircraft speed relative to the surrounding air. Throughout our subsequent development, we assume the atmosphere is stationary (no atmospheric winds); consequently the air speed V is the magnitude of the aircraft velocity vector.

The equations that describe the lift force and the drag force on an aircraft in steady flight with air speed V are

$$L = \frac{1}{2}\rho V^2 SC_L, \tag{15.1}$$

$$D = \frac{1}{2}\rho V^2 SC_D. \tag{15.2}$$

Here C_L is the lift coefficient and C_D is the drag coefficient which are dimensionless aerodynamic variables and S is the total surface area of the wings.

The quadratic drag polar expression characterizes the aircraft aerodynamics and relates the drag coefficient to the lift coefficient by

$$C_D = C_{D_0} + K\, C_L^2. \tag{15.3}$$

Here the zero-lift drag coefficient C_{D_0} and the induced drag coefficient K are aerodynamic parameters. The lift coefficient C_L is a linear function of the angle of attack α of the aircraft as

$$C_L = C_{L_0} + C_{L_\alpha}\alpha. \tag{15.4}$$

The aircraft angle of attack is the angle that the velocity vector of the aircraft makes with the roll axis of the aircraft. Here C_{L_0} is the lift coefficient when the angle of attack is zero and C_{L_α} is the lift slope coefficient.

The pitch moment on an aircraft in steady flight with air speed V is

$$M = \frac{1}{2}\rho V^2\, S c_w C_M, \tag{15.5}$$

where c_w is the mean wing chord of the aircraft. The pitch moment coefficient C_M is a linear function of the angle of attack α and the elevator deflection δ_e as

$$C_M = C_{M_0} + C_{M_\alpha}\alpha + C_{M_{\delta_e}}\delta_e. \tag{15.6}$$

Here C_{M_0}, C_{M_α}, and $C_{M_{\delta_e}}$ are aerodynamic pitching moment coefficients.

7.3 Steady Flight Relationships

It is important to make a distinction between the flight parameters and the flight variables. This perspective is consistent with the viewpoint that the flight parameters are given constants that characterize the physical flight properties of the aircraft, while the flight variables are constants that define the specific steady flight conditions.

The weight of the aircraft is W so that its mass is $\frac{W}{g}$ where g is the acceleration of gravity. The lift force on the aircraft acts perpendicular to the velocity vector of the aircraft in the plane of mass symmetry of the aircraft; the lift force is rotated around the velocity vector by the bank angle μ. The bank angle $\mu = 0$ corresponds to a vertical lift force or, equivalently, wings level flight. The drag force on the aircraft acts in the direction opposite to the velocity vector of the aircraft. The thrust vector, with magnitude T, acts along the aircraft axial axis. The flight path angle γ is the angle between the velocity vector and the local horizontal plane. Also r denotes the instantaneous turn radius, the radius of the projection of the helical flight path onto a horizontal plane.

As is standard, we assume the thrust is much less than the lift. We also assume that the angle of attack and the flight path angle are small angles, if measured in radians, and we approximate the sine of the angle by the angle and the cosine of the angle by unity. We do not make a small angle assumption for the bank angle. Consequently Newton's equations of motion for steady flight are derived in [6]:

$$T = W\gamma + D, \tag{15.7}$$

$$\frac{W}{g}\frac{V^2}{r} = L\sin\mu, \tag{15.8}$$

$$L\cos\mu = W. \tag{15.9}$$

The first Eq. (15.7) is a balance of the components of the forces in the direction of the velocity vector of the aircraft, taking into account that there is no aircraft acceleration along the velocity vector. The second Eq. (15.8) is that the mass of the aircraft multiplied by the steady flight acceleration equals the force components in the (horizontal) radial direction. The third Eq. (15.9) is that the sum of the forces, projected onto the direction perpendicular to the velocity vector of the aircraft in the vertical plane, is zero since there is no aircraft acceleration in this direction.

A positive flight path angle γ corresponds to climbing flight, where the velocity vector lies above the local horizontal plane; a negative flight path angle corresponds to descending flight, where the velocity vector lies below the local horizontal plane. A positive bank angle μ corresponds to a right turn; a negative bank angle corresponds to a left turn, but the sign of the lift term in (15.8) must be changed in this case. The subsequent development holds for any flight path angle, be it positive, negative, or zero, and any bank angle, be it positive, negative, or zero.

In addition, the aircraft pitch moment $M = 0$ in steady flight so that

$$C_M = C_{M_0} + C_{M_\alpha}\alpha + C_{M_{\delta_e}}\delta_e = 0. \tag{15.10}$$

The elevator deflection can be adjusted by the pilot, depending on the flight angle of attack α, to satisfy this condition for steady flight.

These equations form the basis for our subsequent steady flight analysis.

7.4 Flight Constraints

There are three types of important steady flight constraints. These constraints play a central role in analyzing the performance properties of an aircraft in steady flight.

The first constraint is that the aircraft must avoid stalling. This is an aerodynamics constraint that characterizes the loss of lift that occurs when the aircraft angle of attack exceeds a certain threshold value. Stall is a complicated aerodynamic phenomenon that occurs due to substantial changes in the regularity of the flow of air over the wings, and a consequent loss of lift. All aircraft avoid this stall condition, since it can have serious adverse consequences for controlled flight. The stall constraint can

be expressed as an inequality on the angle of attack or equivalently as an inequality on the lift coefficient given by

$$C_L \leq C_{L_{\max}}, \tag{15.11}$$

where $C_{L_{max}}$ is the maximum allowable value of the lift coefficient for safe flight.

The second set of constraints arises from limits on the thrust that can be produced by the propulsion system. In mathematical form, the thrust T required for steady flight must satisfy

$$0 \leq T \leq T^s_{\max} \left(\frac{\rho}{\rho^s} \right)^m, \tag{15.12}$$

where $T^s_{\max}$ is the maximum thrust at sea level, ρ^s is the air density at sea level, and the propulsion exponent m is a positive parameter. These inequalities are subsequently referred to as the propulsion constraints or the jet engine thrust constraints.

The third steady flight constraint is a wing loading constraint that is expressed as

$$L \leq n_{max} W, \tag{15.13}$$

where n_{max} is the maximum value of the load factor. This constraint arises from structural limitations of the wing: the aerodynamic lift force on the wing should not exceed a safe level beyond which the wing might undergo structural damage. Expression (15.9) can be used to express the wing loading constraint as

$$\sec \mu \leq n_{max}. \tag{15.14}$$

7.5 Conditions for Steady Flight

Flight conditions for the most general type of steady flight are analyzed. Equations are obtained that describe the turn radius, the required thrust, the flight path angle, the air speed, and the bank angle for steady flight.

We first make use of Eqs. (15.9) and (15.1) to obtain an expression for the air speed of the aircraft in any steady flight condition

$$V = \sqrt{\frac{2W}{\rho S C_L \cos \mu}}. \tag{15.15}$$

Divide Eq. (15.8) by Eq. (15.9) to obtain

$$\tan \mu = \frac{V^2}{gr}. \tag{15.16}$$

Equation (15.16) can be rewritten so that the turn radius for steady flight is expressed in terms of the air speed and bank angle:

$$r = \frac{V^2}{g \tan \mu}. \tag{15.17}$$

Another expression for the turn radius can be given in terms of the bank angle using Eqs. (15.9), (15.1) and (15.17) to obtain

$$r = \frac{2}{\rho g C_L \sin\mu} \frac{W}{S}. \tag{15.18}$$

This expression for the turn radius of an aircraft in steady flight show that the turn radius depends on the lift coefficient, the bank angle, the wing loading $\frac{W}{S}$, and the flight altitude.

Using Eqs. (15.7) and (15.9) and the relation $\frac{D}{L} = \frac{C_D}{C_L}$, the required thrust for steady flight is

$$T = \left(\gamma + \frac{C_D}{C_L \cos\mu}\right) W. \tag{15.19}$$

Other expressions for the thrust for steady flight can be obtained. Expressing the lift in terms of the dynamic pressure and lift coefficient and using Eq. (15.9), the required lift coefficient for steady flight is

$$C_L = \frac{2W}{\rho V^2 S \cos\mu}. \tag{15.20}$$

Consequently, the required drag coefficient for steady flight is obtained from the quadratic drag polar expression (15.3) as

$$C_D = C_{D_0} + K\left(\frac{2W}{\rho V^2 S \cos\mu}\right)^2. \tag{15.21}$$

The drag on the aircraft in steady flight is

$$D = \frac{1}{2}\rho V^2 S C_{D_0} + \frac{2KW^2}{\rho V^2 S \cos^2\mu}. \tag{15.22}$$

Hence the required thrust for an aircraft in steady flight, expressed in terms of the air speed, bank angle, and flight path angle, is obtained using Eq. (15.7)

$$T = W\gamma + \frac{1}{2}\rho V^2 S C_{D_0} + \frac{2KW^2}{\rho V^2 S \cos^2\mu}. \tag{15.23}$$

There are three terms in the expression (15.23). The first term on the right of the equations describes that part of the thrust that is required for the aircraft is to climb or descend; this thrust term is positive for climbing flight and it is negative for descending flight. The second term represents the drag component that is due to aerodynamic friction and is independent of the lift and the bank angle; this drag term increases as the air speed increases. The third term is that part of the drag that is induced by the lift; this drag term increases as the air speed decreases or the bank angle increases. The total thrust required to maintain steady flight is the sum of all three terms.

These are a sample of important steady flight relationships; more complete development is given in [6]. The above steady flight equations are general in that they are expressed in terms of the aircraft thrust, the aircraft air speed, the flight path angle, and the bank angle. These equations can also be rewritten in terms of the aircraft throttle setting $\frac{T}{T^s_{max}}\left(\frac{\rho^s}{\rho}\right)^m$, the Mach number $\left(\frac{V}{a}\right)$ where a is the speed of sound, the climb rate $(V\gamma)$, and the load factor $\left(\frac{L}{W}\right)$.

7.6 *Conditions for Optimal Steady Flight*

Several flight conditions that characterize the optimal performance of jet aircraft in steady flight are presented. These performance measures characterize optimal flight at a single instant of time, not over a longer time period. The following instantaneous flight performance measures are analyzed: minimum required thrust, maximum air speed, minimum air speed, maximum flight path angle, minimum turn radius, and maximum altitude (flight ceiling). These flight performance measures characterize the flight maneuverability or flight agility of the aircraft. Not all of these results are used in our subsequent case study; but we include them as a sample of optimal flight performance results that are useful for various optimization objectives.

7.6.1 Minimum Thrust

Assume the bank angle is constant and satisfies the wing loading constraint (15.14). The thrust required for steady flight, with a fixed flight path angle and a fixed bank angle, has a minimum value that can be obtained using the methods of calculus.

Based on (15.19), the minimum thrust for steady flight occurs when the aerodynamics ratio $\frac{C_L}{C_D}$ is maximum. It can easily be shown using calculus that the maximum value of this ratio, using the standard quadratic drag polar expression (15.3), occurs when the lift coefficient and the drag coefficient are:

$$C_L = \sqrt{\frac{C_{D_0}}{K}}, \tag{15.24}$$

$$C_D = 2C_{D_0}. \tag{15.25}$$

The minimum value of the thrust required for steady flight, expressed in terms of the flight path angle and the bank angle, is obtained by substituting expression (15.24) into Eq. (15.19) to obtain

$$T = W\gamma + \frac{2W}{\cos\mu}\sqrt{KC_{D_0}}. \tag{15.26}$$

The minimum value of the thrust required for steady flight, for fixed values of the flight path angle and bank angle, does not depend on the air density and hence it does not depend on the flight altitude.

The thrust expressions given above must satisfy the thrust constraints given in (15.12). In particular, for a positive flight path angle the most important thrust constraint is typically the upper thrust limit, while for a negative flight path angle the thrust constraint defined by the lower limit of zero thrust must also be respected.

7.6.2 Minimum and Maximum Air Speeds

Assume the bank angle is constant and satisfies the wing loading constraint (15.14). Flight conditions for minimum and maximum air speeds of a jet aircraft are analyzed for steady flight with a fixed bank angle and a fixed flight path angle.

The maximum thrust that the jet engine can provide limits the minimum air speed and the maximum air speed for which the aircraft can maintain steady flight. In particular, the minimum air speed and the maximum air speed are characterized by the fact that the thrust required is equal to the maximum thrust that the jet engine can provide; this leads to the equation

$$W\gamma + \frac{1}{2}\rho V^2 SC_{D_0} + \frac{2KW^2}{\rho V^2 S\cos^2\mu} = T^s_{\max}\left(\frac{\rho}{\rho^s}\right)^m. \tag{15.27}$$

This equation can be rewritten as a quadratic equation in the square of the air speed; it has two positive solutions. The high speed solution is the maximum possible air speed of the aircraft in steady flight at the fixed flight path angle and the fixed bank angle; the low speed solution is the minimum possible air speed of the aircraft in steady flight at the fixed flight path angle and the fixed bank angle.

Two important qualifications must be made about the maximum air speed and the minimum air speed obtained from this analysis. This computed minimum air speed is the minimum air speed if it satisfies the stall constraint given by inequality (15.11) where the lift coefficient is given by (15.20); otherwise the stall speed, the air speed at which the stall constraint is active, is the minimum air speed for steady flight at the given thrust, flight path angle, and bank angle. The aerodynamics assumptions and model discussed here are only for subsonic flights. Most of the commercial flights are subsonic flights, and we only deal with these flights in this chapter. So, if the computed maximum air speed is near to or exceeds the local speed of sound, it is most likely that an error has been committed in its computation.

7.6.3 Maximum Flight Path Angle

Assume the bank angle satisfies the wing loading constraint (15.14). Flight conditions for the maximum flight path angle of a jet aircraft are analyzed for steady flight with a fixed bank angle.

Equation (15.19) can be expressed as

$$\gamma = \frac{T}{W} - \frac{C_D}{C_L\cos\mu}. \tag{15.28}$$

Thus the maximum flight path angle occurs if the thrust provided by the jet engine is maximum and the aerodynamics ratio $\frac{C_L}{C_D}$ is a maximum. Consequently, the aerodynamic coefficients satisfy (15.24) and (15.25).

The maximum flight path angle is given by

$$\gamma = \frac{T^s_{\max}}{W}\left(\frac{\rho}{\rho^s}\right)^m - \frac{2\sqrt{KC_{D_0}}}{\cos\mu}. \tag{15.29}$$

There is also a minimum flight path angle for steady flight with a given bank angle. This occurs when the thrust provided by the engine is minimum, that is the thrust is zero, and the drag is maximum; a local maximum of the drag occurs at the stall speed, the air speed for which the stall constraint is active. This result is not developed in further detail.

7.6.4 Minimum Radius Turns

Flight conditions for a minimum radius steady turn of a jet aircraft with a fixed flight path angle are now considered. The minimum radius turn optimization problem in steady flight can be formulated as: determine the values of air speed V and bank angle μ for which the steady flight turn radius

$$r = \frac{V^2}{g\tan\mu}, \tag{15.30}$$

is a minimum, subject to satisfaction of the three inequalities defined by the stall constraint, the thrust constraints and the wing loading constraint:

$$\frac{2W}{\rho V^2\, S\cos\mu} \leq C_{Lmax}, \tag{15.31}$$

$$W\gamma + \frac{1}{2}\rho V^2\, SC_{D_0} + \frac{2KW^2}{\rho V^2\, S\cos^2\mu} \leq \left(\frac{\rho}{\rho^s}\right)^m T^s_{max}, \tag{15.32}$$

$$\sec\mu \leq n_{max}. \tag{15.33}$$

This optimization problem is complicated by the fact that the active constraints and the inactive constraints are not a priori known. Any one of the following situations is possible: there is one active constraint, there are two active constraints, and all three constraints are active. Since any one of these cases can characterize the minimum radius turn for a jet aircraft in steady flight, in principle all cases need to be analyzed and the corresponding flight condition and the corresponding steady turn radius computed for each case. The minimum radius turn for the jet aircraft in steady flight is then determined by a direct comparison of the computed turn radius values for which all the constraints are satisfied. In some instances, one can make an educated hypothesis about which are the active constraints and the inactive constraints for the minimum radius turn.

Equation (15.18) makes clear that the smallest possible turn radius for a jet aircraft is achieved, all other factors considered constant, at the lowest possible altitude.

7.6.5 Maximum Altitude

Assume that the flight path angle and the bank angle are zero. There is a maximum altitude, referred to as the steady flight ceiling, at which steady level flight can be maintained. The flight conditions at this flight ceiling are characterized by the fact that the minimum thrust required for steady level flight, given by Eq. (15.26), equals the maximum thrust that the jet engine can produce at this altitude. This leads to the equation

$$W + 2W\sqrt{KC_{D_0}} = T^s_{\max}\left(\frac{\rho}{\rho^s}\right)^m. \tag{15.34}$$

Since the air density depends on the altitude according to the Standard Atmospheric Model, Eq. (15.34) can be viewed as an implicit equation in the altitude; the flight ceiling solves this equation.

7.7 Fuel Consumption in Steady Flight

The prior developments are based on a steady flight assumption that the pilot inputs of thrust, elevator deflection and bank angle are constant and the aircraft responses of air speed, flight path angle and turn radius are constant. Since, in fact the propulsion system consumes fuel so that the weight of the aircraft decreases, this assumption holds for relatively short time periods on the order of seconds or a few minutes. For longer time periods on the order of many minutes, the fuel consumption and the resulting decrease in the weight of the aircraft must be taken into account. Fuel consumption features are included in the subsequent analysis; we obtain important results for the fuel consumed in steady flight of a fixed distance and for the time of steady flight using a fixed amount of fuel.

For an ideal jet engine, the rate at which fuel is burned is proportional to the thrust that the engine produces. Let W denote the total weight of the aircraft, including the fuel. Then the rate at which fuel is consumed is given by

$$\frac{dW}{dt} = -cT, \tag{15.35}$$

where T denotes the thrust produced by the jet engine and c is the thrust specific fuel consumption rate. This fuel consumption rate is an important engine parameter. It denotes the rate at which fuel is burned per unit of thrust produced by the engine. For ideal jet engines this thrust specific fuel consumption rate is assumed to be a positive constant parameter independent of the flight conditions of the aircraft.

The fact that the aircraft weight decreases at a small rate allow us to make use of the prior results for steady flight. This is an approximation but in many cases it leads to accurate conclusions.

7.7.1 Distance Traveled in Steady Flight

We are interested in the distance traveled by an aircraft in flight over long time periods, assuming the bank angle, the flight path angle and the lift and drag coefficients are constants. Since the weight of the aircraft slowly changes as the fuel is consumed, the steady flight relations derived previously remain valid but with the interpretation that the thrust and the air speed may change slowly due to the slow changes in the aircraft weight. Recognizing the abuse of terminology, we still refer to this as steady flight since the aircraft does fly along the arc of a helix with vertical axis.

Let s denote the distance traveled along the arc of a helix with vertical axis. Steady flight implies that the air speed is

$$\frac{ds}{dt} = V. \tag{15.36}$$

The air speed is given by (15.15) so that

$$\frac{ds}{dt} = \sqrt{\frac{2W}{\rho S C_L \cos\mu}}. \tag{15.37}$$

Some of the following equations invlove differentials such as ds; dt; dW; these are the differentials (or increments) of distance travelled s, elapsed flight time t, and aircraft weight W.

Our objective is to obtain an expression for the aircraft distance traveled in terms of the fuel consumed during the flight duration. To this end, Eqs. (15.19), (15.35), (15.36), and (15.37) can be combined to obtain

$$ds = -\frac{1}{c}\sqrt{\frac{2}{\rho S}}\left(\frac{\sqrt{C_L \cos\mu}}{\gamma C_L \cos\mu + C_D}\right)\frac{dW}{\sqrt{W}}.$$

Therefore, the distance traveled by the aircraft during a time interval $[t_i, t_f]$, denoted by d_{i-f} is obtained by integrating the above equation from the initial time t_i, when the weight of the aircraft including fuel is W_i, to the final time t_f, when the weight of the aircraft and any remaining fuel is W_f. Clearly, $W_i > W_f$. Since the bank angle, flight path angle and lift and drag coefficients are constants, the resulting expression for the distance traveled is

$$d_{i-f} = -\frac{1}{c}\left(\frac{\sqrt{C_L \cos\mu}}{\gamma C_L \cos\mu + C_D}\right)\int_{W_i}^{W_f}\sqrt{\frac{2}{\rho S}}\frac{dW}{\sqrt{W}}. \tag{15.38}$$

If the change in altitude is not large, the integral can be approximated using an average constant value for the air density. This gives the algebraic expression for the distance traveled:

$$d_{i-f} = \frac{2}{c}\left(\frac{\sqrt{C_L \cos\mu}}{\gamma C_L \cos\mu + C_D}\right)\sqrt{\frac{2}{\rho S}}\left(\sqrt{W_i} - \sqrt{W_f}\right). \tag{15.39}$$

This formula gives the distance traveled for a jet aircraft in steady flight, assuming that the lift to drag ratio appearing in Eq. (15.39), the bank angle and the flight path angle are maintained constant throughout the flight. The formula is exact if the flight path angle is zero; otherwise it is an approximation that is valid for relatively small changes in flight altitude (several thousand feet).

Equation (15.39) can be solved to obtain an expression for the terminal weight of the aircraft for a given value of the distance traveled in steady flight. This assumes that sufficient fuel is available. The final aircraft weight, taking into account the fuel consumed, to achieve a given flight distance d_{i-f} is

$$W_f = \left(\sqrt{W_i} - \frac{c d_{i-f}}{2} \sqrt{\frac{\rho S}{2}} \left(\frac{\gamma C_L \cos\mu + C_D}{\sqrt{C_L \cos\mu}} \right) \right)^2 . \tag{15.40}$$

The most important results for steady cruise, when the bank angle and flight path angle are zero, are easily obtained from these results. The distance traveled in steady cruise is

$$d_{i-f} = \frac{2}{c} \sqrt{\frac{2}{\rho S}} \left(\frac{\sqrt{C_L}}{C_D} \right) \left(\sqrt{W_i} - \sqrt{W_f} \right) . \tag{15.41}$$

The final aircraft weight in steady cruise of a given distance is

$$W_f = \left(\sqrt{W_i} - \frac{c d_{i-f}}{2} \sqrt{\frac{\rho S}{2}} \frac{C_D}{\sqrt{C_L}} \right)^2 . \tag{15.42}$$

It is seen that the weight of fuel consumed for flight of a given distance depends on the flight altitude. In particular, the fuel consumed for a fixed flight distance decreases as the air density decreases, that is as the altitude increases. Consequently, ignoring the fuel required to climb to a given altitude, the fuel consumed is reduced by steady level flight at the highest possible altitude below the flight ceiling.

7.7.2 Minimum Fuel Consumed in Steady Flight

According to Eq. (15.40), the weight of fuel required to fly a fixed distance is minimized if the value of the lift coefficient and the drag coefficient are selected to maximize the expression

$$\left(\frac{\sqrt{C_L \cos\mu}}{\gamma C_L \cos\mu + C_D} \right) , \tag{15.43}$$

subject to satisfaction of the drag polar expression (15.3). It can easily be shown using calculus that the maximum value of this expression occurs when the lift coefficient is:

$$C_L = \frac{1}{2} \sqrt{\left(\frac{\gamma \cos\mu}{3K} \right)^2 + \frac{4}{3} \frac{C_{D_0}}{K}} - \left(\frac{\gamma \cos\mu}{6K} \right) . \tag{15.44}$$

The corresponding value of the drag coefficient is computed from the drag polar expression (15.3).

If the aircraft is in steady level flight so that the flight path angle is zero, then the minimum fuel consumed in steady level flight occurs when the lift coefficient and the drag coefficient are:

$$C_L = \sqrt{\frac{C_{D_0}}{3K}}, \tag{15.45}$$

$$C_D = \frac{4}{3} C_{D_0}. \tag{15.46}$$

7.7.3 Elapsed Time in Steady Flight

The aircraft time of flight is the time that an aircraft remains in steady flight using a fixed amount of fuel. The procedure to derive an equation for time of flight is similar to that used in the previous section.

Equations (15.19) and (15.35) are rewritten as

$$dt = -\frac{1}{c}\left(\frac{C_L \cos\mu}{\gamma C_L \cos\mu + C_D}\right)\frac{dW}{W}.$$

The time that the airplane remains in steady flight, denoted by t_{i-f}, is obtained by integrating this equation from the initial time t_i, when the aircraft weight is W_i to the final time t_f, when the aircraft weight is W_f:

$$t_{i-f} = -\frac{1}{c}\left(\frac{C_L \cos\mu}{\gamma C_L \cos\mu + C_D}\right)\int_{W_i}^{W_f}\frac{dW}{W}.$$

Evaluating the integral gives the expression for the time of flight

$$t_{i-f} = \frac{1}{c}\left(\frac{C_L \cos\mu}{\gamma C_L \cos\mu + C_D}\right)\ln\frac{W_i}{W_f}. \tag{15.47}$$

This time of flight formula for a jet powered aircraft is based on the assumption that the lift to drag ratio, flight path angle and bank angle that appear in Eq. (15.47) are maintained constant throughout the steady flight segment. Since the time of flight for a jet aircraft does not depend on the air density, it follows that the time of flight is independent of the flight altitude.

8 Characteristics of a Business Jet Aircraft

This section introduces details that describe a typical business jet aircraft that is capable of carrying up to eight passengers plus crew and cargo. The aircraft is fictitious in the sense that it does not represent any specific existing aircraft. The

aircraft data is consistent with typical aircraft of this type. A fuel optimal flight planning problem for this specific jet aircraft is introduced subsequently.

The weight of the aircraft, with a full fuel tank, is 73,000 lbs. The wing surface area is 950 ft^2. The aerodynamic drag polar is given by

$$C_D = 0.015 + 0.05C_L^2.$$

The lift coefficient, in terms of the angle of attack α, is

$$C_L = 0.02 + 0.12\alpha,$$

where the angle of attack is measured in degrees. The maximum lift coefficient at stall is 2.8 which occurs at an angle of attack of 23.2°. The pitch moment coefficient, expressed in terms of the angle of attack α and the elevator deflection δ_e, is given by

$$C_M = 0.24 - 0.18\alpha + 0.28\delta_e,$$

where the angle of attack α and the elevator deflection δ_e are measured in degrees.

The jet aircraft is powered by two jet engines, each of which can provide a maximum sea-level thrust of 6250 lbs at sea level. The engines are configured in the aircraft so that they do not generate a moment on the aircraft. The maximum sea level thrust that can be produced by the propulsion system is T_{max}^s = 12,500 lbs and the exponent in the propulsion system constraint is $m = 0.6$. The maximum fuel that can be carried in the aircraft is 20,000 lbs. The thrust specific fuel consumption rate for the jet engines is 0.69 lbs/hr of fuel consumed for each lb of thrust generated by the engines.

The maximum load factor for the business jet aircraft is 2.0; this corresponds to a maximum bank angle constraint of 60°.

Hence, the flight parameters for the business jet aircraft are:

$$\begin{aligned}
C_{L_0} &= 0.02, & C_{L_\alpha} &= 0.12, & C_{L_{\max}} &= 2.8,\\
C_{D_0} &= 0.015, & K &= 0.05, & &\\
C_{M_0} &= 0.24, & C_{M_\alpha} &= -0.18, & C_{M_{\delta_e}} &= 0.28,\\
S &= 950\,\text{ft}^2, & n_{max} &= 2, & &\\
T_{max}^s &= 12,500\,\text{lbs} & m &= 0.6, & c &= 0.0001917\,1/\text{s},
\end{aligned}$$

9 A Fuel Optimal Flight Planning Case Study

A fuel optimal flight planning problem for the business jet aircraft is formulated as a case study. The objective is that the aircraft take off with a full fuel tank along a run way at a given location and land along a run way at another location, using the least possible fuel.

A sequence of helical flight segments and way points is selected that begins at the take-off location and ends at the landing location. The way points are selected in a natural way to define successive helical path segments corresponding to a steady climb phase, a steady level turn, a steady level cruise phase, and a steady descent phase. The flight conditions for fuel optimal steady flight are obtained for each flight segment. Detailed calculations are given for the specific jet aircraft described in the prior section.

9.1 Description of the Optimal Flight Planning Problem

We now give a formal description of the optimal flight planning problem.

A ground fixed frame is selected with origin at sea level: the x and y axes are in the horizontal plane located at sea level with the x axis pointing due East and the y axis pointing due North; the z axis defines the altitude above the sea level horizontal plane. We specify an aircraft location by a triple (x, y, z), with each coordinate measured in feet.

The takeoff location on an East–West oriented run way is $(0, 0, 2000)$ and the landing location on an East–West run way is $(-200,000, 20,000, 2000)$ in the ground fixed frame. Note that the take off and landing run ways are parallel and are located at an altitude of 2000 ft above sea level.

Formally, the flight planning problem is to determine the fuel optimal flight plan so that the aircraft takes off from an East–West runway at the location given by $(0, 0, 2000)$ and lands on an East–West run way at the location $(-200,000, 20,000, 2000)$. The flight planning problem should provide a minimum fuel flight plan that satisfies these mission constraints and the stall constraint

$$C_L \leq 2.8, \tag{15.48}$$

the propulsion constraints

$$0 \leq T \leq \left(\frac{\rho}{2.3769 \times 10^{-3}}\right)^{0.6} 12,500 \,\text{lbs},$$

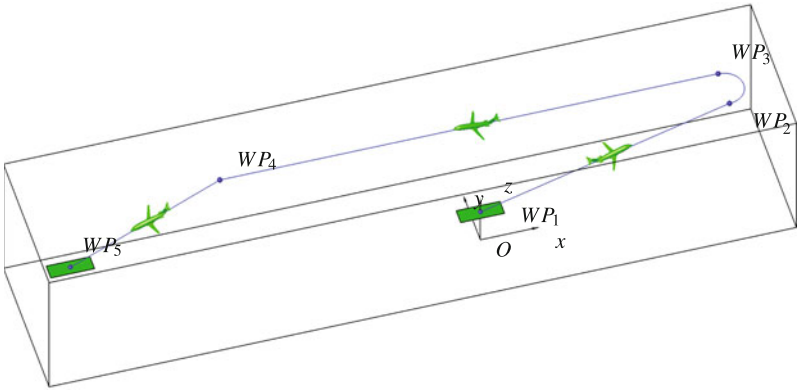
and the wing loading constraint

$$L \leq 2W. \tag{15.49}$$

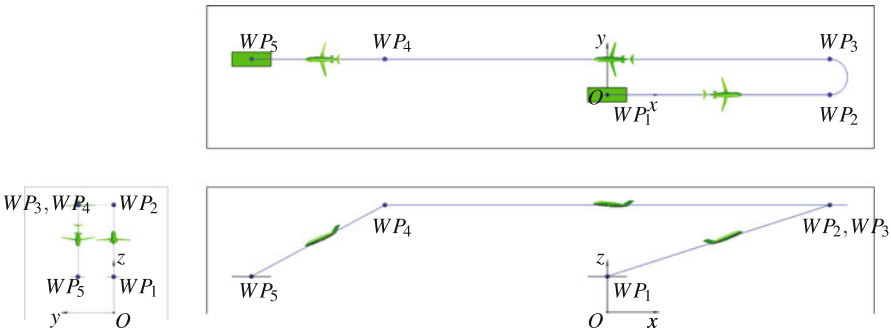
The total time that the aircraft takes, from take-off to landing, is not constrained.

9.2 Selection of Helical Paths and Way Points

There are many ways to construct a sequence of helical flight path segments between the take off location with a due East or due West take off direction and the landing



a 3D perspective



b Three perspectives (looking down, east, and north, counterclockwise from top)

Fig. 15.4 Selected way points and helical flight segments

location with a due East or due West landing direction. These helical flight path segments naturally consist of a straight line climbing flight segment from take off and a straight line descending flight segment to landing; since the take off and landing run ways are not co-linear, there is necessarily a turning flight segment and perhaps a steady cruise flight segment (Fig. 15.4).

This is a relatively simple 3D geometry problem. The choice we make is to take off in the due East direction along a straight line path (a steady climbing flight segment) until reaching an altitude of 6000 ft; then to carry out a level left turn corresponding to 180° of a circular arc of radius 10,000 ft so that the aircraft is flying due West; then to follow a straight line cruise before finally beginning a straight line path (a steady descending flight segment) to the landing location in the due West direction. The heuristic choice of cruising altitude is arbitrary, but our choice represents a trade off between the fuel efficiency of higher altitude flight balanced by the fuel required to climb to and descend from altitude.

There are many other sequences of helical paths that one might select. For example, one possible sequence is defined by a due West take off with a climb, right turn, left turn, and descent. This sequence of flight segments may require less flight time, but it may require aggressive turns that use extra fuel and may not satisfy the aircraft flight constraints. One would have to carry out a similar analysis for this flight plan to determine if the flight paths are feasible and the fuel required to implement the flight plan.

In the following, the way points are numbered and located by their 3D coordinates; the flight segment that follows the way points are indicated.

Way point	Coordinates (x,y,z) in ft	Flight segment
1	$(0, 0, 2000)$	Climb
2	$(125,000, 0, 6000)$	Level turn
3	$(125,000, 20,000, 6000)$	Cruise
4	$(-125,000, 20,000, 6000)$	Descent
5	$(-200,000, 20,000, 2000)$	–

Further, the selected helical flight path segments are described as follows:

- Fuel optimal steady climbing flight between way points 1 and 2 following a straight line path between way points 1 and 2 with a constant flight path angle of 1.833°.
- Fuel optimal steady level left turn between way points 2 and 3 following a horizontal circular path between these two way points with constant turn radius of 10,000 ft; the flight direction changes by 180°, from due East to due West, during this flight segment.
- Fuel optimal steady level cruising flight between way points 3 and 4 following a straight line constant altitude path between these two way points; the distance traveled between these two way points is 250,000 ft.
- Fuel optimal steady descending flight between way points 4 and 5 following a straight line path between these two way points with a constant flight path angle of -3.053°.

The overall flight path consists of a concatenation of these four helical flight path segments. This concatenation of helical flight path segments defines a continuous flight path in 3D. The slope at way point 2 is discontinuous and the slope at way point 4 is discontinuous; this is unavoidable when connecting a climbing or descending straight line path with a steady level circular path. The slope of the flight path at way point 3 is continuous.

Each of the flight segments is analyzed in detail to obtain fuel optimal flight conditions. A summary of results is given.

9.2.1 Fuel Optimal Flight Conditions Between Way Point 1 and Way Point 2

In this and the subsequent sections, we use an integer subscript and a symbol superscript of a flight variable to denote the value of that flight variable at the way point

indicated by that integer. For example, V_1^+ denotes the value of the air speed at way point 1 just after take off, T_4^- denotes the value of the thrust just before way point 4, T_4^+ denotes the value of the thrust just after way point 4, and V_5^- denotes the value of the air speed at way point 5 just before landing. Since the aircraft weight and air density are necessarily continuous at a way point, the superscript symbol is omitted in such cases.

Assume the bank angle is zero so that the aircraft remains in the vertical x–z plane in a steady climb between way points 1 and 2 with flight path angle $\gamma = 0.0319\,\text{rad} = 1.833^\circ$ that is constant throughout this flight segment.

We determine the lift coefficient and the drag coefficient to provide the minimum consumed fuel during this flight segment. The pilot inputs, thrust and elevator deflection, are selected so that the resulting flight path angle and lift coefficient are maintained constant throughout the flight segment. An approximation of the aircraft weight is determined at way point 2, and this is used to determine the flight conditions at way point 2.

The fuel consumed during a steady climbing flight segment is a minimum if the aerodynamics expression in (15.43) is maximized, which occurs if the lift coefficient is given by (15.44)

$$C_L = \frac{1}{2}\sqrt{\left(\frac{\gamma}{3K}\right)^2 + \frac{4}{3}\frac{C_{D_0}}{K}} - \left(\frac{\gamma}{6K}\right) = 0.227.$$

The drag coefficient is thus

$$C_D = C_{D_0} + KC_L^2 = 0.018.$$

The lift coefficient and drag coefficient are constant throughout this climbing flight segment between way point 1 and way point 2.

The take off weight W_1 is assumed to be the aircraft weight with full fuel tank. The air speed at way point 1, just after take off, is given by (15.15) as

$$V_1^+ = \sqrt{\frac{2W_1}{\rho_1 S C_L}} = 530.0\,\text{ft/s}.$$

The value of the thrust at way point 1 is given by (15.19) as

$$T_1^+ = \left(\gamma + \frac{C_D}{C_L}\right) W_1 = 8117.2\,\text{lbs}.$$

Equations (15.4) and (15.10), together with the value of the lift coefficient, are used to determine the values of the the angle of attack and the elevator deflection at way point 1 as:

$$\alpha_1^+ = 1.73^\circ,\ \delta_{e_1}^+ = 0.26^\circ.$$

The distance traveled is $d_{1-2} = 125{,}060$ ft and the average air density is $\rho = 0.00211$ slugs/ft^3, which is the air density value at an altitude of 4000 ft. Thus the aircraft weight at way point 2 is

$$W_2 = \left(\sqrt{W_1} - \frac{c d_{1-2}}{2} \sqrt{\frac{\rho S}{2}} \left(\frac{\gamma C_L + C_D}{\sqrt{C_L}} \right) \right)^2 = 72{,}658.8 \text{ lbs},$$

so that the fuel consumed in flight between way point 1 and way point 2 is 341.2 lbs. The time to complete this climbing flight segment is

$$t_{1-2} = \frac{1}{c} \left(\frac{C_L}{\gamma C_L + C_D} \right) \ln \frac{W_1}{W_2} = 220.1 \text{ s}.$$

The air speed at way point 2 on this climbing flight segment is given by (15.15) as

$$V_2^- = \sqrt{\frac{2W_2}{\rho_2 S C_L}} = 582.5 \text{ ft/s}.$$

The value of the thrust at way point 2 is given by (15.19) as

$$T_2^- = \left(\gamma + \frac{C_D}{C_L} \right) W_2 = 8079.3 \text{ lbs}.$$

Equations (15.4) and (15.10), together with the value of the lift coefficient, are used to determine the values of the the angle of attack and the elevator deflection at way point 2 as:

$$\alpha_2^- = 1.73^\circ, \ \delta_{e_2}^- = 0.26^\circ.$$

Since the lift coefficient is constant throughout this climbing flight segment, the angle of attack and the elevator deflection do not change throughout this flight segment.

The thrust must be adjusted by the pilot throughout the climbing flight segment to compensate for the slow decrease in the aircraft weight as fuel is consumed by the jet engine and the slow decrease in the air density as the aircraft climbs. The value of the thrust at any point on the climbing flight segment between way point 1 and way point 2 can be computed using linear interpolation between thrust values at way point 1 and way point 2.

It is easy to check that all flight constraints are satisfied for this steady climbing flight segment.

9.2.2 Fuel Optimal Flight Conditions Between Way Point 2 and Way Point 3

The bank angle is selected so that the aircraft follows the arc of a circle in a horizontal plane at an altitude of 6000 ft between way point 2 and way point 3. The

distance traveled along 180° of a circular arc of radius 10,000 ft during this flight segment is 31,415.9 ft. The expression for the aircraft weight at way point 3 given by (15.42) implies that the minimum fuel flight condition is obtained if the aerodynamics ratio $\frac{\sqrt{C_L}}{C_D}$ is a maximum throughout the flight segment. The required lift and drag coefficients are computed from (15.45) and (15.46) as

$$C_L = \sqrt{\frac{C_{D_0}}{3K}} = 0.316\,,$$
$$C_D = \frac{4}{3}C_{D_0} = 0.020\,.$$

These should remain constant throughout this turning flight segment. Thus the required bank angle is computed from (15.18) as

$$\sin\mu = \frac{2}{r\rho g C_L}\frac{W_2}{S} = 0.757,$$

so that the constant bank angle is $\mu = -49.2°$, where the negative sign denotes a left turn. The pilot maintains the bank angle constant throughout the turning flight segment.

Using (15.40), the weight of the aircraft at way point 3 is

$$W_3 = \left(\sqrt{W_2} - \frac{cd_{2-3}}{2}\sqrt{\frac{\rho S}{2}}\frac{C_D}{\sqrt{C_L\cos\mu}}\right)^2 = 72587.8\,\text{lbs},$$

where $d_{3-4} = 31{,}415.9$ ft is the distance traveled on this steady level turning flight segment. Thus the fuel consumed in the steady turn between way point 2 and way point 3 is 70.9 lbs.

Using (15.47), the time of flight during the steady level turn is

$$t_{2-3} = \frac{1}{c}\left(\frac{C_L\cos\mu}{C_D}\right)\ln\frac{W_2}{W_3} = 52.7\,\text{s}.$$

The air speed and thrust at way point 2 are given by (15.15) and (15.19) as

$$V_2^+ = \sqrt{\frac{2W_2}{\rho S C_L\cos\mu}} = 610.7\,\text{ft/s},$$

and

$$T_2^+ = \left(\frac{C_D}{C_L\cos\mu}\right)W_2 = 7036.9\,\text{lbs}.$$

Equations (15.4) and (15.10), together with the value of the lift coefficient, are used to determine the values of the angle of attack and the elevator deflection at way point 2 as

$$\alpha_2^+ = 2.47°,\ \delta_{e_2}^+ = 0.73°.$$

The air speed and thrust at way point 3 are given by (15.15) and (15.19) as

$$V_3^- = \sqrt{\frac{2W_3}{\rho S C_L \cos\mu}} = 610.4\,\text{ft/s},$$

and

$$T_3^- = \left(\frac{C_D}{C_L \cos\mu}\right) W_3 = 7030.1\,\text{lbs}.$$

The air speed and thrust values at way point 3 are slightly less than the air speed and thrust values at way point 2, reflecting the fuel consumed by the steady level turn between these way points.

Equations (15.4) and (15.10), and the value of the lift coefficient, are used to determine the values of the angle of attack and the elevator deflection at way point 3 as:

$$\alpha_3^- = 2.47^\circ,\ \delta_{e_3}^- = 0.73^\circ,$$

which are not changed from their values at way point 2.

The thrust must be adjusted by the pilot to compensate for the slow decrease in the aircraft weight as fuel is consumed by the jet engine. The value of the thrust at any point on this steady turning flight segment between way point 2 and way point 3 can be computed using linear interpolation between the thrust values at way point 2 and way point 3.

It is easy to check that all flight constraints are satisfied for this steady climbing flight segment.

9.2.3 Fuel Optimal Flight Conditions Between Way Point 3 and Way Point 4

Assume the bank angle and flight path angle are zero along the straight line cruising flight segment between way point 3 and way point 4. Equation (15.42) implies that the fuel optimal flight conditions are obtained if the aerodynamics ratio $\frac{\sqrt{C_L}}{C_D}$ is a maximum throughout the flight segment. The required lift and drag coefficients are computed from (15.45) and (15.46) as

$$C_L = \sqrt{\frac{C_{D_0}}{3K}} = 0.316\,,$$
$$C_D = \frac{4}{3} C_{D_0} = 0.020\,.$$

Using (15.42), the aircraft weight at way point 4 is

$$W_4 = \left(\sqrt{W_3} - \frac{c d_{3\text{–}4}}{2}\sqrt{\frac{\rho S}{2}}\frac{C_D}{\sqrt{C_L}}\right)^2 = 72,141.6\,\text{lbs},$$

where $d_{4\text{–}5} = 250{,}000$ ft is the distance traveled on this flight segment. Thus the fuel consumed during the steady cruise between way point 3 and way point 4 is 446.2 lbs.

Using (15.47), the time of flight during the steady cruise is

$$t_{3\text{–}4} = \frac{1}{c}\left(\frac{C_L}{C_D}\right)\ln\frac{W_3}{W_4} = 508.2\,\text{s}.$$

The air speed and thrust at way point 3 are given by (15.15) and (15.19) as

$$V_3^+ = \sqrt{\frac{2W_3}{\rho S C_L}} = 493.5\,\text{ft/s},$$

and

$$T_3^+ = \left(\frac{C_D}{C_L}\right)W_3 = 4594.2\,\text{lbs}.$$

Equations (15.4) and (15.10), together with the value of the lift coefficient, are used to determine the values of the angle of attack and the elevator deflection at way point 3 as:

$$\alpha_3^+ = 2.47^\circ,\ \delta_{e_3}^+ = 0.73^\circ.$$

The air speed and thrust at way point 4 are given by (15.15) and (15.19) as

$$V_4^- = \sqrt{\frac{2W_4}{\rho S C_L}} = 491.9\,\text{ft/s},$$

and

$$T_4^- = \left(\frac{C_D}{C_L}\right)W_4 = 4565.9\,\text{lbs}.$$

The air speed and thrust values at way point 4 are slightly less than the air speed and thrust values at way point 3, reflecting the consumption of fuel in flight between these way points.

Equations (15.4) and (15.10), together with the value of the lift coefficient, are used to determine the values of the angle of attack and the elevator deflection at way point 4 as:

$$\alpha_4^- = 2.47^\circ,\ \delta_{e_4}^- = 0.73^\circ,$$

which are not changed from their values at way point 3.

The pilot must adjust the thrust produced by the propulsion system to compensate for the slow decrease in the aircraft weight as fuel is consumed. The value of the thrust at any point on this steady cruising flight segment between way point 3 and way point 4 can be computed using linear interpolation between the thrust values at way point 3 and way point 4.

It is easy to check that all flight constraints are satisfied for this steady climbing flight segment.

9.2.4 Fuel Optimal Flight Conditions Between Way Point 4 and Way Point 5

Assume the bank angle is zero so that the aircraft remains in the vertical x–z plane in a steady descent between way points 4 and 5 with flight path angle $\gamma = -0.0533$ rad $= -3.05^\circ$ that is constant throughout this descending flight segment. This last flight segment terminates just before landing.

We determine the lift coefficient and the drag coefficient that gives the minimum consumed fuel during this flight segment. The pilot inputs, thrust and elevator deflection, are selected so that the resulting flight path angle and lift coefficient are maintained constant throughout the flight segment. An approximation of the aircraft weight is determined at way point 5, and this is used to determine the flight conditions at way point 5.

The fuel consumed during a steady descending flight segment is a minimum if the aerodynamics expression in (15.43) is maximized, which occurs if the lift coefficient is given by (15.44)

$$C_L = \frac{1}{2}\sqrt{\left(\frac{\gamma}{3K}\right)^2 + \frac{4}{3}\frac{C_{D_0}}{K}} - \left(\frac{\gamma}{6K}\right) = 0.541.$$

The drag coefficient is thus

$$C_D = C_{D_0} + KC_L^2 = 0.030,$$

and the lift coefficient and drag coefficient are constant throughout this descending flight segment between way point 4 and way point 5.

The air speed at way point 4 is given by (15.15) as

$$V_4^+ = \sqrt{\frac{2W_4}{\rho_4 S C_L}} = 375.9\,\text{ft/s}.$$

The value of the thrust at way point 4 is given by (15.19) as

$$T_4^+ = \left(\gamma + \frac{C_D}{C_L}\right) W_4 = 101.9\,\text{lbs}.$$

Equations (15.4) and (15.10), together with the value of the lift coefficient, are used to determine the values of the the angle of attack and the elevator deflection at way point 4 as:

$$\alpha_4^+ = 4.34^\circ,\ \delta_{e_4}^+ = 2.55^\circ.$$

The distance traveled is $d_{4\text{–}5} = 75{,}106.6$ ft and the average air density is $\rho = 0.00211$ slugs/ft^3, which is the air density value at an altitude of 4000 ft. Thus the aircraft weight at way point 5, just before landing, is

$$W_5 = \left(\sqrt{W_4} - \frac{cd_{4\text{–}5}}{2}\sqrt{\frac{\rho S}{2}}\left(\frac{\gamma C_L + C_D}{\sqrt{C_L}}\right)\right)^2 = 72{,}136.3\,\text{lbs},$$

so that the fuel consumed in flight between way point 4 and way point 5 is 5.3 lbs. The time to complete this descending flight segment is

$$t_{4\text{–}5} = \frac{1}{c}\left(\frac{C_L}{\gamma C_L + C_D}\right) \ln \frac{W_4}{W_5} = 271.1\,\text{s}.$$

The air speed at way point 5 on this descending flight segment is given by (15.15) as

$$V_5^- = \sqrt{\frac{2W_5}{\rho_5 S C_L}} = 341.3\,\text{ft/s}.$$

The value of the thrust at way point 5 is given by (15.19) as

$$T_5^- = \left(\gamma + \frac{C_D}{C_L}\right) W_5 = 101.9\,\text{lbs}.$$

Equations (15.4) and (15.10), together with the value of the lift coefficient, are used to determine the values of the the angle of attack and the elevator deflection at way point 5 as:

$$\alpha_5^- = 4.34^\circ,\ \delta_{e_5}^- = 2.55^\circ.$$

Since the lift coefficient is constant throughout this climbing flight segment, the angle of attack and the elevator deflection do not change throughout this flight segment.

The thrust must be adjusted by the pilot throughout the descending flight segment to compensate for the slow decrease in the aircraft weight as fuel is consumed by the jet engine and the slow increase in the air density as the aircraft descends. The value of the thrust at any point on the descending flight segment between way point 4 and way point 5 can be computed using linear interpolation between thrust values at way point 4 and way point 5.

It is easy to check that all flight constraints are satisfied for this steady descending flight segment.

9.3 Summary of the Fuel Optimal Flight Plan

The following table describes the cumulative fuel burned by the jet engine from take-off to landing:

Way point	Flight segment	Cumulative fuel (lbs)
1	–	0
2	Climb	341.2
3	Level turn	412.1
4	Level cruise	858.3
5	Descent	863.6

The following table describes the cumulative time of flight from take-off to landing:

Way point	Flight segment	Cumulative flight time (s)
1	–	0
2	Climb	220.1
3	Level turn	272.8
4	Cruise	781.0
5	Descent	1052.1

Hence, this fuel optimal flight plan describes the four steady flight segments of the business jet aircraft as it flies from way point 1 to way point 5. The minimum fuel to complete this flight mission (at least with the selected flight path) is 863.6 lbs and the total required time of flight is 1052.1 s = 17.5 min.

9.4 Comments on the Case Study Results

In this particular optimal flight planning case study, it is interesting to observe that the stall constraints, the thrust constraints, and the wing loading constraints are never active. This is due to the particular formulation of the optimal flight planning problem, namely that the mission way point specifications can be achieved without requiring an aggressive flight maneuver and that the fuel optimality objective tends to lead to balanced flight conditions where none of the flight physics constraints are active.

For purpose of comparison, the time optimal flight planning problem could be solved following the hierarchical approach suggested above. That is, the same way points might be used, but the flight conditions on each flight segment would be selected to be time optimal in the sense that flight on that segment is completed as rapidly as possible. The time optimality objective is likely to lead to steady flight segments for which the thrust constraints are active. Selection of different flight paths between way points or selection of different way points may lead to flight segments where the stall constraint and/or the wing loading constraint are active. In any case, the use of the hierarchical decomposition approach and the use of steady flight analysis would be similar in outline to the development of the fuel optimal flight case study just presented.

9.5 Questions About the Case Study

The case study demonstrates one reasonable solution for the fuel optimal flight plan for the business jet aircraft. There are many ways in which the flight planning problem might be modified. The following questions are posed to encourage thought about

the implications of some possible modifications. You should be able to answer these questions without extensive computation.

- In the steady climbing flight segment after take off, the flight path angle might be selected either slightly larger (in which case the horizontal distance traveled is less) or slightly smaller (in which case the horizontal distance traveled is greater) than 1.833°. What are possible advantages and disadvantages in selecting a slightly larger value of the flight path angle in terms of the total fuel required to complete the mission? What are possible advantages and disadvantages in selecting a slightly smaller value of the flight path angle in terms of the total fuel required to complete the mission.
- Is it possible to combine the climbing flight phase with the turning flight phase? What are possible advantages and disadvantages of this strategy?
- In the steady descending flight segment before landing, the flight path angle might be selected either slightly larger (in which case the horizontal distance traveled in the cruise phase is greater) or slightly smaller (in which case the horizontal distance traveled in cruise phase is less) than -3.053°. What are possible advantages and disadvantages in selecting a slightly larger value of the flight path angle in terms of the total fuel required to complete the mission? What are possible advantages and disadvantages in selecting a slightly smaller value of the flight path angle in terms of the total fuel required to complete the mission.
- Construct a heuristic flight path from take off location to landing location consisting of a sequence of helical flight segments so that the aircraft takes off from the take off location but in a due West direction and lands at the landing location in a due West direction. Can this be done by making a right hand turn of radius 5000 ft followed by a left hand turn of radius 5000 ft? Show that such tight turns cannot satisfy the flight physics constraints. Can this be done by making a right hand turn of radius 20,000 ft followed by a right hand turn of radius 10,000 ft? Construct a sequence of helical flight segments and way points using this heuristic that you think might be feasible.

10 Conclusions

An important class of aircraft flight optimization problems, including mission constraints and aircraft flight constraints, has been introduced. A hierarchical decomposition approach has been suggested that partitions the aircraft flight into a sequence of helical flight segments. Based on flight physics, flight conditions that achieve optimal or near optimal flight for each steady flight segment are developed. This leads to a simplified aircraft flight optimization problem that makes use of both heuristics and formal nonlinear programming methods to obtain efficient, effective and realistic flight planning solutions.

The hierarchical optimization approach, as described, completes the selection of way points and specification of helical flight segments first. Thereafter, it identifies the optimal steady flight conditions on each flight segment. This approach is suitable for

many optimal flight planning problems, but it is also possible to introduce a sequential feature into the decomposition. That is, the first few way points can be selected and the optimal flight conditions on those segments determined. This information can then be used to select one or more succeeding way points and to determine the optimal flight conditions for those flight segments. This sequential decomposition approach introduces extra flexibility that can be useful in some optimal planning problems.

It is possible to formulate more complex versions of the optimal flight planning problem that avoid some of the approximations inherent in the hierarchical decomposition approach. In particular, the aircraft flight optimization problem can be formulated as an optimal control problem that includes the flight dynamics, rather than assuming steady flight between way points. Such optimal control problems, where the optimal flight conditions are determined at each time instant throughout the flight duration, are infinite dimensional, nonconvex optimization problems and require the use of more sophisticated theoretical and computational tools, see [2, 3, 7]. Numerical optimization software tools in the DIDO and the Optimization Toolbox are available for such optimal control problems [4, 5]. Although it may be possible to solve such challenging optimal flight control problems, such optimal solutions may correspond to flight conditions that are not satisfactory to the pilot or flight operators.

References

1. Anderson, J. D. (2000). *Introduction to flight* (4th edn.) Boston: McGraw-Hill.
2. Betts, J. (2001). *Practical methods for optimal control using nonlinear programming, advances in design and control*. Philadelphia: SIAM.
3. Bryson, A., & Ho, Y. C. (1975). *Applied optimal control*. Washington, DC: Hemisphere.
4. DIDO optimal control software. Mathworks. http://www.mathworks.com/products/connections/product_detail/product_61633.html.
5. MATLAB optimization toolbox. Mathworks. http://www.mathworks.com/help/optim/.
6. McClamroch, N. H. (2011). *Steady aircraft flight and performance*. Princeton: Princeton University Press.
7. Ross, I. M., & D'Souza, C. N. (2005). A hybrid optimal control framework for mission planning. *Journal of Guidance, Control and Dynamics, 28*(4), 686–697.

Chapter 16
Freight Transport by Rail

Katta G. Murty, Bodhibrata Nag and Omkar D. Palsule-Desai

1 History of Freight Shipping by Rail in the USA

Following the development of railways in England, private enterprise in USA started designing and constructing rail-tracks and building the US railroads in the late 1820s. By late 1860s transcontinental railroads were operating across North America; this was one of the crowning achievements of the Presidency of Abraham Lincoln. Today most rail transportation in USA consists of freight shipments. US railroads play a major role in the nation's freight shipping, annually moving more than 40 % of the nation's freight; and thus play an important role in the US economy. They haul 40 million rail-carloads (averaging 63 t each) and generate over US$ 80 billion in freight revenue. They carry 1.5 trillion ton-miles, operating 140,490 route miles on standard gauge.

When considered in terms of ton-miles hauled/unit energy consumed, rail transport is more efficient than other means of transportation. However shipment by rail is not as flexible as by highway, which has resulted in much freight being hauled by truck. A freight (or goods) train is a group of freight cars (also called goods wagons) hauled by one or more locomotives on a railway, transporting cargo all or part of the way from the shipper to the intended destination. A single freight car might be

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_16) contains supplementary material, which is available to authorized users.

K. G. Murty (✉)
Department of Industrial and Operations Engineering,
University of Michigan, Ann Arbor, MI 48109-2117, USA
e-mail: murty@umich.edu

B. Nag
Operations Management Group, Indian Institute of Management Calcutta,
Calcutta, West Bengal, India
e-mail: bnag@iimcal.ac.in

O. D. Palsule-Desai
Indian Institute of Management, Indore, India
e-mail: omkardpd@iimidr.ac.in

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_16, © Springer Science+Business Media New York 2015

reclassified or switched in several yards, and hauled by several trains before reaching its final destination.

2 The Train Design Problem

While the freight railroad industry has been in existence for about 200 years, the procedure used for combining freight rail-cars into *blocks* based on their destination yards (also referred to as *nodes* in this chapter) and other attributes, and then assembling blocks into trains has essentially remained the same. Based on customer orders on hand for transporting goods wagons; and the origin, destination, and other attributes like physical dimensions, freight type etc., the railway generates a trip plan detailing the movement of each rail-car from its origin yard to its specified destination yard. The main components of the trip plan are: (a) how to combine the rail-cars at each rail yard into groups called blocks, this is known as the *block design problem*, (b) designing trains to operate; identifying the origin, destination, route for each individual train, and along the route what blocks to pick up, drop off at different yards along the route; this is known as the "Block-to-train assignment (BTA) and train routing" problem, and also referred to as the *Train Design Problem*.

In this chapter we discuss this train design problem only, assuming that blocks have already been formed at each of the yards; and the data on the origin, destination, number of rail-cars, length, tonnage of each block to be transported is given. This is a difficult combinatorial optimization problem encountered in the railroad industry.

This case study is a simplified real-life problem constructed and set-up under the *2011 RAS Problem Solving Competition* by RAS of INFORMS. The statement of the problem and all the data sets in it can be obtained from the following weblink: https://www.informs.org/Community/RAS/Problem-Solving-Competition/2011-RAS-Problem-Solving-Competition

3 Railroad Industry Terminology

Before describing the train design problem and our proposed algorithm to obtain the solution, we explain below the key terminology adopted by the railroad industry.

- **Rail-car:** In freight railroad industry, a rail-car (also referred to as *railroad car*, or *goods wagon*, or just *car* or *wagon*) is the basic container on wheels used to transport goods between locations. There are many different types of rail-cars used in the industry, but for simplicity this case study considers a problem dealing with only one type of rail-cars. In operation, rail-cars are loaded with freight, coupled with other rail-cars to form a train, which is then hauled by locomotives between required rail yards.
- **Railroad Network:** It is the network representation in which the nodes are the various rail yards, and the edges are the various rail links with each link connecting a pair of yards. Each link can be used by trains in both directions. We denote this

network by $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes, and $\mathcal{A}$ is the set of edges. Clearly, G is a connected network.

- **Block:** It is a set of rail-cars formed at a yard as a linked sequence to be moved as a single group from one yard to another. The original origins or destinations of individual cars in a block may be different. Blocks are assembled at rail yards based on the final destinations of rail-cars there and other compatible attributes.
- **Blocking:** It is the process of combining a set of cars at a yard into a block; also called *classification*, or *marshalling*. In operations, a block which is formed at a yard may be separated into individual cars at a subsequent yard and again combined with different cars into a different block. For an individual rail-car, classification into different blocks can happen multiple times at a sequence of intermediate yards before a car reaches its original destination.
- **Block Swap:** This operation transfers a block between trains without reclassification of cars in it at yards. So, in this operation the entire block remains coupled together as a sequence and is moved en masse from one train to another. This operation allows a block to reach its destination using more than one train.
- **Crew Segment:** Typically trains are driven by crew members between a fixed pair of yards, called a *crew district* or *crew segment*. Freight transit in North America typically crosses multiple crew segments, and thus uses multiple crews as it traverses its route. Thus a crew segment is a path in the railroad network between two nodes on which a single crew can operate a train.
  Crews are typically domiciled at a specific location at one end of the crew segment. The crew segments are designed such that it is a full day work to get from one end to the other of a crew segment. Normally crews work from their *home terminal* where they are domiciled, to the *away terminal* one day, and then return the next day.
  The pair of nodes which define a segment need not be directly connected by an arc between them; rather the segment may contain a number of nodes that are directly connected by arcs existing in the network.
- **Trip Plan:** It is the itinerary that a rail-car will follow as it moves from its original origin to its original destination.
- **Original Origin, Destination; Current Origin, Destination:** The original origin of a rail-car is the yard where it begins its journey, i.e., the yard where the customer packs the goods in it and delivers it to the railroad for its dispatch to its original or final destination. A rail-car may travel by several trains on its way to the original destination. While it is traveling on one of these trains, the current origin, destination of this rail-car are the yards where this train picks it up, drops it off respectively.
- **Train Imbalance Instances:** When the number of trains originating at a yard in a solution is not equal to the number of trains terminating at that yard, their difference is defined as the number of train imbalance instances in that solution at that yard. The total number of train imbalance instances in a solution is the sum of train imbalance instances in that solution at the various yards. In the problem we are considering, there is a penalty incurred for each train imbalance instance in the solution proposed.
- **Missed Car Instance:** In a solution for the problem, a rail-car is said to be *missed*, if it is not dropped off at its original destination yard during the period under consideration. The total number of missed car instances in a solution is the number of rail-cars missed in it during this period.

4 Operational Constraints to be Satisfied in the Train Design Problem

Each train design problem involves a set of constraints with respect to assigning blocks and crews to trains and assigning trains to network arcs. These constrains are typically derived from infrastructural and operational limitations and/or specifications. The constraints that need to be satisfied in the train design problem that we consider in this chapter are as follows.

- **Maximum Number of Blocks per Train:** This is an upper bound on the number of blocks a train can carry. Assigning too many blocks to a train may result in an excessive number of stops for the train, which is discouraged for operational efficiency. So this upper bound is an indirect way of limiting the number of stops for each train.
- **Block Swaps per Block:** In practice it has been observed that swapping a block from one train to another too many times, consumes worker time and other resources. That is why this constraint puts an upper bound on the number of times a block can be swapped between trains.
- **Work Events per Train:** A work event is defined as a train stop at an intermediate yard (i.e., not the train's origin or termination yard) to pickup or drop off some blocks, or to both pickup and drop off some blocks. Work events consume labor and contribute to train delays (to the cars, locomotives, crew of the train). Hence this constraint imposes an upper bound on the number of work events per train.
- **Upper Bounds on Length, Tonnage of Trains, and Number of Trains Traversing Edges in the Railroad Network:** Depending on the track attributes and the geographical location of an edge in the railroad network, upper bounds are specified on the length, tonnage, and the number of different trains traversing it in either direction. The upper bound on the number of trains passing through an edge is to avoid congestion on the edge.
- **Origin, Destination Yards of Train Must be Crew Originating, Terminating Yards:** Since every train needs crew on each crew segment, all trains must originate at the start of a crew segment; even if this requires that they have to move part of the way carrying no rail-cars.
  Trains can travel across multiple crew segments. When the termination yard of a train is not the end yard of a crew segment, it is an instance of *crew imbalance*, and additional expenses are incurred for repositioning the crews using an over-the-road taxi service.

5 The Objective Function to be Minimized in the Train Design Problem

The objective function is minimizing the sum of eight different cost components listed below; all components are measured in US Dollars (USD). All cost coefficients given are also in USD.

- **Train Start Cost:** Starting each train incurs a certain cost. This cost term is equal to the number of different trains in the solution multiplied by the train start cost coefficient.
- **Train Travel Cost:** This cost term is equal to the total travel miles of all trains in the solution multiplied by the train travel cost per mile.
- **Freight Car Travel Cost:** This cost term is equal to the total freight car travel miles in the solution multiplied by the freight car travel cost per mile.
- **Work Event Cost:** This cost term is equal to the number of train stops at intermediate yards of the train route (excluding origin and destination yards of the train route) multiplied by the cost per work event.
- **Block Swap Cost:** This cost term is equal to the total number of block swaps in the solution multiplied by the cost per block swap.
- **Crew Imbalance Cost:** This cost term is equal to the total number of crew imbalance instances in the solution multiplied by the specified penalty per crew imbalance instance.
- **Train Imbalance Cost:** This cost term is equal to the total number of train imbalance instances multiplied by the specified penalty per train imbalance instance.
- **Missed Cars Cost:** This cost term is equal to the total number of missed cars in the solution multiplied by the cost per rail-car missed.

6 An Algorithm for the Train Design Problem

The complexity of the train design problem has been well-acknowledged in both practice and academia, see [1–5, 7, 10]. Researchers provide a number of approaches to model and solve this problem (see, e.g., [6], [8]). These approaches typically provide the 0–1 or integer programming (IP) based models that require relatively complex and highly sophisticated solution techniques such as multilevel programming, large scale neighborhood search, etc. While the solutions derived using these models have been highly efficient in a variety of the contexts, if not the optimal in most of the cases, practitioners have been reluctant to using the softwares developed incorporating the IP based models on a routine basis. The inhibitions against using modern softwares primarily come from the level of sophistication required from the user and the complexities involved in customizing the software models from time to time based on specific needs. Moreover, in practice the data keeps on changing on a regular basis. Thereby, it is difficult for practitioners not only to accommodate such unexpected changes in the commercially available softwares, but also to interpret the improvised solutions obtained.

In order to facilitate practitioners in addressing their concerns in solving the typical train design problem using simpler tools and techniques, in this chapter we provide an algorithm to obtain a solution that satisfies all constraints mentioned above. In our algorithm, we first develop train routes in an iterative manner using paths formed in the minimum spanning tree; this tree is obtained using the well-known Dijkstra's algorithm. Our algorithm demonstrates ways by which the Dijkstra's algorithm can be improvised to incorporate operational constraints such as those mentioned in our

problem context. We choose a particular train route in a *greedy* manner, and later we *build* a train on the train route by assigning blocks to it in a *greedy* manner. Likewise, the simplicity in our algorithm essentially comes from the use of the existing tools and techniques that are easy to implement and easy to modify as per needs; the solution obtained is also easy to interpret that provides foundation for further improvements. (More on this to follow in the later sections in this chapter.) To test our algorithm we use Data Set 1 provided in the 2011 RAS Problem Solving Competition.

From the data set, we notice that the major cost components among all the ones specified in the problem are: freight car traveling cost and train traveling cost (Detailed discussion follows in Sect. 7). Hence, our algorithm for the original problem—of minimizing the sum total of the eight cost components mentioned above—is based on minimizing the sum of the two cost components—freight car traveling cost and train traveling cost. With this modified objective, we ensure that the operational constraints in the train design problem are satisfied at various instances that are appropriately chosen in our algorithm. What follows in this section is a detailed description of the algorithm.

Step I. Determining Active Yards and Exogenous Flows at all Yards

Consider a particular iteration of the algorithm. Define a *active yard* as a rail station that has at least one block of rail-cars of goods to be sent out to other yards, or to be received at least one block of rail-cars of goods from all other yards in the network. Let n denote the total number of active yards in this iteration; each active yard is a node in $\mathcal{N}$.

For each active node i let $v_i = v_i^- + v_i^+$. Here, v_i^- is the exogenous flow at node i that is equal to the number of rail-cars to be transported to node i from all other nodes in the network, and v_i^+ is the number of rail-cars to be sent out to other nodes from node i.

Step II. Identifying a Path for a Round-Trip Train Route

We select an active node i corresponding to the maximum value of v_i among $i \in \mathcal{N}$. We ensure that the selected node has at least one block of rail-cars to be sent out from as far as possible, and also ensure that the node is at the beginning of a crew segment. While the former criterion ensures that the train does not travel any distance without carrying any blocks of rail-cars on the *forward* path, the later criterion satisfies the crew segment related constraint. We refer to the selected node as the *center* for this iteration. Suppose that the center is node 1 in this iteration.

We determine the shortest path tree in G rooted at node 1—referred to as T_1. It consists of shortest paths between node 1 and all other nodes in the network.

This minimum spanning tree rooted at node 1 can be obtained using the Dijkstra's algorithm (see, Sect. 4.2 in [9]). (In the remainder of this chapter, we use the notations as described in Chap. 1 in this [9].)

Let j_t, $t = 1$ to r, be the *end nodes* in T_1. For each $t = 1$ to r, let $\mathcal{P}_t$ be the *predecessor path* from j_t to the root node 1 in T_1. It is the shortest path between nodes 1 and j_t. In this iteration, we select one of these $\mathcal{P}_t$ to be used in the forward and reverse directions for the train as the round trip route from node 1 to j_t and back to the root node. For this selection, we use the following simple procedure.

For each $t = 1, \ldots, r$ calculate the total number of rail-cars to send from the root node 1 to all other nodes on $\mathcal{P}_t$ plus the total number of rail-cars to send to node 1 from all other nodes on $\mathcal{P}_t$. We select the $\mathcal{P}_t$ corresponding to the maximum value of this quantity to process in this iteration. We ensure that the selected $\mathcal{P}_t$ is such that node j_t is on the end of a crew segment. From the statement of the problem, we know that the final node of any crew segment is also the beginning node for another crew segment. We also satisfy the condition that there is at least one block to carry in the reverse direction (i.e., in the direction from node j_t to 1) as far as possible. If there is no block to carry in the reverse direction from node j_t to 1 for all $t = 1, \ldots, r$, then we select $\mathcal{P}_t$ that has the maximum number of rail-cars to be sent on the forward path. In this case, we do not allow the train to travel in the reverse direction from node j_1 to node 1 creating a train imbalance instance. Suppose that the selected train route in the iteration is $\mathcal{P}_{j_1}$. Now, we form a train, carrying blocks of rail-cars on the selected train route as described below.

Train Formation on the Forward Path: Processing $\mathcal{P}_{j_1}$ in the Shortest Path Tree T_1 in the Direction of Travel from Root Node 1 to Node j_1

For illustration, refer to Fig. 16.1. Consider a node a on $\mathcal{P}_{j_1}$ that has some blocks of rail-cars to be sent out to a node d in the spanning tree T_1. Node d is such that it is one of the *descendant nodes* (defined in page 25 in [9]) of node a, but it is not on $\mathcal{P}_{j_1}$. Moreover, the portion of the predecessor path of node d between nodes a and d has a common portion with $\mathcal{P}_{j_1}$ that begins with node a and ends with node b, where b is a node on $\mathcal{P}_{j_1}$ whose distance index (or depth label in the rooted tree $\mathcal{P}_{j_1}$) is greater than the depth label of node a. In this case the shortest path from node a to node d consists of two parts: one is the portion of the predecessor path of d in T_1 between nodes b to node d, and the second is the portion of $\mathcal{P}_{j_1}$ between nodes a and b.

So, for processing $\mathcal{P}_{j_1}$ in the forward direction, we consider node b as the *temporary* or *current destination* for all blocks of rail-cars to be sent from node a to node d on a train that we operate on this route. We refer to the blocks of such rail-cars as *eligible* blocks on the train route. The train on the selected train route carries all of the eligible blocks if this violates no capacity constraints, or as many of them as possible as permitted by the various capacity constraints. The eligible blocks that are carried by the train are then detached at yard b. We make their *new origin* as node

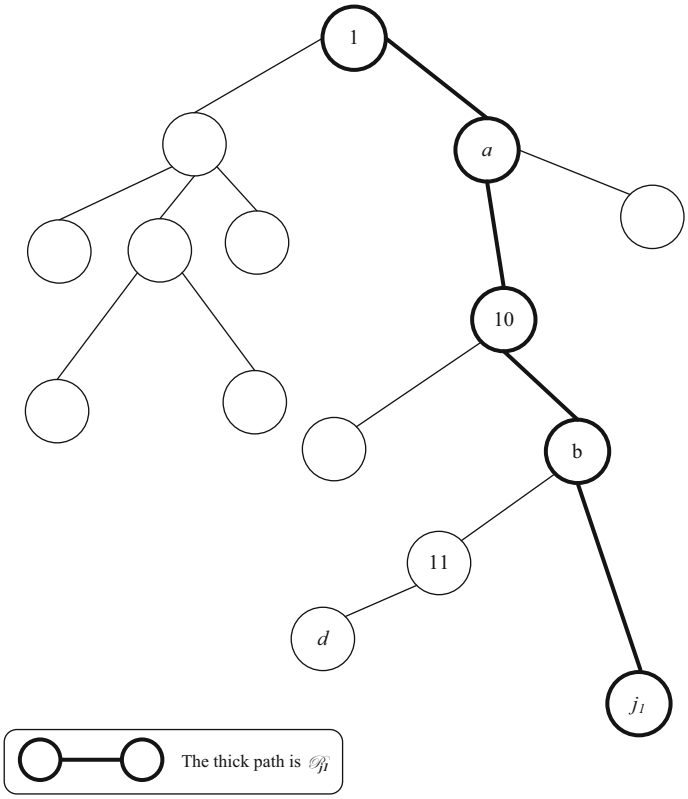


Fig. 16.1 An illustration for train formation on the train route selected: Minimum Spanning Tree rooted at node 1 and path $\mathcal{P}_{j_1}$

b, from which they have to be dispatched to their original destination, node d, in subsequent iterations. We follow this procedure for all nodes (such as node a) on the forward path of the train on the train route $\mathcal{P}_{j_1}$ to determine blocks that are eligible to be assigned to the train. When this work is completed for all the nodes on this path, the processing for train formation begins; the procedure is Step III of the algorithm.

Train Formation on the Reverse Path: Processing $\mathcal{P}_{j_1}$ in the Shortest Path Tree T_{j_1} in the Direction of Travel from Root Node j_1 to Node 1

To form the train on the reverse path by assigning blocks of rail-cars, we apply Dijkstra's method to determine the shortest path tree rooted at node j_1, say T_{j_1}. It consists of the shortest paths between node j_1 and all other nodes in the network, with

$\mathcal{P}_{j_1}$ as the shortest path between nodes j_1 and node 1. To determine the minimum spanning tree rooted at node j_1, we adopt the Dijkstra's algorithm as described in [9] with the shortest path $\mathcal{P}_{j_1}$ already selected and X as the set of nodes on it. We continue the algorithm until all other nodes in $\mathcal{N}$ become the tree nodes.

Once the minimum spanning tree T_{j_1} is developed, we process the path $\mathcal{P}_{j_1}$ in the (reverse) direction of the train travel from root node j_1 to node 1 with the procedure that is similar to that in the forward direction from node 1 to node j_1. After this is completed, we go to the step of the train route formation described below.

Step III. Formation of a Train on the Train Route Selected

Now we form the train consisting of travel along the path $\mathcal{P}_{j_1}$ starting and ending at node 1 in the forward and reverse directions to transport the eligible blocks of rail-cars with origin and current destination nodes on this path to their current destinations as described in Step II above. If there are no bounds on the number of blocks of rail-cars a train can carry and also there are no network-arc related constraints, then this transport can be completed by a train going around this route once. Moreover, in view of the various constraints to be satisfied in the formation of trains, it is possible that not all blocks eligible to be carried on the train route (in both the forward and reverse directions) can be assigned to the train. Therefore, we adopt a greedy technique to select the blocks from the eligible ones to be assigned to the train in this iteration. In particular, we rank the eligible blocks in the decreasing order of their weight and length[1], and select blocks one by one such that the highest ordered blocks are chosen to be assigned to the train first. In the process, we ensure that the following operational constraints are satisfied: (i) maximum number of blocks per train, (ii) maximum number of block swaps per block, (iii) maximum number of work events per train, and (iv) upper bounds on length and tonnage of trains. We also ensure that the blocks of rail-cars assigned to the train are such that each of these *capacities* are consumed to the maximum extent possible. By prioritizing and selecting blocks in the train formation process as described here, we essentially reduce the possibility of creating *more* trains than necessary due to the various operational constraints acting on them. Moreover, to further reduce the train traveling cost, we allow the train to travel in the reverse direction from node j_1 to node 1 on the path $\mathcal{P}_{j_1}$ only if there is at least one block of rail-cars that is eligible to traverse in this direction. If there is no block on the return path to carry on the train, we terminate the further journey of the train in this direction; it creates not only a train imbalance instance, but also crew imbalance instances as all crew segments used on the forward path are not used in the reverse direction.

[1] As described the following section, in the given data set, the ratio of weight to length of a block of rail-cars is constant for all blocks and it is equal to 4/3. Therefore, the eligible blocks can be rank-ordered either on the basis of weight or length.

With the train on the selected train route is formed, all the remaining data in the problem is updated taking into account the blocks of rail-cars transported on both forward and reverse directions of the train trip. With updated data, the algorithm moves to the next iteration to handle the transport of the remaining blocks of rail-cars by using the same procedures as described above by going back to Step I. It may be noted that after completion of the train travel on both forward and reverse directions of the train trip, we also need to update the number times the arcs on the path $\mathcal{P}_{j_1}$ are traversed. If the arcs reach the bound of the number of times an arc can be traversed, then this arc is removed from the network so that it is not in the network in subsequent iterations. The Dijkstra's algorithm in Step I will develop the minimum spanning trees for the reduced network in subsequent iterations.

One can terminate the algorithm when the total number of remaining rail-cars to be dispatched from the nodes in the network becomes *small*. At that stage, we can also stop the algorithm and consider the remaining rail-cars as "missed rail-cars"; as long as the cost of treating them as "missed" is smaller than trying to dispatch them. Nevertheless, in order to examine the performance of our algorithm, in this chapter we allow all blocks of rail-cars to reach to the respective destination nodes. Figure 16.2 summarizes the greedy algorithm.

7 Description of the Data Set

Data Set 1 of the 2011 RAS Problem Solving Competition that we use to test our algorithm consists of 94 nodes and 134 arcs forming the rail network. There are 239 blocks that need to be transshipped on this network from the respective origin nodes to the destination nodes with the help of 154 crew segments available. From these 94 nodes, 72 nodes have at least one block of rail-cars to be sent out to any other node and 62 nodes have at least one block of rail-cars to receive from other nodes in the network. As some nodes have at least one block of rail-cars to be sent out and at least one block to be received from any other node, effectively only 79 nodes have at least one block of rail-cars either to be shipped out to or to be received from any other node in the network. Therefore, the remaining 15 nodes can be used only as transshipment nodes in the network, and likewise, they may become active in some of the iterations in our algorithm. The number of blocks originating from nodes varies from 1 to 16; similarly, the number of blocks terminating at nodes varies from 1 to 23. The absolute difference of the number of blocks originating and terminating at nodes varies from 0 to 10. The aggregate of absolute differences of the number of blocks originating and terminating at all nodes in the network is 116. It implies that if the blocks are transported by forming separate trains for each block, then the number of train imbalance instances will be equal to 116.

From the data set, we note that 17 nodes are *crew-inactive* such that they have neither any crew segment originating from them nor do they have any crew segment terminating at them. This implies that there can be no train that can either originate or terminate at any of these 17 crew-inactive nodes. From these 17 crew-inactive nodes,

```
start
{
    1. input data: nodes, arcs, block details, crew segment details, and cost parameters
    2. create matrices: (i) node-to-node adjacency, (ii) node-to-node distance, and
                        (iii) node-to-node crew segments
    3. sort the blocks in the decreasing order of weight (or length)
    4. run the iteration algorithm for train formation:
       do
       {
            a. determine the active nodes
            b. select the center
            c. using the Dijkstra's algorithm, determine the minimum spanning tree
               rooted at the center
            d. determine the end nodes in the minimum spanning tree
            e. select an end node to obtain the train route in the iteration
            f. determine eligible blocks on both forward and reverse paths of the train
            g. build the train on the train route chosen as follows:
               do
               {
                    i. select a block with the highest rank that is eligible to be assigned to
                       the train
                    ii. if the block is feasible on the train, then
                        {
                            I. change the new origin of the block as the current destination
                            II. adjust the train capacity constraints
                            III. adjust the capacity constraints for arcs on the train routes
                        }
                        else
                        {
                            I. ignore the block
                        }
                        end if
                    iii. select the next eligible block with the highest rank
               }
               do while no more eligible blocks can be assigned to the train
            h. determine if the train is traversed in the reverse direction
            i. adjust the number of times the arcs on the train route are traversed
            j. determine if any arc has reached the maximum number of trains that can traverse:
               if any arc has reached the maximum number of trains that can traverse, then
               {
                    i. remove the arc from the network
               }
               end if
            k. determine if all blocks have reached the respective final destinations:
               if all blocks have not reached the respective final destinations, then
               {
                    i. conclude that one more iteration is required
               }
               end if
       }
       do while more iteations are required
    5. print output
}
end
```

Fig. 16.2 Pseudo code for the greedy algorithm proposed for the train design problem

8 nodes do not have any block of rail-cars to be sent out to or to be received from any other node in the network. It implies that for the blocks originating/terminating at the remaining 9 crew-inactive nodes will require trains to travel without any blocks upto nodes beyond their originating/terminating nodes.

A set of crew-members operate over a particular segment comprising a set of arc(s) from an origin to a destination—no other set of crew operates on that segment. For example, a set of crew will operate all trains running from node 72 to node 74 and from node 74 to node 72. We note that 48 crew segments do not have arcs directly connecting the pair of nodes that define these crew segments, but of course there will be paths (consisting of multiple arcs) connecting these nodes in these cases. Moreover, crew segment distances vary from 19 to 290 miles (considering the shortest distance between nodes defining the crew segment as given in the data set). There might exist a few nodes in the network which do not belong to any of the pair of nodes defining the crew segments. There might also exist arcs, which do not belong to any crew segment; trains cannot use such arcs, since there are no crews for servicing trains running on those arcs. For the particular data set, there are no arcs which do not belong to a crew segment.

Among the 239 blocks of rail-cars that are required to be transported over the network, the lightest block has 4 cars; it weighs 320 t and has a length of 240 ft. The heaviest block has 77 cars; it weighs 6144 t and has a length of 4608 ft. Nevertheless, the ratio of weight to length for each block of rail-cars is constant and it is equal to 4/3. As mentioned earlier, the constant ratio of weight to length of each block of rail-cars will provide the identical rank-ordering of the blocks on the basis of their characteristics—weight and length. Recall that such rank-ordering of blocks is required in Step III (Train Route Formation) of our greedy algorithm. From the given data, we observe that the shortest distance over which a block travels varies from 65 to 1071 miles implying that the freight car travel cost will not be constant for all blocks of rail-cars.

The lengths of the 134 arcs in the network vary from 19 to 265.3 miles. Similarly, all arcs have different bounds for length and weight of a train that can traverse on it. The maximum train length capacity of an arc varies from 4900 to 10,000 ft. The maximum tons carrying capacity varies from 3500 to 15,000 t. While the weight to length ratio for all blocks of rail-cars is constant, the ratio of weight to length of a train that can traverse an arc is significantly different across the arcs. Furthermore, 131 arcs have this ratio more than one, 2 arcs have it less than one, and for the remaining 31 arcs the ratio is equal to one. It implies that the characteristic of a block—weight and/or length—that makes it infeasible for the block to be assigned to a train due to the capacity constraints of an arc on which the train traverses in a particular train route will alter between weight and length. Moreover, no arc can be used for more than six trains to traverse on.

Assuming that there are no constraints on the train and arc capacities, if all blocks in the network are transported on the shortest paths between the respective origin and destination nodes, given the car travel cost of $ 0.75 per mile, the total freight car traveling cost is equal to $ 1,544,246. This is a lower bound for the total car traveling cost in our algorithm as various capacity and operational constraints in

our train design problem may require blocks of rails-cars to be transported from the respective origin node to the destination node on path other than the shortest path between the nodes. It costs $ 400 to start a new train from any of the nodes in the network. Hence, instead of running two trains from A to B and B to C, the total cost can be reduced by running a single train from A to C. If a separate train is created for each block in the network, then the train start cost will be equal to (239 × 400=) $ 95,600. However, we note that 69 pairs of blocks are such that the origin node of one member of pair is the destination node of the other member of the pair, and vice versa. For example, block 26 is sent from node 64 to node 62 and block 29 is sent from node 62 to node 64. Therefore, a single train could be formed which runs from node 64 to node 62 (carrying block 26) and then from node 62 to node 64 (carrying block 29). There are also a few instances where more than one block can be carried in either direction on a single train. It is found that 129 such blocks can be combined to run by 61 trains. Likewise, the maximum of 171 trains will be required for carrying all 239 blocks, and the corresponding train start cost reduces to $ 68,400. This is an upper bound for the train start cost in our algorithm as a train picks-up and drops-off multiple blocks at various locations on its route that reduces the number of trains required. It costs $ 10 per mile of travel of a train. Assuming that there are no constraints on the train and arc capacities, if all blocks are transported on the shortest paths between their origin and destination nodes using separate trains, then the total train travel cost is equal to $ 681,462. This is an upper bound for the total train travel cost in our algorithm as the trains pick-up and drop-off blocks at various nodes on the train route between the respective origin and destination nodes that reduces overall distance traveled by all trains. We observe that all the 72 block-originating-nodes belong to a pair of nodes defining a particular crew segment. Similarly, all the 62 block-terminating-nodes belong to a pair of nodes defining a particular crew segment. If we ignore the train and arc capacity constraints, 94 crew segments are used by routing the blocks through the shortest paths; it results in 167 crew imbalance instances. This an upper bound for the number of crew imbalance instances in our algorithm because a train traveling on the forward path also travels on the return path as far as possible that reduces the number of crew imbalance instances. The corresponding crew imbalance cost is equal to $ 100,200 (The penalty per crew imbalance is $ 600.). It costs $ 1000 per train imbalance; thereby an upper bound for the train imbalance cost in our algorithm is $ 116,000.

Blocks may be transported from origin to destination on a number of trains. In that case, a block will be dropped-off at intermediate node(s) by one train, and it will be picked up by another train passing through the node(s) on subsequent train routes. Each time, a block is detached from a train and attached to another train is known as a block swap. The number of block swaps for any block should be less than three. Each block swap has a cost associated with it, depending on the node where the block swap occurs. Block swap costs vary from $ 30 to $ 100 depending on the node. If separate trains are created for each block as described above, the total block swap cost and the total work event cost will be zero.

It is possible that train- and arc-capacity constraints do not allow transportation of some of the blocks from their origin to destination nodes. The cost of $ 5000 per missed car is incurred in that case. As mentioned earlier, in our algorithm we assume that all blocks are required to be shipped to the respective destination nodes implying that the total missed car cost is equal to zero. Likewise, by ignoring all train and arc capacity constraint in the train design problem and transporting all blocks on the shortest paths, a reasonable estimate of the the total cost is equal to $ 2,510,308. It may be noted that the aggregate of the freight car traveling cost and the train traveling cost ($ 2,225,708) constitutes approximately 89 % of the estimated total cost. Thereby, as mentioned earlier in this chapter we develop an algorithm for the original problem of minimizing the sum total of the eight cost components by minimizing the sum of the two largest cost components—freight car traveling cost and train traveling cost.

8 Solution for the Train Design Problem Using the Greedy Algorithm

We use Code::Blocks[2], version 12.11 rev 8629, to write a C++ programming based code for our greedy algorithm.

Using our greedy algorithm for the given data set, we observe that the solution to the train design problem yields 60 distinct train routes and 60 new trains traversing these routes to transport all of the 239 blocks of rail-cars from the respective origin nodes to the final destination nodes. While the longest train route involves 19 nodes, the shortest train route is between two directly connected nodes. Similarly, the maximum number of crew segments forming a train route is observed to be 14, and the minimum number is one. The train routes formed are such that 12 of the arcs are never traversed by any of the trains; the constraint of the maximum of six trains on any arc in the network is binding for 21 arcs. We observe that four out of the 94 nodes in the network are inactive; neither are there any trains starting and/or terminating at these nodes nor are these used for transshipment of blocks of rail-cars on any of the train routes designed. Moreover, the maximum number of trains that any node accommodates is 21 in the final solution.

From the 60 trains that follow the forward direction—from the root node to an end node—on the respective train routes, 41 trains travel in the reverse direction; it results in 19 train imbalance instances. Likewise, 101 train trips—instead of 171 train trips in the special case described above—are required to transport all blocks of rails-cars to the respective final destination nodes. Four train trips on three separate trains are such that the trains travel with the full capacity constraint of eight blocks on it. Similarly, 12 train trips on nine separate trains are such that the trains require

[2] Code::Blocks is an open source and a full-featured cross-platform IDE (Integrated Development Environment). Weblink: http://www.codeblocks.org.

the maximum work event stops allowed—four—on the respective train routes; 23 train trips involve direct delivery of the blocks assigned to them from the train origin node to the termination node. The remaining 78 train trips require overall 151 work event stops at various nodes on the respective train routes. In parallel with the train imbalances, we observe that there are 71 crew imbalance instances in the solution provided by the algorithm.

Out of 239 blocks of rail-cars to be transported in the network, 189 blocks of rail-cars are shipped directly to the final destination nodes from their origin nodes without swapping between trains at intermediate nodes. Only 50 blocks of rail-cars are transported to the respective destination nodes using more than one train. Moreover, these 50 blocks are swapped 57 times at various nodes. None of the blocks requires the maximum number of block swaps allowed—three—for any block.

The cost implications in the solution for the train design problem provided by the greedy algorithm are reported in Table 16.1. We observe that the total cost in the train design problem considering the various constraints described above reduces by approximately 6.5 % when compared with the total cost in the scenario of separate trains for each block with no capacity constraints for trains and arcs. The reduction in the total cost is essentially the result of combining multiple blocks on various trains within the capacity constraints applicable in the train design problem. In particular, the total cost reduces as the number of trains started are less, and hence, the number of train imbalance and crew imbalance instances are also less. It reduces the aggregate distance traveled in unproductive trips of trains on the same route without entirely consuming the capacities (in terms of maximum allowable length and weight of a train) by carrying more blocks; which further reduces the train travel cost, and hence, the total cost in the train design problem is also lower. While the train travel cost reduces by more than 9 %; it increases both the block swap cost and the work event cost.

From Table 16.1, we note that the freight travel cost increases (by approximately 1 %); the increase is due to the constraint of the maximum of six trains can traverse an arc in any direction. Thereby, some of the blocks of rail-cars are traversed *back* on subsequent trains yielding unproductive travel. Clearly, this scenario creates an opportunity to improve the algorithm further to derive additional cost benefits. Moreover, even though our greedy algorithm reduces the train travel cost, we believe that this cost component can be reduced further by designing train routes and forming trains in a *better* manner. For instance, by altering the basis for selecting the root node in each iteration or by prioritizing the blocks on the basis freight car distance or value, on the basis of missed car cost, etc., may improve our results. Nevertheless, our greedy algorithm provides an important platform to obtain solutions that are easy to interpret and simple to implement for various real-world train design problems without requiring practitioners to resolve the model at every instance of any changes in data and/or operational constraints. Moreover, our algorithm can be converted into a DSS (Decision Support System) for implementation in real-world

Table 16.1 Cost implications for the train design problem

Cost component	Reasonable bounds[a]	Considering various operational constraints[b]
Freight car travel cost	$ 1,544,246[c]	$ 1,556,919
Train travel cost	$ 681,462[d]	$ 648,700
Block swap cost	$ 0	$ 2030
Train start cost	$ 68,400[d]	$ 24,000
Work event cost	$ 0	$ 52,850
Train imbalance cost	$ 116,000[d]	$ 19,000
Crew imbalance cost	$ 100,200[d]	$ 42,600
Missed car cost	$ 0	$ 0
Total cost	$ 2,510,308	$ 2,346,099

[a]Bounds are determined by assuming that there are no constraints on the train and arc capacities and all blocks in the network are transported on the shortest paths between the respective origin and destination nodes by creating separate trains for each block. Even though this procedure gives the smallest possible freight car travel cost, it maximizes each of train start, train travel, train imbalance, and crew imbalance costs; thus leading to a very high total cost (sum of all the cost components) in this column

[b]Considering various constraints of the train and arc capacities as described in our problem context, the solution to the train design problem is obtained using our greedy algorithm proposed in this chapter

[c]Lower bound on the freight car travel cost

[d]Upper bound on the corresponding cost component

applications where complexity of solution technique and time to obtain *reasonable* solutions are quite critical[3].

9 An Illustration

For illustration purpose, in this section we describe the step-wise outcome of our algorithm for the first iteration.

After counting the total number of rail-cars originating from and terminating at each node in the network, we observe that node 89 handles the maximum transportation load of 893 rail-cars; from which 275 rail-cars are to be shipped out of node 89 and the remaining 618 rail-cars are to be shipped to this node. We select node 89 as the root node in this iteration as this node has at least one crew segment either originating from or terminating at it. We observe that there are overall 34 end nodes on the minimum spanning tree rooted at node 89. The maximum number of rail-cars to be transported on the shortest path from the root node to any of these end nods is for node 8. We select node 8 as the other node for the train in this iteration as this node has at least one crew segment either originating from or terminating at it. In this case, overall 189 rail-cars are to be transported on the shortest path between node 89

[3] The interested practitioners and researchers may contact the author Omkar D. Palsule-Desai at omkardpd@iimidr.ac.in to receive the software code developed for the greedy algorithm proposed in this chapter.

Table 16.2 Details of blocks transported on the train

Forward direction			Reverse direction		
Block number	Current origin node	Current destination node	Block number	Current origin node	Current destination node
74	33	39	63	75	89
88	89	17	173	8	79
89	89	33	183	79	75
92	89	79	190	72	75
93	89	39	214	53	72

and node 8; the shortest path between these two nodes (in sequence of node numbers) is $89-33-17-39-75-74-78-6-5-72-53-79-90-93-40-8$. The corresponding sequence of crew segments that can be used on this train route is $90-88-100-27-25-124-34$. From these 189 rail-cars to be transported on the train route, 41 rail-cars are to be transported in the forward direction of the train from node 89 to node 8, and the remaining 148 rail-cars are to be transported in the reverse direction of the train from node 8 to node 89. Therefore, in this iteration we allow the train to travel in the reverse direction from node 8 to node 89. On the forward path, the train stops at nodes 33, 17, 39, and 79 to pick up and drop off blocks. Similarly, on the return path its stops at nodes 79, 53, 72, and 75. A number of blocks are carried on the train in either forward or reverse direction from the *current* origin node to the *current* destination node; the details are as given in Table 16.2.

10 Other Approaches for the Train Design Problem

Approaches obtaining solutions to the train design problem vary in terms of cost and business constraints considered, and most of them decompose the problem into two or more subproblems that are solved in stages. The most common approach discussed generates promising train routes and crew-to-train assignments in the first stage. Dijkstra's shortest path algorithm, and 0–1 MIP (Mixed Integer Programming) models and a variety of algorithms for solving them are also employed in this stage. Then the next stage finds block-to-train assignments using again 0–1 MIP models and algorithms. Interested readers may refer to a few references given at the end of this chapter among [1–8, 10]. What follows below is a brief description of the solution techniques adopted by the four winning teams of the 2011 RAS Problem Solving Competition; the detailed reports can be obtained from the following weblink: https://www.informs.org/Community/RAS/Problem-Competition/2011-RAS-Problem-Solving-Competition

Team "Koppa"—Lozano, L., González, J.E., and Medaglia, A.L.—propose 0–1 MIP based model adopting a decomposition approach that obtains two subproblems: the first, referred to as *Train Design Problem*, determines the set of train routes, and the second, referred to as *Block-To-Train Assignment*, assigns blocks to the train routes. Their model is essentially based on the observation that the cost of missed

cars and block routes are considerably higher than all other costs. Thereby, they select the train routes that provide shortest paths for the most valuable blocks. The total cost in the problem is equal to $ 2,043,471.

Team "OR@UNIMI"—Colombo, F., Cordone, R., and Trubian, M.—propose a model adopting a row and column generation heuristic to solve the train design problem. In the master problem the main columns represent either trains or paths followed by a block. While the train columns are generated with a two level tabu search heuristic, the block-path columns are generated by solving a MIP net-flow model. At the end of the column generation process, the authors impose integrality constraints to the variables of the master problem, and solve the corresponding IP model. The total cost in the train design problem is equal to approximately $ 2,212,163.

Team "NCKU" —Wang, I.L., Lee, H.Y., and Liang, Y.T.—develop a mixed integer programming based model for the train design problem. The main premise of their solution technique is that the optimal solution can be easily obtained if the sets of block routes and train routes with certain desirable properties are already given. Thereby, they develop a heuristic technique based on the shortest path algorithms to obtain block routes, and propose another algorithm to obtain train routes. Using both block and train routes generated, they iteratively solve the reduced MIP to obtain the solution. The authors develop four different block and train generating mechanisms. The average total cost of the problem considering the four different mechanisms is $ 2,027,546.

Team "RAIL-OPT"—Jin, J.G., and Zhao, J.—propose a two-stage hierarchical approach to the train design problem. The first stage determines the routes for each train considering crew-to-train assignment constraints. In the second stage, the block-to-train assignment problem is solved. While they adopt a column generation technique to obtain promising train routes, the second stage problem is formulated and solved as a mixed integer programming problem. The total cost in the train design problem is equal to approximately $ 2,079,308.

11 Exercises

1. In this chapter we have provided an alternate mechanism based on an algorithm to obtain the solution to the train design problem as described in 2011 RAS Problem Solving Competition by RAS of INFORMS. While our algorithm provides a suitable solution to the problem, the algorithm can be easily improved upon to reduce the total cost further. As mentioned in this chapter, one may consider the following alternatives in our algorithm:
   - In view of relatively large train travel cost, select the root node in each iteration on the basis of train travel distance between the root node and all possible end nodes.
   - In view of relatively high cost of missed cars, select the root node on the basis of freight car distance or freight travel cost.

Table 16.3 Exercise 5

Arc capacity		15,000	11,000	11,000	14,000	14,000	10,750	10,750	11,750	9,000	9,000	13,000	
Terminal nodes for arc		89–32	32–17	17–39	39–75	75–74	74–78	78–6	6–5	5–72	72–53	53–79	
Serial number	Block Number												Block weight
1	61					√							1792
2	62					√							1216
3	64					√	√	√	√	√	√	√	1344
4	65					√	√	√	√	√			2560
5	66					√	√	√	√	√	√		960
6	69				√	√	√	√	√	√	√	√	768
7	71			√									832
8	74		√	√									1408
9	78							√	√	√	√	√	1024
10	88	√	√										1600
11	89	√											1408
12	93	√	√	√									1344
13	117									√	√	√	1088
14	118									√			1856
15	172										√	√	2752
16	191										√		832
17	192											√	1024
18	219					√							2240
19	221					√	√	√					2112
20	223					√	√	√	√	√	√	√	3072

- In view of the various capacity constraints on the arcs in the network, select the root node and the train path in each iteration on the basis of characteristics of the blocks, i.e., block length and weight.

2. In order to determine criticality of various operating constraints in the train design problem, one may examine the cost implications of alternate capacity levels separately. For instance,
   - How much will the total cost increase if the maximum number of trains permitted on any arc in the network is decreased to five? Similarly, how much will the total cost decrease if this number is increased to seven?
   - How much will the total cost increase (or decrease) if the maximum number of intermediate work events permitted per train is increased to five?
   - How much will the total cost increase (or decrease) if the maximum number of block swaps permitted per block is increased to four?
3. To analyze the output of our algorithm, one may also introduce additional constraints in the train design problem for operational simplicity. For instance, in order to evenly allocate the load of transshipment across the yards in the network, the number of trains a yard can accommodate can be restricted to fifteen. How much will the total cost increase (or decrease) in that case?
4. The reader may analyze the performance of our algorithm for the other data sets provided in the 2011 RAS Problem Solving Competition by RAS of INFORMS. The reader may contact one of the authors of this chapter, Omkar D. Palsule-Desai at omkardpd@iimidr.ac.in, to receive the C++ code developed for the greedy algorithm proposed here.
5. Let us take the 20 Blocks to be transported from node 89 to node 79, as shown in Table 16.3 with cells marked with $\surd$ indicating the arcs that can be traversed by the blocks. For example, block 66 (given in serial number 5 in the table) has a weight of 960 t (as indicated in the last column) and is required to travel from node 75 to node 53. The first row of the table also gives the restriction for each arc in terms of the weight of train allowed to traverse a particular arc; for example trains traversing arc from node 39 to node 75 cannot have weight exceeding 14,000 t. We have to form trains such every block is accommodated in a certain train for every arc that it is required to travel. Train formation using these 20 blocks has certain restrictions as described earlier. Formulate and solve a MIP model to form trains such that total cost (sum of all cost components) is minimized.

References

1. Assad, A. A. (1980). Models for rail transportation. *Transportation Research, 14A,* 205–220.
2. Carpara, A., Fischetti, M., & Toth, P. (2002). Modeling and solving the train timetabling problem. *Operations Research, 50,* 851–861.
3. Crainic, T. G., & Rousseau, J. M. (1986). Multicommodity, multimode Freight transportation: A general modeling and algorithmic framework for the service network design problem. *Transportation Research, 208,* 225–242.
4. Dorfman, M. J., & Medanic, J. (2004). Scheduling trains on a railway network using a discrete event model of railway traffic. *Transportation Research B, 38,* 81–98.

5. Gorman, M. F. (1998). The Freight railroad operating plan problem. *Annals of Operations research, 78,* 51–69.
6. Jha, K. C., Ahuja, R. K., & Sahin, G. (2008). New approaches for solving the block-to train assignment problem. *Networks, 51,* 48–62.
7. Keaton, M. H. (1989). Designing optimal raiload operating plans: Lagrangian relaxation and heuristic approaches. *Transportation Research, 23B,* 415–431; Also (1992) Designing optimal railroad operating plans: A dual adjustment method for implementing Lagrangian relaxation, *Transportation Science, 26,* 262–279.
8. Lina, B. L., Wanga, Z. M., Jia, L. J., Tiana, Y. M., & Zhoud, G. Q. (2012). Optimizing the Freight train connection service network of a large-scale rail system. *Transportation Research, 46B,* 649–667.
9. Murty, K. G. (1992). *Network programming*. New York: Prentice-Hall.
10. Newton, A. M., & Yano, C. A. (2001). Scheduling trains and containers with due dates and dynamic arrivals. *Transportation Science, 35,* 181–191.

Chapter 17
Cutting Stock Problems in the Paper and Sheet Metal Industries

G. S. R. Murthy and Katta G. Murty

1 Introduction

Paper, textiles, plastic film like cellophane, metallic foil, sheet metal used for producing sheet-metal components for automobile and other industries; are manufactured in rolls of large width referred to as jumbo rolls, stock rolls, master rolls, or raw rolls etc. Customers who use these products in their production processes place orders for rolls in specified widths according to their needs, referred to as final rolls, orders, production rolls, etc.

A final roll of width w inches can be cut from a master roll of width W inches as long as $w < W$. For example, a master roll of width 100 in. (100 inches) can be cut into two final rolls of width 32 in. each, and one final roll of width 35 in. simultaneously in one cutting operation. Combinations of final rolls are cut from master rolls on a winding machine, called the Winder, which is equipped with a number of knives. Most Winders in paper machine companies have ten knives. Required number of adjacent knives are used for cutting the final rolls. As the master roll is unwound on the Winder, the knives set on the Winder slice the master roll into final rolls of required widths. For the above example, four knives are set on the Winder (to cut k final rolls $k + 1$ knives are needed) so that the distances between two adjacent knives are 32 in. each, and the distance between one pair of adjacent knives is 35 in. At the other end, the sheets are wound again into final rolls of widths 32 in., 32 in., 35 in. The deckle outside the extreme knives is called trim loss or trim waste. In this

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_17) contains supplementary material, which is available to authorized users.

G. S. R. Murthy (✉)
SQC & OR Unit, Indian Statistical Institute, Street No.8,
Habsiguda, Hyderabad 500 007, India
e-mail: murthygsr@gmail.com

K. G. Murty
Department of Industrial and Operations Engineering, University of Michigan,
Ann Arbor, MI 48109-2117, USA
e-mail: murty@umich.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_17, © Springer Science+Business Media New York 2015

cutting process, each of the three final rolls has the same diameter as the original 100 in master roll; and the same length of sheet. Cutting the master roll of 100 in., the cutting scheme adopted for cutting it into two rolls of 32 in. width, and one roll of 35 in. width is known as a *cutting pattern*, or just pattern adopted for cutting this master roll. Since the trim waste in this cutting scheme of the master roll of 100 in. width, is of 1 in., the trim waste generated in this scheme is 1 % of the master role.

With a complex list of customer orders to fill, the problem of finding the economical way of cutting the available master rolls (i.e., the cutting patterns to adopt for cutting the master rolls) into the required final rolls, while minimizing the percentage trim waste is known as a cutting stock problem. In this problem, as all the cutting is along the one-dimensional width line of the rolls, this cutting stock problem is classified as a *1-dimensional cutting stock problem*. This problem appears in the paper, steel, and a wide variety of other industries. So, there is a large economic incentive to find optimal or near-optimal solution procedures for this problem. The Russian economist Kantarovich first discussed the problem in [4] and later outlined approaches for solving it using Linear Programming in [5].

Correspondingly, the simplest *2-dimensional cutting stock problem* is that in which we have to fill orders for rectangular sheets of specified lengths and breadths, by cutting available rectangular master sheets of given lengths and breadths. In the 2-dimensional case, there are several different ways of generating feasible cutting patterns. The simplest and most commonly used method in industry is based on *guillotine cuts* in two or more stages, i.e., in each stage each cut is along the length or width, and breaks the piece being cut into two pieces. In this most commonly used scheme, the pattern in the first stage involves cutting the master sheet into a set of strips by making a series of guillotine cuts, either along the length or the width of the master sheet. In the second stage each strip is cut into a set of sheets for some of the orders by a series of guillotine cuts; and the process is continued in additional stages as needed. So, under this scheme, each stage could be a 1-dimensional cutting stock problem, but often additional production constraints may make the problem different.

In this chapter, we will present an application of the 1-dimensional cutting stock problem in the paper industry for cutting jumbo rolls to fulfill customer orders for paper rolls of smaller widths; and an application of the 2-dimensional cutting stock problem under the scheme discussed above in the important application in sheet metal industries to fulfill orders for metal sheet components in automobile companies, and other industries.

2 The 1-Dimensional Cutting Stock Problem

As an example, we consider a company which cuts jumbo rolls into final rolls of smaller widths called the child reels (see Fig. 17.1), all of the same sheet length as the original. On a particular day, suppose they have to fill orders for final rolls using jumbos with the following data: $(w_1, \ldots, w_m)$, $(b_1, \ldots, b_m)$, where m is the number of final rolls of different smaller widths the company has to fill, w_i, for $i = 1$ to m,

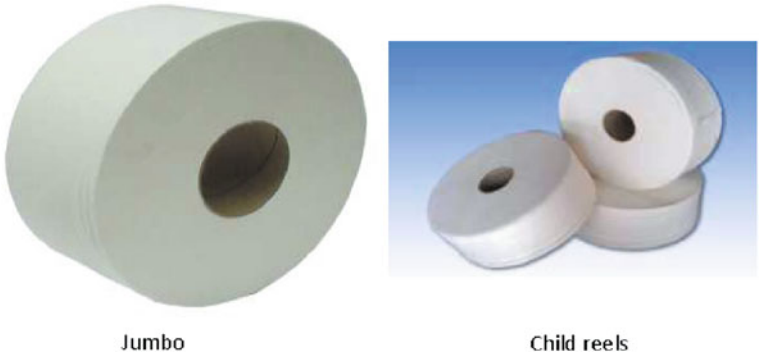


Fig. 17.1 The big Jumbo is slit into child reels of required widths

is the width in inches of the i^{th} order, and b_i is the number of rolls of width w_i of the i^{th} order (all of the same diameter as the original master roll) to be filled on that day. We shall write $w = (w_i : i = 1 \text{ to } m)$, $b = (b_i : i = 1 \text{ to } m)^T$. All the rolls are to be cut from master rolls of width W in. We can fill all the orders using these master rolls if $W > w_i$ for all $i = 1$ to m, which we assume. For example, $w = (40, 35, 30, 20, 10)$, $b = (200, 600, 400, 300, 1000)^T$ is a 1-dimensional cutting stock problem with 5 orders.

A general cutting pattern $a = (a_i : i = 1 \text{ to } m)^T$ yields a_i rolls of width w_i from a single master roll. For the pattern a to be a *feasible cutting pattern*, it has to satisfy the following conditions:

$$\sum_{i=1}^{m} a_i w_i \leq W \tag{17.1}$$

$$\text{and all } \; a_i \geq 0 \quad \text{and integer}$$

J denotes the set of all feasible cutting patterns for cutting master rolls into final rolls in our problem.

$A_{.j} = (a_{ij} : i = 1 \text{ to m})^T$ denotes the jth feasible cutting pattern (i.e., feasible solution of (17.1)) for $j \in J$.

x_j denotes the number of master rolls slit using the j-th feasible cutting pattern for $j \in J$.

$c_j = W - \sum_{i=1}^{m} a_{ij} w_i$ is the width in inches of trim waste in cutting a master roll using pattern $j \in J$.

$|J|$ = the total number of feasible cutting patterns, i.e., the number of different ways of cutting a master roll into required final rolls is finite, but grows exponentially large with m. So, unless m is small, it becomes impractical to enumerate all the feasible cutting patterns. But for explaining the theoretical model for this problem, we will now consider the version for minimizing the trim loss incurred (alternatively, one might consider the problem of minimizing the number of master rolls used) for filling the orders.

Then the problem of minimizing the total trim loss in filling the orders is:

$$\begin{aligned} &\text{Minimize} \sum_{j \in J} c_j x_j \\ &\text{subject to} \sum_{j \in J} a_{ij} x_j \geq b_i, i = 1, \ldots, m \\ &x_j \geq 0, \text{ and integer for all } j \in J. \end{aligned} \tag{17.2}$$

The LP relaxation of this problem is:

$$\begin{aligned} &\text{Minimize} \sum_{j \in J} c_j x_j \\ &\text{subject to} \sum_{j \in J} a_{ij} x_j \geq b_i, i = 1, \ldots, m \\ &x_j \geq 0, \text{for all } j \in J. \end{aligned} \tag{17.3}$$

In the paper and other industries where this problem has frequent applications, customers, mostly regular customers, place orders with the company on a periodic basis like every week, month, etc. In that situation, if the number of final rolls of any required width w_i produced in the optimum solution of the model exceeds the required number for them, b_i in the current order i, the company would have a warehouse where the excess number of these final rolls produced can be stored and used towards fulfilling the order in the next period. In the literature, the additional pieces are called the *off cuts*. That's why the constraints in model (17.2) for the problem are chosen as "$\geq$" inequalities and not equations. Also in this case, the right hand side constants b_i in model (17.2) are normally taken as the ordered quantities minus any final rolls of this width already in storage in the warehouse.

In these industries, each master roll is always sliced completely (i.e., for the full length of the roll), so the decision variables x_j must be integers for real world implementation, that's why (17.2) is an integer program. However, rounding the fractional values of any non integer variables x_j in the optimum solution of the LP relaxation (17.3) up or down, with subsequent ad hoc adjustments as required, has been found to produce a near-optimal solution of the integer model (17.2) that is quite satisfactory in these industries. That is why we will consider (17.3) as the model to derive a solution to these problems.

But there is one major difficulty with model (17.3) for real world implementation. For commonly encountered problems in these industries, the number of feasible cutting patterns, $|J|$ grows exponentially as a function of m, the number of orders for final rolls the company has to fill in any period. For typical periods, this number will be huge, and it is impractical to enumerate all the feasible cutting patterns and generate the whole model (17.3) for the problem. Fortunately in the 1960's Gilmore and Gomory [1, 2] pioneered an alternative approach known as the *delayed column generation approach* for solving this model (17.3) for the 1-dimensional cutting stock problem starting with just a few feasible cutting patterns at the beginning, and

generating additional feasible patterns as needed, one at a time, using an auxiliary problem which is a knapsack problem; and they showed that this approach is guaranteed to converge to the optimum solution of (17.3) without the need to enumerate all the feasible cutting patterns in advance.

In the next section, we will discuss this approach for the 1-dimensional cutting stock problem.

2.1 *The Gilmore-Gomory Delayed Pattern Generation Approach for Solving the 1-Dimensional Cutting Stock Problem*

We consider the original model (17.3) with the $W, w = (w_1 \ldots w_m), b = (b_1 \ldots b_m)^T$ given. This algorithm, is initiated with an initial set of m feasible cutting patterns: $A_{.j} = (a_{1j} \ldots a_{mj})^T$, $j = 1$ to m such that the set of vectors $\{A_{.1} \ldots A_{.m}\}$ is linearly independent, and the matrix B consisting of them as column vectors is a feasible basis for (17.3). Then, starting with B as the first feasible basis for (17.3), the method consists of solving it with Dantzig's revised simplex method, using in each step an auxiliary knapsack model to check optimality of the current basis for (17.3), and generating a new eligible column vector of (17.3) to enter the basis leading to the next basis for (17.3) in the revised simplex method (see [6] for the notation used to describe the revised simplex method).

Let c_B be the row vector of cost coefficients in (17.3) corresponding to the basis B, and $\pi = (\pi_1 \ldots \pi_m) = c_B B^{-1}$, the dual basic solution (basic solution of the dual problem of (17.3)) corresponding to the basis B.

Let $A_{.(m+1)} = (a_{1,m+1} \ldots a_{m,m+1})^T$ denote a column vector of the coefficient matrix in (17.3) which is eligible to enter the basis B in this step of the revised simplex method for solving (17.3), and $c_{m+1} = W - \sum_{i=1}^{m} a_{i,m+1} w_i$ be the cost coefficient in (17.3) associated with it. This column is eligible to enter the basis B to improve the objective value if $c_{m+1} - \sum_{i=1}^{m} \pi_i a_{i,m+1} < 0$.

The auxiliary problem determines the cutting pattern $A_{.m+1} = (a_{1,m+1} \ldots a_{m,m+1})^T$ to minimize $c_{m+1} - \sum_{i=1}^{m} \pi_i a_{i,m+1}$. This leads to the Knapsack problem

$$\begin{aligned} &\text{Maximize} \sum_{i=1}^{m} (\pi_i + w_i) a_{i,m+1} \\ &\text{subject to} \sum_{i=1}^{m} a_{i,m+1} w_i \leq W \\ &a_{i,m+1} \geq 0, \text{and integer for all } i. \end{aligned} \tag{17.4}$$

In the knapsack problem (17.4), the nonnegative integer variables are $a_{i,m+1}$ for $i = 1$ to m. There are efficient commercial software packages for these knapsack problems [http://en.wikipedia.org/wiki/Knapsack_problem#References]. If the optimum objective value in (17.4) is $\leq W$, then the feasible solution of (17.3) corresponding basis B is an optimum solution of (17.3) and the algorithm terminates. Otherwise the optimum solution of (17.4), $(\bar{a}_{1,m+1} \ldots \bar{a}_{m,m+1})^T$, gives the new cutting pattern to

enter into the basis B in this step of the revised simplex method for solving (17.3), and the algorithm is continued with the resulting new basis.

Procedure for Finding A Good Initial Feasible Basis B for (17.3) We will describe the procedure for this discussed in [8], reported to give a near-optimal BFS (Basic Feasible Solution) for (17.3). This procedure is based on the intuition that if the wide finals are out of the way first, the narrow ones are quite versatile to compete for the leftovers. It takes exactly m steps, selects one column of B (i.e., one new cutting pattern) in each step and also simultaneously finds the value of its associated basic variable x_j in the BFS of (17.3) corresponding to it. The BFS x of (17.3) generated satisfies all the constraints in (17.3) as equations.

Reorder the final widths in decreasing order so that $w_1 > w_2 > w_3 > \ldots > w_m$. Define the set $R = \{1 \ldots m\}$ initially. At the end of each step in this procedure, exactly one index will be deleted from the set R.

Define the vector $(b'_i : i = 1, \ldots, m)$ = vector of residual demands (i.e., amounts not filled so far in previous steps in the procedure) which is initially the original demand vector b whose coordinates are permuted, if necessary, so that b_i corresponds to $w_i, i = 1, \ldots, m$. At the end of each step in this procedure, one component of the current vector b' will become 0, and if b'_i is the element that becomes zero in this step, then the index i is deleted from the current set R at the end of this step. If there is a tie for the index i such that b'_i becomes 0 at the end of this step, only one of those indices i that are in the tie is deleted from the set R; and we fix that $b'_i = 0$ for subsequent work in the procedure.

At every stage the number of entries of the current vector b' which are $\neq 0$ will be $\leq$ the current $|R|$. We will now describe the general step j in this procedure for $1 \leq j \leq m$.

General Step j: By the discussion above, clearly at the beginning of this step we will have $|R| = m - (j - 1)$, where R is the current set. Let $b' \geq 0$ be the current vector of residual demands for finals of width w_i for i in the current set R. If $j > 1$, in earlier steps of this procedure, the column vectors $A_{.k}$ of the basis B would have been determined for $k = 1$ to $j - 1$.

Define the j-th column vector $A_{.j} = (a_{ij} : i = 1 \ldots m)^T$ in the basis B for (17.3) being generated, by the pattern: $a_{ij} = 0$ if $i \notin R$, and = the largest integer $\leq (W - \sum_{k=1}^{i-1} w_k a_{kj})/w_i$ if $i \in R$.

Now determine the value of the component x_j associated with this column in the BFS of (17.3) corresponding to the basis B being generated, from $x_j =$ minimum$\{b'_i/a_{ij} :$ over $i \in R$ such that $a_{ij} > 0\}$. From this choice, we will have $x_j a_{ij} \leq b'_i$ for all $i \in R$, and $x_j a_{tj}$ will be equal to b'_t for all indices $t \in R$ which tie for the minimum above.

Update the set R by deleting one of the indices t satisfying $x_j a_{tj} = b'_t$ from the current set R. Update the value b'_i for each i in the updated set R to (current b'_i) $- x_j a_{ij}$. If $j < m$, go to the next step with the updated R and b'. Procedure terminates after the m-th step when the set R becomes $\emptyset$.

Example 1: Consider the cutting stock problem mentioned at the beginning of Sect. 2, with $m = 5$, $W = 150$ in., $w = (w_i) = (40, 35, 30, 20, 10)$in., and $b = (b_i : i = 1 \ldots m) = (200, 600, 400, 300, 1000)^T$.

The entries in w are already in decreasing order. The procedure goes through the following steps.

Step 1: $R = \{1,2,3,4,5\}$, $b' = (200,600,400,300,1000)^T$. So $A_{.1} = (3,0,1,0,0)^T$. $x_1 = \text{minimum}\{200/3, 400/1\} = 200/3$. The index attaining this minimum is 1, so we delete this index 1 from the current R to update it to $\{2,3,4,5\}$, and update b' to $(b'_2, b'_3, b'_4, b'_5)^T = (600, 1000/3, 300, 1000)^T$.

Step 2: Current $R = \{2,3,4,5\}$, $b' = (600, 1000/3, 300, 1000)^T$. So $A_{.2} = (0,4,0,0,1)^T$. $x_2 = \text{minimum}\{600/4, 1000/1\} = 150$. The index attaining this minimum is 2, so we delete this index 2 from the current R to update it to $\{3,4,5\}$, and update b' to $(b'_3, b'_4, b'_5)^T = (1000/3, 300, 850)^T$.

Step 3: Current $R = \{3,4,5\}$, $b' = (1000/3, 300, 850)^T$. So $A_{.3} = (0,0,5,0,0)^T$. $x_3 = \text{minimum}\{1000/15\} = 1000/15$. The index attaining this minimum is 3, so we delete this index 3 from the current R to update it to $\{4,5\}$, and update b' to $(b'_4, b'_5)^T = (300, 850)^T$.

Step 4: Current $R = \{4,5\}$, $b' = (300,850)^T$. So $A_{.4} = (0,0,0,7,1)^T$. $x_4 = \text{minimum}\{300/7, 850/1\} = 300/7$. The index attaining this minimum is 4, so we delete this index 4 from the current R to update it to $\{5\}$, and update b' to $(b'_5)^T = (5650/7)^T$.

Step 5: Current $R = \{5\}$, $b' = (5650/7)^T$. So $A_{.5} = (0,0,0,0,150/10)^T = (0,0,0,0,15)^T$. $x_5 = \text{minimum}\{5650/(7 \times 15)\} = 5650/105$. The index attaining this minimum is 5, so we delete this index 5 from the current R to update it to $\emptyset$, and the procedure terminates.

The basis B obtained consists of column vectors $A_{.1}$ to $A_{.5}$ obtained above in that order; and the BFS of (17.3) corresponding to it is $x = (200/3, 150, 1000/15, 300/7, 5650/105)^T$.

Exercise 1. For the cutting stock problem with data $W = 100$ in., $w = (45,36,31,14)$, $b = (97,610,395,211)^T$, construct a feasible basis B for (17.3), and its associated BFS using the above procedure.

Do the same with data $W = 95$in., $w = (25.5, 22.5, 20, 15)$ in., $b = (81,40,35,35)^T$.

Do the same with data $W = 5600$ mm, $w = (1380, 1520, 1560, 1710, 1820, 1880, 1930, 2000, 2050, 2100, 2140, 2160, 2200)$ mm, b = (22, 25, 12, 14, 18, 18, 20, 10, 12, 14, 16, 18, 20)^T.

Let $\hat{B}$ be the optimum basis with which the algorithm terminates. Then the column vectors of $\hat{B}$, which are $\hat{A}_{.1} \ldots \hat{A}_{.m}$ say, are the optimum set of cutting patterns to use. The BFS of (17.3) corresponding to the basis $\hat{B}$, which is $\hat{x} = (\hat{x}_1 \ldots \hat{x}_m)^T = \hat{B}^{-1}b$ is an optimum solution of (17.3). Rounding up this solution, $\lceil \hat{x}_j \rceil$ = smallest integer $\geq \hat{x}_j$ gives the number of master rolls to cut using the cutting pattern $\hat{A}_{.j}$, for $j = 1$ to m. When this policy is implemented some product rolls may be produced in excess of

the order quantity for them, the excess quantity may be stored in the warehouse and used towards the next order. Also, some customers specify that when the number of product rolls of width w_i ordered is b_i the company may deliver any number of these rolls between b_i and $1.1b_i$, so normally excess number of product rolls produced can be included with the orders.

2.2 *Sequential Heuristic Procedures (SHPs) for the 1-Dimensional Cutting Stock Problems*

SHPs generate one good cutting pattern at a time. Each step in an SHP consists of generating a cutting pattern first, then determines the number of master sheets to cut using that pattern to fill as much of the remaining portions of the current order as possible. Then, the vector of the remaining quantities in the order to fill, $\bar{b} = (\bar{b}_i : i = 1 \text{ to } m)$, say, is computed by updating the current vector with the quantities filled in this step; and with this updated vector the procedure moves to the next step.

The key to getting good results with this type of procedure is to make intelligent choices as to the patterns selected in the early stages of the SHP, in the sense that they should result in low trim loss and high usage. Primary advantages of intelligently developed SHP are that they eliminate the need for rounding and work with only integer values, often they give solutions using much smaller number of cutting patterns than those required by the LP solution to the same problem. Their major disadvantage is that towards the end of the procedure, trim loss incurred increases as the number of orders with remaining quantities to fill decreases, due to limited choices left to create good cutting patterns. See [3, 7] for efficient hybrid methods combining Gilmore–Gomory approach and SHPs.

We will now describe an SHP for the 1-dimensional cutting stock problem with data $W, m, w = (w_i : i = 1 \text{ to } m), b^1 = b = (b_i : i = 1 \text{ to } m)$ as described above at the beginning of Sect. 2. This procedure comprises iterations in a sequence. In each iteration, orders for any width w_i may be fulfilled completely or partially. At the beginning of r^{th} iteration of this procedure, b^r denotes the residual order quantities. Precisely, b_i^r is the number of rolls of width w_i that are yet to be cut. If $b_i^r = 0$, it means that the order for rolls of width w_i is fulfilled, and this order will not be considered in future iterations. This actually reduces the dimensionality of the problem sequentially. To explain, suppose b^1 has exactly two zero coordinates and the rest are positive, say $b_3^1 = 0$ and $b_5^1 = 0$ and $b_i^1 > 0$ for i other than 3 and 5. This means the orders for rolls of widths w_3 and w_5 are completed and they will not be considered as orders in the future iterations to come.

General Step r in the SHP: The remaining requirements vector at the beginning of r^{th} stage or iteration, $r \geq 1$, is $b^r = (b_i^r)$. In this step the cutting pattern to use is generated by solving the following knapsack problem with bounds on the decision variables. In this problem the decision variables are $A_{.r} = (a_{1r} \ldots a_{mr})^T$, where a_{ir} = the number of rolls of width w_i to be cut in this pattern $A_{.r}$, for $i = 1$ to m.

Table 17.1 Data for one dimensional cutting stock problem with master roll width $W = 380$ cm

Order number	1	2	3	4	5	6	7	8	9	10
Order width	113	116	109	40	57	57	71	121	58	52
Order quantity	27	17	26	20	3	14	8	25	15	29

Order quantity is number of pieces required

$$\text{minimize } (W - \sum_{i=1}^{m} a_{ir} w_i)$$

$$\text{subject to } \sum_{i=1}^{m} a_{ir} w_i \leq W \qquad (17.5)$$

$$0 \leq a_{ir} \leq b_i^r \text{ and integer for all } i = 1 \text{ to } m.$$

Let $\bar{A}_{.r} = (\bar{a}_{ir} : i = 1 \text{ to } m)$ be an optimum solution for this knapsack problem. Then $\bar{A}_{.r}$ is the cutting pattern to use in this step.

The number of master rolls to cut in this step with the pattern generated above, denoted by x_r, is:
$x_r = \lfloor \min \{b_i^r / \bar{a}_{ir} : \bar{a}_{ir} > 0, i = 1, 2, \ldots, m\} \rfloor$, where $\lfloor h \rfloor$ is integral part of h. Set $b^{r+1} = (b_i^{r+1} = b_i^r - x_r \bar{a}_{ir} : i = 1, \ldots, m)$. b^{r+1} is the updated requirement vector at the end of Step r. If this vector is nonzero, perform one more iteration. The algorithm terminates when the vector b^{r+1} becomes 0.

It can be verified easily that $\lceil (\sum_i^m w_i b_i)/W) \rceil$ (recall $\lceil h \rceil$ is the smallest integer greater than or equal to h) is the minimum number of master rolls to be cut to fulfill the order $(w,\ b)$, even though it may not be possible to find a feasible solution with this minimum number. By comparing the number of master rolls used in the solution obtained with this lower bound we can get an idea on how good the solution is. We shall now illustrate the method with an example.

Example 2. Consider the 1-dimensional cutting stock data in Table 17.1. This example is taken from a leading paper industry, on "deckle optimization." The word "deckle" means the width of the jumbo or the master roll, and deckle optimization is that of minimizing the trim loss of the deckle. In deckle optimization, the orders are met on a production run basis. The data pertain to one such production run. In this production run there were orders for ten products. Each product requires a roll of 1 m diameter with a specified width. For instance, the first order here is for a roll of width 113 cm and the customer wants 27 such rolls. Similarly, order 2 requires rolls of width 116 cm and the order quantity is 17. The order quantity is the number of pieces or rolls required. Notice that the order numbers 5 and 6 have the same width. These two are actually orders made by two customers for the same width. We can actually combine these into a single order of width 57 cm and quantity required is 17, but it is not necessary to do this. In fact, the solution is obtained without doing it.

We shall now go through the SHP solution method described above with respect to this example. To start with, our $b^1 = (27, 17, 26, 20, 3, 14, 8, 25, 15, 29)^T$. We have $W = 380$ and $w = (113, 116, 109, 40, 57, 57, 71, 121, 58, 52)$. At this stage, we solve the knapsack problem given by (17.5). The solution yields the cutting pattern

Trim Loss	x^r	Width w_j	113	116	109	40	57	57	71	121	58	52
		Req b_j	27	17	26	20	3	14	8	25	15	29
		Actual	27	17	26	20	3	14	8	25	15	29
		Cutting Pattern										
0	1	1	2	0	0	1	2	0	0	0	0	0
0	7	2	2	0	0	1	0	2	0	0	0	0
0	6	3	1	1	0	2	0	0	1	0	0	0
0	13	4	0	0	2	0	0	0	0	0	1	2
1	1	5	0	1	0	0	0	0	2	1	0	0
17	8	6	0	0	0	0	0	0	0	3	0	0
32	3	7	0	3	0	0	0	0	0	0	0	0
35	1	8	1	1	0	0	0	0	0	0	2	0
41	1	9	3	0	0	0	0	0	0	0	0	0
54	1	10	1	0	0	0	1	0	0	0	0	3

Fig. 17.2 Situation at termination of the algorithm

$A_{.1} = (2, 0, 0, 1, 2, 0, 0, 0, 0, 0)^T$ whose trim loss is zero. The minimum ratio x^1 in this case is equal to 1. Now set $b^2 = (25, 17, 26, 19, 1, 14, 8, 25, 15, 29)^T$. Next, we solve (17.5) again with b^2. This yields $A_{.2} = (2, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0)^T$ whose trim loss is zero again. Computing the minimum ratio, we get $x^2 = 7$. This process runs for 10 iterations. The cutting pattern generated in each iteration is given in Fig. 17.2. These patterns are shown as rows against the cutting pattern number in Fig. 17.2. The trim loss and the value of x^r of each cutting pattern are shown in the first and second columns of Fig. 17.2 respectively. The first row of Fig. 17.2 presents the order widths, the second row presents the order quantities and the third row presents, against the heading "Actual", actual number of pieces produced at the end of the algorithm for each order according to the obtained solution.

Example 3. Let us consider one more instance (another run) of the deckle optimization from the same company. In this run we have 25 orders. The input data are given by $m = 25$, $W = 380$, $w = (57,40,74,81,66,85,127,136,88,56,77,113,131,58,116,64,51,77,100,122,114,113,50,105,52)$ and $b = (14,20,26,24,24,16,8,22,21,3,2,18,22,18,17,30,15,10,15,1,16,26,27,15,29)$. The units for all the widths is centimeters. Figure 17.3 displays the solution obtained for this example.

Exercise 2. Verify the solutions of examples 2 and 3.

Exercise 3. For Example 3, show that the number of knapsack problems to be solved in the SPH algorithm discussed above is 24.

Exercise 4. Use the above heuristic and find solution to the following problem for which the data are given below. Also solve it using the Gilmore-Gomery approach and compare the results obtained under the two methods.

Data: $W = 380, w = (88, 120, 78, 105, 116, 71, 59, 85, 52, 81, 127, 40, 77, 122, 110)$, $b = (19, 7, 29, 15, 8, 8, 4, 15, 29, 24, 8, 20, 10, 1, 19)$, all widths in centimeters.

Trim Loss	x_i	Width	57	40	74	81	66	85	127	136	88	56	77	113	131	58	116	64	51	77	108	122	114	113	50	105	52
		Req	14	20	26	24	24	16	8	22	21	3	2	18	22	18	17	30	15	10	15	1	16	26	27	15	29
		Actual	14	20	26	24	24	16	8	22	21	3	2	18	22	18	17	30	15	10	15	1	16	26	27	15	29
		Cutting Pattern																									
0	8	1	1	2	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	4	2	1	1	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	6	3	0	0	1	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	4	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	5	0	0	2	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	6	0	0	0	0	4	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
0	5	7	0	0	0	0	0	0	0	0	3	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
0	2	8	0	0	1	0	2	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
0	6	9	0	0	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
0	2	10	0	0	1	0	1	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	5	11	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0
0	5	12	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	3	0	1	0	0	0	0	0	0
0	1	13	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	11	14	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	5	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	0	0	0
0	1	16	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	0	0	0
0	5	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	2	0	0	0	0	0	1
0	8	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2	0	1
0	1	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	4	0	1
0	7	20	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	2
1	5	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	3	0
2	1	22	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	1
5	4	23	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0
41	1	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0

Fig. 17.3 Situation at termination for example 3

Exercise 5. Explain what changes need to be made in the above algorithms if the number of cutting knives on the cutting machine can be no more than an upper bound u. Solve the numerical problems in Exercises 1, 3, 4 with this upper bound $u = 10$.

3 The 2-Dimensional Cutting Stock Problems

In this section we will discuss an application of the 2-dimensional cutting stock problem at MMC company, producing sheet-metal components for leading automobile companies and other industries. MMC receives orders for sheet-metal components from various companies on a monthly basis regularly. Here we discuss the application to meet orders for components produced from sheet metal of thickness 3.15th of an inch. These components are called *FJ Components*; and the company receives orders for them every month. As an example Table 17.2 shows the requirements for components placed for the months of October, November 2012 (Table 17.3).

MMC uses *master sheets* (also called *standard sheets*) of length 2500 mm and breadth 1250 mm and total area of $2500 \times 1250 = 31250000\ mm^2$ for producing these FJ components. The direction along the length line of 2500 mm of the master sheet is called the *normal direction*, while the direction along the breadth line of 1250 mm is called the *tangential direction*.

There are specifications on the grain structure of sheets of components C1 to C11, for example for component C1, these specifications require that its length line of 87 mm must be in the normal direction (i.e., the length line of the component should be along the length line of the master sheet). This specification is expressed with the entry of "1" in the column with heading "Dir" in Table 17.2. For components C12 to C19 with the entry of "2" in this column, this specification is not there. So,

Table 17.2 FJ component requirements and their particulars

Part	Component	l_j	b_j	Dir	$r_j - Oct$	$r_j - Nov$
C1	B.Tube Bkt	87	32	1	800	700
C2	Ang.SupPil.	83	32	1	800	800
C3	Bracket	167	76	1	500	600
C4	Spare Bkt	60	185	1	1500	1250
C5	Assy Pivot Bkt	178	108	1	800	700
C6	Shift Plate	103	132	1	500	550
C7	Out rig. bkt	168	50	1	100	200
C8	brkt	150	25.4	1	500	300
C9	D.F.MTG BRKT	211	116	1	150	150
C10	D.F.MTG BRKT	160	130	1	150	175
C11	Clamp Ste.	126	25	1	1000	1200
C12	Cut Washer	65	45	2	1500	1300
C13	Rad mtg bkt	54	246	2	1000	1100
C14	X reinf bkt	112	65	2	400	200
C15	D.F.MTG BRKT	140	85	2	150	175
C16	BRKT F. Line	167	25	2	150	150
C17	S.Brake Tube	125	25	2	150	200

Note: l_j, b_j, r_j are the length, breadth in mm,
and the number of pieces of component Cj required;
q_j is the number of tangential strips (explained later
in the text) needed to meet requirements for Cj;
Dir. is "direction" stipulated for components.

Table 17.3 Number of tangential strips needed to fill the order for components C1 to C11 which are constrained to be produced using these strips only

Part	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
s_j	39	39	16	6	11	9	25	49	10	9	50
q_j	21	21	32	250	73	56	4	11	15	17	20

for example, component C12 can be cut from the master sheet with its length line either in the normal or tangential direction on the master sheet.

Process and Constraints for Production of FJ Components The units of all widths and lengths in this section will be in millimeter. We will describe the production process for FJ components using component C1 of dimensions 87×32 as an example. A master sheet is first slit on a *slitting machine* into *strips* of dimensions 87×1250 by slitting the master sheet along its length with guillotine cuts; these strips are called *tangential strips* (strips cut from the master sheet by slitting it along the width are called *normal strips*). To produce components C1, once tangential strips of dimensions 87×1250 are cut, copies of component C1 are then obtained from each strip using blanking or pressing operations. Each strip yields 39 (integer portion of 1250/32) copies of C1. Since the number of components C1 required is 800, to fill the order for the month of October for component C1 requires 21 strips. This consumes $87 \times 21 = 1827$ length of the master sheet along the normal direction. The balance length of the sheet of dimensions 673×1250 is then used to produce the next component in the production sequence. This cutting pattern of strips is shown in Fig. 17.4.

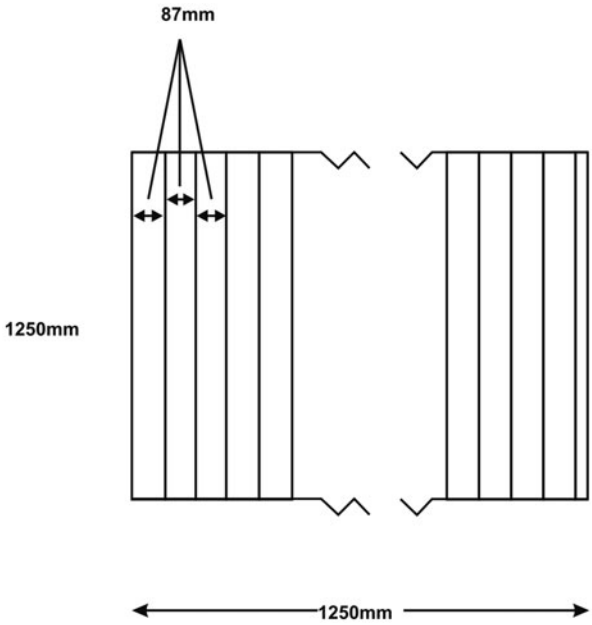


Fig. 17.4 Patterns for slitting master sheet into tangential strips

In the equipment available at the company, the length of the slitting blade on the slitting machine is 1500 mm; so it is not possible to have slitting length greater than 1500 mm. This leads to a constraint on the type of cutting patterns that can be implemented. For example, on the master sheet of dimensions 2500×1250, preparing normal sheets by slitting along the width line of the master sheet requires a slitting length of 2500 (> 1500) which is not possible. So, for using normal sheets to produce components, we first have to slit the master sheet into two sheets of dimensions $L \times 1250$ and $(2500 - L) \times 1250$, and then use sheets to obtain the normal strips for producing the components; here L can be chosen appropriately so that both L and $2500 - L$ are ≤ 1500, i.e., $1000 \leq L \leq 1500$.

The Decision Making Problems For processing each order, the company needs the solutions for the following decision making problems:

(i) Find the minimum number of master sheets reqired to fulfill the order. So, here our objective is to minimize the number of master sheets needed.
(ii) Find the cutting patterns to be used to minimize the overall trim waste. Here the company has decided that only two ways of production will be considered, these are:

   (a) each master sheet is converted into tangential strips first, and then the required number of components are extracted from them; and
   (b) each master sheet is converted into normal strips first, and then the required number of components are extracted from them.

We consider a general order requiring r_j pieces of component j with length and breadth $l_j$mm and b_j mm for $j = 1, 2, \ldots, k$. Define for $j = 1, 2, \ldots, k$,

s_j = integer part of$1250/b_j$, i.e., largest integer $\leq 1250/b_j$
= number of pieces of component Cj that can be cut from a tangential strip of the master sheet of length l_j mm

q_j =smallest integer $\geq r_j/s_j$ = number of tangential strips of length $l_j$of a master sheet needed to fulfill the requirement for this component

In this order suppose for $j = 1, \ldots, p$, component j has direction stipulation 1 (i.e., it can only be made using tangential strips), while components $j = p+1, \ldots, k$ have direction stipulation 2 (i.e., these components can be made using tangential or normal strips).

A Procedure for the 2-D Problem Using 1-D SHP We shall use the 1-dimensional procedure described in Sect. 2.2 to solve this 2-dimensional cutting stock problem. We shall use the October data to illustrate the procedure. As stated earlier, we shall assume that the first p components are unidirectional, that is, their length direction specification is 1 and the direction specification for the rest of the components $(p + 1, \ldots, k)$ is 2. For October data, $p = 11$ and $k = 17$ (k is the total number of component types). Let g be the number of component types with direction specification 2. For October, $g = 6$, and $k = 17 = 11 + 6 = p + g$.

In order to get an efficient solution for the problem, we first introduce, g dummy components corresponding to the components with direction specification 2. With this introduction of dummy components, the total number of component types becomes $h = p + 2g$. For October data, $h = 23$, and the dummy components will be indexed by $k + 1, k + 2, \ldots, h$. Define the length and width requirements for the dummy components as follows. For $k+1 \leq j \leq h$, the dimensions of the j^{th} component are $b_{p+j-k} \times l_{p+j-k}$. That is, for $k + 1 \leq j \leq h$, $l_j = b_{p+j-k}$ and $b_j = l_{p+j-k}$. As an example consider $j = 18$ for October data; here $p+j-k = 11+18-17 = 12$; thus the dimensions of 18th component will be $l_{18} \times b_{18} = b_{12} \times l_{12} = 45 \times 65$.

We shall set the requirements, r_j, for the dummy components ($j = k+1, \ldots, h$) as zero. For October, $r_j = 0$ for $j = 18, \ldots, 23$. The dimensions and the requirements for October data along with dummy components are shown in Table 17.4.

Recall that s_j is the number of j^{th} components that a tangential strip of dimension $l_j \times 1250$ will yield. For $j = 1, \ldots, k$, $s_j = \lfloor \frac{1250}{l_j} \rfloor$; and for $k + 1 \leq j \leq h$, $s_j = \lfloor \frac{1250}{b_j} \rfloor = \lfloor \frac{1250}{l_{p+j-k}} \rfloor$. The s_js are presented in Table 17.4.

To solve the 2-dimensional problem using the 1-dimensional SHP discussed earlier we follow the procedure given below.

Step 1: Set $m = k$, $w_j = l_j$, $j = 1, 2, \ldots, h$, and for $j = 1, \ldots, k$ let r_j be the number of pieces of component j required. For the October data, $m = 17$,

Table 17.4 FJ component along with dummy components

Part	Component	l_j	b_j	A_j (mm^2)	Dir	r_j	s_j
C1	B.tube Bkt	87	32	2784	1	800	14
C2	Ang.SupPil.	83	32	2656	1	800	15
C3	Bracket	167	76	1,2692	1	500	7
C4	Spare Bkt	60	185	1,1100	1	1500	20
C5	Assy pivot Bkt	178	108	1,9224	1	800	7
C6	Shift plate	103	132	1,3596	1	500	12
C7	Out rig. bkt	168	50	8400	1	100	7
C8	brkt	150	25.4	3810	1	500	8
C9	D.F.MTG BRKT	211	116	2,4476	1	150	5
C10	D.F.MTG BRKT	160	130	2,0800	1	150	7
C11	Clamp Ste.	126	25	3150	1	1000	9
C12	Cut washer	65	45	2925	2	1500	19
C13	Rad mtg bkt	54	246	1,3284	2	1000	23
C14	X reinf bkt	112	65	7280	2	400	11
C15	D.F.MTG BRKT	140	85	1,1900	2	150	8
C16	BRKT F. Line	167	25	4175	2	150	7
C17	S.brake tube	125	25	2135	2	150	10
C18	Cut washer	45	65	2925	2	0	27
C19	Rad mtg bkt	246	54	1,3284	2	0	5
C20	X reinf bkt	65	112	7280	2	0	19
C21	D.F.MTG BRKT	85	140	1,1900	2	0	14
C22	BRKT F. Line	25	167	4175	2	0	50
C23	S.brake tube	25	125	2135	2	0	50

Note: l_j, b_j, r_j are the length, breadth in mm, Area $A_j = l_j \times b_j$, and the number of pieces of component Cj required; s_j is the number of Cj components per strip; Dir. is stipulated "direction"

$h = 23$, $(w_1, \ldots, w_{23})^T$ is the column under l_j in Table 17.4; and the r_js are also given in the table but these are to be considered only for $j = 1$ to 17.

Step 2: Solve the following problem for a_js.

$$\text{Minimize } (1250W - \sum_{j=1}^{m} a_j s_j A_j)$$

$$\text{subject to } \sum_{j=1}^{h} a_j w_j \le W \tag{17.6}$$

$$0 \le s_j a_j \le \max(s_j, r_j) \ \ for \ 1 \le j \le p, \tag{17.7}$$

$$0 \le s_j a_j + s_{k+j-p} a_{k+j-p} \le \max(s_j, r_j) \ \ for \ p+1 \le j \le k, \tag{17.8}$$

a_j integer for all $j = 1$ to h.

Note that the middle term $s_j a_j + s_{k+j-p} a_{k+j-p}$ in (17.8) is the total number of pieces of component Cj with direction specification 2 (the first part of this expression $s_j a_j$ is the number obtained from the tangential strips, and $s_{k+j-p} a_{k+j-p}$ is the number obtained from normal pieces) obtained from the cutting pattern $(a_1, \ldots, a_h)$. The RHS expression $\max(s_j, r_j)$ of (17.7) and (17.8) is put as the maximum to take care of the situation when an updated r_j in an iteration falls below s_j (i.e., $0 < r_j < s_j$). If $0 < r_j < s_j$, then a_j will be forced to become 0 which is undesirable.

Step 3: Let $(a_j, 1 \le j \le h)$ denote the solution (the cutting pattern) obtained for the above problem in Step 2. Let x be the largest nonnegative integer for which $r_j - x a_j s_j \ge 0$ for $1 \le j \le p$, and $r_j - x(s_j a_j + s_{k+j-p} a_{k+j-p}) \ge 0$ for $p < j \le k$. Cut x master sheets according to the cutting pattern $(a_j, 1 \le j \le h)$. Reset r_j as $r_j - x a_j s_j$ for $1 \le j \le p$, and $r_j - x(s_j a_j + s_{k+j-p} a_{k+j-p})$ for $p < j \le k$. After this adjustment, for any $1 \le j \le k$, if $r_j \le 0$, it means that the demand for Cj is met and for such js, set $w_j = 0$. If all w_js are equal to 0, then stop, else go to Step 2 with the updated data.

Let us apply the above procedure to October's data. The relevant input data are presented in Table 17.4.

In the initial step, $m = k = 17, h = 23$ the w-vector and r-vector are given in Table 17.4 under the columns titled l_j and r_j respectively. Solving the problem in Step 2, we get the first cutting pattern a in which $a_{13} = 1, a_{18} = 51, a_{22} = 3, a_{23} = 3$ and all other a_js are zero (see the cutting pattern 1 in third row of Fig. 17.5).

Note that according to this cutting pattern, $a_{18} = 51$ means cutting component C12 (as C18 corresponds to C12) in normal direction. Note that $s_{18} = 27$ which implies that each of the 51 strips of 1250 mm length will yield 27 pieces of C12. Therefore, the total number of pieces obtained for C12 from this cutting pattern is $s_{12} a_{12} + s_{18} a_{18} = 19 \times 0 + 27 \times 51 = 1377$. Note that this implies x, the number of master sheets to be cut according to this cutting pattern, cannot be more than 1 and hence $x = 1$. Therefore, the total number of pieces for C12 obtained after the initial step is 1377, and hence the starting $r_{12} = 1500$ will be reset to $r_{12} = 1500 - 1377 = 123$. For component C13, $a_{13} = 1$ and $a_{19} = 0$ (a_{19} corresponds to pieces cut in the normal direction for C13). The number of pieces obtained for C13 is $s_{13} a_{13} + s_{19} a_{19} = 23 \times 1 + 5 \times 0 = 23$. Therefore, the starting $r_{13} = 1000$ will be changed to $r_{13} = 977$. For component C16, $a_{16} = 0$ and $a_{22} = 3$ (a_{22} corresponds to pieces cut in the normal direction for C16). The number of pieces obtained for C16 is $s_{16} a_{16} + s_{22} a_{22} = 7 \times 0 + 50 \times 3 = 150$. Therefore, the starting $r_{16} = 150$ will be changed to $r_{16} = 0$. Similar calculations for C17 will show that the starting $r_{17} = 150$ will be changed to $r_{17} = 0$. Note that the requirements for C16 and C17 are met. Therefore, set $w_{16} = 0$ and $w_{17} = 0$ and go to Step 2.

Continuing the procedure, the algorithm terminated at the end of 26 cutting patterns. These cutting patterns are shown as the the rows in Fig. 17.5.

The last column in the figure presents the number of master sheets, x, to be cut according to corresponding cutting pattern. Figure 17.6 presents the number of pieces

CPs	Components Cjs																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18(12)	19(13)	20(14)	21(15)	22(16)	23(17)	x
1													1					51				3	3	1
2												1	42					2		1				1
3				40									1					1						1
4				33																8				1
5		20		1																12				1
6		30																						1
7	22	3		1		2						1												1
8	27	1										1												1
9	8					10															9			1
10						24																		1
11	1					5		2			11			1							1			1
12						1					19													1
13								4			15													5
14								14			3													2
15					4			11			1													1
16					5		5	1		3					1									1
17					5		5	1		3					1									1
18			12				2			1														2
19			8		2		1			4														1
20			4		4					7														1
21			14							1														2
22					14																			6
23			3		10					1														1
24			4		1				7															1
25			1						11															1
26									11															2

Fig. 17.5 Cutting patterns for October data

Component	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17
Requirement	800	800	500	1500	800	500	100	500	150	150	1000	1500	1000	400	150	150	150
Solution	812	810	504	1500	805	504	105	504	200	154	1008	1515	1012	410	156	150	150
Off Cuts	12	10	4	0	5	4	5	4	50	4	8	15	12	10	6	0	0

Fig. 17.6 Summary of solution to October requirements

produced at the termination of the algorithm. The last row in the figure presents *the off cuts*, the number of additional pieces.

Exercise 6. Compute the residual r_j after using the first two cutting patterns of Fig. 17.5. Determine the components whose orders are met at the end of this iteration (i.e., after cutting according to the 3rd cutting pattern).

Exercise 7. Repeat the above exercise for the fourth iteration, that is, after cutting according to the first three cutting patterns.

Exercise 8. Compare the results obtained using the above procedure to the results that will be obtained if the company's procedure in practice is followed. More precisely, determine the total number of master sheets, total trim loss and the off cuts for the two methods and compare them. The company's practice is as follows: Take up first component C1, cut the required number of tangential strips to meet the requirement of this component, and blank the components. If the left over strip of the master sheet can be used for cutting any other component, use it for producing that component and adjust its requirements accordingly. Then take the second component. Cut the master sheet to meet its requirements just as it was done for the first component, and use the left over master strip for cutting any other component(s). Proceed this way until all component requirements are met.

Exercise 9. Using the procedure explained in this section, work out the solution for November data presented in Table 17.3.

Exercise 10. Compare your results for November data with the results if the company's practice is followed.

References

1. Gilmore, P. C., & Gomory, R. E. (1961). A linear programming approach to the cutting stock problem. *Operations Research, 9*(6), 849–859.
2. Gilmore, P. C., & Gomory, R. E. (1963). A linear programming approach to the cutting stock problem- part II. *Operations Research*, *11*, 863–888.
3. Haessler, R. W. (1988). A new generation of paper machine trim programs. *TAPPI Journal, 71*(August), 127–130.
4. Kantarovich, L. V. (1939). Mathematical methods for organizing and planning production (Leningrad State University). *Managament Science*, *6*(4), 366–422.
5. Kantarovich, L. V., & Zalgaller, V. A. (1951). *Calculation of rational cutting of stock*. Leningrad: Lenizdat.
6. Murty, K. G. (2010). *Optimization for decision making: Linear and quadratic models*. New York: Springer.
7. Sweeney, P. E., & Haessler, R. W. (1991). Cutting stock problems and solution procedures. *E*uropean Journal of Operational Research, *54*(2), 141–150. http://deepblue.lib.umich.edu/handle/2027.42/29128.
8. Vasek, C. (1980). *Linear programming*. New York: W.H. Freeman and Company.

Chapter 18
Inventory Management in Blood Banks

Harshal Lowalekar and N. Ravichandran

The Operations Research (OR) and Operations Management (OM) community has witnessed significant progress in the field of research in blood bank inventory management during the past few decades. The OR and OM community has studied blood inventory management problems at the level of both individual hospital blood banks (HBB) as well as regional blood banks (RBB). The OR and OM literature at the individual HBB level has mainly focused on the determination of optimal policies for collection, ordering, componentizing, cross-matching and issuing of blood and its products[1]. The literature at the RBB level emphasizes on the optimal policies for transshipment, rotation, distribution and scheduling of blood deliveries from the RBB to the HBBs[1]. The objective of this chapter is twofold. In the first place, it provides the readers with a basic understanding of inventory management related problems commonly faced by HBBs. In the second place, it shows how various OR techniques can be used to manage the inventory of blood products in an efficient manner with the help of a real-life case study.

This chapter is organized into seven sections. The first section talks of blood and its components. The second section provides a description of the different types of

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_18) contains supplementary material, which is available to authorized users.

[1] Readers are suggested to refer to Prastacos [27] for an in-depth review of literature on blood bank inventory management.

H. Lowalekar (✉)
Operations Management and Quantitative Techniques Area,
Indian Institute of Management Indore, Indore, Madhya Pradesh, India
e-mail: harshal@iimidr.ac.in

N. Ravichandran
Production and Quantitative Methods,
Indian Institute of Management Ahmedabad, Ahmedabad, Gujarat, India
e-mail: nravi@iimahd.ernet.in

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_18, © Springer Science+Business Media New York 2015

blood groups. Functioning of a typical blood bank is explained in the third section. The fourth section describes various operational decision-making problems at HBBs. The fifth section details out a real-life case study. The penultimate section describes recommendations for real-life implementations based on the insights obtained from the real-life case study. The chapter concludes with a set of exercises from the area of blood bank inventory management.

1 What is Blood?

Blood is a mixture of several different types of cells that float in a water-like substance known as plasma [1]. It also contains proteins, antibodies, nutrients, hormones, clotting agents, salts and wastes [1]. Blood performs several important functions in the human body such as carrying oxygen and nutrients to the various cells and tissues of the body, waste removal, fighting against diseases, regulation of the body temperature, regulation of body acidity, etc. [2, 3].

Blood contains mainly three different kinds of cells: red blood cells (RBCs), white blood cells (WBCs) and platelets; by volume, these cells constitute about 45 % of blood while the remaining 55 % is constituted by plasma [3]. The main functions of various kinds of cells are given below:

1. RBCs: They are also known as erythrocytes. These cells are red in color due to haemoglobin (Hb) and they perform the function of carrying oxygen to the tissues in the human body [1]. The condition in which a human body lacks sufficient number of RBCs is known as anemia [2]. RBCs are transfused to the patients suffering from sickle cell anemia, gastrointestinal bleeding and kidney failure [4, 5]. RBCs are also given to the patients suffering from excessive blood loss due to surgery and trauma [4, 5].
2. WBCs: They are also known as leukocytes. WBCs are responsible for fighting against diseases and foreign matter in the human body [1, 3]. WBC transfusions are not common since they can lead to many complications such as transfer of infections along with some transfusion related reactions in the body of the recipient [6].
3. Platelets: They are also known as thrombocytes. Platelets facilitate the process of clotting of blood and are therefore required in cases where a patient is suffering from bleeding disorders [4]. Platelets are quite commonly required for patients undergoing cancer-therapy and open-heart surgery [4, 5].

Plasma is the fluid portion of blood that contains nutrients, salts, antibodies, waste products and clotting proteins [1–3]. Plasma is given to trauma and burn victims and also to the patients with clotting disorders [4, 5]. Plasma transfusions, however, are relatively less popular since the risks associated with plasma transfusions are very high [7]. Plasma transfusion can lead to circulatory overload and it often causes allergic reactions and infections in the body of the person receiving the blood [7].

2 Types of Blood Groups

Based on the presence (or absence) of certain kinds of antigens on the surface of RBCs, human blood can be classified into several blood groups [8]. Antigens are substances which are responsible for stimulating immune responses, usually by the production of antibodies [9]. There are numerous such antigens (proteins, carbohydrates, etc.) and depending upon the type of the antigen, one can have multiple blood group classification systems [8]. International Society of Blood Transfusion (ISBT) recognizes 30 such blood group systems; the two most important being the ABO and the Rh blood group systems [8]. A brief description of the two blood group systems is provided below:

1. *ABO blood group system:* Based on the presence of A and B type antigens on the surface of RBCs, human blood can be classified into 4 groups under the ABO system (A, B, AB and O) [10]. The development of ABO system is credited to Karl Landsteiner (1901) [10]. Typically, the body of a person will develop an antibody in its plasma against the type of the antigen (A, B) which is not present on the surface of its RBCs [8, 10]. For example, a person with blood group type A (i.e., with type A antigen on the surface of its RBCs) will have anti-B antibodies in its plasma. Thus, if blood of type B is transfused to such person, the anti-B antibodies in the plasma of the person will attack the type B antigens of the donor RBCs and cause a severe transfusion reaction. The person with blood group type A, therefore, cannot be transfused type B blood and vice versa. Both A and B blood types can receive blood from a type O donor. A person with O blood type (with neither A nor B type antigens on the surface of its RBCs) will have both anti-A and anti-B antibodies in the plasma and therefore can receive only type O blood. Similarly, a person with type AB blood (with both A and B type antigens on the surface of its RBCs) will not have any ABO antibodies and can receive blood from all the four blood types (type A, type B, type AB and type O).
2. *Rh blood group system:* Based on the presence (or absence) of a certain type of Rh (Rhesus) antigen, known as the D antigen, the blood groups can be classified into Rh+ or Rh− blood groups [8]. Unlike, many other antigens though, the absence of D antigen on the surface of RBCs does not necessarily imply the presence of anti-D antibodies in the plasma [8]. The body of an Rh− person develops anti-D antibodies only after exposure to an Rh+ blood, which in most of the cases happens during transfusion or pregnancy [8].

Thus, based on the ABO and Rh grouping systems, human blood can be divided into eight blood group types (four combinations based on ABO grouping and two combinations based on Rh grouping). The substitution pattern among these blood groups is shown in Fig. 18.1. An arrow between any two blood groups in Fig. 18.1 denotes a possible donor–recipient combination; the tail of the arrow denoting the donor, the head denoting the recipient. Thus for a pair of B− and B+ blood types the former can donate blood to the latter but not vice-versa. It can be seen from the figure that blood type O− is the universal donor while AB+ is the universal recipient.

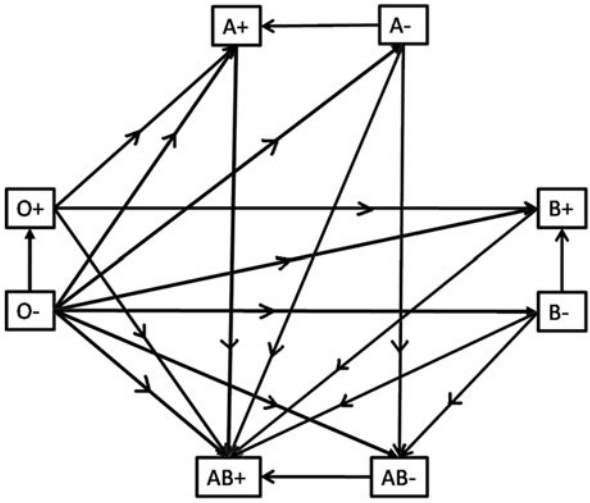


Fig. 18.1 Substitutability among various blood group types

It may be worth noting that even though Fig. 18.1 conveys that a type O− blood can be given to a type A+ blood, such transfusions are never performed in reality[2]. In fact, even when a patient of a certain blood type is to be transfused with blood of the same type, it is extremely important to test, in advance, the possibility of any transfusion reaction that may occur between the recipient and the donor blood. The process is known as cross-matching, where the serum (plasma without the clotting factors) from the patient is mixed with the donor blood to check for any clumping (known as agglutination) which is indicative of a transfusion reaction [11]. It is quite possible that the blood type of the recipient and the donor are same under ABO and Rh system but may not be compatible with each other. It is worthwhile to recall that ABO and Rh are just major blood grouping systems and the incompatibility may arise due to several other antigens which do not form part of the ABO–Rh classification.

3 Blood Bank

A blood bank is a facility which collects, tests, processes and stores blood and its components for future use [12–15]. The primary responsibility of any blood bank is to plan blood collection and respond to the demand of blood and its components from patients. The main objectives of any blood bank are:

1. To ensure that blood products are available in sufficient quantities for the patients who need blood transfusion
2. To ensure that the wastage of blood products is minimized

[2] This concept of substitution between different blood group types is mostly limited to textbooks in medicine. The transfusions between certain blood group types, as shown in Fig. 18.1, are never actually prescribed unless there is an absolute emergency.

The major supply of blood for a typical blood bank comes from blood collections through blood donation drives and camps; other sources of supply include donors who come to the bank and donate blood. Many large hospital-level blood banks collect their own blood through camps and blood bank donations. Some small hospital-based blood banks, however, do not perform blood collections though camps. They order blood from a larger blood bank (or an RBB) at regular intervals. The RBBs usually collect blood in large quantities and supply/replenish blood at regular intervals to a large number of hospitals and/or associated blood banks operating in that region. RBBs also provide blood directly to the patients who require transfusion.

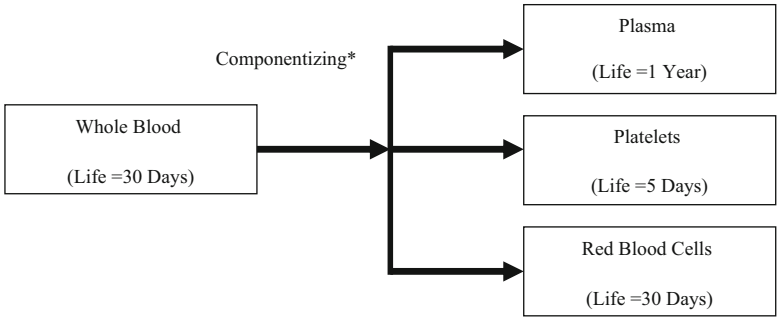
Most of the blood donations fall under the category of homologous donations where blood from anonymous donors is used for transfusion [16]. In some cases, the family of the patient is required to replace the amount of blood transfused to the patient; these are called replacement donations [16]. According to WHO, voluntary non-remunerated donations are the safest [17]. Professional donors are considered to be highly risky since, in many cases, such donors make a living out of selling their blood [18]. They may be donating blood more frequently than what is considered to be medically safe and may not be healthy, which is a very important consideration in ensuring the safety of the blood transfusion process [18]. Replacement donations are also discouraged because in many cases the donations are usually arranged by the patient's family by paying professional donors [18]. In other cases, it is some family member who is donating blood due to necessity and may not otherwise be an ideal candidate for blood donation [18].

3.1 Blood Collection

Blood from donors is usually collected in plastic bags or glass bottles. Blood collected in this manner is generally referred to as whole blood (WB) since it contains all the components like RBCs, WBCs, platelets and plasma at the time of collection [19]. Sometimes, only the required components are collected from a donor through an extracorporeal circulation and the remaining components are returned to the body of the donor [4]. This process is known as apheresis.

Blood is collected and stored in either glass bottles or plastic bags containing anti-coagulant solutions. The use of glass bottles in blood banking is reducing since they can break easily and the blood inside the glass bottles is quite prone to contamination since they need to be opened for the purpose of processing [20]. A single bag is used for storing WB while a double/triple/quadruple bag is used to collect blood which needs to be separated into components (depending upon the number of components required) [21]. A triple/quadruple blood bag is a system consisting of a main blood collection bag and multiple satellite bags connected to it for collecting the components [22]. Single blood bags with capacities of 250 and 300 ml and triple/quadruple bags with capacities of 400 and 450 ml are quite popular in blood banking.

Blood is only collected from those donors who are medically fit for donation. In India, it is common practice to collect blood from donors who weigh more than



*WBCs are usually removed at the time of blood collection by using blood bags with special WBC filters

Fig. 18.2 Componentizing of Whole Blood (Source: Lowalekar and Ravichandran [25])

55 kg, who do not have a history of high blood pressure and who did not donate blood during the 3 months period prior to the day of donation. It is also common for blood banks to collect blood in 450 ml triple/quadruple blood bags from donors who appear to be in good health and weigh more than 65–70 kg and in 250–300 ml single blood bags from the remaining donors.

3.2 Component Separation

In order to separate various components, a triple/quadruple blood bag (filled) is centrifuged at high speed inside a centrifuging machine [23]. Due to centrifugation, various components separate out in the form of multiple layers inside the main bag depending on their densities (Fig. 18.2) [23]. The lightest of the components, plasma, separates out in the topmost layer followed by platelets in the middle and RBCs, the heaviest, at the bottom [23]. These components can be drained from the main bag into their respective satellite bags once the separation of layers is complete. Sometimes blood banks also separate a fourth component called cryoprecipitate from the process of fractionation of plasma. Cryoprecipitate is given to patients with bleeding disorders like hemophilia since it is rich in clotting factors [24]. Since white blood cells (WBCs) can lead to many complications in the patient's body [6], some advanced blood bags come equipped with leukocyte reduction filters. These filters remove most of the WBCs which would otherwise fractionate along with the RBCs.

The biggest advantage of separating WB into components is that only the required components can be given to the patient who needs them. For example, a patient who only needs RBCs would also get platelets and plasma along with RBCs if he is given one unit of WB. This may not be desirable since it can lead to complications arising out of transfusing those components which are not required by the patient.

Table 18.1 Storage conditions for blood products*. (Source: Seeber and Shander [20])

Product	Storage condition	Maximum duration of storage
Red blood cells	1–6 °C	35–49 days depending on storage conditions[a]
Platelets	20–24 °C (in a constantly stirred condition)	5 days
Fresh frozen plasma (FFP)	− 18 °C to − 25 °C	1 year
White blood cells	No storage possible	

*Blood products (or blood items) are same as blood and its components (WB, RBCs, platelets, plasma, WBCs and cryoprecipitate).

[a]Storage conditions refer to the anticoagulant–preservative used in the blood bags along with the sterility and hygiene of the apparatus and the room in which the blood products are stored. In Indian blood banks it is very rare to store blood safely for more than 30 days.

3.3 Typing and Testing

Once the blood unit is collected in blood bags it is tested for infectious diseases such as human immunodeficiency virus (HIV), hepatitis C virus (HCV), hepatitis B surface antigen (HBsAg), malaria, syphilis, etc., that can be transmitted through transfusion from the donor to the recipient [26]. The blood group type (ABO and Rh grouping) of the donated blood units is also determined. The entire process of typing and testing usually takes few hours.

3.4 Storage

The storage conditions play an important role in determining the effective duration for which blood and its components can be used. Typically, the blood collected after donation cannot last for more than 8 hours in room conditions since it is highly prone to bacterial contamination [20]. Due to this technological constraint, the process of component separation has to be completed within a few hours after collection. Once the components have been separated they have to be stored in different conditions as shown in Table 18.1.

In India, the useful life of WB and RBC is considered to be 30 days, which means that WB and RBC units[3] which have not been transfused within 1 month have to be discarded. Similarly, platelets need to be discarded after 5 days if they have not been transfused. Since the life of platelets is very small compared to other components, blood banks typically end up discarding a lot of platelets. Plasma can survive up to almost 1 year in the frozen state. WBCs are usually not transfused. But in those cases where such transfusion is required, WBCs have to be prepared immediately since they cannot be stored.

[3] One unit is equivalent to one blood bag.

3.5 Issuing and Cross-Matching

Whenever a demand for blood unit arrives at the blood bank, the sample of blood from the patient is cross-matched with the samples of blood from various donors of the same blood group type (ABO and Rh) [5, 20]. This is to ensure that no transfusion-related reaction occurs in the patient's body once the donor blood has been transfused [5, 20]. Donor blood is deemed to be fit for transfusion only when there is no indication of any transfusion-related reaction between the donor blood and the patient serum.

A cross-match can be done quickly (15 min) just to check the ABO compatibility or through a long incubation (45 min) to check the compatibility with other antigens [20]. If the donor blood is found to be compatible with patient's serum, the donor blood is reserved (locked) exclusively for that patient for a certain number of days (known as *cross-match release period*) [27]. If the reserved blood is not used within this duration, it becomes available for other patients [27]. Since not all locked units are transfused, a longer cross-match release period can actually increase the shortage and outdate of units at a blood bank.

The flow of blood units in a blood bank is shown in Fig. 18.3.

4 Operational Decision-Making Problems

A blood bank manager faces a wide variety of operational, tactical and strategic decision-making problems [see 27] as shown in Fig. 18.3. The focus of this chapter, however, is to identify the problems that are concerned with operational decision-making at an HBB. The operational decision-making problems that are prevalent in hospital blood banking are described below:

1. *Blood collection:* Blood collection is one of the most important operational decision making problems in the area of blood bank inventory management. Blood collection policies have a significant impact on the overall performance of a blood bank in terms of shortage, wastage[4] and the total cost of blood bank operations. It is important to keep the shortage of blood products under control since the unavailability of blood units may lead to delay in surgeries and, in some cases, even death. Wastage of blood products should be considered as equally serious a problem as shortage. Wastage of blood products needs to be under control since wastage of one unit at a particular blood bank may result in a shortage of one unit at some other blood bank. Also, since the donor who donated one unit today will not be available for donation again for next few months, wasting one unit is also creating a potential shortage of one unit for the next few months.

[4] Terms like wastage, outdate and discards have been used interchangeably throughout the text. They all refer to expiry of blood items once their useful life (shelf life) is over.

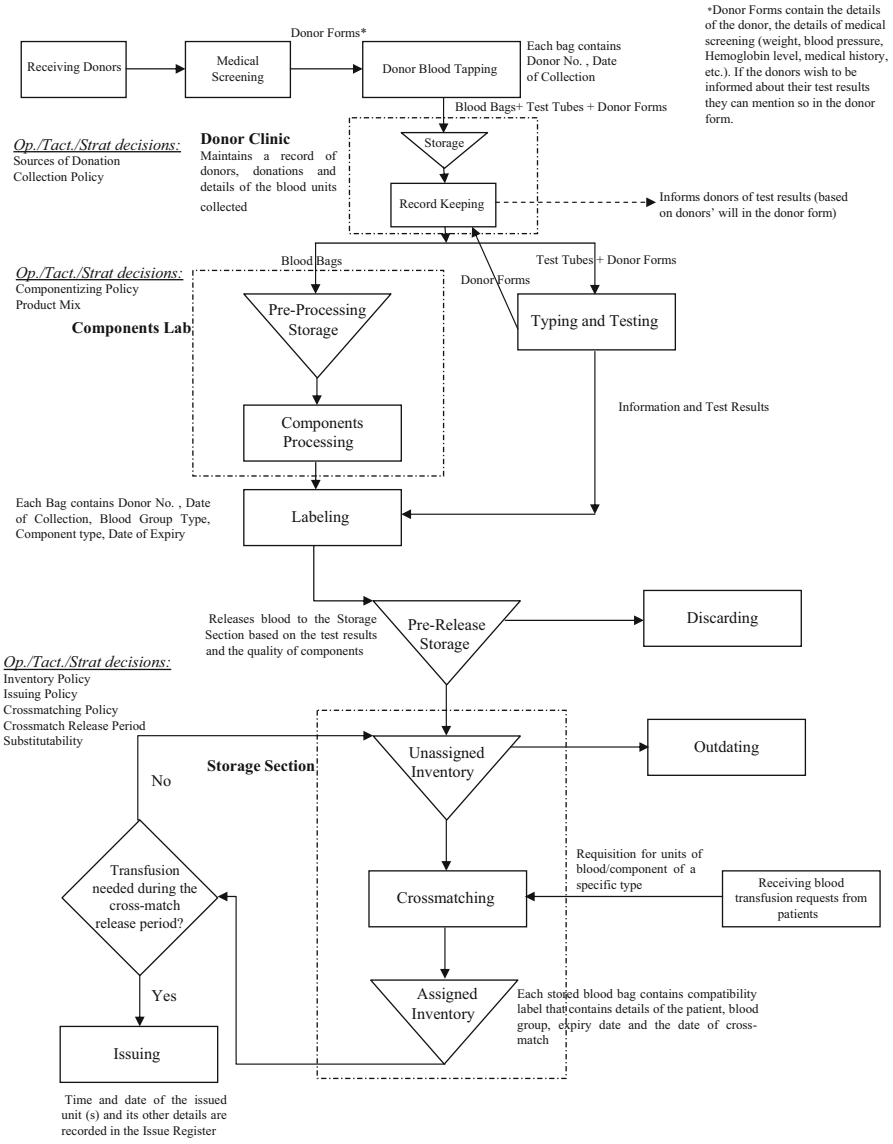


Fig. 18.3 Flow of blood units in a blood bank (Source: Lowalekar and Ravichandran [28, 29])

As mentioned earlier, some small HBBs do not perform their own blood collections by organizing blood donation drives. Instead, they order blood from a larger blood bank at regular intervals. Collection policy in case of such HBBs essentially means determining how frequently should the HBB place orders at the larger blood bank and what should be the ideal order size for a given blood

product every time an order is placed. Again, many HBBs and RBBs do perform their own blood collections by organizing blood donation camps. Collection policy in case of such blood banks essentially means determining how frequently should the blood bank go for blood donation camps and how much blood should be collected during each camp. Collecting too little will result in shortages while collecting too much will result in wastage of blood products.

2. *Blood componentizing:* Componentizing whole blood results in three products which have different demands and lifetimes. The problem of determining the optimal fraction of whole blood to be componentized is, therefore, quite complex. Empty single blood bags (∼300 ml) are used to collect whole blood while empty triple/quadruple bags (∼ 450 ml) are used to collect blood to be componentized. The cost of an empty single blood bag in India is around ₹ 50 while an empty triple bag costs around ₹ 300. The blood from only about half of the donors is suitable for componentizing. Components cannot be prepared once blood has been collected in a single blood bag because there is a risk of contamination while transferring the separated components into other bags. Therefore, it has to be decided even before the camp itself what ratio of single to triple blood bags should be carried to the camp site.
   Componentizing 100 % blood is neither technically feasible nor economically viable for the blood banks. The amount of blood componentized directly affects the cost of blood bank operations. Componentizing too little may lead to shortage while over-componentizing may lead to high costs of purchasing blood bags and holding the inventory of components. Over-componentizing will also result in excessive wastage. The optimal quantity of whole blood to be componentized is a function of the costs of processing, shortage and wastage of blood and its components.
3. *Cross-matching policy:* The blood bank manager needs to decide the optimal length of cross-match release period since it can have a major impact on the shortage and wastage in the system. It has been seen that the number of units cross-matched in hospitals is much higher than the number of units actually transfused to the patients. This happens because the doctors order more number of units on behalf of patients, just to be on the safer side. When the units in the reserved units in the inventory are not eventually transfused during the specified period, they are unreserved and made available to other patients in the free inventory. When the cross-match release period is large, the units are locked in the assigned inventory for a longer time after cross-matching. The units which are returned to the inventory are older and, therefore, prone to outdates if they are not transfused before their useful life is over. If the doctors and patients have a tendency of ordering more than what is required, there will be a lot of wastage when the cross-match release period is large.

It can be seen that the process of operational decision making is quite complex in case of blood banks due to the presence of following factors:

- There are a large number of products due to various blood groups (eight major blood groups).
- A single unit of whole blood can be broken down into various components (four major components). Thus, there are theoretically 32 different products in a blood bank at any given point in time.
- Blood and its components are all perishable in nature and have different useful lifetimes. Due to the perishable nature of the blood components there will be multiple batches of different ages for every blood product at any point in time.
- The order in which the units are issued (First In, First Out (FIFO)), Last In, First Out (LIFO), random order, etc.) may deplete the existing stock of a given blood product in a particular fashion. Even though it is preferable to issue blood units in FIFO manner (to minimize the wastage), the units may not actually get issued as per FIFO[5]. The shortage and the wastage of blood products depend upon the order in which the units are issued from the inventory.
- The supply of blood from a blood donation camp is unknown beforehand. The blood bank may not be able to collect the planned number of units, or it may end up collecting more than the planned number of units during the camp.
- The actual yield of blood products of various blood group types cannot be known during the collection. Only after the units have been typed and tested in the blood bank laboratory is it possible to know exactly how many units of each of the blood group type were added during the camp.
- The demand for the blood products of various blood group types is stochastic. There also exists some substitutability[6] among blood groups and components.
- The cost of shortage and wastage of blood and its components is usually not explicit. In some cases, the result of one shortage can even mean loss of human life. Similarly, the cost of wastage can be very high if the blood bank ends up wasting one unit of blood when the overall supply of blood is extremely low.

5 Blood Bank Inventory Management: A Real-Life Case Study

For the purpose of the case study we chose a not-for-profit blood bank operating in the southern part of India. The blood bank operates within the premises of a charity hospital. It provides blood products to the patients from the same hospital as well as other hospitals operating in the region. The blood bank organizes about 150 camps per year and its annual blood collections amount to approximately 13,000 units per year. The blood bank is equipped with the state-of–the-art facilities for preparing

[5] Some surgeries require fresh blood and in such cases the oldest unit cannot be transfused to the patient. In some cases the patient blood may not be compatible with the oldest unit in the inventory.
[6] For example, whole blood of the same blood group can be given after compatibility testing, when the required component is not available. Also, after fractionation, platelets of any blood group type can be given in case of emergency when the platelets of the required blood group are not available. This is because most of the antibodies, which are responsible for transfusion reactions, remain in the plasma after the fractionation.

Table 18.2 Annual blood collection. (Source: Lowalekar and Ravichandran [25])

Average annual collection from blood donation camps (units*)	12,931
Average annual collection from blood bank donations (units)	575
Average annual collection (camps + blood bank donations)	13,506
% collection from blood donation camps	95.74 %

*One unit is equivalent to one blood bag. Blood is collected from donors in the bags of 300 and 450 ml capacities. A maximum one unit of blood is collected from a donor depending upon the health of the donor. Blood is not separated by gender of the donor. Blood collected from one donor is not mixed with the blood of other donors even if they are of the same blood group type. The difference between the capacities of various blood bags (such as 300 and 450 ml) has been ignored for the rest of the chapter.

components. The subsequent sections describe the data[7] regarding the supply and demand for various blood and components at the blood bank.

5.1 Information Regarding Supply

The main source of collection for the blood bank is blood donation camps that are organized in the various parts of the city. The number of donors who come to donate blood at the blood bank instead of camps is relatively small. All donations at the blood bank are voluntary nonpaid donations. The break-up of the average annual blood collections at the blood bank is shown in Table 18.2.

A typical blood donation camp lasts for 3–4 hours Even though the blood bank tries to organize one blood donation camp every 2 days, the actual time between two consecutive camps varies due to a variety of reasons. The actual number of donors attending these camps is also stochastic and cannot be predicted in advance (see Table 18.3). The distributions of the three supply variables at the blood bank[8] are shown in Fig. 18.4.

Table 18.4 shows the percentage composition of various positive blood groups in the blood obtained through blood donation camps. The negative blood groups constitute only 6 % of the total blood collections. The percentage of blood items discarded due to various failed screening tests (HIV, HCV, HBV, VDRL, malaria, etc.) along with the poor donations amount to roughly 2 % of the total blood collections at the blood bank.

[7] The data organization at the blood bank under study was quite poor due to absence of computerization. The data entry into various registers was performed manually at the blood bank. The data presented in the chapter, therefore, is presented as was made available to us by the blood bank officials in the Excel sheet "Blood_Bank_Case_Study_DATA.xls"

[8] All probability distributions are based on the past data at the blood bank. The data and the distributions will be different for different blood banks.

Table 18.3 Supply variables

Serial number	Random variable	Average and standard deviation	Distribution
1	Time between two successive camps (days)	Average = 2.359 Std. Deviation = 2.22	Empirical[a]
2	Donations at a camp (units)	Average = 83.57 Std. Deviation = 53.09	Empirical
3	Daily donation at the blood bank (units)	Average = 1.574 Std. Deviation = 2.31	Geometric (0.3883)[b]

[a] Empirical distribution is in the form of a histogram created from the past data. Empirical distributions are used to describe random variables when commonly known distributions do not properly fit the historical data. All empirical distributions described in the case study have been given in the Excel file "Blood_Bank_Case_Study _DATA.xls" in the form of probability distributions and histograms.

[b] The number mentioned in the bracket represents the parameter of the geometric distribution. For more details on geometric distribution please refer to the web page: http://en.wikipedia.org/wiki/Geometric_distribution.

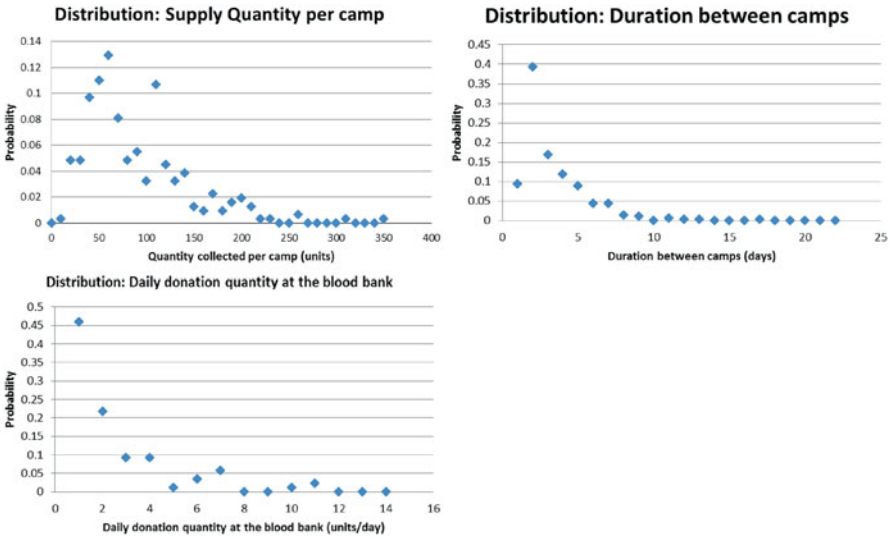


Fig. 18.4 Distributions of supply variables (Source: Lowalekar and Ravichandran [28])

The present policy at the blood bank is to accept blood from all medically fit donors who come to the blood donation camps (*unrestricted collection policy*). Typically the blood to be fractionated into components is collected in empty triple bags of 450 ml capacity. The blood bank componentizes 35–50 % of the fresh blood. The remaining blood is collected in empty single bags and stored as whole blood.

Table 18.4 Percentage composition of positive blood groups

Serial number	Parameter	Values				
1	Percentage composition of blood group "G" in the population	Groups %	A+ 21.19	B+ 29.24	O+ 37.09	AB+ 6.44
2	Percentage of failed screening tests and poor donations	1.99 %				

Table 18.5 Characteristics of daily demand (units/day)

		Blood Group			
		O+	B+	A+	AB+
Whole blood (WB)	Average	7.84	4.52	4.28	1.67
	Standard deviation	5.28	2.92	3.56	2.00
	Distribution	Empirical	Empirical	Empirical	Geometric (0.375)
RBC	Average	3.75	3.01	2.44	0.65
	Standard deviation	3.36	2.65	3.02	0.95
	Distribution	Empirical	Geometric (0.249)	Geometric (0.291)	Geometric (0.606)
Fresh frozen plasma (FFP)	Average	1.24	0.91	0.65	0.09
	Standard deviation	1.73	1.64	1.24	0.63
	Distribution	Empirical	Empirical	Empirical	Empirical
		All Groups			
Platelets	Average	7.35			
	Standard deviation	7.28			
	Distribution	Empirical			

5.2 Information Regarding Demand

The averages, standard deviations and distributions of daily demand for blood and its components for various blood groups have been shown in Table 18.5. Since platelets usually do not carry any blood group charge, the consolidated demand for platelets for all blood groups has been shown. Also, since the percentage of negative blood groups is extremely low we have neglected them for the rest of the chapter. The distributions for the daily demand of whole blood, RBC and fresh frozen plasma (FFP) for one blood group (O+) are shown in Fig. 18.5. The distribution of daily demand for platelets (all groups) is shown in Fig. 18.6. Cryoprecipitate is also manufactured at the blood bank through plasma fractionation in very small quantities (< 300 units/year), to be given away free to hemophilia patients. Since the amount of cryoprecipitate produced at the blood bank is very small we have neglected it for the rest of the chapter.

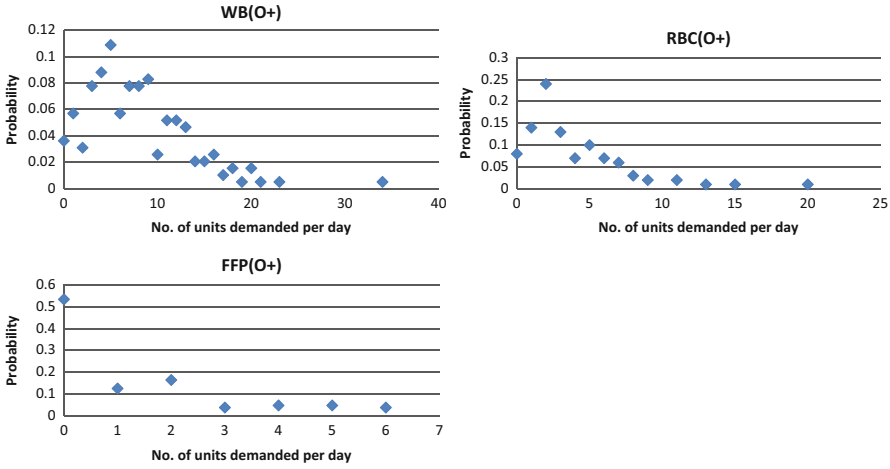


Fig. 18.5 Distribution of daily demand (units/day) for WB, RBC and FFP (O+) (Source: Lowalekar and Ravichandran [28])

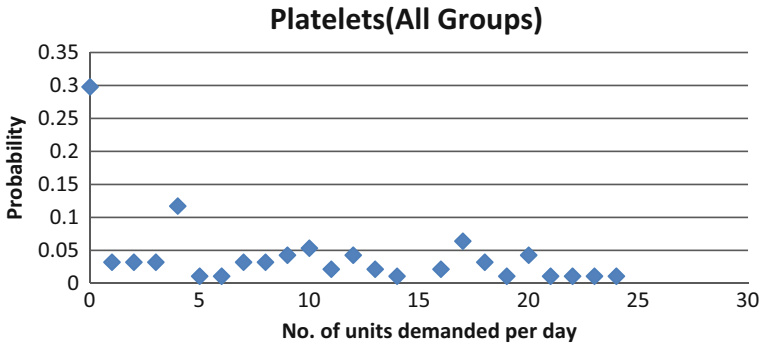


Fig. 18.6 Distribution of daily demand (units/day) for platelets (all groups) (Source: Lowalekar and Ravichandran [28])

5.3 Revenues and Costs

Blood and its components are sold at very reasonable prices to the patients who need transfusion, by the blood bank. WB and RBC are sold at ₹ 700 per bag while platelets and plasma are sold at ₹ 600 per bag. Roughly 10 % of the blood at the blood bank is given away free to the below-poverty-line patients. The fixed cost structure (annual) at the blood bank is shown in Table 18.6, while the variable cost structure (per bag) at the blood bank is shown in Table 18.7.

Table 18.6 Annual fixed costs

Fixed cost	Rupees
Establishment (inclusive of staff salary, allowances, incentives)	2,772,500
Administrative expenses (office supplies, telephones, miscellaneous expenses)	350,000
Repair and maintenance	450,000
Ambulance and disposables expenses	250,000
Camp expenses	700,000
Provision for capital goods	1,000,000
5 % contingency	220,000
Total fixed cost	5,742,500

Table 18.7 Variable costs

Cost	Single Bag (₹)	Triple Bag (₹)
Donor form	0.5	0.5
Hemoglobin (Hb) testing	25	25
Empty blood bag	53	280
Donor diet	15	15
Certificate	6	6
Testing	160	160
Total	259.5	486.5

5.4 Decision Problems at the Blood Bank

The blood bank manager needs to identify optimal policies for blood collection and componentizing. If it is assumed that the frequency of blood donation camps cannot be changed[9], then the operational decision problems for the blood bank can be summarized as:

1. How much blood should be collected during blood donation camps?
2. How much quantity of the fresh blood should be componentized after each camp?

The number of blood units collected during a blood donation drive is a random variable. The actual blood collection during a blood donation camp may differ from the planned collection. Due to the fact that there is usually a shortage of blood items and that the demand of blood units cannot be predicted accurately beforehand, blood banks tend to collect blood from all the donors who come to the donation camp. The tendency to collect blood from all donors stems from their assumption that by collecting one more unit of blood they will be able to save one more human life. This policy, however, results in huge wastage of blood units during the cycles of high collection and low demand.

A cutoff-level policy can be a useful policy in situations where supply quantity is a random variable. Under a cutoff-level policy the blood collection is stopped as

[9] The number of donation camps a blood bank can organize depends on the level of resources at the blood bank and the permission from the various camping sites. It is usually difficult to change the camping frequency in the short run.

soon as the collected quantity reaches the cutoff level. There are two ways in which a blood bank can implement a cutoff-level policy for collection:

a. It can turn away the donors who come to donate blood once the cutoff level is reached.
b. It can collect blood from all donors who come to donate blood and give the excess blood (beyond the cutoff level) to other blood banks that have shortfall in supply.

It is quite possible (due to the randomness in the actual collections) that the blood bank may not be able to collect the desired amount by the end of a given camp.

For the purpose of the case study, it was assumed that the blood bank organizes camp after an interval of T days[10]. The three blood collection policies [28] that were defined for the purpose of this study are discussed below:

1. Unrestricted collection (UC) policy: Blood bank collects blood from all the healthy donors who come to the blood donation camp. It does not turn away any healthy donor.
2. Cutoff-level policies: The two cutoff-level policies for blood collection under random supply environment are described below:
   1. Modified (Q, T) policy: The blood bank tries to collect a fixed quantity (Q units) of blood during each donation camp and stops collection as soon as the desired quantity is collected. In those cases where the desired amount of blood could not be collected, the blood bank returns with just the amount collected.
   2. Modified (R, T) Policy: The blood bank tries to collect an amount which equals the difference between a predetermined target inventory level (R units) and the stock level of blood units (WB + RBC) immediately before the camp. It stops collection as soon as the desired quantity is collected. In those cases where the desired amount of blood could not be collected, the blood bank returns with just the amount collected.

The blood collection policy cannot be looked at in isolation. The optimal amount (percentage) of blood that needs to be componentized is an equally important policy decision since the performance of the system changes dramatically with the amount of blood componentized. If the level of componentizing is too small at a blood bank it would need to increase its collections in order to counter the shortages. On the other hand, if the level of componentizing is too large, it would incur larger costs of processing, separating and holding the components in its inventory. Therefore, it becomes important to take into consideration the level of fractionation while deciding the blood-collection policy for a blood bank.

5.5 Model for Blood Bank Inventory Management

Since blood is a perishable item, several authors have modeled the blood bank inventory problem as a perishable inventory system with lot of simplified assumptions.

[10] Refer to Fig. 18.4 for the distribution of T (interval between two successive camps).

The analytical modeling of perishable systems, however, becomes complex as the size of the problem becomes large [27]. If one were to model the blood bank system as a perishable inventory system, one needs to know the stock position of batches of items of various ages that exist in the system at any point in time in order to calculate the wastage and shortage for a given demand distribution. For example, if blood bank goes for a camp every 2 days there will be 15 different batches of different ages for a given blood product and one will need a state vector of 15 variables in order to calculate the average shortage and wastage in the system. There are 32 such perishable products (8 blood groups, 4 components) with different lifetimes and this makes the mathematical model intractable. The analytical modeling of blood bank system is further complicated due to several other factors as discussed in Sect. 18.4. Therefore for the purpose of this case study, we use simulation as a tool for modeling the blood bank inventory system.

The simulation model is developed using C language (see Fig. 18.7). The run length for the simulation experiment is taken as 1000 years with the total number of iterations for every run as 30. The various performance measures are reported on a per year basis.

5.5.1 Simulation Model Assumptions

The simulation model is developed based on the following assumptions:

- The percentage composition of the various blood groups in a given collection is assumed to be constant (and same as Table 18.4).
- Only the positive blood group types ($\sim 94\,\%$) are considered for the integrated model.
- The percentage of units failing the screening tests is assumed to be constant ($\sim 2\,\%$).
- Only four items are considered for the simulation model: WB, RBC, plasma and platelets. Cryoprecipitate is not considered due to its extremely low production as compared to the other items.
- WB and RBC (of the same blood group) are assumed to be perfectly substitutable[11]. The performance measures (shortage, wastage, fill rates[12], etc.) for WB and RBC are evaluated jointly (WB + RBC).
- No intergroup substitutability is assumed for WB, RBC and plasma.
- Platelets of the various groups are assumed to be perfectly substitutable among each other.

[11] In cases where a unit of RBC is not available, WB of the same blood group can be usually given after minor processing which removes the platelets and plasma from WB. In the cases where WB is not available, RBC can be given (along with certain fluids if required) to the patient.

[12] *Fill rate is defined as the percentage of demand satisfied.* A fill rate of 80 % means that only 80 % of the demanded items were issued from the inventory. Alternatively, it means that 20 % of the demand was not fulfilled.

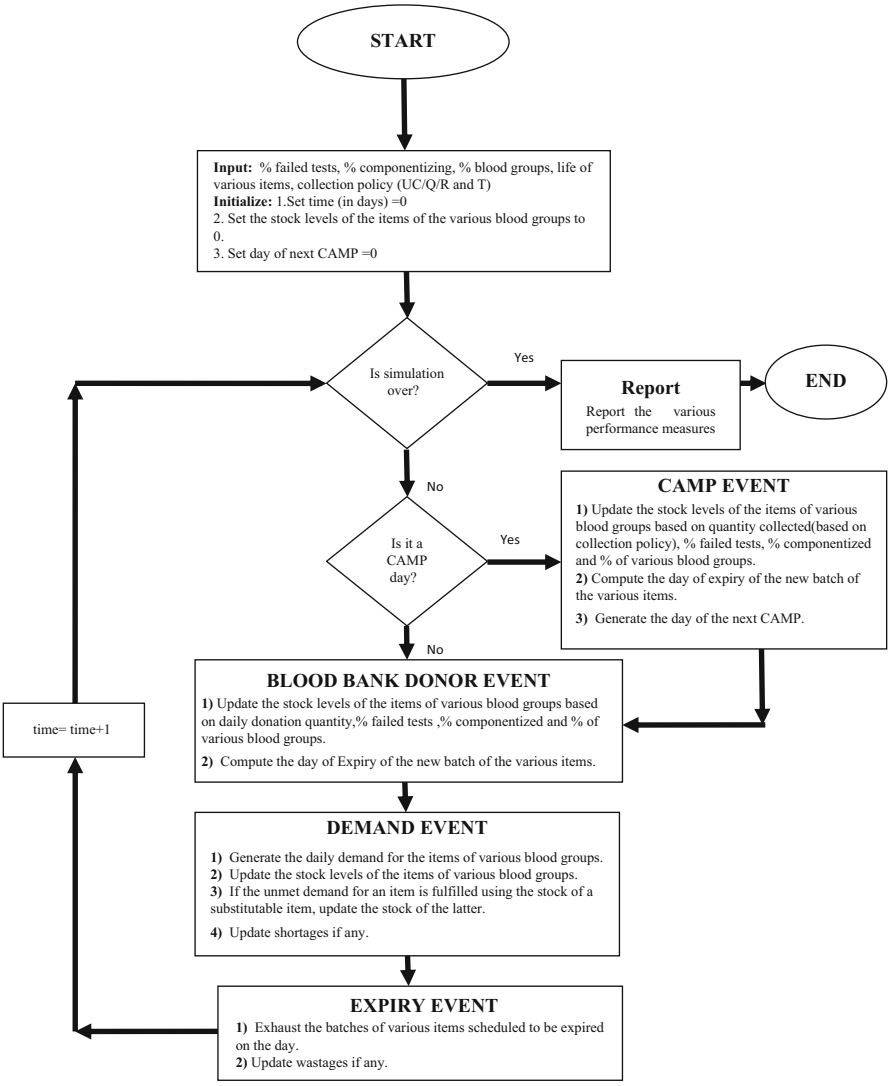


Fig. 18.7 Simulation model (Source: Lowalekar and Ravichandran [25])

- It is assumed that the duration of cross-matching is negligible. It is also assumed that all the blood units of a given blood group type are perfectly safe to be given to a patient of the same blood type[13].

[13] Except for a few cases, the donated blood units of a given blood type are usually compatible with the patient blood of the same blood type (based on the experience of the blood bank chosen for the study).

Table 18.8 Random variables

Serial number	Random variable	Notation
1	Time between two successive camps (days)	T_C
2	Donations at a camp (units)	S_C
3	Daily donation at the blood bank (units)	S_{BB}
4	Daily demand for the item "i" of blood group "g"	D_{ig}

Table 18.9 Model parameters

Serial number	Parameter	Notation
1	Percentage composition of blood group "g"[a] in the population	$Perc_g$
2	Percentage(%) of failed screening tests	perc_fail

[a]The symbol "g" is used throughout the rest of the chapter to refer to a general blood group (O+, A+, B+ and AB+). Similarly, "i" is used to refer to a general blood product (WB, RBC, platelets and plasma) while "n" is used to refer to a batch of a given blood product.

Table 18.10 Performance measures

Serial number	Measures	Description
1	Item fill rate and wastage	Obtained from the profile curves[a]
2	Total annual cost (cost of shortage and wastage of blood products, cost of purchasing blood bags and holding blood products in the inventory)	Obtained from the cost curves

[a]The terms profile curve and cost curve are explained in the next section.

- The blood units are issued in the order in which they are collected (FIFO issuing).
- The holding costs of inventory are assumed to be ₹ 25 per unit per year independent of the type of the component. The costs of the collecting blood[14] are assumed to be ₹ 270 for an empty single bag and ₹ 500 for an empty triple bag.
- The probability distributions of the demand and supply variables are assumed to remain same for the entire period of simulation (Tables 18.3, 18.5 and Figs. 18.4, 18.5 and 18.6).

5.5.2 Model Specifications

The random variables, model parameters, state variables and performance measures for the simulation model are shown in Tables 18.8, 18.9, 18.10 and 18.11.

[14] The cost of collecting blood includes the cost of empty blood bag along with the costs of donor form, Hb testing, donor diet, certificate and testing. As per Table 18.7, the total cost of collecting blood (variable) in an empty single blood bag is ₹ 259.5 while that for an empty triple bag is ₹ 486.5. For simplicity we have assumed these costs to be ₹ 270 and ₹ 500, respectively.

Table 18.11 State variables

Serial Number	Variable	Description
1	Systime	Current simulation time (in days)
2	Stock[n][i][g]	Stock level of the n^{th} batch of item "i" of blood group "g"
3	N[i][g]	Total number of batches of item "i" of blood group "g"
4	Texpiry[n][i][g]	The time of expiry of the n^{th} batch of item "i" of blood group "g"
5	Tcamp	The time of next blood donation camp

Table 18.12 Decision variables

Serial number	Variables	Values
1.	Cutoff Level (Q units[a]) (for Modified (Q, T) Policy)	20,40,60,80,100,120,140, 160, 180,200,220,240,260,280
2.	Cutoff Level (R units) [for Modified (R, T) Policy]	200,300,400,500,600,700,800, 900,1000,1100,1200,1300,1400
3.	% Componentizing [for all the three policies]	0 %, 5 %,10 %,15 %,20 %,25 %, 30 %,35 %,40 %,45 %,50 %

[a]One unit is equivalent to one blood bag

5.5.3 Decision Variables

The simulation model reports the performance of a given policy in terms of fill rate and wastage of blood products along with the total cost of operations (on per year basis) at the blood bank. For an unrestricted collection policy the performance measures will vary only with the level of componentizing (since the interval between two consecutive camps "T" is assumed to be given). The performance of the two cutoff level policies (modified (Q, T) and modified (R, T)) will vary with the cutoff level (Q or R) as well as the level of componentizing. In order to find a good solution using the simulation model, we take multiple values of the decision variables (% componentizing, Q or R) under each policy and measure the system performance in terms of fill rate, wastage and total annual cost. The values of decision variables chosen for the simulation experiment are shown in Table 18.12.

5.5.4 Model Logic

The basic structure of the simulation model (discrete-event) is shown in Fig. 18.7. For the purpose of the simulation, four different events were defined as follows:

a. Camp Event
b. Blood Bank Donor Event
c. Demand Event
d. Expiry Event

At the start of every day the model checks if it is a camp day or not (based on the random variable T_C). If it is a camp day, a certain number of units are randomly

generated (corresponding to random variable S_C). Based on the collection policy, the percentage of componentizing, the composition of blood groups and the percentage of failed screening tests the stock levels of the various blood groups are updated accordingly. The date of expiry of new batch of each of the blood products is also recorded. Similarly, the stock levels of various blood products due to the donations at the blood bank (corresponding to random variable S_{BB}) are updated. The stock of various blood products is then reduced appropriately based on their corresponding demand for the day (given by random variable D_{ig}). The shortage for the blood product is updated accordingly if the stock of the given product (and the substitutable product, if any) is not sufficient to meet the demand. The batches of various blood products which are scheduled to expire on the day are then removed from the stock and the stock level as well as the wastage of various blood products is updated. The simulation clock is advanced to the next day and the same cycle is repeated for a given number of days (known as the run-length of the experiment). The average yearly values of various performance measures, as shown in Table 18.10, are reported at the end of the simulation experiment.

5.5.5 Results and Discussion[15]

In order to determine the best levels of collection and componentizing, two different approaches were considered. Under the first approach, the blood bank manager considers only the total wastage and shortage of various blood products and neglects various operational costs. As per the second approach, the blood bank manager considers operational costs as well as the implicit costs of shortage and wastage of blood products while determining the best levels of collection and componentizing. The output of the first approach is a "Profile Curve" while that of the second approach is a "Cost Curve". These two curves along with the approaches are explained below in detail.

Profile Curves: The profile curves depict the variation of fill rate (percentage of demand satisfied) with the wastage of a given item for a given level of collection and componentizing [28]. These curves are useful in identifying the appropriate collection and componentizing parameters in those situations where a blood bank manager perceives shortage and wastage to be more important than the operational costs.

Cost Curves: The major costs that are incurred at any blood bank typically include the costs of collection, processing and storing blood and its components (see Tables 18.6 and 18.7). The costs of shortage and wastage of blood products are implicit in nature. A blood bank would ideally want to minimize the actual costs of running the operations as well as the implicit costs of shortage and wastage in the system.

[15] The results and discussions in this section are based on the data given in Tables 18.3, 18.4, 18.5, 18.6 and 18.7 for the blood bank chosen for the study. Due to the simulation methodology, these results cannot be generalized for other blood banks operating in different settings.

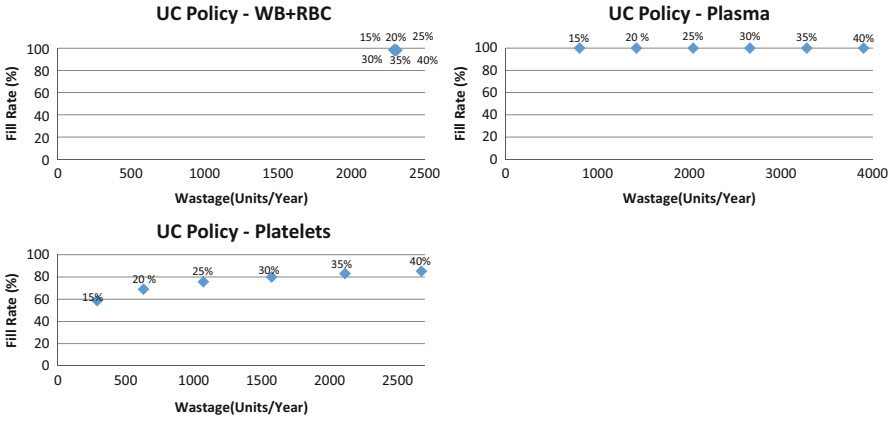


Fig. 18.8 Profile curves-unrestricted collection (UC) policy (Source: Lowalekar and Ravichandran [25])

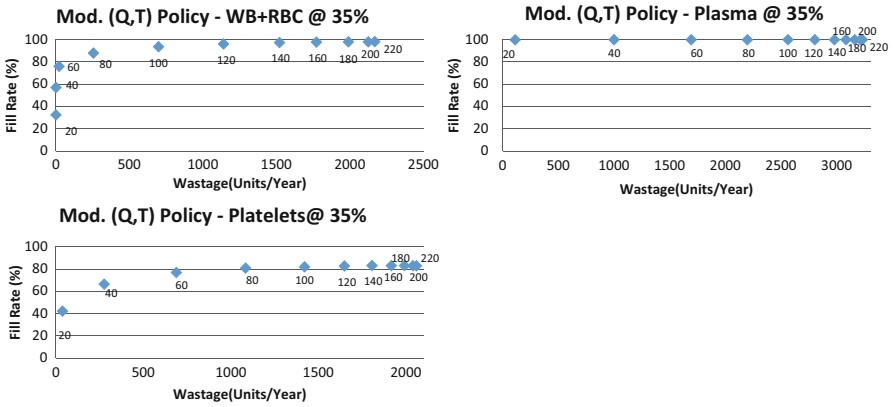


Fig. 18.9 Profile curves-modified (Q, T) policy (@ 35 % componentizing) (Source: Lowalekar and Ravichandran [25])

If the costs of shortage and wastage can be correctly quantified, the blood bank can determine the total cost (includes shortage and wastage costs) for a given level of collection and componentizing. The cost curves depict the variation of total cost at the blood bank with the change in the level of collection and componentizing.

The profile curves for various blood products for the given demand profile at the blood bank have been shown in Figs. 18.8, 18.9 and 18.10. Since the WB and RBC have been assumed to be perfectly substitutable in the model, they have been reported together in the profile curves. The nature of profile curves is similar for various blood products. When the level of collection increases initially, the amount of shortage in the system goes down rapidly, thus increasing the fill rate. But when the collection is increased beyond a certain point, the shortage in the system becomes very low and

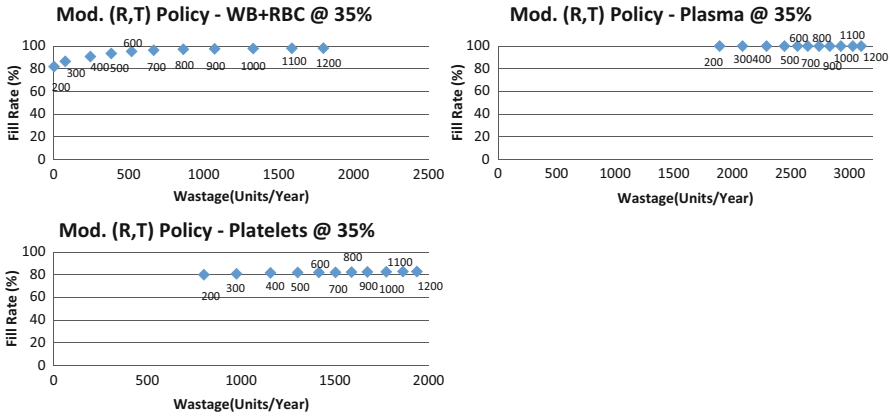


Fig. 18.10 Profile curves-modified (R, T) policy (@ 35 % componentizing) (Source: Lowalekar and Ravichandran [25])

since the collected quantity does not get used, the wastage in the system increases rapidly. Since the demand for platelets is usually high and their shelf life is very low, the fill rates for platelets are the lowest of all the components. On the other hand, since the shelf life of plasma is very high and their demand is very low, the fill rate as well as the wastage of plasma are very high.

Figure 18.8 depicts the profile curves for various blood products under the unrestricted collection (UC) policy at different levels of componentizing. It can be seen that increasing the level of componentizing has no effect on the fill rate and wastage of WB + RBC system under UC policy since they are perfectly substitutable. Increased fractionation will result in production of more RBCs at the expense of WB; the total quantity of WB + RBC remains unchanged. Since the demand for plasma is very low compared to its production, the fill rate for plasma at even 15% componentizing is almost 100%. Increasing the level of componentizing only increases the wastage of plasma. The amount of fresh blood to be fractionated increases the fill rate as well as the wastage of platelets.

Figure 18.9 depicts the profile curves for various products under modified (Q, T) policy at 35% level of componentizing. Increasing the collection quantity (Q) from 20 to 60 units increases the fill rate of WB + RBC without increasing the wastage. Beyond this point, an increase in Q results in an increase in the fill rate as well as the wastage from the WB + RBC system; a similar profile is observed for plasma and platelets. The nature of profile curves will be similar for other levels of componentizing (not shown).

Figure 18.10 depicts the variation of fill rate with wastage for various products, at 35% componentizing, for change in the order-up-to level (R) from 200 to 1200 units. Even though the nature of profile curves for various products is similar to the modified (Q, T) policy, the level of wastage is found to be lower under the modified (R, T) policy for a given fill rate level. The nature of profile curves will be similar for other levels of componentizing (not shown).

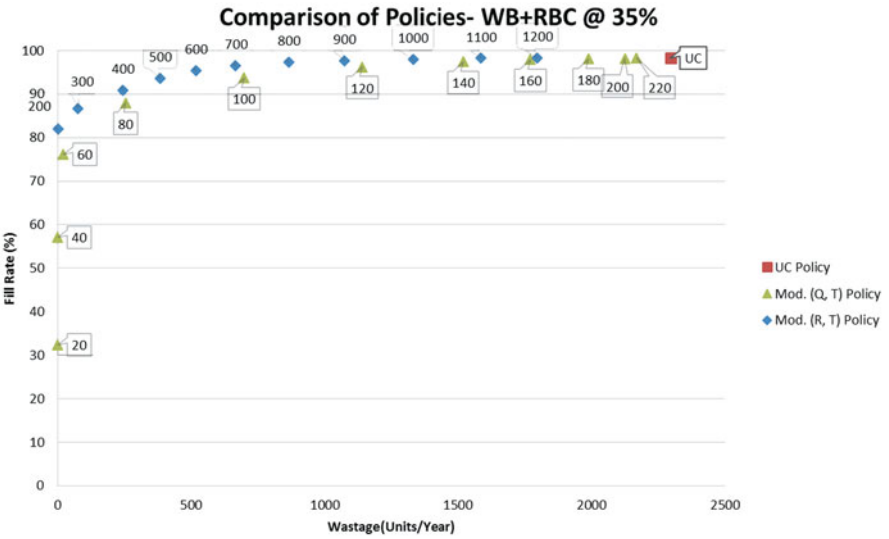


Fig. 18.11 Profile curves-comparison of three policies (WB + RBC @ 35 %) (Source: Lowalekar and Ravichandran [28])

Figure 18.11 depicts the profile curve for WB + RBC system at 35 % level of componentizing for the three collection policies. It can be seen that the profile of a modified (R, T) policy is always above that of a modified (Q, T) policy thus indicating the superiority of the former over the latter. A modified (R, T) policy is better than modified (Q, T) policy since it enables the blood bank in achieving a higher fill rate for a given level of wastage (or lower wastage for a given fill rate). Since the collections are not stopped beyond a certain level (as under modified (Q, T) and modified (R, T) policies) under the unrestricted collection policy, the fill rate as well the wastage of WB + RBC are the highest. It can be seen that UC policy is nothing but the special case where R and Q are infinity.

For the purpose of illustration we show how profile curves can be used to identify the optimal collection and componentizing levels for the three policies at the given blood bank. Let us assume that the minimum fill rate requirement at the blood bank for WB + RBC, plasma and platelets is 95, 95 and 80 % respectively[16]. Using the profile curves obtained from the simulation model the blood bank manager can identify the collection and componentizing levels for the three policies which satisfy the minimum fill rate requirements for various products (see Table 18.13).

The following observations can be made based on Table 18.13 for the demand distributions used:

[16] Since the blood bank needs to ensure the demand from patients is met more often than not, it is assumed arbitrarily for the purpose of illustration that the blood bank would want to satisfy at least 95 % of demand for WB + RBC, 95 % for plasma and 80 % for platelets. The fill rate requirements may be different for different blood banks.

Table 18.13 Fill rate and wastage performance of the collection policies for a given fill rate requirement. (Source: Lowalekar and Ravichandran [25])

Policy	WB+RBC	Plasma	Platelets
UC	$FR_{WB+RBC} = 99\,\%$	$FR_{Plasma} = 100\,\%$	$FR_{Platelets} = 80\,\%$
(@30 %)	$W_{WB+RBC} = 2250$ units	$W_{Plasma} = 2700$ units	$W_{Platelets} = 1550$ units
Mod. (Q, T)	$FR_{WB+RBC} = 95\,\%$	$FR_{Plasma} = 100\,\%$	$FR_{Platelets} = 83\,\%$
(Q=120 @35 %)	$W_{WB+RBC} = 1200$ units	$W_{Plasma} = 2700$ units	$W_{Platelets} = 1600$ units
Mod. (R, T)	$FR_{WB+RBC} = 95\,\%$	$FR_{Plasma} = 100\,\%$	$FR_{Platelets} = 82\,\%$
(R=600 @35 %)	$W_{WB+RBC} = 600$ units	$W_{Plasma} = 2550$ units	$W_{Platelets} = 1400$ units

Minimum Fill Rate Requirement: $FR_{WB+RBC} \geq 95\,\%$, $FR_{Plasma} \geq 95\,\%$, $FR_{Platelets} \geq 80\,\%$
FR denotes the Fill rate (%) while W denotes the wastage for an item

- Under the unrestricted collection (UC) policy, the ideal level of componentizing would be around 30 %. The level of wastage of the various blood products is, however, quite high under this policy.
- Under the modified (Q, T) policy, the blood bank should ideally target collecting 120 units of fresh blood during each donation camp and componentize 35 % of the fresh blood. The wastage of blood and its components is significantly lower when compared to the UC policy.
- A modified (R, T) policy is the best policy among the three since it can help the blood bank achieve minimum level of wastage for blood and its components for the given fill rate requirements. It can be seen that the blood bank should set its order-up-to level (R) as 600 units and keep the level of componentizing at 35 %. If the blood bank follows this policy, it will be able to reduce the annual wastage of WB + RBC by 1650 units, plasma by 250 units and platelets by 150 units platelets as compared to the unrestricted collection policy, for the given fill rate requirements.

An alternate approach would be to quantify the cost of shortage and wastage of one unit of blood and its components and find the level of collection and componentizing which would minimize the total cost of collection, processing, holding, shortage and wastage of blood units at the blood bank. Since different blood banks may perceive the costs of shortage and wastage for blood products differently, Table 18.14 illustrates the optimum levels of collection and componentizing assuming different values of shortage and wastage costs.

Fig. 18.12 depicts the variation of total annual cost with a change in the level of collection and componentizing. For a UC policy, the total annual cost decreases with the increase in level of componentizing up to a certain point and then increases again. For a given level of componentizing, the total annual cost decreases with an increase in collection up to a certain point and then increases again under modified (Q, T) and modified (R, T) policies. The initial decrease in the total cost with the increase in collection (or componentizing) is due to reduction in the shortage in the system. Beyond a certain point (as in profile curves), the increase in collection does not result in a significant reduction in the shortage but an increase the wastage in the system hence increasing the total cost. For the purpose of illustration

Table 18.14 Cost comparison (multiple scenarios) (Source: Lowalekar and Ravichandran [25])

	Shortage Cost/Unit (₹)			Wastage Cost/Unit (₹)			UC		Mod. (Q,T)			Mod. (R,T)		
S.No.	WB/RBC	Plasma	Platelets	WB/RBC	Plasma	Platelets	% Comp.[a]	Total Annual Cost (₹)[a]	% Comp.[a]	Q[a]	Total Annual Cost (₹)[a]	% Comp.[a]	R[a]	Total Annual Cost (₹)[a]
1	1000	1000	1000	1000	1000	1000	10	8,223,301	15	80	6,387,816	10	600	6,062,578
2	1000	100	500	1000	100	500	10	7,255,892	10	80	5,344,310	10	600	5,099,336
3	500	100	500	500	100	500	10	6,032,501	15	80	4,615,251	15	300	4,499,857
4	500	100	300	500	100	300	5	5,689,018	10	80	4,255,273	10	300	4,147,060
5	1000	1000	1000	500	500	500	10	6,956,645	15	100	6,025,500	15	600	5,754,648
6	1000	100	500	500	50	250	10	6,090,114	10	100	5,125,881	10	600	4,854,677
7	500	100	500	250	50	250	10	5,434,904	15	80	4,515,206	15	400	4,451,295
8	500	100	300	250	50	150	10	5,120,452	10	80	4,190,422	10	400	4,131,337
9	1000	1000	1000	300	300	300	10	6,449,983	15	100	5,760,775	15	700	5,525,932
10	1000	100	500	300	30	150	10	5,623,803	15	100	4,974,386	15	600	4,733,124
11	500	100	500	150	30	150	10	5,195,865	15	80	4,475,188	15	400	4,410,626
12	500	100	300	150	30	100	10	4,885,510	10	80	4,164,720	10	500	4,104,098
13	1000	100	500	2000	200	1000	10	9,587,448	10	80	5,598,182	10	300	5,310,332
14	1000	1000	1000	0	0	0	25	5,339,681	25	120	5,098,647	25	700	4,909,706
15	2000	2000	2000	0	0	0	30	6,084,976	35	140	5,943,255	35	800	5,741,108

[a]The best value obtained from simulation model

(see Fig. 18.12) we assume the shortage and wastage cost arbitrarily[17] for various blood products. We then determine, using the cost curves, the optimum levels of collection and componentizing for the three policies. The following observations can be made based on Fig. 18.12:

- A level of componentizing of 10 % will be ideal for the blood bank if it follows unrestricted (UC) policy. The total annual cost at the blood bank corresponding to 10 % componentizing is approximately ₹ 80 lakhs[18].
- The blood bank should try and collect 80 units of blood each time it goes for a donation camp and fractionate 20 % of the blood if it follows a modified (Q, T) policy. The total annual cost corresponding to Q = 80 units and level of componentizing at 20 % is approximately ₹ 60 lakhs.
- The blood bank should follow[19] a modified (R, T) policy since it results in the minimum total annual cost (approximately ₹ 58 lakhs) among the three policies. The blood bank should set the order up to level at 500 units and fractionate 20 % of the collected fresh under this policy.

6 Recommendations for Real-World Implementation

It can be clearly seen from the case study of the Indian blood bank that the policy of collecting all the available blood supply during a camp is not necessarily the best one for blood banks. It results in high costs of operation and leads to excessive wastage in the system. It was also shown how stopping the supply beyond a certain level does not increase the shortage but definitely reduces the wastage in the system.

It can also be argued from the case study that the policy of fractionating all the available blood, as recommended by doctors, may not be the best policy for blood banks. Fractionating fresh blood beyond a certain level only leads to increased costs of processing, holding and wastage of the blood products.

The inventory management problem discussed in the chapter considered the planning horizon of 1 year for deciding the best levels of collection of collection and componentizing at the blood bank. In some cases, the planning horizon may be quite small (such as a week or a month) and the blood bank needs to decide how much blood it should plan to collect in the next period (a period can be a convenient planning horizon like a week or a month). In such cases, the blood bank can estimate the mean and the standard deviation of the demand of the various blood products

[17] For the purpose of cost curves shown in Fig. 18.12, the cost of shortage per unit was assumed to be ₹1000 for WB/RBC, ₹100 for plasma and ₹1000 for platelets. The cost of wastage per unit was assumed to be ₹1000 for WB/RBC, ₹100 for plasma and ₹1000 for platelets.

[18] One lakh is equivalent to one hundred thousand (100,000)

[19] It is worth mentioning here that the modified (R, T) policy is not the optimum policy for blood collection for the blood bank. It only performs the best among the three collection policies discussed in the chapter. The blood bank may actually come up with a superior policy for blood collection and test its performance using the simulation model described in this chapter.

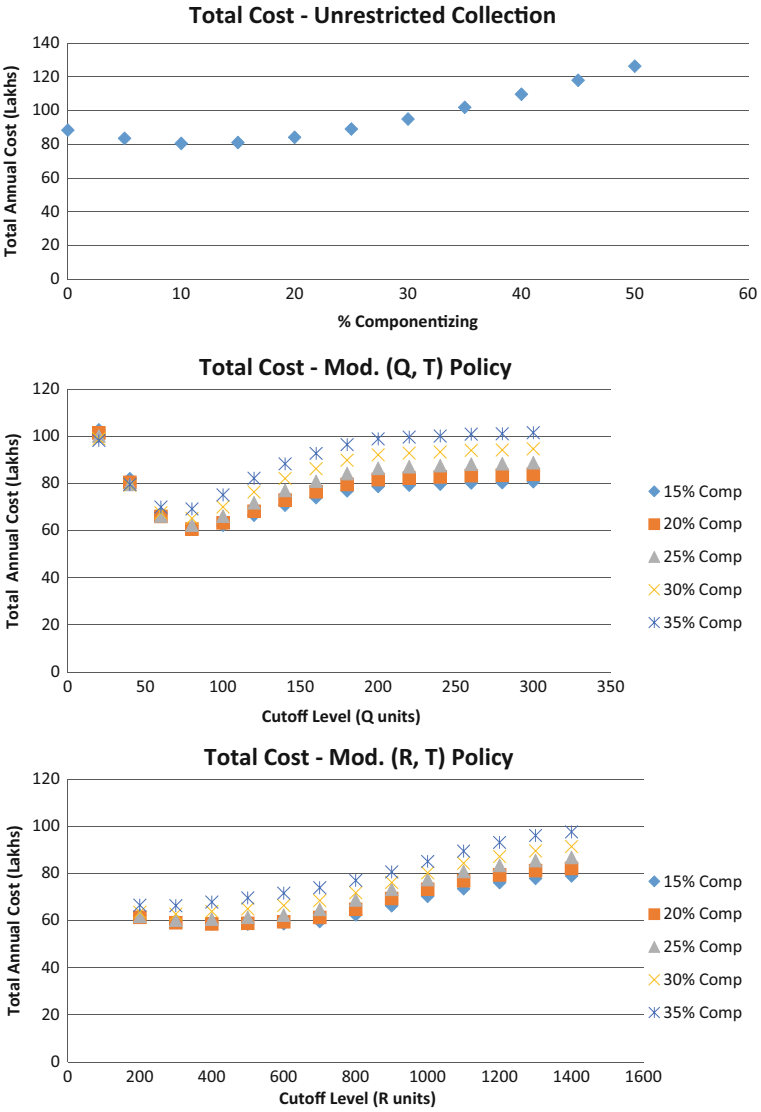


Fig. 18.12 Total cost curves (Source: Lowalekar and Ravichandran [25])

based on the recent data. The blood bank can develop a forecast for the expected demand of each blood product for the next period by some method like the moving average method or some other method. The target level of blood product to be kept during the next period can be then computed adding a factor of safety (1.5 or 2 standard deviation of demand/period of the blood product) to the expected demand. The actual quantity of blood product to be manufactured can be computed by taking

the difference between the target level and the on hand stock of the blood product at the start of the period. Once the estimates of amounts of various blood products are available, the blood bank can estimate the amount of blood to be collected and componentized for the next period[20].

Since blood bank situations are too complex for analytical modeling, simulation as a tool can be extremely powerful for operational decision making in blood banks. Simulation models can be an integral part of the Decision Support Systems (DSS) at the blood banks and they can help a blood bank manager to identify the best levels of collection and componentizing.

7 Exercises

Q.1 The requirement of components of various positive blood groups (in units[21]) at a blood bank on a given day is given below:

Products	Blood Groups			
	O+	A+	B+	AB+
WB	7	9	5	3
RBC	5	3	2	1
Platelets	3	4	3	2
Plasma	1	2	1	3

It can be assumed for the purpose of this exercise that the blood bank can arrange any number of donors of a given blood group type for donation at the start of the day. Also, it can collect blood from such donors in empty single bags (for WB) or empty triple bags (for components) depending on its requirement. Componentizing one unit of fresh blood produces one unit each of RBC, platelets and plasma. The cost of an empty single bag is ₹ 50 while the cost of an empty triple bag is ₹ 300. It can be assumed that WB and RBC of the same blood group type are perfectly substitutable (in case one is not available the other of the same blood group can be given to the patient). The platelets of all the four blood group types are also substitutable. The demand for all the components of various blood group types must be met. If the objective of the blood bank is to minimize the total cost of blood bags (single and triple) for the given day, how many donors[22] of each blood type should be called for donating blood and how many units of fresh blood of each blood type should be separated into components at the start of the day? (Since the planning horizon is only 1 day you can neglect the perishable nature of blood and its components for the purpose of this problem.)

[20] See Q.1 in the exercises given at the end of the chapter.
[21] One unit is equivalent to one blood bag.
[22] Hint: Use linear programming.

Q.2 Assume that the blood bank in the previous problem can arrange any number of donors for donating blood but the blood groups of these donors is not known beforehand[23]. However, the percentage composition of various blood groups in any collection remains constant as shown below (For example, in every 100 units collected 40 units are of type O+, 30 units of type A+, etc.):

	O+	A+	B+	AB+
%	40	30	20	10

If everything else remains same as Q.1, how many donors should be called for donating blood and how many units of fresh blood of each blood type should be separated into components at the start of the day? (Neglect the integrality constraint arising out of percentage composition of various blood groups in a given collection, for the purpose of this problem.)

Q.3 In Q.2 assume that the blood bank need not satisfy the demand of all the components of various blood groups. The selling price of one unit of WB, RBC, platelets and plasma are ₹700, ₹700, ₹500 and ₹200, respectively. If everything else remains same as Q.2, how many donors should be called for donating blood and how many units of fresh blood of each blood type should be separated into components at the start of the day? The objective of the blood bank in this case is to maximize the profit by selling blood and its components during the day.

Q.4 A blood bank collects blood at the start of every month to be stored as whole blood (WB) in its inventory. The cost of one blood bag is ₹200 while the selling price of one unit of WB is ₹500. The shelf-life of WB is exactly 1 month. Each time the blood bank goes for collection, the unused units of WB from the previous month, if any, need to be discarded. If the number of units of WB demanded monthly demand follows a Poisson distribution with an average of 50 units, what is the optimal number of units that the blood bank should collect at the start of every month?

Q.5 Assume that a blood bank organizes blood donation camp after every 2 weeks. Blood is collected in single bags and stored as whole blood (WB) in its inventory. The life of whole blood is 1 month. The blood bank follows an order up to level policy where the blood bank collects, each time, the difference between a target inventory level (R) and the stock on hand immediately before the camp. The duration of the camp can be neglected for the purpose of this problem. If the monthly demand of WB follows a Poisson distribution with an average of 60 units, what is the optimal target inventory level (R) for blood collection for the blood bank?

[23] In Q.1 it was assumed that the blood bank can arrange any number of donors of a given blood group type for donation at the start of the day. For the purpose of this problem it is assumed that the blood bank can arrange any number of donors to donate blood but their blood group can only be determined after the blood has been physically collected.

Q.6 Resolve Q.5 assuming that a blood bank organizes blood donation camp after every 10 days.

Q.7 Based on the information given in the chapter and the data provided in the file "Blood_Bank_Case_Study_DATA.xls" develop a comprehensive simulation model for computing the optimal level of collection and componentizing at the blood bank chosen for the purpose of the study. The details required for the simulation model are summarized below:

- The blood bank organizes roughly 150 camps per year. The time interval between two successive camps (T) can be assumed to be a random variable with an average of 2.359 days and a standard deviation of 2.22 days. The actual distribution of T is shown in Fig. 18.4. Refer to the sheet "Duration between camps" for the actual dataset.
- The number of units donated at blood donation camps is a random variable with an average of 83.57 units and a standard deviation of 53.09 units. The actual distribution of supply quantity per camp is shown in Fig. 18.4. Refer to the sheet "Supply Quantity Per Camp" for the actual dataset.
- The number of units donated daily at the blood bank is a random variable with an average of 1.574 units and a standard deviation of 2.31 units. The actual distribution of daily donation quantity at the blood bank is shown in Fig. 18.4. Refer to the sheet "BB_Donor" for the actual dataset.
- The percentage composition of positive blood groups in a given collection can be assumed to be fixed (see Table 18.4).The negative blood groups are rare (<6 %) and can be neglected for the purpose of simulation model.
- The percentage of units failing screening tests along with the poor donations can be assumed to be 2 % of the supply.
- The shelf lives of blood and its components have been shown in Fig. 18.2.
- The averages and the standard deviations of daily demand of blood and its components have been shown in Table 18.5. The daily demand data for various blood products has been given in the sheet "Consolidated (Only Data)". Since platelets do not carry any charge the consolidated demand of platelets has been provided in the sheet.
- Model Logic: At the beginning of every day the stock of blood units collected through camps and blood bank donations is added to the existing inventory. Depending upon the quantity collected and the number of units fractionated, the stock of WB and components of various blood groups is updated. Then the demand for various products is generated and the stock of the respective items is reduced appropriately. The batches of various blood products scheduled to expire on the day, if any, are removed from the inventory. The simulation clock is advanced to the next day and the same procedure is repeated till the experiment is over. Refer to Fig. 18.7 for the model logic to be used in the simulation model.
- WB and RBC of the same blood group can be assumed to be perfectly substitutable. Similarly, the platelets of all the four blood groups are assumed to be perfectly substitutable.

- The cross-match release period can be neglected. It can be assumed that the units are issued instantaneously as and when the demand arrives. The units are issued in FIFO order where the oldest unit from the stock is issued first whenever the demand for a blood product arrives at the blood bank.
- The holding cost of one unit for all the blood products is assumed to be ₹ 25 per unit per year.
- The cost of single blood bag (empty) is assumed to be ₹ 270 while that of a triple bag (empty) is assumed to be ₹ 500.
- Three policies can be considered for blood collection through camps (Unrestricted Collection Policy, Modified (Q, T) policy and Modified (R, T) policy) as discussed in Sect. 18.5.4 of the chapter. The values of Q, R and the percentage of componentizing to be considered for the purpose of simulation have been shown in Table 18.12.
- The performance of a collection policy for a given level of parameters can be evaluated in terms of yearly shortage and wastage of blood and its components along with the costs of purchasing and holding blood bags.

References

1. Wilson, S. (web-link). Blood in the body: What is blood made of? Red Gold: The Epic Story of Blood. http://www.pbs.org/wnet/redgold/basics/whatisblood.html. Accessed 29 Dec 2012.
2. American Society of Hematology. (web-link). Blood Basics. http://www.hematology.org/Patients/Blood-Basics/5222.aspx. Accessed on 29 Dec 2012.
3. Wikipedia. (web-link). Blood. http://www.en.wikipedia.org/wiki/Blood. Accessed 29 Dec 2012.
4. 5 Points of Life. (web-link). Apheresis donation. http://www.fivepointsoflife.com/teach-learn/what-are-the-five-points/apheresis-donation/. Accessed 29 Dec 2012.
5. World Health Organization. (2002). *The Clinical Use of Blood—Handbook. Blood Transfusion Safety*. Geneva: World Health Organization.
6. Cantt, L. (2006). Life-saving decisions: A model for optimal blood inventory management. Thesis, Princeton University, Princeton.
7. Pandey, S., Vyas, G. N. (2012). Adverse effects of plasma transfusion. *Transfusion, 52*(S1), 65S–79S (Special Issue: Plasma transfusion: Current status and future directions: The Bernard Fantus, MD Symposium).
8. Wikipedia. (web-link). Blood type. http://www.en.wikipedia.org/wiki/Blood_type. Accessed 29 Dec 2012.
9. Daniels, G., Bromilow, I. (2007). *Essential Guide to Blood Groups*. Hoboken: Blackwell.
10. O'Neil, D. (web-link). ABO blood types. http://www.anthro.palomar.edu/blood/ABO_system.htm. Accessed 29 Dec 2012.
11. Wikipedia. (web-link). Cross-matching. http://www.en.wikipedia.org/wiki/Cross-matching. Accessed on 21 May 2013.
12. Society for the Advancement of Blood Management. (2012). Glossary, Iron Corner. http://www.iron.sabm.org/glossary/index.php. Accessed 26 Sept 2012.
13. Jennings, J. B. (1973) Blood bank inventory control. *Management Science, 19,* 637–645 (Application series).
14. Elston, R. C., Pickerel, J. C. (1970). Blood bank inventories. *Critical Reviews in Clinical Laboratory Sciences, 1*(3), 527–548.
15. Wikipedia. (web-link). Blood bank. http://www.en.wikipedia.org/wiki/Blood_bank. Accessed 29 Dec 2012.

16. Wikipedia. (web-link). Blood donation. http://www.en.wikipedia.org/wiki/Blood_donation. Accessed 21 May 2013.
17. World Health Organization. (2010). *Towards 100 % voluntary blood donation. A global framework for action*. http://www.who.int/bloodsafety/publications/9789241599696eng.pdf. Accessed 21 May 2013.
18. Blood Index. (web-link). Voluntary, altruistic blood donation has proven to be safer than paid or replacement donation. http://www.bloodindex.org/viewarticle.php?id=9. Accessed 21 May 2013.
19. Red Cross. (web-link). Blood Components. http://www.redcrossblood.org/learn-about-blood/blood-components. Accessed 21 May 2013.
20. Seeber, P., Shander, A. (2007). *Basics of blood management* (1st edn.). Hoboken: Blackwell.
21. Central Drugs Standard Control Organization. (2012). Regulatory requirements of blood and/or its components including blood products. http://www.cdsco.nic.in/html/guideline.htm. Accessed 26 Sept 2012.
22. Lane Blood Center. (web-link). Blood facts. http://www.laneblood.org/blood-facts/. Accessed 21 May 2013.
23. Animal Health Diagnostic Center. (web-link). Blood components. http://www.ahdc.vet.cornell.edu/clinpath/modules/coags/comp.htm. Accessed 22 May 2013.
24. AABB. (2013). Whole blood and blood components. http://www.aabb.org/resources/bct/bloodfacts/pages/fabloodwhole.aspx. Accessed 26 Sept 2012.
25. Lowalekar, H., Ravichandran, N. (2011). A model for blood components processing. *Transfusion, 51*(7 Pt 2), 1624–1634. (Special Issue: Journal of Blood Services Management).
26. The Blood Bank. (web-link). Facts about blood. http://www.thebloodbank.org/factsaboutblood.asp. Accessed 22 May 2013.
27. Prastacos, G. P. (1984). Blood inventory management: An overview of theory and practice. *Management Science, 30,* 777–800.
28. Lowalekar, H., Ravichandran, N. (2010). Model for blood collections management. *Transfusion, 50*(12 Pt 2), 2778–2784 (Special Issue: Journal of Blood Services Management).
29. Lowalekar, H., Ravichandran, N. (2014). Blood bank inventory management in India. OPSEARCH, 51(3), 376–399.

Chapter 19
Assembling Cells in Wet Cell Traction Batteries that Power Forklifts

G. S. R. Murthy and Katta G. Murty

1 Forklifts and Their History

Now-a-days forklifts are used extensively worldwide on the shopfloors of manufacturing companies and in warehouses to carry loads from one place to another. In modern warehouses, items are stored in many different layers vertically one above the other; they use forklifts that can lift heavy items to 24 ft or more vertically to store items in higher layers, or retrieve stored items from higher layers.

Forklifts, also called lift trucks, forktrucks, or forklift trucks, are basic materials handling equipment to lift and transport materials. Their development originated in mid-nineteenth century. At the beginning of twentieth century railroads started using them in the form of platform trucks for moving luggage in train stations; and soon after, manufacturing companies developed and started using them in their factories. Both World Wars I and II spurred the development of efficient methods for storing materials in warehouses, and the consequent development of newer models of forklifts that are more maneuverable and which can reach greater heights. By the 1960s the modern forklift was developed and became indispensable in manufacturing, warehousing, and luggage handling operations. Along with the pallet working as a platform for holding the load, the pallet–forklift combination is the tool used for moving loads in manufacturing plants and other areas where moving loads is part of the operations.

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_19) contains supplementary material, which is available to authorized users.

G. S. R. Murthy (✉)
SQC & OR Unit, Indian Statistical Institute, Street No.8,
Habsiguda, 500 007 Hyderabad, India
e-mail: murthygsr@gmail.com

K. G. Murty
Department of Industrial and Operations Engineering, University of Michigan,
Ann Arbor, MI 48109-2117, USA
e-mail: murty@umich.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_19, © Springer Science+Business Media New York 2015

Fig. 19.1 A forklift and one of the batteries powering it. The cells in the battery can be seen in the figure

Now there are many different types of forklifts in use. Some are powered by an internal combustion engine using diesel, kerosene, gasoline, natural gas, butane, or propane as fuel. But the most common are electric forklifts powered by rechargeable lead-acid batteries [2]. A battery lasts 6 consecutive hours of work, or a complete 8-hr shift with 2–3 breaks, while in use powering forklifts. In food processing and health care-related facilities, forklifts powered by batteries are required to prevent harmful emissions (Figs. 19.1–19.4).

In this chapter we will discuss applications of optimum decision making techniques in the process of recharging these batteries in daily use. For convenience, the data presented in the subsequent sections (data of Tables 19.1, 19.3, 19.5, and 19.6) of this chapter can also be accessed in Excel file titled Forklifts.xlsx.

2 Background on Batteries Powering Forklifts

Each battery consists of 24 cells connected in series. Two important characteristics of the cells in these batteries are their voltage (measured in volts) and specific gravity.

The specific gravity of a cell refers to the specific gravity of the electrolyte in it (the electrolyte is sulphuric acid), which is defined as the ratio of the weight of a certain volume of the electrolyte to the weight of an equal volume of water at a specified temperature. So the specific gravity is a unitless number; it is measured using a hydrometer. As the power in the cell is being discharged while the battery is powering the forklift in its work, the specific gravity of the electrolyte in it keeps on decreasing in proportion to the ampere-hours of power discharged. Correspondingly when the battery is recharged, the specific gravity associated with each cell in it keeps on increasing. Thus, the specific gravity of the electrolyte in a cell is an indicator of its state of charge. The industry standard for the specific gravity of a fully charged GB Industrial Battery is 1.285, and the specifications for it are to be between 1.285 and 1.295, and the difference between the specific gravities of any two cells in a battery should not exceed 0.030.

While the specific gravity is defined for each individual cell in a battery, and not for the whole battery; voltage measured in volts is defined for each individual cell, and also for the battery as a whole. We shall use the notation L for the voltage (in volts) of the whole battery under full charge (100 %) condition. Since the cells in a battery are connected in series, L will be equal to the sum of the voltages of the

Fig. 19.2 Wood pallet. The load to be carried is placed on top of it. The forklift can lift the pallet with the load on it by inserting its two forks through the left and right slots of the pallet and then carry it

Fig. 19.3 Forklift moving a load of rubber sheets in a tyre manufacturing plant

individual cells in it. Voltage specifications for the individual cells are 2.55–2.65 V. Voltages of individual cells are measured after recharging the battery to 80 % level of charge or higher. The specification on L is that it should be above 62.15 V, to make sure that the time interval between two consecutive rechargings of the batteries powering a forklift is reasonably long.

For a single cell, the approximate relationship between its voltage and specific gravity under fully charged state is:

$$(\text{Voltage in volts}) = (\text{Specific gravity}) + 0.845.$$

For more details see Fig 19.5 and http://www.giantbattery.com/GLOSSARY/Specific.Gravity-Industrial.Batteries.html.

3 The Decision Making Problems

This chapter describes work carried out at M/s Birla Tyres Limited at Laksar, India by the SQC & OR Division, Indian Statistical Institute, Hyderabad, India, and is prepared with the company's permission. The company uses forklifts powered by 10 batteries, and each battery consists of 24 cells connected in series. Once the charge in

Fig. 19.4 Forklift truck. (The source for this figure is the Wikipedia article on forklifts at http://en.wikipedia.org/wiki/Forklift_truck. Accessed 30 June 2014)

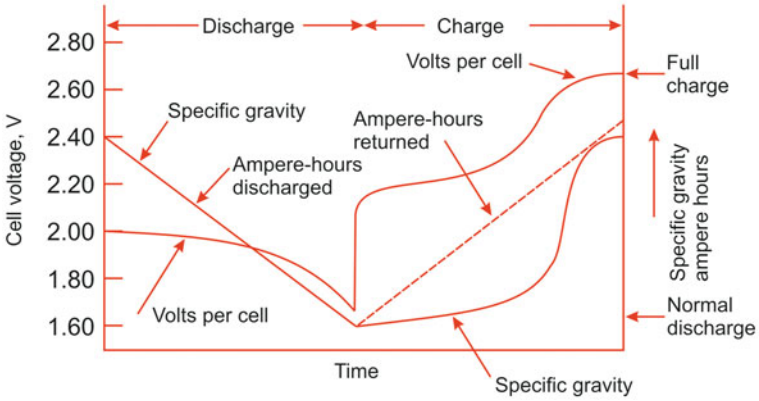


Fig. 19.5 Approximate relationship between voltage and specific gravity of a cell

the batteries in a forklift reaches a level where the functioning of the forklift becomes ineffective (about 20 % of full charge), all the batteries in the forklift are recharged in the company's recharging unit. When the cumulative voltage of the 24 cells in a battery put together reaches 80 % of the full charge level or greater, measurements of the specific gravity and the voltage of each of the cells in the battery are taken before the batteries are put to work again.

The cycle length of the batteries in a forklift is the length of the working time interval between two consecutive recharging operations for them. The company has the goal of maximizing this cycle length, which depends on $\bar{v}$ = minimum of the voltages at full charge of the ten batteries powering the forklift. If all the batteries

powering the forklift satisfy the condition on the minimum required voltage under full charge of L; then this $\bar{v}$ will be $\geq L$. If, however, this condition is not satisfied by the batteries in a forklift, then the company faces the following decision making problem.

DMP-1: Given the vector of voltages, $v = (v_1, \ldots, v_{240})^T$ of all the 240 cells in the 10 batteries in a forklift that have just been recharged, recombine these 240 cells into 10 new batteries of 24 cells each to maximize V = minimum of the voltages of the 10 new batteries created.

If the V value corresponding to the solution obtained for DMP-1 is satisfactory, then the 240 cells in the forklift can be recombined into 10 batteries corresponding to that solution; which are then fitted into the forkilift and the forklift is put into operation.

On other hand if the value of the V corresponding to the solution obtained for DMP-1 is smaller than the specified value for it and is not satisfactory, the company now faces the following decision making problem.

DMP-2: Given the vector of voltages $v = (v_1, \ldots, v_{240})^T$ of all the 240 cells in the 10 batteries in a forklift that have just been recharged, what is the maximum number of batteries each consisting of 24 cells among these, with their voltage values $L \geq 62.15$ V, that can be formed?

If the optimum solution of DMP-2 yields $r < 10$ batteries each with voltage $L \geq 62.15$ V, then the remaining cells from the existing batteries in the forklift are combined with cells from other forklifts in the charging unit at that time to form the remaining $10 - r$ batteries needed for this forklift, and the forklift is put into operation. Or sometimes the remaining cells from the batteries in the forklift may be recognized as having some maintenance problems, then they are sent to the battery maintenance unit for repair; and the forklift is put to work after being fitted with new batteries as needed.

4 Mathematical Models for DMP-1 and 2

Model for DMP-1 We will consider a general version of DMP-1. Consider a set $S = \{1, \ldots, n\}$ of cells that have just been fully charged, where $n \geq 24k$; n, k are both positive integers (in the version of DMP-1 described in Sect. 3, $k = 10$, and $n = 240$). $v = (v_1, \ldots, v_n)^T$ is the vector of voltages (in volts) of cells in S in that order. The problem is to form k groups of 24 cells each from the cells in S, each group to be assembled into a battery, such that V = minimum of the voltages of the batteries formed is maximized. We model this problem as a binary integer program. Define the decision variables:

$x_{ij} = 1$ if the i-th cell in S is included in group j; 0 otherwise, for $i = 1$ to n, $j = 1$ to k

T_j = voltage in volts of the j-th battery formed by assembling cells in the jth group into a battery, for $j = 1$ to k

$V = \text{minimum}\{T_1, \ldots, T_k\}$.

Then the model is

$$\text{Maximize} \quad V$$

$$\text{subject to} \quad \sum_{i=1}^{n} x_{ij} = 24 \quad \text{for } j = 1 \text{ to } k$$

$$\sum_{j=1}^{k} x_{ij} \leq 1 \quad \text{for } i = 1 \text{ to } n$$

$$T_j = \sum_{i=1}^{n} v_i x_{ij} \geq V \quad \text{for } j = 1 \text{ to } k$$

$$x_{ij} = 0 \text{ or } 1 \quad \text{for all} \quad i = 1 \text{ to } n,\ j = 1 \text{ to } k$$

The first constraint states that each of the k groups formed must consist of exactly 24 cells from S. The second constraint states that each of the cells in S can be included in at most one of the k groups formed. The third constraint states that for each $j = 1$ to k, T_j, voltage of the battery formed by assembling the cells in the jth group, is the sum of the voltages of cells in this group, and that its value is $\geq V$. These constraints, and the fact that we are maximizing V subject to these constraints, automatically guarantee that V will be equal to minimum$\{T_j : j = 1 \text{ to } k\}$.

Example 1 Data on the voltages of 240 cells which have just been recharged are given in Table (19.1). Form this set into 10 groups of 24 cells each, each group to be assembled into a battery, to maximize V = minimum of the voltages of the 10 batteries formed.

The model for this problem has been solved using Lingo software (see http://www.lindo.com/for Lingo; a trial version is available). For this example, $n = 240$ and $k = 10$. The solver could not get the optimal solution even after 40 min of CPU running time. As the problem is a binary integer programming problem with 2401 decision variables, it is not surprising that the solver could not end up finding an optimal solution. However, the solver reports the bound for the objective function and the best solution obtained at the end of each iteration. The upper bound for the objective function is 61.7 V. It is interesting to note that the feasible solution obtained within 3 s of run time from the beginning is 61.61 V, and at the end of the 32nd s the solver found a feasible solution with an objective of 61.68 and thereafter it remained at 61.68 V even after 40 min of CPU running time. At the end of the 40th min the solver was interrupted manually and the best solution that was found thus far was taken as a feasible near optimum. The solution obtained is presented in Table 19.2. Each entry in the table has numbers in the format $\frac{a}{b}$ where b is the cell number (as in Table 19.1) and a is its voltage.

Exercise 1 Consider the data of Table 19.1 and the solution presented in Table 19.2. Identify x_{ij}s for this solution. Suppose you form the batteries by assembling the cells as follows: put cells 1–24 to form the first battery, 25–48 to form the second, and so on, 217–240 to form the tenth battery (see the note below Table 19.1 for the

Table 19.1 Cell voltages (in volts) after recharging

2.70	2.60	2.95	2.87	3.24	2.28	2.06	2.85
2.78	2.26	2.63	2.55	2.60	2.69	1.99	2.41
2.87	2.28	2.83	2.60	2.84	2.74	2.65	1.90
2.51	2.21	2.85	2.25	2.24	2.50	2.03	2.72
2.48	2.22	2.99	2.93	2.21	2.64	2.92	1.85
3.50	3.16	2.13	2.30	3.03	2.96	2.71	2.56
2.77	2.58	2.29	2.28	3.16	2.59	2.54	2.93
2.22	2.77	2.59	2.71	2.77	2.65	2.23	2.43
2.74	2.67	2.34	2.79	2.29	2.27	2.66	2.39
2.54	2.64	2.24	2.98	2.25	2.20	2.69	2.77
2.57	2.86	2.64	2.62	2.57	3.30	2.85	2.91
2.68	2.28	2.97	2.70	2.19	2.67	3.19	2.72
2.45	1.95	2.88	2.56	2.73	3.17	2.43	2.47
3.24	2.28	2.51	2.32	1.93	2.55	2.58	2.57
2.67	2.37	2.49	2.60	2.14	2.00	2.27	3.36
2.42	2.75	2.97	2.90	3.13	2.32	2.85	2.36
2.04	2.38	2.09	2.49	2.99	1.36	2.24	2.46
2.00	3.09	2.71	2.42	2.59	2.99	2.87	2.44
2.37	2.94	2.60	2.39	2.45	2.43	2.09	2.57
2.47	3.37	2.41	2.44	2.39	2.66	2.67	2.74
2.22	2.70	2.68	1.82	2.32	2.85	2.56	2.67
3.32	3.01	2.22	3.11	2.18	2.63	2.54	2.73
2.82	2.24	2.40	2.65	2.38	2.58	2.65	2.54
2.69	2.70	3.19	2.85	1.89	2.01	2.88	2.03
2.92	2.27	3.01	2.65	2.51	2.48	2.32	2.23
2.39	2.58	2.20	2.27	2.82	2.64	2.58	2.43
2.57	2.86	2.67	2.57	2.95	2.52	2.49	2.75
2.85	3.22	2.81	1.90	2.53	2.90	2.68	2.68
2.61	2.41	2.89	2.55	2.67	2.72	3.22	1.85
2.42	2.40	2.24	2.24	2.26	2.79	2.26	2.11

Entry of i^{th} row and j^{th} column is the voltage of $(8(i-1)+j)^{th}$ cell in a set of 240 cells

cell numbering and the corresponding voltage). Identify the assignment matrix for this assignment and compute the battery voltages for this assembly. Compare this assignment with the solution with the one presented in Table 19.2.

Exercise 2 Consider another set of data given in Table 19.3. Identify n and k for this problem. Form the maximum number of batteries from these data using your intuition or any heuristic you can think of. Solve the model for DMP-1 for this data (you may use any optimization software such as MATLAB, Excel solver, etc.). Compare different solutions you have obtained. Suppose that the battery voltage should be a minimum of 62.15 V. How many batteries can you form meeting this requirement from the data given in Table 19.3? This is actually a question meant for DMP-2. Pondering over this will help in understanding the next section.

Table 19.2 Solution to the problem

B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}
2.69/14	1.99/15	3.24/5	2.26/10	2.70/1	2.60/2	2.87/4	2.28/6	2.95/3	2.83/19
2.28/18	1.90/24	2.60/13	2.41/16	2.63/11	2.78/9	2.55/12	2.06/7	2.85/8	2.21/26
2.25/28	2.99/35	2.51/25	2.24/29	2.65/23	2.74/22	2.84/21	2.87/17	2.60/20	2.64/38
1.85/40	2.56/48	2.85/27	2.22/34	2.50/30	2.72/32	2.71/47	2.03/31	2.93/36	2.13/43
2.58/50	2.29/69	2.67/66	2.96/46	2.92/39	2.48/33	2.29/51	3.16/42	2.77/49	2.59/59
2.93/56	3.30/86	2.57/85	2.54/55	3.50/41	2.21/37	2.79/68	2.30/44	2.59/54	2.65/62
2.22/57	2.68/89	2.85/87	2.23/63	3.03/45	3.16/53	2.98/76	2.28/52	2.43/64	2.39/72
2.71/60	2.19/93	2.56/100	2.24/75	2.74/65	2.77/58	2.69/79	2.77/61	2.25/77	2.86/82
2.34/67	2.58/111	2.49/115	2.57/81	2.66/71	2.54/73	2.43/103	2.77/80	2.20/78	2.70/92
2.27/70	2.75/122	2.27/119	3.19/95	2.64/83	2.62/84	2.47/104	2.67/94	2.28/90	2.72/96
2.64/74	2.97/123	2.04/129	2.88/99	2.28/106	3.17/102	2.36/128	2.67/113	2.97/91	2.45/97
2.91/88	2.00/137	3.09/138	2.73/101	2.51/107	1.93/109	2.38/130	2.37/114	1.95/98	2.57/112
3.24/105	2.43/150	2.37/145	2.55/110	2.90/124	2.60/116	1.36/134	3.13/125	2.14/117	2.42/121
2.32/108	2.57/152	2.41/155	2.24/135	2.09/131	2.42/140	2.44/144	2.32/126	2.99/133	2.22/161
2.00/118	3.37/154	2.39/157	2.99/142	2.46/136	2.39/148	2.68/163	2.49/132	2.71/139	2.73/176
3.36/120	3.32/169	2.70/162	2.44/156	1.82/164	2.09/151	2.56/167	2.87/143	2.59/141	2.54/184
2.85/127	3.11/172	2.65/183	2.67/159	2.22/171	2.47/153	2.58/182	2.45/149	2.66/158	2.85/188
2.94/146	2.40/179	2.51/197	2.67/168	2.65/180	2.74/160	2.88/191	2.32/165	3.01/170	2.67/211
2.60/147	2.01/190	2.39/201	2.63/174	2.38/181	2.18/173	2.92/193	2.85/166	2.54/175	2.81/219
1.89/189	2.27/194	2.58/207	2.82/177	2.70/186	2.24/178	2.49/215	2.20/203	2.69/185	1.90/220
2.03/192	2.65/196	2.43/208	2.27/204	2.23/200	3.19/187	2.68/223	2.61/225	2.48/198	2.90/222
3.01/195	2.57/209	2.85/217	2.64/206	2.82/205	2.32/199	2.72/230	2.89/227	2.52/214	2.68/224
2.58/202	2.95/213	2.42/233	2.57/212	2.41/226	2.86/210	2.26/237	3.22/231	2.40/234	2.55/228
3.22/218	1.85/232	2.24/236	2.75/216	2.26/239	2.53/221	2.79/238	2.11/240	2.24/235	2.67/229
61.71	61.70	61.68	61.71	61.70	61.75	61.72	61.69	61.74	61.68

Note: The ten columns give the ten batteries formed (B_js stand for batteries); the entries are cell voltage/cell number; and last row gives battery voltages

5 Approach to Solve DMP-2

We will consider a general version of DMP-2. Consider a set $S = \{1, \ldots, n\}$ of cells that have just been fully charged; with $v = (v_1, \ldots, v_n)^T$ as the vector of voltages of these cells in that order. The question is to determine the maximum number of batteries (each consisting of 24 cells) that can be formed from S, such that the voltage of each battery formed is greater than or equal to a specified minimum voltage $L \geq 62.15$.

Let p be the integer part of $n/24$, that is, p satisfies $24\ p \leq n < 24(p+1)$. Clearly p is an upper bound for the maximum number of batteries that can be formed. Define

Table 19.3 Cell voltages (in volts) after recharging

2.64	2.50	2.87	3.04	2.67	2.56	2.68	2.43
2.99	2.43	2.55	2.67	2.22	2.33	2.64	2.95
2.34	2.88	2.69	2.60	3.02	3.07	2.79	2.39
3.00	2.72	3.20	2.53	2.76	2.78	3.11	2.80
2.65	2.91	2.48	2.94	2.42	2.61	3.08	2.47
2.67	2.60	2.79	2.96	2.91	2.74	2.23	2.84
2.52	2.77	3.13	2.83	2.80	2.44	2.89	2.80
2.39	2.43	2.85	3.07	2.59	3.15	2.74	2.43
2.70	2.88	2.88	1.99	2.69	2.57	2.92	2.44
2.56	2.86	2.82	2.53	2.48	2.52	2.22	3.08
2.25	2.28	2.23	2.51	2.41	2.61	2.65	2.91
2.81	2.11	2.53	2.91	2.48	2.47	2.64	2.67
2.96	2.22	2.61	1.93	2.73	2.61	3.09	2.60
2.63	2.87	2.50	2.20	2.68	2.42	2.70	2.49
2.38	2.28	2.60	2.63	2.62	2.42	2.54	2.65
2.68	2.57	2.98	2.40	2.69	2.65	3.00	2.36
2.96	2.72	2.56	2.66	2.64	2.44	2.35	2.53
2.45	2.34	2.66	2.54	2.35	2.43	2.57	2.96
2.23	2.69	2.53	1.83	2.79	3.06	2.63	2.55
3.02	2.63	2.49	2.92	2.95	2.75	2.40	2.47
2.35	2.52	2.63	2.92	2.44	2.83	2.64	2.48
2.65	2.54	2.81	2.83	2.59	2.35	2.35	2.93
2.69	2.68	2.62	2.58	2.24	2.85	2.61	2.59
2.28	2.58	2.54	2.26	2.49	2.94	2.56	2.17
2.17	2.86	2.94	2.63	2.33	3.00	2.01	2.72

Entry of i^{th} row and j^{th} column is the voltage of $(8(i-1)+j)^{th}$ cell in a set of 200 cells

binary decision variables $x_{ij} = 1$ if the i^{th} cell in S is included in group j; 0 otherwise, for $i = 1$ to n, $j = 1$ to p.

Let r be an integer between 1 and p. Clearly we can form r batteries each with voltage $\geq L$ from the given set S of cells, if the following system has a feasible solution.

$$\begin{aligned}
&\sum_{i=1}^{n} x_{ij} = 24 \quad \text{for } j = 1 \text{ to } r \\
&\sum_{j=1}^{r} x_{ij} \leq 1 \quad \text{for } i = 1 \text{ to } n \\
&\sum_{i=1}^{n} v_i x_{ij} \geq 62.15 \quad \text{for } j = 1 \text{ to } r \\
&x_{ij} = 0 \text{ or } 1 \quad \text{for all } i = 1 \text{ to } n,\ j = 1 \text{ to } r
\end{aligned} \tag{19.1}$$

In fact if $\bar{x} = (\bar{x}_{ij})$ is a feasible solution of the above system, then for $j = 1$ to r, form the jth battery by assembling all the cells $i \in S$ satisfying $\bar{x}_{ij} = 1$; and vice versa.

We will now describe a simple enumerative approach for solving this problem. First check if the above system (19.1) has a feasible solution $\bar{x}$ for $r = p$; if so p is the answer to the question (i.e., p batteries can be formed from the set of cells in S such that the voltage of every battery formed is ≥ 62.15).

If system (19.1) has no feasible solution for $r = p$, then check whether it has a feasible solution for $r = p - 1$. Continue the same way decreasing r by 1 until you find $\bar{r}$ = the largest value of r for which system (19.1) has a feasible solution. Then $\bar{r}$ is the answer to the question, and from that feasible solution $\bar{\bar{x}}$ to system (19.1) corresponding to $\bar{r}$; for $j = 1$ to $\bar{r}$, the jth battery formed consists of all cells $i \in S$ satisfying $\bar{\bar{x}}_{ij} = 1$.

This simple enumerative approach is quite satisfactory for the forklift application, since the p that we have to deal with in this application is 10, quite small.

Example 2 Consider the data of Table 19.1. Suppose it is required that each battery should have a minimum voltage of 62.15 V. How many batteries can we make up from the cells with voltages of Table 19.1? We can answer this question by using the approach mentioned above. When this problem is solved using Lingo, for $p = 10$, there is no solution (the answer was flashed by Lingo within a second); and when the problem is solved using $p = 9$ a feasible solution was found within a second. The solution is presented in Table 19.4. It may be noted that all nine batteries have their voltages above 62.15 V.

Exercise 3 Solve the model DMP-2 for the data in Table 19.3.

Exercise 4 Consider the problem of forming the batteries with a given set of cells. Consider the following heuristic. Let p be the number of batteries you want to form from the given n cells. Let $v_1, v_2, \ldots, v_n$ be the n voltages after arranging them in the descending order, that is, $v_1 \geq v_2 \geq \ldots \geq v_n$. Determine p sets $S_1, S_2, \ldots, S_p$ as follows. Put $v_j, v_{p+j}, v_{2\,p+j}, \ldots$ in S_j, $j = 1, 2, \ldots, p$. Assess this heuristic and comment.

Exercise 5 Suppose $v_1, \ldots, v_{24}$ are the voltages at full charge of a single battery. Then the variance of the voltages of cells within this battery is defined as $\sum_{i=1}^{24} (v_i - \bar{v})^2/24$, where $\bar{v} = \left(\sum_{i=1}^{24} v_i\right)/24$. Consider the following problem: we are given $n = 24k$ cells with voltages $v_1, \ldots, v_n$ volts. It is required to divide these n cells into k groups, to form each group into a battery; such that each battery has voltage $\geq L$ volts, while minimizing the sum of the variances of the voltages of cells of the k batteries formed. Formulate this problem. Solve this problem with the data in Table 19.1.

6 Summary

In this chapter we presented a decision making problem that was encountered in a tyre manufacturing company. Basically the problem is about making decisions on selective assemblies of charged cells to form batteries. Two objectives were considered for this problem. The first objective aims at forming a specified number of batteries

Table 19.4 Solution to the problem

B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
2.06/7	2.95/3	2.60/2	2.78/9	2.70/1	2.28/6	3.24/5	2.69/14	2.87/4
2.85/8	2.28/18	2.60/13	2.26/10	2.63/11	2.83/19	2.55/12	2.25/28	2.51/25
2.60/20	2.50/30	2.48/33	2.87/17	2.84/21	2.74/22	2.65/23	2.29/51	2.22/34
2.96/46	2.99/35	1.85/40	2.85/27	2.21/26	2.28/52	2.24/29	2.54/55	2.93/36
3.16/53	2.21/37	3.50/41	2.71/47	2.72/32	2.65/62	2.92/39	2.64/74	2.71/60
2.59/54	2.56/48	3.16/42	2.93/56	2.64/38	2.79/68	2.59/59	2.24/75	2.54/73
2.69/79	2.58/50	3.03/45	2.23/63	2.30/44	2.25/77	2.67/66	2.64/83	2.98/76
2.57/85	2.85/87	2.77/49	2.29/69	2.77/58	3.19/95	2.66/71	3.30/86	2.20/78
3.17/102	2.19/93	2.74/65	2.68/89	2.43/64	2.45/97	2.28/90	2.67/94	2.57/81
2.28/106	2.72/96	2.34/67	3.24/105	2.62/84	2.47/104	2.73/101	2.88/99	2.86/82
1.93/109	2.56/100	2.77/80	2.49/115	2.91/88	2.32/108	2.55/110	2.43/103	2.57/112
2.60/116	2.90/124	2.37/114	2.00/118	2.97/91	3.36/120	2.14/117	2.51/107	2.67/113
2.27/119	2.85/127	2.32/126	3.13/125	2.70/92	2.49/132	2.04/129	2.75/122	2.42/121
2.87/143	2.44/144	2.47/153	2.09/131	2.58/111	2.00/137	2.46/136	2.97/123	2.99/133
2.09/151	2.45/149	3.37/154	2.99/142	2.36/128	2.60/147	2.94/146	2.71/139	2.42/140
2.67/159	2.66/158	3.32/169	2.43/150	2.24/135	2.39/148	2.39/157	2.41/155	2.74/160
2.67/168	3.01/170	2.18/173	2.85/166	3.09/138	2.70/162	2.63/174	2.32/165	1.82/164
2.69/185	2.70/186	2.03/192	2.24/178	2.37/145	2.56/167	2.92/193	2.73/176	2.38/181
2.27/194	1.89/189	2.51/197	2.65/180	2.68/163	3.11/172	3.01/195	2.88/191	3.19/187
2.58/202	2.48/198	2.23/200	2.65/183	2.82/177	2.58/182	2.52/214	2.39/201	2.20/203
2.86/210	2.43/208	2.81/219	2.64/206	2.40/179	2.85/188	2.68/224	2.27/204	2.58/207
3.22/218	2.67/211	1.90/220	2.57/209	2.54/184	2.82/205	2.41/226	2.75/216	2.85/217
1.85/232	2.95/213	3.22/231	2.89/227	2.65/196	2.53/221	2.72/230	2.68/223	2.90/222
2.79/238	2.42/233	2.11/240	2.24/235	2.61/225	2.26/237	2.40/234	2.67/229	2.24/236
62.29	62.24	62.68	62.7	62.78	62.5	62.34	62.61	62.36

As in Table 19.2 columns represent batteries and the entries are cell voltage/cell number; and last row gives battery voltages

with given voltages of the cells so as to maximize the minimum of the battery voltages. The second objective aims at extracting as many batteries as possible when it is required that the voltage of each battery must exceed a minimum specified voltage. Though the problems are addressed in the context of managing forklift batteries, the models discussed here are useful to a wide range of applications. For instance, a similar problem is faced in the manufacture of lightning arresters [1, 3, 4].

Table 19.5 Cell voltages (in volts) after recharging

2.57	2.57	2.67	2.56	2.53	2.63	2.63	2.54
2.59	2.56	2.58	2.63	2.63	2.66	2.56	2.51
2.50	2.60	2.62	2.57	2.54	2.69	2.51	2.63
2.59	2.58	2.56	2.70	2.64	2.54	2.52	2.60
2.54	2.60	2.52	2.52	2.56	2.69	2.54	2.55
2.52	2.60	2.54	2.51	2.61	2.68	2.65	2.57
2.66	2.60	2.53	2.67	2.68	2.53	2.51	2.62
2.56	2.54	2.59	2.59	2.70	2.63	2.51	2.68
2.51	2.62	2.65	2.65	2.64	2.56	2.66	2.62
2.61	2.70	2.51	2.69	2.61	2.55	2.62	2.59
2.58	2.55	2.60	2.65	2.67	2.59	2.55	2.54
2.51	2.63	2.65	2.61	2.53	2.60	2.58	2.59
2.51	2.64	2.69	2.57	2.67	2.54	2.56	2.63
2.52	2.57	2.54	2.59	2.62	2.54	2.67	2.51
2.55	2.58	2.68	2.53	2.59	2.66	2.53	2.67
2.65	2.57	2.69	2.66	2.65	2.69	2.57	2.55

Entry of i^{th} row and j^{th} column is the voltage of $(8(i-1)+j)^{th}$ cell

Table 19.6 Cell voltages (in volts) after recharging

2.58	2.58	2.58	2.58	2.59	2.5	2.5	2.58
2.58	2.58	2.59	2.57	2.58	2.58	2.58	2.58
2.58	2.59	2.59	2.58	2.59	2.58	2.58	2.59
2.58	2.58	2.39	2.58	2.59	2.58	2.59	2.59
2.58	2.58	2.58	2.50	2.58	2.58	2.58	2.58
2.58	2.58	2.58	2.59	2.58	2.58	2.58	2.59
2.58	2.57	2.57	2.57	2.57	2.57	2.58	2.58
2.59	2.59	2.57	2.56	2.56	2.56	2.57	2.57
2.58	2.58	2.57	2.57	2.57	2.57	2.56	2.56
2.56	2.57	2.57	2.58	2.56	2.57	2.57	2.57
2.58	2.58	2.57	2.56	2.56	2.56	2.57	2.57
2.58	2.58	2.57	2.57	2.57	2.57	2.56	2.56
2.56	2.57	2.57	2.58	2.56	2.57	2.57	2.57
2.58	2.58	2.50	2.59	2.57	2.58	2.58	2.58
2.57	2.56	2.56	2.57	2.57	2.57	2.58	2.58
2.58	2.58	2.58	2.58	2.57	2.58	2.57	2.57
2.58	2.58	2.58	2.59	2.57	2.59	2.58	2.58
2.58	2.57	2.58	2.57	2.56	2.56	2.56	2.57
2.57	2.57	2.58	2.59	2.58	2.58	2.57	2.56

Entry of i^{th} row and j^{th} column is the voltage of $(8(i-1)+j)^{th}$ cell

7 Problems

1. Consider the cell voltage data in Table 19.5. Lightning arrestors use the same cells which are used for the forklift batteries but each line arrestor requires six cells. The minimum voltage of line arrestor is 61 V (as the cells within a line arrestor will be connected in series, the arrestor voltage will be sum of the voltages of the cells in it). What is the maximum number of line arrestors you can form from the cells with voltages given in Table 19.5. Supposing it is desired that the voltage of

the arrestors should be as high as possible, group the cells in such a way that you get the maximum number of arrestors meeting the minimum voltage requirement of 61 V. Consider the data in Table 19.5. What is the maximum number of batteries you can form from the cells with voltages as given in Table 19.5? Is your solution best? Explain your criterion for the "best"? If not, explain your procedure for finding the best.

2. Repeat the above exercise for the data in Table 19.6 below.

Acknowledgements The authors wish to thank M/s Birla Tyres management for providing an opportunity to work on this live problem and allowing us to publish this work. In particular, special thanks are due to Mr. Jai Kishan, the Technical Executive, who has introduced the problem and Mr. Akhilesh Sinha, Chief Operating Officer of the company, who is responsible for driving these applications in their industry.

References

1. Htwe, N. K. (2008). Analysis and design selection of lightning arrester for distribution substation. *World Academy of Science, Engineering and Technology*, *2,* 12–29.
2. Linden, D., & Reddy, T. B. (Ed.). (2002). *Handbook of batteries* (3rd ed.) (p. 23–25). New York: McGraw-Hill.
3. Pinceti, P., & Giannettoni, M. (1999). A simplified model for zinc oxide surge arresters. *IEEE Transactions on Power Delivery, 14*(2), 393–398.
4. Zanetta Jr., L. C. (2003). Evaluation of line surge arrester failure rate for multipulse lightning stresses. *IEEE Transactions on Power Delivery, 18*(3), 796–801.

Chapter 20
Giving Appointments to Patients for Surgeries, and Scheduling Surgeries to Operating Rooms in a Day

Katta G. Murty, Weidong Chen and Mary Goulet Duck

1 Introduction

Surgery (from the Greek word "cheirourgia" ("cheir" means hand + "ergan" means work)) is the branch of medicine dealing with the physical manipulation of an organ in the body to diagnose, prevent, or cure an ailment. The effort by many human societies all over the world to develop surgical techniques for treating injuries and traumas have been traced from the study of prehistoric human remains and archaeological records to 12,000 BCE. However, until modern times surgeons were not able to overcome the three principal obstacles that plagued the medical profession from its infancy—bleeding, pain, and infection.

In early nineteenth century, even a minor procedure like tooth extraction used to cause excruciating pain to the patient, so surgeons learned to work with lightning speed. Every surgical procedure involved assistants pinning patients down as they screamed and struggled until they fainted from the intense pain. Then, in 1846, a Boston dentist named William Morton approached the famous local surgeon of the day Henry Jacob Bigelow, to arrange a demonstration of a gas he had discovered that could render patients insensitive to the pain of surgery. The demonstration was held on 16 October 1846 at Massachusetts General Hospital by administrating the gas

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_20) contains supplementary material, which is available to authorized users.

K. G. Murty (✉) · W. Chen
Department of Industrial and Operation Engineering, University of Michigan,
Ann Arbor, MI 48109-2117, USA
e-mail: murty@umich.edu

W. Chen
e-mail: aschenwd@umich.edu

M. G. Duck
University of Michigan Hospitals and Health Centers,
Ann Arbor, MI 48105-2969, USA
e-mail: mgduck@umich.edu

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_20, © Springer Science+Business Media New York 2015

through an inhaler in the mouth of a young man undergoing the excision of a tumor in his jaw. The patient only uttered minor sounds to himself in a semi-conscious state during the procedure. Next day, the gas left a woman undergoing the cutting of a large tumor from her upper arm completely silent and motionless, when she woke up she said that she experienced no pain during the procedure. Thus, anesthesia was born, and it spread rapidly all over the world.

The other great scourge of surgery was sepsis-infection. Infection was so prevalent that the discharge of pus from a surgical wound was thought to be an essential part of healing. Then, reading Louis Pasteur's papers on microorganisms as the cause of infection, and the use of some chemicals to eliminate them, in 1860s, the Edinburgh surgeon Joseph Lister perfected a way to use carbolic acid for cleansing hands and wounds and destroying the germs entering the operating field. His antiseptic method resulted in the modern standard of asepsis, i.e., entirely excluding germs from the surgical field using heat-sterilized instruments and surgical teams clad in sterile gowns and gloves, leading to strikingly lower rates of sepsis and death from surgeries.

With such phenomenal developments in medicine in modern times, all the major problems that plagued the medical profession in earlier times have now been mostly overcome, and surgery has been transformed from a highly risky "art" into a scientific discipline capable of treating many diseases and illnesses.

A brief history of surgery, and of medicine itself is given in Chap. 5; for more details the reader is referred to the articles in Wikipedia on "Surgery" (http://wikipedia.org/wiki/Surgery) and "History of Surgery" (http://en.wikipedia.org/wiki/History_of_surgery).

In this chapter, we consider only *elective surgery* which generally refers to a surgical procedure that can be scheduled in advance because it does not involve a medical emergency. It is done to correct a non-life-threatening condition, and is carried out at the patient's request, subject to the surgeons and the surgical facility's availability. Every year more than 15 million people have elective surgery at 7230 hospitals in the US to relieve pain, or to reduce a symptom, or improve some body function, or to diagnose some problem (an example of this is a biopsy which involves removing a piece of tissue to examine under a microscope), or to save life. Forecasts indicate that the demand for surgery may increase between 14 and 47 % by 2020 (see [1]). Also, financial constraints are an issue for American hospitals especially in that insurance companies and medicare tend to pay the same or less each year for the same procedure. This points out the importance for hospitals to have efficient operating rooms and surgery scheduling systems.

Types of Surgery Surgical procedures are commonly classified by several criteria: urgency, type of procedure, degree of invasiveness, special instrumentation needed, body parts involved. For the application discussed in this chapter, the classification involves the following surgical specialities: general surgery, cardiac surgery, thoracic surgery, vascular surgery, pediatric surgery, plastic surgery, transplant surgery, otolaryngology, gynecology, oral and maxillofacial surgery, dental surgery, orthopaedic surgery, neurosurgery, ophthalmology, pediatric surgery, and urology. Some subspecialities include: colorectal surgery, surgical oncology, trauma and burn surgery (or

also known as acute care surgery), endocrine surgery, gastrointestinal surgery, spine surgery, minimally invasive surgery, and sports medicine.

Various Components in a Surgical Procedure Now-a-days surgeries are carried out in an *operating theater* (also called *operating room*), using surgical instruments, an operating table for the patient, and other surgical equipment, under sterile conditions.

Activities for the Patient Before S/He Enters the Operating Room When the patient arrives at the hospital, s/he goes to pre-op holding prior to the operating room. There the patient changes out of her/his street clothes into the clothes provided by the hospital for the surgery, her/his vital signs are recorded, an IV line is placed, the pre-operative medications are given, consents are validated, the site is marked, the pre-op nurse prepares the patient, the anesthesiologist and the surgeon individually review what they will do in the case.

The procedure starts when the patient enters the operating room, and ends when s/he exits it. The room is then cleaned for the next patient. The surgical procedure consists of three components or phases one after the other, as described below.

1. Set-up: We will call this phase as set-up, which includes the time to prepare the operating room, induce the patient with anesthesia and prepare her/him for the surgery, which we will describe below. Here are the various activities that go on in this phase.

a: Pre-operative Preparation of the Operating Room: This activity is called *room preparation* or *room set-up*. This work involves preparing the operating room for the surgery, getting all the equipment, instrumentation and surgical and anesthesia supplies, and everything else ready for the surgery. The majority of this work occurs prior to the patient entering the operating room.

b. Induction: Anesthesia has lead for moving the patient from pre-op holding area to the OR suite, to begin the intra-op time. The circulating nurse and the surgical resident may assist. When they arrive in the room, they move the patient to the OR table and cover her/him with warm blankets. Anesthesia staff intubate the patient and begin administering anesthesia (local, general, or spinal as appropriate for the surgery) to her/him to prevent pain from incision, tissue manipulation, and suturing. They also connect any monitoring equipment to the patient to ensure the patients' vitals are stable and pain is stable during the procedure. While anesthesia staff are intubating, others are continuing to set-up the room and ensure documentation of the case is in place.

c. Prep/Position/Drape: This activity is to prepare the patient for the surgery. The patient is positioned into how s/he should be for the procedure, with the surgical resident leading that activity with assistance from the circulating nurse. Any hair present at the surgical site (skin surface to be operated on) is clipped off, the surgical site is cleaned and smeared with an antiseptic. Sterile drapes are used to cover all of the patient's body except for the head and the surgical site, and the drapes are clipped to a pair of poles near the head of the bed to form an ether screen separating the unsterile working area of the anesthesiologist from the sterile surgical site. The surgical equipment needed for the surgery is staged around the sterile field by the circulating nurse and scrub tech. Most times, during this activity the surgeon will not be

in the operating room, all other members of his/her surgical team carry out this work. At the end of this activity, the surgeon will arrive and discuss the case, and ensure that the set-up is as desired. The surgeon and resident will then scrub in for the case (completing a surgical scrub of their hands and arms, and applying sterile gowns).

2. Surgery: This phase is also called *procedure time*, and begins with the incision time after the team takes what is called a time out to ensure all team members are doing the right work on the correct patient. In this phase, the surgeon performs the surgery, and carries out the work to correct the problem in the patient's body. Once the procedure is completed, sutures or staples are used to close the incision, and the surgeon leaves the operating room to inform the family of how the surgery went. The resident applies any dressings to the patient's wounds.

3. Extubation and Operating Room Clean-Up: We will call this phase under the combined name *clean-up*. For the patient, extubation includes the stopping of the anesthetic agents. The circulating nurse cleans up the patient. The scrub tech puts instrumentation back together. The resident is generally documenting the case at this point, but may assist the circulator. When anesthesia deems that the patient is awake and stable, s/he is ready to leave the Operating Room. Then, the anesthesiologist and the surgical resident move the patient to a gurney (commonly used word for a "stretcher") and transport to the post anesthesia care unit (PACU).

Room Clean-up: This phase is also called clean-up time or post-intra-op time for the operating room. It is carried out right after the surgery is completed, and it involves cleaning-up the operating room and getting it ready for the next surgery to be carried out in it. Once this clean-up is completed, the procedure is considered to be over.

Activities of the Patient After the Procedure: The patient recovers in the PACU. However, about 10 % of critical in-patients bypass the PACU and go directly to an intensive care unit (ICU). When the patient arrives at PACU, the anesthesiologist and resident hand-off and report to the PACU nurse how the case went and what to expect in recovery. The PACU nurse monitors the patient and assists her/him to recover from the anesthesia and surgery. When certain vitals are reached, the patient is judged to have recovered from the anesthesia. Inpatients are then transferred to a surgical ward, and patients going home move to what is called Phase II to finish recovering until they feel stable to be discharged home.

At this point many processes are happening in parallel with multiple workers keeping the aspects of the surgical suites moving for the patients. The preparation time for the next surgery in this operating room can start right after this clean-up process for the surgery just carried out.

In [2], it has been reported that at one hospital, about 21 % of time of an operating room time is spent on set-up and clean-up, 15% of time is anesthesia time, and 64 % is time spent conducting surgery on an average.

For the University of Michigan Health system (UMHS), the total operating case time (i.e., the patient in the OR to patient out of the OR time) average is 3 h. Seventeen percent is spent on anesthesia time (before and end of procedure). Sixty-nine percent is spent on the actual procedure and 14 % is spent on prepping, positioning, and draping the patient. The patient spends an average of almost 2 h in pre-op holding

area, and 3 h in the PACU. Outpatients going home would spend almost another 2 h in the PACU. So, on average a patient coming for a day surgery would spend about 10 h at UMHS. Average turnover for clean-up and set-up between cases for UMHS's operating rooms is about 42 min, i.e., 21 % of their allocated OR time, similar to benchmarks.

Operating room management deals with the problem of running the surgical suite at maximum efficiency, maximizing the number of surgeries carried out/day while at the same time minimizing the cost of resources used and other related costs. In this problem, people have considered the objective functions to be optimized: minimizing the idle time of the operating room, minimizing surgeon's waiting time between pairs of surgeries the surgeon has to perform in a day, minimizing any overtime needed to complete all the surgeries that are scheduled to be performed in a day, minimizing inpatient's length of stay waiting for surgery, maximizing the hospital's revenue, or minimizing the patient's total length of stay in the hospital.

2 The Decision Making Problems Considered

Our project was carried out at the University of Michigan, Ann Arbor (UM) Hospital abbreviated as UMUH (University of Michigan University Hospital). UM has hospitals or clinics in several places around Ann Arbor, and some types of surgical procedures are carried out only at some of these branches; and UMUH refers to its main adult hospital at the UM campus.

UMUH has 27 operating rooms with most of them able to accommodate any type of surgery. A typical working day is from 7 A.M. to 5:30 P.M. during Monday to Friday, but some of the operating rooms run till 7:30 P.M., and a few of them till 11:30 P.M. Surgeons usually take work shifts for 8, 10, or 12 h a day. Surgeons in academic institutions with teaching and other duties perform surgeries for about 3–4 days a month, most other surgeons in an academic hospital generally do surgeries 7–10 days a month. Surgeons in private practice typically would be performing surgeries more often than this.

About four operating rooms are used on weekends and holidays, but we limit our discussion to weekdays only. If some planned work continues into overtime (i.e., beyond the normal working hours in the day), the overtime cost is 50 % more expensive than that for normal time.

The surgery section of UMUH is divided into departments, each one handling one type of surgeries. The abbreviated titles of various departments at UMUH are: Oto (Otolaryngology), Oral, Neuro (Neurosurgery), Ortho (Orthopaedic), Plastic, Uro (Urology), Gyn (Gynecology), GSA (General Surgery), and STX (transplant surgery). The other types of surgeries are carried out in different locations and are not included in our study.

Study of past data at UMUH has revealed that the time taken for any surgical procedure varies considerably depending on the skill, experience, and seniority of the surgeon, and the other personnel in the surgical team; and the condition of

Table 20.1 Data on the average and standard deviation in minutes for set-up, surgery, clean-up, and the total time in minutes for a "Laparoscopic Cholecystectomy" case by 17 different surgeons

Surgeon	Cases/year	Time in minutes for Laparoscopic Cholecystectomy case							
		Set-up		Surgery		Clean-up		Total	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
ACS1	40	62.4	18.1	133.6	41.6	21.1	8.7	217.1	53.0
ACS10	20	72.6	16.3	94.0	25.3	19.2	5.7	185.8	40.8
ACS2	28	81.1	22.9	178.7	55.5	19.4	4.2	279.3	62.0
ACS4	32	70.9	20.7	86.3	23.8	18.5	7.0	175.6	43.9
ACS5	20	80.6	21.8	140.0	95.0	18.4	5.2	239.0	115.5
ACS6	32	64.6	17.4	107.0	49.8	18.1	5.1	189.8	61.7
ACS7	28	74.6	24.1	142.7	44.9	22.4	9.4	239.7	49.2
ACS8	10	87.5	9.2	131.0	87.7	46.0	5.7	264.5	102.5
ACS9	12	68.7	14.0	103.0	20.2	20.3	6.8	192.0	22.6
COL2	11	77.0	9.9	86.0	31.1	19.5	3.5	182.5	37.5
HEP2	44	74.5	13.0	81.5	24.4	17.8	6.8	173.7	32.2
HEP3	12	56.7	3.1	82.7	17.9	17.3	3.5	156.7	17.6
HEP4	24	66.2	12.8	62.8	24.2	23.3	5.4	152.3	34.2
HEP6	16	54.8	5.7	88.8	11.0	18.0	3.6	161.5	11.0
MIS2	11	63.0	11.3	95.0	38.2	15.5	0.7	173.5	50.2
MIS3	24	60.2	10.2	92.0	29.5	17.0	5.1	169.2	39.3
MIS4	56	57.0	12.1	79.6	29.2	20.9	12.3	157.6	36.9

the patient being treated which itself varies. As an example, Table 20.1 gives the average, and the standard deviation of the time in minutes taken by the surgical teams of 17 different surgeons within the General Surgery Department and four of its subspecialities at UMUH to perform a laparoscopic cholecystectomy (gall bladder removal using the laparoscopic approach).

The Patient Arrival Process When a patient has a health condition, s/he consults her/his primary care physician (PCP). The PCP makes a preliminary diagnosis, and if it involves some type of surgery, s/he asks the patient to consult a surgeon in the relevant department in the surgical section. The PCP's referral may be to a specific surgeon, or any surgeon in that area. The patient will then call the area's department. The receptionist in that department gives the names of their available surgeons, and asks the patient to consult them. The patient may talk to one or more than one among them, and if the diagnosis involves surgery, the patient selects one of the surgeons consulted to perform the surgery. When a selection has to be made among several surgeons, the manner in which the patient makes this selection, is similar to the process described in Chap. 5. The surgeon then determines if the patient needs surgery, and if so, the date of the surgery depending on his own work schedule in consultation with the department (this is described below in detail).

Of course there are several decision-making problems in running the surgical suite efficiently; but in this chapter we will only consider three of the most important decision-making problems among them; and describe the approaches adopted to solve them.

Table 20.2 Operating rooms allocated to departments. M, Tue, Wed, Thu, F are Monday to Friday, respectively. GSA is general surgery. "FCFS and shared" mean that these rooms are shared by all departments whichever comes first; we can also call these rooms as "open" rooms; sometimes they are are also used for urgent or emergency cases which we do not consider in our study

Procedure	Rooms and room times allocated
Oto	Room 1, 2, 3 M–F full time;
Oral	Room 4 M, Thu full time; Room 9,11 Tue full time;
Neuro	Room 6, 7, 8 M–F full time; Room 4 Wed full time;
Ortho	Room 5, 10, 14, 15, 16 M–F full time;
GSA	Room 9 M; Room 13 M–F; Room 17 F; Room 18 Wed; Room 19 Tue–Wed; Room 20 M–Thu; Room 27 M, F; Room 28 M, W, F—full time, Room 17 M part time
Plastics	Room 12 M–F—full time;
STX	Room 17 Tue–Thu—full time;
Uro	Room 19 Wed–F; Room 22, 26, 29 M–F; Room 28 Tue, Thu—full time;
Gyn	Room 20 F; Room 21 M–F—full time;
ACS	Room 27 Tue, Thu—full time;
FCFS	Room 9 W–F; Room 18 M, Thu, F—full time; Room 24 full time
Shared	Room 4 Tu, F Shared between FCFS, Oto & Neuro Room 11 M, Shared bet. FCFS Room 17, 18 M, shared bet. FCFS & GSA; Room 27 shared bet. FCFS & GSA

2.1 The First Decision-Making Problem: Allocation of Operating Room Time to Departments

The first decision-making problem is to allocate operating room time to various departments, so that they can fix surgery dates for the arriving patients. This is carried out on a monthly basis, but the allocations for a month are decided annually at budget time for the year.

This decision-making problem has been discussed extensively in the published literature, and most research publications discuss models for solving this problem based on linear programming, or integer programming, or goal programming techniques etc. These models do not seem appropriate for solving this problem mainly because at UMUH, the patient, when s/he arrives, is allocated the first available convenient slot for her/his surgery; and also because patients arrive for service one by one individually, and the patient arrival process involves the surgeon selecting the date of surgery convenient to his schedule and again the decision is reached at the time he speaks individually to the department about the patient. Also, the actual surgery time varies a lot depending on the surgeon who performs it and the resident assisting.

At UMUH this allocation for the planning month is made using what is known in the literature as a *block allocation strategy*. This strategy allocates operating rooms and continuous blocks of working time in them to departments, based on the actual usage pattern, and trends in them at the time this decision is made, in proportion to the average (or median) usage, i.e., overall cases treated per month in the departments. Table 20.2 gives an example of an allocation of operating rooms (and working time in them) to the departments in a typical working week in a month at UMUH.

The important thing is that there is flexibility in implementing the allocations determined, as patients begin to arrive and fill up the lists of surgeries scheduled in the planning month in the various departments. At some stage, if say Department 1, has used up all the operating room time allotted to it for the planning month, and more patients are lining up for service from them; but another Department 2 has spare operating room time left in its allocation, then some of that spare time will be transferred to Department 1 as needed, this happens the day before when the blocks are released. Otherwise Department 1 will overflow into the designated first come first serve (FCFS) rooms.

2.2 The Second Decision-Making Problem: Fixing the Surgery Date for a Patient

At UMUH the policy adopted for making this decision uses a FCFS strategy for scheduling patients. This decision is made at the time the surgeon who is going to perform the surgery calls the department for fixing this date for her/his patient.

The total amount of time (in minutes) this surgical procedure will take, from preparation to clean-up, say Tt, is needed for making this decision. But this Tt = the total amount of time (in minutes) a surgical procedure of this type will take when performed by this particular surgeon, is a random variable, estimates of its expected value μ, and its standard deviation σ can be computed from our data base.

Suppose in our schedule we allow an amount, t minutes of time, for this surgery on the surgery date. If the realized value of Tt is $< t$, then the operating room will be idle for $t - Tt$ minutes before the next surgery scheduled for this room begins in it. On the other hand if the realized value of Tt is $> t$, then the starting of the next surgery in this room will be delayed by $Tt - t$ minutes, and this may eventually lead to an overtime expense for $Tt - t$ minutes for this surgical team at the end of the day. We should determine the value for t so as to minimize z = expected value of the sum of either of these two expenses (cost of operating room idle time + cost of possible overtime) that may occur as a result of the value picked for t.

We carried out a simulation with $t = \mu + a\sigma$ for several values of a, and found that $a = 0.5$ minimizes the value for z. So, we will take $t = \mu + 0.5\sigma$.

When the department hires a new surgeon, there will be a short interval of time after this new surgeon performs his first one or two surgeries during which we do not have enough data for estimating the standard deviation of the time in minutes of the surgical procedure when performed by this particular surgeon. During that short interval of time we will take this surgeon's standard deviation for surgery time to be 0.

So when the surgeon calls the department to fix the surgery date for her/his patient, the department uses the following procedure to determine this date: they look up the surgeries for which dates are fixed already among the dates in the future, and find the earliest date in the future when an operating room suitable for this type of surgery has a continuous interval of length $\mu + 0.5\sigma$ minutes free, and fix that date and room

Table 20.3 Number of cases of surgeries treated in each department in July, August, September

Number of cases per month per department				
Department	July	August	September	Total
General Surgery	389	367	358	1114
General Surgery Transplant	74	67	59	200
Gynecology	130	100	78	308
Neurosurgery	158	179	161	498
Oral Surgery	72	48	58	178
Orthopaedics	257	249	255	761
Otolaryngology	207	231	229	667
Plastic Surgery	119	124	107	350
Urology Surgery	283	336	290	909
Total	1689	1701	1595	4985

Table 20.4 Average number of minutes/case (total of set-up, surgery, clean-up phases) in each department

Average total time per case per month per department				
Department	July	August	September	Total
General Surgery	230	219	227	225
General Surgery Transplant	270	281	265	272
Gynecology	221	215	219	218
Neurosurgery	329	290	296	305
Oral Surgery	241	229	255	242
Orthopaedics	230	246	226	233
Otolaryngology	211	240	249	234
Plastic Surgery	212	229	235	225
Urology Surgery	186	171	183	179
Grand average	230	227	231	229

for this surgery, and set aside for this surgery the earliest interval of time on that day during which this room has no other procedure scheduled in it earlier, and inform the surgeon about it so that he can mark it in his schedule and also inform this date to the patient.

Exercise 1 In our data base in file "ch20Exer1data.xlsx," we give data on various surgeons and the various types of surgeries they performed over a consecutive period of 3 months. Use this data to determine how much time of the available 27 operating rooms during the next month (month number 4) to allocate to each department using the procedure similar to that described above.

Exercise 2 At UMUH the data on all the cases of surgeries carried out in the Departments GSA, STX, Gyn, Neuro, Oral, Ortho, OTO, Plastic, and Uro over a 3-month period are shown in Tables 20.3, 20.4, and 20.5. Table 20.3 gives the number of cases of surgeries treated in each of these departments in July, August, September, and the totals for these 3 months. Table 20.4 gives the average number of minutes/case (total of set-up, surgery, clean-up phases) in each department in each of those months. Table 20.5 gives the average number of minutes of time of surgery phase/case in each department in each of those months.

Table 20.5 The average number of minutes of time of surgery phase/case in each department

Average surgery time per case per month in each department				
Department	July	August	September	Total
General Surgery	132	125	134	130
General Surgery Transplant	157	172	159	163
Gynecology	128	126	126	127
Neurosurgery	184	161	166	170
Oral Surgery	131	120	139	131
Orthopaedics	120	134	120	125
Otolaryngology	114	143	150	136
Plastic Surgery	118	130	139	129
Urology Surgery	100	88	99	95
Grand average	127	127	132	129

Using this information, determine the allocation of time of the available 27 operating rooms in October to these departments, by the approach described above.

2.3 The Third Decision-Making Problem, Sequencing the Various Surgeries to be Performed By a Surgeon Over the Time of the Day

This decision-making problem is faced by surgeons who perform more than one surgery in a day. Such surgeons are found commonly in all departments for certain types of surgeries. The average time for such surgeries is about 3 h/patient and a surgeon typically performs surgeries on three such patients/day. Neurosurgery has the longest case times at 4.28 h per case and those surgeons tend to only do one or two cases per day.

When a surgeon has to perform two or more surgeries in a day, the time gap between a consecutive pair of surgeries performed is known as *the turnover time* which becomes *the surgeon's waiting time* between that pair. The sum of a surgeon's waiting times between various consecutive pairs of surgeries performed on that day is called his *total waiting time* during that day. Surgeons usually like to have this total waiting time to be as small as possible, even though typically during the waiting time between surgeries, the surgeon has other duties to attend to, including visiting the family of the case just completed, completing post-operative documentation, and visiting the next patient in the pre-op area.

If all these surgeries are to be performed in the same operating room, the waiting time between every consecutive pair of surgeries will include both a clean-up time and a set-up time, and as a result of this the surgeon's total waiting time will be large. Kildea [3] suggested that the surgeon's total waiting time can be reduced by using two or more operating rooms for holding surgeries, because while the surgeon is busy in performing a surgery on a patient in one room, another patient can undergo

set-up in another room, and as soon as the surgery is finished in the first room, the surgeon can transfer to perform the surgery in the other room, with 0 waiting time.

We will now discuss a heuristic algorithm to arrange the surgeries to be performed by a surgeon in one day, using two operating rooms, to minimize the total waiting time of the surgeon performing them on that day; that gives good results in practice.

Let $r \geq 2$ denote the number of surgeries to be performed by the surgeon on that day. We will use the indices $i, j = 1$ to r to denote the various surgeries. The data that we need is the mean and standard deviation of the preparation time, surgery time, clean-up time in minutes for each of these surgeries. Let:

$OR1, OR2$	denote the two operating rooms available to the surgeon to perform these surgeries.
tp_i, ts_i, tc_i	denote the (mean) + 0.5(standard deviation) in minutes of the preparation time, surgery time, clean-up time, respectively, of the ith surgery, $i = 1$ to r. We will use these as the estimates of the time needed to carry out these tasks.
r_1, r_2	will denote the number of surgeries allocated by the algorithm in OR1, OR2, respectively, $r_1 + r_2 = r$. The algorithm needs to determine these and also the other things listed below.
T	$T = 0$ will denote the clock time when the surgical team of the surgeon starts setting up the first surgery. We assume that the team will continue working until all the r surgeries for the team on that day are completed.
h_k	will denote the kth surgery in the sequence of surgeries performed by the surgeon on that day, for $k = 1$ to r.
R_k	will denote the operating room (OR1 or OR2) in which the surgeon performs the kth surgery of the day, for $k = 1$ to r.
T_k	will denote the clock time when the surgeon just finishes the kth surgery of the day, and the clean-up for that surgery has just started in room R_k.
s_k	will denote the total surgeon waiting time incurred until clock time T_k for $k = 1$ to r, $s_1 = 0$.
a_k	$= T_k + tc_{h_k}$ = clock time at which room R_k is ready for the set-up of the next surgery after the surgery h_k has been completed in it.

Step 1 of the algorithm determines the first surgery to be carried out in each of OR1, OR2; and surgeon will start work in OR1. Once this pair has been selected, we start clock time at $T = 0$, and start advancing the clock time. We will measure clock time in minutes beginning with $T = 0$. As clock time advances, whenever the clean-up phase for the surgery being carried out in any of the two operating rooms has been completed, at that point of time the algorithm has to determine which among the remaining surgeries will be carried out next in that operating room; the General step in the algorithm describes how this is determined.

At any point of clock time, the *remaining set of surgeries* to be performed on this day denotes the set of surgeries to be performed on this day for which even the preparation has not commenced at that time.

2.4 The Algorithm to Schedule the r Surgeries for the Day in OR1, OR2

Step 1 Among the r surgeries to be performed by the surgical team of this surgeon, look for an ordered pair (i, j) attaining the minimum $\{|tp_i + ts_i - tp_j| : \text{over all } i, j\}$. If the ordered pair $(\bar{i}, \bar{j})$ attains this minimum, make surgeries $\bar{i}, \bar{j}$ as the first surgeries to be carried out in OR1, OR2, respectively, and let the surgeon start the work in OR1, make clock time as $T = 0$, and start advancing the clock time.

Before we describe the general step in the algorithm, we will mention the assumption on which it is based. This assumption holds at UMUH by the way it runs the Surgery section, as it helps to minimize the surgeon's total waiting time between consecutive pairs of surgeries s/he performs on that day, and also the idle time of the operating rooms.

Assumption As soon as the surgeon completes a surgery in an operating room, the clean-up procedure of that room begins and is completed without any delay; and as soon as this clean-up is completed, the next surgery to be performed in this room by this surgeon is selected, and preparation for that surgery is started and completed. Then, the room waits for the surgeon to return to carry out that surgery.

In the general step, we will describe the criterion that we will use in this algorithm to select the next surgery to be performed in an operating room, once the clean-up for a surgery has been completed in that room.

General Step Suppose we reached the point of time T_k at which the surgeon has just completed the kth surgery of the day in room R_k, and the clean-up for that surgery has commenced. This clean-up will be completed at clock time $a_k = T_k + tc_{h_k}$.

Let us denote the room among OR1, OR2 other than R_k as ORp. In ORp, at time point a_k either the surgeon may be performing a surgery, or clean-up after a surgery may be going on, or preparation step for another surgery may be going on.

At this clock time a_k select the next surgery to be carried out in this room R_k to be the one which attains the minimum in: minimum $\{tp_i$: over all remaining surgeries i for which set-up has not even commenced$\}$. Call this surgery as the *current surgery* for this room R_k, it replaces the previous current surgery for this room if it had one before. Let:

Room ready time for the surgeon for this room R_k be a_k + (preparation time in minutes for the current surgery for room R_k). It is the earliest time at which room R_k will be ready for the surgeon to begin the current surgery for this room. It replaces any earlier room ready time for this room that may have been defined earlier.

We will now describe what the surgeon will do at clock time T_k when he just finished the surgery h_k in room R_k. It depends on the situation of the other operating room ORp at this clock-time. We enumerate the various possibilities below:

1. Clean-up for a surgery completed before clock time T_k may be going on in ORp: In this case, the surgeon goes into the waiting time interval between the surgery h_k and the next one. Apply for the room ORp, the work described above for R_k, select the next surgery to be set-up in ORp after this clean-up, and determine the room ready time for the surgeon for ORp.

Table 20.6 Some details for ten different examples in the AIMMS-MOPTA contest for 2013

Example	T (OR working hours)	Surgeries to be sequenced
1	4	A, A, C, J
2	4	A, A, G, H, J
3	4	A, D, G, G, J
4	8	A, B, F, G, G, H
5	8	C, D, F, H, J, J, J
6	8	A, A, A, C, D, G, I, J, J, J
7	8	A, A, F, F, G, H, H, H, I, I, J, J
8	12	A, B, D, E, G, G, J
9	12	A, A, B, D, G, G, I, I, J, J
10	12	A, A, C, E, E, F, G, H, I, I, J

At the clock time: Minimum {room ready time for the surgeon, in the two rooms}, the current waiting time interval for the surgeon ends, and the surgeon will begin the surgery set-up in the room attaining this minimum. That room is R_{k+1}, and h_{k+1} = the $(k+1)$th surgery to be performed by the surgeon on this day, is that surgery set-up in R_{k+1}; and consequently T_{k+1} = (room ready time for the surgeon in room R_{k+1}) + $ts_{h_{k+1}}$; and finally, $s_{k+1} = s_k + [($ room ready time for the surgeon in room $R_{k+1}) - T_k]$.

2. Set-up for the next surgery in this room after a surgery is completed may be going on in ORp: In this case, the clock time at which that set-up procedure ends is the present room ready time for the surgeon for ORp.

Again at the clock time Minimum{room ready time for the surgeon, in the two rooms}, the current waiting time interval for the surgeon ends, and the surgeon will begin the surgery set-up in the room attaining this minimum. That room is R_{k+1} in this case. The updating of the other quantities in this case is similar to that in Case 1.

3. The room ORp has already undergone the set-up for the next surgery in it, and is ready for the surgeon's arrival: In this case, the surgeon just walks into this room ORp from the room R_k where he just finished the surgery h_k, and right away starts the surgery set-up in ORp which becomes h_{k+1}; R_{k+1} = ORp; surgeon waiting time between surgeries h_k, h_{k+1} is 0 and consequently $s_{k+1} = s_k$; $T_{k+1} = T_k + ts_{h_{k+1}}$.

Advance clock time to T_{k+1} go to the next step.

2.5 Example and Solution for AIMMS/MOPTA Contest Problem

The following examples are from the 5th AIMMS-MOPTA Optimization Modeling Competition-2013 [4]—*Operating Room Management under Uncertainty*. Here, the various types of surgeries are denoted by letters A to J. The original data on individual surgeries can be accessed at http://www.aimms.com.

In Table 20.6, we show ten different examples set-up in the competition. T in Column 2 gives the number of hours for which each of the two different operating rooms OR1, OR2 are made available to complete all the surgeries in this example, any extra time taken beyond this time will be considered as overtime. Column 3

Table 20.7 S.type in Column 1 is "surgery type." In the row for each surgery type, is given the average and standard deviation in minutes of the set-up, surgery, clean-up times for that type

	Set-up		Surgery		Clean-up	
S. type	Avg.	Std.	Avg.	Std.	Avg.	Std.
A	5.03	2.16	9.82	3.33	4.10	1.88
B	39.99	17.12	81.46	28.36	40.36	17.94
C	23.68	8.95	59.62	24.43	43.71	17.30
D	14.79	7.92	34.57	17.50	18.53	8.56
E	46.81	20.80	120.90	48.65	47.46	24.23
F	25.97	9.12	47.76	15.29	23.92	7.23
G	31.77	14.01	43.94	21.83	23.58	11.85
H	21.26	5.74	39.92	11.60	21.98	6.56
I	137.21	37.79	94.75	48.16	66.51	17.98
J	18.07	3.30	19.51	9.79	43.01	8.07

Table 20.8 S. type is surgery type. The times in minutes tp, ts, tc for each of set-up, surgery, clean-up phases are obtained as explained above for each type of surgery

	Surgery type							
Phase time	A	C	E	F	G	H	I	J
tp	6.11	28.16	57.21	30.53	38.78	24.13	156.11	19.72
ts	11.49	71.84	145.23	55.11	54.86	45.72	118.83	24.41
tc	5.04	52.36	59.58	27.54	29.51	25.26	75.50	47.05

gives the types of surgeries required to be carried out by the surgeon in this example, denoted by letters of the alphabet A to J. There are multiple surgeries in this list for some types. Our goal is to find the sequence of surgeries carried out in each of the two rooms that minimizes the total overtime, and the surgeon's waiting time between consecutive pairs of surgeries.

From the data given in the competition, we computed the average and standard deviation in minutes per surgery of each phase (set-up, surgery, clean-up), of the types A to J by this surgeon; and show these in Table 20.7.

As an illustration, we will show the solution of Example 10 (given in the last row of Table 20.6) using the Heuristic algorithm discussed in Sect. 2.4. The surgeries to be carried out using rooms OR1, OR2 in this example are: A, A, C, E, E, F, G, H, I, I, J. For each of the set-up, surgery, clean-up of each surgery in this example, we will take the time required in minutes tp, ts, tc as (the average time) + 0.5 (standard deviation) given in Table 20.7 for that type of surgery. These times are shown in Table 20.8.

Selection and sequencing of the surgeries to be performed in each of OR1, OR2 for Example 10 involving 11 surgeries (last row of Table 20.6):

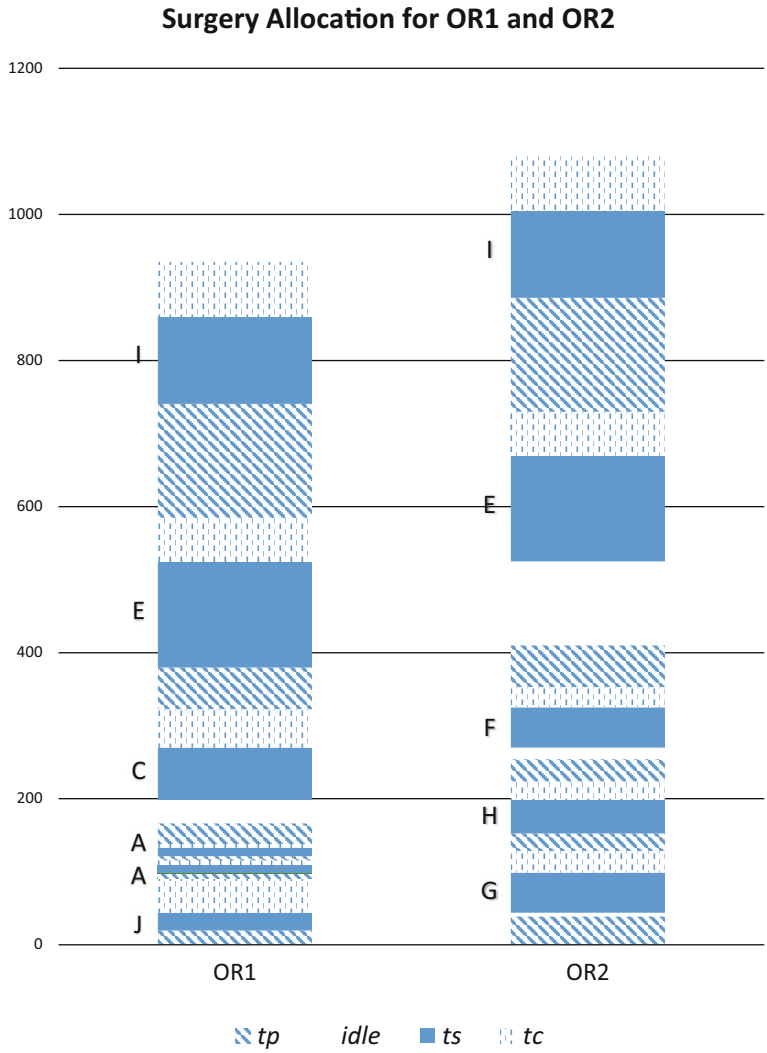


Fig. 20.1 In this column chart, time is quoted in minutes, "tp" indicates set-up time, which is shown in *diagonal pattern*; "idle" indicates operating room idle time, which is shown in *blank*; "ts" indicates surgery time, which is shown in *solid pattern*; "tc" indicates clean-up time, which is shown in *dashed vertical pattern*; The surgery type is shown besides each block

Initial Surgery to Perform in Each of OR1, OR2 Among the 11 surgeries to be performed by the surgical team of this surgeon, look for the pair of surgeries (i, j) which attain the minimum $\{|tp_i + ts_i - tp_j| : \text{over all } i, j\}$. Then, put surgery i in OR1, and surgery j in OR2. In this example, this pair is (J, G) so the first surgeries scheduled in OR1, OR2 are taken as J, G, respectively. So, in the notation of Sect. 2.3, $h_1 =$ J, $R_1 =$ OR1, $T_1 = 44.13$, $s_1 = 0$.

Subsequent Surgeries in OR1, OR2 The first surgery scheduled in OR2 is G, and the set-up time for it from Table 20.8 is $38.78 < T_1 = 44.13$. So, by clock time T_1 this set-up will be completed and OR2 will be ready for the surgeon to perform this surgery, so at clock time T_1 after completing surgery J in OR1, the surgeon moves over without any waiting directly to OR2 to perform surgery G. This surgery will be completed at clock time $44.13 + 54.86 = 98.99$ min. So, $h_2 = \text{G}$, $R_2 = \text{OR2}$, $T_2 = 98.99$, $s_2 = 0$

At clock time $T_1 = 44.13$ min, clean-up begins in room OR1 after surgery J, this will be completed at clock time $44.13 + 47.05 = 91.18$ min. At this clock time 91.18 min, surgery G is going on in OR2, so among the remaining surgeries A, A, C, E, E, F, H, I, I at this stage, the one with the smallest set-up time, an A type, will be set-up in OR1. This set-up will be completed at clock time $91.18 + 6.11 = 97.29$ min, which is the room ready time for the surgeon for this room at this stage.

In OR2, the first surgery G scheduled in it will be completed at clock time $T_2 = 98.99$ min, then clean-up for this surgery will begin there and will be completed at clock time $98.99 + 29.51 = 128.50$ min.

At clock time $T_2 = 98.99$ min, the surgeon completes surgery G in OR2. At that time the set-up for surgery A would have been completed in OR1, so the surgeon can walk over to OR1 without any waiting from OR2 to OR1 to perform surgery A which he will complete at clock time $98.99 + 11.49 = 110.48 = T_3$ minutes, and so $h_3 = \text{A}$, $R_3 = \text{OR1}$, $s_3 = 0$. Then in OR1 clean-up for surgery A just completed begins, and this will be completed at clock time $110.48 + 5.04 = 115.52$ min. So, at clock time $T_3 = 110.48$ min the surgeon begins a waiting period; and as clean-up for surgery G will be going on in OR2, we begin setting-up for the next surgery in OR1, this will be the remaining surgery with the smallest set-up time which is also a type A surgery. This set-up in OR1 will be completed at clock time $115.52 + 6.11 = 121.63$ min. At this clock time, in OR2 clean-up will be going on, so the surgeon will come from his waiting back to OR1 to perform surgery A just set-up in OR1. So, $h_4 = \text{A}$, $R_4 = \text{OR1}$, and this surgery will be completed at clock time $T_4 = 121.63 + 11.49 = 133.12$ min. Clean-up for this surgery begins in OR1 at clock time $T_4 = 133.12$ min and it will be completed at clock time $133.12 + 5.04 = 138.16$ min.

When clean-up is completed in room OR2 at clock time 128.50 min, in OR1 the surgeon is busy performing a surgery, so at that time in OR2 set-up begins for the remaining surgery with the smallest set-up time which is surgery type H, this will be completed at clock time $128.50 + 24.13 = 152.63$ min.

The algorithm continues the same way until all the required surgeries are completed. The final schedule obtained is shown below, and also in Fig. 20.1.

In the final schedule obtained by the algorithm, in OR1 the surgeries scheduled are J, A, A, C, E, I, and those scheduled in OR2 are G, H, F, E, I in that order over time. Measuring clock time from 0 min, OR1 remains idle only between clock time minute 166.32 and 198.35, and all the surgeries scheduled in it are completed at clock time at minute 935.01. OR2 remains idle from clock time minute 38.78 to 44.13, 254.14 to 279.19, 410.05 to 524.99, and all the surgeries scheduled in it are completed at clock time minute 1080.24. The result of h_k, R_k, T_k for all $k = 1$ to 11 is shown in Table 20.9.

Table 20.9 h_k, R_k, T_k denote the kth surgery performed, the operating room number in which it is performed, and the clock time for the surgeon to finish kth surgery, as described before in the algorithm

h_k (Surgery type)	R_k	T_k (in minutes)
1 (J)	1	44.13
2 (G)	2	98.99
3 (A)	1	110.48
4 (A)	1	133.12
5 (H)	2	198.35
6 (C)	1	270.19
7 (F)	2	325.3
8 (E)	1	524.99
9 (E)	2	670.22
10 (I)	1	859.51
11 (I)	2	1004.74

In Fig. 20.1, time is plotted on the vertical axis, and in each of OR1 and OR2 we show the various activities (tp = set-up, ts = surgery, tc = clean-up) going on in that room over time, and show the idle time in blank.

Exercise 3 Find the optimum schedules for each of the remaining nine examples in AIMMS/MOPTA contest.

References

1. Etzioni, D. A., Liu, J. H., Maggard, M. A., & Ko C. Y. (2003). The aging population and its impact on the surgery workforce. *Annals of Surgery, 238*(2), 170–177.
2. Kuo, P. C., Schroeder, R. A., & Mahaffey, S. (2003). Optimization of operating room allocation using linear programming techniques. *Journal of the American College of Surgeons, 197*(6), 889–895. (Elsevier Inc.).
3. Kildea, J. (1970). Operating room scheduling methods: The two room system. *Hospitals, 44,* 99–101.
4. AIMMS/MOPTA contest problem. (2013). See at the website http://www.aimms.com.

Chapter 21
The Satellite Downlink Scheduling Problem: A Case Study of RADARSAT-2

Daniel Karapetyan, Snezana Mitrovic-Minic, Krishna T. Malladi and Abraham P. Punnen

1 Introduction

Mission planning operations of Earth observing satellites involve acquisition of images and downlinking (downloading) the acquired images of prescribed areas of the Earth to one or more ground stations. Efficient scheduling of image acquisition and image downlinking plays a vital role in successful satellite mission planning. The image acquisition and downlinking operations are often interlinked and solved using heuristic algorithms that take advantage of the flexibility allowed within such integrated systems. In this chapter, we study the mission planning operations of Canada's Earth observing synthetic aperture radar (SAR) satellite, RADARSAT-2. Figure 21.1 shows RADARSAT-2 in its orbit.

As noted in [10], the RADARSAT program was born out of the necessity to monitor icy waters of Canada. In addition, the program supports various applications covering marine surveillance, disaster management, soil moisture measurement, creation of Digital Elevation Models (DEM)s, environment management and security, oil pollution monitoring, geological exploration activities, tracking temporal

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_21) contains supplementary material, which is available to authorized users.

D. Karapetyan (✉)
ASAP Research Group, School of Computer Science, University of Nottingham, Nottingham NG8 1BB, UK
e-mail: daniel.karapetyan@gmail.com

S. Mitrovic-Minic
MDA Systems Ltd., Richmond, BC, V6V 2J3, Canada
e-mail: snezanam@mdacorporation.com

K. T. Malladi · A. P. Punnen
Department of Mathematics, Simon Fraser University, Surrey, BC V3T 0A3, Canada
e-mail: kmalladi@sfu.ca

A. P. Punnen
e-mail: apunnen@sfu.ca

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_21, © Springer Science+Business Media New York 2015

Fig. 21.1 RADARSAT-2 in its orbit. This image is the result of a collaboration between the Canadian Space Agency and MacDonald, Dettwiler and Associates Ltd. (MDA). (Reproduced by permission of MDA)

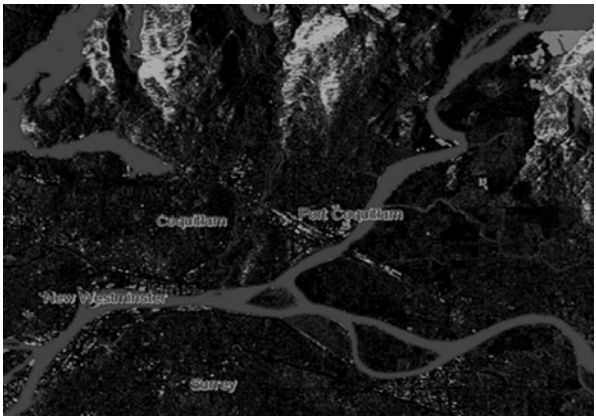


Fig. 21.2 RADARSAT-2 image (in grayscale) showing flooding along the Fraser River, New Westminster and Port Coquitlam area, BC, Canada.[1] ©Canadian Space Agency, 2014. The original color version of this can be viewed at http://www.asc-csa.gc.ca/images/satellites/radarsat2/image-vedette/amerique-nord-coquitlam.jpg

changes in soil and crop conditions, etc. For additional details of these applications, we refer to [10]. Figure 21.2 presents an image (converted to grayscale) showing flooding along the Fraser River, BC, Canada. The image was taken on July 27, 2012 by RADARSAT-2.[1]

[1] The flood extent products are derived from RADARSAT-2 images with a system developed and operated by the Earth Sciences Sector of Natural Resources Canada (NRCan). Flood extent boundaries ©NRCan. RADARSAT-2 Data and Products ©MDA (2012)—All Rights Reserved. RADARSAT is an official mark of the Canada Space Agency (CSA). *Disclaimer*: These products are for demonstration purposes only. The CSA is not responsible for the accuracy, reliability or currency of the information or services provided by external sources.

For RADARSAT-2 mission planning, image acquisition is scheduled by an online algorithm processing client requests as they arrive and information about available time windows. Because of various operational restrictions we are required to use the existing image acquisition scheduling scheme. Thus our study is focused on 'optimally' scheduling image downlinking operations assuming that an image acquisition schedule is already available.

The downlink scheduling process currently in operation for the RADARSAT-2 mission makes use of a greedy type algorithm followed by human intervention where necessary. The operators of RADARSAT-2 are facing increased demand for its imagery and are exploring various methods to improve the efficiency of the downlink scheduling algorithms. We will discuss efficient heuristics to handle the downlink scheduling with the focus of achieving operational efficiency and reducing the number of unsuccessful downlink requests.

The satellite image downlink scheduling problem (SIDSP) and its variations have been studied by many authors in the past [7–9, 11, 12, 14, 19, 21–23]. However, each mission planning problem has its own restrictions and properties that can be exploited to achieve improvements over general purpose algorithms. The heuristic algorithm we propose integrates very large scale neighborhood (VLSN) search techniques [1, 2] and special ejection chain propagation schemes [15] that exploits the problem structure culminating in algorithms with superior computational performance and streamlined implementation challenges.

2 Problem Description and Model Development

The SIDSP is defined by one satellite that orbits the Earth (see Fig. 21.3)[1] to acquire images and a set G of stationary ground stations that receive images from the satellite for further processing.

As indicated earlier, the image acquisition schedule is already available and is part of the input. Let R be a set of n image-downlink-requests (IDR) that need to be processed (downlinked to a ground station) during the planning horizon of 24 h. For each IDR $r \in R$, a release time b_r, deadline e_r, downlink duration d_r, priority p_r, and a ground station $g_r \in G$ are prescribed. The interval $[b_r, e_r]$ is called the *time window* of IDR r.

It may be noted that image downlinking needs to satisfy a variety of constraints and it may not be possible to downlink all the available images. Thus, rejection of an IDR is allowed. An image that is not downlinked before its deadline is considered *unscheduled*. The SIDSP deals with finding an image downlink schedule so that an appropriate utility function is maximized. The utility function takes into account a number of parameters such as the number of downlinks scheduled, their priority values, and the earliness of the downlink start time relative to its time window. The SIDSP is NP-hard since several NP-hard machine scheduling problems are special cases of it. Hereafter, the terminology *satellite* refers to the spacecraft RADARSAT-2, which is the focus of this study.

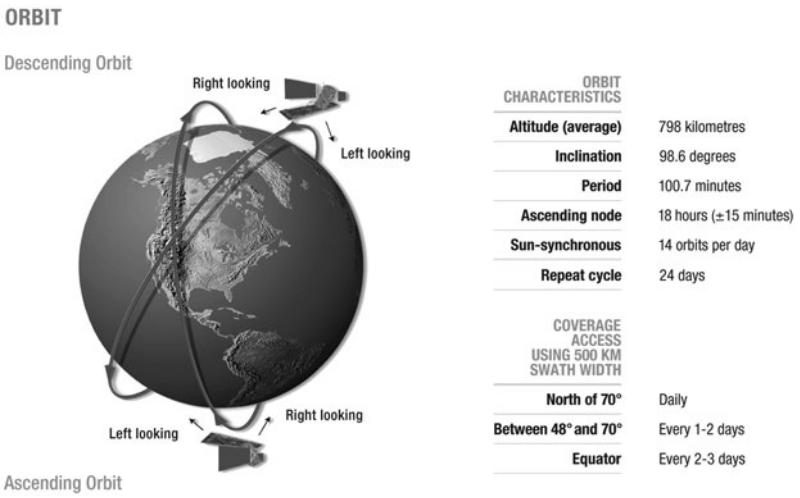


Fig. 21.3 Orbit characteristics of RADARSAT-2.[1] (© Canada Space Agency. Reproduced with permission)

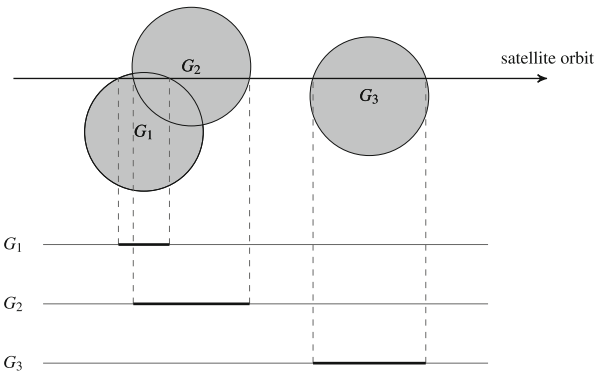


Fig. 21.4 Three ground stations G_1, G_2, G_3, and the associated *station visibility masks* are given as the shaded region. The *thick black line* segments represent the time intervals in which the satellite is within the visibility mask of the corresponding ground station

A downlink activity can be carried out only when the satellite is passing over the ground station coverage area, called *station visibility mask* (see Fig. 21.4).

The satellite has two antennas that are used for downlinking and each ground station $g \in G$ has one or two channels that are used for receiving the downlinked image. If the ground station has two channels, two downlinks can happen in parallel (simultaneously) under certain circumstances which will be discussed later. In our case study, approximately two-thirds of the ground stations have one-channel capability and one-third of the stations have two-channel capability. Ground station channels work independently. Satellite antennas also work independently and are capable of downlinking to two different ground stations simultaneously. Figure 21.5 shows the image of a ground station and its antenna.

Fig. 21.5 A ground station with its antenna. (Reproduced by permission of MacDonald, Dettwiler and Associates Ltd. (MDA). © MDA Ltd.)

Ground stations are also classified based on their transmission power. Let G_1 be the set of ground stations that are in *half-power setting* and G_2 be the set of ground stations that are in *full power setting*. Then $G = G_1 \cup G_2$ and $G_1 \cap G_2 = \emptyset$. When two images are downlinked one after another to stations with the same power setting, there must be a gap of δ seconds between the two downlinks. When two images are downlinked consecutively to two stations with different power settings, the required gap between the downlinks is $\Delta > \delta$ seconds. For the full power ground stations, only one downlink is allowed at a time, regardless of the number of channels.

An image downlinking can be carried out in *pass-through mode* or *store-forward mode*. In pass-through mode, an image is downlinked as it is being acquired. This is possible only when the area to be imaged is within the visibility mask of the ground station to which it is downloaded. In store-forward mode, images taken earlier are stored in the satellite recorder and are downlinked as opportunity arises.

The visibility mask of a ground station $g \in G$ can be represented by a collection V_g of nonoverlapping time intervals. We call V_g the *normal visibility mask*. Certain downlink requests require better reliability and in such cases we use a finer visibility mask V_g^1 which we call *high-reliability visibility mask*. Thus, for each $g \in G$, we have normal and high reliability visibility masks.

Let t_r be the scheduled downlink start time of IDR $r \in R$. If the IDR r is unscheduled, the value of t_r is undefined and this is denoted by $t_r = \varnothing$. The set $S = \{t_r : r \in R\}$ determines the solution to the SIDSP and it is called a *schedule*. Let $R(S) = \{r : r \in R \text{ and } t_r \neq \varnothing\}$ be the set of scheduled requests under S.

Since we are interested in (1) scheduling more IDRs with higher priority, and (2) scheduling an IDR as early as possible, we used objective function $f(S) = \sum_{r \in R(S)} p_r \left(1 - \alpha \cdot \frac{t_r - b_r}{e_r - d_r - b_r}\right)$, where $0 \leq \alpha \leq 1$ is a parameter reflecting the importance of scheduling each request downlink as early as possible. (Note that when

$\alpha = 0$, the objective function reduces to maximizing $\sum_{r \in R(S)} p_r$.) Thus, our model for the SIDSP reduces to

$$\text{Maximize} f(S) = \sum_{r \in R(S)} p_r \left(1 - \alpha \cdot \frac{t_r - b_r}{e_r - d_r - b_r}\right) \tag{21.1}$$

$$\text{subject to } S \in \mathbb{F},$$

where $\mathbb{F}$ is the collection of all schedules S satisfying the following constraints and assumptions:

1. A downlink cannot start earlier than the release time of the corresponding request and must be finished by its deadline. If a downlink cannot be done within this time window, the request is unscheduled.
2. If there are more than one requests to downlink one image to different ground stations, either all or none must be scheduled.
3. There must be a gap of at least δ seconds between two consecutive downlinks under the same power setting.
4. The satellite antennas work either in half-power setting or full-power setting. The transition from one power setting to another requires $\Delta > \delta$ seconds. During the transition, none of the two antennas can work. Thus, if two consecutive requests are scheduled to two different ground stations with different power settings, then there has to be a time gap of at least Δ seconds between these downlinks.
5. If a satellite antenna is in the full-power setting, then only this one antenna can work and the other has to be idle. For both of the satellite antennas to be working independently, the antennas have to be in the half-power setting. An antenna in the full-power setting is used if and only if the corresponding ground station is in the full-power setting.
6. We assume that the satellite is in the half-power setting at the beginning of the planning horizon and guarantee that it is in the half-power setting at the end. We also assume that downlinking (in half-power setting) can start right from the beginning of the planning horizon, i.e., there was no downlink activity for at least δ seconds before the beginning of the planning horizon.
7. No downlink activity happens outside the planning horizon.
8. Once a downlink starts, it cannot be preempted, i.e., the start time of a downlink has to satisfy the condition that once a downlink to a ground station starts, it can (and must) be completed within that time window.
9. A pass-through request is considered as *urgent* and, if possible, has to be scheduled at the earliest possible time in its time window.

Hereafter we use the terminologies IDR and request interchangeably. Note that our objective function does not explicitly distinguish between urgent and nonurgent requests. However, this is taken care off by running our algorithm in two phases, where in the first phase all urgent requests are scheduled followed by scheduling of nonurgent requests in the second phase.

3 Development of the Algorithms

Our solution approach is based on a local search with an ejection chain neighborhood. A high-level description of the algorithm is given as follows:

Preprocessing:	This step modifies the data to handle the gap constraint (3).
Urgent requests processing:	This step schedules all the urgent requests using the ejection chain algorithm.
Nonurgent requests processing:	Starting with the fixed downlink schedule of urgent requests, we schedule nonurgent requests using the ejection chain algorithm, while maintaining the scheduled urgent requests intact.

3.1 Preprocessing

We first perform a preprocessing step which modifies the data to simplify the presentation of the algorithm and efficiently handle some of the constraints. One of the constraints of the SIDSP is to have a gap of at least δ units between two consecutive downlinks. To accommodate this, we add δ to d_r and e_r for all requests $r \in R$. Also, we increase by δ the upper bounds of each of the time intervals in normal and high reliability visibility masks for all $g \in G$. Thus, even if two downlinks are scheduled such that one is followed immediately by another, a gap of size δ between the downlinks is guaranteed.

Note that such an adjustment in the data prohibits some downlink activity at the end of the planning horizon which otherwise could have been scheduled. One can increase the end of the planning horizon by δ to take care of this. In that case, the conflicts between downlinks at the end of the previous planning horizon and the beginning of the current planning horizon can be taken care of by using appropriate perturbations and careful data handling.

Further, we replace Δ by $\Delta - \delta$ because the gap between downlinks to the stations when different power settings are used will always include the additional δ units of time we added. Hereafter, we assume that problem data is modified to reflect the changes discussed above and, hence, we can ignore explicit treatment of constraints (3).

3.2 Schedule Generator

This section describes the greedy algorithm we used to generate a schedule from a given ordered subset of requests R. Let us now discuss some terminology and notations to simplify the presentation of the algorithm and its validity proof.

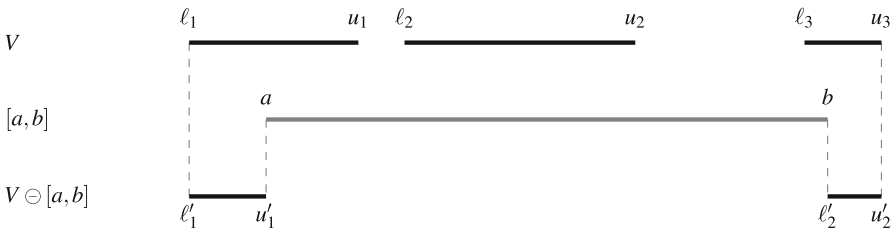


Fig. 21.6 Subtracting the interval $[a, b]$ from V

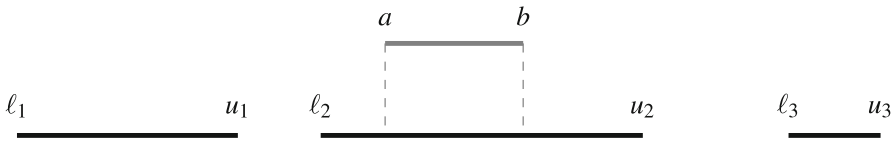


Fig. 21.7 $[a, b] \Subset V$

Let V be a finite set where elements of V are nonintersecting intervals. We call sets of the type V *interval set*. Since the elements of V are nonintersecting intervals, V can also be viewed as an ordered set with the natural order produced by the position of these intervals on the real line. Thus, V is represented as $V = \{[\ell_1, u_2], [\ell_2, u_2], \ldots, [\ell_v, u_v]\}$ where $\ell_1 < u_1 < \ell_2 < u_2 < \cdots < \ell_v < u_v$ and $v = |V|$. We refer to the ith interval in V as V_i.

Consider an arbitrary interval $[a, b]$. Let us introduce sets of intervals B, C, and D as follows:

$$B = \{[\ell_j, u_j] \in V : \ \ell_j < b \text{ and } u_j > a\},$$

$$C = \{[\ell_j, a] : \text{there exists } j \text{ such that } [\ell_j, u_j] \in V \text{ and } \ell_j < a < u_j\}, \text{ and}$$

$$D = \{[b, u_j] : \text{there exists } j \text{ such that } [\ell_j, u_j] \in V \text{ and } \ell_j < b < u_j\}.$$

Then subtracting the interval $[a, b]$ from V generates the set $V \ominus [a, b]$ of nonintersecting intervals given by

$$V \ominus [a, b] = (V \setminus B) \cup C \cup D.$$

The notation $V \ominus [a, b]$ is read as V minus interval $[a, b]$ (see Fig. 21.6).

An interval $[a, b]$ said to be a *subinterval* of the intervals set V if there exists k such that $\ell_k \le a < b \le u_k$ and $[\ell_k, u_k] \in V$. This relationship is denoted by $[a, b] \Subset V$ (see Fig. 21.7). (Note that $[a, b]$ need not be a member of V.)

Let X and Y be two interval seats. Then the *intersection* of X and Y, denoted by $X \Cap Y$, is defined as $X \Cap Y = \{x \cap y : \ x \in X, \ y \in Y \text{and } x \cap y \neq \emptyset\}$ (see Fig. 21.8).

Let R^* be an ordered subset of R. Given R^*, we now present a *greedy algorithm* to schedule requests in R^* following the order prescribed for R^*. The algorithm maintains several indicator interval sets representing channel availability at ground

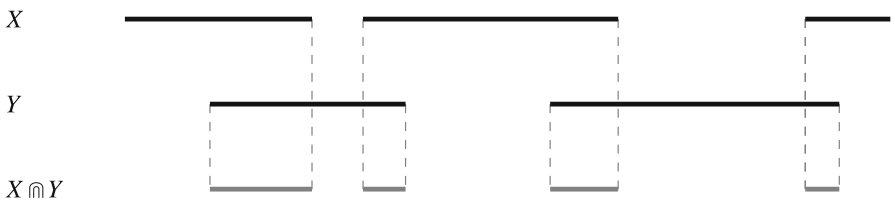


Fig. 21.8 Intersection of interval sets

stations and antenna availability on the satellite. After scheduling a request, the algorithm updates these indicator sets.

Let A be an interval set that represents the time intervals when both satellite antennas are available. Let H and F be interval sets indicating the time intervals when half-power and full-power downlinks can happen, respectively. Let Q_g and Q_g^1 be interval sets indicating the availability of the ground station $g \in G$ in normal and high-reliability visibility, respectively. Note that we need only one pair of indicator sets for a two-channel ground station. Indeed, the number of channels in this case is not limiting since the number of simultaneous downlinks is constrained by the number of antennas. Hence, only the visibility of the ground station need to be considered.

A high-level description of the greedy algorithm is given below. The notation $U \leftarrow V$ indicates the standard assignment statement. (i.e., assign the contents of V to U.)

1. Let $R^{**} \leftarrow R^*$.
2. Initialize the sets A, H and F to $\{[0, 24\text{ h}]\}$ which is the planning horizon. Initialize the set Q_g to the normal visibility mask of g and Q_g^1 to the high-reliability visibility mask of g for every ground station $g \in G$. Note that for every $[\ell_j^1, u_j^1] \in Q_g^1$ there is an interval $[\ell_k, u_k] \in Q_g$ such that $\ell_k < \ell_j^1 < u_j^1 < u_k$.
3. Select the first request $r \in R^{**}$.
4. Suppose that r is downlinked to a half-power station $g \in G_1$. Let $Q \leftarrow Q_g$ if r requires normal reliability and $Q \leftarrow Q_g^1$ otherwise. Choose the smallest $i \in \{1, 2, \ldots, |H|\}$ and $j \in \{1, 2, \ldots, |Q|\}$ such that $|H_i \cap Q_j \cap [b_r, e_r]| \geq d_r$. If no such i and j exist, the request r is unscheduled and $t_r \leftarrow \varnothing$. Otherwise $t_r \leftarrow x$, where $H_i \cap Q_j \cap [b_r, e_r] = [x, y]$.
   If $t_r \neq \varnothing$, the indicator sets are updated as follows. Let $X = [t_r, t_r + d_r]$. Apply $X \leftarrow X \ominus [\ell_i, u_i]$ for every $[\ell_i, u_i] \in A$. Now X contains all the intervals inside $[t_r, t_r + d_r]$ during which exactly one antenna was available. Apply $H \leftarrow H \ominus [\ell_i, u_i]$ for every $[\ell_i, u_i] \in X$. Also apply $A \leftarrow A \ominus [t_r, t_r + d_r]$ and $F \leftarrow F \ominus [t_r - \Delta, t_r + d_r + \Delta]$. Finally, if g is a one channel ground station, update $Q_g \leftarrow Q_g \ominus [t_r, t_r + d_r]$ and $Q_g^1 \leftarrow Q_g^1 \ominus [t_r, t_r + d_r]$.
5. Suppose that r is downlinked to a full-power station $g \in G_2$. Let $Q \leftarrow Q_g$ if r requires normal reliability and $Q \leftarrow Q_g^1$ otherwise. Choose the smallest $i \in \{1, 2, \ldots, |F|\}$ and $j \in \{1, 2, \ldots, |Q|\}$ such that $|F_i \cap Q_j \cap [b_r, e_r]| \geq d_r$. If no such i and j exist, the request r is unscheduled and $t_r \leftarrow \varnothing$. Otherwise, $t_r \leftarrow x$, where $F_i \cap Q_j \cap [b_r, e_r] = [x, y]$.

If $t_r \neq \varnothing$, the indicator sets are updated as follows: $F \leftarrow F \ominus [t_r, t_r + d_r]$ and $H \leftarrow H \ominus [t_r - \Delta, t_r + d_r + \Delta]$. Note that it is not necessary to update the indicator set A since no downlink can happen if neither H nor F is available.
6. Update R^{**} to $R^{**} \setminus \{r\}$ and, if $|R^{**}| > 0$, proceed to Step 3.
7. For every pair (r, u) of dual requests, if either $t_r = \varnothing$ and $t_u \neq \varnothing$ or $t_r \neq \varnothing$ and $t_u = \varnothing$, remove both r and u from R^* and proceed to Step 2.
8. Assign the downlinks to the particular ground station channels and satellite antennas.

Theorem 1 *The greedy algorithm terminates with a feasible schedule of requests in R^* in $O(n^3)$ time, where $n = |R^*|$.*

Proof Our updating scheme of the intervals sets A, H, and F ensures that no antenna conflict arises and all the downlinks happen within the planning horizon. By subtracting $[t_r - \Delta, t_r + d_r + \Delta]$ from F for every half-power downlink request r, we guarantee that no full-power downlink will happen within Δ units from the downlink r. Similarly, no half-power downlink can happen within Δ units from a full-power downlink r. Also, the updating of the indicator sets Q_g and Q_g^1 guarantees that no channel conflicts occur and the downlinks obey visibility mask constraints. The preprocessing of data assures that there is a gap of at least δ units between two consecutive downlinks. Finally, the dual-requests constraint is satisfied due to Step 7 of the algorithm. Thus, the schedule generated is feasible. ■

Let us now analyze the complexity of the algorithm. The primary operations in each iteration are: (1) to find the smallest i and j to satisfy certain condition and (2) update the indicator sets A, H, F, Q_g, and Q_g^1. Note that the size of the indicator sets Q_g and Q_g^1 for some $g \in G$ may increase but it is bounded by $O(n_g)$ where n_g is the number of requests to be scheduled to the station g. Similarly, the sizes of the indicator sets A, H, and F are limited by $O(n)$. Thus, operation (1) for these sets can be performed in $O(n)$ time by simultaneous scanning of the sets. Operation (2) can also be performed in $O(n)$ time. Indeed, we only need to update a fixed number of indicator sets, and updating each of them takes $O(n)$ time (for a downlink to a half-power setting ground station, manipulating with X also needs only $O(n)$ time; note that each of the $H \leftarrow H \ominus [\ell_i, u_i]$ for $[\ell_i, u_i] \in X$ operations needs only $O(1)$ since all of these operations affect only one interval in H). The number of iterations is at most $O(n^2)$ and, thus, the algorithm complexity is $O(n^3)$.

It may be noted that the greedy algorithm may not schedule all requests in R^*.

3.3 Construction Heuristics

The greedy scheme discussed above can be used to obtain various construction heuristics for the SIDSP. Choose $R^* = R$ and depending on how we order elements of R^* we obtain different construction heuristics. We considered sorting the requests based on one or more of the following fields:

1. Priority p_r.
2. Time window range $w_r^* = \lfloor e_r - b_r - d_r \rfloor$. Note that if we do not use rounding, time windows for all the requests could be different and, thus, sorting using a secondary criterion could become somewhat superfluous.
3. Downlink duration d_r.

After extensive computational experiments using various combinations of the above fields to sort requests, it is determined that sorting elements of R^* with respect to p_r first, ties are broken by sorting with respect to w_r^* values and further ties, if any, are broken by d_r values provides the best performance. The resulting solution is used as the initial solution for our improvement heuristic that uses ejection-chain neighborhoods. This ejection chain algorithm is described in the next section.

3.4 Ejection-Chain Improvement Heuristic

Ejection chain is a terminology introduced by Glover [15] in the context of meta-heuristic algorithms where transition from one solution to another is achieved using compound moves consisting of a sequence of ejection and replacement operations. We now show how ejection chains can be used to achieve possible improvements over greedy solutions.

Note that the quality of the solution produced by the greedy algorithm depends on the ordering of elements in R^*. By repeating the algorithm with different orderings we get possibly different solutions and the overall best solution can be chosen. However, exhaustive searching over all such orderings is impractical. Our ejection-chain algorithm chooses "good" ordering for R^* so that the output generated is of a high quality. Let $\wp$ be the set of all permutations of elements of R. For any $\pi \in \wp$, let S_π be the schedule produced by the greedy algorithm by choosing $R^* = R$ and the ordering of elements of R is given by π. Thus, we concentrate on solutions $\{S_\pi : \pi \in \wp\}$ to design our ejection algorithm. In this sense, we consider $\wp$ as the family of feasible solutions of the SIDSP. Thus, to solve the SIDSP, we essentially need to solve the *downlink requests permutation problem* (DRPP) given by:

$$\text{Maximize } \phi(\pi) = f(S_\pi)$$
$$\text{subject to } \pi \in \wp.$$

Note that the objective function $f(.)$ is defined in Eq. (21.1).

Ejection chain-based neighborhoods are commonly used in developing VLSN search algorithms [1, 2] for complex combinatorial optimization problems. For example, the well known Lin–Kernighan heuristic and its variations are ejection-chain algorithms that are extremely efficient for the traveling salesman problem [13]. We now use the idea of the ejection chains [15] to develop a simple and effective heuristic to solve DRPP (and hence, the SIDSP).

The data structure used in our ejection-chain algorithm is a pair (π, h), where π is a permutation of all the requests R and $h \in \{1, 2, \ldots, n\}$ is a position in this

permutation called *hole*. Here $n = |R|$. The "hole" can be viewed as an empty position generated by "ejecting" the request corresponding to the position h of π. At the end of the ejection-chain propagation, this request will be inserted to the available "hole" at that stage. In a general stage of the ejection-chain propagation, we can "eject" any request, say $\pi(r)$, from its position r and insert into the position of "hole," making r a new "hole," etc. The process can be continued to an appropriate depth level which is an input parameter for the algorithm. The resulting ejection chain is closed by inserting the first ejected request to the last "hole" generated by the ejection process. In order to calculate the objective $\phi(\pi, h)$ of the "solution" (π, h), we choose $R^* = R$, order the element of R^* according to the permutation π and then remove the hth element from R^*. Then, by applying the greedy scheduling algorithm, we obtain a schedule (excluding request at position h, and possibly other requests that are not scheduled by greedy) and calculate its objective function value using Eq. (21.1).

The basic move in our ejection chain algorithm is swapping of an element with the "hole," i.e., swapping $\pi(h)$ with $\pi(i)$ and updating h with i for some $i \neq h \in \{1, 2, \ldots, n\}$. There are $n - 1$ options for this move, and we select the one that produces the best solution (π, h). Every time we obtain a new solution (π, h), we also evaluate the schedule S_π. If S_π is better than the best schedule known so far, we save it. For details of our ejection-chain algorithm see Algorithms 1 and 2.

Algorithm 1 uses a procedure *improvement* (π, h, σ, d) (described in Algorithm 2) which takes input a permutation π, a hole position h, the current best solution σ, and the depth d of the ejection-chain construction and outputs an improved solution (as a permutation π), if one is found.

Algorithm 1 Ejection Chain Algorithm.

Require: Requests R.
Require: Priorities p_r and time window durations w_r for each $r \in R$.
Require: Objective functions $\phi(\pi)$ and $\phi(\pi, h)$.
Create a permutation π of $(R_1, R_2, \ldots, R_n)$ such that $p_{\pi(i)} > p_{\pi(i+1)}$ (or $p_{\pi(i)} = p_{\pi(i+1)}$ and $\lfloor w_{\pi(i)} \rfloor < \lfloor w_{\pi(i+1)} \rfloor$, or $p_{\pi(i)} = p_{\pi(i+1)}$, $\lfloor w_{\pi(i)} \rfloor = \lfloor w_{\pi(i+1)} \rfloor$ and $d_{\pi(i)} < d_{\pi(i+1)}$) for every $i = 1, 2, \ldots, n-1$.
Save π to a permutation σ.
Initialize the idle counter c with n. c keeps track of the number of non-improving iterations.
Initialize the hole position h with 1.
While there are no more then n consecutive non-improving iterations:
while $c > 0$ **do**
  if *improvement* $(\pi, h, \sigma, 10) = 1$ **then**
    Reset the idle counter c with n.
    Save π to σ.
  else
    Reduce the idle counter c by one.
  end if
  Go to the next hole position: if $h = n$ set $h \leftarrow 1$ otherwise $h \leftarrow h + 1$.
end while
return σ.

4 Data Analysis and Generation of Test Instances

The algorithms developed in this paper were tested using real data. We used 13 *low-density instances* (relatively low number of requests per day) collected between October 25 and December 13, 2011, each containing approximately 100 requests per planning horizon. We also used 31 *high-density instances* (large number of requests per day) collected in August 2011, each containing approximately 300 requests. There are about ten ground stations involved in each instance and these ground stations are typically visible during the planning horizon from 4 to 10 times.

For the low-density data, two solutions were given to us: *machine scheduled* and *human rescheduled*. The machine-scheduled solutions are generated by the algorithm currently in use for mission planning. Such a solution may be infeasible and the human-rescheduled solution modifies the machine-scheduled solution to remove violations and to satisfy some additional considerations known to the operators at that time. Such data (and solution) may be imprecise. For example, our model uses crisp visibility mask boundaries but a human operator may use his/her judgement to schedule a job even if a job goes beyond the prescribed visibility mask by very small amounts but such a solution will be infeasible as per our model.

Algorithm 2 Ejection Chain Algorithm: *improvement* (π,h,σ,d).

Require: Permutation π with and a hole position h.
Require: The best solution σ found so far.
Require: The remaining search depth d, i.e., potentially the longest ejection chain
  if $d = 0$ **then**
    return 0.
  end if
  Save the objective value of the partial solution $\phi(\pi,h)$ to ϕ_{p}.
  Initialize j with ϕ .
  for $i \leftarrow 1,2,...,\ h-1,h+1,...,\ n$ **do**
    Swap $\pi(h)$ with $\pi(i)$.
    if $\phi(\pi) > \phi(\sigma)$ **then**
      Save the solution π to σ .
      return 1.
    end if
    if $\phi(\pi,\mathrm{i}) > \phi_{\mathrm{p}}$ **then**
      Update j with i.
      Update ϕ_{p} with $\phi(\pi,i)$.
    end if
    Swap $\pi(h)$ with $\pi(i)$.
  end for
  if $\mathrm{j} \neq \phi$ **then**
    Apply the modification by swapping $\pi(h)$ and $\pi(j)$.
    if *improvement* $(\pi,j,\sigma,d-1) = 1$ **then**
      return 1.
    end if
  end if
  return 0.

The high-density instances were given to us as a single file containing data for the month of August 2011. In particular, we were provided with a table of all the requests and the schedule generated by human intervention. Since we were not provided any explicit instances for each planning horizon of 24 h, we used the following rules to construct the instances. Let T_i be the planning horizon of the ith instance, $i = 1, 2, \ldots, 31$.

1. If a request r is present in the provided schedule, we assign it to the ith instance, where $t_r \in T_i$.
2. If an urgent request r is withdrawn in the provided schedule, we assign it to the first instance where it can be downlinked.
3. If a regular request r is withdrawn in the provided schedule, we assign it to the ith instance, where i is chosen such that the length of the interval $T_j \cap [b_r, e_r - d_r]$ is maximized over $i \in \{1, 2, \ldots, 31\}$ (if there are several such is, we chose the smallest one).

Several of the 31 instances were excluded from our test bed because of a significant amount of violations found in the provided solution. These are instances corresponding to `Aug 7`, `Aug 8`, and `Aug 31`.

5 Computational Experiments

The ejection chain algorithm is coded in `C++` and tested on a PC with Intel i7-2600 CPU. The low- and high-density instances as discussed above were used as our test bed. We compared our results with the solutions provided by our industrial partner. The summary of our experimental results on low-density instances is given in Table 21.1.

Each row of the table corresponds to one problem instance. For each of the algorithms (M-S for machine scheduled, H-R for human rescheduled, and EC for ejection chain) and for each of the instances we provided how much worse the solution is compared to the best-known solution for this instance ("To the best (%)"), the number of unscheduled urgent requests ("Urg. unsch."), the total number of unscheduled requests ("Total unsch."), the average delay of the urgent downlinks ("Avg. urg. del. (s)") and the average delay of the regular downlinks ("Avg. reg. del. (s)"). Since the machine-scheduled solutions are sometimes infeasible, we consider a downlink to be scheduled if it falls into the planning horizon and is not present in the violations table provided with the machine-scheduled solution. The delay of a request r is measured as $t_r - \tau_r$, where τ_r is the earliest time when r can be downlinked.

Note that the comparison of our schedules to the human-rescheduled solutions may be incorrect. Indeed, we may not know some of the constraints that are taken into consideration by the operator. Thus, we focus on the comparison to the machine-scheduled solutions.

Table 21.1 Comparison of the given machine-scheduled solutions (*M-S*), human-rescheduled solutions (*H-R*), and the results of our ejection-chain algorithm (*EC*) for the low-density instances

			To the best (%)			Urg. unsch.			Total unsch.			Avg. urg. del. (s)			Avg. reg. del.(s)		
Instance	n	Urg.	M-S	H-R	EC	M-S	H-R	EC	M-S	H-R	EC	M-S	H-R	EC	M-S	H-R	EC
Oct 25	110	31	1.7	1.7	0.0	0	0	0	1	1	0	0.4	0.3	0.0	712	762	135
Oct 28	108	34	2.1	1.2	0.0	0	0	0	3	1	0	0.4	0.5	0.0	445	1001	229
Nov 1	115	29	1.9	1.9	0.0	0	0	0	1	1	0	0.7	0.7	0.0	881	881	94
Nov 15	105	25	2.5	2.5	0.0	0	0	0	1	1	0	0.7	0.7	0.0	1613	1613	513
Nov 17	104	32	1.5	2.0	0.0	0	0	0	1	2	0	6.9	0.4	0.0	711	701	184
Nov 22	105	33	1.1	1.1	0.0	0	0	0	1	1	0	0.3	1.3	0.0	201	202	119
Nov 23	90	30	1.2	1.1	0.0	0	0	0	1	1	0	0.4	3.1	0.0	371	379	79
Nov 29	108	31	1.0	1.0	0.0	0	0	0	1	1	0	4.5	4.5	0.0	234	234	102
Dec 1	110	28	1.9	1.6	0.0	0	0	0	2	1	0	5.7	11.3	0.0	536	975	237
Dec 2	134	27	0.3	0.3	0.0	0	0	0	0	0	0	0.4	0.4	0.0	564	564	65
Dec 7	108	36	1.1	1.1	0.0	0	0	0	1	1	0	1.4	0.5	0.0	573	575	151
Dec 13	94	24	2.1	2.1	0.0	0	0	0	2	2	0	0.2	0.2	0.0	566	566	40
Average	107.6	30.0	1.5	1.5	0.0	0.0	0.0	0.0	1.3	1.1	0.0	1.8	2.0	0.0	617	704	162

Table 21.2 Comparison of the human-rescheduled (*H-R*) solutions with the results of our ejection-chain algorithm (*EC*) for the high-density instances

Instance	n	Urg.	To the best, %		Urg. unsch.		Total unsch.		Avg. urg. del., s		Avg. reg. del., s	
			H-R	EC	H-R	EC	H-R	EC	H-R	EC	H-R	EC
Aug 1	211	34	13.5	0.0	0	0	53	19	0.3	0.0	2013	457
Aug 2	301	35	20.5	0.0	0	0	123	67	0.6	0.0	3788	2725
Aug 3	324	42	20.7	0.0	0	0	120	48	0.5	0.0	2329	2248
Aug 4	294	41	21.1	0.0	2	0	116	52	0.5	0.0	2025	1248
Aug 5	260	36	13.3	0.0	0	0	85	49	0.5	0.0	2621	1021
Aug 6	356	46	20.0	0.0	2	1	130	61	0.4	0.0	2555	1563
Aug 9	259	38	18.7	0.0	3	0	87	30	92.4	0.0	2448	1867
Aug 10	304	54	20.9	0.0	1	0	98	33	0.4	0.0	2230	1338
Aug 11	278	39	19.4	0.0	3	0	90	36	0.5	0.5	2463	904
Aug 12	230	25	16.8	0.0	0	0	80	28	0.5	0.0	2880	2096
Aug 13	324	41	16.7	0.0	1	0	101	38	0.4	0.0	2045	868
Aug 14	285	34	18.8	0.0	4	3	104	54	0.4	0.0	1558	818
Aug 15	259	30	18.6	0.0	0	0	93	43	0.4	0.0	1692	1303
Aug 16	299	39	29.5	0.0	4	0	139	54	0.4	0.0	1374	1849
Aug 17	303	30	23.0	0.0	0	0	128	65	0.5	0.0	2664	1193
Aug 18	275	33	20.6	0.0	2	1	115	54	0.4	0.0	2557	1819
Aug 19	254	30	18.8	0.0	0	0	93	34	0.4	0.0	2210	1817
Aug 20	295	39	19.4	0.0	0	0	96	35	0.4	0.0	2768	1124
Aug 21	272	24	18.7	0.0	0	0	100	41	0.4	0.0	2531	1479
Aug 22	253	36	18.0	0.0	0	0	91	45	0.5	0.0	1737	1486
Aug 23	307	44	18.6	0.0	0	0	111	45	0.3	0.0	2450	1421
Aug 24	300	40	22.9	0.0	1	0	122	56	0.4	0.0	1465	1146
Aug 25	269	40	36.2	0.0	11	0	112	48	2208.0	0.0	6051	1664
Aug 26	301	37	19.2	0.0	0	0	116	44	0.5	0.0	1955	2152
Aug 27	282	40	18.5	0.0	0	0	89	33	0.4	0.0	2720	921
Aug 28	282	34	21.9	0.0	5	0	108	41	0.5	8.0	2765	1014
Aug 29	267	36	20.5	0.0	3	1	95	46	0.4	0.0	2991	1393
Aug 30	315	44	21.0	0.0	1	0	116	46	0.3	0.0	2081	1582
Average	284.3	37.2	20.2	0.0	1.5	0.2	104.0	44.5	82.6	0.3	2463	1447

The computational result for the low-density instance show that the solutions produced by our algorithm dominate the provided machine-scheduled solutions for all of the instances. In particular, our algorithm always schedules all the downlinks while usually there were several unscheduled requests in the given solutions. Our algorithm never delays the urgent requests while it sometimes happens in the given machine-scheduled solutions, i.e., there are large delays (around 3 min) for the `Nov 17` and `Nov 29` instances. In each case, the delay is mainly caused by only one downlink that the algorithm failed to schedule to the beginning of the time window.

Note that, since our algorithm schedules all the requests for every low-density instance, the optimality of schedules does not depend on the value of the constant α (see Sect. 2).

The running time of our algorithm is always very reasonable and never exceeds 10 s in our experiments.

Computational results for high-density instances are given in Table 21.2.

Table 21.3 Comparison of the given human-rescheduled solution with the results of our construction heuristic and our ejection-chain algorithm for the rolling instances

Algorithm	Objective	Scheduled	Unscheduled	Avg. delay (s)	Running time (h)
Human-rescheduled	219.1	5056	3428	9486.6	—
Construction	293.8	6513	1971	12,286.6	0.0
Ejection chain	304.1	7084	1400	8995.6	3.9

Unlike for the low-density instances, our ejection-chain algorithm is not able to schedule all the requests in the high-density instances. However, it schedules 60 requests more, on average, than in the given machine-scheduled solutions. It also schedules more urgent requests and provides a smaller average downlink delay. The running time of the algorithm is below 4 min on average and it never exceeds 10 min.

Another experiment that we conducted was to schedule as many requests from the high-density instances as possible. For this purpose we defined the new instances from the high-density instance as follows. The `Aug 1` instance includes all the requests that can be downlinked on August 1. The `Aug 2` instance includes all the requests that can be downlinked on August 2 except the requests scheduled for `Aug 1`. The `Aug 3` instance includes all the requests that can be downlinked on August 3 except the requests scheduled for `Aug 1` and `Aug 2`, etc.

We call such instances *rolling*. Note that rolling instances may be different for different algorithms, and thus, comparison of the performance of the algorithms for every particular day-instance is meaningless.

The results of our experiment with the rolling instances are reported in Table 21.3.

It takes around 4 h for our ejection-chain algorithm to schedule 7086 out of 8484 requests for the entire month of August, compared to 5056 requests scheduled in the human-rescheduled solution. Note that our algorithm also improved the average delay and, hence, the quality of the solution.

Our construction heuristic proceeds in almost no time and schedules more requests than in the given human-rescheduled solutions. However, the average delay in this solution is higher.

Although such an experiment may not reflect the real procedure of scheduling downlink requests, it shows the potential for improving downlink schedules.

5.1 The Parameter α

Recall that the objective function for the SIDSP uses a parameter α that represents the relative preference of the downlink delays over unscheduled requests. Our ejection-chain algorithm was able to schedule all the requests for the low-density instances and, hence, the solutions obtained for these instances were almost independent of the value of α. In contrast, for the high-density instances, careful selection of the value of α was crucial in achieving high-quality solutions. In Fig. 21.9 we show how

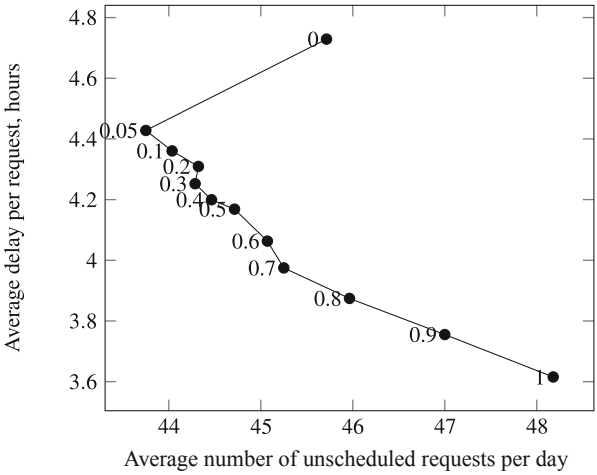


Fig. 21.9 How the value of α influences the performance of our ejection-chain algorithm. The experiments were conducted for the high-density instances

the performance of our ejection-chain algorithm depends on the value of α for the high density instances.

The tendency is that as α increases, the delay in scheduling a request decreases and the number of unscheduled requests increases. Hence, α can be used as a parameter to adjust the trade-off between the number of scheduled requests and the average delay.

Note that the performance of the algorithm is poor for $\alpha = 0$. This is because the objective in this case does not depend on the delays making the search landscape more discrete, and thus, complicated for optimization. It is also worth mentioning that the difference in the solutions obtained for different values of α is relatively small.

One of our assumptions was that the rejection of requests is allowed because it provides for a larger number of requests to be submitted to the scheduler in order to efficiently utilize the resources. If the rejection is not allowed, the large number of requests as an input to the scheduler would create an infeasible scheduling problem. Selecting the requests to schedule from a larger set of requests is a common approach when aiming to maximize resource utilization.

6 Concluding Remarks

In this chapter, we formalized the satellite downlink scheduling problem and proposed an efficient heuristic solution approach. Computational experiments show that our algorithm clearly outperforms the currently implemented algorithm, and the running time of our algorithm is practically negligible to reasonable.

It is worth noting that we understand that the proposed problem definition may not be precise and some additional constraints may be found out later. Our approach, however, is very flexible and we believe that it can easily incorporate potential additional constraints.

7 Exercises

1. Show that the image downlink scheduling problem is NP-hard.
2. Show that $V \ominus [a,b] = \bigcup_{[\ell,u]\in V} ([\ell,u] \ominus [a,b])$, where

$$[\ell,u] \ominus [a,b] = \begin{cases} \{[\ell,a],[b,u]\} & \text{if } \ell < a < b < u \\ \{[\ell,\min\{a,u\}]\} & \text{if } \ell < a \\ \{[\max\{b,\ell\},u]\} & \text{if } b < u \\ \emptyset & \text{otherwise.} \end{cases}$$

3. Describe an appropriate data structure for representing an interval set V and discuss the complexity of computing $V \ominus [a,b]$.
4. Discuss the complexity of computing $X \Cap Y$ where X and Y are interval sets assuming appropriate representations of X and Y.
5. Show that the preprocessing step discussed in the chapter indeed takes care of constraint 3.
6. Identify some nontrivial polynomially solvable cases of the SIDSP by restricting the problem data.
7. Introduce diversification strategies within the ejection chain algorithm to possibly improve performance.
8. What can you say about the dominance number of the ejection-chain algorithm?

Acknowledgement This work was partially supported by an NSERC CRD grant awarded to Abraham P. Punnen and supported by MacDonald, Dettwiler and Associates Ltd. (MDA). We are thankful to MDA R&D managers Harold Zwick and Christian Nadeau, and RADARSAT-2 flight operation manager Philippe Rolland for their support in various aspects of this work. Extensive comments of Katta G. Murty on an earlier version of this chapter improved the presentation.

References

1. Ahuja, R. K., Ergun, O., Orlin, J. B., & Punnen, A. P. (2002). A survey of very large scale neighborhood search techniques. *Discrete Applied Mathematics, 123,* 75–102.
2. Ahuja, R. K., Ergun, O., Orlin, J. B., & Punnen, A. P. (2007). Very large scale neighborhood search: Theory, algorithms and applications. In T. Gonzalez (ed.), *Approximation algorithms and metaheuristics*. Boca Raton: CRC.
3. Barbulescu, L., Watson, J. P., Whitley, L. D., & Howe, A. E. (2004). Scheduling space-ground communications for the Air Force Satellite Control Network. *Journal of Scheduling*, *7*, 7–34.
4. Barbulescu, L., Whitley, L. D., & Howe, A. E. (2004). Leap before you look: An effective strategy in an oversubscribed scheduling problem. In *Proceedings of the nineteenth national conference on artificial intelligence*, San Jose, USA.
5. Bard, J. F., & Rojanasoonthon, S. (2006). A branch and bound algorithm for parallel machine scheduling with time windows and job priorities. *Naval Research Logistics, 53,* 24–44.
6. Beaumet, G., Verfaillie, G., & Charmeau, M.-C. (2011). Feasibility of autonomous decision making on board an agile Earth-observing satellite. *Computational Intelligence, 27,* 123–139.
7. Benoist, T., & Rottembourg, B. (2004). Upper bounds for revenue maximization in a satellite scheduling problem, *4OR, 2*, 235–249.

8. Bianchessi, N., & Righini, G. (2008). Planning and scheduling algorithms for the COSMO-SkyMed constellation. *Aerospace Science and Technology, 12,* 535–544.
9. Bianchessi, N., Cordeau, J.-F., Desrosiers, J., Laporte, G., & Raymond, V. (2007). A heuristic for the multi-satellite, multi-orbit and multi-user management of Earth observation satellites. *European Journal of Operational Research, 177*, 750–762.
10. Canada Space Agency. http://www.asc-csa.gc.ca/eng/satellites/radarsat2/applications.asp. Accessed on 1 July 2014; last modified on 28 Nov 2007.
11. Cordeau, J.-F., & Laporte, G. (2005). Maximizing the value of an earth observation satellite orbit. *Journal of the Operational Research Society, 56,* 962–968.
12. Gabrel, V., & Murat, C. (2003). Mathematical programming for Earth observation satellite mission planning. In T. A. Ciriani, et al. (Eds.), *Operations research in space and air*. Boston: Kluwer Academic.
13. Gamboa, D., Osterman, C., Rego, C., & Glover, F. (2006). *An experimental evaluation of ejection chain algorithms for the traveling salesman problem*. Technical report, School of Business Administation, University of Missisippi.
14. Globus, A., Crawford, J., Lohn, J., & Pryor, A. (2004). A comparison of techniques for scheduling earth observing satellites, In *Proceedings of the sixteenth innovative applications of artificial intelligence conference (IAAA-04)*, San Jose, USA.
15. Glover, F. (1996). Ejection chains, reference structures and alternating path methods for traveling salesman problems. *Discrete Applied Mathematics, 65,* 223–253.
16. Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research, 207,* 1–14.
17. Marinelli, F., Rossi, F., Nocella, S., & Smriglio, S. (2006). A Lagrangian heuristic for satellite range scheduling with resource constraints. *Optimization Online*.
18. Rojanasoonthon, S., & Bard, J. (2005). A GRASP for parallel machine scheduling with time windows. *INFORMS Journal on Computing, 17,* 32–51.
19. Sun, B., Wang, W., Xie, X., & Qin, Q. (2010). Satellite mission scheduling based on genetic algorithm. *Kybernetes, 39,* 1255–1261.
20. Vasquez, M., & Hao, J.-K. (2003). A logic-constrained knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite. *Computational Optimization and Applications, 7,* 87–103.
21. Verfaillie, G., Pralet, C., & Lemaitre, M. (2010). How to model planning and scheduling problems using constraint networks on timelines. *Knowledge Engineering Review, 25,* 319–336.
22. Wang, P., Reinelt, G., Gao, P., & Tan, Y. (2011). A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation. *Computers & Industrial Engineering, 61,* 322–335.
23. Zufferey, N., Amstutz, P., & Giaccari, P. (2008). Graph colouring approaches for a satellite range scheduling problem. *Journal of Scheduling, 11,* 263–277.

Chapter 22
An Integer Programming Model for the Ferry Scheduling Problem

Daniel Karapetyan and Abraham P. Punnen

1 Introduction

Routing and scheduling of public transport vehicles and other commercial vehicles is one of the most extensively studied areas in the operations research literature. Such problems include airline scheduling [5], scheduling of passenger and freight trains, transit bus routing and scheduling, and a variety of other vehicle routing problems [3, 4, 15, 16]. The routing and scheduling of passenger ferries is yet another problem in this class that received relatively little attention.

In a ferry scheduling problem, we are given a set of ports, a set of ferries, and a planning horizon (examples of a ferry and a terminal are given in Figs. 22.2 and 22.3). Then we need to find a routing and scheduling scheme for the ferries so that the travel demands emanating at the ports at different periods of the planning horizon are satisfied while the operating cost and passenger dissatisfaction are kept at a minimum. Note that the transition of travel demand from a lower time state to a higher time state is permitted but the reverse transition is not permitted. As reported in [7], periodic changes in ferry schedules are required for a variety of reasons. These include changing demographics within the service area, seasonal changes, changes in fleet size and characteristics, changes in port configuration, altered service level restrictions, responses to customer feedback, and/or changes in government regulations. Developing an "optimal" schedule is crucial in managing operating costs efficiently while maintaining sufficient level of satisfaction of the ferry users.

The online version of this chapter (doi: 10.1007/978-1-4939-1007-6_22) contains supplementary material, which is available to authorized users.

D. Karapetyan (✉)
ASAP Research Group, School of Computer Science, University of Nottingham, Nottingham NG8 1BB, UK
e-mail: daniel.karapetyan@gmail.com

A. P. Punnen
Department of Mathematics, Simon Fraser University, Surrey, BC V3T 0A3, Canada
e-mail: apunnen@sfu.ca

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6_22, © Springer Science+Business Media New York 2015

Even for scheduling problems that appear to be of smaller size (e.g., four ferries, seven ports with a 20-h planning horizon), the ferry scheduling problem is complex and requires systematic scientific approaches to achieve high-quality schedules and identifying operational bottlenecks. Fig. 22.1 provides a sample schedule taken from our case study using real data [7].

Lai and Lo [10] studied the ferry scheduling problem and provided an integer programming formulation with the assumption that all ferries are of the same speed. That assumption, however, can easily be relaxed with appropriate modifications of the model. They also provided heuristic algorithms for the problem with excellent computational results. Wang et al. [17] considered a slightly more general problem with demand estimation included in the model, and Mitrovic-Minic and Punnen [13] studied the ferry scheduling problem of reconfigurable ferries. Recently, Karapetyan and Punnen [7] proposed yet another integer programming model for the ferry scheduling problem and reported experimental results using real data. For other related works on ferry scheduling we refer to [8, 18, 19].

Modern general-purpose integer programming solvers embed powerful heuristic algorithms and sophisticated valid inequality generators to augment branch-and-cut or branch-and-bound type enumerative schemes. This together with advances in hardware technologies increased our problem solving capabilities multifold simply by using integer programming as a stand-alone solver or as a major component of developing powerful metaheuristics [2, 6, 9, 12]. These observations provided additional motivation to study integer programming formulations for the ferry scheduling problem.

In this chapter we discuss two integer programming models for the ferry scheduling problem and present a data set to test new and old algorithms. The data set is created based on insights gained from real-world applications. Some details of experimental analysis with one of the models are presented. Students using the contents of this chapter and data sets can experiment with the models as they are or with appropriate modifications to implement additional constraints and/or objectives. Most of the notations and results we use in this chapter are taken from our paper [7]. The data set provided here and related computational results are new.

2 Notations and Definitions

We first give a formal mathematical definition of the ferry scheduling problem (FSP). Let $P = \{1, 2, \ldots, n\}$ be a set of ports and $F = \{1, 2, \ldots, m\}$ be a set of ferries. For each ferry $f \in F$, a home port $h^f \in P$ is assigned. Ferry f starts and ends its service at its home port h^f. Let $[\ell, L]$ be the planning horizon. For example, if $\ell = 5{:}00$ a.m. and $L = 12{:}00$ noon, then the planning horizon is the time interval between 5:00 a.m. and 12:00 noon. Travel demands originate at a port with a prescribed destination port at discrete times within the planning horizon. Starting from the home port at time ℓ, a ferry visits a fixed number of ports, possibly multiple times (including the home ports) and returns to the home port no later than time L. We call this a *ferry traversal*. An arrangement of all the arrival times and departure times of ferry f at each of the ports it visits in a traversal is called a schedule of the ferry f (see Fig. 22.1).

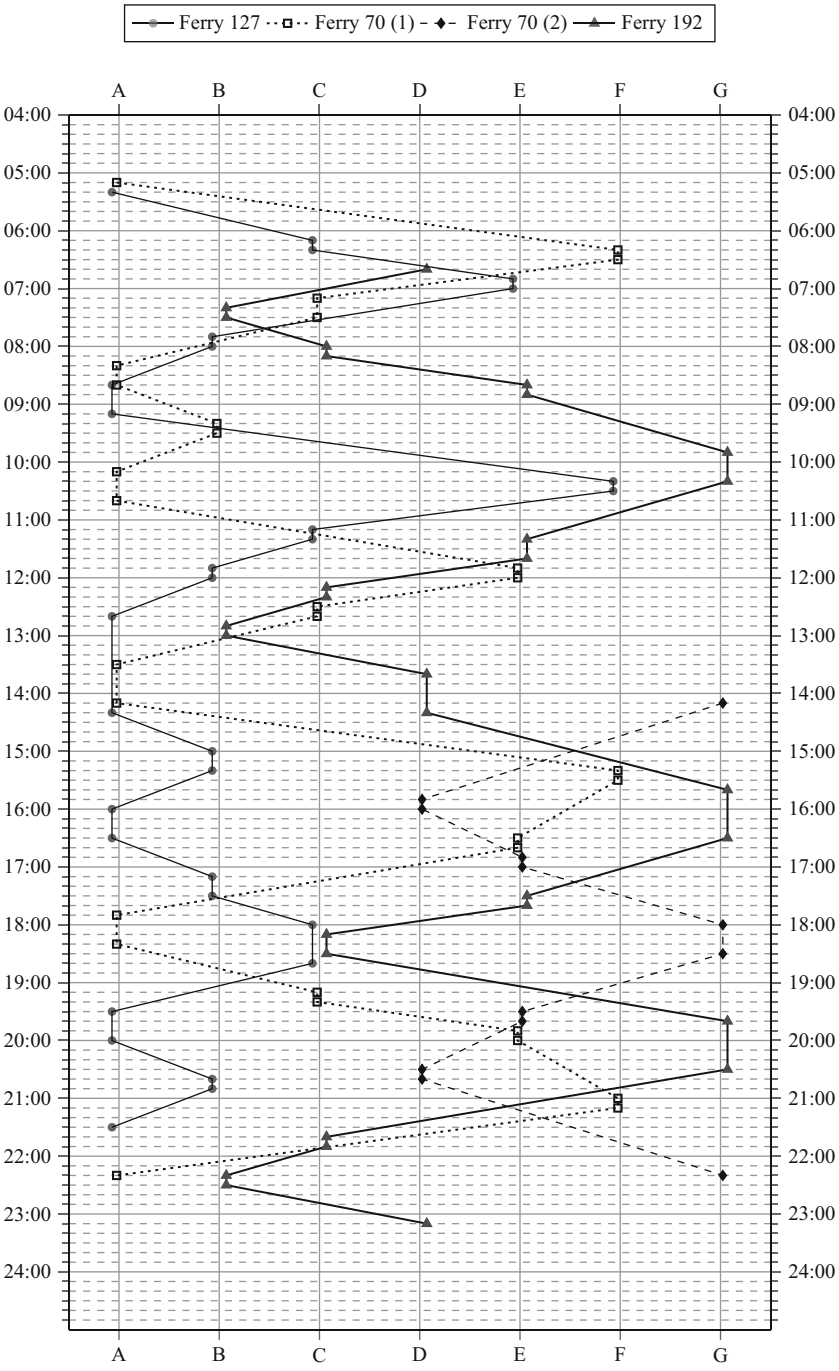


Fig. 22.1 A schedule we constructed for a peak Friday with four ferries and seven ports using real data for a ferry company

Fig. 22.2 A ferry owned by BC Ferries serving between Tsawwassen and Swartz Bay, BC, Canada

A ferry carries passengers as well as vehicles of various types such as trucks, SUVs, buses, cars, other commercial vehicles, etc. To simplify the demand types, we consider a measure called *automobile equivalent* (AEQ) which is calculated from the itemized demand using a conversion formula. Thus, hereafter we assume that demand is given in terms of AEQ. Each travel data (demand) can be represented by a 4-tuple (o, σ, t, λ) where, o is the origin port, σ is the destination port, t is the departure time, and λ is the demand volume in AEQ. The demand volume is deterministic and is part of the input data. Then the ferry scheduling problem is to develop a schedule for all the ferries to move the demand volume from the respective

Fig. 22.3 An areal view of the BC Ferries terminal at Tsawwassen, BC, Canada

Table 22.1 Main symbols and their meanings used in the chapter

m	Number of ferries
n	Number of ports
P	Set of ports
h^f	Home port of ferry f
C^f	Capacity of ferry f
β_k	Number of berths at that port k
k_i	Time state i of port k
V_k	$\{k_1, k_2, \ldots, k_q\}$
$[\ell, L]$	Planning horizon
$T^f(s,k)$	Direct travel time for ferry f from port s to port k
W_p^f	Load/unload time for ferry f at port p
b	The demand volume in AEQ
δ	Time increment factor to discretize planning horizon
ψ	The length of the planning horizon
q	$1 + \frac{\psi}{\delta}$, number of time states
$G^f = (V, E^f)$	The ferry flow network corresponding to ferry f
$F(u,v)$	$\{f : (u,v) \in E^f\}$
$\tau(k_i)$	The exact time represented by the node k_i in G^f
y_{uv}^f	Decision variable representing ferry flow along (u,v)
$\Omega^k = (U, A^k)$	Passenger flow network with destination port k
x_{uv}^k	Decision variable representing the number of passengers (measured in AEQ) traveling along arc (u,v) in Ω^k
$d_{k_i}^p$	The travel demand (in terms of AEQ) originating at node k_i
T_k	The transfer time required at port k
c_{uv}^k	The cost of arc (u,v) in Ω^k
g_e^f	Operating cost of ferry f along arc e

origin ports to destination ports so that the weighted sum of total operating cost and total travel time is minimized.

Let $T^f(s,k)$ be the direct travel time for ferry f from port s to port k without intermediate stops, $k \neq s$. If travel between ports s and k are not permitted, $T^f(s,k)$ is set to a large number. We call s the origin port, k the destination, and (s,k) an origin–destination pair (OD pair). The load/unload times mostly depend on the number of vehicles uploading to and downloading from the ferry. For busy ports, load/unload times are higher, and it is reasonably short for all other ports. Further, large ferries are likely to carry more load, and hence load/unload time also depends on ferries. We denote W_p^f the load/unload time for ferry f at port p.

A summary of all symbols used in this chapter is given in Table 22.1.

3 Integer Programming Formulations

Let us now consider our two integer programming formulations for the FSP. Two models are discussed. The first one called the compact model is taken from our paper [7]. We then provide a larger model that has an increased problem size but is more flexible. This can be viewed as an enhancement to the models discussed in [7, 11]. We also present a comparison of the relative merits of the two models.

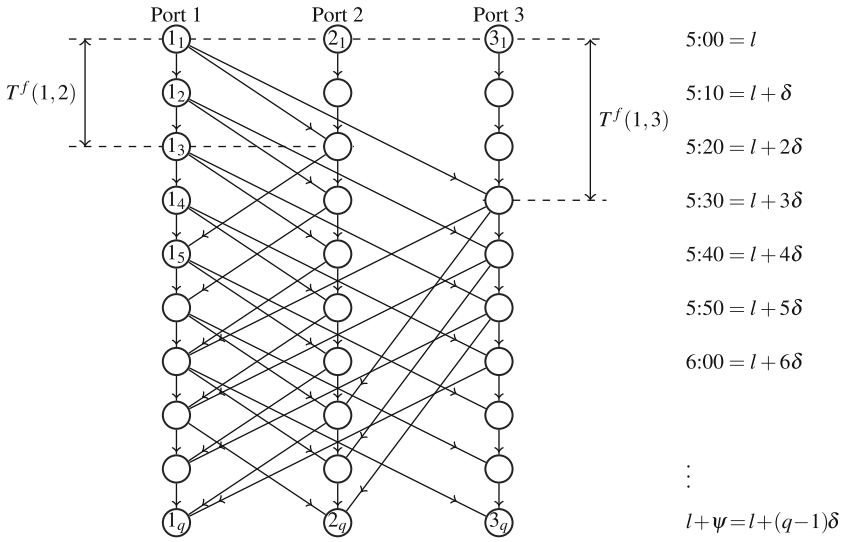


Fig. 22.4 A sample ferry flow network having three ports with *Port 1* as home port. The travel time for the ferry between *Port 1* and *Port 2* is 20 min, *Port 1* and *Port 3* is 30 min, and *Port 2* and *Port 3* is 40 min. The discretization parameter δ is set to 10 min

The backbone of our models is two networks, the *ferry flow network* and the *passenger flow network*. This is a standard approach used in many passenger vehicle models. A general overview of the model is as follows: The ferries are routed through ferry flow networks and passengers are routed through passenger flow networks. These networks are joined by imposing the constraints that a passenger cannot travel if a ferry is not available. This provides the general framework of the models. There are additional constraints that need to be handled efficiently. These include berth conflicts, transfer time constraints, load/unload times, crew exchange restrictions, among others.

Let us first construct our *ferry flow networks*. Note that the planning horizon is $[\ell, L]$ and let ψ be the length of the planning horizon. Let δ be a given *time increment factor* which is used to discretize the planning horizon. For example, ℓ represents 5:00 a.m., $\psi = 1200$ min and $\delta = 10$ min. We assume that ψ is a multiple of δ and let $q = 1 + \frac{\psi}{\delta}$.

Let $G^f = (V, E^f)$ be the ferry flow network corresponding to ferry f. For each port $k \in P$, we denote $V_k = \{k_1, k_2, \ldots, k_q\}$ to be the set of nodes in G^f representing "different time states" of port k. For example, if $\ell =$ 5:00 a.m. and $\delta = 10$ min, then k_1 represents 5:00 a.m., k_2 represents 5:10 a.m., and so on. The node set V of G^f is given by $V = \bigcup_{k \in P} V_k$. The nodes in V can be considered as a rectangular arrangement as illustrated in Fig. 22.4. We denote the exact time represented by the node k_i as $\tau(k_i)$. Thus, $\tau(k_i) = \ell + \delta(i-1)$ for all $k_i \in V$.

Introduce an arc (k_i, h_j) in E^f for $i = 1, 2, \ldots, q$ and $k, h \in P$, $k \neq h$, if a direct service from k to h is allowed and $j \leq q$ is the smallest index such that $\tau(h_j) \geq \tau(k_i) + T^f(k,h)$. Such arcs connecting time state nodes of two different

ports are called *service arcs*. Also, for each $k \in P$ and $i = 1, 2, \ldots, q-1$, we introduce an arc (k_i, k_{i+1}). These arcs connect two consecutive time state nodes of the same port and are called *waiting arcs*.

An example of a ferry flow network $G^f = (V, E^f)$ with three ports is shown in Fig. 22.4. Each service arc $(k_i, h_j) \in E^f$ represents a potential service of ferry f from port k to port h with departure time $\tau(k_i)$. Each waiting arc (k_i, k_{i+1}) represents a portion (or full) in-port time of ferry f at port k between times $\tau(k_i)$ and $\tau(k_{i+1})$.

For any $v \in V$, let $I^f(v) = \{v' : (v', v) \in E^f\}$ and $O^f(v) = \{v' : (v, v') \in E^f\}$. For each $(u, v) \in E^f$ consider the 0–1 variable y^f_{uv} defined by

$$y^f_{uv} = \begin{cases} 1 & \text{if ferry } f \text{ traverses arc } (u, v) \\ 0 & \text{otherwise.} \end{cases}$$

Define

$$b^f_v = \begin{cases} -1 & \text{if } v = h^f_1 \\ 1 & \text{if } v = h^f_q \\ 0 & \text{otherwise.} \end{cases}$$

The flow of ferry f along the ferry flow network from the beginning of the planning horizon to the end of the planning horizon is governed by the *ferry flow balancing constraints* which are given by:

$$\sum_{v' \in I^f(v)} y^f_{v'v} - \sum_{v' \in O^f(v)} y^f_{vv'} = b^f_v \text{ for every } v \in V \text{ and } f \in F. \tag{22.1}$$

Constraints of the type (22.1) are widely used in the network flow literature and generally known as flow balancing constraints. For an excellent treatment of network flow problems we refer to the books [1, 14]. The ferry flow balancing constraints allow ferry f to pass through the arcs of the ferry flow network and eventually reach back to the home port at the end of the planning horizon.

To facilitate load and unload operations at a node k_i, $i \neq q$ of G^f, we want to make sure that a ferry that enters k_i through a service arc stays at k_i for at least W^f_k min before departing from k_i. This can be achieved with the following inequalities forcing that the ferry traverse at least $w^f_k = \left\lceil \frac{W^f_k}{\delta} \right\rceil$ consecutive waiting arcs:

$$|\Delta^f_{k_i}| \sum_{v' \in I^f_{\text{serv}}(k_i)} y^f_{v'k_i} \leq \sum_{(u,v) \in \Delta^f_{k_i}} y^f_{uv} \text{for every } k_i \in V \text{ and } f \in F, \tag{22.2}$$

where $\Delta^f_{k_i} = \{(k_i, k_{i+1}), (k_{i+1}, k_{i+2}), \ldots (k_{r-1}, k_r)\}$, $r = \min\{i + w^f_k, q\}$, and $I^f_{\text{serv}}(k_i) = \{v' : (v', k_i) \in E^f, v' \notin V_k\}$. We call (22.2) the *load/unload constraints*.

Our next task is to make sure there are no berth conflicts arise at any of the ports. Note that at any time, the number of ferries staying at port k should not exceed the number of berths β_k at that port. We guarantee this by introducing the *berth constraints* given below:

$$\sum_{f \in F} y^f_{k_i k_{i+1}} \leq \beta_k \text{ for every waiting arc } (k_i, k_{i+1}), k_i \in V_k, k \in P, i \neq q. \quad (22.3)$$

The ferry flow network is the same for both integer programming models that we consider. However, the passenger flow networks in these two models are different in size and structure.

3.1 Passenger Flow Networks for the Compact Model

Let $G = (V, E)$ be the minimal supergraph of all $G^f, f \in F$. Thus, $(i, j) \in E$ implies $(i, j) \in E^f$ for at least one f. Note that all ferry flow networks have the same vertex set V which is also the vertex set of G. The network topology of G^f will be the same as G for all f if all ferries take the same time to travel between two specified ports. The only difference in this case would be the home port designation, which does not affect the network topology. However, we permit ferries to operate at different speed levels, and hence in our case, G does not need to be identical to G^f. Create new nodes $\zeta_1, \zeta_2, \ldots, \zeta_n$ and arcs (k_q, ζ_i), called *infeasibility arcs*, for $i \in P \setminus \{k\}$, $k \in P$. The infeasibility arcs are introduced to detect infeasibility easily and to measure the "magnitude" of infeasibility. In an optimal solution, if there is a positive passenger flow along an infeasibility arc, then the ferry scheduling problem is infeasible. Obviously, there are other ways to detect infeasibility of the system. We also introduce the arcs (k_i, ζ_k) called destination arcs for port k, for $i = 1, 2, \ldots, q$. The resulting graph $\Omega^k = (U, A^k)$ is called the *passenger flow network for destination port k* (see Fig. 22.5). Note that $U = V \cup \{\zeta_1, \zeta_2, \ldots \zeta_n\}$, and A^k consists of E together with all destination arcs for port k and the infeasibility arcs.

Let $d^p_{k_i}$ be the travel demand (in terms of AEQ) originating at node k_i, $k \in P$, $i = 1, 2, \ldots, q$ with destination port p. If there is no travel demand originating at node k_i with destination port p, we choose $d^p_{k_i} = 0$. The travel demand with destination port p at the node ζ_t of Ω^p is given by

$$d^p_{\zeta_t} = \begin{cases} -\sum_{k \in P} \sum_{i=1}^{q} d^p_{k_i} & \text{if } t = p \\ 0 & \text{otherwise.} \end{cases}$$

Let x^k_{uv} be the number of passengers (measured in AEQ) traveling along arc (u, v) in Ω^k. Thus the destination port of passengers $x^k_{uv}, (u, v) \in \Omega^k$ is port k. To guarantee that all passengers reach their destination port, we have *passenger flow balancing constraints* for each passenger flow network, i.e.,

$$\sum_{v' \in I_k(v)} x^k_{v',v} - \sum_{v' \in O_k(v)} x^k_{v,v'} = -d^k_v \text{ for every } v \in U \text{ and for every } k \in P, \quad (22.4)$$

where $I_k(v) = \{v' : (v', v) \in A^k\}$ and $O_k(v) = \{v' : (v, v') \in A^k\}$. Note that service arcs are the same in Ω^k for all k. The primary difference between different passenger

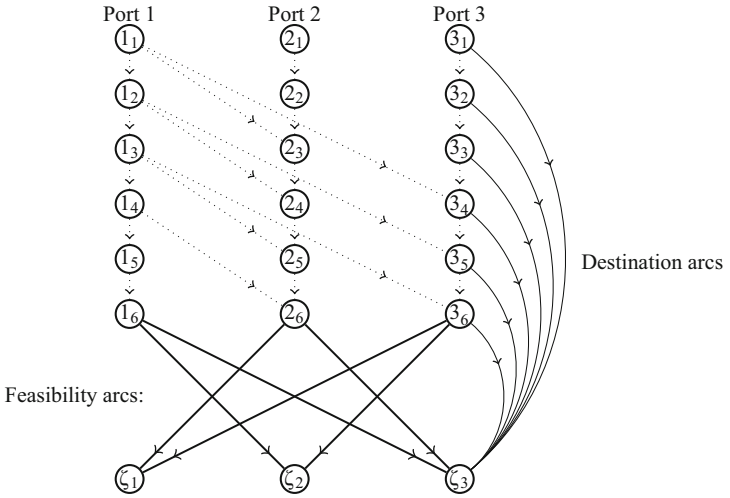


Fig. 22.5 A passenger flow network Ω^3 for aggregated decision variables with $q = 6$ and destination *Port 3*

flow networks is the destination and infeasibility arcs. Let Ω be the minimal supergraph of Ω^k for all k, i.e., each arc in Ω is an arc in at least one Ω^k for some k. For each service arc (u, v) of Ω, let $F(u, v) = \{f : (u, v) \in E^f\}$. The *capacity* of a service arc (u, v) in Ω is given by $\sum_{f \in F(u,v)} C^f y^f_{uv}$, where C^f is the capacity of ferry f. A passenger can travel along a service arc only if a ferry traverses that arc. To insure this, we require that the sum of passenger flows to all destinations along a service arc should not exceed the capacity of that arc, i.e.,

$$\sum_{k \in P} x^k_{uv} \leq \sum_{f \in F(u,v)} C^f y^f_{uv} \text{ for every service arc } (u, v) \in \Omega \text{ and } k \in P. \tag{22.5}$$

The inequalities (22.5) are referred to as *capacity constraints*.

It is possible that some passengers might not travel to their destinations directly in the ferry and might need to transfer from one ferry to another one or more times. Although multiple transfers are not desirable, the model does not impose any penalty for such travel. However, to make such transfers feasible, we need to allocate sufficient time for passengers to unload from their ferry and load onto another ferry.

For simplicity, we assume that the transfer time at a given port is independent of the type of ferries and time state nodes. Let T_k be the transfer time required at port k, $\hat{T}_k = \left\lceil \frac{T_k}{\delta} \right\rceil$ and $\Delta_{k_i} = \{k_i, k_{i+1}, \ldots, k_{r-1}\}$, where $r = \min\{i + \hat{T}_k, q\}$. Let $I_{\text{serv}}(k_i) = \{v' : (v', k_i) \in E, v' \notin V_k\}$. Then the *transfer constraints* can be stated as

$$\sum_{j=i}^{t} \sum_{v' \in I_{\text{serv}}(k_j)} x^p_{v'k_j} \leq x^p_{k_t k_{t+1}} \text{ for } t = i \text{ to } r-1, k_i \in V, i \neq q \text{ and } p \in P \setminus \{k\}. \tag{22.6}$$

Note that (22.6) affects not only the transferring passengers but also the passengers remaining on a ferry. Thus, we require $T_k \leq \min_{f \in F} W_k^f$.

The transfer constraints force that the flow with destination port p along each of the arcs $(k_j, k_j + 1), k_j \in \Delta_{k_i}$ is at least the total flow with destination port p arriving at node k_i through service arcs. Note that we have $|\Delta_{k_i}|$ transfer inequalities for each $k_i \in V$, $k \in P$. If port k has only one berth or $\hat{T}_k = 1$, we can replace the $|\Delta_{k_i}|$ transfer inequalities at node k_i by a single inequality

$$\sum_{v' \in I_{\text{serv}}(k_i)} x^p_{v' k_i} \leq x^p_{k_i k_{i+1}} \text{ for every } k_i \in V, i \neq q \text{ and } p \in P. \tag{22.7}$$

The next ingredient in the development of our model is the objective function. This depends on our "objective," as the name suggests. One can consider different factors in defining the objective which is a combination of operating cost, passenger dissatisfaction, number of transfers, etc. We restrict our goal to "minimize" the operating cost and "maximize" the level of service. Since these two objectives are contradictory in nature, we define the objective function as a compromise between these competing goals. Let μ^f be the cost of operating ferry f for 1 h. This cost includes fuel costs and crew salary. For each edge $e \in E^f$ define its ferry operating cost g_e^f as

$$g_e^f = \mu^f \cdot (\tau(v) - \tau(u)),$$

where $e = (u, v)$.

Let c^k_{uv} be the cost of arc (u, v) in the passenger flow network Ω^k. Then c^k_{uv} is given by

$$c^k_{uv} = \begin{cases} M & \text{if } (u, v) \text{ is an infeasibility arc} \\ 0 & \text{if } (u, v) \text{ is a destination arc} \\ \tau(h_j) - \tau(k_i) & \text{otherwise, where } (u, v) = (k_i, h_j), \end{cases}$$

where M is a very large number. Note that the destination arcs have zero cost and any passenger that does not reach the correct destination port will travel through the infeasibility arcs incurring a very large cost. This is to make sure the passengers reach their destination if there is a feasible solution. Using these cost elements, we define our objective function as

$$\phi(x, y) = \lambda \sum_{f \in F} \sum_{e \in E^f} g_e^f y_e^f + \nu \sum_{k \in P} \sum_{e \in A^k} c_e^k x_e^k, \tag{22.8}$$

where $\lambda \geq 0$ and $\nu > 0$ are coefficients controlling the cost/level of service balance and we want to minimize $\phi(x, y)$.

The compact integer programming model for the ferry scheduling problem discussed above is summarized as follows:

$$\text{Minimize} \quad \lambda \sum_{f \in F} \sum_{e \in E^f} g_e^f y_e^f + \nu \sum_{k \in P} \sum_{e \in A^k} c_e^k x_e^k$$

subject to:

$$\sum_{v' \in I^f(v)} y^f_{v'v} - \sum_{v' \in O^f(v)} y^f_{vv'} = b^f_v \text{ for every } v \in V \text{ and } f \in F,$$

$$|\Delta^f_{k_i}| \sum_{v' \in I^f_{\text{serv}}(k_i)} y^f_{v'k_i} \leq \sum_{(u,v) \in \Delta^f_{k_i}} y^f_{uv} \text{ for every } k_i \in V \text{ and } f \in F,$$

$$\sum_{f \in F} y^f_{k_i k_{i+1}} \leq \beta_k \text{ for every waiting arc } (k_i, k_{i+1}), k_i \in V_k, k \in P, i \neq q,$$

$$\sum_{v' \in I_k(v)} x^k_{v',v} - \sum_{v' \in O_k(v)} x^k_{v,v'} = -d^k_v \text{ for every} v \in U \text{ and for every } k \in P,$$

$$\sum_{k \in P} x^k_{uv} \leq \sum_{f \in F(u,v)} C^f y^f_{uv} \text{ for every service arc}(u,v) \in \Omega,$$

$$\sum_{j=i}^{t} \sum_{v' \in I_{\text{serv}}(k_j)} x^p_{v'k_j} \leq x^p_{k_t k_{t+1}} \text{ for } t = i \text{ to } r-1,$$

$$k_i \in V, i \neq q \text{ and } p \in P \setminus \{k\},$$

$$y^f_{uv} \in \{0,1\} \text{ for } (u,v) \in E^f, f \in F,$$

$$x^k_{uv} \text{ is a nonnegative integer } for (u,v) \in A^k, k \in P.$$

The model discussed above has $O(qmn^2)$ constraints and $O(qmn^3)$ variables. This gives a reasonably compact model that can be solved using general-purpose integer programming solvers for moderate size problems. The compactness, however, comes with a disadvantage that the model is less flexible. We cannot distinguish passengers traveling to the same destination in terms of their origin ports. However, in our case it is not necessary because the model allows us to calculate the total traveling time of all the passengers.

3.2 *Enlarged Integer Programming Model*

Let us now consider a more elaborate model by increasing the number of variables to gain additional flexibility. If we use demand types as OD pair, as considered in [11, 13, 17], we can get an integer programming formulation for the ferry scheduling problem with $O(qmn^3)$ constraints and $O(qmn^4)$ variables. This increases the model size by a factor of $O(n)$ compared to the compact model discussed above.

We now consider a further elaborate model where passengers are completely distinguishable in terms of origin port, destination port, and time of arrival at the origin port. This obviously increases the number of decision variables, but allows considerable flexibility. For example, if we know that most of the passengers arriving at a given port, say port k, at 7:30 a.m. and going to port t are students who need to

reach their destination by 8:30 a.m., we will be able to impose this by a hard constraint or modifying the cost of the corresponding variables in the objective function.

For each pair (k_i, t), where k_i represents port k at time state i (essentially a node k_i of the ferry flow network) and t represents a destination port, we define a passenger flow network $\Omega^{k_i,t} = (U^{k_i,t}, A^{k_i,t})$. The network $\Omega^{k_i,t}$ is the same as Ω^t (i.e. $U^{k_i,t} = U^t$ and $A^{k_i,t} = A^t$) except that the demand $d_{r_j}^{k_i,t}$ at node r_j, for $r \in P$ and $j = 1, 2, \ldots q$ is given by

$$d_{r_j}^{k_i,t} = \begin{cases} d_{k_i}^t & \text{if} r_j = k_i \\ -d_{k_i}^t & \text{if} r_j = \zeta_t \\ 0 & \text{otherwise.} \end{cases}$$

Let $x_{uv}^{k_i,t}$ be the number of passengers with origin node k_i (i.e., origin port k and time of arrival at port k corresponds to the time state i in the network), destination node t, and traveling along arc (u, v) of the passenger flow network $\Omega^{k_i,t}$. Then the passenger flow balancing constraints become:

$$\sum_{v' \in I_v^{k_i,t}} x_{v',v}^{k_i,t} - \sum_{v' \in O_v^{k_i,t}} x_{v,v'}^{k_i,t} = -d_v^{k_i,t} \text{for every } v \in U^{k_i,t} \text{ and for every } (k_i, t) \in (V \times P), \tag{22.9}$$

where, $I_v^{k_i,t} = \{v' : (v', v) \in A^{k_i,t}\}$ and $O_v^{k_i,t} = \{v' : (v, v') \in A^{k_i,t}\}$.

As in the case of our compact model, the service arcs are the same in $\Omega^{k_i,t}$ for all $(k_i, t) \in V \times P$. It can be verified that be the minimal supergraph of $\Omega^{k_i,t}$ for all $(k_i, t) \in V \times P$ is the same the graph Ω defined in the previous section,. i.e., each arc in Ω is an arc in at least one $\Omega^{k_i,t}$ for some (k_i, t). Recall that for each service arc (u, v) of Ω, we define $F(u, v) = \{f : (u, v) \in E^f\}$ and the *capacity* of a service arc (u, v) in Ω by $\sum_{f \in F(u,v)} C^f y_{uv}^f$, where C^f is the capacity of ferry f. A passenger can travel along a service arc only if a ferry traverses that arc. Thus, the sum of passenger flows to all destinations along a service arc should not exceed the capacity of that arc, i.e.,

$$\sum_{(k_i,t) \in v \times P} x_{uv}^{k_i,t} \leq \sum_{f \in F(u,v)} C^f y_{uv}^f \text{ for every service arc } (u, v) \in \Omega\,. \tag{22.10}$$

Again, as in our previous model, we call constraints (22.10) the *capacity constraints*. We now discuss the transfer constraints for our model. Recall that T_k is the transfer time required at port k, $\hat{T}_k = \left\lceil \frac{T_k}{\delta} \right\rceil$ and $\Delta_{k_i} = \{k_i, k_{i+1}, \ldots, k_{r-1}\}$, where $r = \min\{i + \hat{T}_k, q\}$. Let $I_{\text{serv}}(k_i) = \{v' : (v', k_i) \in E, v' \notin V_k\}$. Then the *transfer constraints* can be stated as

$$\sum_{j=i}^{t} \sum_{v' \in I_{\text{serv}}(k_j)} x_{v'k_j}^p \leq x_{k_t k_{t+1}}^p \text{ for } t = i \text{ to } r - 1, k_i \in V, i \neq q \text{ and } p \in P \setminus \{k\}. \tag{22.11}$$

Note that (22.11) affects not only the transferring passengers but also the passengers remaining on a ferry. Thus, we require $T_k \leq \min_{f \in F} W_k^f$.

To complete this model, we need to define the objective function and transfer constraints. We leave it for the reader to complete this, using insights gained from our compact model.

3.3 Model Refinements and Implementation Details

The formulations discussed in the previous sections are reasonably general representations of the ferry scheduling problem. Additional refinements may be necessary to enhance the model, depending on specific application and availability of information. We illustrate this with a case study using five ports and three ferries. The data used in this study is simulated based on our experience in a real case study consisting of seven ports and four ferries [7]. The problem size is slightly reduced in the present case, yet kept reasonably large, so that the readers can experiment with the data and make reasonable conclusions. These refinements are taken directly from our paper [7] keeping the same notations and terminology, although the data and experimental results reported are different.

We restrict ourselves to the compact model that is used in our experiments. The readers can make appropriate changes to the distributed model and run experiments with our data. This is indicated as an exercise at the end of this chapter.

Note that to facilitate load/unload operations, we need to make sure that a ferry f entering port k through node k_i must stay at port k for at least $w_k^f \delta$ time. The load/unload constraints (22.2) assures this in our model. However, there are alternative ways to handle this. In our experiments, constraints (22.2) are replaced by

$$\sum_{v' \in O^f_{\text{serv}}(k_i)} y_{k_i,v'} \leq y_{k_j,k_{j+1}} \text{ for every } k_i \in V \text{ and } f \in F\,, \tag{22.12}$$

where $O^f_{\text{serv}}(v) = \{v' \in O^f(v) : (v, v') \text{ is a service arc}\}$ and j is the largest index such that $\tau(k_i) - \tau(k_j) \geq W_k^f$. If no such j exists, we do not have a constraint for the corresponding node k_i. This constraint is invalid if $W_k^f \geq \min_{h \in P} T^f(k,h) + T^f(h,k)$ and our applications are consistent with this requirement.

We also replaced constraints (22.6) by constraints (22.7) in our case study. It reduces the number of constraints and simplifies the model without affecting solution quality. Again, it is a restriction on the general model but the data used in our case study permits this simplification.

The cost of a waiting arc $e = (k_i, k_{i+1})$ is set to $g_e^f = \delta \mu_f$ in the general model. If all ferries are in operation exactly at the beginning of the planning horizon and continue service till the end of the planning horizon, this cost assignment is perfectly valid. However, this can lead to over estimation of cost (and hence affect optimality) in some applications. It is not necessary that all ferries need to be in operation exactly at 5:00 a.m. A ferry could start service late, say at 10:00 a.m. and terminate service at 4:00 p.m. It can also skip all morning or serve only in the morning. The general

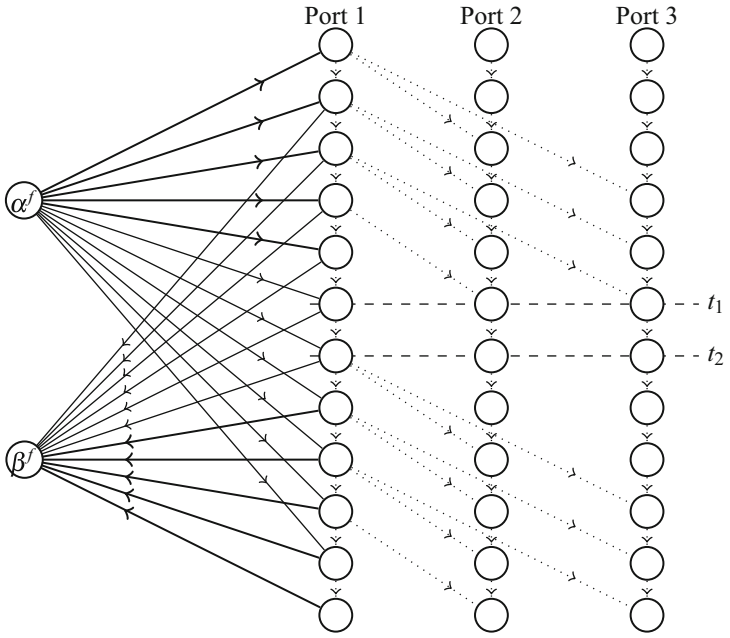


Fig. 22.6 Ferry flow network with nodes α^f and β^f that allow efficient calculation of the crew salaries and loading/unloading stays

discussions given in the last section does not take care of this, leading to suboptimal solutions. With a minor change in the model, we can address this difficulty.

Introduce two new nodes α^f and β^f to the ferry flow network G^f, $f \in F$ and connect α^f to all nodes in $V_{h^f} \setminus \{h_q^f\}$ using arcs (α^f, h_i^f), $i = 1, 2, \ldots, q-1$, referred to as *in-port arcs*. Similarly, connect each node of $V_{h^f} \setminus \{h_1^f\}$ to β^f and the resulting arcs (h_i^f, β^f), $i = 2, 3, \ldots, q$ are called *out-port arcs* (see Fig. 22.6).

We set $g_e^f = 0$ whenever e is an in-port or out-port arc. It is assumed that ferry f is stationed at node α in G^f and returns to node β in G^f at the end of service. Consequently, we change the definition of b_v^f for this modified G^f as

$$b_v^f = \begin{cases} -1 & \text{if } v = \alpha^f \\ 1 & \text{if } v = \beta^f \\ 0 & \text{otherwise.} \end{cases}$$

Normally, the ferry crew operates in two shifts per day. Dividing the planning horizon into two is not a good approach as some passengers may start their travel in the morning and finish it in the afternoon. Let $[t_1, t_2]$ be a given time interval during which the ferry must come back to its home port for crew exchange, regardless of the number of hours the ferry was in service. Let σ^f be the total crew salary for ferry f per shift.

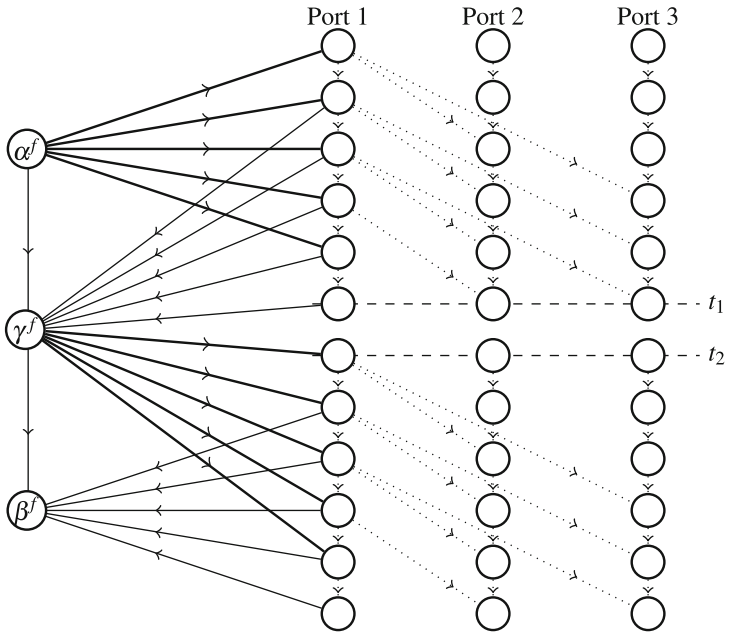


Fig. 22.7 Ferry flow network with nodes α^f, β^f, and γ^f that allow better calculation of the crew salaries and correct calculation of the fuel charges

In G^f, introduce three nodes α^f, β^f, and γ^f. Join α^f to $h_i^f \in V_{h^f}$ by an arc with cost σ^f whenever $\tau(h_i^f) < t_1$. Likewise, join h_i^f to γ^f by an arc (h_i^f, γ^f) of zero weight for every $i > 1$ such that $\tau(h_i^f) \leq t_1$. Set the cost g_e^f of the waiting arcs e involving time state nodes v with $t_1 \leq \tau(v) \leq t_2$ to a very large number M.[1] Then connect γ^f to h_i^f by an arc (γ^f, h_i^f) of weight σ^f for every $i < q$ such that $\tau(h_i^f) \geq t_2$. Also, connect each node h_i^f to β^f by an arc (h_i^f, β^f) of zero weight for every i such that $\tau(h_i^f) > t_2$. Finally connect α^f to γ^f and γ^f to β by arcs of zero weights, see Fig. 22.7.

We leave it to the reader to verify that the modifications discussed here achieve the desired objective. We used this modified model in the experimental analysis discussed in the next section.

4 Experimental Analysis

The integer programming model with aggregated variables, incorporating modifications discussed in the previous section was tested on simulated data with three ferries and five ports. The data were generated using insights gained from an earlier

[1] We assume that such arcs exist.

Table 22.2 Travel time for the ferries. 999 indicates that the corresponding travel is not permitted

	Small ferry					Large ferry				
Ports	Port 1	Port 2	Port 3	Port 4	Port 5	Port 1	Port 2	Port 3	Port 4	Port 5
Port 1	0	40	50	40	999	0	40	50	40	999
Port 2	40	0	50	30	999	40	0	50	30	70
Port 3	50	50	0	40	999	50	50	0	40	75
Port 4	40	30	40	0	999	40	30	40	0	65
Port 5	999	999	999	999	0	999	70	75	65	0

case study involving four ferries and seven ports. We have two sets of data: peak season and off-peak season. We only consider vehicles to define the demand value. However, if no vehicles were transferred between two ports but there were some foot passengers, we assume the demand volume to be 1 AEQ to ensure that these passengers will be taken into account by the model. The terminology passenger (in terms of AEQ) refers to both foot passengers as well as vehicles. The total demand in the considered problem was from 1000 to 2000 AEQ per day.

The integer programming model was solved using CPLEX 12.5 and the problem input was prepared using a C# code. The experiments were run on a Windows 7 PC with an Intel i7-3820 3.6 GHz CPU and 16 GB of memory. The parameter values for the model were set as follows: $\delta = 10$ min, $\ell =$ 5:00, $\psi = 19$ h. In our test data, we have three ferries, two are of capacity 100 AEQ called small ferries and one having capacity of 200 AEQ, which is called large ferry. For small ferries, fuel cost is £ 400/h enroute and £ 160/h in port. Similarly for large ferry, fuel cost is £ 600/h enroute and £ 240/h in port. The crossing times for these ferries are given in Table 22.2.

Table 22.3 represents travel demand in terms of AEQ.

Figure 22.8 gives the schedule generated by CPLEX running 4 h and it reached within 0.43 % of optimality using the value 10 for cost coefficient. Figure 22.9 gives the schedule generated by CPLEX running 4 h and it reached within 0.44 % of optimality using the value 1 for cost coefficients.

5 Conclusions

In the current chapter, we presented a compact integer programming model for the ferry scheduling problem. This model has reduced size compared to the standard models by a factor of approximately $O(n)$. As shown, this is achieved at the cost of considerably lower flexibility. However, it efficiently handles all the operational constraints including load/unload times and passenger transfers. The model was able to produce high-quality solutions in 4 h using CPLEX 12.5. We also provided a more flexible model with increased number of variables and constraints.

Table 22.3 Demand data used in our experiments. *Origin* origin port, *Destination* destination port, *Desired dep. time* desired departure time, *Desired arr. time* desired arrival time, *AEQ* demand in terms of AEQ

Origin	Destination	Desired dep. time	Desired arr. time	AEQ
Port1	Port4	06:30	07:30	21
Port3	Port5	06:30	07:30	32
Port1	Port2	07:00	08:00	4
Port1	Port3	07:00	08:00	6
Port2	Port1	08:00	08:30	91
Port3	Port1	08:00	08:30	80
Port4	Port1	08:00	08:30	40
Port5	Port2	08:00	09:00	31
Port5	Port3	08:00	09:00	76
Port5	Port4	08:00	10:00	7
Port4	Port2	09:00	10:00	8
Port1	Port2	10:00	11:00	45
Port1	Port3	10:00	11:00	65
Port2	Port1	11:00	12:00	49
Port2	Port5	11:00	12:00	34
Port3	Port1	11:00	12:00	77
Port3	Port5	11:00	12:00	48
Port4	Port1	11:00	12:00	33
Port4	Port5	11:00	12:00	20
Port1	Port3	12:00	13:00	15
Port2	Port5	14:00	15:00	40
Port3	Port5	14:00	15:00	79
Port2	Port4	15:00	16:00	8
Port3	Port1	15:00	16:00	23
Port1	Port2	16:00	16:30	32
Port4	Port5	16:00	17:00	32
Port1	Port3	16:30	17:00	40
Port1	Port4	16:30	17:00	54
Port1	Port2	18:00	19:00	40
Port1	Port3	18:00	19:00	32
Port4	Port1	18:00	19:00	20
Port2	Port1	19:00	20:00	21
Port3	Port1	19:00	20:00	8
Port5	Port2	19:00	20:00	31
Port5	Port3	19:00	20:00	93
Port1	Port2	20:00	21:00	42
Port1	Port3	20:00	21:00	30
Port1	Port4	20:00	21:00	19
Port3	Port5	20:00	21:00	52
Port5	Port2	20:00	21:00	11
Port5	Port3	20:00	21:00	39
Port5	Port4	20:00	21:00	14

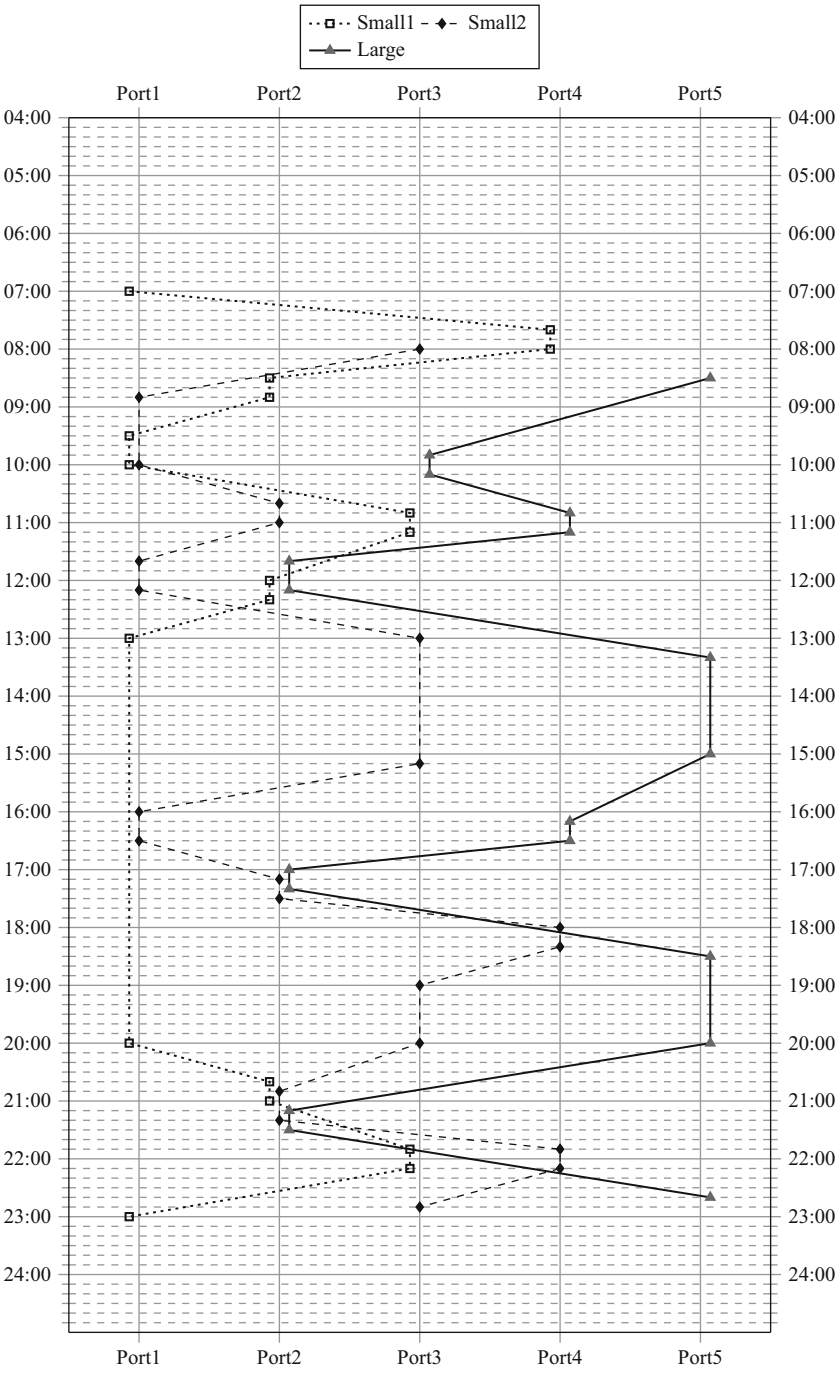


Fig. 22.8 Schedule generated by CPLEX for the cost coefficient 10 after running 4 h. The optimality gap reached is 0.43 %

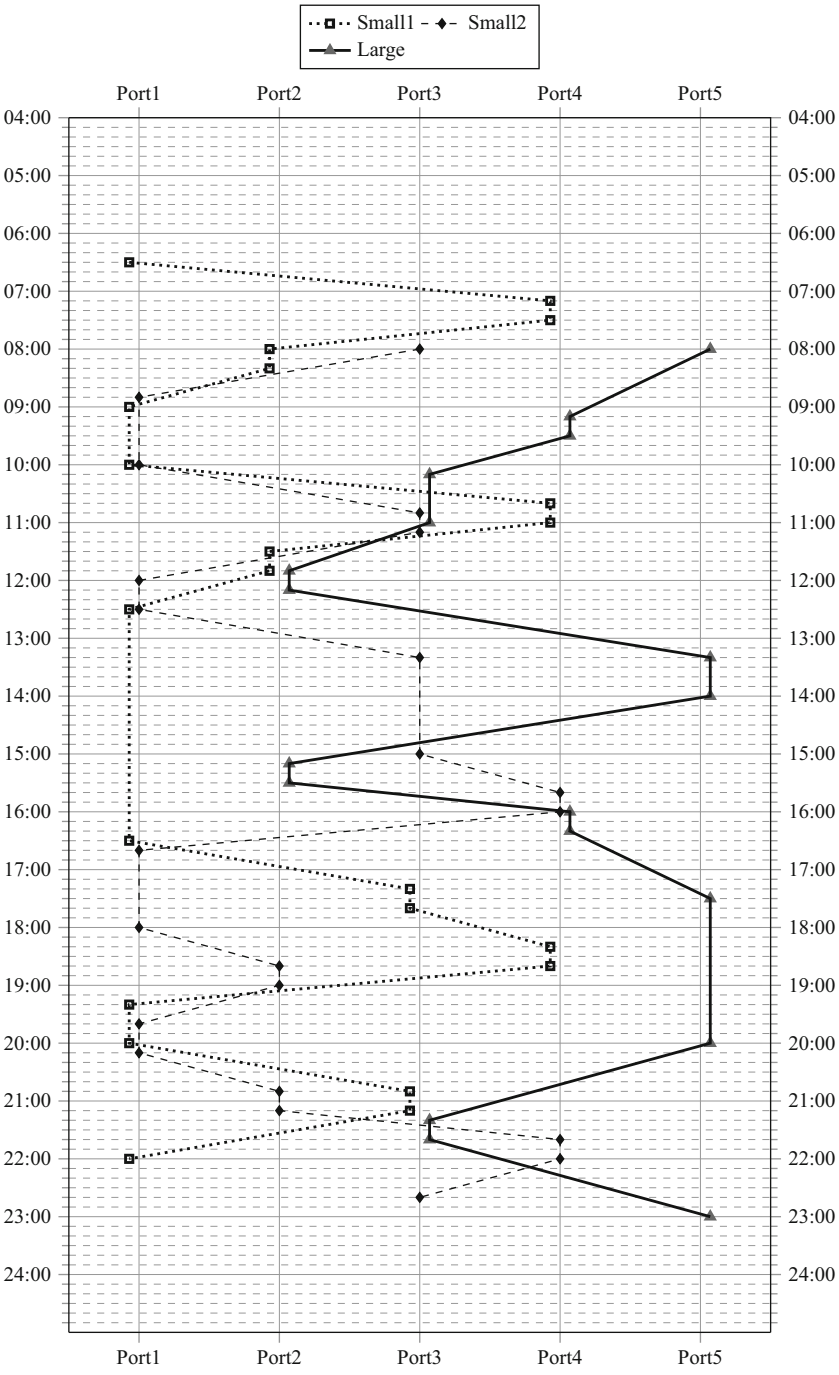


Fig. 22.9 Schedule generated by CPLEX for the cost coefficient 1 after running 4 h. The optimality gap reached is 0.44 %

6 Exercises

1. Develop the objective function and transfer constraints for the second integer programming model. If a desired arrival time for passengers is known, how do you integrate this in this model?
2. Solve the model above using CPLEX and the data set provided.
3. Compare and contrast the advantages and disadvantages of the two models given.
4. Introduce an additional group of passengers and explain how to handle this in an integer programming model where these groups are distinguishable.
5. Identify new meaningful constraints that can be added to the models and solve the resulting problems.
6. Develop metaheuristic algorithms to solve the FSP and report results of experimental analysis with the data provided.

Acknowledgment We are thankful to Steve Anderson and Peter Simpson at BC Ferries for their help and support in this work. Comments of Katta G. Murty, editor of this book, improved the presentation of this chapter. This work was partially supported by BC Ferries and MITACS. The chapter is primarily based on our research paper [7], but it also includes additional new material and experimental results.

References

1. Ahuja, R. K., Orlin, J. B., & Magnanti, T. L. (1993). *Network flows: Theory, algorithms, and applications*. Upper Saddle River: Prentice Hall.
2. Bixby, R. E. (2010). Mixed-integer programming: it works better than you may think. *FERC Conference*, Washington: DC.
3. Ceder, A. (2003). Designing public transport network and routes. In W. Lam and M. Bell (eds.), *Advanced modeling for transit operations and service planning*. Amsterdam: Elsevier.
4. Goetschalckx, M. (2011). *International series in operations research & management science. Supply chain engineering*. Berlin: Springer.
5. Gopalan, R., & Talluri, K. T. (1998). Mathematical models in airline schedule planning: A survey. *Annals of Operations Research, 76,* 155–185.
6. Hvattum, L. M., Lokketangen, A., & Glover, F. (2012). Comparisons of commercial MIP solvers and an adaptive memory (tabu search) procedure for a class of 0-1 integer programming problems. *Algorithmic Operations Research, 7,* 13–20.
7. Karapetyan, D., & Punnen, A. P. (2013). A reduced integer programming model for the ferry scheduling problem. *Public Transport, 4,* 151–163.
8. Khairy Adly Wan Zaimi, W. M., Abu, M. S., Junoh, A. K., Ariffin, W. N. M. (2011). Ferry scheduling model using linear programming technique. In *2011 3rd International Conference on Computer Research and Development (ICCRD)* (pp. 337–341), vol. 4, 11–13 March 2011
9. Koch, T., Achterberg, T., Anderson, E., Bastert, O., Berthold, T. Bixby, R. E., Danna, E., Gamrath, G., Gleixner, A. M., Heinz, S., Lodi, A., Mittelmann, H., Ralphs, T., Salvagnin, D., Steffy, D. E., & Wolter, K. (2011). MILP 2010: Mixed integer programming library version 5. *Mathematical Programming Computation, 3,* 103–163.
10. Lai, M. F., Lo, H. K. (2004). Ferry service network design: Optimal fleet size, routing, and scheduling. *Transportation Research Part A: Policy and Practice, 38,* 305–328.

11. Linderoth, J. T., & Lodi, A. (2011). MILP software. In J. J. Cochrane (Ed.), *Wiley Encyclopedia of Operations Research and Management Science*. Hoboken: Wiley.
12. Lodi, A. (2012). *MIP computation and beyond*. Technical report ARRIVAL-TR-0229.
13. Mitrovic-Minic, S., & Punnen, A. P.. (2011). Routing and scheduling of a heterogeneous fleet of re-configurable ferries: A model, a heuristic, and a case study. In *OR 2011 - International Conference on Operations Research*, Zurich, Switzerland, 2011.
14. Murty, K. G.. (1992). *Network programming*. Upper Saddle River: Prentice Hall.
15. Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Philadelphia: SIAM.
16. Voß, S., & Daduna, J. R. (Eds.) (2001). *Springer lecture notes in economics and mathematical systems, vol. 505. Computer-aided scheduling of public transport*. Berlin: Springer.
17. Wang, Z. W., Lo, H., & Lai, M. F. (2008). Mixed-fleet ferry routing and scheduling. In M. Hickman, P. Mirchandani, and S. Voss (Eds.), *Lecture notes in economics and mathematical systems 600. Computer-aided systems in public transport* (pp. 181–194). Berlin: Springer.
18. Wang, D. Z. W., & Lo, H. K. (2008). Multi-fleet ferry service network design with passenger preferences for differential services. *Transportation Research Part B, 42,* 798–822.
19. Yan, S., Chen, C.-H., Chen, H.-Y., & Lou, T.-C. (2007). Optimal scheduling models for ferry companies under alliances. *Journal of Marine Science and Technology, 15,* 53–66.

Index

K. G. Murty (ed.), *Case Studies in Operations Research,*
International Series in Operations Research & Management Science 212,
DOI 10.1007/978-1-4939-1007-6, © Springer Science+Business Media New York 2015