



QUANTITATIVE APPROACHES  
TO DECISION MAKING COLLECTION

Donald N. Stengel, *Editor*

# Business Applications of Operations Research

**Bodhibrata Nag**



[www.busessexpertpress.com](http://www.busessexpertpress.com)

# Business Applications of Operations Research



# Business Applications of Operations Research

Bodhibrata Nag



*Business Applications of Operations Research*

Copyright © Business Expert Press, LLC, 2014.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopy, recording, or any other except for brief quotations, not to exceed 400 words, without the prior permission of the publisher.

First published in 2014 by  
Business Expert Press, LLC  
222 East 46th Street, New York, NY 10017  
[www.busessexpertpress.com](http://www.busessexpertpress.com)

ISBN-13: 978-1-60649-526-1 (paperback)

ISBN-13: 978-1-60649-527-8 (e-book)

Business Expert Press Quantitative Approaches to Decision Making  
Collection

Collection ISSN: 2163-9515 (print)

Collection ISSN: 2163-9582 (electronic)

Cover and interior design by Exeter Premedia Services Private Ltd.,  
Chennai, India

First edition: 2014

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America.

## **Abstract**

Operations Research is a bouquet of mathematical techniques that have evolved over the last six decades to improve the process of business decision making. Operations Research offers tools to optimize and find the best solutions to myriad decisions that managers have to take in their day to day operations or while carrying out strategic planning. Today, with the advent of operations research software, these tools can be applied by managers even without any knowledge of the mathematical techniques that underlie the solution procedures.

The book starts with a brief introduction to various tools of operations research, such as linear programming and integer programming together with simple examples formulated and solved using the operations research software LINGO.

The book intends to make the readers aware of the power and potential of operations research in addressing decision making in areas of operations, supply chain, and financial and marketing management. The approach of this book is to demonstrate the solution to specific problems in these areas using operations research techniques and LINGO software. The reader is encouraged to use the accompanying software models to solve these problems, using detailed do-it-yourself instructions and the limited version of LINGO software available at the “Downloads-Try Lingo” tab of the website [www.lindo.com](http://www.lindo.com). The intended outcome for readers of this book will be gaining familiarity with and an intuitive understanding of the various tools of operations research and their applications to various business situations. It is expected that this will give the readers the ability and confidence to devise models for their own business needs.

## **Keywords**

operations research, linear programming, integer programming, heuristics, queues, supply chain models, marketing models, operations models, financial models



# Contents

<b>Section 1</b>	<b>Introduction to Operations Research.....</b>	<b>1</b>
Chapter 1	Introduction to Operations Research and Guide to LINGO Software .....	3
<b>Section 2</b>	<b>Applications in Operations Management.....</b>	<b>31</b>
Chapter 2	Product Mix .....	33
Chapter 3	Facility Location .....	35
Chapter 4	Cable Layout .....	45
Chapter 5	Planning Check-in Counters .....	53
Chapter 6	Scheduling of a Production Line.....	55
Chapter 7	Shift Staff Planning.....	61
Chapter 8	Production Planning.....	67
Chapter 9	Blending of Dog Diet.....	71
Chapter 10	Paper Roll Trimming .....	73
<b>Section 3</b>	<b>Applications in Supply Chain Management.....</b>	<b>77</b>
Chapter 11	Multicommodity Transport Planning .....	79
Chapter 12	Single Delivery Truck Routing.....	87
Chapter 13	Multiple Delivery Trucks Routing .....	93
Chapter 14	Supplier Selection with Multiple Criteria.....	97
<b>Section 4</b>	<b>Applications in Marketing Management.....</b>	<b>103</b>
Chapter 15	Revenue Management .....	105
<b>Section 5</b>	<b>Applications in Financial Management.....</b>	<b>109</b>
Chapter 16	Portfolio Management.....	111

Chapter 17 Capital Budgeting .....115

Chapter 18 Bank Asset Liability Management.....117

Chapter 19 Index Fund Construction .....123

**Section 6 Applications in Transport Management..... 127**

Chapter 20 Airline Network Design.....129

Chapter 21 Performance Measurement Using Data  
Envelopment Analysis .....137

*Notes*.....143

*References*.....147

*Index* .....149

## SECTION 1

# Introduction to Operations Research



## CHAPTER 1

# Introduction to Operations Research and Guide to LINGO Software

### 1. Introduction to Operations Research

Operations Research (or Management Science or Decision Sciences) is a scientific manner of decision making, mostly under the pressure of scarce resources. The objective of decision making is often the maximization of profit, minimization of cost, better efficiency, better operational or tactical or strategic planning or scheduling, better pricing, better productivity, better recovery, better throughput, better location, better risk management, or better customer service. The scientific manner involves extensive use of mathematical representations or models of real life situations. These representations allow employment of mathematical techniques to arrive at optimal or near optimal decisions after consideration of all possible options. The mathematical representations also allow the decision maker to understand the system better through sensitivity and scenario analyses.<sup>1-2</sup>

Operations Research had its origins in the Second World War when a group of scientists and engineers was formed to aid the military in proper radar deployment, convoy management, anti-submarine operations, and mining operations.

The development of the linear programming solution methodology by George Dantzig and the advent of computers in the early 1950s led to enormous interest in operation research applications in business. Industries set up operations research groups especially in areas relating to oil refineries and finance. Simultaneously, other methodologies such

as integer programming solution methodology, queuing theory, graph and network theory, non-linear programming, stochastic programming, game theory, dynamic programming, Markov decision processes, meta-heuristic procedures such as simulated annealing, genetic and tabu search, neural networks, multi-criteria or multi-objective analysis, analytic hierarchy process, simulation, and Petri nets were being developed to tackle business problems.

The advent of computers with large memories and high speeds in the 1990s led to the second major wave of interest in operations research. The major areas where operations research was applied in the second wave were yield or revenue management, crew management and network design in the airlines sector, and finance and supply chain management. Today, the opportunities of exploiting operations research techniques have multiplied manifold with the huge amount of data available through ERP,<sup>3</sup> CRM,<sup>4</sup> or point of sale capture for analysis and decision making.

Academic journals such as *Interfaces*, *Management Science*, *Transportation Science*, the *European Journal of Operational Research*, *Computers and Industrial Engineering*, *Computers and Operations Research*, *Location Science*, *Omega*, *Transportation Research*, and the *Journal of Operational Research Society* regularly carry papers on business applications of operations research. Societies such as International Federation of Operational Research Societies (IFORS),<sup>5</sup> the Association of European Operational Research Societies (EURO),<sup>6</sup> Institute for Operations Research and the Management Sciences (INFORMS),<sup>7</sup> The OR Society,<sup>8</sup> Operational Research Society of India,<sup>9</sup> INFORMS Simulation Society<sup>10</sup> and the Airline Group of IFORS (AGIFORS)<sup>11</sup> host conferences to discuss the advances in operations research theory and applications.

INFORMS holds an annual Franz Edelman competition<sup>12</sup> to select the best applications of operations research in both profit and non-profit sectors; the January/February issue of *Interfaces* features the Franz Edelman winners' papers. The INFORMS Rail Applications Section<sup>13</sup> holds an annual competition on the best solutions to operations research problems typically encountered in the railroad industry. Similar annual competitions are held by Paragon Decision Technology<sup>14</sup> and College Industry Council on Material Handling Education<sup>15</sup> to name a few.

The breadth of applications of operations research in business can be gauged from a reading of the annual Franz Edelman competition winners' accomplishments, few of which are listed below:

- 2013: Dutch Delta Program Commissioner used mixed integer nonlinear programming to derive an optimal investment strategy for strengthening dikes for protection against high water and keeping freshwater supplies up to standard, resulting in savings of €8 billion in investment costs.
- 2012: TNT Express developed a portfolio of multi-commodity and vehicle routing models for package and vehicle routing and scheduling, planning of pickup and delivery, and supply chain optimization for its operations across 200 countries using 2,600 facilities, 30,000 road vehicles, and 50 aircraft resulting, in savings of €207 million over the period 2008–2011 and reduction in CO<sub>2</sub> emissions by 283 million kilograms.
- 2011: Midwest Independent Transmission System Operator used mixed integer programming to determine when each power plant should be on or off, the power plant output levels and prices to minimize the cost of generation, start-up and contingency reserves for 1000 power plants with total capacity of 146,000 MW spread over 13 Midwestern states of U.S. and Manitoba (Canada) owned by 750 companies supplying 40 million users, resulting in savings of \$2 billion over the period 2007–2010.
- 2010: Mexico's central security depository, INDEVAL, used linear programming to develop a secure and automatic clearing and settlement engine to determine the set of transactions that can be settled to maximize the number of traded securities, thereby efficiently processing transactions averaging \$250 billion daily and optimally using available cash and security balances.
- 2009: Hewlett-Packard developed an efficient frontier analysis based Revenue Coverage Optimization tool for analysis of

its order history to manage product variety, thereby enabling increased operational focus on most critical products, making data-driven decisions and increasing its market share, customer satisfaction, and profits by more than \$500 million since 2005.

- 2008: Netherlands Railways developed a constraint programming based railway timetable for scheduling about 5,500 trains daily, while ensuring maximum utilization of railway network, improving the robustness of the timetable, and optimal utilization of rolling stock and crew thereby resulting in additional annual profit of €40 million.
- 2005: Motorola used mixed integer programming to develop Internet-enabled supplier negotiation software with flexible bidding formats, multistage negotiation capabilities, multiple online negotiation formats (with reverse e-auction, online competitive bidding facilities), and optimized selection of vendors to support its global procurement function, automated negotiations, and the management of a heterogeneous supply base of more than thousand vendors with different product portfolios and delivery capabilities, thereby resulting in savings of more than \$600 million.

## 2. Linear Programming

One of the most popular tools of Operations Research is Linear Programming. The subsequent sub-sections will explore the nature of Linear Programming through real life examples.

### 2.1 Advertising Problem

How can we mathematically represent or model real life situations? To understand this, let us take the case of a company which is planning to advertise its newly launched product in two TV channels, TV1 and TV2. The company has set aside a budget of 1,000,000 GMD<sup>16</sup> for this TV advertising campaign. An advertisement costs 100,000 GMD and 200,000 GMD per minute in channels TV1 and TV2 respectively. TV1

and TV2 have informed that advertisement slots can last a maximum of 3 and 4 minutes respectively. The company has found that there is a linear relationship between the time of advertising and the number of viewers of the advertisements. The company has estimated that the viewership is 2 and 3 million per minute of advertisements on TV1 and TV2 respectively. Thus if the advertisement is shown for 3 minutes, the viewership would be 6 and 9 million for TV1 and TV2 respectively. The company wishes to determine the number of minutes that the advertisement should be aired on TV1 and TV2 such that the number of viewers is maximized.

It is obvious that there can be a number of combinations of time of advertisement of TV1 and TV2 channels, few of which are listed as follows.

Let us look at each of these combinations in turn:

- Combination Number 1: Here, we have been able to spend only 300,000 GMD out of the budget of 1,000 thousand GMD. Thus there must be a better combination than this sub-optimal solution.
- Combination Number 2: Here, we have used the maximum time allowed by the channels TV1 and TV2. But we need 1,100 thousand GMD whereas our budget is only 1,000 thousand GMD. This combination is therefore not feasible on account of the cost exceeding the budget.
- Combination Number 3: Here, though the total cost equals the advertising budget, the time period chosen is 4 minutes for TV1, whereas TV1 allows only 3 minutes of advertising. This combination is therefore not feasible on account of the time period chosen for TV1 exceeding the maximum time allowed for advertising slots.
- Combination No 4: Here we have been able to spend only 900,000 GMD out of the budget of 1,000 thousand GMD. Is this the best solution? Or is there a better solution?

We can find the best solution through mathematical representation or modelling of the problem. For this purpose, we use letters  $x$  and  $y$  to represent the number of minutes that the advertisement will be aired on TV1

Table 1.1

Combination number	Minutes on TV1	Minutes on TV2	Viewers on TV1 (in million)	Viewers on TV2 (in million)	Total viewers (in million)	Cost for TV1 (in thousands)	Cost for TV2 (in thousands)	Total Cost (in thousands)
1	1	1	2	3	5	100	200	300
2	3	4	6	12	18	300	800	1100
3	4	3	8	9	17	400	600	1000
4	3	3	6	9	15	300	600	900

and TV2. These letters are termed **variables** because the numbers that these letters represent can vary. Thus  $x$  could take a value of 1 or 1.5 or 2.3 or 4 and so on. Further, because the problem involves taking a decision regarding the numbers that these letters can represent, these letters are termed as **decision variables**. Deciding the decision variables is the first step in the modelling process.

The next step in the modelling process decides the manner in which the decision maker evaluates the choice of the number of minutes that the advertisement will be aired on TV1 and TV2. How is a choice of 2 minutes each on TV1 and TV2 better than 1 minute each on TV1 and TV2? We are given that “The Company wishes to determine the number of minutes that the advertisement will be aired on TV1 and TV2 such that the number of viewers is maximized.” This tells us that the evaluation of the choice of numbers for the decision variables is done on the basis of viewership. The viewership is 2 and 3 million per minute of advertisements on TV1 and TV2 respectively. We thus create an algebraic expression  $2x + 3y$  of viewership using the decision variables  $x$  and  $y$ . The company wishes to maximize the viewership,  $2x + 3y$ , with feasible choice of numbers for decision variables  $x$  and  $y$ . The algebraic expression  $2x + 3y$  is known as the **objective function** because it captures the objective of the decision maker. The coefficients of the decision variables in the objective function are known as **objective function coefficients**. Thus, the objective function coefficient of decision variable  $x$  is 2.

The third step in the modelling process is creating algebraic expressions to ensure feasibility of the choice of numbers for the decision variables  $x$  and  $y$ . These algebraic expressions could either be equalities or inequalities. These expressions are known as **constraints**.

Let us consider Combination 2 again. This combination is infeasible because the cost resulting from the choice of numbers for the decision variables  $x$  and  $y$  exceeded the advertising budget. In order to ensure that such a situation does not occur, we create an algebraic equation for cost (in thousands) as  $100x + 200y$  because it costs 100,000 GMD and 200,000 GMD per minute in channels TV1 and TV2 respectively. Because cost cannot exceed the budget of 1,000 thousand GMD, the algebraic expression or constraint  $100x + 200y \leq 1000$  will ensure the appropriate choice of numbers for the decision variables  $x$  and  $y$ . Thus, we have created the first constraint of the model. The coefficients of the decision variables in

the constraints are known as **technological coefficients**. Thus, the technological function coefficient of decision variable  $x$  is 100 for the first constraint. The number on the right hand side of the constraint expression is known as **right hand side** (or **RHS**). The RHS represents the quantity of resource (in this case the advertising budget) that is available. The objective function coefficients, technological coefficients and RHS are collectively known as the **parameters** of the model.

Let us consider Combination 3. This combination was infeasible because the time period chosen for TV1 exceeded the maximum time allowed for advertising slots. TV1 and TV2 have informed that advertisement slots can last a maximum of 3 and 4 minutes respectively. In order to ensure that such situations do not occur, we create algebraic equations  $x \leq 3$  and  $y \leq 4$  for TV1 and TV2 respectively. Thus, we have created the second and third constraints of the model.

Because negative numbers cannot be chosen for the decision variables  $x$  and  $y$ , we create algebraic equations  $x \geq 0$  and  $y \geq 0$  as the fourth and fifth constraints of the model.

Thus, the mathematical representation or model of the situation is:

$$\text{Maximize } 2x + 3y$$

Subject to the following constraints:

$$100x + 200y \leq 1000 \text{ (Constraint 1)}$$

$$x \leq 3 \text{ (Constraint 2)}$$

$$y \leq 4 \text{ (Constraint 3)}$$

$$x \geq 0 \text{ (Constraint 4)}$$

$$y \geq 0 \text{ (Constraint 5)}$$

Let us look at two aspects of the above model carefully:

- i. The objective functions and constraints are linear functions. A linear function is a polynomial expression of degree zero or one. A polynomial is an expression constructed from variables and constants using the operations of addition, subtraction, multiplication, and non-negative integer exponents. Thus  $2x + y - \frac{z}{4} + 8$  and  $2x + y^2 - \frac{z}{4} + 8$

are polynomials while  $2x + y - \frac{4}{z} + 8$  and  $2x + y^{\frac{3}{2}} - \frac{z}{4} + 8$  are not polynomials (because  $z$  has a negative exponent in the first expression, and  $y$  has a non-integer exponent in the second expression). The degree of a term of the polynomial is the sum of the exponents of the variables that appear in it. Thus the degree of term  $2xyz^4$  is 6 in the polynomial  $2xyz^4 + y^2 - \frac{z}{4} + 8$ . The degree of a polynomial is the highest degree of its terms. Thus the degree of the polynomial  $2xyz^4 + y^2 - \frac{z}{4} + 8$  is 6.

- ii. The decision variables  $x$  and  $y$  are allowed to assume fractional values (such as 2.3).

The above model is thus a **linear programming (LP)** model because it satisfies the conditions i and ii. Linear Programming models are solved using primal simplex algorithm, dual simplex algorithm, and barrier or interior point methods.

## 2.2 Other Types of Models

The other types of models are:

- If the objective function is nonlinear, while constraints are linear, the model is known as **Linearly Constrained Optimization** model. If the objective function is quadratic (some terms involve square of a variable or product of two variables), whereas constraints are linear, the model is known as **Quadratic Programming** model. An example of Quadratic Programming model is *Maximize*  $2x + 3y - x^2 - xy$  subject to  $x + 4y \leq 28$ . If the objective function is nonlinear and there are no constraints, the model is known as **Unconstrained Nonlinear Programming** model.
- If a few decision variables are constrained to be integers, the model is known as **Mixed Integer Linear Programming** model. If all decision variables are constrained to be integers, the model is known as **Pure Integer Linear Programming** model. For example, if the TV channels require that the time periods be integers, we have to incorporate additional

constraints, restricting the choice of decision variables as positive integers. In many cases, we may model a situation with **binary decision variables** which take a value of 0 or 1. If all the decision variables are constrained to be binary integers, the model is known as **Binary Programming** model.

- If the parameters are not known with certainty, we use **Stochastic Programming** models.

### 2.3 Modeling Software

Modelling and solver software are available on Windows, Linux, and MacOS platforms to formulate the models in a modelling language, solve them, and obtain the solutions in the desired format. A few examples of such software are LINGO (*LINDO Systems*), AIMMS (*Paragon Decision Technology*), AMPL (*AMPL Optimization*), CPLEX Optimization Studio (*IBM Corporation*), GAMS (*GAMS Development Corporation*), MPL Modeling System (*Maximal Software*), Premium Solver (*Frontline Systems*) and Gurobi (*Gurobi Optimization*). Readers can access FAQs and surveys of available software in the OR-MS Today magazine (<http://www.orms-today.org/ormsmain.shtml>),<sup>17</sup> Analytics magazine (<http://analytics-magazine.org/>),<sup>18</sup> Decision Tree for Optimization Software website (<http://analytics-magazine.org/>),<sup>19</sup> OR-Exchange website,<sup>20</sup> NEOS Guide website<sup>21</sup> and COIN-OR website.<sup>22</sup> Solver software can also be accessed through the NEOS server.<sup>23</sup>

We will use LINGO software for the purpose of demonstration of models in this book. The demo version of LINGO is sufficient for solving the problem instances discussed in this book. The demo version of LINGO for Windows and Linux operating systems can be accessed at the “Downloads” page of the website <http://www.lindo.com>.

### 2.4 Using LINGO Software

We will use the TV advertising example described in Section 2.1 to introduce the LINGO software. The entry of the above model in the LINGO software is done as follows:

After installation and on clicking on the LINGO icon, the screen given in Figure 1.1. will appear.

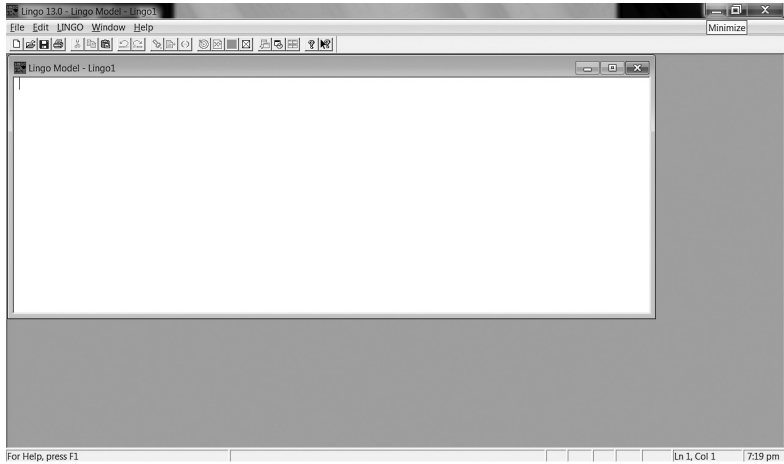


Figure 1.1

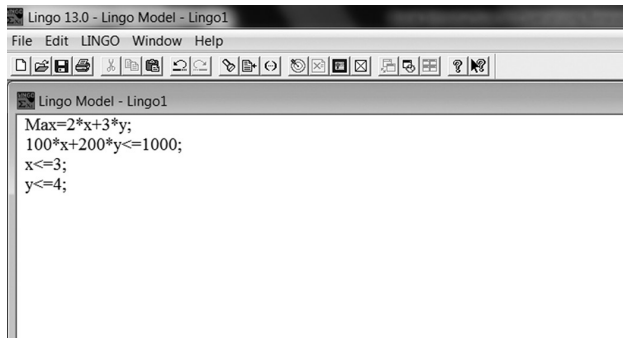


Figure 1.2

Type the model in the child window labelled “Lingo Model-Lingo1” (known as Model Window) as given in Figure 1.2.

The following points may be noted here:

- The objective function is to be maximized. Hence we have written “Max=” followed by the objective function expression. If the function had to be minimized, we would have written “Min=” followed by the objective function expression.
- Mathematical products are represented by the “\*” symbol. For example,  $2x$  is written as “2\*x.”
- All expressions (whether objective function or constraint function) have to be terminated by a semicolon, “;.”

- The less than or equal to expression in the  $x \leq 3$  constraint has to be written as a combination of two symbols, “<” followed by “=.” Similarly, a greater than or equal to expression in a constraint has to be written as a combination of two symbols, “>” followed by “=.”
- The  $x \geq 0, y \geq 0$  constraints are not required to be included in the LINGO model because these constraints are assumed. However, if  $x$  is constrained to be a positive integer variable, a constraint given by “@GIN(x);” has to be included. If  $x$  is constrained to be a binary integer variable, a constraint given by “@BIN(x);” has to be included.
- The LINGO modelling language is not case sensitive. Hence, “x” and “X” mean the same in LINGO modelling language. Hence, we cannot have two different variables “x” and “X”—they should be differentiated by suffixes or prefixes say, “x1” and “x2.”
- Comments may be included in the program for ease of referencing using a prefix “!” and suffix “;.”

#### 2.4.1 Solving the LINGO Program

The model is solved by clicking on the bull’s eye or using the shortcut “Control+U” (which requires pressing both “Ctrl” and “U” buttons simultaneously). If the model has an error (say “;” is missing after the first constraint), an error message will be given as shown in Figure 1.3.

Else, if the model is correct and an optimal solution is obtained, a solution output of the form given in Figure 1.4 is displayed.

The following information is obtained from the solution report:

- The second line from the top: The objective value 16.5 implies that the optimal value of the objective function is 16.5. Thus the maximum viewership that can be obtained under the circumstances is 16.5 million.
- The seventh and sixth lines from the bottom give the optimal values of  $x = 3$  minutes and  $y = 3.5$  minutes.
- The sixth, seventh and eight lines from the top gives the total number of variables, non-linear and integer variables

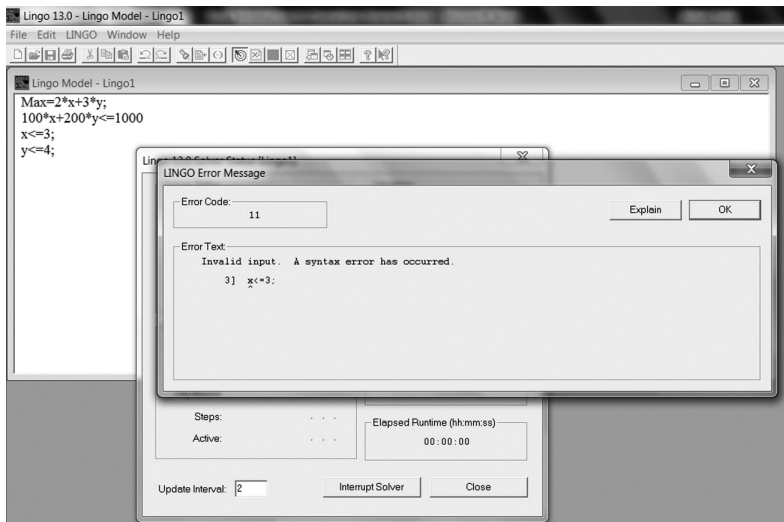


Figure 1.3

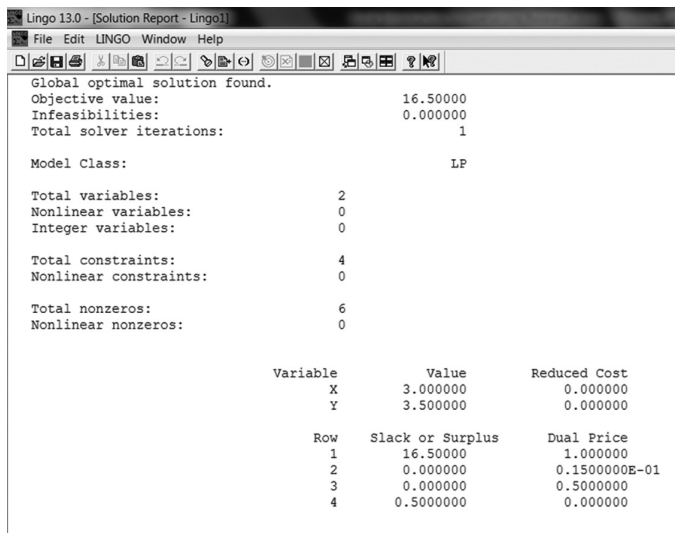


Figure 1.4

used in the model. It is seen that only two variables are used in the model. Software usually has limits on the number of variables that can be used for the model. For example, the demo version of LINGO used for this book has limits of 500 variables, 50 integer variables, and 50 non-linear variables.

- The ninth and tenth lines from the top give the total number of constraints and non-linear constraints used in the model. The demo version of LINGO has limits of 250 constraints. All the examples in this book use models which are within the variable and constraint limits of the demo version of LINGO.

### 2.4.2 Slack and Dual Prices

The second, third, and fourth lines from the bottom give the slack or surplus and the dual prices associated with each constraint. The slack gives the resource remaining for the optimal solution. For example, the slack associated with constraint 3 (constraint  $y \leq 4$  associated with the maximum time period allowed for TV2) is 0.5 because the optimal solution is  $y = 3.5$ . The surplus is discussed later in Section 2.6.1. The dual price or shadow price is the marginal value of resource associated with the constraint. Thus, the dual price indicates the rate at which the objective would improve on slight increase of the associated resource. For example, the dual price for constraint 1 (which is associated with the advertising budget) is 0.015 (0.1500000E-01 given in the solution report means  $0.15 \times 10^{-1}$ ) as given in the solution report (third line from the bottom). This implies that if the budget is increased from 1000 to 1001, the objective value will improve from the current value of 16.5 to  $16.515 (= 16.5 + 1 \times 0.015)$ . If the budget is increased from 1,000 to 1,003, the objective value will improve from the current value of 16.5 to  $16.545 (= 16.5 + 3 \times 0.015)$ . If the budget is decreased from 1000 to 999, the objective value will reduce from the current value of 16.5 to  $16.485 (= 16.5 - 1 \times 0.015)$ . You can check the changes in the objective value by running the program with different budgets in the right hand side of constraint 1.

### 2.4.3 Reduced Cost

It will be seen that the sixth and seventh eight lines from the bottom of the solution report give the optimal values of  $x$  and  $y$  and their associated reduced costs. The reduced cost is defined as the amount by which the objective function coefficient of a decision variable must be improved such that the optimal solution of that decision variable becomes non-zero.

Obviously, if the optimal solution of a variable is non-zero, its reduced costs are zero. Thus the reduced costs given for  $x$  and  $y$  are zero.

To understand reduced costs, change the objective function from *Maximize*  $2x + 3y$  to *Maximize*  $2x + 30y$  and constraint 3 from  $y \leq 4$  to  $y \leq 40$  in the LINGO program and run the LINGO program again. It will be seen that we get optimal values as  $x = 0$  and  $y = 5$  and the associated reduced costs as 13 and 0 respectively. Now, if we improve the coefficient of  $x$  in the objective function from 2 to 16 ( $>2 + 13$ ) and run the LINGO program again, it will be seen that the optimal value of  $x$  changes from 0 to 3.

#### 2.4.4 Multiple Solutions

Let us now explore another situation using a slightly changed model:

$$\text{Maximize } x + 2y$$

Subject to the following constraints:

$$100x + 200y \leq 1000 \text{ (Constraint 1)}$$

$$x \leq 3 \text{ (Constraint 2)}$$

$$y \leq 4 \text{ (Constraint 3)}$$

Here, we have changed the coefficients of the variables in the objective function, whereas the constraints remain the same. If we enter the above model in the LINGO software and run it, we find that we get the optimal value of the objective function as 10 and the optimal solution of variables as  $x = 3$  and  $y = 3.5$ . We will also find from the second row from the bottom that both the slack and dual price associated with the Constraint 2 is zero. Whenever the slack and dual price associated with a constraint is zero, it signifies that the problem has multiple optimal solutions. This situation arises because the straight lines given by the objective function and constraint 1 are parallel to each other.<sup>24</sup> To find the other multiple solutions, we slightly change the coefficient of any variable in the objective function. For example, we change the objective function to  $x + 2.01y$ . If we change the objective function in the LINGO software and run it,

we get the optimal solution of the decision variables as  $x = 2$  and  $y = 4$ . Note that the optimal value of the objective function  $x + 2y$  is 10 for both the optimal solutions (i)  $x = 3$  and  $y = 3.5$  and (ii)  $x = 2$  and  $y = 4$ . Once we have two optimal solutions, we can generate other optimal solutions using the equations  $x = 3w_1 + 2w_2$ ,  $y = 3.5w_1 + 4w_2$  where  $w_1 + w_2 = 1$ . For example, if we choose  $w_1 = 1$  and  $w_2 = 0$ , we get  $x = 3$  and  $y = 3.5$ ; if we choose  $w_1 = 0$  and  $w_2 = 1$ , we get  $x = 2$  and  $y = 4$ ;  $w_1 = 0.5$  and  $w_2 = 0.5$ , we get  $x = 2.5$  and  $y = 3.75$ ;  $w_1 = 0.2$  and  $w_2 = 0.8$ , we get  $x = 2.2$  and  $y = 3.9$  and so on. It can be seen that the optimal value of the objective function  $x + 2y$  is always 10 for these different multiple solutions of  $x$  and  $y$ .

2.4.4.1 Let us now explore again another situation using a slightly changed model:

$$\text{Maximize } 2x + 3y$$

Subject to the following constraints:

$$200x + 300y \leq 1000 \text{ (Constraint 1)}$$

$$x \leq 3 \text{ (Constraint 2)}$$

$$y \leq 4 \text{ (Constraint 3)}$$

Here, we have changed the coefficients of the variables in constraint 1, whereas the other constraints and the objective function remain the same. If we enter the above model in the LINGO software and run it, we find that we get the optimal value of the objective function as 10 and the optimal solution of variables as  $x = 0$  and  $y = 3.33$ . We also find from the seventh row from the bottom of the solution report, that both the value and reduced cost associated with variable  $x$  is zero. Whenever the value and reduced cost associated with a decision variable is zero, it signifies that the problem has multiple optimal solutions. This situation arises because the straight lines given by the objective function and constraint 1 are parallel to each other. To find the other multiple solutions, we slightly change the coefficient variable  $y$  in constraint 1. For example, we change the constraint 1 to  $200x + 300.1y \leq 1000$ . If we change the

constraint 1 in the LINGO software and run it, we get the optimal solution of variables as  $x = 3$  and  $y = 1.33$ . Note that the optimal value of the objective function  $2x + 3y$  is 10 for both the optimal solutions (i)  $x = 0$  and  $y = 3.33$  and (ii)  $x = 3$  and  $y = 1.33$ . Once we have two optimal solutions, we can generate other optimal solutions using the equations  $x = 0w_1 + 3w_2$ ,  $y = 3.33w_1 + 1.33w_2$  where  $w_1 + w_2 = 1$ .

It is thus seen that the dual price and reduced cost generated in the solution reports of linear programs allow us to gain insights into the problem. This analysis using dual prices and reduced costs is known as **sensitivity analysis**. The sensitivity analysis allows us to understand the impact of changes in parameters on the optimal solution. This manner of sensitivity analysis is only possible for problems using linear programming.

## 2.5 Unbounded Model

Let us examine the mathematical representation or model of the TV advertising situation given below again:

$$\text{Maximize } 2x + 3y$$

Subject to the following constraints:

$$100x + 200y \leq 1000 \text{ (Constraint 1)}$$

$$x \leq 3 \text{ (Constraint 2)}$$

$$y \leq 4 \text{ (Constraint 3)}$$

If there no constraints, the value of the objective function can increase without any limits because there are no limits on the values taken by the decision variables  $x$  and  $y$ . Hence we cannot arrive at any optimal solution. This situation is known as an **unbounded objective**. If we were to run the LINGO program with all the constraints removed, we will get an “Unbounded Solution” message.

## 2.6 Minimization Objective

Let us look at the advertising problem in another way. We take the same case of a company which is planning to advertise its newly launched

product in two TV channels, TV1 and TV2. Here, the company has not set aside a budget for this TV advertising. However it wishes to minimize its cost of advertising. An advertisement costs 100,000 GMD and 200,000 GMD per minute in channels TV1 and TV2 respectively. TV1 and TV2 have informed that advertisement slots can last a maximum of 3 and 4 minutes respectively. The company has found that there is a linear relationship between the time of advertising and the number of viewers of the advertisements. The company has estimated that the viewership is 2 and 3 million per minute of advertisements on TV1 and TV2 respectively. Thus, if the advertisement is shown for 3 minutes, the viewership would be 6 and 9 million for TV1 and TV2 respectively. The company wishes to determine the number of minutes that the advertisement will be aired on TV1 and TV2 such that the number of viewers is more than 10 million.

Here too,  $x$  and  $y$  representing the number of minutes that the advertisement will be aired on TV1 and TV2 will be the decision variables. However, the objective of the decision maker here is to minimize the cost of advertising. Hence the objective function here is:

$$\text{Minimize } 100x + 200y$$

TV1 and TV2 have informed that advertisement slots can last a maximum of 3 and 4 minutes respectively. In order to ensure that such situations do not occur, we create algebraic equations  $x \leq 3$  and  $y \leq 4$  for TV1 and TV2 respectively. Thus, we have created the first and second constraints of the model.

The decision maker wishes to ensure that the number of viewers is more than 10 million. This is modelled by the third constraint:

$$2x + 3y \geq 10$$

Thus, the mathematical representation or model of the situation is:

$$\text{Minimize } 100x + 200y$$

Subject to the following constraints:

$$x \leq 3 \text{ (Constraint 1)}$$

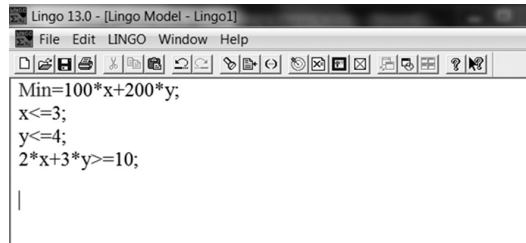


Figure 1.5

$$y \leq 4 \quad (\text{Constraint 2})$$

$$2x + 3y \geq 10 \quad (\text{Constraint 3})$$

We type the model in the LINGO Model Window as given in Figure 1.5.

On solving the model using “Control+U” or clicking on the bull’s eye, we get the optimal solution as  $x = 3$  and  $y = 1.33$  minutes and an advertising expenditure of 567,000 GMD.

### 2.6.1 Slack Variables

If the company required that a minimum of 400 thousand GMD must be spent on advertising, we will need a fourth constraint  $100x + 200y \geq 400$ . On solving the LINGO model, we get the same optimal solution of  $x = 3$  and  $y = 1.33$  minutes and a cost of 566.67 thousand. Thus the optimal cost is 166.67 thousand more than the minimum required by the fourth constraint. This figure of 166.67 is the **surplus** associated with the fourth constraint at the optimal solution. Surplus is thus associated with “ $\geq$ ” constraints. The LINGO solution report (last line from the bottom) will report this surplus against the fourth constraint.

### 2.6.2 Infeasible Model

Now let us take the case of a company wishing to determine the number of minutes that the advertisement will be aired on TV1 and TV2 such that the number of viewers is more than 100 million (instead of 10 million earlier). Thus the model is revised as follows:

$$\text{Minimize } 100x + 200y$$

Subject to the following constraints:

$$x \leq 3 \text{ (Constraint 1)}$$

$$y \leq 4 \text{ (Constraint 2)}$$

$$2x + 3y \geq 100 \text{ (Constraint 3)}$$

The company has estimated that the viewership is 2 and 3 million per minute of advertisements on TV1 and TV2 respectively. Thus, if the advertisement is shown for  $x$  minutes on TV1 and  $y$  minutes on TV2, the viewership will be  $2x + 3y$  million. However, constraint 1 requires that  $x \leq 3$ , and constraint 2 requires that  $y \leq 4$ . If we take the maximum allowed values of  $x$  and  $y$  as 3 and 4 minutes respectively, the maximum possible viewership is 18 million ( $= 2 * 3 + 3 * 4$ ). Hence, we can never get a viewership more than 100 million that is required by constraint 3. Thus the model is **infeasible**. If we were to run the LINGO program with this model, we will get a “No feasible solution found” message.

### 3. Integer Programming Model

Let us consider the TV advertising problem described in Section 2.1. Let us assume that the TV channels require that the number of minutes that the advertisement will be aired must be integers. The model thus needs to be augmented with additional requirements that both the decision variables  $x$  and  $y$  have to be integers. The integer requirement for the decision variables  $x$  and  $y$  can be modelled in LINGO using the fourth and fifth constraints  $@GIN(x)$  and  $@GIN(y)$ . The model is entered in LINGO software as given in Figure 1.6.

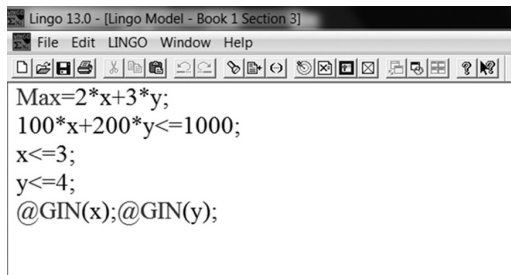


Figure 1.6

Solving the model, we get the optimal solution as  $x = 2$  and  $y = 4$  minutes.

### 3.1 Binary Integer Programming Model

Let us consider the TV advertising problem described in Section 2.1 along with the requirement that the number of minutes that the advertisement will be aired must be integers. Further, let us assume that the company wishes to air its advertisement in either channel TV1 or channel TV2, but not in both.

Binary decision variables which take only two values 0 and 1 become very useful to model such situations where we have to choose between two or more options. Let us choose a decision variable  $z_1$ , which takes value 1 if the advertisement is aired on channel TV1 and 0 if the advertisement is not aired on channel TV1. Similarly we choose a decision variable  $z_2$  which takes value 1 if the advertisement is aired on channel TV2 and 0 if the advertisement is not aired on channel TV2. Because the advertisement has to be aired on either channel TV1 or channel TV2 but not in both, we have to incorporate an additional sixth constraint  $z_1 + z_2 = 1$  so that

- i.  $z_1, z_2$  both do not become 0, which means that the advertisement is not aired on either channel (a situation which is not desired)
- ii.  $z_1, z_2$  both do not become 1, which means that the advertisement is aired on both channels (a situation which is also not desired).

Further we need to model that decision variable  $z_1$  will take value 1 whenever the decision variable  $x > 0$ . This is modelled by the constraint  $x \leq Mz_1$ , where  $M$  is a very large number (say 500 in the context of the problem being discussed). If  $x > 0$ , the constraint forces the decision variable  $z_1$  to take the value 1. For example if  $x = 5$ , the decision variable  $z_1$  will have to become 1 in the expression  $x \leq Mz_1$ . A similar constraint  $y \leq Mz_2$  ensures that decision variable  $z_2$  takes value 1 whenever the decision variable  $y > 0$ .

Thus the model is:

$$\text{Maximize } 2x + 3y$$

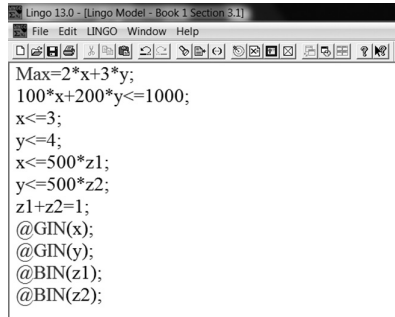


Figure 1.7

Subject to the following constraints:

$$100x + 200y \leq 1000 \text{ (Constraint 1)}$$

$$x \leq 3 \text{ (Constraint 2)}$$

$$y \leq 4 \text{ (Constraint 3)}$$

$$x \leq Mz_1 \text{ (Constraint 4)}$$

$$y \leq Mz_2 \text{ (Constraint 5)}$$

$$z_1 + z_2 = 1 \text{ (Constraint 6)}$$

$$x, y \geq 0 \text{ and integers (Constraints 7 \& 8)}$$

$$z_1, z_2 = 0, 1 \text{ (Constraints 9 \& 10)}$$

The binary integer requirement for the decision variables  $z_1, z_2$  can be modelled in LINGO using the ninth and tenth constraints  $@BIN(z_1)$  and  $@BIN(z_2)$ . The model is entered in LINGO software as given in Figure 1.7.

Solving the model, we get the optimal solution as  $x = 0$  and  $y = 4$  minutes. Thus the advertisement is aired on TV2 only.

## 4. Using SETS in LINGO

Let us consider the TV advertising problem described in Section 2.1 with a few modifications. We have the company which is planning to advertise its newly launched product in two TV channels TV1 and TV2. It is now

considering two prime time slots S1 (7 am to 8 am) and S2 (8 pm to 9 pm). The cost of advertisement (in thousands per minute) for slots S1 and S2 in channels TV1 and TV2 respectively; the maximum time period (in minutes) allowable for slots S1 and S2 in channels TV1 and TV2; the company's estimate of the viewership (in millions) per minute of advertisements for slots S1 and S2 in TV1 and TV2 are all given in the table below (Table 1.2).

**Table 1.2**

	TV1		TV2	
	S1	S2	S1	S2
Cost of advertisement (in thousands per minute)	80	200	100	150
Maximum time period (in minutes) allowable	3	2	5	4
Company's estimate of the viewership (in million) per minute of advertisements	2	3	4	2

The company has found that there is a linear relationship between the time of advertising and the number of viewers of the advertisements. The company has set aside a budget of 1,000,000 GMD for this TV advertising. The company wishes to determine the number of minutes that the advertisement will be aired on slots S1 and S2 in TV1 and TV2 such that the number of viewers is maximized. Let us assume that the TV channels require that the number of minutes that the advertisement will be aired must be integers.

Let us assume that  $x_{ij}$  be the decision variable giving the number of minutes that the advertisement is aired in slot  $i$  (where  $i = 1$  indicates slot S1 and  $i = 2$  indicates slot S2) on TV channel  $j$  (where  $j = 1$  indicates channel TV1 and  $j = 2$  indicates channel TV2).

We indicate

- $c_{ij}$  as the cost of advertisement (in thousands per minute) aired in slot  $i$  of TV channel  $j$
- $t_{ij}$  as the maximum time period (in minutes) allowable for advertisement aired in slot  $i$  of TV channel  $j$
- $v_{ij}$  as the estimated viewership (in million) per minute of advertisement aired in slot  $i$  of TV channel  $j$ . We can read their values from the table above; thus, for example  $c_{12} = 100,000$ ,  $t_{11} = 3$  minutes and  $v_{21} = 3$  million.

Using the same logic used for obtaining the objective function in Section 2.1, we can obtain the objective function here as follows:

$$\text{Maximize } 2x_{11} + 4x_{12} + 3x_{21} + 2x_{22}$$

or,

$$\text{Maximize } v_{11}x_{11} + v_{12}x_{12} + v_{21}x_{21} + v_{22}x_{22}$$

or,

$$\text{Maximize } \sum_{i=1}^2 (v_{i1}x_{i1} + v_{i2}x_{i2})$$

or,

$$\text{Maximize } \sum_{i=1}^2 \sum_{j=1}^2 v_{ij}x_{ij}$$

Similarly, the constraint 1 of Section 2.1 relating to the budget can be framed as follows:

$$\sum_{i=1}^2 \sum_{j=1}^2 c_{ij}x_{ij} \leq 1000$$

Constraints 2 and 3 of Section 2.1 relating to the maximum allowable time are modelled here as follows:

$$x_{ij} \leq t_{ij}, \forall i = 1, 2 \text{ and } j = 1, 2$$

Here the symbol  $\forall$  indicates “for every.” Thus, this constraint implies that  $x_{11} \leq t_{11}$ ,  $x_{12} \leq t_{12}$ ,  $x_{21} \leq t_{21}$ , and  $x_{22} \leq t_{22}$ .

Further, there are constraints to model the integer property of the decision variable  $x_{ij}$  as follows:

$$x_{ij} \text{ integer}, \forall i = 1, 2 \text{ and } j = 1, 2$$

We introduce the SETS function of LINGO software to model the problem. A set is a group of similar objects. For example, in this case, we have a set “TV” of TV channels and another set “TS” of time slots. We can further obtain **derived sets** through combination of **primitive sets**. Thus we have a derived set S1 obtained by combining the primitive sets

of time sets “TS” and TV channels “TV.” Primitive and derived sets have characteristics associated with them called **attributes**. For example, the derived set S1 has four attributes discussed earlier associated with each combination of time slot and TV channel:

- i. C being the cost of advertisement (in thousands per minute) aired
- ii. T as the maximum time period (in minutes) allowable for advertisement
- iii. V as the estimated viewership (in million) per minute of advertisement aired
- iv. X as the decision variable giving the number of minutes that the advertisement is aired.

These SETS are declared in a SETS section which starts with a keyword “SETS:” and ends with the keyword “ENDSETS” as follows:

```
SETS:
TV;
TS;
S1(TS,TV):C,T,V,X;
ENDSETS
```

The order in which the primitive sets “TS” and “TV” are written within the brackets for the derived set S1 must be followed throughout the rest of the program.

The next DATA section of the LINGO program will contain the data for the members of the primitive and derived sets declared in the SETS section. The DATA section starts with a keyword “DATA:” and ends with the keyword “ENDDATA” as follows:

```
DATA:
TV=TV1 TV2;
TS=S1 S2;
C=80 100 200 150;
T=3 5 2 4;
V=2 4 3 2;
ENDDATA
```

The following aspects may be noted here:

- Data are separated by blank space(s) or comma(s).
- Data for derived set attribute C is read by LINGO software in the order  $c_{11}$ ,  $c_{12}$ ,  $c_{21}$ , and  $c_{22}$ . This is related to the declaration of S1 as (TS, TV). Hence, LINGO software first chooses the first value of TS and reads off the data for different values of TV, followed by choosing the second value of TS and reading off the data for different values of TV. The principle is that the data corresponding to the last primitive set is read before incrementing the next primitive set in the derived set.

The advantage of a separate DATA section is that the same model can be reused with changes in the DATA section alone for changes in parameters. Scaling of the model becomes very easy too; once we have a working model with 2 TV channels and 2 time slots, the same model can be used for 80 TV channels and 10 time slots with changes in the DATA section alone.

The power of SET modelling lies in set looping functions “@SUM” and “@FOR.” The “@SUM” function computes the sum of all members of a set. The function @SUM(S1(I,J):V(I,J)\*X(I,J)) is equivalent to  $v_{11}x_{11} + v_{12}x_{12} + v_{21}x_{21} + v_{22}x_{22}$  or  $\sum_{i=1}^2 \sum_{j=1}^2 v_{ij}x_{ij}$ . Note that we specify the derived set S1 over which the product of V(I,J) and X(I,J) has to be summed. A “@SUM” function must contain the specification of a set over which the summation has to be done.

The constraint  $x_{ij} \leq t_{ij}$ ,  $\forall i=1,2$  and  $j=1,2$  can be similarly  $x_{ij} \leq t_{ij}$ ,  $\forall i=1,2$  and  $j=1,2$  represented by @FOR(S1(I,J):X(I,J) <=T(I,J)). Note that here too, we specify the derived set S1 over which the X(I,J) is compared with T(I,J).

Thus the constraints are written in LINGO as follows (Table 1.3):

**Table 1.3**

Constraint	LINGO Representation
$\sum_{i=1}^2 \sum_{j=1}^2 c_{ij}x_{ij} \leq 1000$	@SUM(S1(I,J):C(I,J)*X(I,J))<=1000
$x_{ij} \leq t_{ij}$ , $\forall i=1,2$ and $j=1,2$	@FOR(S1(I,J):X(I,J)<=T(I,J))
$x_{ij}$ integer, $\forall i=1,2$ and $j=1,2$	@FOR(S1(I,J):@GIN(X(I,J)))

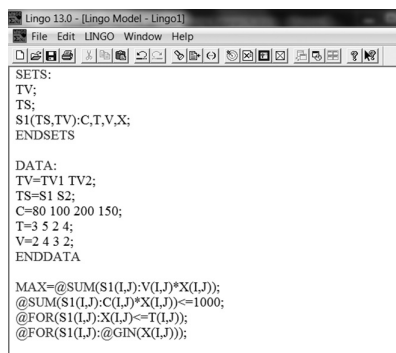


Figure 1.8

Table 1.4

TV1		TV2	
S1	S2	S1	S2
3	1	5	0

The model is entered in LINGO software as given in Figure 1.8.

The solution  $X$  in terms of minutes of advertisement in the time slots  $S1$  and  $S2$  of channels  $TV1$  and  $TV2$  is obtained as given in Table 1.4.

SETS have been used in the formulation of Facility Location Problems (Chapter 3), Multi-commodity Transport Planning Problem (Chapter 11), Supplier Selection Problem (Chapter 14), Portfolio Management Problem (Chapter 16), Bank Asset Liability Management Problem (Chapter 18), Index Fund Construction Problem (Chapter 19), and Performance Measurement Problem (Chapter 21).

## 5. Problems Discussed in Book

The Table 1.5 gives the list of examples given in this book, which use Linear Programming and Integer Programming. There are a few problems covered in this book that use neither linear programming nor integer programming. The Portfolio Management Problem (Chapter 16) uses Non-linear programming. Few situations exist where solutions are easily obtained by algorithms or heuristics. Examples where heuristics have been used in this book are Cable Layout Problem (Chapter 4),

Single Delivery Truck Routing Problem (Chapter 12) and Multiple Delivery Trucks Routing Problem (Chapter 13). Also, there exist other operations research techniques such as queuing theory, which has been used for designing the number of check-in counters (Chapter 5).

Table 1.5

Linear Programming	Integer Programming
Production Planning (Chapter 8) Blending of Dog Diet (Chapter 9) Paper Roll Trimming (Chapter 10) Revenue Management (Chapter 15) Bank Asset Liability Management (Chapter 18) Performance Measurement using DEA (Chapter 21)	Product Mix (Chapter 2) Facility Location (Chapter 3) Scheduling of a Production Line (Chapter 6) Shift Staff Planning (Chapter 7) Multi-commodity Transport Planning (Chapter 11) Supplier Selection with Multiple Criteria (Chapter 14) Capital Budgeting (Chapter 17) Index Fund Construction (Chapter 19) Airline Network Design (Chapter 20)

## SECTION 2

# Applications in Operations Management



## CHAPTER 2

# Product Mix

Let us take the case of a company that manufactures four products P1, P2, P3, and P4. Each of these products requires processing in three treatment plants T1, T2, and T3. The time required for processing (in hours) of each unit of products P1, P2, P3, and P4 in the plants T1, T2, and T3 is given in Table 2.1.

Each of the treatment plants T1, T2, and T3 are available for 100 hours per week. The profit obtained by selling the products P1, P2, P3, and P4 is 50, 100, 30, and 27, respectively. Whereas there is enough demand for products P1, P3, and P4, the company forecasts a maximum demand of 3 units for product P2. How many units of the products P1, P2, P3, and P4 should be produced to maximize the demand?

To solve this problem, let us define integer decision variables  $x_i$  as the number of units of product  $P_i$  to produce. We have to maximize the profit. Hence the objective function is given by:

$$\sum_{i=1}^4 x_i f_i$$

where  $f_i$  is the profit obtained by selling the product  $P_i$ .

If  $t_{ij}$  is defined as the time required for processing product  $P_i$  on machine  $T_j$ , then the following constraints model the hours of availability  $a_j$  of machine  $T_j$ :

$$\sum_{i=1}^4 x_i t_{ij} \leq a_j, \text{ for } j = 1, 2, 3$$

**Table 2.1**

	P1	P2	P3	P4
T1	4	2	1	2
T2	3	6	3	6
T3	5	4	8	6

*Table 2.2*

LINGO Program
<pre>@GIN(x1);@GIN(x2);@GIN(x3);@GIN(x4); max=50*x1+100*x2+30*x3+27*x4; 4*x1+2*x2+x3+2*x4&lt;=100; 3*x1+6*x2+3*x3+6*x4&lt;=100; 5*x1+4*x2+8*x3+6*x4&lt;=100; x2&lt;=3;</pre>

Because the company forecasts a maximum demand of 3 units for product P2, there will be an additional constraint:

$$x_2 \leq 3$$

The LINGO program is given in Table 2.2. The solution to this problem is the production of 17 units of product P1, 3 units of product P2 and none of products P3 and P4.

# CHAPTER 3

## Facility Location

### 1. Supermarkets Serving the Maximum Number of People

Let us consider six neighbouring towns A, B, C, D, E, and F. The distance  $k_{ij}$  between towns  $i$  and  $j$  (in miles) is indicated in Table 3.1 for towns that have roads connecting them. The population of the towns  $P_i$  is also given in the first row of Table 3.1.

A supermarket chain wishes to set up supermarkets at two towns such that the two supermarkets are accessible to the maximum number of people of the six neighbouring towns.<sup>1</sup> Supermarkets are considered accessible if the distance is less than 30 miles.

We define a binary variable  $d_{ij}$  which equals 1 if the town  $j$  is less than 30 miles from town  $i$  and 0 otherwise. Thus  $d_{AB}$  equals 0 (because town B is more than 30 miles away from town A), whereas  $d_{AE}$  equals 1 (because town E is less than 30 miles away from town A). Further,  $d_{AA}$  equals 1, because town A is less than 30 miles away from town A. Also  $d_{AD}$  equals 0, because there is no road connecting towns A and D. The other values of  $d_{ij}$  can be similarly obtained and are given in Table 3.2.

**Table 3.1**

Population (in thousands)	3	8	2	12	7	10
	A	B	C	D	E	F
A	–	50	26	–	15	32
B	50	–	–	34	28	–
C	26	–	–	29	40	17
D	–	34	29	–	18	24
E	15	28	40	18	–	43
F	32	–	17	24	43	–

**Table 3.2**

	A	B	C	D	E	F
A	1	0	1	0	1	0
B	0	1	0	0	1	0
C	1	0	1	1	0	1
D	0	0	1	1	1	1
E	1	1	0	1	1	0
F	0	0	1	1	0	1

We define a decision binary variable  $S_i$  which equals 1 if a supermarket is located at town  $i$ , and 0 otherwise. We also define a decision binary variable  $X_i$  which equals 1 if a town  $i$  is served by at least one supermarket.

Because our goal is to ensure that maximum number of people of the six neighbouring towns can access the super markets, the objective function is:

$$\text{Max} \sum_{i=A,B,\dots}^F P_i X_i$$

Because we intend to have only 2 super markets, there is a constraint:

$$\sum_{i=A,B,\dots}^F S_i \leq 2 \quad (\text{Constraint 1})$$

The decision binary variable  $S_i$  equals 1 if a supermarket is located at town  $i$ . This supermarket at town  $i$  will serve town  $i$  and other towns  $j$  accessible to town  $i$  for which  $X_j$  equals 1. If  $X_A$  equals 1, then  $S_j$  has to equal 1 for at least one of the three towns (A, C, or E), which are accessible from town A (given in the Ath row of Table 3.2) because that supermarket will serve that town A. If  $X_A$  equals 0, it does not matter if  $S_j$  equals 0 or 1 for the towns which are accessible from town A. This can be modelled by the following constraint:

$$X_i \leq \sum_{j=A,B,\dots}^F d_{ij} S_j \text{ for } i = A, B, \dots, F \quad (\text{Constraint 2})$$

The LINGO program is given in Table 3.3. It will be noticed that SETS has been used in the program. There is only one primitive set TOWNS declared in the SETS section (line 2 of Table 3.3). The attributes of

Table 3.3

Line	LINGO Program
1	SETS:
2	TOWNS:P,S,X;
3	S1(TOWNS,TOWNS):D;
4	ENDSETS
5	
6	DATA:
7	NOS=2;
8	TOWNS=A B C D E F;
9	P=3 8 2 12 7 10;
10	D=
11	1 0 1 0 1 0
12	0 1 0 0 1 0
13	1 0 1 1 0 1
14	0 0 1 1 1 1
15	1 1 0 1 1 0
16	0 0 1 1 0 1;
17	ENDDATA
18	
19	@FOR(TOWNS(I):@BIN(S(I)));
20	@FOR(TOWNS(I):@BIN(X(I)));
21	MAX=@SUM(TOWNS(I):P(I)*X(I));
22	@SUM(TOWNS(I):S(I))<=NOS;
23	@FOR(TOWNS(I):X(I)<=@SUM(TOWNS(J):D(I,J)*S(J)));

TOWNS set are P (denoting population), S (denoting decision binary variable  $S_j$ ) and X (denoting decision binary variable  $X_j$ ). A derived set S1 is formed using the primitive sets TOWNS and TOWNS (line 3 of Table 3.3). The attribute of S1 is D (denoting binary variable  $d_{ij}$ ).

The variable NOS in the DATA section (line 7 of Table 3.3) denotes the maximum number of supermarkets that the supermarket chain wishes to set up. The advantage of using the variable NOS is that the optimal solutions for different numbers of supermarkets can be obtained by changing the value of this variable.

The values of attribute P (given in line 9 of Table 3.3) are obtained from the first row of Table 3.1. The values of attribute D (given in lines 11–16 of Table 3.3) are obtained from Table 3.2.

The decision binary variables  $S_i$  and  $X_i$  are declared in lines 19 and 20 of the program given in Table 3.3. The objective function is given in line 21. Constraints 1 and 2 are given in lines 22 and 23, respectively.

The solution for this problem is  $S_D = S_E = 1$ , which implies that setting up supermarkets at D and E cater to the maximum population of all the six towns in the neighbourhood. Towns C, D, and F will be catered by the supermarket in town D. Towns A, B, D, and E will be catered by the supermarket in town E.

## 2. Supermarkets Near the Towns

In case, we want to locate supermarkets in such a way that we wish to minimize the distance between the towns and the supermarkets,<sup>2</sup> the formulation will be different.

We define here a decision binary variable  $S_i$ , which equals 1 if a supermarket is located at town  $i$ , and 0 otherwise. Because we intend to have only two super markets, there is a constraint:

$$\sum_{i=A,B,\dots}^F S_i \leq 2 \quad (\text{Constraint 1})$$

We also define a decision variable  $Y_{ij}$ , which is the fraction of the population of town  $i$  served by a supermarket at town  $j$ . Thus:

$$\sum_{j=A,B,\dots}^F Y_{ij} = 1, \text{ for } i = A, B, \dots, F \quad (\text{Constraint 2}),$$

which implies that all the population of town  $i$  are served by some supermarket. For example, if 50% of the population of town  $A$  is served by a supermarket at town  $C$  and the remaining 50% of population of town  $A$  is served by a supermarket at town  $E$ , then  $Y_{AA} = 0$ ,  $Y_{AB} = 0$ ,  $Y_{AC} = 0.5$ ,  $Y_{AD} = 0$ ,  $Y_{AE} = 0.5$ , and  $Y_{AF} = 0.5$ .

The distance  $W$  which we wish to minimize can be obtained from the constraint:

$$W \geq \sum_{j=A,B,\dots}^F P_i Y_{ij} k_{ij}, \text{ for } i = A, B, \dots, F \quad (\text{Constraint 3})$$

where  $P_i$  is the population of the towns (given in the first row of Table 3.1) and  $k_{ij}$  is the distance between towns  $i$  and  $j$  (indicated in Table 3.1). The distance  $k_{ij}$  for towns which have no roads connecting them is assumed to be a very large number (say 500). The distance  $k_{ii}$  is 0.

The decision binary variable  $S_i$  equals 1 if a supermarket is located at town  $i$ . The decision variable  $Y_{iA}$  which is the fraction of population of town  $i$  served by a supermarket at town A. If  $Y_{iA}$  is more than zero, then  $S_A$  has to equal 1. If  $Y_{iA}$  equals 0, it does not matter if  $S_A$  equals 0 or 1. This can be modelled by the constraint:

$$Y_{ij} \leq S_j \text{ for } i, j = A, B, \dots, F \text{ (Constraint 4).}$$

The LINGO program is given in Table 3.4. There is only one primitive set, TOWNS, declared in the SETS section (line 2 of Table 3.4). The attributes of the TOWNS set are P (denoting population) and S (denoting decision binary variable  $S_j$ ). A derived set S1 is formed using the primitive sets, TOWNS and TOWNS (line 3 of Table 3.4). The attributes of S1 are K (denoting distance  $k_{ij}$ ) and Y (denoting decision variable  $Y_{ij}$ ).

**Table 3.4**

Line	LINGO Program
1	SETS:
2	TOWNS:P,S;
3	S1(TOWNS,TOWNS):K,Y;
4	ENDSETS
5	
6	DATA:
7	NOS=2;
8	TOWNS=A B C D E F;
9	P=3 8 2 12 7 10;
10	K=
11	0 50 26 500 15 32
12	50 0 500 34 28 500
13	26 500 0 29 40 17
14	500 34 29 0 18 24
15	15 28 40 18 0 43
16	32 500 17 24 43 0
17	;
18	ENDDATA
19	
20	@FOR(TOWNS(I):@BIN(S(I)));
21	Min=W;
22	@SUM(TOWNS(I):S(I))<=NOS;
23	@FOR(TOWNS(I):@SUM(TOWNS(J):Y(I,J))=1);
24	@FOR(TOWNS(I):@SUM(TOWNS(J):P(I)*Y(I,J)*K(I,J))<=W);
25	@FOR(S1(I,J):Y(I,J)<=S(J));

The variable NOS in the DATA section (line 7 of Table 3.4) denotes the maximum number of supermarkets that the supermarket chain wishes to set up. The values of attribute P (given in line 9 of Table 3.4) are obtained from the first row of Table 3.1. The values of attribute K (given in lines 11–16 of Table 3.4) are obtained from Table 3.1.

The decision binary variable  $S_i$  is declared in line 20 of the program given in Table 3.4. The objective function is given in line 21. Constraints 1, 2, 3, and 4 are given in lines 22, 23, 24, and 25, respectively.

The solution for this problem is  $S_E = S_F = 1$ , which implies that setting up supermarkets at E and F cater to all the six towns in the neighbourhood at the minimum distance. Towns A, B, D, and E are served by the supermarket in town E. Towns C and F are served by the supermarket in town F.

### 3. Considering Supermarket Capital Costs and Customer Travelling Costs

Building a supermarket requires a certain capital cost, which varies from town to town. Let the capital cost  $C_i$  be 700, 300, 450, 260, 900, and 150 GMD for building supermarkets in the towns A, B, C, D, E, and F, respectively. Further, customers also incur costs in travelling from the towns to the super markets; let the cost be 80 GMD per person per mile. In case, we want to locate supermarkets in such a way that we wish to minimize the total cost incurred in capital costs of building the supermarkets as well as the cost of travelling between the towns and the supermarkets,<sup>3</sup> the formulation will be slightly different from that described in Section 2.

Here, the objective function is the sum of capital costs  $\sum_{i=A,B,\dots}^F C_i S_i$  and the sum of travelling costs given by  $80 \sum_{i=A,B,\dots}^F \sum_{j=A,B,\dots}^F P_{ij} k_{ij} Y_{ij}$ , where  $S_i$  and  $Y_{ij}$  are the same as defined in Section 2.

The following three constraints pertaining to Section 2 will also be required here:

$$\sum_{i=A,B,\dots}^F S_i \leq 2 \quad (\text{Constraint 1})$$

$$\sum_{j=A,B,\dots}^F Y_{ij} = 1, \text{ for } i = A, B, \dots, F \quad (\text{Constraint 2})$$

$$Y_{ij} \leq S_j \text{ for } i, j = A, B, \dots, F \text{ (Constraint 3)}$$

The LINGO program is given in Table 3.5. There is only one primitive set TOWNS declared in the SETS section (line 2 of Table 3.5). The attributes of the TOWNS set are P (denoting population), C (denoting capital cost  $C_j$ ) and S (denoting decision binary variable  $S_j$ ). A derived set S1 is formed using the primitive sets, TOWNS and TOWNS (line 3 of Table 3.5). The attributes of S1 are K (denoting distance  $k_{ij}$ ) and Y (denoting decision variable  $Y_{ij}$ ).

The variable NOS in the DATA section (line 7 of Table 3.5) denotes the maximum number of supermarkets that the supermarket chain wishes to set up. The values of attribute P (given in line 9 of Table 3.5) are obtained from the first row of Table 3.1. The variable TRCOST in the

**Table 3.5**

Line	LINGO Program
1	SETS:
2	TOWNS:P,C,S;
3	S1(TOWNS,TOWNS):K,Y;
4	ENDSETS
5	
6	DATA:
7	NOS=2;
8	TOWNS=A B C D E F;
9	P=3 8 2 12 7 10;
10	C=700 300 450 260 900 150;
11	TRCOST=80;
12	K=
13	0    50    26    500    15    32
14	50    0    500    34    28    500
15	26    500    0    29    40    17
16	500    34    29    0    18    24
17	15    28    40    18    0    43
18	32    500    17    24    43    0
19	;
20	ENDDATA
21	
22	@FOR(TOWNS(I):@BIN(S(I)));
23	Min=@SUM(TOWNS(I):C(I)*S(I))+TRCOST*@SUM(TOWNS(I):@SUM(TOWNS(J):P(I)*Y(I,J))*K(I,J)));
24	@SUM(TOWNS(I):S(I))<=NOS;
25	@FOR(TOWNS(I):@SUM(TOWNS(J):Y(I,J))=1);
26	@FOR(S1(I,J):Y(I,J)<=S(J));

DATA section (line 11 of Table 3.5) denotes the costs in travelling, per person per mile. The values of attribute K (given in lines 13–18 of Table 3.5) are obtained from Table 3.1.

The decision binary variable  $S_i$  is declared in line 22 of the program given in Table 3.5. The objective function is given in line 23. Constraints 1, 2, and 3 are given in lines 24, 25, and 26, respectively.

The solution for this problem is  $S_E = S_F = 1$ , which implies that setting up supermarkets at E and F cater to all the six towns in the neighbourhood at the minimum capital and travelling cost. Towns A, B, D, and E are served by the supermarket located in town E. Towns C, and F are served by the supermarket located in town F.

#### 4. Capacity Limits of Supermarkets

The problem discussed in Section 4 could have a capacity limitation of supermarkets at different towns.<sup>4</sup> For example, supermarkets in towns could have a capacity constraint  $t_i$  in terms of number of persons that can be handled. Let the capacity constraint be 23, 25, 26, 14, 28, and 12 thousands for supermarkets located in towns A, B, C, D, E, and F, respectively. This capacity constraint can be handled by an additional constraint:

$$\sum_{i=A,B,\dots}^F PY_{ij} \leq t_j S_j, \text{ for } j = A, B, \dots, F \text{ (Constraint 4)}$$

The LINGO program is given in Table 3.6. This program is similar to that in Table 3.5 except for a few changes. Here the TOWNS set has an additional attribute T (which denotes capacity constraint  $t_i$  in thousands of persons), the values of which are given in line 11. Constraint 4 is given in line 28.

The solution for this problem is  $S_B = S_C = 1$ , which implies that setting up supermarkets at B and C cater to all the six towns in the neighbourhood at the minimum cost. Towns A, C, D and F are served by the supermarket located in town C. Towns B and E are served by the supermarket located in town B.

Table 3.6

Line	LINGO Program
1	SETS:
2	TOWNS:P,C,S,T;
3	S1(TOWNS,TOWNS):K,Y;
4	ENDSETS
5	
6	DATA:
7	NOS=2;
8	TOWNS=A B C D E F;
9	P=3 8 2 12 7 10;
10	C=700 300 450 260 900 150;
11	T=23 25 26 14 28 12;
12	TRCOST=80;
13	K=
14	0     50     26     500   15     32
15	50     0     500   34     28     500
16	26     500   0     29     40     17
17	500   34     29     0     18     24
18	15     28     40     18     0     43
19	32     500   17     24     43     0
20	;
21	ENDDATA
22	
23	@FOR(TOWNS(I):@BIN(S(I)));
24	Min=@SUM(TOWNS(I):C(I)*S(I))+@SUM(TOWNS(I):@SUM(TOWNS(J):P(I)*Y(I,J)*K(I,J)));
25	@SUM(TOWNS(I):S(I))<=NOS;
26	@FOR(TOWNS(I):@SUM(TOWNS(J):Y(I,J))=1);
27	@FOR(S1(I,J):Y(I,J)<=S(J));
28	@FOR(TOWNS(J):@SUM(TOWNS(I):P(I)*Y(I,J))<=T(J)*S(J));



# CHAPTER 4

## Cable Layout

Let us consider 10 neighbouring towns A, B, C, D, ..., J. The roads connecting the 10 towns are shown in Figure 4.1.

The distances (in miles) between pairs of neighbouring towns are given in Table 4.1. Distances are not given for towns that do not have a

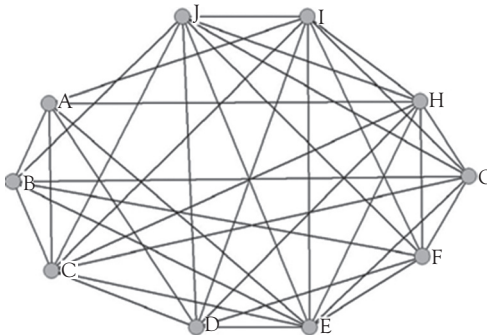


Figure 4.1

Table 4.1

	A	B	C	D	E	F	G	H	I	J
A	–	20	30	10	40	–	–	15	25	–
B	20	–	35	–	15	20	25	–	–	10
C	30	35	–	12	18	–	32	22	28	16
D	10	–	12	–	17	20	–	24	32	28
E	40	15	18	17	–	13	19	12	21	34
F	–	20	–	20	13	–	15	30	25	22
G	–	25	32	–	19	15	–	18	12	17
H	15	–	22	24	12	30	18	–	22	27
I	25	–	28	32	21	25	12	22	–	32
J	–	10	16	28	34	22	17	27	32	–

direct road connecting them; for example, towns A and F do not have a road connecting them.

A utilities company wishes to lay cables along the roads connecting the towns such that all towns are connected with each other using the minimum length of cable.

We use the following heuristic<sup>1</sup> for finding the roads along which the cables are to be laid such that all towns are connected using the minimum length of cable:

Step 1: Form three sets **P**, **Q**, and **R**. The towns A, B, C, D, ... J belong to either set **P** or set **Q**. Thus, if set **P** contains towns A and C (represented as  $\mathbf{P} = \{A, C\}$ ), the set **Q** contains the remaining towns (represented as  $\mathbf{Q} = \{B, D, E, F, G, H, I, J\}$ ). Set **R** will contain the pairs of neighbouring towns that will be connected by cables. For example, if towns A and H are connected by cables then set **R** will contain the pair (A, H). To start with, **P** and **R** are empty sets. Go to Step 2.

Step 2: Let us start with any town that is inserted in set **P**. Set **Q** thus contains the remaining towns. Go to Step 3.

Step 3: Choose the town  $i$  in set **Q** such that the length of a road from any town  $j$  in set **P** to town  $i$  is the minimum of all the distances from each town in set **P** to each town in set **Q**. Include town  $i$  in set **P** and delete that town from set **Q**. Include pair  $(i, j)$  in set **R**. Go to Step 4.

Step 4: If set **Q** is empty, stop. Else go to Step 3.

The heuristic is applied to the problem of finding the roads along which the cables are to be laid, such that all the ten towns are connected using the minimum length of cable as follows:

Step 1: $P = \{\}$ , $Q = \{A, B, C, D, E, F, G, H, I, J\}$ , $R = \{\}$
Step 2: Choose A to start with. Hence $P = \{A\}$ , $Q = \{B, C, D, E, F, G, H, I, J\}$
Step 3: $i = D$ and $j = A$ . Hence $P = \{A, D\}$ , $Q = \{B, C, E, F, G, H, I, J\}$ , $R = \{(D, A)\}$ (shown in Figure 4.2)

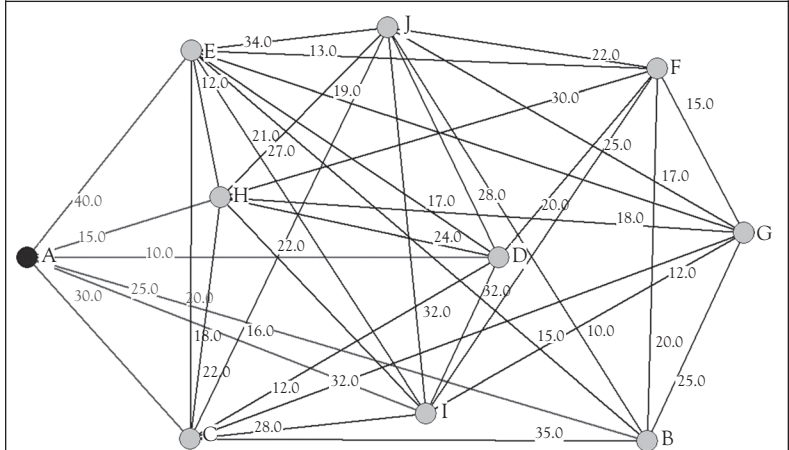


Figure 4.2

Step 3: $i = C$ and $j = D$ . Hence $P = \{A, D, C\}$ , $Q = \{B, E, F, G, H, I, J\}$ , $R = \{(D, A), (C, D)\}$ (shown in Figure 4.3)
--

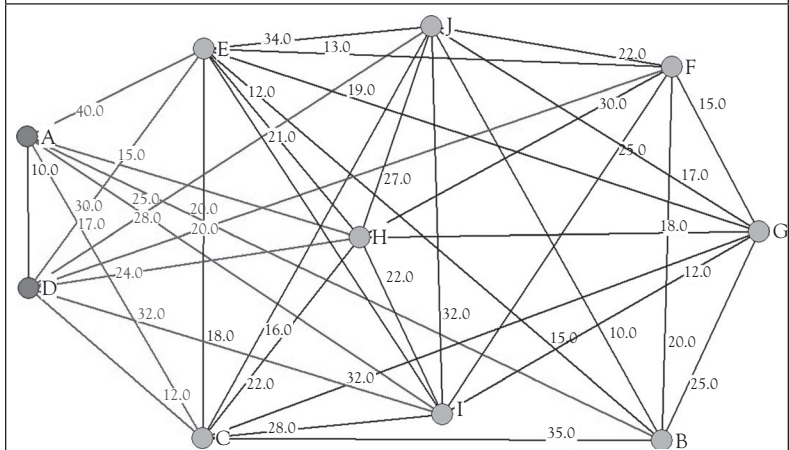


Figure 4.3

Step 3:  $i = H$  and  $j = A$ . Hence  $P = \{A, D, C, H\}$ ,  $Q = \{B, E, F, G, I, J\}$ ,  $R = \{(D, A), (C, D), (H, A)\}$  (shown in Figure 4.4)

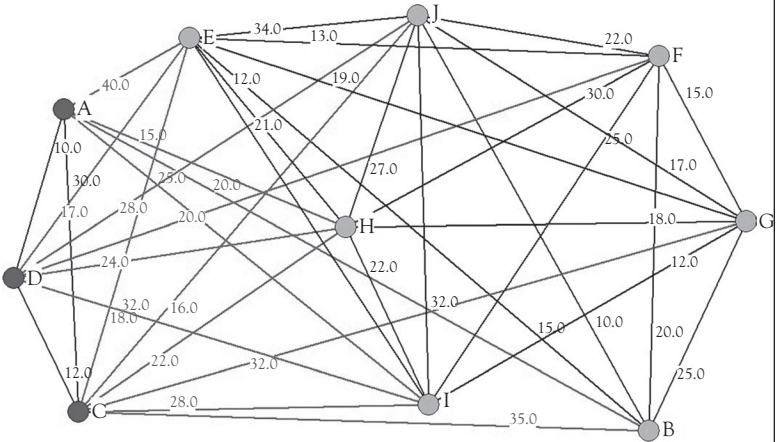


Figure 4.4

Step 3:  $i = E$  and  $j = H$ . Hence  $P = \{A, D, C, H, E\}$ ,  $Q = \{B, F, G, I, J\}$ ,  $R = \{(D, A), (C, D), (H, A), (E, H)\}$  (shown in Figure 4.5)

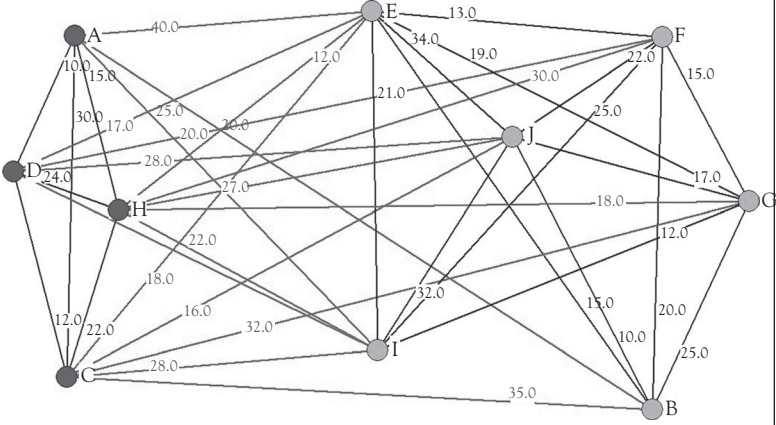
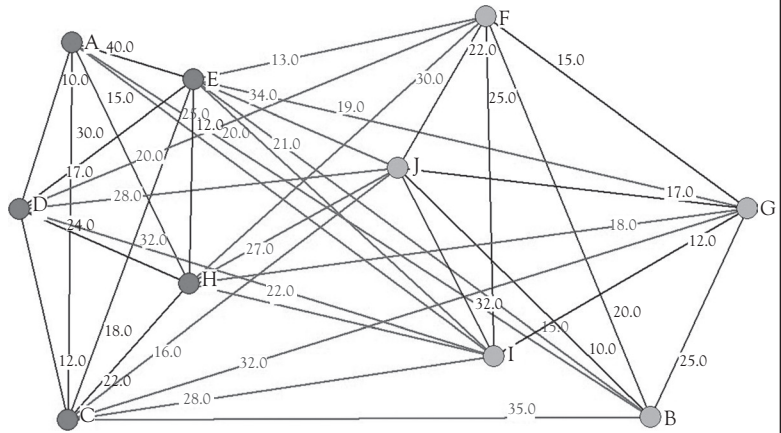


Figure 4.5

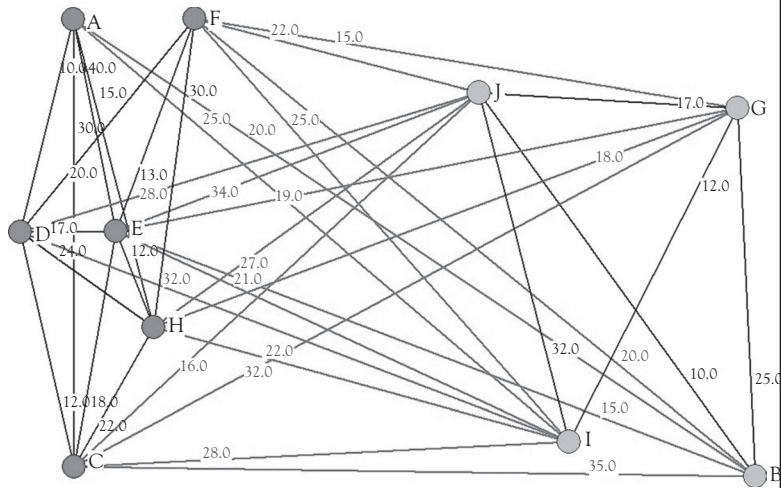
Step 3:  $i = F$  and  $j = E$ . Hence  $P = \{A, D, C, H, E, F\}$ ,  $Q = \{B, G, I, J\}$ ,  $R = \{(D, A), (C, D), (H, A), (E, H), (F, E)\}$  (shown in Figure 4.6)



**Figure 4.6**

Step 3:  $i = G$  and  $j = F$ . Hence  $P = \{A, D, C, H, E, F, G\}$ ,  $Q = \{B, I, J\}$ ,  $R = \{(D, A), (C, D), (H, A), (E, H), (F, E), (G, F)\}$  (shown in Figure 4.7)

We could also have chosen  $i = B$  and  $j = E$ , because distance from B to E is the same as that from G to F.



**Figure 4.7**

Step 3:  $i = I$  and  $j = G$ . Hence  $P = \{A, D, C, H, E, F, G, I\}$ ,  $Q = \{B, J\}$ ,  $R = \{(D, A), (C, D), (H, A), (E, H), (F, E), (G, F), (I, G)\}$  (shown in Figure 4.8)

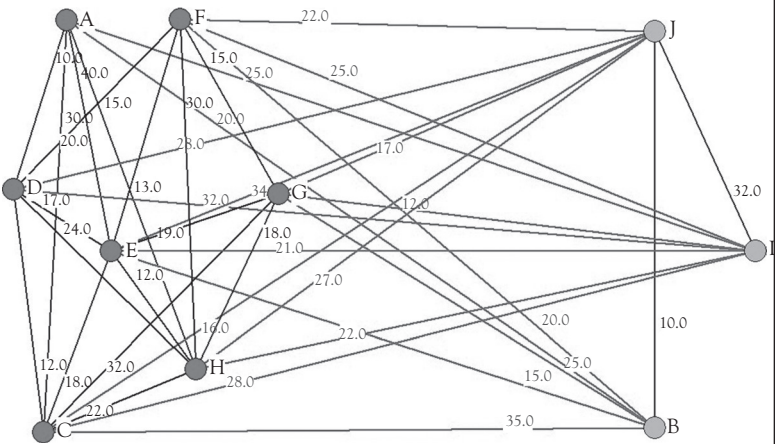


Figure 4.8

Step 3:  $i = B$  and  $j = E$ . Hence  $P = \{A, D, C, H, E, F, G, I, B\}$ ,  $Q = \{J\}$ ,  $R = \{(D, A), (C, D), (H, A), (E, H), (F, E), (G, F), (I, G), (B, E)\}$  (shown in Figure 4.9)

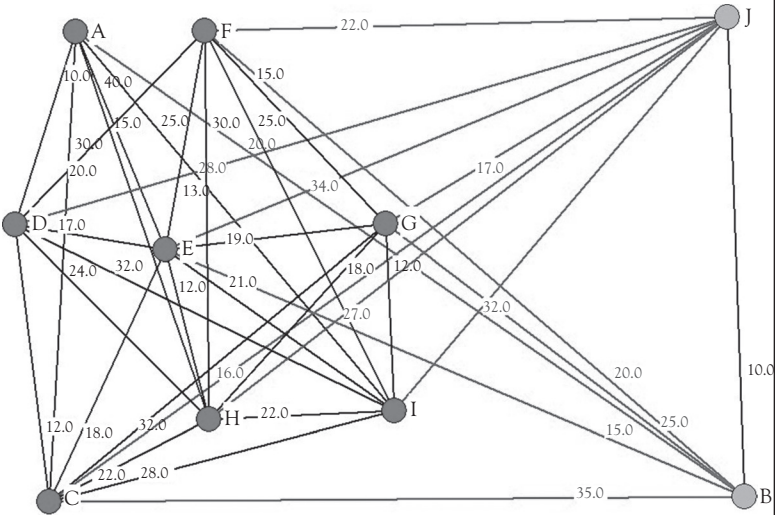
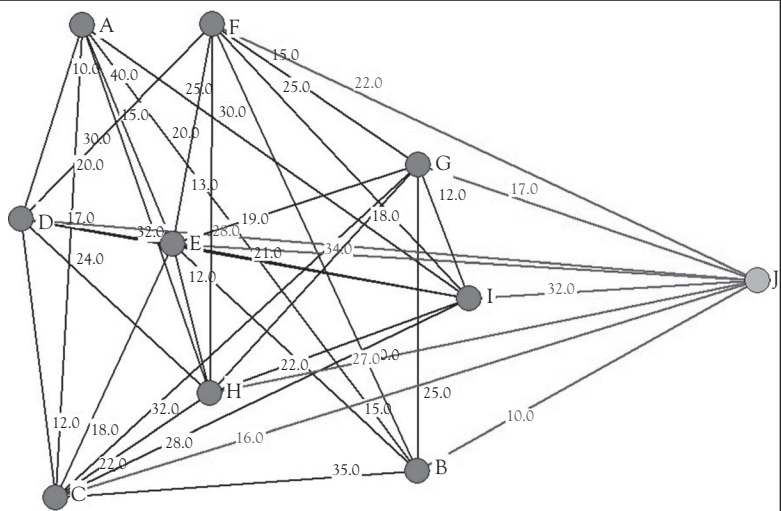


Figure 4.9

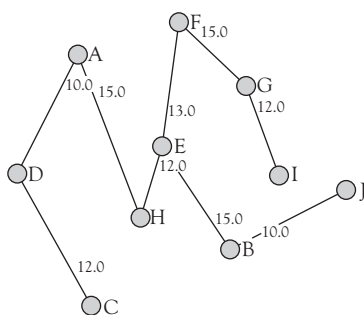
Step 3:  $i = J$  and  $j = B$ . Hence  $\mathbf{P} = \{A, D, C, H, E, F, G, I, B, J\}$ ,  $\mathbf{Q} = \{\}$ ,  $\mathbf{R} = \{(D, A), (C, D), (H, A), (E, H), (F, E), (G, F), (I, G), (B, E), (J, B)\}$  (shown in Figure 4.10)



**Figure 4.10**

Step 4: Stop.

The cable layout is given by the set  $\mathbf{R} = \{(D, A), (C, D), (H, A), (E, H), (F, E), (G, F), (I, G), (B, E), (J, B)\}$ , which is shown in Figure 4.11. The total length of the cable used is 114.



**Figure 4.11**



## CHAPTER 5

# Planning Check-in Counters

We frequently encounter the frustration of waiting in queues that never seem to move. Let us explore the mathematics underlying queue behaviour in this chapter using an example.

Let us take the case of a fast food drive-in sales counter. If a customer arrives and finds no one in the queue, the customer gives the order at the sales counter, makes the payment, and drives to the delivery bay to await delivery. If the customer arrives and finds a queue ahead, the customer has to wait. How long does a customer wait, and how long would the queue be on an average?

To answer these questions, we have to understand the arrival process of customers and the service process at the sales counter. For most situations, customer arrivals occur randomly and independently, which can be modelled by a Poisson probability distribution. The Poisson probability function  $P(x) = \frac{\lambda^x e^{-\lambda}}{x!}$  gives the probability of  $x$  arrivals in a time period, where  $\lambda$  is the average number of customer arrivals over a time period,  $e = 2.71828$  and  $x! = x(x-1)(x-2)\dots(2)(1)$ . Thus, if the average number of customer arrivals over an hour is 90 (which works out to  $\lambda = 1.5$  per minute), we can compute the probabilities of 1 to 5 customer arrivals per minute using the Poisson probability function  $P(x) = \frac{\lambda^x e^{-\lambda}}{x!}$  as given in Table 5.1.

**Table 5.1**

Customer arrivals per minute	Probability
$x = 1$	0.33
$x = 2$	0.25
$x = 3$	0.13
$x = 4$	0.05
$x = 5$	0.01

**Table 5.2**

Service time	Probability
$t \leq 2$ minutes	0.98
$t \leq 1$ minute	0.86
$t \leq 0.5$ minute	0.63
$t \leq 0.2$ minute	0.33
$t \leq 0.1$ minute	0.18

The time for handling each order by the sales counter can be modelled by an exponential probability distribution. The exponential probability distribution  $P(t) = 1 - e^{-\mu t}$  gives the probability that the time for handling will be less than  $t$ , where  $\mu$  is the average number of customers that can be served during a particular time period. Thus, if the sales counter can service an average number of 120 customers per hour (or  $\mu = 2$  per minute), we compute the probabilities of various service times as follows, using the exponential probability distribution  $P(t) = 1 - e^{-\mu t}$  as given in Table 5.2.

If the service rate  $\mu$  exceeds the customer arrival rate  $\lambda$ , the following relationships can be obtained:<sup>1</sup>

$$(a) \text{ Average number of customers in queue} = \frac{\lambda^2}{\mu(\mu - \lambda)}$$

$$(b) \text{ Average waiting and servicing time} = \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu}$$

If  $\lambda = 1.5$  per minute and  $\mu = 2$  per minute, (a) the average number of customers in queue works out to 2.25 and (b) average waiting and servicing time works out to 2 minutes. Similarly, we can find the service performance (in terms of queue length and average waiting and servicing time) for different combinations of  $\lambda$  and  $\mu$ . Thus, the service performance can be improved by changing the service rate  $\mu$  depending on the expected customer arrival rate  $\lambda$ .

# CHAPTER 6

## Scheduling of a Production Line

Production systems require jobs to be processed on different machines. The order of processing of a particular job through different machines is given by the requirements for that job. Each machine can process only one job at a time. The time of processing varies from job to job. In this chapter, we discuss the problem of determination of an optimal schedule for all jobs across the machines so that the processing of all the jobs is completed in the minimum possible time.<sup>1</sup>

Let us take the case of processing three jobs J1, J2, and J3 on four machines M1, M2, M3, and M4. It is required that job J1 be processed first in M2, followed by processing in M3, followed by processing in M4 and finally processed in M1. Jobs J2 is required to be processed in machines M2, M1, and M4 in that order and J3 is required to be processed in machines M1, M4, M2, and M3 in that order. The processing time of the jobs on different machines is given in Table 6.1, wherein it is seen that, whereas jobs J1 and J2 require processing on all four machines, job J2 requires processing on machines M1, M2, and M4 only.

A feasible sequence of scheduling of these three jobs is shown in Table 6.2. Because both jobs J1 and J2 start with processing on M2, we have to choose one of them to start processing on M2. Because we choose J1 to start processing first on M2 at time 0, J2 can start processing on M2

*Table 6.1*

	M1	M2	M3	M4
J1	10	5	8	3
J2	3	14	–	9
J3	5	8	9	12

Table 6.2

	M1		M2		M3		M4	
	Start	End	Start	End	Start	End	Start	End
J1	22	32	0	5	5	13	17	20
J2	19	22	5	19	–	–	22	31
J3	0	5	19	27	27	36	5	17

only when J1's processing is over at time 5. J1 starts processing on M3 next at time 5 and finishes processing at time 13. In the mean time, J3 has processed on machines M1 and started processing on machine M4 from time 5 onwards. Thus though J1 can start processing on M4 at 13, it has to wait till time 17 for job J3's processing to be over.

In this manner, it will be observed that we obtain the completion times of jobs J1, J2, and J3 as 32, 31, and 36, respectively. Hence all the jobs are over at time 36. Let us verify whether the completion time of 36 is the minimum possible time using a linear program.

Let us define a decision variable  $x_{ij}$  as the time of starting the processing of job  $j$  on machine  $i$ . Processing of a job on machine  $i$  cannot start unless it has been processed on the earlier machine scheduled. Job J1 cannot be processed on machine M3 unless it has been processed on machine M2. Hence, the time of starting processing of job J1 on machine M3 has to be more than the sum of time of starting the processing of job J1 on machine M2 and the processing time of job J1 on M2. Thus, we have the following eight constraints pertaining to the three jobs J1, J2, and J3:

$$x_{31} \geq x_{21} + 5 \text{ (Constraint 1)}$$

$$x_{41} \geq x_{31} + 8 \text{ (Constraint 2)}$$

$$x_{11} \geq x_{41} + 3 \text{ (Constraint 3)}$$

$$x_{12} \geq x_{22} + 14 \text{ (Constraint 4)}$$

$$x_{42} \geq x_{12} + 3 \text{ (Constraint 5)}$$

$$x_{43} \geq x_{13} + 5 \text{ (Constraint 6)}$$

$$x_{23} \geq x_{43} + 12 \text{ (Constraint 7)}$$

$$x_{33} \geq x_{23} + 8 \text{ (Constraint 8)}$$

We define  $C_{max}$  as the maximum time for completion of all the three jobs. The objective is to minimize  $C_{max}$ .  $C_{max}$  is related to the completion time of processing of all the jobs on the last machine and is modelled by the following three constraints:

$$C_{max} \geq x_{11} + 10 \text{ (Constraint 9)}$$

$$C_{max} \geq x_{42} + 9 \text{ (Constraint 10)}$$

$$C_{max} \geq x_{33} + 9 \text{ (Constraint 11)}$$

All jobs J1, J2, J3 require processing on machine M1. The time of starting processing of job J1 on M1 could either be before or after the processing of job J2 on M1. This can be modelled by the following three constraints:

$$-x_{11} \leq -x_{12} - 3 + My_1 \text{ (Constraint 12)}$$

$$-x_{12} \leq -x_{11} - 10 + My_2 \text{ (Constraint 13)}$$

$$y_1 + y_2 = 1 \text{ (Constraint 14)}$$

where  $y_1$  and  $y_2$  are binary variables and  $M$  is a very large positive number.

Similar constraints for processing of jobs J1 or J3 on machine M1 are:

$$-x_{11} \leq -x_{13} - 5 + My_3 \text{ (Constraint 15)}$$

$$-x_{13} \leq -x_{11} - 10 + My_4 \text{ (Constraint 16)}$$

$$y_3 + y_4 = 1 \text{ (Constraint 17)}$$

Similar constraints for processing of jobs J2 or J3 on machine M1 are:

$$-x_{12} \leq -x_{13} - 5 + My_5 \text{ (Constraint 18)}$$

$$-x_{13} \leq -x_{12} - 3 + My_6 \text{ (Constraint 19)}$$

$$y_5 + y_6 = 1 \text{ (Constraint 20)}$$

We formulate similar constraints for other pairs of jobs on machines M2, M3, and M4. Thus constraints 21–29 pertain to processing of jobs on machine M2, Constraints 30–32 pertain to processing of jobs on machine M3 (which is used by jobs J1 and J3 only for processing) and Constraints 33–42 pertain to processing of jobs on machine M4.

The LINGO program is given in Table 6.3. Constraints 1–8 are given in lines 2–9. Constraints 9–11 are given in lines 10–12. Constraints 12–20 pertaining to M1 are given in lines 13–24 along with declarations for binary variables. Constraints 21–29 pertaining to M2 are given in lines 25–36 along with declarations for binary variables. Constraints 30–32 pertaining to M3 are given in lines 37–40 along with declarations for binary variables. Constraints 33–41 pertaining to M4 are given in lines 41–52 along with declarations for binary variables.

**Table 6.3**

Line	LINGO Program	Line	LINGO Program
1	min=Cmax;	27	y7+y8=1;
2	x31>=x21+5;	28	@BIN(y7);@BIN(y8);
3	x41>=x31+8;	29	-x21<=-x23-8+500*y9;
4	x11>=x41+3;	30	-x23<=-x21-5+500*y10;
5	x12>=x22+14;	31	y9+y10=1;
6	x42>=x12+3;	32	@BIN(y9);@BIN(y10);
7	x43>=x13+5;	33	-x22<=-x23-8+500*y11;
8	x23>=x43+12;	34	-x23<=-x22-14+500*y12;
9	x33>=x23+8;	35	y11+y12=1;
10	Cmax>=x11+10;	36	@BIN(y11);@BIN(y12);
11	Cmax>=x42+9;	37	-x31<=-x33-9+500*y13;
12	Cmax>=x33+9;	38	-x33<=-x31-8+500*y14;
13	-x11<=-x12-3+500*y1;	39	y13+y14=1;
14	-x12<=-x11-10+500*y2;	40	@BIN(y13);@BIN(y14);
15	y1+y2=1;	41	-x41<=-x42-9+500*y15;
16	@BIN(y1);@BIN(y2);	42	-x42<=-x41-3+500*y16;
17	-x11<=-x13-5+500*y3;	43	y15+y16=1;
18	-x13<=-x11-10+500*y4;	44	@BIN(y15);@BIN(y16);
19	y3+y4=1;	45	-x41<=-x43-12+500*y17;
20	@BIN(y3);@BIN(y4);	46	-x43<=-x41-3+500*y18;
21	-x12<=-x13-5+500*y5;	47	y17+y18=1;
22	-x13<=-x12-3+500*y6;	48	@BIN(y17);@BIN(y18);
23	y5+y6=1;	49	-x42<=-x43-12+500*y19;
24	@BIN(y5);@BIN(y6);	50	-x43<=-x42-9+500*y20;
25	-x21<=-x22-14+500*y7;	51	y19+y20=1;
26	-x22<=-x21-5+500*y8;	52	@BIN(y19);@BIN(y20);

*Table 6.4*

	M1		M2		M3		M4	
	Start	End	Start	End	Start	End	Start	End
<b>J1</b>	22	32	0	5	5	13	19	22
<b>J2</b>	19	22	5	19	–	–	22	31
<b>J3</b>	0	5	19	27	27	36	5	17

The solution given by the LINGO program is given in Table 6.4. The solution gives the minimum completion time of 36 for all three jobs.



## CHAPTER 7

# Shift Staff Planning

Let us take the case of a call centre where the staff requirements vary by the hour. The minimum requirements of staff in each hour are given in Table 7.1.<sup>1</sup>

Staff can start work at the beginning of any hour. The staff works for 8 hours with a rest break of 2 hours after 4 hours of duty. How many staff should be scheduled to join at the beginning of each hour such that the requirements given in Table 7.1 are met with the minimum number of staff?

Let us take the case of staff starting work at 1 am. They are able to take care of the requirements from 1 am to 5 am and 7 am to 11 am. We can find out similar duty hours for all staff joining at the beginning of each hour from 00 am to 11 pm as listed in Table 7.2.

Let us define integer decision variables  $x_i$  as the number of staff joining duty at the beginning of hour  $i$  (with the hours in 24 hour clock format), where  $i = 1, 2, \dots, 24$ .

Now, if we take the time period 00 am to 1 am, Table 7.2 indicates that the staff available during this period is the sum of staff who have joined duty at 3 pm, 4 pm, 5 pm, 6 pm, 9 pm, 10 pm, 11 pm, and 00 am.

**Table 7.1**

Period	Minimum requirement
00 am to 3 am	20
3 am to 6 am	16
6 am to 9 am	25
9 am to 12 noon	40
12 noon to 4 pm	60
4 pm to 8 pm	30
8 pm to 00 am	25

**Table 7.2**

Time of joining duty	Pre-break hours of duty	Post-break hours of duty
00 am	00 am to 4 am	6 am to 10 am
1 am	1 am to 5 am	7 am to 11 am
2 am	2 am to 6 am	8 am to 12 noon
3 am	3 am to 7 am	9 am to 1 pm
4 am	4 am to 8 am	10 am to 2 pm
5 am	5 am to 9 am	11 am to 3 pm
6 am	6 am to 10 am	12 noon to 4 pm
7 am	7 am to 11 am	1 pm to 5 pm
8 am	8 am to 12 noon	2 pm to 6 pm
9 am	9 am to 1 pm	3 pm to 7 pm
10 am	10 am to 2 pm	4 pm to 8 pm
11 am	11 am to 3 pm	5 pm to 9 pm
12 noon	12 noon to 4 pm	6 pm to 10 pm
1 pm	1 pm to 5 pm	7 pm to 11 pm
2 pm	2 pm to 6 pm	8 pm to 00 am
3 pm	3 pm to 7 pm	9 pm to 1 am
4 pm	4 pm to 8 pm	10 pm to 2 am
5 pm	5 pm to 9 pm	11 pm to 3 am
6 pm	6 pm to 10 pm	00 am to 4 am
7 pm	7 pm to 11 pm	1 am to 5 am
8 pm	8 pm to 00 am	2 am to 6 am
9 pm	9 pm to 1 am	3 am to 7 am
10 pm	10 pm to 2 am	4 am to 8 am
11 pm	11 pm to 3 am	5 am to 9 am

Thus, we have the following constraint that the sum of these staff should exceed the minimum requirement of 20 persons:

$$x_{15} + x_{16} + x_{17} + x_{18} + x_{21} + x_{22} + x_{23} + x_{24} \geq 20$$

Similar equations can be obtained for the remaining 23 time periods as listed below.

$$x_{16} + x_{17} + x_{18} + x_{19} + x_{22} + x_{23} + x_{24} + x_1 \geq 20 \text{ (1 am to 2 am)}$$

$$x_{17} + x_{18} + x_{19} + x_{20} + x_{23} + x_{24} + x_1 + x_2 \geq 20 \quad (2 \text{ am to } 3 \text{ am})$$

$$x_{18} + x_{19} + x_{20} + x_{21} + x_{24} + x_1 + x_2 + x_3 \geq 16 \quad (3 \text{ am to } 4 \text{ am})$$

$$x_{19} + x_{20} + x_{21} + x_{22} + x_1 + x_2 + x_3 + x_4 \geq 16 \quad (4 \text{ am to } 5 \text{ am})$$

$$x_{20} + x_{21} + x_{22} + x_{23} + x_2 + x_3 + x_4 + x_5 \geq 16 \quad (5 \text{ am to } 6 \text{ am})$$

$$x_{21} + x_{22} + x_{23} + x_{24} + x_3 + x_4 + x_5 + x_6 \geq 25 \quad (6 \text{ am to } 7 \text{ am})$$

$$x_{22} + x_{23} + x_{24} + x_1 + x_4 + x_5 + x_6 + x_7 \geq 25 \quad (7 \text{ am to } 8 \text{ am})$$

$$x_{23} + x_{24} + x_1 + x_2 + x_5 + x_6 + x_7 + x_8 \geq 25 \quad (8 \text{ am to } 9 \text{ am})$$

$$x_{24} + x_1 + x_2 + x_3 + x_6 + x_7 + x_8 + x_9 \geq 40 \quad (9 \text{ am to } 10 \text{ am})$$

$$x_1 + x_2 + x_3 + x_4 + x_7 + x_8 + x_9 + x_{10} \geq 40 \quad (10 \text{ am to } 11 \text{ am})$$

$$x_2 + x_3 + x_4 + x_5 + x_8 + x_9 + x_{10} + x_{11} \geq 40 \quad (11 \text{ am to } 12 \text{ noon})$$

$$x_3 + x_4 + x_5 + x_6 + x_9 + x_{10} + x_{11} + x_{12} \geq 60 \quad (12 \text{ noon to } 1 \text{ pm})$$

$$x_4 + x_5 + x_6 + x_7 + x_{10} + x_{11} + x_{12} + x_{13} \geq 60 \quad (1 \text{ pm to } 2 \text{ pm})$$

$$x_5 + x_6 + x_7 + x_8 + x_{11} + x_{12} + x_{13} + x_{14} \geq 60 \quad (2 \text{ pm to } 3 \text{ pm})$$

$$x_6 + x_7 + x_8 + x_9 + x_{12} + x_{13} + x_{14} + x_{15} \geq 60 \quad (3 \text{ pm to } 4 \text{ pm})$$

$$x_7 + x_8 + x_9 + x_{10} + x_{13} + x_{14} + x_{15} + x_{16} \geq 30 \quad (4 \text{ pm to } 5 \text{ pm})$$

$$x_8 + x_9 + x_{10} + x_{11} + x_{14} + x_{15} + x_{16} + x_{17} \geq 30 \quad (5 \text{ pm to } 6 \text{ pm})$$

$$x_9 + x_{10} + x_{11} + x_{12} + x_{15} + x_{16} + x_{17} + x_{18} \geq 30 \quad (6 \text{ pm to } 7 \text{ pm})$$

$$x_{10} + x_{11} + x_{12} + x_{13} + x_{16} + x_{17} + x_{18} + x_{19} \geq 30 \quad (7 \text{ pm to } 8 \text{ pm})$$

$$x_{11} + x_{12} + x_{13} + x_{14} + x_{17} + x_{18} + x_{19} + x_{20} \geq 25 \quad (8 \text{ pm to } 9 \text{ pm})$$

$$x_{12} + x_{13} + x_{14} + x_{15} + x_{18} + x_{19} + x_{20} + x_{21} \geq 25 \quad (9 \text{ pm to } 10 \text{ pm})$$

$$x_{13} + x_{14} + x_{15} + x_{16} + x_{19} + x_{20} + x_{21} + x_{22} \geq 25 \quad (10 \text{ pm to } 11 \text{ pm})$$

$$x_{14} + x_{15} + x_{16} + x_{17} + x_{20} + x_{21} + x_{22} + x_{23} \geq 25 \quad (11 \text{ pm to } 00 \text{ am})$$

Because the objective is to minimize the total staff, we have the following objective function:

$$\text{Min} \sum_{i=1}^{24} x_i$$

**Table 7.3**

Line	LINGO Program
1	@GIN(x1);@GIN(x2);@GIN(x3);@GIN(x4);@GIN(x5);
2	@GIN(x6);@GIN(x7);@GIN(x8);@GIN(x9);@GIN(x10);
3	@GIN(x11);@GIN(x12);@GIN(x13);@GIN(x14);@GIN(x15);
4	@GIN(x16);@GIN(x17);@GIN(x18);@GIN(x19);@GIN(x20);
5	@GIN(x21);@GIN(x22);@GIN(x23);@GIN(x24);
6	min=x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+ x11+x12+x13+x14+x15+x16+x17+x18+x19+x20+ x21+x22+x23+x24;
7	x15+x16+x17+x18+x21+x22+x23+x24>=20;
8	x16+x17+x18+x19+x22+x23+x24+x1>=20;
9	x17+x18+x19+x20+x23+x24+x1+x2>=20;
10	x18+x19+x20+x21+x24+x1+x2+x3>=16;
11	x19+x20+x21+x22+x1+x2+x3+x4>=16;
12	x20+x21+x22+x23+x2+x3+x4+x5>=16;
13	x21+x22+x23+x24+x3+x4+x5+x6>=25;
14	x22+x23+x24+x1+x4+x5+x6+x7>=25;
15	x23+x24+x1+x2+x5+x6+x7+x8>=25;
16	x24+x1+x2+x3+x6+x7+x8+x9>=40;
17	x1+x2+x3+x4+x7+x8+x9+x10>=40;
18	x2+x3+x4+x5+x8+x9+x10+x11>=40;
19	x3+x4+x5+x6+x9+x10+x11+x12>=60;
20	x4+x5+x6+x7+x10+x11+x12+x13>=60;
21	x5+x6+x7+x8+x11+x12+x13+x14>=60;
22	x6+x7+x8+x9+x12+x13+x14+x15>=60;
23	x7+x8+x9+x10+x13+x14+x15+x16>=30;
24	x8+x9+x10+x11+x14+x15+x16+x17>=30;
25	x9+x10+x11+x12+x15+x16+x17+x18>=30;
26	x10+x11+x12+x13+x16+x17+x18+x19>=30;
27	x11+x12+x13+x14+x17+x18+x19+x20>=25;
28	x12+x13+x14+x15+x18+x19+x20+x21>=25;
29	x13+x14+x15+x16+x19+x20+x21+x22>=25;
30	x14+x15+x16+x17+x20+x21+x22+x23>=25;

**Table 7.4**

00 am	1 am	2 am	3 am	4 am	5 am	6 am	7 am	8 am	9 am	10 am	11 am
–	12	–	4	7	12	23	–	8	4	6	–
12 noon	1 pm	2 pm	3 pm	4 pm	5 pm	6 pm	7 pm	8 pm	9 pm	10 pm	11 pm
4	8	5	8	4	8	–	–	–	–	–	–

The LINGO program is given in Table 7.3. The integer decision variables  $x_i$  are declared in lines 1–5 and the constraints pertaining to the 24 time periods are given in lines 7–30. The solution obtained in terms of staff joining at each hour, is given in Table 7.4.

**Table 7.5**

Time period	Staff available	Staff requirement
00–1 am	20	20
1–2 am	24	20
2–3 am	20	20
3–4 am	16	16
4–5 am	23	16
5–6 am	23	16
6–7 am	46	25
7–8 am	54	25
8–9 am	55	25
9–10 am	51	40
10–11 am	41	40
11 am–12 noon	41	40
12 noon–1 pm	60	60
1–2 pm	60	60
2–3 pm	60	60
3–4 pm	60	60
4–5 pm	43	30
5–6 pm	43	30
6–7 pm	34	30
7–8 pm	30	30
8–9 pm	25	25
9–10 pm	25	25
10–11 pm	25	25
11 pm–00 am	25	25

The staff, thus, available (computed using the optimal solution obtained in Table 7.4 and the constraints pertaining to the 24 time periods) and the requirement in each time period is given in Table 7.5. It will be observed that the staff thus available in each time period sometimes exceeds the requirements of that particular time period.



## CHAPTER 8

# Production Planning

Here, we take the case of a company that has forecasted the demand (in a number of units) for a processed food product over the next six months (April–September) as given in Table 8.1.

The company has an inventory of 50 units that was produced in March. The company has strict policies regarding keeping of processed food products in inventory according to which inventory can only be kept for a maximum of 2 months. Thus, anything produced in April has to be sold latest by June end. Further, due to storage capacity restrictions the maximum inventory can be only 100 units. The company has estimated that keeping inventory costs 15 dollars per unit per month of storage.

The company can produce a maximum of 500 units with regular labour at a cost of 50 dollars per unit. Labour can be paid overtime to produce an additional 200 units at a cost of 80 dollars per unit.

How much should the company produce each month to meet the demand at minimum cost? The company wishes to end the product line at the end of September, and thus, would not like to have any inventory remaining after the end of September.

To solve this problem, we define a decision variable  $x_{ijl}$  as the number of units produced in month  $i$  for sale in month  $j$  by labour type  $l$ . Months  $i$  and  $j$  range from 1–6 pertaining to the months April, May, June, ..., September. Labour types  $l$  take the value 1 or 2 depending on whether regular or overtime labour is used.

Further, in regard to the decision variable  $x_{ijl}$  it will be obvious that  $i \leq j \leq i + 2$ , because products manufactured in month  $i$  can either be sold in month  $i$ , or month  $i + 1$  or month  $i + 2$ . The cost of production  $c_{ijl}$  for

*Table 8.1*

Month	April	May	June	July	August	September
Demand	200	400	700	800	300	200

the product manufactured in month  $i$ , which is sold in month  $i$ , is 50 or 80 dollars per unit depending on whether it is produced by regular or overtime labour. The cost of production and inventory  $c_{ijl}$  for the product manufactured in month  $i$ , which is sold in month  $i + 1$ , is 65 or 95 dollars per unit (considering inventory cost for a month) depending on whether it is produced by regular or overtime labour. The cost of production and inventory  $c_{ijl}$  for the product manufactured in month  $i$ , which is sold in month  $i + 2$ , is 80 or 110 dollars per unit (considering inventory cost for two months) depending on whether it is produced by regular or overtime labour. Hence, the objective function is given as follows:

$$\text{Minimize } \sum_{i=1}^4 \sum_{j=i}^{i+2} \sum_{l=1}^2 x_{ijl} c_{ijl}$$

Because the monthly production is restricted to a maximum of 500 and 200 units for regular and overtime labour, respectively, the following first set of constraints are added to the model:

$$\sum_{j=1}^{i+2} x_{ijl} \leq t_l, \text{ for } i = 1, 2, 3, 4 \text{ and } l = 1, 2$$

where  $t_l$  is the maximum production of 500 and 200 units for regular and overtime labour, respectively.

The inventory can be a maximum of 100 units. Thus, at the end of April, the number of units produced in April for sale in May and June has to be less than 100 units. At the end of May, the sum of (a) number of units produced in April for sale in June and (b) number of units produced in May for sale in June and July has to be less than 100 units. This will require a second set of constraints.

The demand in each month has to be met by the production in that month and the production in the earlier two months. This will require a third set of constraints.

The LINGO program is given in Table 8.2. The first set of constraints is given in lines 2–13. The second set of constraints is given in lines 14–18. The third set of constraints is given in lines 19–24.

Table 8.2

Line	LINGO Program
1	min=50*(x111+x221+x331+x441+x551+x661)+ 80*(x112+x222+x332+x442+x552+x662)+ 65*(x121+x231+x341+x451+x561)+ 95*(x122+x232+x342+x452+x562)+ 80*(x131+x241+x351+x461)+ 110*(x132+x242+x352+x462);
2	x111+x121+x131<=500;
3	x112+x122+x132<=200;
4	x221+x231+x241<=500;
5	x222+x232+x242<=200;
6	x331+x341+x351<=500;
7	x332+x342+x352<=200;
8	x441+x451+x461<=500;
9	x442+x452+x462<=200;
10	x551+x561<=500;
11	x552+x562<=200;
12	x661<=500;
13	x662<=200;
14	x121+x131+x122+x132<=100;!end of April;
15	x131+x132+x231+x241+x232+x242<=100;!end of May;
16	x241+x242+x341+x351+x342+x352<=100;!end of June;
17	x351+x352+x451+x461+x452+x462<=100;!end of July;
18	x461+x462+x561+x562<=100;!end of August;
19	x111+x112>=200;
20	x121+x122+x221+x222>=400;
21	x131+x132+x231+x232+x331+x332>=700;
22	x241+x242+x341+x342+x441+x442>=800;
23	x351+x352+x451+x452+x551+x552>=300;
24	x461+x462+x561+x562+x661+x662>=200;

The solution is given in Table 8.3. It is observed that whatever is produced in the months April, July, August, and September is consumed in that month only. Overtime production has been resorted only in the months of June and July.

Table 8.3

Month	April	May	June	July	August	September
Demand	200	400	700	800	300	200
April Reg	200					
April Ov						
May Reg		400	100			
May Ov						
June Reg			400	100		
June Ov			200			
July Reg				500		
July Ov				200		
Aug Reg					300	
Aug Ov						
Sept Reg						200
Sept Ov						

## CHAPTER 9

# Blending of Dog Diet

Let us take the problem of preparing a dog's daily diet by mixing appropriate quantities of raw meat, brown rice, and vegetables to meet the diet requirements at the minimum cost.

A 80 pound dog requires a minimum of 2000 calories per day along with a diet in which the protein content should range from 1–40% of diet by weight, carbohydrate content should range from 5–55% of diet by weight, and vegetable content should range from 1.5–25% of diet content by weight.

Raw meat contains 15% protein by weight and 60 calories per ounce. Brown rice contains 25% carbohydrates by weight and 30 calories per ounce. Vegetables contain 10 calories per ounce. The cost of raw meat, brown rice, and vegetables is 12, 80, and 150 per ounce, respectively.

Let us choose decision variables  $x_1$ ,  $x_2$ , and  $x_3$  as the weight (in ounces) of raw meat, brown rice, and vegetables used for preparing the daily diet.

The following constraints model the minimum and maximum requirements of protein content:

$$0.15x_1 \geq 0.01(x_1 + x_2 + x_3)$$

$$0.15x_1 \leq 0.4(x_1 + x_2 + x_3)$$

The following constraints model the minimum and maximum requirements of carbohydrate content:

$$0.25x_2 \geq 0.05(x_1 + x_2 + x_3)$$

$$0.25x_2 \leq 0.55(x_1 + x_2 + x_3)$$

**Table 9.1**

LINGO Program
min=12*x1+80*x2+150*x3; 0.15*x1>=0.01*(x1+x2+x3); 0.15*x1<=0.40*(x1+x2+x3); 0.25*x2>=0.05*(x1+x2+x3); 0.25*x2<=0.55*(x1+x2+x3); x3>=0.015*(x1+x2+x3); x3<=0.25*(x1+x2+x3); 60*x1+30*x2+10*x3>=2000;

The following constraints model the minimum and maximum requirements of vegetable content:

$$x_3 \geq 0.015(x_1 + x_2 + x_3)$$

$$x_3 \leq 0.25(x_1 + x_2 + x_3)$$

The following constraints model the minimum calories requirement:

$$60x_1 + 30x_2 + 10x_3 \geq 2000$$

Because the objective is to minimize the total cost of the diet, the objective function is given as:

$$\text{Min } 12x_1 + 80x_2 + 150x_3$$

The LINGO program is given in Table 9.1. The optimal solution comprises 29.5 ounces of raw meat, 7.5 ounces of brown rice, and 0.6 ounces of vegetables.

# CHAPTER 10

## Paper Roll Trimming

Steel, paper, and textiles are produced in long rolls of different widths. These rolls need to be sliced to appropriate lengths to satisfy requirements of fabrication of machinery or paper cartons or apparel. For example, an 8 feet width roll could be sliced to obtain rolls of 5 feet and 3 feet width as shown in Figure 10.1. Slicing has to be done in a manner such that the total cost of material used is minimized.<sup>1</sup>

Let us take the example of three paper rolls: one of 1000 feet length and of 8 feet width, another of 500 feet length and of 5 feet width, and a third of 600 feet length and 3 feet width. We need 500 feet length of 1 foot width paper, 600 feet length of two feet width paper, 800 feet length of 4 feet width paper, and 400 feet length of 6 feet width paper. Let us assume that the cost is 100 per foot of 8 feet width rolls, 70 per foot of 5 feet width rolls, and 40 per foot of 3 feet width rolls.

Now, 1 foot length of 1 foot width paper can be obtained by any one of the following ways: (a) slicing a  $1/8$  foot length of 8 feet width roll in eight equal parts, (b) slicing a  $1/5$  foot length of the 5 feet width roll in five equal parts, (c) slicing a  $1/3$  foot length of the 3 feet width roll in three equal parts, (d) slicing a  $1/2$  foot length of the 8 feet width roll in three equal parts of  $1/2$  foot length of 6 feet width,  $1/2$  foot length of 1 foot width, and  $1/2$  foot length of 1 feet width. We can, thus, find different ways of obtaining the requirements of 1 foot width paper, 2 feet width

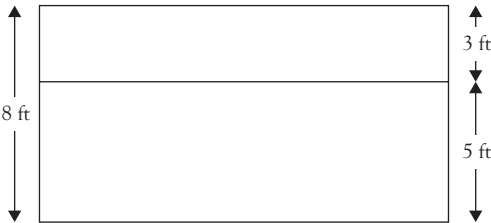


Figure 10.1

Table 10.1

Roll used	Feet length obtained →	1 foot width	2 feet width	4 feet width	6 feet width
	feet length used ↓				
8 feet width	$x_{81}$	$8x_{81}$	—	—	—
	$x_{82}$	$6x_{82}$	$x_{82}$	—	—
	$x_{83}$	$4x_{83}$	—	$x_{83}$	—
	$x_{84}$	$2x_{84}$	—	—	$x_{84}$
	$x_{85}$	—	$4x_{85}$	—	—
	$x_{86}$	—	$2x_{86}$	$x_{86}$	—
	$x_{87}$	—	$x_{87}$	—	$x_{87}$
	$x_{88}$	—	—	$2x_{88}$	—
5 feet width	$x_{51}$	$5x_{51}$	—	—	—
	$x_{52}$	$3x_{52}$	$x_{52}$	—	—
	$x_{53}$	$x_{53}$	—	$x_{53}$	—
3 feet width	$x_{31}$	$3x_{31}$	—	—	—
	$x_{32}$	$x_{32}$	$x_{32}$		

paper, 4 feet width paper, and 6 feet width paper, all of which is listed in Table 10.1.

The first row of Table 10.1 implies that  $x_{81}$  feet of 8 feet width roll yields  $8x_{81}$  feet of 1 foot width paper by slicing the roll in eight equal parts. Similarly, the last row of the table implies that  $x_{32}$  feet of 3 feet width roll yields  $x_{32}$  feet of 1 foot width paper and  $x_{32}$  feet of 2 feet width paper by slicing the roll in two parts of 1 foot and 2 feet widths, respectively.

Thus the requirement of 1 foot width paper has to be met by the summation of the 1 foot length widths obtained from 8 feet width, 5 feet width, and 3 feet width rolls. Thus we obtain the constraint for the requirement of 1 foot width paper as follows:

$$8x_{81} + 6x_{82} + 4x_{83} + 2x_{84} + 5x_{51} + 3x_{52} + x_{53} + 3x_{31} + x_{32} \geq 500$$

Similar constraints for the requirement of 2 feet width, 4 feet width, and 6 feet width paper are obtained as follows:

$$x_{82} + 4x_{85} + 2x_{86} + x_{87} + x_{52} + x_{32} \geq 600$$

$$x_{83} + x_{86} + 2x_{88} + x_{53} \geq 800$$

$$x_{84} + x_{87} \geq 400$$

It is given that only 1000 feet length of paper roll of 8 feet width is available. Thus the sum of  $x_{81}, x_{82}, \dots, x_{88}$  feet length has to be less than 1000 feet. Thus the constraint for availability of 8 feet width paper is as follows:

$$x_{81} + x_{82} + x_{83} + x_{84} + x_{85} + x_{86} + x_{87} + x_{88} \leq 1000$$

Similar constraints for availability of 5 feet and 3 feet width rolls are obtained as follows:

$$x_{51} + x_{52} + x_{53} \leq 500$$

$$x_{31} + x_{32} \leq 600$$

The objective is to minimize the cost of 8 feet, 5 feet, and 3 feet width rolls used. The objective function is:

$$\begin{aligned} \text{Min } & 100 * (x_{81} + x_{82} + x_{83} + x_{84} + x_{85} + x_{86} + x_{87} + x_{88}) \\ & + 70 * (x_{51} + x_{52} + x_{53}) + 40 * (x_{31} + x_{32}) \end{aligned}$$

The LINGO program is given in Table 10.2. The solution to the problem is:  $x_{82} = 83.33$ ,  $x_{86} = 58.33$ ,  $x_{87} = 400$ , and  $x_{88} = 370.833$ . This implies that (a) 83.33 feet length of 8 feet width roll is sliced in seven parts to yield 500 feet length of 1 feet width paper and 83.33 feet length of 2 feet width paper, (b) 58.33 foot length of 8 feet width roll is sliced in three parts to

**Table 10.2**

LINGO Program
min=100*(x81+x82+x83+x84+x85+x86+x87+x88)+70*(x51+x52+x53)+40*(x31+x32); 8*x81+6*x82+4*x83+2*x84+5*x51+3*x52+x53+3*x31+x32>=500; x82+4*x85+2*x86+x87+x52+x32>=600; x83+x86+2*x88+x53>=800; x84+x87>=400; x81+x82+x83+x84+x85+x86+x87+x88<=1000; x51+x52+x53<=500; x31+x32<=600;

yield 116.66 feet length of 2 feet width paper and 58.33 feet length of 4 feet width paper, (c) 400 feet length of 8 feet width roll is sliced in two parts to yield 400 feet length of 2 feet width paper and 400 feet length of 6 feet width paper, and (d) 370.833 feet length of 8 feet width roll is sliced in two parts to yield 741.666 feet length of 4 feet width paper. Thus we use 912.5 feet length of 8 feet width roll to obtain 500 feet length of 1 foot width paper, 600 feet length of 2 feet width paper, 800 feet length of 4 feet width paper, and 400 feet length of 6 feet width paper.

## SECTION 3

# Applications in Supply Chain Management



# CHAPTER 11

## Multicommodity Transport Planning

Let us take the case of a company producing five commodities CA, CB, CC, CD, and CE in four plants located in Vadodara, Visakhapatnam, Nagpur, and Kochi as shown in Figure 11.1.

Each plant has a certain maximum production capacity for each commodity, as given in Table 11.1. It is assumed that the cost of production of these commodities is the same at all the plants.



Figure 11.1

Table 11.1

	Vadodara	Visakhapatnam	Nagpur	Kochi
CA	200	400	800	400
CB	400	600	400	600
CC	800	200	600	800
CD	400	400	200	200
CE	600	800	400	400

Table 11.2

	Mumbai	Chennai	Bangalore	Hyderabad	Bhopal	Kanpur
CA	50	50	100	400	50	160
CB	100	200	150	300	100	100
CC	400	100	200	100	100	100
CD	50	100	200	50	200	50
CE	100	500	50	50	50	300

The customers for these five commodities are located at different parts of the country around the cities of Mumbai, Chennai, Bangalore, Hyderabad, Bhopal, and Kanpur shown in Figure 11.1. The demand of the customers for each of the commodities is known and given in Table 11.2.

The company is planning to set up at most three distribution centers to streamline the supply from the plants to the customers.<sup>1</sup> Commodities will be transported from the production plants to the distribution centers, from where they will be sent to the customers. Each customer will be served by a single distribution center. No commodity will be transported directly from any plant to any customer. The company has identified four candidate locations at Sholapur, Indore, Vijayawada, and Hubli (as shown in Figure 11.1) for the planned distribution centers. The distances (in kilometres) from the plants and customers to the candidate locations are given in Table 11.3.

The company wishes to select the distribution center locations such that the transportation cost from the plants to the distribution centers and the distribution centers to the customers and the cost of the distribution centers is minimized. In the process, the allocation of the customers to distribution centers will also be determined.

**Table 11.3**

		Solapur	Indore	Vijayawada	Hubli
Plants	Vadodara	791	347	1386	972
	Visakhapatnam	949	1287	349	1127
	Nagpur	587	510	709	884
	Kochi	1173	1973	1090	964
Customers	Mumbai	402	585	997	583
	Chennai	956	1455	458	763
	Bangalore	618	1418	660	409
	Hyderabad	310	829	272	511
	Bhopal	844	196	1061	1202
	Kanpur	1331	703	1452	1709

**Table 11.4**

	CA	CB	CC	CD	CE
Transportation cost per unit per kilometre	1	0.2	5	2	10

**Table 11.5**

	Solapur	Indore	Vijayawada	Hubli
Fixed cost	2000	5000	3000	1000
Variable cost	2	3	4	1
Minimum annual throughput	200	100	500	200
Maximum annual throughput	1000	1100	2000	2000

We assume that transportation cost is directly proportional to the distance traversed. The transportation cost per unit per kilometre for the five commodities CA, CB, CC, CD, and CE are given in Table 11.4.

Each distribution center location has different fixed costs and variable costs as given in Table 11.5. The variable cost of each location is directly proportional to the annual throughput. The minimum and maximum annual throughput of the distribution centers are also given in the Table 11.5.

Let us use the following notation for the model:

$C_p$ ,  $i = 1, 2, 3, 4, 5$  denotes the commodities CA, CB, CC, CD, CE

$P_j$ ,  $j = 1, 2, 3, 4$  denotes the plants located at Vadodara, Visakhapatnam, Nagpur, and Kochi, respectively

$W_k$ ,  $k = 1, 2, 3, 4$  denotes the candidate distribution center locations at Solapur, Indore, Vijaywada, and Hubli, respectively

$N_l$ ,  $l = 1, 2, 3, 4, 5, 6$  denotes the customer locations at Mumbai, Chennai, Bangalore, Hyderabad, Bhopal, and Kanpur, respectively

$S_{ij}$  denotes the maximum production capacity of commodity  $i$  at plant  $j$

$D_{il}$  denotes the demand for commodity  $i$  by customer  $l$

$\bar{V}_k, \underline{V}_k$  denotes the maximum and minimum throughput for candidate distribution center location  $k$

$f_k$  denotes the fixed cost of operation of candidate distribution center location  $k$

$v_k$  denotes the variable cost of operation of candidate distribution center location  $k$  for each unit of throughput

$m_i$  denotes the cost of transportation of commodity  $i$  per unit per kilometre

$d'_{jk}$  denotes the distance (in kilometres) from plant  $j$  to distribution center  $k$

$d''_{kl}$  denotes the distance (in kilometres) from distribution center  $k$  to customer  $l$

$cst_{ijkl} = m_i(d'_{jk} + d''_{kl})$  denotes the cost of transportation per unit of commodity  $i$  from plant  $j$  through distribution center  $k$  to customer  $l$ .

$x_{ijkl}$  is a decision variable denoting the amount of commodity  $i$  transported from plant  $j$  through distribution center  $k$  to customer  $l$

$y_{kl}$  is a binary integer decision variable, which is 1 if distribution center  $k$  serves customer  $l$  and 0 otherwise

$z_k$  is a binary integer decision variable, which is 1 if candidate distribution center  $k$  is chosen and 0 otherwise

The cost of transportation from the production plants to the customers

through the distribution centers is given by  $\sum_{i=1}^5 \sum_{j=1}^4 \sum_{k=1}^4 \sum_{l=1}^6 cst_{ijkl} x_{ijkl}$ . The

cost of the distribution centers is given by  $\sum_{k=1}^4 \left( f_k z_k + v_k \sum_{i=1}^5 \sum_{l=1}^6 D_{il} y_{kl} \right)$ .

Hence, the objective function is given by the following expression:

$$\text{Minimize } \sum_{i=1}^5 \sum_{j=1}^4 \sum_{k=1}^4 \sum_{l=1}^6 \text{cst}_{ijkl} x_{ijkl} + \sum_{k=1}^4 \left( f_k z_k + v_k \sum_{i=1}^5 \sum_{l=1}^6 D_{il} y_{kl} \right)$$

Because each plant  $j$  has a limited capacity of production of commodity  $i$ , the total amount of commodities supplied to all customers  $l$  through all distribution centers  $k$  has to be less than the maximum production capacity of that plant for that commodity. This is modelled by the following constraint:

$$\sum_{k=1}^4 \sum_{l=1}^6 x_{ijkl} \leq S_{ij}, \text{ for all } i, j \text{ (Constraint 1)}$$

The amount of a particular commodity  $i$  received at the distribution center  $k$  for a particular customer  $l$  from all the plants must equal the demand for that commodity by that customer. This is modelled by the following constraint:

$$\sum_{j=1}^4 x_{ijkl} = D_{il} y_{kl}, \text{ for all } i, k, l \text{ (Constraint 2)}$$

Each customer is served by only one distribution center. This is modelled by the following constraint:

$$\sum_{k=1}^4 y_{kl} = 1, \text{ for all } l \text{ (Constraint 3)}$$

The total amount of commodities passing through a distribution center for customers assigned to that distribution center must exceed the minimum annual throughput and must be lower than the maximum annual throughput for that distribution center. This is modelled by the following two constraints:

$$\sum_{i=1}^5 \sum_{l=1}^6 D_{il} y_{kl} \geq \underline{V}_k z_k, \text{ for all } k \text{ (Constraint 4)}$$

$$\sum_{i=1}^5 \sum_{l=1}^6 D_{il} y_{kl} \leq \overline{V}_k z_k, \text{ for all } k \text{ (Constraint 5)}$$

Table 11.6

Line	LINGO Program	Line	LINGO Program
1	SETS:	36	d1=
2	C:m;	37	791 347 1386 972
3	P;	38	949 1287 349 1127
4	W:VU,VL,f,v,z;	39	587 510 709 884
5	N;	40	1173 1973 1090 964
6	S1(C,P):S;	41	;
7	S2(C,N):D;	42	d2=
8	S3(P,W):d1;	43	402 956 618 310 844 1331
9	S4(W,N):d2,y;	44	585 1455 1418 829 196 703
10	S5(C,P,W,N):CST,x;	45	997 458 660 272 1061 1452
11	S6(C,W,N);	46	583 763 409 511 1202 1709
12	ENDSETS	47	;
13		48	ENDDATA
14	DATA:	49	
15	C=1 2 3 4 5;	50	@FOR(S4(K,L):@BIN(y(K,L)));
16	P=1 2 3 4;	51	@FOR(W(K):@BIN(z(K)));
17	W=1 2 3 4;	52	MIN=@SUM(S5(I,J,K,L):
18	N=1 2 3 4 5 6;		CST(I,J,K,L)*x(I,J,K,L))+
19	S=		@SUM(W(K):(f(K)*z(K)+v(K)*
20	200 400 800 400		@SUM(S2(I,L):D(I,L)*y(K,L)));
21	400 600 400 600	53	@FOR(S5(I,J,K,L):CST(I,J,K,L)
22	800 200 600 800		=M(I)*(d1(J,K)+d2(K,L)));
23	400 400 200 200	54	@FOR(S1(I,J):@SUM(S4(K,L):
24	600 800 400 400;		x(I,J,K,L))<=S(I,J));
25	D=	55	@FOR(S6(I,K,L):@SUM(P(J):
26	50 50 100 400 50 160		x(I,J,K,L))>=D(I,L)*y(K,L));
27	100 200 150 300 100 100	56	@FOR(N(L):@SUM(W(K):y(K,L))=1);
28	400 100 200 100 100 100		@FOR(W(K):@
29	50 100 200 50 200 50	57	SUM(S2(I,L):D(I,L)*y(K,L))
30	100 500 50 50 50 300;		>=VL(K)*z(K));
31	VU=1000 1100 2000 2000;	58	@FOR(W(K):@SUM(S2(I,L):D(I,L)*
32	VL=200 100 500 200;		y(K,L))<=VU(K)*z(K));
33	f=2000 5000 3000 1000;	59	@FOR(W(K):@SUM(N(L):y(K,L))
34	v=2 3 4 1;		<=5000*z(K));
35	m=1 0.2 5 2 10;	60	@SUM(W(K):z(K))<=3;

The integer binary variable  $z_k$  should be 1 if the candidate distribution center  $k$  serves at least one customer (or  $y_{kl}$  is 1 for at least one customer  $l$ ). This is modelled by the following constraint:

$$\sum_{l=1}^6 y_{kl} \leq Mz_k \quad (\text{Constraint 6})$$

where  $M$  is a very large number.

Further, because the management wishes to have at most three distribution centers, we have the following constraint:

$$\sum_{k=1}^4 z_k \leq 3 \quad (\text{Constraint 7})$$

The LINGO program is given in Table 11.6. There are four primitive sets C, P, W, and N declared in the SETS section (lines 2–5 of Table 11.6) corresponding to  $C_i$ ,  $P_j$ ,  $W_k$ , and  $N_l$ . The attribute of set C is m (corresponding to  $m_i$ ). The attributes of set W are VU (corresponding to  $\overline{V_k}$ ), VL (corresponding to  $\underline{V_k}$ ), f (corresponding to  $f_k$ ), v (corresponding to  $v_k$ ) and z (corresponding to  $z_k$ ). Six derived sets S1, S2, S3, S4, S5, and S6 are formed using the primitive sets (line 6–11). The attribute of S1 is S (denoting  $S_{ij}$ ). The attribute of S2 is D (denoting  $D_{il}$ ). The attribute of S3 is d1 (denoting  $d'_{jk}$ ). The attributes of S4 are d2 (denoting  $d''_{kl}$ ) and  $\gamma$  (denoting  $y_{kl}$ ). The attributes of S5 are CST (denoting  $cst_{ijkl}$ ) and  $x$  (denoting  $x_{ijkl}$ ).

The values of attribute S (given in lines 20–24 of Table 11.6) are obtained from Table 11.1. The values of attribute D (given in lines 26–30) are obtained from Table 11.2. The values of attributes VU and VL (given in lines 31 and 32) are obtained from the fourth and fifth rows of Table 11.5, respectively. The values of attribute f and v (given in lines 33 and 34) are obtained from the second and third rows of Table 11.5, respectively. The value of attribute m (given in line 35) is obtained from Table 11.4. The values of attribute d1 (given in lines 37–40) are obtained from the second to fifth rows of Table 11.3. The values of attribute d2 (given in lines 43–46) are obtained from last six rows of Table 11.3.

The decision binary variables  $y_{kl}$  and  $z_k$  are declared in lines 50 and 51. The objective function is given in line 52. The expression  $cst_{ijkl} = m_i(d'_{jk} + d''_{kl})$  is given in line 53. Constraints 1–7 are given in lines 54–58, respectively.

The solution indicates that distribution centers should be located at Indore, Vijayawada, and Hubli. The customers being served by each distribution center is given in Table 11.7.

**Table 11.7**

Distribution centers	Indore	Vijaywada	Hubli
Customers served	Kanpur	Chennai, Hyderabad, Mumbai	Bangalore, Bhopal

Table 11.8

Customers	CA	CB	CC	CD	CE
Mumbai	Nagpur	Nagpur	Nagpur Kochi	Nagpur	Nagpur
Chennai	Visakhapatnam	Visakhapatnam	Visakhapatnam	Visakhapatnam	Visakhapatnam
Bangalore	Nagpur	Nagpur	Nagpur	Vadodara Kochi	Nagpur
Hyderabad	Visakhapatnam Nagpur	Visakhapatnam	Visakhapatnam	Visakhapatnam	Visakhapatnam
Bhopal	Nagpur	Nagpur	Nagpur	Nagpur Kochi	Nagpur
Kanpur	Vadodara	Vadodara	Vadodara	Vadodara	Vadodara

The plants supplying the commodities to the customers are given in Table 11.8.

Because the LINGO program given in Table 11.6 might exceed the variable limits of the demo version, it is advised to modify the program given in Table 11.6 as follows: (a) change line 15 to “C = 1 2 3 4” (b) delete the last 4 figures “600 800 400 400” in line 24 (c) delete the last 6 figures “100 500 50 50 50 300” in line 30 and (d) delete the last figure “10” in line 35.

# CHAPTER 12

## Single Delivery Truck Routing

Let us consider a distribution center at A that has to send a truck daily with refills to retail centers located in nine neighbouring towns B, C, D, ..., J. The road map connecting the 10 towns is shown in Figure 12.1.

The distances (in miles) between pairs of neighbouring towns are given in Table 12.1. Distances are not given for towns that do not have a direct road between them; for example A and F do not have a road connecting them.

The distribution center wishes to route the truck from A to the nine towns and back to A such that the truck passes through each town only once and returns back to A such that the distance covered is the minimum. For example, a possible route is A–B–C–D–E–F–G–H–J–I–A with a total distance of 214 miles. Can we find a shorter route?

Let us use the following heuristic<sup>1</sup> for solving this problem:

Step I: Let us start with any possible route  $R = t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_1$  with a total distance  $d$  as the shortest route known. For example, if we start

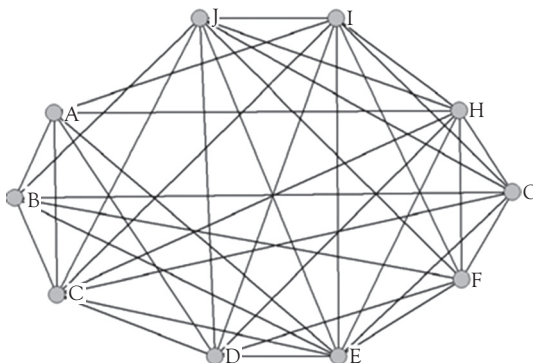


Figure 12.1

Table 12.1

	A	B	C	D	E	F	G	H	I	J
A	–	20	30	10	40	–	–	15	25	–
B	20	–	35	–	15	20	25	–	–	10
C	30	35	–	12	18	–	32	22	28	16
D	10	–	12	–	17	20	–	24	32	28
E	40	15	18	17	–	13	19	12	21	34
F	–	20	–	20	13	–	15	30	25	22
G	–	25	32	–	19	15	–	18	12	17
H	15	–	22	24	12	30	18	–	22	27
I	25	–	28	32	21	25	12	22	–	32
J	–	10	16	28	34	22	17	27	32	–

with route is A–B–C–D–E–F–G–H–J–I–A covering 214 miles,  $t_1 = A$ ,  $t_2 = B$ ,  $t_3 = C$ ,  $t_4 = D$ ,  $t_5 = E$ ,  $t_6 = F$ ,  $t_7 = G$ ,  $t_8 = H$ ,  $t_9 = J$ ,  $t_{10} = I$  and the shortest distance  $d = 214$ .

Step II: Set  $i = 1$

Step III: Set  $j = i + 2$

Step IV: Interchange the towns  $t_{i+1}$  and  $t_j$  in the route R if it is feasible; if not feasible, go to Step V. Calculate the total distance  $d'$  with the interchanged route. If the distance  $d'$  is less than  $d$ , set the new shortest route R with towns  $t_{i+1}$  and  $t_j$  interchanged with shortest distance  $d = d'$  and go to Step II. If the distance  $d'$  is more than  $d$ , go to Step V.

Step V: Increment  $j$  by 1. If  $j \leq 10$ , go to Step VI. If  $j > 10$ , increment  $i$  by 1. If  $i \leq 8$ , go to Step III. If  $i > 8$ , stop.

Let us follow the heuristic to find the shortest route.

1. Step I:  $R = t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_1$ , where  $t_1 = A$ ,  $t_2 = B$ ,  $t_3 = C$ ,  $t_4 = D$ ,  $t_5 = E$ ,  $t_6 = F$ ,  $t_7 = G$ ,  $t_8 = H$ ,  $t_9 = J$ ,  $t_{10} = I$ , the route being A–B–C–D–E–F–G–H–J–I–A and distance  $d = 214$ .
2. Step II: Set  $i = 1$
3. Step III: Set  $j = i + 2 = 3$
4. Step IV: Interchange the towns  $t_{i+1} = t_2(B)$  and  $t_j = t_3(C)$  in the route R if it is feasible. The route then becomes A–C–B–D–E–F–G–H–

J–I–A. Because towns B and D do not have a road connection, it is not feasible.

5. Step V:  $j = 3 + 1 = 4$
6. Step IV: Interchange the towns  $t_{i+1} = t_2$  (B) and  $t_j = t_4$  (D) in the route R if it is feasible. The route then becomes A–D–C–B–E–F–G–H–J–I–A, which is feasible. The distance of route  $d'$  is 202. Because  $d' = 202 < d = 214$ , the shortest route R is changed to  $R = t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_1$ , where  $t_1 = A$ ,  $t_2 = D$ ,  $t_3 = C$ ,  $t_4 = B$ ,  $t_5 = E$ ,  $t_6 = F$ ,  $t_7 = G$ ,  $t_8 = H$ ,  $t_9 = J$ ,  $t_{10} = I$  and the shortest distance  $d = 202$ .
7. Step II: Set  $i = 1$
8. Step III: Set  $j = i + 2 = 3$
9. Step IV: Interchange the towns  $t_{i+1} = t_2$  (D) and  $t_j = t_3$  (C) in the route R if it is feasible. The route then becomes A–C–D–B–E–F–G–H–J–I–A, which is not feasible because there is no road connecting towns D and B.
10. Step V:  $j = 3 + 1 = 4$
11. Step IV: Interchange the towns  $t_{i+1} = t_2$  (D) and  $t_j = t_4$  (B) in the route R if it is feasible. The route then becomes A–B–C–D–E–F–G–H–J–I–A, which is feasible. The distance of route  $d'$  is 214, which is greater than  $d = 202$ .
12. Step V:  $j = 4 + 1 = 5$
13. Step IV: Interchange the towns  $t_{i+1} = t_2$  (D) and  $t_j = t_5$  (E) in the route R if it is feasible. The route then becomes A–E–C–B–D–F–G–H–J–I–A, which is not feasible, because there is no road connecting towns B and D.
14. Step V:  $j = 5 + 1 = 6$
15. Step IV: Interchange the towns  $t_{i+1} = t_2$  (D) and  $t_j = t_6$  (F) in the route R if it is feasible. The route then becomes A–F–C–B–E–D–G–H–J–I–A, which is not feasible, because there is no road connecting towns A and F.
16. Step V:  $j = 6 + 1 = 7$
17. Step IV: Interchange the towns  $t_{i+1} = t_2$  (D) and  $t_j = t_7$  (G) in the route R if it is feasible. The route then becomes A–G–C–B–E–F–D–H–J–I–A, which is not feasible, because there is no road connecting towns A and G.
18. Step V:  $j = 7 + 1 = 8$

19. Step IV: Interchange the towns  $t_{i+1} = t_2(D)$  and  $t_j = t_8(H)$  in the route R, if it is feasible. The route then becomes A–H–C–B–E–F–G–D–J–I–A, which is not feasible, because there is no road connecting towns G and D.
20. Step V:  $j = 8 + 1 = 9$
21. Step IV: Interchange the towns  $t_{i+1} = t_2(D)$  and  $t_j = t_9(J)$  in the route R if it is feasible. The route then becomes A–J–C–B–E–F–G–H–D–I–A, which is not feasible, because there is no road connecting towns A and J.
22. Step V:  $j = 9 + 1 = 10$
23. Step IV: Interchange the towns  $t_{i+1} = t_2(D)$  and  $t_j = t_{10}(I)$  in the route R if it is feasible. The route then becomes A–I–C–B–E–F–G–H–J–D–A, which is feasible. The distance of route  $d'$  is 214, which is greater than  $d = 202$ .
24. Step V:  $j = 10 + 1 = 11$ . Because  $j > 10$ , we increment  $i$  by 1.  $i = 1 + 1 = 2$
25. Step III: Set  $j = i + 2 = 4$
26. Step IV: Interchange the towns  $t_{i+1} = t_3(C)$  and  $t_j = t_4(B)$  in the route R if it is feasible. The route then becomes A–D–B–C–E–F–G–H–J–I–A, which is not feasible because there is no road connecting towns D and B.
27. Step V:  $j = 4 + 1 = 5$
28. Step IV: Interchange the towns  $t_{i+1} = t_3(C)$  and  $t_j = t_5(E)$  in the route R if it is feasible. The route then becomes A–D–E–B–C–F–G–H–J–I–A, which is not feasible because there is no road connecting towns C and F.
29. Step V:  $j = 5 + 1 = 6$
30. Step IV: Interchange the towns  $t_{i+1} = t_3(C)$  and  $t_j = t_6(F)$  in the route R, if it is feasible. The route then becomes A–D–F–B–E–C–G–H–J–I–A, which is feasible. The distance of route  $d'$  is 217, which is greater than  $d = 202$ .
31. Step V:  $j = 6 + 1 = 7$
32. Step IV: Interchange the towns  $t_{i+1} = t_3(C)$  and  $t_j = t_7(G)$  in the route R if it is feasible. The route then becomes A–D–G–B–E–F–C–H–J–I–A, which is not feasible because there is no road connecting D to G.

Table 12.2

Shortest route R	Shortest distance $d$	$i$	$j$	Route	$d'$
A-D-J-B-E-F-G-H-C-I-A	184	1	10	A-I-J-B-E-F-G-H-C-D-A	172
A-I-J-B-E-F-G-H-C-D-A	172	1	8	A-H-J-B-E-F-G-I-C-D-A	157
A-H-J-B-E-F-G-I-C-D-A	157	2	2	A-H-E-B-J-F-G-I-C-D-A	151

33. Step V:  $j = 7 + 1 = 8$

34. Step IV: Interchange the towns  $t_{i+1} = t_3(\text{C})$  and  $t_j = t_8(\text{H})$  in the route R if it is feasible. The route then becomes A-D-H-B-E-F-G-C-J-I-A, which is not feasible because there is no road connecting H to B.

35. Step V:  $j = 8 + 1 = 9$

36. Step IV: Interchange the towns  $t_{i+1} = t_3(\text{C})$  and  $t_j = t_9(\text{J})$  in the route R if it is feasible. The route then becomes A-D-J-B-E-F-G-H-C-I-A, which is feasible. The distance of route  $d'$  is 184. Because  $d' = 184 < d = 202$ , R is changed to  $R = t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_1$  where  $t_1 = \text{A}$ ,  $t_2 = \text{D}$ ,  $t_3 = \text{J}$ ,  $t_4 = \text{B}$ ,  $t_5 = \text{E}$ ,  $t_6 = \text{F}$ ,  $t_7 = \text{G}$ ,  $t_8 = \text{H}$ ,  $t_9 = \text{C}$ ,  $t_{10} = \text{I}$  and  $d = 184$ .

The remaining steps are summarized in Table 12.2. The steps where the route is not feasible or there is no improvement in the shortest path distance are not shown.

The shortest route obtained is thus A-H-E-B-J-F-G-I-C-D-A with a total distance of 151 miles.



# CHAPTER 13

## Multiple Delivery Trucks Routing

Let us consider a distribution center at A that has to send a truck daily with refills to nine neighbouring towns B, C, D, ..., J. The distances (in miles) between pairs of neighbouring towns are given in Table 13.1.

The daily demand (in cartons) for each of the nine towns B, C, D, ..., J is given in Table 13.2.

The trucks used for distribution can carry only 60 cartons. How many trucks will be required for distribution? How should these trucks be routed from the distribution center A to the towns and back to A such that the total miles covered by these trucks is minimum?

**Table 13.1**

	A	B	C	D	E	F	G	H	I	J
A	–	20	30	10	40	20	32	15	25	14
B	20	–	35	24	15	20	25	19	11	10
C	30	35	–	12	18	21	32	22	28	16
D	10	24	12	–	17	20	29	24	32	28
E	40	15	18	17	–	13	19	12	21	34
F	20	20	21	20	13	–	15	30	25	22
G	32	25	32	29	19	15	–	18	12	17
H	15	19	22	24	12	30	18	–	22	27
I	25	11	28	32	21	25	12	22	–	32
J	14	10	16	28	34	22	17	27	32	–

**Table 13.2**

	B	C	D	E	F	G	H	I	J
Demand	20	30	40	10	25	15	35	10	20

We use the following heuristic<sup>1</sup> for finding the solution:

- Step I: Find the shortest possible route **R** starting and ending at the distribution center and passing through all the  $n$  towns only once. If the sequence of towns visited is obtained as A–B–C–D–E–F–G–H–I–J–A, the first town visited is B, the second town visited is C, and so on.
- Step II: Start with truck  $i = 1$  and  $j = 1$ . The full capacity of truck  $i$  is 60 cartons.
- Step III: Select the  $j$ th town in the route **R**. If the demand of town  $j$  is less than or equal to the full or remaining capacity of truck  $i$ , allocate the town  $j$  to truck  $i$ , reduce the full or remaining capacity of truck  $i$  by the demand of town  $j$  and go to Step IV. If the demand of town  $j$  is more than the full or remaining capacity of truck  $i$ , go to Step V.
- Step IV: Increment  $j$  by 1. If  $j$  is more than  $n$ , stop. Otherwise go to Step III.
- Step V: Increment  $i$  by 1 and go to Step III.

We apply the heuristic to the problem as follows:

1. Step I: We obtain the shortest possible route **R** using the method described in Chapter 12. The steps followed are summarized in Table 13.3. The steps where there are no improvement in the shortest path distance are not shown.  
Hence, the shortest possible route **R** is A–B–I–H–G–F–E–D–C–J–A, with the first town being B, second town being I and so on.
2. Step II:  $i = 1$  and  $j = 1$ . The full capacity of truck 1 is 60.
3. Step III: The first town in **R** is B with demand 20. The full capacity of truck 1 is 60. Hence town B is allocated to truck 1. The remaining capacity of truck 1 is revised to 40.
4. Step IV:  $j = 2$

Table 13.3

Shortest route <b>R</b>	Shortest distance $d$	$i$	$j$	Route	$d'$
A–B–C–D–E–F–G–H–I–J–A	198	1	4	A–D–C–B–E–F–G–H–I–J–A	186
A–D–C–B–E–F–G–H–I–J–A	186	2	9	A–D–I–B–E–F–G–H–C–J–A	166
A–D–I–B–E–F–G–H–C–J–A	166	1	8	A–H–I–B–E–F–G–D–C–J–A	162
A–H–I–B–E–F–G–D–C–J–A	162	4	7	A–H–I–B–G–F–E–D–C–J–A	160
A–H–I–B–G–F–E–D–C–J–A	160	1	4	A–B–I–H–G–F–E–D–C–J–A	158

5. Step III: The second town in **R** is I with demand 10. The remaining capacity of truck 1 is 40. Hence, town I is allocated to truck 1. The remaining capacity of truck 1 is revised to 30.
6. Step IV:  $j = 3$
7. Step III: The third town in **R** is H with demand 35. The remaining capacity of truck 1 is 30. Hence town H cannot be allocated to truck 1.
8. Step V:  $i = 2$ . The full capacity of truck 2 is 60.
9. Step III: The third town in **R** is H with demand 35. The remaining capacity of truck 2 is 60. Hence, town H is allocated to truck 2. The remaining capacity of truck 2 is revised to 25.
10. Step IV:  $j = 4$
11. Step III: The fourth town in **R** is G with demand 15. The remaining capacity of truck 2 is 25. Hence, town G is allocated to truck 2. The remaining capacity of truck 2 is revised to 10.
12. Step IV:  $j = 5$
13. Step III: The fifth town in **R** is F with demand 25. The remaining capacity of truck 2 is 10. Hence, town F cannot be allocated to truck 2.
14. Step V:  $i = 3$ . The full capacity of truck 3 is 60.
15. Step III: The fifth town in **R** is F with demand 25. The full capacity of truck 3 is 60. Hence, town F is allocated to truck 3. The remaining capacity of truck 3 is revised to 35.
16. Step IV:  $j = 6$
17. Step III: The sixth town in **R** is E with demand 10. The remaining capacity of truck 3 is 35. Hence, town E is allocated to truck 3. The remaining capacity of truck 3 is revised to 25.
18. Step IV:  $j = 7$
19. Step III: The seventh town in **R** is D with demand 40. The remaining capacity of truck 3 is 25. Hence, town D cannot be allocated to truck 3.
20. Step V:  $i = 4$ . The full capacity of truck 4 is 60.
21. Step III: The seventh town in **R** is D with demand 40. The full capacity of truck 4 is 60. Hence, town D is allocated to truck 4. The remaining capacity of truck 4 is revised to 20.
22. Step IV:  $j = 8$
23. Step III: The eighth town in **R** is C with demand 30. The remaining capacity of truck 4 is 20. Hence, town C cannot be allocated to truck 4.

Table 13.4

Truck	Cartons loaded	Truck capacity utilization(%)	Route	Miles covered
1	30	50	A-B-I-A	56
2	50	83	A-H-G-A	65
3	35	58	A-F-E-A	73
4	40	67	A-D-A	20
5	50	83	A-C-J-A	60

- 24. Step V:  $i = 5$ . The full capacity of truck 5 is 60.
- 25. Step III: The eighth town in **R** is C with demand 30. The full capacity of truck 5 is 60. Hence, town C is allocated to truck 5. The remaining capacity of truck 5 is revised to 30.
- 26. Step IV:  $j = 9$
- 27. Step III: The ninth town in **R** is J with demand 20. The remaining capacity of truck 5 is 30. Hence, town J is allocated to truck 5. The remaining capacity of truck 5 is revised to 10.
- 28. Step IV:  $j = 10$ . Stop.

Thus, the number of trucks required is 5. The routes taken by each of these trucks is given in Table 13.4.

## CHAPTER 14

# Supplier Selection with Multiple Criteria

A company wishes to place an order for 800 pieces of a particular item, for which 4 suppliers (S1, S2, S3, and S4) are available. Each supplier accepts orders in lot sizes that vary from supplier to supplier. The lot size  $l_j$  is 40, 20, 100, and 50 for suppliers S1, S2, S3, and S4, respectively. Each supplier  $j$  can supply only a maximum quantity  $m_j$  given by 400, 400, 600, and 300 for suppliers S1, S2, S3, and S4, respectively.

The company wishes to decide the quantity to be ordered from each supplier on the basis of four criteria<sup>1</sup> given by price, quality, delivery consistency, and process capability. Selection of suppliers using multiple objectives is difficult. For example, if a supplier is selected only on price criterion, the supplier may fare poorly in the other three criteria.

Analytic Hierarchy Process(AHP)<sup>2</sup> provides a powerful tool for handling multiple objective decision making, which the company decides to adopt for the purpose. This method is described in this chapter.

To apply AHP, the company first carries out a brain storming session amongst its stakeholders to decide the pair-wise comparison of the objectives. For example, the quality objective may be rated as seven times more important than the price objective. The final comparison obtained on a scale of one to nine (with one being the lowest and nine the highest) is given in Table 14.1.

We denote  $a_{ij}$  as the pair-wise comparison of objectives  $i$  and  $j$ . For example,  $a_{41} = 5$  gives the pair-wise comparison of objectives 4 (process capability) and 1 (price) and implies that the process quality objective is rated five times more important to the stakeholders than the price objective.

Second, Table 14.1 is normalized by dividing each element by the sum of the elements in each column. Thus  $a_{41} = 5$  is divided by

Table 14.1

	Price (1)	Quality (2)	Delivery consistency (3)	Process capability (4)
Price (1)	1	1/7	1/4	1/5
Quality (2)	7	1	4	5
Delivery consistency (3)	4	1/4	1	3
Process capability (4)	5	1/5	1/3	1

Table 14.2

	Price (1)	Quality (2)	Delivery consistency (3)	Process capability (4)
Price (1)	1/17	20/223	3/67	1/46
Quality (2)	7/17	140/223	48/67	25/46
Delivery consistency (3)	4/17	35/223	12/67	15/46
Process capability (4)	5/17	28/223	4/67	5/46

Table 14.3

	Price (1)	Quality (2)	Delivery consistency (3)	Process capability (4)
Weight	0.0538	0.5749	0.2244	0.1470

17(= 1 + 7 + 4 + 5) to obtain  $a'_{41} = \frac{5}{17}$ . The normalized table thus obtained is given in Table 14.2.

Third, the weight  $w_i$  of each objective  $i$  is then obtained by averaging the entries in the row pertaining to that objective. Thus, the weights of the objectives are obtained as given in Table 14.3. It will be noticed that the sum of the weights of the objectives is one.

Fourth, the company again carries out another brain storming session to decide the pair-wise comparison of the suppliers for each of the four objectives. The comparison is given in Table 14.4.

Table 14.4

Price					Quality				
	S1	S2	S3	S4		S1	S2	S3	S4
S1	1	7	6	3	S1	1	1/6	1	2
S2	1/7	1	1/4	1/3	S2	6	1	6	2
S3	1/6	4	1	1/4	S3	1	1/6	1	1/2
S4	1/3	3	4	1	S4	1/2	1/2	1	1
Delivery consistency					Process capability				
	S1	S2	S3	S4		S1	S2	S3	S4
S1	1	1/3	1/6	2	S1	1	4	3	1
S2	3	1	1/2	1	S2	1/4	1	2	1/3
S3	6	2	1	3	S3	1/3	1/2	1	1/2
S4	1/2	1	1/3	1	S4	1	3	2	1

Table 14.5

Price					Quality				
	S1	S2	S3	S4		S1	S2	S3	S4
S1	14/23	7/15	8/15	36/55	S1	2/17	1/11	1/9	4/11
S2	2/23	1/15	1/45	4/55	S2	12/17	6/11	2/3	4/11
S3	7/69	4/15	4/45	3/55	S3	2/17	1/11	1/9	1/11
S4	14/69	1/5	16/45	12/55	S4	1/17	3/11	1/9	2/11
Delivery consistency					Process capability				
	S1	S2	S3	S4		S1	S2	S3	S4
S1	2/21	1/13	1/12	2/7	S1	12/31	8/17	3/8	6/17
S2	2/7	3/13	1/4	1/7	S2	3/31	2/17	1/4	2/17
S3	4/7	6/13	1/2	3/7	S3	4/31	1/17	1/8	3/17
S4	1/21	3/13	1/6	1/7	S4	12/31	6/17	1/4	6/17

Fifth, Table 14.4 is normalized by dividing each element by the sum of the elements in each column. The normalized table thus obtained is given in Table 14.5.

Sixth, the score  $s_j$  of each supplier  $j$  for each objective is then obtained by averaging the entries in the row pertaining to that supplier and that

**Table 14.6**

Weight	0.0538	0.5749	0.2244	0.1470
	Price	Quality	Delivery consistency	Process capability
S1	0.57	0.17	0.14	0.40
S2	0.06	0.57	0.23	0.15
S3	0.13	0.10	0.49	0.12
S4	0.24	0.16	0.15	0.34

**Table 14.7**

Supplier	S1	S2	S3	S4
Final Score	0.2186	0.4046	0.1921	0.1885

objective. Thus the scores of the suppliers for the different objectives are obtained as given in Table 14.6.

Seventh, the final score  $s'_j$  for each supplier  $j$  is calculated using weights of each objective obtained earlier in Table 14.3 (and as given in the first row of Table 14.6) and scores  $s_j$ . Thus,  $s'_1$  for supplier S1 is given by  $(0.0538 * 0.57) + (0.5749 * 0.17) + (0.2244 * 0.14) + (0.1470 * 0.4) = 0.2186$ . The final score  $s'_j$  for all suppliers is given in Table 14.7.

In the eight step, we formulate a linear integer program as follows: Let  $x_j$  be integer decision variables denoting the number of lots ordered from supplier  $j$ . Because the company wishes to meet all the four objectives, the objective function is given by the following expression:

$$\text{Maximize } \sum_{j=1}^4 s'_j x_j l_j$$

The company has to obtain the 800 pieces from all the suppliers with whom the orders are placed. This is modelled by the following constraint:

$$\sum_{j=1}^4 x_j l_j \leq 800 \quad (\text{Constraint 1})$$

Each supplier  $j$  can supply only a maximum quantity  $m_j$ . This is modelled by the following constraint:

$$x_j l_j \leq m_j, \text{ for all } j \quad (\text{Constraint 2})$$

Table 14.8

Line	LINGO Program
1	SETS:
2	P:l,m,s,x;
3	ENDSETS
4	
5	DATA:
6	P=1 2 3 4;
7	l=40 20 100 50;
8	m=400 400 600 300;
9	s=0.2186 0.4046 0.1921 0.1885;
10	ENDDATA
11	
12	@FOR(P(J):@GIN(x(J)));
13	MAX=@SUM(P(J):s(J)*x(J)*l(J));
14	@SUM(P(J):x(J)*l(J))<=800;
15	@FOR(P(J):x(J)*l(J)<=m(J));

The LINGO program is given in Table 14.8. There is only one primitive set  $P$  declared in the SETS section corresponding to the four suppliers  $S_1$ ,  $S_2$ ,  $S_3$ , and  $S_4$ . The attributes of set  $P$  are  $l$  (corresponding to lot size  $l_j$ ),  $m$  (corresponding to maximum quantity  $m_j$ ),  $s$  (corresponding to final score  $s'_j$ ), and  $x$  (corresponding to integer decision variables  $x_j$  denoting the number of lots ordered) for each supplier  $j$ .

The integer decision variables  $x_j$  is declared in line 12 of the program. Constraints 1 and 2 are given in lines 14 and 15 of the program, respectively.

The solution gives an optimal order of 10 lots with supplier  $S_1$  and 20 lots with supplier  $S_2$ . No orders are placed with the suppliers  $S_3$  and  $S_4$ .



## SECTION 4

# Applications in Marketing Management



# CHAPTER 15

## Revenue Management

Let us take the case of an airline that has two 180 seater aircraft operating between four cities A, B, C, and D shown in Figure 15.1. Aircraft 1 leaves A and flies to C via B in the morning and returns to A in the evening through the same route once every day. Aircraft 2 leaves D and flies to C via B in the morning and returns to D in the evening through the same route once every day. For the purpose of this example, we will consider only the morning flights.

The airline uses two fare classes—Class I and Class II targeted towards the business and leisure traveller segments, respectively. The airline has developed the forecasts of demand for the two fare classes for the different origin-destination pairs served by the two aircraft in the morning, as given in Table 15.1.

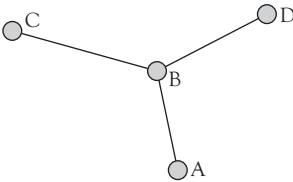


Figure 15.1

Table 15.1

Origin–destination	Class I fare	Class I demand	Class II fare	Class II demand
A–C	300	40	90	100
A–B	180	32	50	130
B–C	200	80	70	110
D–C	500	20	150	60
D–B	300	55	100	90

The airline has to decide on the allocation of number of aircraft seats to each of the fare classes on each aircraft to maximize the revenue.

In order to solve the problem,<sup>1,2</sup> we define integer decision variables  $x_{ijkl}$  which indicate the number of seats reserved for fare class  $k$  (which equals 1 and 2 for Class I and II fares, respectively) on aircraft number  $l$  operating between origin  $i$  and destination  $j$ . Thus we have 12 integer decision variables  $x_{AC11}$ ,  $x_{AC21}$ ,  $x_{AB11}$ ,  $x_{AB21}$ ,  $x_{BC11}$ ,  $x_{BC21}$ ,  $x_{BC12}$ ,  $x_{BC22}$ ,  $x_{DB12}$ ,  $x_{DB22}$ ,  $x_{DC12}$ , and  $x_{DC22}$ .

Because the objective is to maximize the revenue, the objective function is given by:

$$\text{Maximize } \sum x_{ijkl} f_{ijk}$$

where  $f_{ijk}$  is the fare of class  $k$  between origin  $i$  and destination  $j$ .

Because the seats allocated have to be less than or equal to the aircraft capacity, constraints have to be incorporated in the model accordingly. For example, the number of seats allocated on the AB leg has to be less than the aircraft capacity, which is modelled by the constraint  $x_{AB11} + x_{AB21} \leq 180$ . These constitute the first set of constraints.

Further, the seats allocated for each fare class on each flight leg has to be less than or equal to the demand forecast. For example the number of seats allocated on the AB leg for Class I fare has to be less than 32, which is modelled by the constraint  $x_{AB11} \leq 32$ . These constitute the second set of constraints.

The LINGO program is given in Table 15.2. The first set of constraints is given in lines 2–5. The second set of constraints is given in lines 6–15. The 12 integer decision variables  $x_{AC11}$ ,  $x_{AC21}$ ,  $x_{AB11}$ ,  $x_{AB21}$ ,  $x_{BC11}$ ,  $x_{BC21}$ ,  $x_{BC12}$ ,  $x_{BC22}$ ,  $x_{DC12}$ ,  $x_{DC22}$ ,  $x_{DB12}$ , and  $x_{DB22}$  are declared in lines 16–18.

Table 15.3 gives the seat allocations on the two aircraft for the different flight legs with a total revenue of 88,860 obtained by solving the Lingo program.

In practice, the capacity constraints in the above program are revised after regular intervals to account for reservations done for each flight leg and fare class in the preceding time interval. The programs are re-run to obtain revised seat allocations.

Table 15.2

Line	LINGO Program
1	$\text{max}=300*x_{AC11}+90*x_{AC21}+180*x_{AB11}+50*x_{AB21}+200*x_{BC11}+70*x_{BC21}+200*x_{BC12}+70*x_{BC22}+500*x_{DC12}+150*x_{DC22}+300*x_{DB12}+100*x_{DB22};$
2	$x_{AC11}+x_{AC21}+x_{AB11}+x_{AB21}\leq 180; \text{!constraint 1};$
3	$x_{AC11}+x_{AC21}+x_{BC11}+x_{BC21}\leq 180; \text{!constraint 2};$
4	$x_{DC12}+x_{DC22}+x_{DB12}+x_{DB22}\leq 180; \text{!constraint 3};$
5	$x_{DC12}+x_{DC22}+x_{BC12}+x_{BC22}\leq 180; \text{!constraint 4};$
6	$x_{AC11}\leq 40; \text{!constraint 5};$
7	$x_{AC21}\leq 100; \text{!constraint 6};$
8	$x_{AB11}\leq 32; \text{!constraint 7};$
9	$x_{AB21}\leq 130; \text{!constraint 8};$
10	$x_{BC11}+x_{BC12}\leq 80; \text{!constraint 9};$
11	$x_{BC21}+x_{BC22}\leq 110; \text{!constraint 10};$
12	$x_{DC12}\leq 20; \text{!constraint 11};$
13	$x_{DC22}\leq 60; \text{!constraint 12};$
14	$x_{DB12}\leq 55; \text{!constraint 13};$
15	$x_{DB22}\leq 90; \text{!constraint 14};$
16	$@GIN(x_{AC11});@GIN(x_{AC21}); @GIN(x_{AB11}); @GIN(x_{AB21});$
17	$@GIN(x_{BC11});@GIN(x_{BC21}); @GIN(x_{BC12}); @GIN(x_{BC22});$
18	$@GIN(x_{DC12});@GIN(x_{DC22}); @GIN(x_{DB12}); @GIN(x_{DB22});$

Table 15.3

Flight leg	Aircraft 1		Aircraft 2	
	Class I Fare	Class II Fare	Class I Fare	Class II Fare
A-C	40	50		
A-B	32	58		
B-C	0	90	80	20
D-C			20	60
D-B			55	45



## SECTION 5

# Applications in Financial Management



## CHAPTER 16

# Portfolio Management

Let us take a scenario where a certain sum of money  $M$  is available for investment in  $n$  stocks. The investor would like to ensure a minimum expected return  $r$  (in percent) on the investment and minimize the risk of the portfolio. The risk of the portfolio is measured by the variance of return earned by the portfolio.

Let  $x_i$  be the decision variable representing the amount invested in stock  $i$ . Let  $S_i$  be the random variable representing the return on one dollar invested in stock  $i$ . Let  $E(S_i)$  and  $\text{var}(S_i)$  represent the expected value and variance of the random variable  $S_i$ . Let  $\text{cov}(S_i, S_j)$  represent the covariance between the random variables,  $S_i, S_j$  pertaining to stocks,  $i$  and  $j$ , respectively.

Because the objective of the problem is to minimize the variance of return earned by the portfolio, the objective function<sup>1,2</sup> will be given by:

$$\min \sum_{i=1}^n x_i^2 \text{var}(S_i) + \sum_{i=1}^n \sum_{j=1, i \neq j}^n 2x_i x_j \text{cov}(S_i, S_j)$$

It will be observed that the objective function is non-linear.

Because the money available for investment in stocks is  $M$ , we have the following constraint:

$$\sum_{i=1}^n x_i = M \quad (\text{Constraint 1})$$

Because the investor wishes a minimum expected return of  $r$  percent on the investment, we have the following constraint:

$$\sum_{i=1}^n x_i E(S_i) \leq \frac{Mr}{100} \quad (\text{Constraint 2})$$

To illustrate this model, let us take the example of three stocks  $i=1,2,3$  for which data of increase in price over the last 10 years is given in Table 16.1.

**Table 16.1**

	Stock 1	Stock 2	Stock 3
Year 1	1.24	1.33	0.98
Year 2	1.23	1.25	1.29
Year 3	1.40	1.12	1.23
Year 4	1.10	0.96	1.14
Year 5	1.30	1.13	1.65
Year 6	0.90	1.43	1.34
Year 7	1.20	1.22	1.27
Year 8	0.80	1.06	1.03
Year 9	1.13	1.23	1.21
Year 10	1.21	1.32	1.12

**Table 16.2**

	Stock 1	Stock 2	Stock 3
<b>Stock 1</b>	0.0326	-0.0004167	0.0098
<b>Stock 2</b>	-0.0004167	0.0196	0.000356
<b>Stock 3</b>	0.0098	0.000356	0.0352

The expected value  $E(S_i)$  for the stock returns calculated from the data<sup>3</sup> given in Table 16.1 is 1.151, 1.205, and 1.226 for stocks 1, 2, and 3, respectively. The variance  $\text{var}(S_i)$  and the  $\text{cov}(S_i, S_j)$  are also calculated and given in Table 16.2.

The LINGO program is given in Table 16.3. Only one primitive set STOCKS is declared in the SETS section (line 2). The attributes of set STOCKS are E (denoting the expected value  $E(S_i)$  for the stock returns), VAR (denoting the variance  $\text{var}(S_i)$  for the stock returns), and X (denoting the decision variable  $x_i$  representing the amount invested in stock  $i$ ). A derived set S1 is formed using the primitive set. The attribute of set S1 is COVAR (denoting the co-variance  $\text{cov}(S_i, S_j)$  for stock returns of stocks  $i$  and  $j$ ).

The sum of money  $M$  is available for investment, and the minimum expected return  $r$  (in percent) is given in lines 7 and 8 of the DATA section. The values of attributes VAR and COVAR given in lines 11 and

Table 16.3

Line	LINGO Program
1	SETS:
2	STOCKS=E,VAR,X;
3	S1(STOCKS,STOCKS):COVAR;
4	ENDSETS
5	
6	DATA:
7	M=10000;
8	r=12;
9	STOCKS=1 2 3;
10	E=1.151 1.205 1.226;
11	VAR=0.0326 0.0196 0.0352;
12	COVAR=
13	0.0326        -0.0004167    0.0098
14	-0.0004167    0.0196        0.000356
15	0.0098        0.000356        0.0352;
16	ENDDATA
17	
18	min=@SUM(STOCKS(I):X(I)*X(I)*VAR(I))+
	@SUM(S1(I,J)   I#NE#J:2*X(I)*X(J)*COVAR(I,J));
19	@SUM(STOCKS(I):X(I))=M;
20	@SUM(STOCKS(I):X(I)*E(I))>=M*(r/100);

13–15, respectively, are obtained from Table 16.2. Constraints 1 and 2 are given in lines 19 and 20, respectively.

The optimal portfolio obtained from the solution of the LINGO program is investment of 2618, 5772, and 1610 dollars in stocks 1, 2, and 3, respectively, for a total investment of 10,000 dollars and a minimum return of 12%.



# CHAPTER 17

## Capital Budgeting

Let us take the case of a company that is considering five different projects P1, P2, ..., P5. All the projects take five years to complete. The funds outflow for the projects is given in Table 17.1 along with the net present value (NPV) of returns obtained from the projects in thousands of dollars.

The company can invest a maximum of 25 thousand dollars every year over the next five years. If the investment in any year is less than 25 thousand dollars, the remaining amount can be invested in subsequent years.

The company wishes to determine the projects to be taken up for investment such that the returns are maximized.

To solve this problem,<sup>1</sup> we define binary decision variable  $x_i$ , which equals 1 if project  $P_i$  is taken up for investment and 0 otherwise. We also define  $s_j$  as the amount of funds remaining in year  $j$  after investments;  $s_j$  can be utilized for investments in subsequent years.

Because the objective of the problem is the maximization of returns, the objective function of the problem is:

$$\text{Maximize } \sum_{i=1}^n x_i R_i$$

where,  $R_i$  is the net present value of returns of project P1 and  $n = 5$  is the number of projects under consideration.

**Table 17.1**

	P1	P2	P3	P4	P5
Year 1 outflow	5	10	10	15	5
Year 2 outflow	10	35	10	15	35
Year 3 outflow	25	25	20	20	30
Year 4 outflow	15	10	25	10	25
Year 5 outflow	30	20	10	5	10
NPV of returns	89	94	110	75	120

**Table 17.2**

Line	LINGO Program
1	@BIN(x1);@BIN(x2);@BIN(x3);@BIN(x4);@BIN(x5);
2	max=89*x1+94*x2+110*x3+75*x4+120*x5;
3	5*x1+10*x2+10*x3+15*x4+5*x5+s1=25;
4	10*x1+35*x2+10*x3+15*x4+35*x5+s1=25+s2;
5	25*x1+25*x2+20*x3+20*x4+30*x5+s2=25+s3;
6	15*x1+10*x2+25*x3+10*x4+25*x5+s3=25+s4;
7	30*x1+20*x2+10*x3+5*x4+10*x5+s4=25+s5;

The funds required for the projects should equal the funds available. This condition can be modelled by the following constraint:

$$\left( \sum_{i=1}^n x_i f_{ij} \right) + s_{j-1} = I_j + s_j, j = 1, 2, \dots, t \text{ (Constraint 1)}$$

Where  $t$  is the total number of years over which the projects require funds outflow (in this case, it is 5 years),  $f_{ij}$  is the cash outflow required for project  $i$  in year  $j$  (if that project is taken up for investment) and  $I_j$  is the funds available for investment in year  $j$ . The value of  $s_0$  is zero.  $I_j$  equals 25 thousand dollars for all years ( $j = 1, 2, 3, 4, 5$ ).

The LINGO program is given in Table 17.2. The binary decision variables  $x_i$  are declared in line 1. The constraint 1 for the five years is given in lines 3–7.

The solution of the problem using the LINGO program indicates that the company will maximize its returns if the projects P2, P3, and P5 are taken up for investment.

# CHAPTER 18

## Bank Asset Liability Management

A bank's liabilities are the funds used by the banks for lending and investment activities. The bank's capital, reserves, surplus, deposits, and borrowings typically constitute the liabilities. The bank's investments, advances, fixed assets, balances with central bank, and other banks constitute the assets of the bank. Asset Liability Management<sup>1</sup> seeks to manage the volume and mix of various assets and liabilities to minimize the risks, achieve the goals of the bank, ensure liquidity, and adhere to central bank norms.

In this example, we take the case of a bank which wishes to maximize its profit for a given liability whilst ensuring liquidity and adhering to central bank norms. We consider eight time periods (or buckets) for assets and liabilities as follows:

- (a) Bucket 1 of 1 to 14 days
- (b) Bucket 2 of 15 to 30 days
- (c) Bucket 3 of one to three months
- (d) Bucket 4 of three to six months
- (e) Bucket 5 of six months to one year
- (f) Bucket 6 of one to three years
- (g) Bucket 7 of three to five years
- (h) Bucket 8 of more than five years.

The following four liabilities are considered in the model for each bucket  $i$ , along with associated costs:

- (a) Demand deposit  $LDD_i$  with a cost  $CLDD_i$  of 0% for all buckets
- (b) Savings deposit  $LSD_i$  with a cost  $CLSD_i$  of 3.5% for all buckets
- (c) Term deposit  $LTD_i$  with cost  $CLTD_i$  of 3.5, 4.25, 5.75, 6.25, 8.5, 8.75, 9, and 10% for buckets 1, 2, 3, 4, 5, 6, 7, and 8, respectively

- (d) Borrowings  $LB_i$  with cost  $CLB_i$  of 3.5% for buckets 1 and 2, 5.5% for buckets 3, 4, and 5, 8.5% for bucket 6, 9% for bucket 7, and 9.5% for bucket 8.

The estimated demand deposits, savings deposits, and term deposits for all buckets are given in Table 18.1.

The following five assets are considered in the model for each bucket  $i$ , along with associated returns:

- (a) Balance with central bank  $ABCB_i$  with return  $RABCB_i$  of 3.5% for buckets 1 and 2, 5.5% for buckets 3, 4, and 5, 6% for bucket 6, 6.5% for bucket 7, and 7% for bucket 8
- (b) Balance with other banks  $ABOB_i$  with return  $RABOB_i$  of 3.5% for bucket 1, 4.25% for bucket 2, 5.75% for bucket 3, 6.25% for bucket 4, 8.5% for bucket 5, 8.75% for bucket 6, 9.5% for bucket 7, and 10% for bucket 8
- (c) Investment in government securities  $AGS_i$  with return  $RAGS_i$  of 3.5% for buckets 1 and 2, 5.5% for buckets 3, 4, and 5, 8.5% for bucket 6, 9% for bucket 7, and 9.5% for bucket 8
- (d) Investments in debentures and bonds  $ADB_i$  with rate of return  $RADB_i$  of 3.5% for buckets 1 and 2, 5.5% for buckets 3, 4 and 5, 8.5% for bucket 6, 9% for bucket 7, and 9.5% for bucket 8
- (e) Advances  $AA_i$  with return  $RAA_i$  of 5% for buckets 1 and 2, 6.5% for buckets 3 and 4, 8% for bucket 5, 9% for bucket 6, 9.5% for bucket 7, and 10% for bucket 8.

**Table 18.1**

	Demand deposit	Savings deposit	Term deposit
Bucket 1	5	5	12
Bucket 2	10	10	13
Bucket 3	10	10	18
Bucket 4	10	10	17
Bucket 5	10	10	10
Bucket 6	10	10	10
Bucket 7	10	10	10
Bucket 8	10	10	10

The profit is given by total revenue from the assets less the total cost of funds. Hence the objective function of the model is given by the following expression:

$$\text{Maximize } \sum_{i=1}^8 (R_i) - (C_i)$$

where

$$R_i = ABCB_i RABCB_i + ABOB_i RABOB_i + AGS_i RAGS_i \\ + ADB_i RADB_i + AA_i RAA_i$$

is the revenue earned in bucket  $i$ ,

$C_i = LDD_i CLDD_i + LSD_i CLSD_i + LTD_i CLTD_i + LB_i CLB_i$  is the cost incurred in bucket  $i$ .

The total revenues must equal the total costs for bucket 1 to maintain liquidity of funds. This is modelled by the following constraint:

$$ABCB_1 + ABOB_1 + AGS_1 + ADB_1 + AA_1 - LDD_1 \\ - LSD_1 - LTD_1 - LB_1 = 0$$

Similarly, the total revenues must equal the total cost for buckets 1 and 2 to maintain liquidity of funds. This is modelled by the following constraint:

$$\sum_{i=1}^2 (ABCB_i + ABOB_i + AGS_i + ADB_i + AA_i - LDD_i \\ - LSD_i - LTD_i - LB_i) = 0$$

Liquidity requirements for buckets 1–3, 1–4, ..., and 1–8 can be modelled similarly.

Banks also have certain restrictions on assets and liabilities, which have evolved through experience of the markets. These restrictions are:

- (a) Advances in each bucket should exceed 5% of total term advances in all eight buckets. Hence  $AA_j \geq 0.05 \sum_{i=1}^8 AA_i, j = 1, 2, \dots, 8$ .
- (b) Balance with central bank in each bucket should exceed 5% of total assets in all eight buckets. Hence

$$ABCB_j \geq 0.05 \sum_{i=1}^8 \left( ABCB_i + ABOB_i + AGS_i \right. \\ \left. + ADB_i + AA_i \right), j = 1, 2, \dots, 8.$$

- (c) Balance with central bank in bucket 8 should exceed 5% of total balances with central bank in all eight buckets. Hence

$$ABCB_8 \geq 0.05 \sum_{i=1}^8 ABCB_i.$$

- (d) Investment in government securities in bucket 8 should exceed 5% of total investment in government securities in all eight buckets.

$$\text{Hence } AGS_8 \geq 0.05 \sum_{i=1}^8 AGS_i.$$

- (e) Investment in debentures and bonds in bucket 8 should exceed 5% of total investment in debentures and bonds in all eight buckets.

$$\text{Hence } ADB_8 \geq 0.05 \sum_{i=1}^8 ADB_i.$$

- (f) The total investment in government securities in all buckets should exceed 24% of the total demand deposits, savings deposits, and term deposits in all buckets. Hence

$$\sum_{j=1}^8 AGS_j \geq 0.24 \sum_{i=1}^8 (LDD_i + LSD_i + LTD_i).$$

- (g) The total investment in assets in each bucket should be less than the total demand deposits, savings deposits, and term deposits in all buckets. Hence

$$ABCB_j + ABOB_j + AGS_j + ADB_j + AA_j \leq \sum_{i=1}^8 \begin{pmatrix} LDD_i + LSD_i \\ +LTD_i \end{pmatrix},$$

$$j = 1, 2, \dots, 8.$$

- (h) Borrowings in bucket 1 should exceed 80% of total borrowings in all buckets. Hence  $LB_1 \geq 0.8 \sum_{i=1}^8 LB_i$ .

- (i) Borrowings in each of the buckets 6, 7, and 8 should exceed 5% of total borrowings in all buckets. Hence  $LB_j \geq 0.05 \sum_{i=1}^8 LB_i, j = 6, 7, 8$ .

The LINGO program is given in Table 18.2. The optimal solution obtained is given in Table 18.3.

Table 18.2

LINGO Program
<p>SETS:            BUCKET:LDD,CLDD,LSD,CLSD,LTD,CLTD,LB,CLB,            ABCB,RACB,ABOB,RABOB,AGS,RAGS,ADB,RADB,AA,RAA,            R,C;            ENDSETS</p> <p>DATA:            BUCKET=1 2 3 4 5 6 7 8;            CLDD=0 0 0 0 0 0 0;            CLSD=3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5;            CLTD=3.5 4.25 5.75 6.25 8.5 8.75 9 10;            CLB=3.5 3.5 5.5 5.5 5.5 5.5 8.5 9 9.5;            RACB=3.5 3.5 5.5 5.5 5.5 5.5 6 6.5 7;            RABOB=3.5 4.25 5.75 6.25 8.5 8.75 9.5 10;            RAGS=3.5 3.5 5.5 5.5 5.5 5.5 8.5 9 9.5;            RADB=3.5 3.5 5.5 5.5 5.5 5.5 8.5 9 9.5;            RAA=5 5 6.5 6.5 8 9 9.5 10;            ENDDATA</p> <p>LSD(1)=5;LSD(2)=10;LSD(3)=10;LSD(4)=10;LSD(5)=10;LSD(6)=10;LSD(7)=10;LSD(8)=10;            LDD(1)=5;LDD(2)=10;LDD(3)=10;LDD(4)=10;LDD(5)=10;LDD(6)=10;LDD(7)=10;LDD(8)=10;            LTD(1)=12;LTD(2)=13;LTD(3)=18;LTD(4)=17;LTD(5)=10;LTD(6)=10;LTD(7)=10;LTD(8)=10;            MAX=@SUM(BUCKET(I):R(I)-C(I));            @FOR(BUCKET(I):R(I)=ABCB(I)*RACB(I)+ABOB(I)*RABOB(I)+AGS(I)*RAGS(I)              +ADB(I)*RADB(I)+AA(I)*RAA(I));            @FOR(BUCKET(I):C(I)=LDD(I)*CLDD(I)+LSD(I)*CLSD(I)+LTD(I)*CLTD(I)+LB(I)*CLB(I));            @SUM(BUCKET(I)   I=LE#1: ABCB(I)+ABOB(I)+AGS(I)+ADB(I)+AA(I)-LDD(I)-LSD(I)              -LTD(I)-LB(I))=0;            @SUM(BUCKET(I)   I=LE#2: ABCB(I)+ABOB(I)+AGS(I)+ADB(I)+AA(I)-LDD(I)-LSD(I)              -LTD(I)-LB(I))=0;            @SUM(BUCKET(I)   I=LE#3: ABCB(I)+ABOB(I)+AGS(I)+ADB(I)+AA(I)-LDD(I)-LSD(I)              -LTD(I)-LB(I))=0;            @SUM(BUCKET(I)   I=LE#4: ABCB(I)+ABOB(I)+AGS(I)+ADB(I)+AA(I)-LDD(I)-LSD(I)              -LTD(I)-LB(I))=0;            @SUM(BUCKET(I)   I=LE#5: ABCB(I)+ABOB(I)+AGS(I)+ADB(I)+AA(I)-LDD(I)-LSD(I)              -LTD(I)-LB(I))=0;            @SUM(BUCKET(I)   I=LE#6: ABCB(I)+ABOB(I)+AGS(I)+ADB(I)+AA(I)-LDD(I)-LSD(I)              -LTD(I)-LB(I))=0;            @SUM(BUCKET(I)   I=LE#7: ABCB(I)+ABOB(I)+AGS(I)+ADB(I)+AA(I)-LDD(I)-LSD(I)              -LTD(I)-LB(I))=0;            @SUM(BUCKET(I)   I=LE#8: ABCB(I)+ABOB(I)+AGS(I)+ADB(I)+AA(I)-LDD(I)-LSD(I)              -LTD(I)-LB(I))=0;            @FOR(BUCKET(J):AA(J))&gt;=0.05*@SUM(BUCKET(I):AA(I));!constraint(a);            @FOR(BUCKET(I):ABCB(I))&gt;=0.05*@SUM(BUCKET(J):ABCB(J)+ABOB(J)+AGS(J)+ADB(J)              +AA(J));!constraint(b);            ABCB(8)&gt;=0.05*@SUM(BUCKET(I):ABCB(I));!constraint(c);            AGS(8)&gt;=0.05*@SUM(BUCKET(I):AGS(I));!constraint(d);            ADB(8)&gt;=0.05*@SUM(BUCKET(I):ADB(I));!constraint(e);            @SUM(BUCKET(I):AGS(I))&gt;=0.24*@SUM(BUCKET(J):(LDD(J)+LSD(J)+LTD(J)));!con-              straint(f);            @FOR(BUCKET(I):ABCB(I)+ABOB(I)+AGS(I)+ADB(I)+AA(I)&lt;=@SUM(BUCKET(J):LDD(J)              +LSD(J)+LTD(J)));!constraint(g);            LB(1)&gt;=0.80*@SUM(BUCKET(I):LB(I));!constraint(h);            LB(6)&gt;=0.05*@SUM(BUCKET(I):LB(I));!constraint(i);            LB(7)&gt;=0.05*@SUM(BUCKET(I):LB(I));!constraint(i);            LB(8)&gt;=0.05*@SUM(BUCKET(I):LB(I));!constraint(i);</p>

*Table 18.3*

	Borrowings	Balance with Central Bank	Balance with other banks	Investment in government securities	Investment in debentures and bonds	Advances
Bucket 1	206.3	25.4	0	43.8	0	159.1
Bucket 2	4.6	25.4	0	0	0	12.2
Bucket 3	0	25.4	0	0.4	0	12.2
Bucket 4	0.6	25.4	0	0	0	12.2
Bucket 5	7.6	25.4	0	0	0	12.2
Bucket 6	12.9	25.4	0	5.3	0	12.2
Bucket 7	12.9	25.4	0	5.3	0	12.2
Bucket 8	12.9	25.4	0	5.3	0	12.2

# CHAPTER 19

## Index Fund Construction

Index fund portfolio<sup>1</sup> is a set of stocks chosen such that the movement of these stocks closely follow that of a population of stocks.

Let us take an example where the population contains six stocks, A, B, C, D, E, and F. We wish to choose an index fund portfolio consisting of only three stocks. How do we choose these 3 stocks from the set of 6 stocks such that the movement of these three stocks closely follow that of a population of six stocks.

First, we find the correlation  $\rho$  between the returns of each of the 6 stocks and the remaining stocks in the population set, as given in Table 19.1.

We define a binary decision variable  $x_{ij}$ , which equals 1 if stock  $i$  is represented by stock  $j$  of the index fund portfolio, and equals 0 otherwise. Because each stock  $i$  is represented by only one stock of the index fund portfolio, it follows that:

$$\sum_{j=1}^3 x_{ij} = 1, \text{ for } i = A, B, C, \dots, F \text{ (Constraint 1)}$$

Because our objective is to ensure that the index fund portfolio closely follows the population of stocks, we have to ensure that the sum of the

*Table 19.1*

	A	B	C	D	E	F
A	1	0.6	0.2	0.5	0.1	0.8
B	0.6	1	0.3	0.8	0.2	0.5
C	0.2	0.3	1	0.8	0.6	0.7
D	0.5	0.8	0.8	1	0.4	0.1
E	0.1	0.2	0.6	0.4	1	0.2
F	0.8	0.5	0.7	0.1	0.2	1

product of the correlation  $\rho_{ij}$  and the binary variable  $x_{ij}$  is maximized. We thus have the objective function:

$$\text{Max} \sum_{i=A,B,\dots}^F \sum_{j=1}^3 x_{ij} \rho_{ij}$$

We should have only 3 stocks in the index fund portfolio. For this purpose, we define a binary decision variable  $y_j$ , which equals 1 if stock  $j$  is included in the index fund portfolio, and equals 0 otherwise. Hence:

$$\sum_{j=A,B,\dots}^F y_j = 3 \quad (\text{Constraint 2})$$

Now, the binary variable  $x_{ij}$  equals 1 if stock  $i$  is represented by stock  $j$  of the index fund portfolio. Thus, if  $x_{ij} = 1$ , it is essential that  $y_j = 1$ . However, if  $x_{ij} = 0$ , the binary variable  $y_j$  can take values 0 or 1. These conditions can be represented by the following constraint:

$$x_{ij} \leq y_j, \text{ for } i, j = A, B, C, \dots, F \quad (\text{Constraint 3})$$

The LINGO program is given in Table 19.2. Only one primitive set STOCKS is declared in the SETS section (line 2). The attribute of set STOCKS is Y (denoting the binary decision variable  $y_j$ ). A derived set S1 is formed using the primitive set. The attributes of S1 are RHO (denoting correlation  $\rho$ ) and X (denoting the binary decision variable  $x_{ij}$ ).

The number of stocks in the index fund portfolio should equal 3. This is indicated by NOINDEX=3 given in line 8 of DATA section. The values of attribute RHO given in lines 10–15 are obtained from Table 19.1.

The binary decision variables  $x_{ij}$  and  $y_j$  are declared in lines 18 and 19. Constraints 1, 2, and 3 are given in lines 22, 21, and 23, respectively.

The solution to this problem is that stocks D, E, and F should be included in the index funds portfolio because they together reflect the population of stocks. Stocks A, B, and C are represented by stocks F, D, and D, respectively.

Table 19.2

Line	LINGO Program
1	SETS:
2	STOCKS=Y;
3	S1(STOCKS,STOCKS):RHO,X;
4	ENDSETS
5	
6	DATA:
7	STOCKS=A B C D E F;
8	NOINDEX=3;
9	RHO=
10	1    0.6   0.2   0.5   0.1   0.8
11	0.6   1    0.3   0.8   0.2   0.5
12	0.2   0.3   1    0.8   0.6   0.7
13	0.5   0.8   0.8   1    0.4   0.1
14	0.1   0.2   0.6   0.4   1    0.2
15	0.8   0.5   0.7   0.1   0.2   1;
16	ENDDATA
17	
18	@FOR(S1(I,J):@BIN(X(I,J)));
19	@FOR(STOCKS(I):@BIN(Y(I)));
20	max=@SUM(S1(I,J):RHO(I,J)*X(I,J));
21	@SUM(STOCKS(I):Y(I))=NOINDEX;
22	@FOR(STOCKS(I):@SUM(STOCKS(J):X(I,J))=1);
23	@FOR(STOCKS(I):@FOR(STOCKS(J):X(I,J)<=Y(J)));



## SECTION 6

# Applications in Transport Management



# CHAPTER 20

## Airline Network Design

Let us take the case of an airline company that is launching airline services between the six most populous cities of India, namely, Mumbai, Kolkata, Delhi, Chennai, Hyderabad, and Bangalore. The locations of the cities are given in Figure 20.1.

The aerial distances (in miles)  $a_{ij}$  between cities  $i$  and  $j$  for all pairs are given in Table 20.1.

The daily demand  $d_{ij}$  between the cities  $i$  and  $j$  are estimated for all pairs and given in Table 20.2.



Figure 20.1

**Table 20.1**

	Mumbai	Kolkata	Delhi	Chennai	Hyderabad	Bangalore
Mumbai	–	1095	724	657	404	532
Kolkata	1095	–	875	863	769	995
Delhi	724	875	–	1096	781	1082
Chennai	657	863	1096	–	323	184
Hyderabad	404	769	781	323	–	311
Bangalore	532	995	1082	184	311	–

**Table 20.2**

	Mumbai	Kolkata	Delhi	Chennai	Hyderabad	Bangalore
Mumbai	–	2825	2793	1998	1856	1856
Kolkata	2825	–	2505	1792	1665	1665
Delhi	2793	2505	–	1772	1646	1646
Chennai	1998	1792	1772	–	1178	1178
Hyderabad	1856	1665	1646	1178	–	1094
Bangalore	1856	1665	1646	1178	1094	–

The airline possesses two types of aircraft A1 and A2 with seating capacity  $b_k$ ,  $k = 1$  and 2 of 100 and 180, respectively. The relative cost per miles  $c_k$ ,  $k = 1$  and 2 for aircrafts A1 and A2 are 0.65 and 1. It may be noted that  $\frac{c_1}{b_1} > \frac{c_2}{b_2}$ .

Further, the airline has the following restrictions in operations to and from Hyderabad: (a) aircraft A2 cannot be used for any services to and from Hyderabad and Bangalore and (b) flights are operated from Hyderabad to Bangalore and Bangalore to Hyderabad only; thus, no flights are operated from Delhi or Kolkata or Mumbai or Chennai to Hyderabad and back.

The airline wishes to design the services between the cities using the two aircrafts A1 and A2 such that the demand is satisfied at the minimum total cost. There could either be a direct service between two cities or a service with one stop between the two cities.

We define the following decision variables for solving the problem:

- $y_{ijk}$  integer number of aircraft of type  $k$  used between cities  $i$  and  $j$
- $x_{ijl}$  fraction of demand between cities  $i$  and  $j$  met by a flight passing through city  $l$  ( $l \neq i, j$ )

Because we wish to minimize the total cost of transportation between all the cities, the objective function is given by the cost incurred by the aircraft plying between all pairs of cities as given below:

$$\text{Minimize } y_{ijk} c_k a_{ij}$$

The flights running between any pair of cities  $i$  and  $j$  will have a total capacity given by  $\sum_{k=1}^2 y_{ijk} b_k$ . This capacity should meet the demand between cities  $i$  and  $j$ ; and the fraction of other demands should be met by the flights running between any pair of cities  $i$  and  $j$ . This is given by the constraint below.

$$d_{ij} + \sum_{l \neq i, j} (x_{ijl} d_{il} + x_{lij} d_{lj} - x_{ilj} d_{ij}) \leq \sum_{k=1}^2 y_{ijk} b_k \quad (\text{Constraint 1})$$

Further, the sum of all fractions of demand between cities  $i$  and  $j$  met by a flight passing through cities  $l$  ( $l \neq i, j$ ) should be less than 1. This is given by the following constraint:

$$\sum_{l \neq i, j} x_{ilj} \leq 1 \quad (\text{Constraint 2})$$

The LINGO program is given in Table 20.3. The integer decision variables  $y_{ijk}$  are declared in lines 1–10. The relative cost per miles  $c_k$  and seating capacity  $b_k$  are given in line 12. The aerial distances (in miles)  $a_{ij}$  are given in lines 14–18. The daily demands  $d_{ij}$  are given in lines 20–24. Constraints pertaining to Constraint 1 are given in lines 28–54. Constraints pertaining to Constraint 2 are given in lines 56–88.

The solution of the number of flights operated between the cities using the two aircraft is given in Table 20.4.

It will be seen that there are no direct flights from Bangalore to Kolkata in the optimal solution though there is a demand of 1,665 daily. It will be seen from the solution report that  $x_{642} = 1$ , which means that the entire traffic from Bangalore to Kolkata passes through Chennai.

Table 20.3

Line	LINGO Model
1	@GIN(y121);@GIN(y131);@GIN(y141);@GIN(y161);
2	@GIN(y122);@GIN(y132);@GIN(y142);
3	@GIN(y211);@GIN(y231);@GIN(y241);@GIN(y261);
4	@GIN(y212);@GIN(y232);@GIN(y242);
5	@GIN(y321);@GIN(y311);@GIN(y341);@GIN(y361);
6	@GIN(y322);@GIN(y312);@GIN(y342);
7	@GIN(y421);@GIN(y431);@GIN(y411);@GIN(y461);
8	@GIN(y422);@GIN(y432);@GIN(y412);
9	@GIN(y561);
10	@GIN(y621);@GIN(y631);@GIN(y641);@GIN(y651);@GIN(y611);
11	
12	c2=1;c1=0.65;b2=180;b1=100;
13	
14	a12=1095;a13=724;a14=657;a15=404;a16=532;a21=a12;a31=a13;a41=a14;a51=a15;a61=a16;
15	a23=875;a24=863;a25=769;a26=995;a32=a23;a42=a24;a52=a25;a62=a26;
16	a34=1096;a35=781;a36=1082;a43=a34;a53=a35;a63=a36;
17	a45=323;a46=184;a54=a45;a64=a46;
18	a56=311;a65=a56;
19	
20	d12=2825;d13=2793;d14=1998;d15=1856;d16=1856;d21=d12;d31=d13;d41=d14;d51=d15;d61=d16;
21	d23=2505;d24=1792;d25=1665;d26=1665;d32=d23;d42=d24;d52=d25;d62=d26;
22	d34=1772;d35=1646;d36=1646;d43=d34;d53=d35;d63=d36;
23	d45=1178;d46=1178;d54=d45;d64=d46;
24	d56=1094;d65=d56;

25	min=	
26		c1*a12*y121+c1*a13*y131+c1*a14*y141+c1*a16*y161+ c2*a12*y122+c2*a13*y132+c2*a14*y142+c1*a21*y211+c1*a23*y231+c1*a24*y241+c1*a26*y261+ c2*a21*y212+c2*a23*y232+c2*a24*y242+ c1*a32*y321+c1*a31*y311+c1*a34*y341+c1*a36*y361+ c2*a32*y322+c2*a31*y312+c2*a34*y342+ c1*a42*y421+c1*a43*y431+c1*a41*y411+c1*a46*y461+ c2*a42*y422+c2*a43*y432+c2*a41*y412+ c1*a56*y561+ c1*a62*y621+c1*a63*y631+c1*a64*y641+c1*a65*y651+c1*a61*y611;
27		
28		d12+(x123*d13+x124*d14+x126*d16+x312*d32+x412*d42+x612*d62-x142*d12-x162*d12)<=y121*b1+y122*b2;
29		d13+(x132*d12+x134*d14+x136*d16+x213*d23+x413*d43+x613*d63-x123*d13-x143*d13-x163*d13)<=y131*b1+y132*b2;
30		d14+(x142*d12+x143*d13+x146*d16+x214*d24+x314*d34+x614*d64-x124*d14-x134*d14-x164*d14)<=y141*b1+y142*b2;
31		d16+(x162*d12+x163*d13+x164*d14+x165*d15+x216*d26+x316*d36+x416*d46-x126*d16-x136*d16-x146*d16)<=y161*b1;
32		
33		d21+(x213*d23+x214*d24+x216*d26+x321*d31+x421*d41+x621*d61-x231*d21-x241*d21-x261*d21)<=y211*b1+y212*b2;
34		d23+(x231*d21+x234*d24+x236*d26+x123*d13+x423*d43+x623*d63-x213*d23-x243*d23-x263*d23)<=y231*b1+y232*b2;
35		d24+(x241*d21+x243*d23+x246*d26+x124*d14+x324*d34+x624*d64-x214*d24-x234*d24-x264*d24)<=y241*b1+y242*b2;
36		d26+(x261*d21+x263*d23+x264*d24+x265*d25+x126*d16+x326*d36+x426*d46-x216*d26-x236*d26-x246*d26)<=y261*b1;
37		
38		d31+(x312*d32+x314*d34+x316*d36+x231*d21+x431*d41+x631*d61-x321*d31-x341*d31-x361*d31)<=y311*b1+y312*b2;
39		d32+(x321*d31+x324*d34+x326*d36+x132*d12+x432*d42+x632*d62-x312*d32-x342*d32-x362*d32)<=y321*b1+y322*b2;
40		d34+(x341*d31+x342*d32+x346*d36+x134*d14+x234*d24+x634*d64-x314*d34-x324*d34-x364*d34)<=y341*b1+y342*b2;

(Continued)

Table 20.3 (Continued)

Line	LINGO Model
41	d36+(x361*d31+x362*d32+x364*d34+x365*d35+x136*d16+x236*d26+x436*d46-x316*d36-x326*d36-x346*d36)<=y361*b1;
42	
43	d41+(x412*d42+x413*d43+x416*d46+x241*d21+x341*d31+x641*d61-x421*d41-x431*d41-x461*d41)<=y411*b1+y412*b2;
44	d42+(x421*d41+x423*d43+x426*d46+x142*d12+x342*d32+x642*d62-x412*d42-x432*d42-x462*d42)<=y421*b1+y422*b2;
45	d43+(x431*d41+x432*d42+x436*d46+x143*d13+x243*d23+x643*d63-x413*d43-x423*d43-x463*d43)<=y431*b1+y432*b2;
46	d46+(x461*d41+x462*d42+x463*d43+x465*d45+x146*d16+x246*d26+x346*d36-x416*d46-x426*d46-x436*d46)<=y461*b1;
47	
48	d56+(x561*d51+x562*d52+x563*d53+x564*d54)<=y561*b1;
49	
50	d61+(x612*d62+x613*d63+x614*d64+x261*d21+x361*d31+x461*d41+x561*d51-x621*d61-x631*d61-x641*d61)<=y611*b1;
51	d62+(x621*d61+x623*d63+x624*d64+x162*d12+x362*d32+x462*d42+x562*d52-x612*d62-x632*d62-x642*d62)<=y621*b1;
52	d63+(x631*d61+x632*d62+x634*d64+x163*d13+x263*d23+x463*d43+x563*d53-x613*d63-x623*d63-x643*d63)<=y631*b1;
53	d64+(x641*d61+x642*d62+x643*d63+x164*d14+x264*d24+x364*d34+x564*d54-x614*d64-x624*d64-x634*d64)<=y641*b1;
54	d65+(x165*d15+x265*d25+x365*d35+x465*d45)<=y651*b1;
55	
56	x132+x142+x162<=1;
57	x123+x143+x163<=1;
58	x124+x134+x164<=1;
59	x165<=1;
60	x126+x136+x146+x156<=1;
61	
62	x231+x241+x261<=1;
63	x213+x243+x263<=1;

64	$x_{214}+x_{234}+x_{264}\leq 1;$
65	$x_{265}\leq 1;$
66	$x_{216}+x_{236}+x_{246}+x_{256}\leq 1;$
67	
68	$x_{312}+x_{342}+x_{362}\leq 1;$
69	$x_{321}+x_{341}+x_{361}\leq 1;$
70	$x_{324}+x_{314}+x_{364}\leq 1;$
71	$x_{365}\leq 1;$
72	$x_{326}+x_{316}+x_{346}+x_{356}\leq 1;$
73	
74	$x_{432}+x_{412}+x_{462}\leq 1;$
75	$x_{423}+x_{413}+x_{463}\leq 1;$
76	$x_{421}+x_{431}+x_{461}\leq 1;$
77	$x_{465}\leq 1;$
78	$x_{426}+x_{436}+x_{416}+x_{456}\leq 1;$
79	
80	$x_{562}\leq 1;$
81	$x_{563}\leq 1;$
82	$x_{564}\leq 1;$
83	$x_{561}\leq 1;$
84	
85	$x_{632}+x_{642}+x_{612}\leq 1;$
86	$x_{623}+x_{643}+x_{613}\leq 1;$
87	$x_{624}+x_{634}+x_{614}\leq 1;$
88	$x_{621}+x_{631}+x_{641}\leq 1;$

Table 20.4

	Mumbai		Kolkata		Delhi		Chennai		Hyderabad		Bangalore	
	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2
Mumbai			0	15	1	15	0	12	0	0	19	0
Kolkata	0	16			0	14	0	19	0	0	0	0
Delhi	1	15	0	14			0	10	0	0	16	0
Chennai	0	11	0	20	0	10			0	0	29	0
Hyderabad	0	0	0	0	0	0	0	0			11	0
Bangalore	19	0	0	0	16	0	29	0	11	0		

## CHAPTER 21

# Performance Measurement Using Data Envelopment Analysis

Let us take the case of railroad companies for demonstrating the measurement of productivity and efficiency using the Operations Research technique known as Data Envelopment Analysis (DEA).

Railroad companies around the world transport either freight or passenger or both. The output of railroad companies is measured in terms of ton-kilometres (freight weighing a ton transported over a kilometre) for freight traffic and passenger-kilometres (a passenger transported over a kilometre) for passenger traffic.

Performance or the efficiency of transformation is measured in terms of units of output produced by a unit of input (given by  $\frac{\text{output}}{\text{input}}$ ).

Let us consider the manpower employed in the railroad as the only input under consideration for the moment. If the railroads under comparison transport only freight or passenger, the performance of the railroads can be compared in terms of ton-kilometres per manpower or passenger-kilometre per manpower, respectively. However, if we are comparing railroads carrying both freight and passenger, it becomes difficult to compare the performance of the railroads in this manner because it is difficult to add ton-kilometres and passenger-kilometres straightaway. Table 21.1 gives the output (in terms of freight ton-kilometres and passenger-kilometres) and input (in terms of manpower) of major railways of the world (source: UIC database at <http://www.uic.org/spip.php?article1352>).

If we were to assign weights to metrics of freights and passenger outputs, it is difficult to decide the weights because each railroad might value its freight and passenger traffic outputs in various ways. For example, if we

**Table 21.1**

No.	Railroads(acronym)	Input-Manpower	Output-Freight Ton-kilometres (in millions)	Output-Passenger-km (in millions)
1	Queensland Railway-Australia (QR)	15,439	106,314	1,546
2	Chinese Railways (CR)	2,041,600	3,473,660	787,890
3	Indian Railways (IR)	1,386,011	1,013,114	838,032
4	Japan Railways (JR)	128,761	55,665	244,235
5	Kazakhstan Railway (KR)	93,251	344,257	14,860
6	France (SNCF)	156,434	57,873	85,697
7	Germany (DB)	239,888	202,975	76,772
8	Italy FS Spa	12,215	39,673	44,404
9	Poland (PKP)	113,107	62,998	16,454
10	Russian Federation (RZD)	1,030,878	3,261,827	153,575
11	Turkey (TCDD)	29,966	17,183	5,374
12	Ukraine (UZ)	364,189	408,294	48,327
13	South Africa Spoornet (SA)	24,811	165,800	991
14	Canada all railroads (CAND)	32,310	558,549	1,422
15	USA all railroads (USR)	171,741	3,939,804	9,518

take the case of the Indian Railways and the Chinese Railways, assigning a weight of 0.7 to freight traffic output and a weight of 0.3 to passenger traffic output would show that the Chinese Railways is far more efficient than the Indian Railways. Again assigning a weight of 0.1 to freight traffic output and a weight of 0.9 to passenger traffic output would show that the Indian Railways is more efficient than the Chinese Railways.

DEA<sup>1</sup> is a technique that allows each railroad to value its freight and passenger traffic outputs in the best possible way while evaluating a railroad in relation to other railroads. The DEA technique starts with the assumption that the efficiency will lie between 0 and 1. Here weights are assigned for the outputs and inputs based on the data for the inputs and outputs of the all railroads being compared in such a way that the efficiency of the railroad under evaluation is maximized and the efficiency of

none of the other railroad exceeds 1. This technique thus allows a railroad to choose the best possible weights so that its efficiency is maximized while ensuring that those weights are feasible. Feasible weights imply that the efficiency of none of the railroads should exceed 1 using those weights. If the railroad under evaluation obtains an efficiency of 1 while choosing the best possible weights, it is efficient in relation to the other railroads under comparison. Else, the railroad is inefficient in comparison with the railroads under comparison. It may also be noted that the efficiency obtained by this method does not depend on the units used for the inputs and outputs, provided the same units have been used for all the railroads under comparison.

The DEA formulation for determination of weights for evaluation of Indian Railways (railroad number 3 in the table above) in comparison with 14 other railroads given in the table above would be given as:

$$\text{Max } e_3$$

$$e_j = \frac{x_f o_{jf} + x_p o_{jp}}{y_m i_{jm}}, j = 1, 2, \dots, 15$$

$$e_j \leq 1, j = 1, 2, \dots, 15$$

where,  $e_j$  is the efficiency of railroad  $j$ ,  $o_{jf}$  is the freight ton-kilometres of railroad  $j$ ,  $o_{jp}$  is the passenger-kilometres of railroad  $j$ ,  $i_{jm}$  is the manpower employed in railroad  $j$ ,  $x_f$  is the decision variable determining the weight of freight traffic output,  $x_p$  is the decision variable determining the weight of passenger traffic output, and  $y_m$  is the decision variable determining the weight of manpower input.

The above formulation cannot be solved by linear programming software because there are fractions involved. However, keeping in view that the maximum efficiency attainable is 1, we can equate the denominator of  $e_3$  to 1 and maximize the numerator of  $e_3$ . Again because  $e_j \leq 1$ , the second constraint above can be converted to the constraint:

$$x_f o_{jf} + x_p o_{jp} \leq y_m i_{jm}, j = 1, 2, \dots, 15$$

Thus the final DEA formulation will be as follows:

$$\text{Max } x_f o_{3f} + x_p o_{3p}$$

subject to the constraints:

$$y_m i_{3m} = 1 \text{ (Constraint 1)}$$

$$x_f o_{jf} + x_p o_{jp} \leq y_m i_{jm}, j = 1, 2, \dots, 15 \text{ (Constraint 2)}$$

The LINGO program is given in Table 21.2. There is only one primitive set RAILROAD declared in line 2. The attributes of RAILROAD set are

**Table 21.2**

Line	LINGO Program
1	SETS:
2	RAILROAD:MANPOWER,FREIGHTTONKM,PASSKM,OUT,IN;
3	ENDSETS
4	
5	DATA:
6	R=3;
7	RAILROAD=1 2 3 4 5 6 7 8 9 10 11 12 13 14 15;
8	MANPOWER=
9	15.439    2041.6    1386.011    128.761    93.251
10	156.434    239.888    12.215    113.107    1030.878
11	29.966    354.189    24.811    32.31    171.741
12	;
13	FREIGHTTONKM=
14	106.314    3473.66    1013.114    55.665    344.257
15	57.873    202.975    39.673    62.998    3261.827
16	17.183    408.294    165.8    558.549    3939.804
17	;
18	PASSKM=
19	1.546    787.89    838.032    244.235    14.86
20	85.697    76.772    44.404    16.454    153.575
21	5.374    48.327    0.991    1.422    9.518
22	;
23	ENDDATA
24	
25	max=@SUM(RAILROAD(J)   J#EQ#R:xf*FREIGHTTONKM(J)+ xp*PASSKM(J));
26	@SUM(RAILROAD(J)   J#EQ#R:ym*MANPOWER(J))=1;
27	@FOR(RAILROAD(J):(xf*FREIGHTTONKM(J)+xp*PASSKM (J))<=ym*MANPOWER(J));
28	@FOR(RAILROAD(J):OUT(J)=xf*FREIGHTTONKM(J)+xp*PASSKM(J));
29	@FOR(RAILROAD(J):IN(J)=ym*MANPOWER(J));

MANPOWER (corresponding to  $i_{jm}$ ), FREIGHTTONKM (corresponding to  $o_{jf}$ ), PASSKM (corresponding to  $o_{jp}$ ), OUT (corresponding to the weighted output:  $x_f o_{jf} + x_p o_{jp}$ ), and IN (corresponding to weighted input:  $y_m i_{jm}$ ). The variable R given in line 6 indicates the railroad for which the efficiency calculation is being made. This corresponds to the column “No” of Table 21.1. Because efficiency calculations are being made for Indian Railways, R has been set equal to 3.

It will be observed that the inputs and outputs for all the railroads are given in thousands in the LINGO program. This is done because very large or very small numbers should be avoided in any program by scaling the data appropriately. Constraints 1 and 2 are given in lines 26 and 27, respectively. The weighted outputs and inputs are calculated in lines 28 and 29, respectively.

Solving the LINGO program, we get  $x_f = 0.00003$ ,  $x_p = 0.00017$  and  $y_m = 0.00072$ . The efficiencies of the Indian Railways (railroad 3) works out to 0.175, whereas the efficiencies of the other railroads are given in Table 21.3. Because the efficiency of the Indian Railways is below 1, it is termed as inefficient.

It will be observed that the Italy FS Spa and USA Railroads have an efficiency of 1.00. It may be noted that these two railroads (known as the reference set to the railroad under evaluation) forces the Indian Railways to be inefficient.

Similar calculations of efficiency can be carried out for other railroads by using different values of R in line 6 of the LINGO program. The results for all the railroads are summarized in Table 21.4.

It will be observed that only Italy FS Spa and USA Railroads having an efficiency of 1.00 are efficient. The rest of the railroads are inefficient. Italy FS Spa and USA Railroads constitute the reference set for the inefficient railroads.

**Table 21.3**

Railroad no	1	2	4	5	6	7	8	9	10	11	12	13	14	15
Railroad	QR	CR	JR	KR	SNCF	DB	FS	PKP	RZD	TCDD	UZ	SA	CAND	USR
Efficiencies	0.32	0.17	0.47	0.20	0.15	0.11	1.00	0.06	0.17	0.07	0.08	0.30	0.75	1.00

**Table 21.4**

No.	Railroads (acronym)	$x_f$	$x_p$	$y_m$	Efficiency	Reference set
1	Queensland Railway -Australia (QR)	0.00279	0.01533	0.06477	0.32	FS, USR
2	Chinese Railways (CR)	0.00002	0.00012	0.00049	0.165	FS, USR
3	Indian Railways (IR)	0.00003	0.00017	0.00072	0.175	FS, USR
4	Japan Railways (JR)	0	0.00214	0.00777	0.522	FS
5	Kazakhstan Railway (KR)	0.00046	0.00254	0.01072	0.197	FS, USR
6	France (SNCF)	0	0.00176	0.00639	0.151	FS
7	Germany (DB)	0.00018	0.00099	0.00417	0.112	FS, USR
8	Italy FS Spa	0.00352	0.01937	0.08187	1	USR
9	Poland (PKP)	0.00038	0.00209	0.00884	0.058	FS, USR
10	Russian Federation (RZD)	0.00004	0.00023	0.00097	0.171	FS, USR
11	Turkey (TCDD)	0.00144	0.00790	0.03337	0.067	FS, USR
12	Ukraine (UZ)	0.00012	0.00067	0.00282	0.082	FS, USR
13	South Africa Spoornet (SA)	0.00173	0.00954	0.04030	0.297	FS, USR
14	Canada all railroads (CAND)	0.00133	0.00732	0.03095	0.754	FS, USR
15	USA all railroads (USR)	0.00025	0.00138	0.00582	1	FS

# Notes

## Chapter 1

1. <http://www.scienceofbetter.org/>
2. <http://www.learnaboutor.co.uk/>
3. Enterprise Resource Planning (ERP) systems enable financial accounting and control, manufacturing planning, plant maintenance, quality management, materials management, sales and distribution management and human resources management through integration of databases across the organization (Shutb, 2002). It involves integration of Material Requirement Planning (MRP) systems that support production planning and control with Database Management Systems (DBMS), Decision Support Systems (DSS) and Management Information Systems (MIS) to meet the data and information requirements of all parts of an organization. ERP systems typically consist of a model base and databases. The model base containing models for forecasting, material management, inventory management, capacity planning, scheduling, sequencing etc. are used along with the database for automated decision making for integrated production and order management. Further ERP may be extended for supplier relationship management, customer relationship management, supply chain management and warehouse management
4. Customer Relationship Management (CRM) is defined by the information technology research and advisory company Gartner (Yunker & Nelson, 2004) as “a business strategy that maximizes profitability, revenue and customer satisfaction by (a) Organizing around customer segments, (b) Fostering behaviour that satisfies customers and (c) Implementing customer-centric processes....thereby enabling greater customer insight, increased customer access, more effective interactions and integration throughout all customer channels and back-office enterprise functions.” Customer insight is sought to understand the drivers of customer behaviour in terms of relationship length (length of buyer-seller relationship), depth(frequency of buying) and breadth (cross buying behaviour), (Wubben, 2008). CRM involves (i) designing the customer experience through redesigned touchpoints and processes along with customer service level agreements (SLA) for key processes (ii) creation of customer information system which is integrated with operational and analytical systems and (iii) designing and monitoring customer experience metrics.
5. <http://ifors.org/web/>

6. (<http://www.euro-online.org/web/pages/1/home>)
7. (<https://www.informs.org/>)
8. (<http://www.theorsociety.com/>)
9. (<http://www.orsi.in/>)
10. (<https://www.informs.org/Community/Simulation-Society>)
11. (<http://www.agifors.org/>)
12. (<https://www.informs.org/Recognize-Excellence/Franz-Edelman-Award>)
13. (<https://www.informs.org/Community/RAS>)
14. (<http://www.aimms.com/community/modeling-competitions>)
15. (<http://www.mhi.org/cicmhe/competition>)
16. Golden Mountain Dollars (acronym GMD) is the currency of the fictitious country Republic of Golden Mountains; this currency is used throughout the book.
17. (<http://www.orms-today.org/ormsmain.shtml>)
18. (<http://analytics-magazine.org/>)
19. (<http://plato.asu.edu/guide.html>)
20. (<https://www.or-exchange.org/>)
21. (<http://neos-guide.org/>)
22. (<http://www.coin-or.org/index.html>)
23. (<http://www.neos-server.org/neos/>)
24. Lines are given by the equation  $y = mx + b$ , where  $m$  is the slope of the line and  $b$  is the  $y$ -axis intercept. Two lines  $y = m_1x + b_1$  and  $y = m_2x + b_2$  are parallel if  $m_1 = m_2$ . For any objective function value of  $V$ , the objective function can be written as  $x + 2y = V$  or  $y = \frac{V}{2} - \frac{x}{2}$ . The straight line associated with constraint 1 is  $100x + 200y = 1000$  or  $y = 5 - \frac{x}{2}$ . Thus the straight lines associated with both the objective function and constraint 1 are parallel because  $m_1 = m_2 = -\frac{1}{2}$ .

## Chapter 3

1. Daskin (1995), pp. 110–113.
2. Daskin (1995), pp. 160–162.
3. Daskin (1995), pp. 250–251.
4. Daskin (1995), pp. 276–277.

## Chapter 4

1. Hillier et al. (2012), pp. 403–404.

## Chapter 5

1. Anderson et al. (2012), pp. 508–509.

## Chapter 6

1. Pinedo (2009), pp. 88–89.

## Chapter 7

1. Gueret et al. (2002), pp. 214–215.

## Chapter 10

1. Bisschop (2012), pp. 235–237.

## Chapter 11

1. Geoffrion and Graves (1974), pp. 822–823.

## Chapter 12

1. Srinivasan (2010), p. 366.

## Chapter 13

1. Goetschalckx (2011), pp. 288–293.

## Chapter 14

1. Sharma and Dubey (2010), pp. 1171–1178.
2. Winston (2003), pp. 785–791.

## Chapter 15

1. Anderson et al. (2012), pp. 224–229.
2. Schrage (1997), pp. 191–192.

## Chapter 16

1. Winston and Venkataramanan (2003), pp. 723–724.
2. Schrage (1997), pp. 372–374.
3. Expected value  $\bar{d}$  of a set of  $N$  data  $(d_1, d_2, \dots, d_N)$  is given by  $\frac{\sum d_i}{N}$ .  
 Variance  $var$  is given by  $\frac{\sum (d_i - \bar{d})^2}{N-1}$ . Co-variance between two sets of data  $(d_1, d_2, \dots, d_N)$  and  $(t_1, t_2, \dots, t_N)$  is given by  $\frac{\sum (d_i - \bar{d})(t_i - \bar{t})}{N-1}$ .

## Chapter 17

1. Winston and Venkataramanan (2003), pp. 479–480.

## Chapter 18

1. Dash and Pathak (2011), pp. 53–58.

## Chapter 19

1. Cornuejols and Tutuncu (2006), pp. 217–218.

## Chapter 20

1. Jaillet et al. (1996), pp. 197–198.

## Chapter 21

1. Cooper et al. (2002), pp. 23–24.

# References

- Anderson, D. R., Sweeney, D. J., Williams, T. A., Camm, J. D., & Martin, K. (2012). *An introduction to management science: Quantitative approaches to decision making* (XIIIth ed.). Mason, OH: South-Western Cengage Learning.
- Bisschop, J. (2012). *AIMMS optimization modeling*. Haarlem, NL: Paragon Decision Technology BV.
- Cooper, W. W., Seiford, L. M., & Tone, K. (2002). *Data envelopment analysis*. New York, NY: Kluwer Academic Publishers.
- Cornuejols, G., & Tutuncu, R. (2006). *Optimization methods in finance*. Cambridge, GB: Cambridge University Press.
- Dash, M., & Pathak, R. (2011). A linear programming model for assessing asset-liability management in banks. *The IUP Journal of Financial Risk Management* 8(1), 50–67.
- Daskin, M. S.(1995). *Network and discrete location: Models, algorithms, and applications* (1st ed.). New York, NY: John Wiley and Sons.
- Geoffrion, A. M., & Graves, G. W.(1974). Multicommodity distribution system design by bender's decomposition. *Management Science* 20(5), 822–844.
- Goetschalckx, M. (2011). *Supply chain engineering*(1st ed.). New York, NY: Springer Science + Business Media.
- Gueret, C., Prins, C., Sevaux, M., & Heipcke, S. (2002). *Applications of optimization with Xpress-MP* (1st ed.). Northants, GB: Dash Optimization Ltd.
- Hillier, F. S., Lieberman, G. J., Nag, B. & Basu, P. (2012). *Special indian edition of introduction to operations research* (IXth ed.). New Delhi, DL: Tata McGraw Hill Education Private Limited.
- Jaillet, P., Song, G., & Yu, G. (1996). Airline network design and hub location problems. *Location Science* 4(3), 195–212.
- Pinedo, M. L. (2009). *Planning and scheduling in manufacturing and services* (IIInd ed.). New York, NY: Springer Science + Business Media.
- Schrage, L. (1997). *Optimization modeling with LINGO* (Vth ed.). Chicago, IL:LINDO Systems.
- Sharma, S., & Dubey, D. (2010). Multiple sourcing decisions using integrated AHP and knapsack model: A case on carton sourcing. *International Journal of Advanced Manufacturing Technology* 51, 1171–1178.
- Shutb, A. (2002). *Enterprise resource planning (ERP): The dynamics of operations management*. Kluwer Academic Publishers.

- Srinivasan, G. (2010). *Operations research: Principles and applications* (IIInd ed.). New Delhi, DL: PHI Learning Private Limited.
- Winston, W. L. (2003). *Operations research: Applications and algorithms* (IVth ed.). Independence, KY: Cengage Learning.
- Winston, W. L., & Venkataramanan, M. (2003). *Introduction to mathematical programming* (IVth ed.). Belmont, CA: Brooks/Cole Cengage Learning.
- Wubben, M. (2008). *Analytical CRM-developing and maintaining profitable customer relationships in non-contractual settings*. Dissertation of Technische Universität München, published by Gabler | GWV Fachverlage GmbH, Wiesbaden.
- Yunker, E., & Nelson, S. (2004). Gartner's Research Paper No IGG-06252003-01 "Management update: The eight building blocks of CRM"; retrieved from [www.gartner.com/2\\_events/crmawards/2004/emea/building\\_blocks.pdf](http://www.gartner.com/2_events/crmawards/2004/emea/building_blocks.pdf) on 15 October 2013.

# Index

## A

Advertising problem, 6–11  
AHP. *See* Analytic hierarchy process  
Airline network design, 129–136  
Analytic hierarchy process (AHP), 97

## B

Bank asset liability management,  
117–122  
Binary decision variable, 12  
Binary integer programming model,  
23–24  
Binary programming model, 12

## C

Cable layout, 45–51  
Capital budgeting, 115–116  
Constraints, 9

## D

Data envelopment analysis (DEA),  
137–142  
DEA. *See* Data envelopment analysis  
Decision sciences, 3  
Decision variables  
binary, 12  
definition, 9  
Diet planning. *See* Dog's diet blending  
Dog's diet blending, 71–72  
Dual prices, 16

## E

Exponential probability distribution,  
54

## F

Facility location problem, 35–43  
Franz Edelman competition, 5

## I

Index fund portfolio, 123  
Infeasible model, 21–22  
Integer programming model, 22–24

## L

Linearly constrained optimization  
model, 11  
Linear programming (LP)  
dual prices, 16  
infeasible model, 21–22  
modeling software, 12  
multiple solutions, 17–19  
parameters of model, 10  
reduced cost, 16–17  
slack, 16  
surplus, 21  
unbounded model, 19  
LINGO software  
description, 12–14  
SETS function, 24–29  
solving problem, 14–16  
types of models, 11–12  
*vs.* integer programming model,  
30  
LINGO software  
airline network design, 129–136  
bank asset liability management,  
117–122  
capital budgeting, 115–116  
data envelopment analysis,  
137–142  
dog's diet, 71–72  
index fund construction, 123–125  
multicommodity transport  
planning, 79–86  
paper roll trimming, 73–76  
portfolio management, 111–113  
production line scheduling,  
55–59  
production planning, 67–70  
product mix, 33–34  
revenue management, 105–107  
shift staff planning, 61–65  
supermarket  
capacity limitation, 42  
capital costs, 40–42  
customer travelling costs, 40–42

- number of people, 35–38
  - supplier selection with multiple criteria, 97–101
- LP. *See* Linear programming
- M
- Management science, 3
- Mixed integer linear programming model, 11
- Model parameters. *See* Linear Programming (LP)
- Multicommodity transport planning, 79–86
- Multiple criteria optimization. *See* Analytic hierarchy process (AHP)
- Multiple delivery truck routing, 93–96
- Multiple solutions. *See* Linear Programming (LP)
- N
- Network Design. *See* Airline Network Design.
- O
- Objective function, 9
- Objective function coefficients, 9
- Operations research, 3
- P
- Paper roll trimming, 73–76
- Parameters of model. *See* Linear Programming (LP)
- Performance measurement, data envelopment analysis, 137–142
- Poisson probability function, 53
- Portfolio management, 111–113
- Production line scheduling, 55–59
- Production planning, 67–70
- Product mix, 33–34
- Pure integer linear programming model, 11
- Q
- Quadratic programming model, 11
- Queue behaviour, Poisson probability function, 53
- R
- Railroads performance, 137
- Reduced Cost. *See* Linear Programming (LP)
- Revenue management, 105–107
- Routing. *See* Truck Routing
- S
- SETS function, 24–29
- Scheduling Production Lines. *See* Production Line Scheduling
- Shift staff planning, 61–65
- Single delivery truck routing, 87–91
- Slack variables, 21
- Stochastic programming models, 12
- Supermarket location, LINGO program
  - capacity limitation, 42
  - capital costs, 40–42
  - customer travelling costs, 40–42
  - number of people, 35–38
- Supplier selection with multiple criteria, 97–101
- Surplus. *See* Linear Programming (LP)
- T
- Technological coefficients, 10
- Transport Planning. *See* Multicommodity transport planning
- Truck routing
  - multiple delivery, 93–96
  - single delivery, 87–91
- U
- Unbounded model, 19
- Unconstrained nonlinear programming model, 11

## OTHER TITLES IN QUANTITATIVE APPROACHES TO DECISION MAKING COLLECTION

Donald Stengel, California State University, Fresno, Editor

- *Working With Sample Data: Exploration and Inference* by Priscilla Chaffe-Stengel and Donald N. Stengel
- *Business Applications of Multiple Regression* by Ronny Richardson
- *Operations Methods: Waiting Line Applications* by Ken Shaw
- *Regression Analysis: Understanding and Building Business and Economic Models Using Excel* by J. Holton Wilson, Barry P. Keating and Mary Beal-Hodges

## FORTHCOMING IN THIS COLLECTION

- *Forecasting Across the Organization 12/31/2013* by Ozgun Caliskan Demirag, Diane Parente and Carol L. Putman
- *Effective Applications of Statistical Process Control 4/15/2014* by Ken Shaw
- *Leveraging Business Analysis for Project Success 5/15/2014* by Vicki James
- *Construction Projects and the Power of Design-Build: Ensuring Safety and Control in Project Delivery using the SAFEDB Methodology 6/15/2014* by Sherif Hashem
- *Project Risk: Concepts, Process, and Tools 7/15/2014* by Tom R. Wielicki and Donald N. Stengel
- *Effective Applications of Supply Chain Logistics 7/15/2014* by Ken Shaw

## Announcing the Business Expert Press Digital Library

*Concise E-books Business Students Need  
for Classroom and Research*

This book can also be purchased in an e-book collection by your library as

- a one-time purchase,
- that is owned forever,
- allows for simultaneous readers,
- has no restrictions on printing, and
- can be downloaded as PDFs from within the library community.

Our digital library collections are a great solution to beat the rising cost of textbooks. e-books can be loaded into their course management systems or onto student's e-book readers.

The **Business Expert Press** digital libraries are very affordable, with no obligation to buy in future years. For more information, please visit [www.busessexpertpress.com/librarians](http://www.busessexpertpress.com/librarians). To set up a trial in the United States, please contact **Adam Chesler** at [adam.chesler@busessexpertpress.com](mailto:adam.chesler@busessexpertpress.com) for all other regions, contact **Nicole Lee** at [nicole.lee@igroupnet.com](mailto:nicole.lee@igroupnet.com).



## THE BUSINESS EXPERT PRESS DIGITAL LIBRARIES

### EBOOKS FOR BUSINESS STUDENTS

Curriculum-oriented, born-digital books for advanced business students, written by academic thought leaders who translate real-world business experience into course readings and reference materials for students expecting to tackle management and leadership challenges during their professional careers.

### POLICIES BUILT BY LIBRARIANS

- *Unlimited simultaneous usage*
- *Unrestricted downloading and printing*
- *Perpetual access for a one-time fee*
- *No platform or maintenance fees*
- *Free MARC records*
- *No license to execute*

The Digital Libraries are a comprehensive, cost-effective way to deliver practical treatments of important business issues to every student and faculty member.

**For further information, a  
free trial, or to order, contact:**

[sales@businessexpertpress.com](mailto:sales@businessexpertpress.com)

[www.businessexpertpress.com/librarians](http://www.businessexpertpress.com/librarians)

## Business Applications of Operations Research

**Bodhibrata Nag**

Operations Research is a bouquet of mathematical techniques that have evolved over the last six decades to improve the process of business decision making. Operations Research offers tools to optimize and find the best solutions to myriad decisions that managers have to take in their day to day operations or while carrying out strategic planning. Today, with the advent of operations research software, these tools can be applied by managers even without any knowledge of the mathematical techniques that underlie the solution procedures.

The book starts with a brief introduction to various tools of operations research, such as linear programming and integer programming together with simple examples in each of the areas. The introductory chapter also covers operations research software, along with examples. Other tools of operations research such as multi-objective programming, queuing theory, and network theory are discussed in subsequent chapters along with their appropriate applications in business decision making.

The book will give you real awareness of the power and potential of operations research in addressing decision making in areas of operations, supply chain, and financial and marketing management. You'll be encouraged to use the accompanying software models to solve these problems, using detailed do-it-yourself instructions, and as you progress, you'll gain familiarity with and an intuitive understanding of the various tools of operations research and their applications to various business situations.

**Bodhibrata Nag** is an associate professor of Operations Management at Indian Institute of Management Calcutta. His research interests are applications of operations research in areas of logistics, transportation, energy, and public sector management. He has authored a book, *Optimal Design of Timetables for Large Railways*, and co-authored the Special Indian Edition of Hillier & Lieberman's classic textbook, *Introduction to Operations Research*. In addition he has co-authored three chapters for the forthcoming book *Case Studies of Realistic Applications of Optimum Decision Making* being published by Springer.

### QUANTITATIVE APPROACHES TO DECISION MAKING COLLECTION

Donald N. Stengel, Editor



[www.businessexpertpress.com](http://www.businessexpertpress.com)

ISBN: 978-1-60649-526-1



9 781606 495261