M.Sc Final Year

Mathematics

OPERATIONS RESEARCH

वाग्देवी तामुपास्महे

मध्यप्रदेश भोज (मुक्त) विश्वविद्यालय – भोपाल

MADHYA PRADESH BHOJ (OPEN) UNIVERSITY - BHOPAL

COURSE WRITERS

S Kalavathy Dean, Department of Science and Humanities, K Ramakrishnan College of Technology, Samayapuram, Trichy

Units (1.0-1.2, 1.4, 1.5.1-1.5.3, 1.6-1.15, 2, 4.0-4.3, 4.5-4.10, 5)

CR Kothari, Ex-Principal, University Commerce College, Jaipur

Units(1.3, 1.5, 3, 4.4)

Information contained in this book has been published by VIKAS® Publishing House Pvt. Ltd. and has been obtained by its Authors from sources believed to be reliable and are correct to the best of their knowledge. However, the Madhya Pradesh Bhoj (Open) University, Bhopal, Publisher and its Authors shall in no event be liable for any errors, omissions or damages arising out of use of this information and specifically disclaim any implied warranties or merchantability or fitness for any particular use.

Published by Registrar, MP Bhoj (Open) University, Bhopal in 2020

VIKAS®

Vikas® is the registered trademark of Vikas® Publishing House Pvt. Ltd.

VIKAS® PUBLISHING HOUSE PVT. LTD.
E-28, Sector-8, Noida - 201301 (UP)
Phone: 0120-4078900 • Fax: 0120-4078999
Regd. Office: A-27, 2nd Floor, Mohan Co-operative Industrial Estate, New Delhi 1100 44
• Website: www.vikaspublishing.com • Email: helpline@vikaspublishing.com

SYLLABI-BOOK MAPPING TABLE

Operations Research

Syllabi	Mapping in Book
Unit-1: Operations Research and its Scope. Linear Programming - Simplex Method. Theory of the Simplex Method. Duality and Sensitivity Analysis. Other Algorithms for Linear Programming - Dual Simplex Method, Parametric Linear Programming, Upper Bound Technique, Interior Point Algorithm. Linear Goal Programming.	**Unit-1:** Operations Research and its Scope **(Pages 3-60)**
Unit-2: Transportation and Assignment Problems. Network Analysis - Shortest Path Problem. Minimum Spanning Tree Problem, Maximum Flow Problem. Minimum Cost Flow Problem, Network Simplex Method. Project Planning and Control with PERT-CPM.	**Unit-2:** Transportation and Assignment Problems **(Pages 61-135)**
Unit-3: Dynamic Programming - Deterministic and Probabilistic Dynamic programming. Game Theory - Two-Person, Zero-Sum Games. Games with Mixed Strategies. Graphical Solution, Solution by Linear Programming.	**Unit-3:** Dynamic Programming **(Pages 137-201)**
Unit-4: Integer Programming - Branch and Bound Technique, Simulation Replacement Problems, sequencing.	**Unit-4:** Integer Programming **(Pages 203-249)**
Unit-5: Nonlinear Programming - One and Multi-Variable Unconstrained Optimization. Kuhn-Tucker Conditions for Constrained Optimization. Quadratic Programming. Separable Programming. Convex Programming. Non-convex Programming.	**Unit-5:** Nonlinear Programming **(Pages 251-291)**

CONTENTS

INTRODUCTION

NOTES

Operational Research, or simply OR, originated in the context of military operations, but today it is widely accepted as a powerful tool for planning and decision-making, especially in business and industry. The OR approach has provided a new tool for managing conventional management problems. In fact, operational research techniques do constitute a scientific methodology of analysing the problems of the business world. They provide an improved basis for taking management decisions. The practice of OR helps in tackling intricate and complex problems, such as that of resource allocation, product mix, inventory management, sequencing and scheduling, replacement and a host of similar problems of modern business and industry.

Operations research is an interdisciplinary branch of applied mathematics and formal science that uses mathematical methods, such as mathematical modelling, statistics and algorithms to arrive at optimal or near optimal solutions to complex problems. Basically, it is concerned with optimizing the maxima (profit, assembly line performance, bandwidth, etc.) or minima (loss, risk, etc.) of some objective function. It also helps management achieve its goals using scientific methods. The field of operations research is closely related to the Industrial Engineering and hence the Industrial Engineers consider operations research techniques as their major toolset. Some of the primary tools used by operations researchers are statistics, optimization, probability theory, queuing theory, game theory, graph theory, decision analysis and simulation. Because of the computational nature of these fields, OR is linked to computer science and OR professionals use specific custom-written software for computation of data and decision-making. The uniqueness of OR prompted industries to use its formal tools, such as operations analysis, system analysis, management science, decision science, etc. Commercial industries, such as airlines, automobiles, communications, electronics, transportation, chemicals and mining use OR techniques to optimally utilize their limited resources and thereby maximize profits. Hence, OR is the application of the methods of science to complex problems arising in the direction and management of large systems of men, machines, materials and money in industry, business, government and defence. The distinctive approach is to develop a scientific model of the system, incorporating measurement of factors, such as chance and risk, with which to predict and compare the outcomes of alternative decision strategies and controls.

Operations research provides top-level administrators a quantitative basis for taking decisions which will help organizations to carry out their functions, such as planning, controlling and organizing, effectively. Decision-making is the key responsibility of managers and OR provides a scientific approach to them for solving problems. Decisions in an organization should be such that they can compete in the market. We can say that the OR and decision-making processes are interlinked. There are intangible factors also, such as human behaviour, which OR has to take into account when calculating for a solution.

This book, *Operations Research* is divided into five units that follow the self-instruction mode with each unit beginning with an Introduction to the unit, followed by an outline of the Objectives. The detailed content is then presented in a simple but structured manner interspersed with Check Your Progress Questions to test the student's understanding of the topic. A Summary along with a list of Key Terms and a set of Self-Assessment Questions and Exercises is also provided at the end of each unit for recapitulation.

UNIT 1 OPERATIONS RESEARCH AND ITS SCOPE

NOTES

Structure

1.0 INTRODUCTION

Operations Research (OR) was coined by J.F. McCloskey and F.N. Trefethen in 1940 in Bawdsey in the United Kingdom. This innovative science was discovered during World War II for a specific military situation, when military management sought decisions based on the optimal consumption of limited military resources with the help of an organized and systematized scientific approach. This was termed as Operations Research (OR) or operation research. Thus, OR was known as an ability to win a war without really going into a battlefield or fighting it. It is a new field of scientific and managerial application and an exact definition that is acceptable uniformly is not possible and different experts have given different definitions for it.

Linear Programming (LP, also known as linear optimization) is a technique for achieving the optimal result (such as, highest profit or lowest cost) in a mathematical model with linear criteria. Mathematical programming is a subset of linear programming (also known as mathematical optimization). In more formal terms, linear programming is a method for optimizing a linear objective function that is constrained by linear equality and inequality constraints. It has a convex polytope as its feasible region, which is a set defined as the intersection of a finite number of half spaces, each of which is specified by a linear inequality. This polyhedron's objective function is a real-valued affine (linear) function. If there is a location in the polytope where this function has the smallest (or greatest) value, a linear programming approach identifies it.

Simplex algorithm (or simplex method) is a widely used linear programming algorithm in mathematics. Simplex method is an iterative procedure for solving LPP (Linear Programming Problem) in a finite number of steps. This method provides an algorithm which consists of moving from one vertex of the region of feasible solution to another in such a manner that the value of the objective function at the succeeding vertex is less or more as the case may be that at the previous vertex. This procedure is repeated and since the number of vertices is finite, the method leads to an optimal vertex in a finite number of steps or indicates the existence of unbounded solution.

The principle of duality, often known as the duality principle, states that optimization issues can be seen from two perspectives: the primal problem or the dual problem. The dual problem's answer gives a lower constraint on the primal (minimization) problem's solution. However, the optimal values of the primal and dual issues do not have to be equivalent in most cases. The dualism gap is the name given to their disparity. Under a constraint qualifying condition, the duality gap for convex optimization problems is zero. The term sensitivity analysis, often known as post-optimality refers to the optimal solution of a linear programming problem, formulated using various method.

Parametric linear programming is a type of mathematical optimization, where the optimization problem is solved as a function of one or multiple parameters. Developed in parallel to sensitivity analysis, its earliest mention can be found in a thesis from 1952. Since then, there have been considerable developments for the cases of multiple parameters, presence of integer variables as well as nonlinearities. A Linear Programming Problem (LPP) may have, in addition to the regular constraints, lower or upper bounds on some or all the variables. Interior Point Methods (also referred to as barrier methods or IPMs) are a certain class of algorithms that solve linear and nonlinear convex optimization problems.

In this unit, you will learn about the basics of operations research, linear programming, simplex method, duality and sensitivity analysis, dual simplex method, parametric linear programming, upper bound techniques, interior point algorithm and linear goal programming.

1.1 OBJECTIVES

After going through this unit, you will be able to:

- Understand the definition and scope of Operations Research (OR)
- Know about the applications of OR
- Understand the significance of linear programming problem
- Elaborate on the concept of linear programming
- Define the meaning of simplex method
- Explain the concept of duality and sensitivity analysis
- Apply principles of IPP in real life situations
- Know about the parametric linear programming
- Define upper bound techniques

- Discuss about the interior point algorithm
- Explain the linear goal programming

1.2 BASICS OF OPERATIONS RESEARCH

NOTES

The term, operations research was first coined in 1940 by J.F. McCloskey and F.N. Trefethen in a small town Bowdsey, the United Kingdom. This new science came into existence in a military context. During World War II, military management called on scientists from various disciplines and organized them into teams to assist in solving strategic and tactical problems, relating to air and land defence. Their mission was to formulate specific proposals and plans for aiding the Military commands to arrive at decisions on optimal utilization of scarce military resources and attempts to implement these decisions effectively. This new approach to the systematic and scientific study of the operations of the system was called Operations Research (OR) or operational research. Hence, OR can be associated with 'An art of winning the war without actually fighting it.'

Definitions

Operations Research (OR) has been defined so far in various ways and it is perhaps still too young to be defined in some authoritative way. It is not possible to give uniformly acceptable definitions of OR. A few opinions about the definition of OR are given below. These have been changed according to the development of the subject.

> OR is a scientific method of providing executive departments with a quantitative basis for decisions regarding the operations under their control.
>
> *Morse and Kimball (1946)*
>
> OR is the scientific method of providing executive with an analytical and objective basis for decisions.
> *P.M.S. Blackett (1948)*
>
> OR is a systematic method-oriented study of the basic structures, characteristics, functions and relationships of an organization to provide the executive with a sound, scientific and quantitative basis for decision-making.
>
> *E.L. Arnoff and M.J Netzorg*
>
> OR is a scientific approach to problem solving for executive management.
>
> *H.M. Wagner*
>
> OR is an aid for the executive in making his decisions by providing him with the quantitative information based on the scientific method of analysis.
>
> *C.Kittee*
>
> OR is the scientific knowledge through interdisciplinary team effort for the purpose of determining the best utilization of limited resources.
>
> *H.A. Taha*

The various definitions given here bring out the following essential characteristics of operations research:

(*i*) System Orientation
(*ii*) Use of Interdisciplinary Terms
(*iii*) Application of Scientific Methods
(*iv*) Uncovering New Problems
(*v*) Quantitative Solutions
(*vi*) Human Factors

1.2.1 Scope of Operations Research

There is a great scope for economists, statisticians, administrators and the technicians working as a team to solve problems of defence by using the OR approach. Besides this, OR is useful in various other important fields like:

(*i*) Agriculture
(*ii*) Finance
(*iii*) Industry
(*iv*) Marketing
(*v*) Personnel Management
(*vi*) Production Management
(*vii*) Research and Development

Phases of Operations Research

The procedure to be followed in the study of OR generally involves the following major phases:

(*i*) Formulating the Problem
(*ii*) Constructing a Mathematical Model
(*iii*) Deriving the Solution from the Model
(*iv*) Testing the Model and its Solution (updating the model)
(*v*) Controlling the Solution
(*vi*) Implementation

1.3 LINEAR PROGRAMMING

Decision-making has always been very important in the business and industrial world, particularly with regard to the problems concerning production of commodities. Which commodity/commodities to produce, in what quantities and by which process or processes, are the main questions before a production manager. English economist Alfred Marshall pointed out that the businessman always studies his production function and his input prices and substitutes one input for another till his costs become the minimum possible. All this sort of substitution, in the opinion of Marshall, is being done by businessman's trained instinct rather than with formal calculations. But now there does exist a method of formal calculations often termed as Linear Programming. This method was first formulated by a Russian mathematician L.V. Kantorovich, but it was developed later in 1947 by George B. Dantzig 'for the purpose of scheduling the complicated procurement activities of the United States Air Force'. Today, this method is being used in solving a wide range of practical business problems. The advent of electronic computers has further increased its applications to solve many other problems in industry. It is being considered as one of the most versatile management tools.

Meaning of Linear Programming

Linear Programming (LP) is a major innovation since World War II in the field of business decision-making, particularly under conditions of certainty. The word 'Linear' means that the relationships are represented by straight lines, i.e., the relationships are of the form $y = a + bx$ and the word 'Programming' means taking decisions systematically. Thus, LP is a decision-making technique under given constraints on the assumption that the relationships amongst the variables representing different phenomena happen to be linear. In fact, Dantzig originally called it 'Programming of interdependent activities in a linear structure' but later shortened it to 'Linear Programming'. LP is generally used in solving maximization (sales or profit maximization) or minimization (cost minimization) problems subject to certain assumptions. Putting in a formal way, 'Linear Programming is the maximization (or minimization) of a linear function of variables subject to a constraint of linear inequalities.' Hence, LP is a mathematical technique designed to assist the organization in optimally allocating its available resources under conditions of certainty in problems of scheduling, product-mix and so on.

Fields Where Linear Programming can be Used

The problem for which LP provides a solution may be stated to maximize or minimize for some dependent variable which is a function of several independent variables when the independent variables are subject to various restrictions. The dependent variable is usually some economic objectives, such as profits, production, costs, work weeks, tonnage to be shipped, etc. More profits are generally preferred to less profits and lower costs are preferred to higher costs. Hence, it is appropriate to represent either maximization or minimization of the dependent variable as one of the firm's objective. LP is usually concerned with such objectives under given constraints with linearity assumptions. In fact, it is powerful to take in its stride a wide range of business applications. The applications of LP are numerous and are increasing every day. LP is extensively used in solving resource allocation problems. Production planning and scheduling, transportation, sales and advertising, financial planning, portfolio analysis, corporate planning, etc., are some of its most fertile application areas. More specifically, LP has been successfully applied in the following fields:

(*i*) *Agricultural Applications:* LP can be applied in farm management problems as it relates to the allocation of resources such as acreage, labour, water supply or working capital in such a way that is maximizes net revenue.

(*ii*) *Contract Awards:* Evaluation of tenders by recourse to LP guarantees that the awards are made in the cheapest way.

(*iii*) *Industrial Applications:* Applications of LP in business and industry are of most diverse kind. Transportation problems concerning cost minimization can be solved by this technique. The technique can also be adopted in solving the problems of production (product-mix) and inventory control.

Thus, LP is the most widely used technique of decision-making in business and industry in modern times in various fields as stated above.

1.4 SIMPLEX METHOD

Simplex method is an iterative procedure for solving LPP in a finite number of steps. This method provides an algorithm which consists of moving from one vertex of the region of feasible solution to another in such a manner that the value of the objective function at the succeeding vertex is less or more as the case may be that at the previous vertex. This procedure is repeated and since the number of vertices is finite, the method leads to an optimal vertex in a finite number of steps or indicates the existence of unbounded solution.

Definition

(*i*) Let X_B be a basic feasible solution to the LPP.

Max $Z = C_X$

Subject to $A_X = b$ and $X \geq 0$, such that it satisfies $X_B = B^{-1}b$,

Where B is the basic matrix formed by the column of basic variables.

The vector $C_B = (C_{B1}, C_{B2} \ldots C_{Bm})$, where C_{Bj} are components of C associated with the basic variables is called the cost vector associated with the basic feasible solution X_B.

(*ii*) Let X_B be a basic feasible solution to the LPP.

Max $Z = C_X$, where $A_X = b$ and $X \geq 0$.

Let C_B be the cost vector corresponding to X_B. For each column vector a_{ij} in A_1, which is not a column vector of B, let

$$a_j = \sum_{i=1}^{m} a_{ij} b_j$$

Then the number $Z_j = \sum_{i=1}^{m} C_{Bi} a_{ij}$ is called the evaluation corresponding to a_{ij} and the number $(Z_j - C_j)$ is called the net evaluation corresponding to j.

Simplex Algorithm

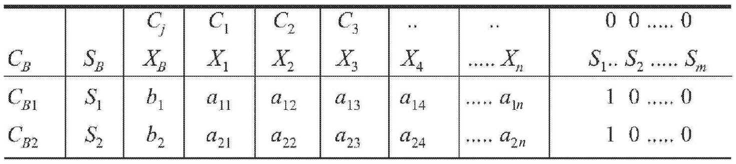
For the solution of any LPP by simplex algorithm, the existence of an initial basic feasible solution is always assumed. The steps for the computation of an optimum solution are as follows:

Step 1: Check whether the objective function of the given LPP is to be maximized or minimized. If it is to be minimized then we convert it into a problem of maximization by, Min $Z = -\text{Max}\ (-Z)$

Step 2: Check whether all b_i ($i = 1, 2, \ldots, m$) are positive. If any one of b_i is negative, then multiply the inequation of the constraint by -1 so as to get all b_i to be positive.

Step 3: Express the problem in the standard form by introducing slack/surplus variables to convert the inequality constraints into equations.

Step 4: Obtain an initial basic feasible solution to the problem in the form $X_B = B^{-1}b$ and put it in the first column of the simplex table. Form the initial simplex table shown as follows:



		C_j	C_1	C_2	C_3	..	..	0 0 0
C_B	S_B	X_B	X_1	X_2	X_3	X_4	..... X_n	S_1.. S_2 ..... S_m
C_{B1}	S_1	b_1	a_{11}	a_{12}	a_{13}	a_{14}	..... a_{1n}	1 0 0
C_{B2}	S_2	b_2	a_{21}	a_{22}	a_{23}	a_{24}	..... a_{2n}	1 0 0

Step 5: Compute the net evaluations $Z_j - C_j$ by using the relation:

$Z_j - C_j = C_B\ (a_j - C_j)$

Examine the sign of $Z_j - C_j$:

(*i*) If all $Z_j - C_j \geq 0$, then the initial basic feasible solution X_B is an optimum basic feasible solution.

(*ii*) If at least one $Z_j - C_j > 0$, then proceed to the next step as the solution is not optimal.

Step 6: To find the entering variable, i.e., key column.

If there are more than one negative $Z_j - C_j$ choose the most negative of them. Let it be $Z_r - C_r$ for some $j = r$. This gives the entering variable X_r and is indicated by an arrow at the bottom of the *r*th column. If there are more than one variable having the same most negative $Z_j - C_j$, then any one of the variable can be selected arbitrarily as the entering variable.

(*i*) If all $X_{ir} \leq 0$ ($i = 1, 2, \ldots, m$) then there is an unbounded solution to the given problem.

(*ii*) If at least one $X_{ir} > 0$ ($i = 1, 2, \ldots, m$), then the corresponding vector X_r enters the basis.

Step 7: To find the leaving variable or key row:

Compute the ratio ($X_{Bi}/X_{kr}, X_{ir}{>}0$)

If the minimum of these ratios be X_{Bi}/X_{kr}, then choose the variable X_k to leave the basis called the key row and the element at the intersection of the key row and the key column is called the key element.

Step 8: Form a new basis by dropping the leaving variable and introducing the entering variable along with the associated value under C_B column. The leaving element is converted to unity by dividing the key equation by the key element and all other elements in its column to zero by using the formula:

New element = Old element

$$-\left[\frac{\text{Product of elements in key row and key column}}{\text{Key element}}\right]$$

Step 9: Repeat the procedure of Step (5) until either an optimum solution is obtained or there is an indication of unbounded solution.

Example 1.1: Use simplex method to solve the following LPP.

Maximize, $Z = 3X_1 + 2X_2$

Subject to, $X_1 + X_2 \le 4$

$X_1 - X_2 \le 2$

$X_1, X_2 \ge 0$

Solution: By introducing the slack variables S_1 and S_2 convert the problem into standard form.

Max, $Z = 3X_1 + 2X_2 + 0S_1 + 0S_2$

Subject to, $X_1 + X_2 + S_1 = 4$

$X_1 - X_2 + S_2 = 2$

$X_1, X_2, S_1, S_2 \ge 0$

$$\begin{bmatrix} X_1 & X_2 & S_1 & S_2 \\ 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \\ S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

An initial basic feasible solution is given by,

$X_B = B^{-1}b,$

Where, $B = I_2, X_B = (S_1, S_2)$

i.e., $(S_1, S_2) = I_2 = (4, 2)$

Initial Simplex Table

$Z_j = C_B\, a_j$

$$Z_1 - C_1 = C_B a_1 - C_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} - 3 = -3$$

$$Z_2 - C_2 = C_B a_2 - C_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} - 2 = -2$$

$$Z_3 - C_3 = C_B a_3 - C_3 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} - 0 = -0$$

$$Z_4 - C_4 = C_B a_4 - C_4 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix} - 0 = -0$$

		C_j	3	2	0	0	
C_B	B	X_B	X_1	X_2	S_1	S_2	Min $\frac{X_B}{X_1}$
0	S_1	4	1	1	1	0	4/1 = 4
←0	S_2	2	①	–1	0	1	2/1 = 2
	Z_j	0	0	0	0	0	
	$Z_j - C_j$		–3↑	–2	0	0	

Since, there are some $Z_j - C_j = 0$, the current basic feasible solution is not optimum.

Since, $Z_1 - C_1 = -3$ is the most negative, the corresponding non-basic variable X_1 enters the basis.

The column corresponding to this X_1 is called the key column.

$$\text{Ratio } = \text{Min}\left\{\frac{X_{Bi}}{X_{ir}}, X_{ir} > 0\right\}$$

$$= \text{Min}\left\{\frac{4}{1}, \frac{2}{1}\right\}, \text{ which corresponds to } S_2$$

∴ The leaving variable is the basic variable S_2. This row is called the key row. Convert the leading element X_{21} to units and all other elements in its column n, i.e., (X_1) to zero by using the formula:

New element = Old element –

$$\left[\frac{\text{Product of elements in key row and key column}}{\text{Key element}}\right]$$

To apply this formula, first we find the ratio, namely

$$\frac{\text{The element to be zero}}{\text{Key element}} = \frac{1}{1} = 1$$

Apply this ratio for the number of elements that are converted in the key row. Multiply this ratio by key row element shown as follows:

1×2

1×1

1×-1

1×0

1×1

Now, subtract this element from the old element. The element to be converted into zero is called the old element row. Finally, we have

$4 - 1 \times 2 = 2$

$1 - 1 \times 1 = 0$

$1 - 1 \times -1 = 2$

$1 - 1 \times 0 = 1$

$0 - 1 \times 1 = -1$

∴ The improved basic feasible solution is given in the following simplex table.

First Iteration

		C_j	3	2	0	0	
C_B	B	X_B	X_1	X_2	S_1	S_2	Min $\frac{X_B}{X_2}$ ⟵
←0	S_1	2	0	$\boxed{2}$	1	–1	2/2 = 1
3	X_1	2	1	–1	0	1	–
	Z_j	6	3	–3	0	0	
	$Z_j - C_j$		0	–5↑	0	0	

Since, $Z_2 - C_2$ is the most negative, X_2 enters the basis.

To find Min $\left(\frac{X_B}{X_{i2}}, X_{i2} > 0\right)$

$$\text{Min}\left[\frac{2}{2}, \frac{2}{-1}\right] = 1$$

(∴ Negative or zero values are not considered)

This gives the outgoing variables. Convert the leaving element into one. This is done by dividing all the elements in the key row by 2. The remaining elements are converted to zero by using the following formula.

Here, $-\frac{1}{2}$ is the common ratio. Put this ratio 5 times and multiply each ratio by the key row element.

$-\frac{1}{2} \times 2$

$-\frac{1}{2} \times 0$

$-\frac{1}{2} \times 2$

$-1/2 \times 1$

$-1/2 \times -1$

Subtract this from the old element. All the row elements which are converted into zero are called the old elements.

$2 - \left(-\frac{1}{2} \times 2\right) = 3$

$1 - (-1/2 \times 0) = 1$

$-1 - (-1/2 \times 2) = 0$

$0 - (-1/2 \times 1) = 1/2$

$1 - (-1/2 \times -1) = 1/2$

Second Iteration

	C_j		3	2	0	0
C_B	B	X_B	X_1	X_2	S_1	S_2
2	X_2	1	0	1	1/2	–1/2
3	X_1	3	1	0	1/2	1/2
	Z_j	11	3	2	5/2	1/2
	Z_j–C_j		0	0	5/2	1/2

Since all $Z_j - C_j \geq 0$, the solution is optimum. The optimal solution is Max $Z = 11$, $X_1 = 3$, and $X_2 = 1$.

Example 1.2: Solve the LPP

$$\text{Max } Z = 3x_1 + 2x_2$$

Subject to,

$$4x_1 + 3x_2 \leq 12$$
$$4x_1 + x_2 \leq 8$$
$$4x_1 - x_2 \leq 8$$
$$x_1,\ x_2 \geq 0$$

Solution: Convert the inequality of the constraint into an equation by adding slack variables S_1, S_2, S$_3$.

$$\text{Max } Z = 3x_1 + 2x_2 + 0S_1 + 0S_2 + 0S_3$$

Subject to,

$$4x_1 + 3x_2 + S_1 = 12$$
$$4x_1 + x_2 + S_2 = 8$$
$$4x_1 - x_2 + S_3 = 8$$
$$x_1,\ x_2,\ S_1,\ S_2,\ S_3 \geq 0$$

$$\begin{bmatrix} x_1 & x_2 & S_1 & S_2 & S_3 \\ 4 & 3 & 1 & 0 & 0 \\ 4 & 1 & 0 & 1 & 0 \\ 4 & -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \\ 8 \end{bmatrix}$$

Initial Table

		C_j	3	2	0	0	0	
C_B	*Basis*	x_B	x_1	x_2	S_1	S_2	S_3	*Min* $\frac{x_B}{x_1}$
0	S_1	12	4	3	1	0	0	12/4 = 3
0	S_2	8	4	1	0	1	0	8/4 = 2
←0	S_3	8	4	–1	0	0	1	8/4 = 2
	Z_j	0	0	0	0	0	0	
	Z_j – C_j		–3	–2	0	0	0	
			↑					

$Z_1 - C_1$ is most negative, x_1 enters the basis. And the $\min\left(\frac{x_B}{x_{il}},\ x_{il} > 0\right) = \min$ $(3, 2, 2) = 2$ gives S_3 as the leaving variable.

NOTES

Convert the leading element into 1, by dividing key row element by 4 and the remaining elements into 0.

Initial Simplex Table

		C_j	3	2	0	0	0	
C_B	***Basis***	x_B	x_1	x_2	S_1	S_2	S_3	***Min*** $\frac{x_B}{x_2}$
0	S_1	4	0	4	1	0	–1	4/4 = 1
←0	S_2	0	0	2	0	1	–1	0/2 = 0
3	x_1	2	1	–1/4	0	0	¼	—
	Z_j	6	3	–3/4	0	0	¾	
	$Z_j - C_j$		0	–11/4	0	0	¾	
				↑				

$$8-\left(\frac{4}{4}\times 8\right) = 0 \qquad 12-\left(\frac{4}{4}\times 8\right) = 4$$

$$4-\left(\frac{4}{4}\times 4\right) = 0 \qquad 4-\left(\frac{4}{4}\times 4\right) = 0$$

$$1-\left(\frac{4}{4}\times -1\right) = 2 \qquad 3-\left(\frac{4}{4}\times -1\right) = 4$$

$$0-\left(\frac{4}{4}\times 0\right) = 0 \qquad 1-\left(\frac{4}{4}\times 0\right) = 1$$

$$1-\left(\frac{4}{4}\times 0\right) = 1 \qquad 0-\left(\frac{4}{4}\times 0\right) = 0$$

$$0-\left(\frac{4}{4}\times 1\right) = –1 \qquad 0-\left(\frac{4}{4}\times 1\right) = –1$$

Since, $Z_2 - C_2 = -\frac{11}{4}$ is the most negative, x_2 enters the basis.

To find the outgoing variable, find $\text{Min}\left(\frac{x_B}{x_{i2}}, x_{i2} > 0\right)$

$$\text{Min}\left(\frac{4}{4}, \frac{0}{2}, -\right) = 0$$

First Iteration

Therefore, S_2 leaves the basis. Convert the leading element into 1 by dividing the key row elements by 2 and make the remaining elements in that column as zero using the formula.

$$\text{New element} = \text{Old element}$$

$$-\left[\frac{\text{Product of elements in key row and key column}}{\text{Key element}}\right]$$

		C_j	3	2	0	0	0	
C_B	*Basis*	x_B	x_1	x_2	S_1	S_2	S_3	*Min* $\frac{x_B}{S_3}$
←0	S_1	4	0	0	1	–2	1	4/1 = 4
2	x_2	0	0	1	0	1/2	–1/2	—
3	x_1	2	1	0	0	1/8	1/8	$\frac{2}{1/8} = 16$
	Z_j	6	3	2	0	11/8	–5/8	
	$Z_j - C_j$		0	0	0	11/8	–5/8	
								↑

Second Iteration

Since $Z_5 - C_5 = -5/8$ is most negative, S_3 enters the basis and

$$\text{Min}\left(\frac{x_B}{S_{13}}, S_{i3}\right) = \text{Min}\left(\frac{4}{1}, 16\right) = 4.$$

Therefore, S_1 leaves the basis. Convert the leading element into one and remaining elements as zero.

Third Iteration

		C_j	3	2	0	0	0
C_B	*Basis*	x_B	x_1	x_2	S_1	S_2	S_3
0	S_3	4	0	0	1	–2	1
2	x_2	2	0	1	1/2	–1/2	0
3	x_1	3/2	1	0	–1/8	3/8	0
	Z_j	17/2	3	2	5/8	1/8	0
	$Z_j - C_j$		0	0	5/8	1/8	0

Since all $Z_j - C_j \geq 0$, the solution is optimum and it is given by $x_1 = 3/2$, $x_2 = 2$ and Max $Z = 17/2$.

Example 1.3: Using simplex method solve the LPP.

$$\text{Max } Z = x_1 + x_2 + 3x_3$$

Subject to,

$$3x_1 + 2x_2 + x_3 \leq 3$$
$$2x_1 + x_2 + 2x_3 \leq 2$$
$$x_1, x_2, x_3 \geq 0$$

Solution: Rewrite the inequality of the constraints into an equation by adding slack variables.

$$\text{Max } Z = x_1 + x_2 + 3x_3 + 0S_1 + 0S_2$$

Subject to,

$$3x_1 + 2x_2 + x_3 + S_1 \leq 3$$
$$2x_1 + x_2 + 2x_3 + S_2 \leq 2$$

Initial basic feasible solution is,

$$x_1 = x_2 = x_3 = 0$$
$$S_1 = 3,\ S_2 = 2 \text{ and } Z = 0$$

NOTES

$$\begin{bmatrix} x_1 & x_2 & x_3 & S_1 & S_2 \\ 3 & 2 & 1 & 1 & 0 \\ 2 & 1 & 2 & 0 & 1 \\ 1 & 1 & 3 & 0 & 0 \end{bmatrix}$$

		C_j	1	1	3	0	0	
C_B	**Basis**	x_B	x_1	x_2	x_3	S_1	S_2	***Min*** $\frac{x_B}{x_3}$
0	S_1	3	3	2	1	1	0	3/1 = 3
←0	S_2	2	2	1	2	0	1	2/2 = 1
	Z_j	0	0	0	0	0	0	
	$Z_j - C_j$		–1	–1	–3	0	0	
					↑			

Since $Z_3 - C_3 = –3$ is the most negative, the variable x_3 enters the basis. The column corresponding to x_3 is called the *key column*.

To determine the key row or leaving variable, find Min$\left(\frac{x_B}{x_{i3}}, x_{i3} > 0\right)$ Min$\left(\frac{3}{1}, \frac{2}{2}\right) = 1$

Therefore, the leaving variable is the basic variable S_2, the row is called the *key row* and the intersection element 2 is called the *key element*.

Convert this element into one by dividing each element in the key row by 2 and the remaining elements in that key column as zero using the formula

$$\text{New element} = \text{Old element} - \left[\frac{\text{Product of elements in key row and key column}}{\text{Key element}}\right]$$

First Iteration

		C_j	1	1	3	0	0
C_B	**Basis**	x_B	x_1	x_2	x_3	S_1	S_2
0	S_1	2	2	3/2	0	1	–1/2
3	x_3	1	1	1/2	1	0	1/2
	Z_j	3	3	3/2	3	0	3/2
	$Z_j - C_j$		2	1/2	0	0	3/2

Since all $Z_j - C_j \geq 0$, the solution is optimum and it is given by $x_1 = 0$, $x_2 = 0$, $x_3 = 1$, Max $Z = 3$.

Example 1.4: Use simplex method to solve the LPP.

$$\text{Min } Z = x_2 - 3x_3 + 2x_5$$

Subject to,

$$3x_2 - x_3 + 2x_5 \leq 7$$

$$-2x_2 + 4x_3 \leq 12$$

$$-4x_2 + 3x_3 + 8x_5 \leq 10$$

$$x_2, x_3, x_5 \geq 0$$

Solution: Since the given objective function is of minimization, we shall convert it into maximization using Min Z = –Max(–Z) = –Max Z^*

$$\text{Max } Z^* = -x_2 + 3x_3 - 2x_5$$

Subject to,
$$3x_2 - x_3 + 2x_5 \leq 7$$
$$-2x_2 + 4x_3 \leq 12$$
$$-4x_2 + 3x_3 + 8x_5 \leq 10$$

We rewrite the inequality of the constraints into an equation by adding slack variables S_1, S_2, S_3 and the standard form of LPP becomes.

$$\text{Max } Z = -x_2 + 3x_3 - 2x_5 + 0S_1 + 0S_2 + 0S_3$$

Subject to,
$$3x_2 - x_3 + 2x_5 + S_1 = 7$$
$$-2x_2 + 4x_3 + S_2 = 12$$
$$-4x_2 + 3x_3 + 8x_5 + S_3 = 10$$
$$x_2, x_3, x_5, S_1, S_2, S_3 \geq 0$$

∴ The initial basic feasible solution is given by $S_1 = 7$, $S_2 = 12$, $S_3 = 10$. ($x_2 = x_3 = x_5 = 0$)

Initial Table

		C_j	-1	3	-2	0	0	0	
C_B	*Basis*	x_B	x_2	x_3	x_5	S_1	S_2	S_3	*Min* $\frac{x_B}{x_3}$
0	S_1	7	3	–1	2	1	0	0	—
←0	S_2	12	–2	4	0	0	1	0	12/4 = 3
0	S_3	10	–4	3	8	0	0	1	10/3 = 3.33
	Z_j	0	0	0	0	0	0		
	$Z_j - C_j$		1	–3	2	0	0	0	
				↑					

Since $Z_2 - C_2 = -3 < 0$, the solution is not optimum.

The incoming variable is x_3 (key column) and the outgoing variable (key row) is given by,

$$\text{Min}\left(\frac{x_B}{x_{i3}}\, x_{i3} > 0\right) = \text{Min}\left(-, \frac{12}{4}, \frac{10}{3}\right) = 3.$$

Hence, S_2 leaves the basis.

First Iteration

		C_j	-1	3	-2	0	0	0	
C_B	B	x_B	x_2	x_3	x_5	S_1	S_2	S_3	*Min* $\frac{x_B}{x_2}$
$\leftarrow 0$	S_1	10	5/2	0	2	1	1/4	0	$\frac{10}{5/2} = 4$
3	x_3	3	–1/2	1	0	0	1/4	0	—
0	S_3	1	5/2	0	8	0	–3/4	1	2/5
	Z_j	9	–3/2	3	0	0	3/4	0	
	$Z_j - C_j$		–1/2	0	2	0	3/4	0	
			↑						

Since $Z_1 - C_1 < 0$, the solution is not optimum. Improve the solution by allowing the variable x_2 to enter into the basis and the variable S_1 to leave the basis.

Second Iteration

		C_j	-1	3	-2	0	0	0
C_B	B	x_B	x_2	x_3	x_5	S_1	S_2	S_3
–1	x_2	4	1	0	4/5	2/5	1/10	0
3	x_3	5	0	1	2/5	1/5	3/10	0
0	S_3	11	0	0	10	1	– 1/2	1
	Zj	11	– 1	3	2/5	1/5	8/10	0
	$Z_j - C_j$		0	0	12/5	1/5	8/10	0

Since all $Z_j - C_j \geq 0$, the solution is optimum.

$\therefore$ The optimal solution is given by Max $Z^* = 11$

$$x_2 = 4, x_3 = 5, x_5 = 0$$

$$\therefore \quad \text{Min } Z = -\text{Max }(-Z) = -11$$

$$\therefore \quad \text{Min } Z = -11, x_2 = 4, x_3 = 5, x_5 = 0.$$

Example 1.5: Solve the following LPP using simplex method.

$$\text{Max } Z = 15x_1 + 6x_2 + 9x_3 + 2x_4$$

$$\begin{aligned} \text{Subject to,} \quad 2x_1 + x_2 + 5x_3 + 6x_4 &\leq 20 \\ 3x_1 + x_2 + 3x_3 + 25x_4 &\leq 24 \\ 7x_1 + x_4 &\leq 70 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

Solution: Rewriting the inequality of the constraint into an equation by adding slack variables S_1, S_2 and S_3, the standard form of LPP becomes.

$$\text{Max } Z = 15x_1 + 6x_2 + 9x_3 + 2x_4 + 0S_1 + 0S_2 + 0S_3$$

$$\begin{aligned} \text{Subject to,} \quad 2x_1 + x_2 + 5x_3 + 6x_4 + S_1 &= 20 \\ 3x_1 + x_2 + 3x_3 + 25x_4 + S_2 &= 24 \\ 7x_1 + x_4 + S_3 &= 70 \\ x_1, x_2, x_3, x_4, S_1, S_2, S_3 &\geq 0 \end{aligned}$$

The initial basic feasible solution is $S_1 = 20$, $S_2 = 24$, $S_3 = 70$ ($x_1 = x_2 = x_3 = x_4 = 0$ non-basic)

The initial simplex table is given by:

		C_j	15	6	9	2	0	0	0	
C_B	**Basis**	x_B	x_1	x_2	x_3	x_4	S_1	S_2	S_3	*Min* $\frac{x_B}{x_1}$
0	S_1	20	2	1	5	6	1	0	0	20/2 = 10
←0	S_2	24	3	1	3	25	0	1	0	24/3 = 8←
0	S_3	70	7	0	0	1	0	0	1	70/7 = 10
	Z_j	0	0	0	0	0	0	0	0	
	$Z_j - C_j$		–15	–6	–9	–2	0	0	0	
			↑							

∴ As some of $Z_j - C_j \leq 0$ the current basic feasible solution is not optimum. $Z_1 - C_1 = -15$ is the most negative value and hence x_1 enters the basis and the variable S_2 leaves the basis.

First Iteration

		C_j	15	6	9	2	0	0	0	
C_B	**Basis**	x_B	x_1	x_2	x_3	x_4	S_1	S_2	S_3	*Min* $\frac{x_B}{x_2}$
←0	S_1	4	0	1/3	3	– 32/3	1	– 2/3	0	$\frac{4}{1/3} = 12$
15	x_1	8	1	1/3	1	25/3	0	1/3	0	$\frac{8}{1/3} = 24$
0	S_3	14	0	– 7/3	– 7	– 172/3	0	– 7/3	1	—
	Z_j	120	15	5	15	125	0	5	0	
	$Z_j - C_j$		0	– 1	6	123	0	5	0	
				↑						

Since $Z_2 - C_2 = -1 < 0$ the solution is not optimal therefore, x_2 enters the basis and the basic variable S_1 leaves the basis.

Second Iteration

		C_j	–5	6	9	–2	0	0	0
C_B	**B**	x_B	x_1	x_2	x_3	x_4	S_1	S_2	S_3
6	x_2	12	0	1	9	–32	3	–2	0
15	x_1	4	1	0	–2	57/3	–1	1	0
0	S_3	42	0	0	14	–132	7	–7	1
	Z_j	132	15	6	24	93	3	3	0
	$Z_j - C_j$		0	0	15	91	3	3	0

Since all $Z_j - C_j \geq 0$, the solution is optimal and is given by,

Max Z =132, $x_1 = 4$, $x_2 = 12$, $x_3 = 0$, $x_4 = 0$.

NOTES

Check Your Progress

1. How was the concept of operations research started?
2. State one definition of OR.
3. What is linear programming problem?
4. Define the linear programming.
5. What are the fields where linear programing can be used?
6. Define simplex method.

1.5 DUALITY AND SENSITIVITY ANALYSIS

For every given linear programming problem, there is another intimately related linear programming problem referred to as its dual. The duality theorem states that 'For every maximization (or minimization) problem in linear programming, there is a unique similar problem of minimization (or maximization) involving the same date which describes the original problem'. The original problem is referred to as the 'Primal'. The 'Dual' of a dual problem is the primal. Thus the primal and dual problems are replicas of each other. Further, the maximum feasible value of the primal objective function equals to the minimum feasible value of the dual objective function. This means that the solutions of the primal and the dual problems are related which infact yields several advantages.

The transformation of a given primal problem into a dual problem involves the following considerations:

(1) If the objective of the primal is maximization, the objective of the dual is minimization.

(2) The primal has m-constraints while its dual has m-unknowns.

(3) The primal has n-unknowns while its dual has n-constraints.

(4) The n-coefficients of the objective function of primal (C_j) become the n-constant terms (b_i) of its dual.

(5) The m-constant terms of the primal (b_i) become the m-constant terms of the objective function (C_j) of its dual.

(6) The coefficients of the variables of the primal are transformed in their position in the dual. This means that the first column of the coefficients in the primal becomes the first row in the dual, the second column becomes the second row and so on.

(7) The n-variables (X_n) of the primal are replaced by the m new variables (Y_m) of its dual. This change affects the system of restrictions as well as the objective function.

(8) The sign of the inequalities in the set of restrictions of the primal ($\leq$) is reversed in the set of restrictions in its dual ($\geq$). In other words, if the inequalities in the primal are of the type $\leq$, then, they are of $\geq$ type in the dual.

(9) The sign of the inequalities restricting the variable ($\geq X_j$) to non-negative values in the primal is equal to the inequality sign of the new variable ($\geq Y_j$)of its dual.

(10) For writing the dual of the given maximization problem, we should first ensure that all the constraint inequalities are of the $\leq$ type and for writing the dual of the given minimization problem, the constraint inequalities should be of the $\geq$ type. We can see the application of these considerations with the help of given examples.

NOTES

Duality in Linear Programming

Every LPP (called the primal) is associated with another LPP (called its dual). Either of the problem can be considered as primal with the other as dual.

The importance of the duality concept is due to two main reasons:

(*i*) If the primal contains a large number of constraints and a smaller number of variables, the labour of computation can be considerably reduced by converting it in to the dual problem and then solving it. (*ii*) The interpretation of the dual variables from the cost or economic point of view, proves extremely useful in making future decisions in the activities being programmed.

Formulation of Dual Problems

For formulating a dual problem, we first write the problem in the canonical form. The following changes are used in formulating the dual problem:

(1) Change the objective function of maximization in the primal into minimization in the dual, and vice versa.

(2) The variables in the primal should be equal to the constraints in the dual and vice versa.

(3) The cost coefficients $C_1, C_2 \ldots C_n$ in the objective function of the primal should be the RHS constant of the constraints in the dual and vice versa.

(4) In forming the constraints for the dual, we consider the transpose of the body matrix of the primal problem.

(5) The variables in both the problems are non-negative.

(6) If the variable in the primal is unrestricted in sign, then the corresponding constraint in the dual will be an equation, and vice versa.

Definition of the Dual Problem

A dual problem refers to a linear program in which the objective function is a linear combination of *m* values that are the limits in the *m* constraints of the primal problem.

Let the primal problem be:

Max $Z = C_1x_1 + C_2x_2 + \ldots + C_nx_n$

Subject to,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\leq b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\leq b_m \\ x_1, x_2 \cdots x_n &\geq 0 \end{aligned}$$

NOTES

Dual: The dual problem is defined as,

Min $Z' = b_1w_1 + b_2w_2 + ... + b_mw_m$

Subject to,
$$\begin{aligned} a_{11}w_1 + a_{21}w_2 + \ldots + a_{m1}w_m &\geq C_1 \\ a_{12}w_1 + a_{22}w_2 + \ldots + a_{m2}w_m &\geq C_2 \\ &\vdots \\ a_{1n}w_1 + a_{2n}w_2 + \ldots + a_{mn}w_n &\geq C_n \\ w_1, w_2 \cdots w_m &\geq 0 \end{aligned}$$

where $w_1, w_2, w_3 ... w_m$ are called dual variables.

Example 1.6: Write the dual of the primal LP problem given as follows:

Max $Z = x_1 + 2x_2 + x_3$

Subject to,
$$\begin{aligned} 2x_1 + x_2 - x_3 &\leq 2 \\ -2x_1 + x_2 - 5x_3 &\geq -6 \\ 4x_1 + x_2 + x_3 &\leq 6 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

Solution: Since the problem is not in the canonical form, we interchange the inequality of the second constraint.

Max $Z = x_1 + 2x_2 + x_3$

Subject to,
$$\begin{aligned} 2x_1 + x_2 - x_3 &\leq 2 \\ 2x_1 - x_2 + 5x_3 &\leq 6 \\ 4x_1 + x_2 + x_3 &\leq 6 \\ \text{and } x_1, x_2, x_3 &\geq 0 \end{aligned}$$

Dual: Let w_1, w_2, w_3 be the dual variables.

Min $Z' = 2w_1 + 6w_2 + 6w_3$

Subject to,
$$\begin{aligned} 2w_1 + 2w_2 + 4w_3 &\geq 1 \\ +w_1 - w_2 + w_3 &\geq 2 \\ -w_1 + 5w_2 + w_3 &\geq 1 \\ w_1, w_2, w_3 &\geq 0 \end{aligned}$$

Example 1.7: Write the dual of the following LPP.

Min $Z = 2x_2 + 5x_3$

Subject to,
$$\begin{aligned} x_1 + x_2 &\geq 2 \\ 2x_1 + x_2 + 6x_3 &\leq 6 \\ x_1 - x_2 + 3x_3 &= 4 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

Solution: Since the given primal problem is not in the canonical form, we interchange the inequality of the constraint. Also, the third constraint is an equation. This equation can be converted into two inequations.

Min $Z = 0x_1 + 2x_2 + 5x_3$

Subject to,
$$x_1 + x_2 + 0x_3 \geq 2$$
$$-2x_1 - x_2 - 6x_3 \geq 6$$
$$x_1 - x_2 + 3x_3 \leq 4$$
$$x_1 - x_2 + 3x_3 \geq 4$$
$$x_1, x_2, x_3 \geq 0$$

Again, on rearranging the constraint, we have,

Min $Z = 0x_1 + 2x_2 + 5x_3$

Subject to,
$$x_1 + x_2 + 0x_3 \geq 2$$
$$-2x_1 - x_2 - 6x_3 \geq -6$$
$$x_1 - x_2 + 3x_3 \geq 4$$
$$-x_1 + x_2 - 3x_3 \geq -4$$
$$x_1, x_2, x_3 \geq 0$$

Dual: Since there are four constraints in the primal, we have four dual variables, namely w_1, w_2, w_3', w_3''.

Max $Z' = 2w_1 - 6w_2 + 4w_3' - 4w_3''$

Subject to,
$$w_1 - 2w_2 + w_3' - w_3'' \leq 0$$
$$w_1 - w_2 - w_3' + w_3'' \leq 2$$
$$0w_1 - 6w_2 + 3w_3' - 3w_3'' \leq 5$$
$$w_1, w_2, w_3', w_3'' \geq 0$$

Let $w_3 = w_3' - w_3''$

Max $Z' = 2w_1 - 6w_2 + 4(w_3' - w_3'')$

Subject to,
$$w_1 - 2w_2 + (w_3' - w_3'') \leq 0$$
$$w_1 - w_2 - (w_3' - w_3'') \leq 2$$

Finally, we have, $0w_1 - 6w_2 + 3(w_3' - w_3'') \leq 5$

Max $Z' = 2w_1 - 6w_2 + 4w_3$

Subject to,
$$w_1 - 2w_2 + w_3 \leq 0$$
$$w_1 - w_2 - w_3 \leq 2$$
$$0w_1 - 6w_2 + 3w_3 \leq 5$$
$$w_1, w_2 \geq 0, w_3 \text{ is unrestricted.}$$

Example 1.8: Find the dual of the LPP given as follows:

Max $Z = 3x_1 - x_2 + x_3$

Subject to
$$\begin{aligned} 4x_1 - x_2 &\le 8 \\ 8x_1 + x_2 + 3x_3 &\ge 12 \\ 5x_1 - 6x_3 &\le 13 \\ x_1, x_2, x_3 &\ge 0 \end{aligned}$$

Solution: Since the problem is not in the canonical form, we interchange the inequality of the second constraint.

Max $Z = 3x_1 - x_2 + x_3$

Subject to,
$$\begin{aligned} 4x_1 - x_2 + 0x_3 &\le 8 \\ -8x_1 - x_2 - 3x_3 &\le -12 \\ 5x_1 + 0x_2 - 6x_3 &\le 13 \\ x_1, x_2, x_3 &\ge 0 \end{aligned}$$

Max $Z = Cx$

Subject to, $Ax \le B$
$x \ge 0$

$$C = (3 - 1 1) \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} b = \begin{pmatrix} 8 \\ -12 \\ 13 \end{pmatrix}$$

$$A = \begin{pmatrix} 4 & -1 & 0 \\ -8 & -1 & -3 \\ 5 & 0 & -6 \end{pmatrix}$$

Dual: Let w_1, w_2, w_3 be the dual variables. The dual problem is:

Min $Z' = b^T W$

Subject to, $A^T W \ge C^T$ and $W \ge 0$

$$\text{i.e., Min } Z' = (8 \quad -12 \quad 13)\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}$$

$$\text{Subject to, } \begin{pmatrix} 4 & -8 & 5 \\ -1 & -1 & 0 \\ 0 & -3 & 6 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \ge \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

Min $Z_1 = 8w_1 - 12w_2 + 13w_3$

Subject to,
$$\begin{aligned} 4w_1 - 8w_2 + 5w_3 &\ge 3 \\ -w_1 - w_2 + 0w_3 &\ge -1 \\ 0w_1 - 3w_2 + 6w_3 &\ge 1 \\ w_1, w_2, w_3 &\ge 0 \end{aligned}$$

Example 1.9: Give the dual of the problem given as follows:

Max $Z = x + 2y$

Subject to, $2x + 3y \geq 4$

$3x + 4y = 5$

$x \geq 0$ and y unrestricted.

NOTES

Solution: Since the variable y is unrestricted, it can be expressed as $y = y' - y''$, $y', y'' \geq 0$. On reformulating the given problem, we have,

Max $Z = x + 2(y' - y'')$

Subject to, $-2x - 3(y' - y'') \leq -4$

$3x + 4(y' - y'') \leq 5$

$3x + 4(y' - y'') \geq 5$

$x, y', y'' \geq 0$

Since the problem is not in the canonical form, we rearrange the constraints.

Max $Z = x + 2y' - 2y''$

Subject to, $-2x - 3y' + 3y'' \leq -4$

$3x + 4y' - 4y'' \leq 5$

$-3x - 4y' + 4y'' \leq -5$

Dual: Since there are three variables and three constraints, in the dual we have three variables namely, w_1, w_2', w_2''.

Min $Z' = -4w_1 + 5w_2' - 5w_2''$

Subject to, $-2w_1 - 3w_2' - 3w_2'' \geq 1$

$-3w_1 + 4w_2' - 4w_2'' \geq 2$

$3w_1 - 4w_2' + 4w_2'' \geq -2$

$w_1, w_2', w_2'' \geq 0$

Let $w_2 = w_2' - w_2''$, so that the dual variable w_2 is unrestricted in sign. Hence, the dual is:

Min $Z' = -4w_1 + 5(w_2' - w_2'')$

Subject to, $-2w_1 + 3(w_2' - w_2'') \geq 1$

$-3w_1 + 4(w_2' - w'') \geq 2$

$3w_1 + 4(w_2' + w'') \geq -2$

i.e., Min $Z' = -4w_1 + 5w_2$

Subject to, $-2w_1 + 3w_2 \geq 1$

$-3w_1 + 4w_2 \geq 2$

$3w_1 - 4w_2 \geq -2$

NOTES

$w_1 \geq 0$ and w_2 is unrestricted.

i.e., Min $Z' = -4w_1 + 5w_2$

Subject to, $-2w_1 + 3w_2 \geq 1$

$-3w_1 + 4w_2 \geq 2$

$-3w_1 + 4w_2 \geq 2$

i.e., Min $Z' = -4w_1 + 5w_2$

Subject to, $-2w_1 + 3w_2 \geq 1$

$-3w_1 + 4w_2 = 2$, $w_1 \geq 0$ and w_2 is unrestricted.

Example 1.10: Write the dual of the following primal LPP.

Min $Z = 4x_1 + 5x_2 - 3x_3$

Subject to, $x_1 + x_2 + x_3 = 22$

$3x_1 + 5x_2 - 2x_3 \leq 65$

$x_1 + 7x_2 + 4x_3 \geq 120$

$x_1 + x_2 \geq 0$ and x_3 is unrestricted.

Solution: Since the variable x_3 is unrestricted, $x_3 = x_3' - x_3''$. Also, bring the problem into canonical form by rearranging the constraints.

Min $Z = 4x_1 + 5x_2 - 3\,(x_3' - x_3'')$

Subject to, $x_1 + x_2 + (x_3' - x_3'') \leq 22$

$x_1 + x_2 + x_3' - x_3'' \geq 22$

$-3x_1 - 5x_2 + 2\,(x_3' - x_3'') \geq -65$

$x_1 + 7x_2 + 4\,(x_3' - x_3'') \geq 120$

$x_1, x_2, x_3' - x_3'' \geq 0$

Min $Z = 4x_1 + 5x_2 - 3x_3' + 3x_3''$

Subject to, $x_1 + x_2 + x_3' - x_3'' \geq 22$

$-x_1 - x_2 - x_3' + x_3'' \geq -22$

$-3x_1 - 5x_2 + 2x_3' - 2x_3'' \geq -65$

$x_1 + 7x_2 + 4x_3' - 4x_3'' \geq 120$

$x_1, x_2, x_3' - x_3'' \geq 0$

Dual: Since there are four constraints in the primal problem, in the dual there are four variables, namely w_1', w_1'', w_2, w_3 so that the dual is given by:

Max $Z' = 22(w_1' - w_1'') - 65w_2 + 120w_3$

Subject to
$$\begin{aligned} w_1' - w_1'' - 3w_2 + w_3 &\le 4 \\ w_1' - w_1'' - 5w_2 + 7w_3 &\le 5 \\ w_1' - w_1'' + 2w_2 + 4w_3 &\le -3 \\ -w_1' + w_1'' - 2w_2 - 4w_3 &\le 3 \\ w_1', w_1'', w_2, w_3 &\ge 0 \end{aligned}$$

Let $w_1 = w_1' - w_{11}''$, i.e., the variable w_1 is unrestricted.

i.e., Max $Z' = 22(w_1' - w_1'') - 65w_2 + 120w_3$

Subject to,
$$\begin{aligned} w_1' - w_1'' - 3w_2 + w_3 &\le 4 \\ w_1' - w_1'' - 5w_2 + 7w_3 &\le 5 \\ -(w_1' - w_1'') - 2w_2 - 4w_3 &\ge 3 \\ -(w_1' - w_1'') - 2w_2 - 4w_3 &\le 3 \end{aligned}$$

i.e., Max $Z' = 22w_1 - 65w_2 + 120w_3$

Subject to,
$$\begin{aligned} w_1 - 3w_2 + w_3 &\le 4 \\ w_1 - 5w_2 + 7w_3 &\le 5 \\ -w_1 - 2w_2 - 4w_3 &\ge 3 \\ -w_1 - 2w_2 - 4w_3 &\le 3 \end{aligned}$$

Thus, we have,

Min $Z' = 22w_1 - 65w_2 + 120w_3$

Subject to,
$$\begin{aligned} w_1 - 3w_2 + 4w_3 &\le 4 \\ w_1 - 5w_2 + 7w_2 &\le 5 \\ -w_1 - 2w_2 - 4w_3 &= 3 \end{aligned}$$

$w_2, w_3 \ge 0$ and w_1 is unrestricted.

Important Results in Duality

1. The dual of the dual is primal.
2. If one is a maximization problem, then the other is a minimization one.
3. The necessary and sufficient condition for any LPP and its dual to have an optimal solution is that both must have a feasible solution.
4. The fundamental duality theorem states that if either the primal or the dual problem has a definite optimal solution, then the other problem also has a definite optimal solution and the maximum values of the objective function in both the problems are the same, i.e., Max Z = Min Z. The solution of the other problem can be read from the $Z_j - C_j$ row below the columns of slack and surplus variables.
5. The existence theorem states that if either problem has an unbounded solution, then the other problem has no feasible solution.

NOTES

6. Complementary slackness theorem: According to this,
   (*i*) If a primal variable is positive, then the corresponding dual constraint is an equation at the optimum, and vice versa.
   (*ii*) If a primal constraint is a strict inequality, then the corresponding dual variable is zero at the optimum, and vice versa.

Economic Interpretation of Dual Variables

If we interpret our primal LP problem as a classical 'Resource Allocation' problem, then its dual can be interpreted as a 'Resource Valuation' problem. Thus the values of the optimal dual variables have fascinating economic interpretation. The primal problem is used to describe a production problem in which the objective function represents the gain obtained from the production of goods, while the constraints characterize bounds on the production amounts due to the presence of limited resources. The available quantity of each resource is then measured.

Economic Interpretation: Consider the following primal problem:

Maximize $C_1x_1 + \ldots + C_nx_n$

Subject to, All $x_i \geq 0$

$a_{11}x_1 + \ldots + a_{1n}x_n \leq b_1$

$a_{m1}x_1 + \ldots + a_{mn}x_n \leq b_m$

Where,

n = Economic Activities

m = Resources

C_j = Revenue Per Unit of Activity j

b_i = Maximum Availability of Resource i

a_{ij} = Consumption of Resource i Per Unit of Activity j

If $(x_1, \ldots, x_n)$ is optimal for the primal and $(y_1, \ldots, y_m)$ is optimal for the dual, then we can state that:

$$C_1x_1 + \ldots + C_nx_n = b_1y_1 + \ldots + b_ny_m$$

Here,

Left Hand Side = Maximal Revenue

Right Hand Side = Resource *i* (Availability of Resource *i*) × Revenue Per Unit of Resource *i*.

In other words we can say that the value of y_i at optimal is dual price of resource *i*.

1.5.1 Sensitivity Analysis

The term sensitivity analysis, often known as post-optimality analysis refers to the optimal solution of a linear programming problem, formulated using various methods. You will learn how sensitivity analysis helps to solve repeatedly the real problem in a little different form. Generally, these scenarios crop up as an end result of parameter changes due to the involvement of new advanced technologies and the accessibility of well-organized latest information for key (input) parameters or the ' What-If' questions. Thus, sensitivity analysis helps to produce optimal solution of simple perturbations for the key parameters. For optimal solutions, consider the simplex algorithm as in a 'Back Box', which accepts the input key parameters to solve LPP as shown in Figure 1.1.

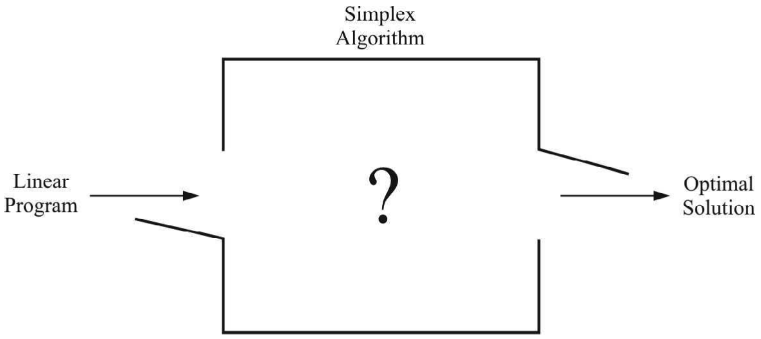


Fig. 1.1 *Black Box to Solve LPP*

Example 1.11: Illustrate sensitivity analysis using simplex method to solve the following LPP:

Maximize $Z = 20x_1 + 10x_2$

Subject to, $x_1 + x_2 \leq 3$

$3x_1 + x_2 \leq 7$

And $x_1, x_2 \geq 0$

Solution: Sensitivity analysis is done after making the initial and final tableau using the simplex method. Add slack variables to convert it into equation form.

Maximize $Z = 20x_1 + 10x_2 + 0S_1 + 0S_2$

Subject to, $x_1 + x_2 + S_1 + 0S_2 + = 3$

$3x_1 + x_2 + 0S_1 + S_2 = 7$

Where $x_1, x_2 \geq 0$

To find basic feasible solution, we put $x_1 = 0$ and $x_2 = 0$. This gives $Z = 0$, $S_1 = 3$ and $S_2 = 7$. The initial table will be as follows:

Initial Table

		C_j	20	10	0	0	
C_B	B	x_B	x_1	x_2	S_1	S_2	Min $\frac{x_B}{x_i}$
0	S_1	3	1	1	1	0	3/1 = 3
←0	S_2	7	③	1	0	1	7/3 = 2.33
	Z_j	0	0	0	0	0	
	$Z_j - C_j$		–20↑	–10	0	0	

Find $\frac{x_B}{x_i}$ for each row and also find minimum for the second row. Here, $Z_j - C_j$ is maximum negative (–20). Hence, x_1 enters the basis and S_2 leaves the basis. It is shown with the help of arrows.

Key element is 3, key row is second row and key column is x_1. Now, convert the key element into entering key by dividing each element of the key row by key element using the following formula:

$$\text{New element} = \text{Old element} - \left[\frac{\text{Product of elements in the key row and key column}}{\text{Key element}}\right]$$

The following is the first iteration table:

		C_j	20	10	0	0	
C_B	B	x_B	x_1	x_2	S_1	S_2	
←0	S_1	2/3	0	2/3	1	–1/3	$\frac{2}{3}/\frac{2}{3}=1$
20	x_1	7/3	1	1/3	0	1	$\frac{7}{3}/\frac{1}{3}=7$
	Z_j	140/3	20	20/3	0	20	
	$Z_j - C_j$	–	0	–10/3↑	0	20	

Since $Z_j - C_j$ has one value less than zero, i.e., negative value, hence, this is not yet the optima solution. Value –10/3 is negative hence x_2 enters the basis and S_1 leaves the basis. Key row is upper row.

		C_j	20	10	0	0
C_B	B	x_B	x_1	X_2	S_1	S_2
10	x_2	1	0	1	3/2	–1/2
20	x_1	4/3	1	0	0	4/3
	Z_j	110/3	20	10	0	25
	$Z_j - C_j$		0	0	0	25

$Z_j - C_j \geq 0$ for all, hence optimal solution is reached, where $x_1 = \frac{4}{3}, x_2 = 1$,

$Z = \frac{80}{3} + 10 = \frac{110}{3}$.

1.5.2 Shadow Price

The price or value of any item is its exchange ratio, which is relative to some standard item. Thus, we may say that shadow price, also known as marginal value, of a constraint *i* is the change it induces in the optimal value of the objective function due to the result of any change in the value, of on the right-hand side of the constraint *i*.

This can be formulated assuming,

z = Objective Function

b_i = Right-Handed Side of Constraint *i*

π^* = Standard Price of Constraint *i*;

At optimal solution,

$z^* = v^* = b^T \pi^*$ (Non-degenerate solution)

Under this situation, the change in the value of z per change of b_i for small changes in b_i is obtained by partially differentiating with objective function z, with respect to the right-handed side b_i, which is further illustrated as:

$$\frac{\delta z}{\delta b_i} = \pi_i^*$$

where,

π_i^* = price associated with the right-handed side.

It is this price which was interpreted by Paul Sammelson as shadow price.

1.5.3 Economic Interpretation

We have often seen that shadow prices are being frequently used in the economic interpretation of the data in linear programming.

Example 1.12: To find the economic interpretation of shadow price under non-degeneracy, you will need to consider the linear programming to find out minimum of objective function $z, x \geq 0$, which is as follows:

$$-x_1 - 2x_2 - 3x_3 + x_4 = z$$
$$x_1 + 2x_4 = 6$$
$$x_2 + 3x_4 = 2$$
$$x_3 - x_4 = 1$$

Now, to get an optimal basic solution, we can calculate the numericals:

$x_1 = 6,\ x_2 = 2,\ x_3 = 1,\ x_4 = 0,\ z = -13.$

The optimal solution for the shadow price is:

$\pi_1^\circ = -1,\ \pi_2^\circ = -2,\ \pi_3^\circ = -3,$

as, $z = b_1\, \pi_1 + b_2\, \pi_2 + b_3\, \pi_3$, where $b = (6, 2, 1)$;

it denotes,

$$\frac{\partial z}{\partial b_1} = \pi_1 = -1,\quad \frac{\partial z}{\partial b_2} = \pi_2 = -2,\quad \frac{\partial z}{\partial b_3} = \pi_3 = -3.$$

As these shadow prices and the changes take place in a non-degenerate situation, they do not impact the small changes of b_i. Now, if this same situation is repeated in a degenerate situation, we will have to replace $b_3 = 1$ by $b_3 = 0$; thereby $\partial z/\partial b_3^+ = -3$, only if the change in b_3 is positive. However, we need to keep in mind that if, b_3 is negative, then x_3 will drop out of the basis and x_4 transcends as the basic and the shadow price may be illustrated as:

$\pi^\circ{}_1 = -1,\ \pi^\circ{}_2 = -2,\ \pi^\circ{}_3 = \partial z/\partial b_3^- = -9$

Here, we see that the interpretation of the dual variables π, and dual objective function υ corresponds to column j of the primal problem. So, the goal of linear programming (Simplex method) is to determine whether there is a basic feasibility for optimal solution, in the most cost-effective manner. Thus, at iteration t is the total cost of the objective function and this can be illustrated as:

$$\upsilon = \pi^T b = \sum_{i=1}^{m} \pi_i \, b_i$$

here, π_i = simplex multipliers which is associated with the basis B.

So, we may say that the prices of the problem of the dual variables are selected in such a manner, that there is maximization of the implicit indirect costs of the resources that are consumed by all the activities. Whenever any basic activity is conducted, it is done at a positive level and all non-basic activities are kept at a zero level.

Hence, if the primal—dual variable system-is utilized, then the slack variable is maintained at a positive level in an optimal solution and the corresponding dual variable is equal to zero.

1.6 DUAL SIMPLEX METHOD

The dual simplex method is very similar to the regular simplex method, the only difference lies in the criterion used for selecting a variable to enter the basis and to leave the basis. In the dual simplex method, we first select the variable to leave the basis and then the variable to enter the basis. This method yields an optimal solution to the given LPP in a finite number of steps, provided no basis is repeated.

The dual simplex method is used to solve problems that start as dual feasible (i.e., whose primal is optimal but infeasible). In this method, the solution starts as optimum, but infeasible, and remains infeasible until the true optimum is reached, at which point the solution becomes feasible. The advantage of this method is that it avoids the artificial variables introduced in the constraints along with the surplus variables, as all '≥' constraints are converted into the '≤' type.

Dual Simplex Algorithm

The iterative procedure for the dual simplex method is listed as follows:

Step 1: Convert the problem into the maximization form if it is initially in the minimization form.

Step 2: Convert the '≥' type constraints, if any, to '≤' types by multiplying both sides by –1.

Step 3: Express the problem in the standard form by introducing slack variables. Obtain the initial basic solution and display this solution in the simplex table.

Step 4: Test the nature of $Z_j - C_j$ (optimal condition).

Case i: If all $Z_j - C_j \geq 0$ and all $x_{Bi} \geq 0$, then the current solution is an optimum feasible solution.

Case ii: If all $Z_j - C_j \geq 0$ and at least one $x_{Bi} < 0$, then the current solution is not the optimum basic feasible solution. In this case, go to the next step.

Case iii: If any $Z_j - C_j < 0$, then the method fails.

Step 5: In this step, we find the leaving variable, which is the basic variable corresponding to the most negative value of x_{Bi}. Let x_k be the leaving variable, i.e., $x_{BK} = \min\{x_{Bi}, x = 0\}$.

To find out the variable entering the basis, we compute the ratio between the $Z_j - C_j$ row and the key row, i.e., we compute Max $\{Z_j - C_j/c_{ik}, a_{ik} < 0\}$. (Consider the ratios with negative denominators alone). The entering variable is the one having the maximum ratio. If there is no such ratio with a negative denominator, then the problem does not have a feasible solution.

NOTES

Step 6 Convert the leading element into unity and all the other elements of the key column into zero to get an improved solution.

Step 7 Repeat Steps 4 and 5 until either an optimum basic feasible solution is attained or till an indication of no feasible solution is obtained.

Example 1.13: Use the dual simplex method to solve the following LPP.

$$\text{Max } Z = -3x_1 - x_2$$

Subject to,
$$x_1 + x_2 \geq 1$$
$$2x_1 + 3x_2 \geq 2$$
$$x_1, x_2 \geq 0$$

Solution: Convert the given constraints into < type.

$$\text{Max } Z = -3x_1 - x_2$$

Subject to,
$$-x_1 - x_2 \leq -1$$
$$-2x_1 - 3x_2 \leq -2$$
$$x_1, x_2 \geq 0$$

Introducing slack variables S_1 and $S_2 \leq 0$, we get:

$$\text{Max } Z = -3x_1 - x_2 + 0S_1, + 0S_2$$

Subject to,
$$-x_1 - x_2 + S_1 \leq -1$$
$$-2x_1 - 3x_2 + S_2 \leq -2$$
$$x_1, x_2, S_1, S_2 \geq 0$$

An initial basic (infeasible) solution of the modified LPP is $S_1 = -1, S_2 = -2$.

		C_j	-3	-1	0	0
C_B	B	X_B	W_1	W_2	S_1	S_2
0	S_1	-1	-1	-1	1	0
$\leftarrow 0$	S_2	-2	-2	$\boxed{-3}$	0	1
	Z_j	0	0	0	0	0
	$Z_j - C_j$		3	$1\uparrow$	0	0

Since all $Z_j - C_j \geq 0$ and all $x_{Bi} < 0$, the current solution is not an optimum basic feasible solution.

Since $x_{B2} = -2$ is the most negative, the corresponding basic variable S_2 leaves the basis. Also, since max $\{Z_j - C_j/a_{ik}, a_{ik} < 0\}$, where x_k is the leaving variable max $\{3/-2, 1/-3\} = -1/3 = Z_2 - C_2/a_{22}$, the non-basic variable x_2 enters the basis.

NOTES

Drop S_2 and introduce x_2.

First Iteration

		C_j	-3	-1	0	0
C_B	B	X_B	X_1	X_2	S_1	S_2
$\leftarrow 0$	S_1	$-1/3$	$-1/3$	0	1	$\boxed{-1/3}$
-1	X_2	$2/3$	$2/3$	1	0	$-1/3$
	Z_j	$-2/3$	$-2/3$	-1	0	$1/3$
	Z_j-C_j		$7/3$	0	0	$1/3\uparrow$

Since all $Z_j - C_j \geq 0$ and $x_{B1} = -1/3 < 0$, the current solution is not an optimum basic feasible solution.

$\because$ $x_{B1} = -1/3$, the basic variable S leaves the basis. Also, since max $\{Z_j - C_j/a_{i1}, a_{i1} < 0\} = \max \{(1/3)/(-1/3)...(1/3)/(-1/3)\} = -1$, it corresponds to the non-basic variable S_2

$\therefore$ Drop S_1 and introduce S_2.

Second Iteration

		C_j	-3	-1	0	0
C_B	B	x_B	x_1	x_2	S_1	S_2
0	S_2	1	1	0	-3	1
-1	x_2	1	1	1	-1	0
	Z_j	-1	-1	-1	1	0
	Z_j-C_j		2	0	1	0

Since all $Z_j - C_j \leq 0$ and also $x_{Bi} \geq 0$, an optimum basic feasible solution has been reached. The optimal solution to the given LPP is $x_1 = 0$; $x_2 = 1$, Maximum $Z = -1$.

Example 1.14: Solve by the dual simplex method the following LPP.

Min $Z = 5x_1 + 6x_2$

Subject to,
$$x_1 + x_2 \geq 2$$
$$4x_1 + x_2 \geq 4$$
$$x_1,\ x_2 \geq 0$$

Solution: The given LPP is Max Z = $5x_1 - 6x_2$

Subject to,
$$-x_1 - x_2 \leq -2$$
$$-4x_1 - x_2 \leq -4 \text{ and}$$
$$x_1,\ x_2 \geq 0$$

By introducing slack variables S_1 and S_2, the standard form of LPP becomes,

Max $Z = -5x_1 - 6x_2 + 0S_1 + 0S_2$

Subject to, $- x_1 - x_2 + S_1 = -2$

$- 4x_1 - x_2 + S_2 = -4$

Initial Table

		C_j	–5	–6	0	0
C_B	B	X_B	X_1	X_2	S_1	S_2
0	S_1	– 2	– 1	– 1	1	0
←0	S_2	– 4	$\boxed{-4}$	– 1	0	1
	Z_j	0	0	0	0	0
	Z_j–C_j		5↑	6	0	0

Since all $Z_j - C_j \geq 0$ and $x_{Bi} \leq 0$, the current solution is not an optimum basic feasible solution.

Since $x_{B2} = -4$ is the most negative, the corresponding basic variable S_2 leaves the basis.

Also, Max $Z_j - C_j/a_{i2}a_{\ i2} < 0$ = Max {– 5/4, 6/ –1,...} = –5/4 gives the non-basic variable, and x_1 enters into the basis.

First Iteration

		C_j	–5	–6	0	0
C_B	B	X_B	X_1	X_2	S_1	S_2
←0	S_1	– 1	– 1	– 3/4	1	$\boxed{-1/4}$
–5	X_1	1	– 4	1/4	0	– 1/4
	Z_j	– 5	– 5	– 5/4	0	5/4
	Z_j–C_j		0	19/4	0	5/4↑

Since all $Z_j - C_j \geq 0$ and also $x_{B1} = -1 < 0$, the current basic feasible solution is not optimum. As $x_{B1} = -1 < 0$, the basic variable S_1 leaves the basis.

Also, since Max $\left\{\dfrac{Z_j - C_j}{a_{i1}}, a_{i1} < 0\right\}$ = Max $\left\{\dfrac{19/4}{-3/4}, \dfrac{5/4}{-1/4}\right\} = 5/4$, it corresponds to the non-basic variable S_2.

∴ Drop S_1 and introduce S_2.

Second Iteration

		C_j	–5	–6	0	0
C_B	B	X_B	X_1	X_2	S_1	S_2
0	S_2	4	0	3	– 4	1
–5	X_1	1	– 4	1	– 1	0
	Z_j	– 10	– 5	– 5	5	0
	Z_j–C_j		0	1	5	0

Since all $Z_j - C_j \geq 0$ and also all $x_{Bi} \geq 0$, the current basic feasible solution is optimum. The optimal solution is given by $x_1 = 2, x_2 = 0$, Max $Z = -10$

i.e., Min $Z = 10$.

Example 1.15: Use the dual simplex method to solve the following LPP.

$$\text{Max } Z = -3x_1 - 2x_2$$

Subject to,

$$x_1 + x_2 \geq 1$$
$$x_1 + x_2 \leq 1$$
$$x_1 + 2x_2 \geq 10$$
$$x_2 \leq 3$$
$$x_1, x_2 \geq 0$$

Solution: Interchanging the > inequality of the constraints into <, the given LPP becomes,

$$\text{Max } Z = -3x_1 - 2x_2$$

Subject to,

$$-x_1 - x_2 \leq -1$$
$$x_1 + x_2 \leq 7$$
$$-x_1 - 2x_2 \leq -10$$
$$0x_1 + x_2 \leq 3$$

By introducing the non-negative slack variables S_1, S_2, S_3 and S_4, the standard form of the LPP becomes,

$$\text{Max } Z = -3x_1 - 2x_2 + 0S_1 + 0S_2 + 0S_3 + 0S_4$$

Subject to,

$$-x_1 - x_2 + S_1 = -1$$
$$x_1 + x_2 + S_2 = 1$$
$$-x_1 - 2x_2 + S_3 = -10$$
$$0x_1 + x_2 + S_4 = 3$$

The initial solution is given by:

$S_1 = -1,\ S_2 = 7,\ S_3 = -10,\ S_4 = 3$

Initial Table

		C_j	-3	-2	0	0	0	0
C_B	B	X_B	X_1	X_2	S_1	S_2	S_3	S_4
0	S_1	-1	1	1	1	0	0	0
0	S_2	7	1	1	0	1	0	0
$\leftarrow 0$	S_3	-10	-1	$\boxed{-2}$	0	0	1	0
0	S_4	3	0	1	0	0	0	1
	Z_j	0	0	0	0	0	0	0
	$Z_j - C_j$		3	$2\uparrow$	0	0	0	0

NOTES

Since all $Z_j - C_j \geq 0$ and some $x_{Bi} \leq 0$, the current solution is not the main workable solution. $= x_{B3} = -10$ being the most negative, the basic variable S_3 leaves the basis.

Also, since Max $\{Z_j - C_j/a_{i2}, a_{i2} < 0\}$ = Max $\{3/{-1}, 2/{-2}\}$ = -1, the non-basic variable x_2 enters the basis.

First Iteration

		C_j	-3	-2	0	0	0	0
C_B	B	X_B	X_1	X_2	S_1	S_2	S_3	S_4
0	S_1	4	$-1/2$	0	1	0	$-1/2$	0
0	S_2	2	$1/2$	0	0	1	$1/2$	0
-2	X_2	5	$1/2$	1	0	0	$-1/2$	0
$\leftarrow 0$	S_4	-2	$\boxed{-1/2}$	0	0	0	$1/2$	1
	Z_j	-10	-1	-2	0	0	1	0
	$Z_j - C_j$		$2\uparrow$	0	0	0	1	0

Second Iteration

Drop S_4 and introduce x_1.

$\because x_{B4} = -2 < 0$, S_4 leaves the basis.

$$\text{Max}\left\{\frac{Z_j - C_j}{a_{1i}} a_{1i} < 0\right\} = \text{Max}\left\{\frac{2}{-1/2}\cdots\right\} = -4$$

Hence, x_1 enters the basis.

		C_j	3	2	0	0	0	0
C_B	B	X_B	X_1	X_2	S_1	S_2	S_3	S_4
0	S_1	2	0	0	1	0	-1	-1
0	S_2	0	0	0	0	1	1	1
-2	X_2	3	0	1	0	0	0	1
-3	S_4	4	1	0	0	0	-1	-2
	Z_j	-18	-3	-2	0	0	3	4
	$Z_j - C_j$		0	0	0	0	3	4

Since all $Z_j - C_j \geq 0$ and all $x_{Bi} \geq 0$, the current solution is an optimum basic feasible solution.

$\therefore$ The optimum solution is Max $Z = -18$, $x_1 = 4$, $x_2 = 3$.

Example 1.16: Use dual simplex method to solve the following LPP.

Max $Z = -2x_1 - x_3$

Subject to, $x_1 + x_2 - x_3 \geq 5$

$x_1 - 2x_2 + 4x_3 \geq 8$

$x_1, x_2, x_3 \geq 0$

NOTES

Solution: The given problem can be written as:

$$\text{Max } Z = -2x_1 - 0x_2 - x_3$$

Subject to, $-x_1 - x_2 + x_3 \leq -5$

$$-x_1 + 2x_2 - 4x_3 \leq -8$$

$$x_1,\ x_2,\ x_3 \geq 0$$

Adding the slack variables S_1 and S_2, we get the constraints,

$$-x_1 - x_2 + x_3 + S_1 = -5$$

$$-x_1 + 2x_2 - 4x_3 + S_2 = -8$$

		C_j	-2	0	-1	0	0
C_B	B	X_B	X_1	X_2	X_3	S_1	S_2
0	S_1	-5	-1	-1	1	1	0
←0	S_2	-8	-1	2	$\boxed{-4}$	0	1
	Z_j	0	0	0	0	0	0
	$Z_j - C_j$		2	0	1↑	0	0

Since all $Z_j - C_j \geq 0$ and also some $x_{Bi} \geq 0$, the solution is not optimum.

$\because x_{B2} = -8$ is the most negative, the basic variable S_2 leaves the basis. Also, since

$$\text{Max}\left\{\frac{Z_j - C_j}{a_{iK}}, a_{ik} < 0\right\} \text{ gives}$$

$$\text{Max}\left\{\frac{2}{-1}, -, \frac{1}{-4}, --\right\} = -\frac{1}{4}$$

x_3 enters the basis. Drop S_2 and introduce x_3.

First Interation

		C_j	-2	0	-1	0	0
C_B	B	X_B	X_1	X_2	X_3	S_1	S_2
←0	S_1	-7	$-5/4$	$-1/2$	1	1	$1/4$
-1	X_3	2	$1/4$	$-1/2$	0	0	$-1/4$
	Z_j	-2	$-1/4$	$1/2$	-1	0	$1/4$
	$Z_j - C_j$		$7/4$	$1/2$↑	0	0	$1/4$

Drop S_1 and introduce x_2.

Since $x_{B1} = -7 < 0$, S_1 the basic variable leaves the basis.

Therefore, the non-basic variable x_2 enters the basis.

$$\text{Max }\left\{\frac{Z_j - C_j}{a_{iK}}, a_{ik} < 0\right\} \text{ gives}$$

$$\text{Max}\left\{\frac{7/4}{-5/4},\frac{1/2}{-1/2},--\right\} = -1$$

Second Iteration

		C_j	-2	0	-1	0	0
C_B	B	X_B	X_1	X_2	X_3	S_1	S_2
0	S_2	14	5/2	1	0	– 2	– 1/2
– 1	X_3	9	3/2	0	1	– 1	– 1/1
	Z_j	– 9	– 3/2	0	– 1	1	1/2
	$Z_j - C_j$		1/2	0	0	1	1

Since all $Z_j - C_j \geq 0$ and all $x_{Bi} \geq 0$, the current feasible solution is optimum. The optimal solution is given by $x_1 = 0, x_2 = 14, x_3 = 9$.

$$\text{Max } Z = -9.$$

1.7 PARAMETRIC LINEAR PROGRAMMING

Sensitivity analysis discussed the effect of discrete changes in the input coefficients of the linear programming problem on its optimal solution. Further, these input coefficients (c_j or b_i or a_{ij}) were changed one at a time. However, if there is continuous change in the values of the coefficients or if simultaneous variations (changes in all components of c_j or b_i or a_{ij}) occur in their values.

Parametric linear programming investigates the effect of predetermined continuous variations of these coefficients on the optimal solution. It is simply an extension of sensitivity analysis and aims at finding the various basic solutions that become optimal, one after the other, as the coefficients of the problem change continuously. The coefficients change as a linear function of a single parameter, hence the name parametric linear programming for this computational technique. As in sensitivity analysis, the purpose of this technique is to reduce the additional computations required to obtain the changes in the optimal solution. The various types of parametric problems that one may come across are:

1. Parametric Cost Problem: In which the cost coefficients c_j vary linearly as a function of parameter λ.
2. Parametric Right-Hand Side Problem: In which the resources availability coefficients b_i vary linearly as a function of parameter λ.
3. Parametric problem involving linear variations in the non-basic vector P_j of A.
4. Parametric problem involving simultaneous linear variations in c_j, b_i and P_j.

The parametric right-hand side linear programming (parametric RHS LP) problem as follows:

$$(P_\lambda)\ \min\{c^T x | Ax = b + \lambda \bar{b}\ , x \geq 0|\},$$

NOTES

where A is an m x n matrix and b, $\overline{b}$, and c are vectors of dimensions m, m, and n, respectively. The parametric RHS LP problem (P_λ), $\lambda \in \mathbf{R}$, consists of solving each Linear Programming (LP) problem (P_λ) for all values of $\lambda \in \mathbf{R}$ (or for λ in a certain required nterval). If (P_λ) denotes the optimal value of (P_λ) it is well known that the function $\lambda \in \mathrm{R} \rightarrow \varphi(\lambda)$ is a convex piecewise linear continuous function. In view of this property, only a finite amount of information is necessary to solve the parametric RHS LP problem. Basically, it consists of finding the 'Breakpoints' of $\varphi(\lambda)$ and an optimal solution of (P_λ) for all breakpoints A.

We present a way of approaching this problem which differs from the usual method based on the simplex method. Our main motivation to look back into this problem was the introduction of new methods for solving LP problems like the ellipsoid method introduced by Khachiyan and the new interior point algorithm presented by Karmarkar.

The existing method to solve this problem is the parametric RHS LP simplex method which was first discussed by Gass and Saaty a few years after the simplex method was developed by Dantzig. The theory of sensitivity and parametric analysis both in discrete and continuous linear (and nonlinear) optimization has been the subject of intensive research.

Both the existing theory of sensitivity and parametric analysis depends crucially on the concept of the optimality (or characteristic) interval associated with an optimal basis, that is, the set of values of λ. for which this basis is optimal for the LP problem (P_λ)

Definition of optimality intervals and derive an algorithm for solving the parametric RHS LP problem which can be implemented with the aid of any LP solver. As a first step we have to get rid of the concept of basis and introduce another invariant associated with the problem in order to define our optimality intervals. This is done by considering those partitions $(B,$ N), which we call optimal partitions, such that, $B \cup N = \{1, .. ., n\}$, $B \cap \ N = \varnothing$ and $(x_j \geq 0, j \in N)$ and $(A_j^T y \leq c_j, j \in B)$ are, respectively, the set of always-active constraints with respect to the primal optimal face and the dual optimal face of some problem (P_λ). We then show that an optimality interval is either an open interval between two consecutive breakpoints of $\varphi(\lambda)$ or consists of a breakpoint itself. This shows that the real line is covered in a unique way using these optimality intervals. This is in contrast with the basis optimality intervals where even the closed interval determined by two consecutive breakpoints of $\varphi(\lambda)$ can be covered in many ways with possibly an exponential number of these intervals.

The second step is to provide an algorithm for the parametric RHS LP problem, based on any LP solver, that computes a sequence of optimal partitions and their associated optimality intervals so that at the end we have covered the required interval by these optimality intervals. The approach is the same as in the existing pivot method which successively finds adjacent optimality intervals either going to the left or to the right of the real line. However, our approach solves an LP problem to find the adjacent partition and the corresponding optimality interval.

It is well known that the parametric RHS LP problem cannot be solved in polynomial time due the existence of instances of the problem whose corresponding function $\varphi(\lambda)$ exhibits an exponential number of breakpoints. One of the main consequences of our algorithm is an affirmative answer to the following related computational complexity issue: Can the parametric RHS LP problem be solved in time polynomially bounded by the size of the input and the number of breakpoints of $\varphi(\lambda)$? For nondegenerate problems, the answer to this issue is rather trivial and is provided by the parametric RHS simplex method discussed above. The parametric RHS LP problem can be solved in *O(kZ)* where *k* is the number of breakpoints and *Z* is the complexity of solving a single LP problem of the same dimension.

Problem Description and Some Theoretical Background

In this section we introduce the class of problems with which this paper is concerned. We also review some results pertinent to the present work.

1. ***Basic Notations*: $\mathbf{R}^n$, $\mathbf{R}^n_+$, $\mathbf{R}^n_{++}$** denotes the sets of *n*-vectors with reals components, nonnegative reals components, and (strictly) positive reals components, respectively. Let *A* be an *m* x *n* matrix. Given a subset *B* of the index set $\{1,\ldots,n\}$, we denote by *An* the submatrix of *A* associated with the index set *B*. Also, the subspace generated by the columns of *A* is denoted by *range(A)*. The *i*th row and *j*th column of *A* is denoted by A_i and A_j respectively. A closed interval in **R** with extreme points α and β is denoted by $[\alpha,\beta]$ even when either $\alpha = -\infty$ or $\beta = \infty$. The closure of a set $X \subseteq \mathbf{R}^n$ is denoted by cl *X*. $\bar{\mathbf{R}}$ denotes the set of extended reals, that is, $\bar{\mathbf{R}} = \mathbf{R} \cup \{-\infty, \infty\}$.
2. ***True Inequalities*:** Consider a polyhedron $Q \subseteq \mathbf{R}^n$, that is, $Q = \{x \in \mathbf{R}^n | Ax \le b;\ Cx = d\}$. The system of linear constraints $Ax \le b$, $Cx = d$ is then said to be a representation of *Q*. A subset *F* of *Q* is a face of *Q* if either $F = \varnothing$ or *F* is the set of optimal solutions for min$\{c^T x | x \in Q\}$ for some $c \in \mathbf{R}^n$. We say that an inequality $ax \le \beta$ of the system $Ax \le b$ is a *true inequality for the face F if* $ax < \beta$ *for some* $x \in F$. When the face *F* is the whole polyhedron $\{x | Ax \le b, Cx = d\}$, the set of true inequalities for the face *F* is referred to simply as the set of *true inequalities for the system* $Ax \le b$, $Cx = d$. Let $A'x \le b'$ be the true inequalities for the face *F* and $A''x \le b''$ be the other inequalities from $Ax \le b$.

$$F = \{x \in \mathbf{R}^n | A'x \le b';\ A''x = b'';\ Cx = d\}. \qquad \text{...(2.1)}$$

The following results can be obtained from Equation (1.1) and as per the definition of true inequalities.

Note 1: Let *F* and *F'* be two faces of the polyhedron *Q*. Then $F \subseteq F'$ if and only if every true inequality for *F* is also a true inequality for *F'*.

The following result can be easily proved.

Result 1: *Let the face F and the matrices A', A'', and C be as above and let* $f \in \mathrm{R}^n$ *be given. Then the linear function* $x \rightarrow f^T x$ *is constant on F if and only if* $f \in range[(A'')^T, C^T]$.

NOTES

3. *Optimal Partitions:* Consider the LP problem in standard form

(P) $= \min\{c^{T}x \mid Ax = b;\ x \geq 0\}$ and its dual

(D) $= \max\{b^{T}y \mid A^{T}y \leq c\}$,

where A is an $m \times n$ matrix, c is an n-vector, and b is an m-vector. Let X^* and Y^* denote the set of optimal solutions of problems *(P)* and *(D),* respectively. Assume that X^* (and consequently Y^**)* is nonempty. Clearly, X^* (respectively. Y^**)* is a face of the polyhedron of feasible solutions for problem *(P)* (respectively. *(D)).* Let the inequalities. $x_j \geq 0$ with $j \in B \subseteq \{1, \ldots, n\}$ and the inequalities $A^{T}_{j}y \leq ci$ with $j \in N \subseteq \{1, \ldots, n\}$ be the set of true inequalities for the faces X^* and Y^*, respectively. This is equivalent to saying that,

$$B = \{j \mid x_j > 0 \text{ for some } x \in X^* \text{ and } j = 1, \ldots, n\}, \qquad \text{...(1.2)}$$

$$N = \{j \mid c_j - A_j^{T}y > 0 \text{ for some } y \in Y^* \text{ and } j = 1, \ldots, n\}. \qquad \text{...(1.3)}$$

We then have the following well-known result.

Result 2: *Assume that* X^* *is nonempty and let* $B \subseteq \{1, \ldots, n\}$ *and* $N \subseteq \{1, \ldots, n\}$ *be as in* (1.2) *and* (1.3). *Then* $B \cap N = \phi$ *and* $B \cup N = \{1, \ldots, n\}$.

The pair of index sets *(B, N)* then determines a partition of the index set $\{1, \ldots, n\}$. We refer to *(B, N)* as the optimal partition associated with problem *(P).* In terms of the partition *(B, N)*, the optimal faces X^* and Y^* can be written as,

$$X^* = \{x \mid A_B x_B = b;\ x_B\ 2\ 0;\ x_N = 0\},$$

$$Y^* = \{y \mid \boldsymbol{A}^{T}{}_{B}\, y = c_{B};\ Ai\, y \qquad c_N\} \cdot \qquad \text{...(1.4)}$$

Description of the Parametric RHS LP Problem and Related Preliminary Results. In this subsection we introduce the problem which is the object of our analysis in this paper.

Consider the parametrized family of LP problems in standard form

(P_λ) $\min\{c^{T}x \mid Ax = b + \lambda\bar{b};\ x \geq 0\}$, and the corresponding parametrized family of dual problems

$(D) = \max\{(b + \bar{b})^{T}y \mid A^{T}y \leq c\}$,

where b, $\bar{b}$ are m-vectors, c is an n-vector, A is an m x n matrix, and $\lambda \in$ R.

Solving (P_λ) for all $\lambda \in$ R is known as the parametric RHS LP problem. We denote the optimal value of (P_λ) by) $\varphi(\lambda)$ with the convention that $\varphi(\lambda) = \infty$ if (P_λ) is infeasible and $\varphi(\lambda) = -\infty$ if (P_λ) is feasible and unbounded. With this convention, we then obtain an extended convex function $\varphi: \mathbf{R} \to \bar{\mathbf{R}}$, that is, a function taking values on $\bar{\mathbf{R}}$ and whose epigraph epi $\varphi \equiv \{(x, \theta) \in \mathbf{R}^n \times \mathbf{R} \mid \theta \geq \varphi(x)\}$ is a convex set (Refer Result 3 below).

The next proposition characterizes the 'Shape' of the function φ.

Result 3: *There exists a closed interval* [α, β] *(possibly empty) such that:*

(a) $\varphi(\lambda) = \infty$ *for all* $\lambda \notin [\alpha, \beta]$.

(b) *Either* $\varphi(\lambda) = -\infty$ *for all* $\lambda \in [\alpha, \beta]$ *or* $\varphi\ [\alpha, \beta] \subseteq \mathbf{R}$ *and in this case* $\varphi\ [\alpha, \beta]$ *is a continuous convex piecewise linear function.*

The cases in which either [α, β] is empty or $\varphi(\lambda) = -\infty$ for all $\lambda \in [\alpha, \beta]$ present no difficulty to our analysis. Henceforth, we make the following assumption.

Theory 1.1. [α, β] *is nonempty and* $\varphi(\lambda) \in \mathbf{R}$ for all $\lambda \in [\alpha, \beta]$.

Proof: There exists a finite set of points $\alpha = \lambda_0 < \lambda_1 < \cdots < \lambda_k = \beta$ and real constants g_i, h_i, $i \in \{1, \ldots, k\}$, such that, $\varphi(\lambda) = g_i \lambda + hi$ for all $\lambda \in [\lambda_{i-1}, \lambda_i]$. The convexity of φ implies that $g_1 < g_2 < \cdots < g_k$. Obviously, φ has left and right derivatives for all $\lambda \in [\alpha, \beta]$ Indeed, for all $i \in \{1, \ldots, k\}$, $\varphi'(\lambda) = g_i$ if $\lambda \in (\lambda_{i-1}, \lambda_i)$, $\varphi'_+(\lambda_{i-1}) = g_i$, and $\varphi'_-(\lambda_i) = g_i$ By convention, if α (respectively, β) is finite, we let

$\varphi'_-(\alpha) = -\infty$ (resp. $\varphi'_+(\beta) = \infty$).

Throughout this paper we let $X(\lambda)$ and $Y(\lambda)$, where $\lambda \in [\alpha, \beta]$, denote the primal and dual optimal faces for problems (P_λ) and (D_λ), respectively. Also ($B(\lambda)$, N(λ)) denotes the optimal partition associated with problem (P_λ) where $\lambda \in [\alpha, \beta]$.

The next proposition expresses the left and right derivatives of the function φ in terms of certain LP problems.

Result 4. *For any* $\lambda \in [\alpha, \beta]$, *the left and right derivatives* $\varphi'_-(\lambda)$ *and* $\varphi'_+(\lambda)$ *are given by*

$$\varphi'_-(\lambda) = \min\{\bar{b}^T y \mid y \in Y(\lambda)\},$$
$$\varphi'_+(\lambda) = \max\{\bar{b}^T y \mid y \in Y(\lambda)\},$$

Characterization of Optimality Sets of Optimal Partitions

The existing theory of sensitivity and parametric analysis depends crucially on the concept of the optimality (or characteristic) interval associated with an optimal basis, that is, a basis which is primal and dual feasible for some problem (P_λ) where $\lambda \in [\alpha,\beta]$ In this case, the optimality set of an optimal basis is defined to be the set of all $\lambda \in [\alpha,\beta]$ for which such basis is optimal for the LP problem (P_λ). Hence, this theory studies the invariance of the optimality of a basis with respect to change on the parameter *λ*. Instead, our analysis is based on the optimality set of an optimal partition. In this section we introduce and characterize the optimality set of an optimal partition. We also present some results relating 'Adjacent' optimal partitions.

We start by defining the optimality set of an optimal partition. Let $(B, N) \equiv (B(\lambda^*), N(\lambda^*))$ be an optimal partition associated with (P_{λ^*}) where *)* $\lambda^* \in [\alpha, \beta]$ The optimality set of *(B, N)* is the set defined as

$$\Lambda(B, N) \equiv \{\lambda \mid (B(\lambda), N(\lambda)) = (B, N)\}.$$

We also consider the set associated with the optimal partition *(B, N)* as follows:

$$\begin{aligned}\bar{\Lambda}(B, N) &\equiv \{\lambda \mid B(\lambda) \subseteq B\}\\ &= \{\lambda \mid N(\lambda) \supseteq N\}.\end{aligned}$$

Note that $\Lambda(B, N) \subseteq \bar{\Lambda}(B, N)$. Note also that there are a finite number of optimality sets, one for each distinct optimal partition that appears in [α,β]. Moreover, these optimality sets form a partition of [α,β]. Using Note 1, we can easily present equivalent definitions of the sets $\Lambda(B, N) \subseteq \bar{\Lambda}(B, N)$. in terms of the dual optimal faces $Y(\lambda)$as follows:

$$\Lambda(B, N) \equiv \{\lambda \mid Y(\lambda) = Y(\lambda^*)\},$$
$$\bar{\Lambda}(B, N) \equiv \{\lambda \mid Y(\lambda) \supseteq Y(\lambda^*)\},$$

where we recall that λ^* is such that, $(B, \mathrm{N}) = (B(\lambda^*), \mathrm{N}(\lambda^*))$.

Our main result, the characterization of the optimality sets, is given in the following theorem.

Theorem 1.2: *Let $(B, N) = (B(\lambda^*), N(\lambda^*))$ for $\lambda^* \in [\alpha,\beta]$. Then:*

(a) *If $\lambda^* = \lambda_i$ for some $i \in \{0, 1, \ldots, k\}$, that is, λ^* is a breakpoint of $\varphi(\lambda)$, then $\Lambda(B, N) = \bar{\Lambda}(B, N) = \{\lambda^*\}$.*

(b) *If $\lambda^* \in (\lambda_{i-1}, \lambda_i)$ for some $i \in \{1,\ldots, k\}$, then $\Lambda(B, N) = (\lambda_{i-1}, \lambda_i)$ and $\bar{\Lambda}(B, N) = [\lambda_{i-1}, \lambda_i]$.*

Consider an optimal partition $(B, N) = (B(\lambda), \mathrm{N}(\lambda))$ for some $\lambda \in [\alpha, \beta]$ First, we show that the optimality sets $\Lambda(B, N)$ and $\bar{\Lambda}(B, N)$ can be expressed as linear projections of certain polyhedral sets (Lemma 1.1). This fact immediately implies that $\Lambda(B, N)$ and $\bar{\Lambda}(B, N)$ are intervals which, in Lemma 1.2, are characterized by a certain algebraic condition on A_B and $\bar{b}$. In view of this result, from now on we refer to the optimality sets $\Lambda(B, N)$ as optimality intervals.

We now introduce new sets as follows. For $B \subseteq \{1, \ldots, n\}$ and $N = \{1, \ldots, n\} - B$, let

$$\begin{aligned}X_B(\lambda) &= \{x \in \mathbf{R}^n \mid Ax = b + \lambda\bar{b};\ x_B > 0;\ x_N = 0\},\\ \bar{X}_B(\lambda) &= \{x \in \mathbf{R}^n \mid Ax = b + \lambda\bar{b};\ x_B \geq 0;\ x_N = 0\},\\ \Gamma(B) &= \{\lambda \mid X_B(\lambda) \neq \varnothing\},\\ \bar{\Gamma}(B) &= \{\lambda \mid \bar{X}_B(\lambda) \neq \varnothing\},\\ Y_B &= \{y \in \mathbf{R}^m \mid A_B^T y = c_B;\ A_N^T y < c_N\}.\end{aligned}$$

An equivalent definition of the sets $\Gamma(B)$ and $\bar{\Gamma}(B)$ which we use later is as follows. Let $\pi\colon \mathbf{R} \times \mathbf{R}^n \to \mathbf{R}$ denote the projection $\pi(\lambda, x) = \lambda,\ (\lambda, x) \in \mathbf{R} \times \mathbf{R}^n$ and consider the sets Q_B and $\bar{Q}_B$ as follows:

$$\begin{aligned}Q_B &= \{\lambda, x) \in \mathbf{R} \times \mathbf{R}^n \mid A_B x_B - \lambda\bar{b} = b,\ x_B > 0,\ x_N = 0\},\\ \bar{Q}_B &= \{(\lambda, x) \in \mathbf{R} \times \mathbf{R}^n \mid A_B x_B - \lambda\bar{b} = b,\ x_B \geq 0,\ x_N = 0\}.\end{aligned}$$

Then the sets $\Gamma(B)$ and $\bar{\Gamma}(B)$ are the projection sets $\pi(Q_B)$ and $\pi(\bar{Q}_B)$, respectively. The following lemma relates the projection sets $\Gamma(B)$ and $\bar{\Gamma}(B)$ to the sets $\Lambda(B, N)$ and $\bar{\Lambda}(B, N)$, $N = \{1, \ldots, n\} - B$, when *(B, N)* is an optimal partition.

Lemma 1: *Let (B, N) = (B(*λ**), N(*λ**)) for some* $\lambda^* \in [\alpha, \beta]$. *Then:*

(a) $\Lambda(B, N) = \Gamma(B)$.
(b) $\bar{\Lambda}(B, N) = \bar{\Gamma}(B)$.
(c) $X(\lambda) = \bar{X}_B(\lambda)$ *for all* $\lambda \in \bar{\Gamma}(B)$ *(and hence for all* $\lambda \in \bar{\Lambda}(B, N)$*)*.

Proof: Note first that Result 2 easily implies that *(B, N) = (B(λ), N(λ))* if and only if $X_B(\lambda) \neq \emptyset$ and $YB \neq \emptyset$. Clearly, $YB \neq \emptyset$ since *(B, N) = (B(λ*), N(λ*))* by assumption. These two observations imply (a). We next show (b) and (c). By complementary slackness condition, $\bar{X}_B(\lambda) \neq \emptyset$ implies $Y_B \subseteq Y(\lambda)$ which implies that $X(\lambda) = \bar{X}_B(\lambda)$. This shows (c). Clearly, (c) implies that $B(\lambda) \subseteq B$ for all $\lambda \in \bar{\Gamma}(B)$.

Hence, $\bar{\Gamma}(B) \subseteq \bar{\Lambda}(B, N)$.. The inclusion $\bar{\Gamma}(B) \subseteq \bar{\Lambda}(B, N)$. is immediate since $B(\lambda)$implies $\emptyset \neq X(\lambda) \equiv \bar{X}_{B(\lambda)}(\lambda) \subseteq \bar{X}_B(\lambda)$. Hence (b) follows.

Lemma 2: *Let* $B \subseteq \{1, \ldots, n\}$ *be given. Assume that* $\Gamma(B) \neq \emptyset$ *and let* $\lambda^* \in \Gamma(B)$. *Then:*

(a) $\Gamma(B) = \{\lambda^*\}$ *if and only if* $\bar{b} \notin range(A_B)$.
(b) $\Gamma(B)$ *is an open interval (containing* λ^**) if and only if* $\bar{b} \in range(A_B)$.
(c) $\bar{\Gamma}(B) = \mathrm{cl}(\Gamma(B))$.

Proof: Consider the sets Q_B and $\bar{Q}_B$ and the projection $\pi(\lambda,, x) = x$ mentioned before the statement of Lemma 1. We know that $\pi(Q_B) = \Gamma(B)$ and $\pi(\bar{Q}_B) = \bar{\Gamma}(B)$. Since both Q_B and $\bar{Q}_B$ are convex sets and π is linear, it follows that both $\Gamma(B)$ and $\bar{\Gamma}(B)$ are convex sets and hence intervals. We first show (c). Clearly, since $QB \neq \emptyset$, we have $\bar{Q}_B$ = cl Q_B. Since π is continuous, this implies that $\bar{\Gamma}(B) = \pi(\bar{Q}_B) \subseteq \mathrm{cl}\, \pi(Q_B) = \mathrm{cl}\, \Gamma(B)$. Next, observe that $\bar{\Gamma}(B)$ is closed since $\bar{Q}_B$ is a poly- hedron and $\pi(\bar{Q}_B) = \bar{\Gamma}(B)$. Since $\Gamma(B) \subseteq \bar{\Gamma}(B)$, it follows that cl $\Gamma(B) \subseteq \bar{\Gamma}(B)$. Hence (c) follows. We next show (a) and (b). Assume first that $\bar{b} \notin$ range(A_B). We will show that $\Gamma(B) = \{\lambda^*\}$. Indeed, since $\lambda^* \in \Gamma(B)$, we have $b + \lambda^* \bar{b} \in$ *range(*A_B*)*. Since $\bar{b} \notin$ range(A_B), we can easily see that $b + \lambda \bar{b} \notin$ range(A_B) for any *)*.$\lambda \neq \lambda^*$ Therefore, $X_B(\lambda) = \emptyset$ for any $\lambda \neq \lambda^*$ which implies that $\Gamma(B) = \{\lambda^*\}$. Assume next that $\bar{b} \in$ range(A_B)· We will show that $\Gamma(B)$ is open and hence an open interval. Indeed, let $\bar{\lambda} \in \Gamma(B)$. Then we have $A_B x_B = b + \bar{\lambda}\bar{b}$ for some $x_B \in \mathbf{R}^{|B|}_{++}$. Also, since $\bar{b} \in$ range(A_B), we have $A_B u = \bar{b}$ for some $u \in \mathrm{R}^{|B|}$. Since $\bar{x}_B + \theta u > 0$ for all sufficiently small real number $\theta$and $A_B(x_B + \theta u) = b + (\bar{\lambda} + \theta)\bar{b}$, it follows that $\bar{\lambda} + \theta \in \Gamma(B)$ for all sufficiently small θ. We have thus shown that any $\bar{\lambda} \in \Gamma(B)$ has a neighborhood contained in $\Gamma(B)$. Hence, $\Gamma(B)$ is an open interval. This completes the proof of (a) and (b).

An Algorithm for the Parametric RHS LP Problem

In this section we present an algorithm for the parametric RHS LP problem. During the run of the algorithm, a sequence of LP problems is generated in such a way that the solution of one problem determines the next one to be solved. By solving this sequence of LP problems, we also obtain the solution to the given parametric RHS LP problem. As in the parametric RHS LP pivot algorithm, given a parameter $\lambda^* \in [\alpha,\beta]$ (Refer Result 3), the algorithm looks for the breakpoints of $\varphi(\lambda)$ first in one side of λ^* and then in the other side of λ^*.

By solving the parametric RHS LP problem, we mean:

(1) Find α and β satisfying the conditions of Result 3.

(2) Determine whether $\varphi([\alpha, \beta]) = \{-\infty\}$. If so, the solution to the parametric

RHS LP problem is completely specified and we stop. Otherwise, we have $\varphi([\alpha, \beta]) = R$ and we continue to find the following additional information:

(*a*) The breakpoints $\alpha = \lambda_0 < \lambda_1 < \ldots < \lambda_k = \beta$ of thew function $\varphi(\lambda)$;

(*b*) The slopes g_i such that, $g_i = \varphi'(\lambda)$ for $\lambda \in (\lambda_{i-1}, \lambda_i)$, i $\{1, \ldots, k\}$;

(*c*) The optimal partitions (B_i, N_i) such that, $(B_i, N_i) = (B(\lambda), N(\lambda))$ for $\lambda \in (\lambda_{i-1}, \lambda_i)$, $i \in \{1, \ldots, k\}$;

(*d*) The optimal partitions $(B(\lambda_i), N(\lambda_i))$ for $i \in \{0, 1, \ldots, k\}$;

(*e*) $x^i \in X(\lambda_i)$ for $i \in \{0, 1, \ldots, k\}$;

(*f*) $y^i \in Y_i$ for $i \in \{1, \ldots, k\}$.

To find the values $\varphi(\lambda)$ for all $\lambda \in [\alpha, \beta]$, all we need is (*a*) and (*b*) together with the value $\varphi(\lambda^*)$ for some $\lambda^* \in [\alpha, \beta]$. Points in the sets $X(\lambda)$ and $Y(\lambda)$ for all $\lambda \in [\alpha, \beta]$ can be obtained from information (*e*) and (*f*).

1.8 UPPER BOUND TECHNIQUE

An upper bound or majorant of a subset S of some preordered set $(K, \leq)$ is an element of *K* which is greater than or equal to every element of *S*. Dually, a lower bound or minorant of *S* is defined to be an element of *K* which is less than or equal to every element of *S*. A set with an upper (respectively, lower) bound is said to be bounded from above or majorized (respectively bounded from below or minorized) by that bound. The terms bounded above (bounded below) are also used in the mathematical literature for sets that have upper (respectively lower) bounds.

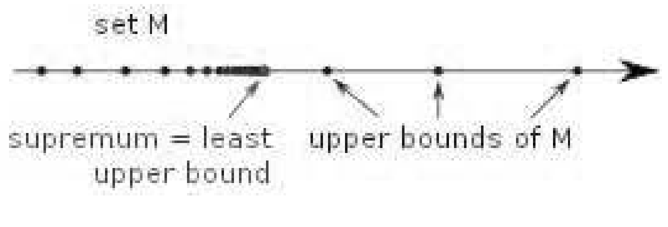


Fig1.2 *A Set with its Upper Bounds and its Least Upper Bound*

Examples

For example, 5 is a lower bound for the set *S* = {5, 8, 42, 34, 13934} (as a subset of the integers or of the real numbers, etc.), and so is 4. On the other hand, 6 is not a lower bound for *S* since it is not smaller than every element in *S*.

The set $S = \{42\}$ has 42 as both an upper bound and a lower bound; all other numbers are either an upper bound or a lower bound for that *S*.

Every subset of the natural numbers has a lower bound since the natural numbers have a least element (0 or 1, depending on convention). An infinite subset of the natural numbers cannot be bounded from above. An infinite subset of the integers may be bounded from below or bounded from above, but not both. An infinite subset of the rational numbers may or may not be bounded from below, and may or may not be bounded from above.

Every finite subset of a non-empty totally ordered set has both upper and lower bounds.

Bounds of Functions

The definitions can be generalized to functions and even to sets of functions.

Given a function f with domain *D* and a preordered set $(K, \leq)$ as codomain, an element *y* of *K* is an upper bound of f if $y \geq f(x)$ for each *x* in *D*. The upper bound is called sharp if equality holds for at least one value of *x*. It indicates that the constraint is optimal, and thus cannot be further reduced without invalidating the inequality.

Similarly, function g defined on domain *D* and having the same codomain $(K, \leq)$ is an upper bound of f, if $g(x) \geq f(x)$ for each *x* in *D*. Function g is further said to be an upper bound of a set of functions, if it is an upper bound of each function in that set.

The notion of lower bound for (sets of) functions is defined analogously, by replacing $\geq$ with $\leq$.

Tight Bounds

An upper bound is said to be a tight upper bound, a least upper bound, or a supremum, if no smaller value is an upper bound. Similarly, a lower bound is said to be a tight lower bound, a greatest lower bound, or an infimum, if no greater value is a lower bound.

Exact Upper Bounds

An upper bound u of a subset S of a preordered set $(K, \leq)$ is said to be an exact upper bound for *S* if every element of *K* which is strictly majorized by u is also majorized by some element of *S*. Exact upper bounds of reduced products of linear orders play an important role in PCF theory.

1.9 INTERIOR POINT ALGORITHM

Interior Point Methods (also referred to as barrier methods or IPMs) are a certain class of algorithms that solve linear and nonlinear convex optimization problems.

An interior point method was discovered by Soviet mathematician I. I. Dikin in 1967 and reinvented in the U.S. in the mid-1980s. In 1984, Narendra Karmarkar developed a method for linear programming called Karmarkar's algorithm, which runs in provably polynomial time and is also very efficient in practice. It enabled solutions of linear programming problems that were beyond the capabilities of the simplex method. Contrary to the simplex method, it reaches a best solution by traversing the interior of the feasible region. The method can be generalized to convex programming based on a self-concordant barrier function used to encode the convex set.

Any convex optimization problem can be transformed into minimizing (or maximizing) a linear function over a convex set by converting to the epigraph form. The idea of encoding the feasible set using a barrier and designing barrier algorithm was studied by Anthony V. Yurii Nesterov, and Arkadi Nemirovski came up with a special class of such barriers that can be used to encode any convex set. They guarantee that the number of iterations of the algorithm is bounded by a polynomial in the dimension and accuracy of the solution.

Karmarkar's breakthrough revitalized the study of interior point methods and barrier problems, showing that it was possible to create an algorithm for linear programming characterized by polynomial complexity and, moreover, that was competitive with the simplex method. Already Khachiyan's ellipsoid method was a polynomial-time algorithm; however, it was too slow to be of practical interest.

The class of primal-dual path-following interior-point methods is considered the most successful. Mehrotra's predictor–corrector algorithm provides the basis for most implementations of this class of methods.

Primal-Dual Interior Point Methods for Nonlinear Optimization

The primal-dual method's idea is easy to demonstrate for constrained nonlinear optimization. For simplicity, consider the all-inequality version of a nonlinear optimization problem:

Minimize $f(x)$ subject to $c_i\ (x) \geq 0$ for $i = 1,\ldots,m$, $x \in \mathbb{R}^n$, where $f\colon \mathbb{R}^n \rightarrow \mathbb{R}$, $c_i\colon \mathbb{R}^n \rightarrow \mathbb{R}$ (1.5)

This inequality-constrained optimization problem is then solved by converting it into an unconstrained objective function whose minimum we hope to find efficiently. Specifically, the logarithmic barrier function associated with Equation (1.5) is

$$B(x,\mu) = f(x) - \mu \sum_{i=1}^{m} \log(c_i(x)). \tag{1.6}$$

Here, μ is a small positive scalar, sometimes called the 'Barrier Parameter'. As μ converges to zero the minimum of $B(x, \mu)$ should converge to a solution of Equation (1.5).

The barrier function gradient is,

$$g(x) := \nabla f(x)\ g_b(x,\mu) := \nabla B(x,\mu) = g(x) - \mu \sum_{i=1}^{m} \frac{1}{c_i(x)} \nabla c_i(x), \tag{1.7}$$

where $g(x) := \nabla f(x)$ is the gradient of the original function $f(x)$ and $\nabla\ c_i$ is the gradient of c_i.

In addition to the original 'Primal' variable x we introduce a Lagrange multiplier-inspired dual variable $\lambda \in \mathbb{R}^n$

$$c_i(x)\lambda_i = \mu, \forall i = 1, \ldots, m. \tag{1.8}$$

Equation (1.8) is sometimes called the 'Perturbed Complementarity' condition, for its resemblance to 'Complementary Slackness' in KKT conditions. (x_μ, λ_μ) for which the gradient of the barrier function is zero.

Applying Equation (1.8) to Equation (1.7), we get an equation for the gradient:

$$g - A^T\lambda = 0, \tag{1.9}$$

where the matrix A is the Jacobian of the constraints $c(x)$.

The intuition behind Equation (1.9) is that the gradient of $f(x)$ should lie in the subspace spanned by the constraints' gradients. The 'Perturbed Complementarity' with small μ Equation (1.9) can be understood as the condition that the solution should either lie near the boundary $c_i(x) = 0$, or that the projection of the gradient g on the constraint component $c_i(x)$ normal should be almost zero.

Applying Newton's method to Equation (1.8) and Equation (1.9), we get an equation for (x, λ) update (p_x, p_λ):

$$\begin{pmatrix} W & -A^T \\ \Lambda A & C \end{pmatrix}\begin{pmatrix} p_x \\ p_\lambda \end{pmatrix} = \begin{pmatrix} -g + A^T\lambda \\ \mu 1 - C\lambda \end{pmatrix},$$

Where W is the Hessian matrix of $B(x, \mu)$, ∇ is a diagonal matrix of λ, and C is a diagonal matrix with $C_{ii} = C_i(x)$.

Because of Equations (1.5) and (1.8) the condition

$\lambda \geq 0$

should be enforced at each step. This can be done by choosing appropriate α:

$(x, \lambda) \rightarrow (x + \alpha p_x, \lambda + \alpha p_\lambda)$

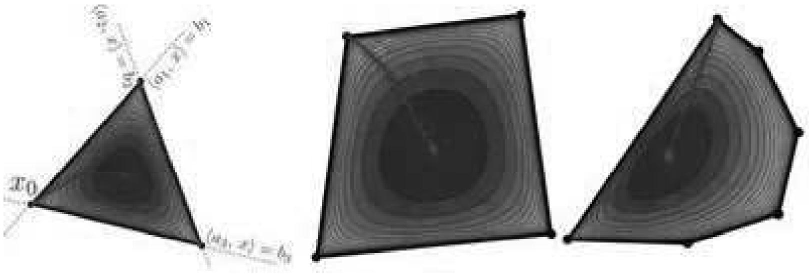


Fig.1.3 *Trajectory of the Iterates of x by using the Interior Point Algorithm*

1.10 LINEAR GOAL PROGRAMMING

Goal programming, as a concept, was first used by Charnes, Cooper and Ferguson in 1955. However, it was only in 1961 that the actual name appeared in a text introduced by Charnes and Cooper. In this text, they suggested the usage of a method that could be used for solving the multi-criteria dilemma faced due to the constraints of linear programming. Critical works on goal programming by Lee (1972) and Ignizio (1976) followed. This led to wide-scale usage of goal programming in planning, resource allocation, policy analysis and functional management issues. The first application of goal programming was done on an engineering application for design and placement of the antennas. This was during the second stage of Saturn V, which was used to launch the Apollo space capsule (this had landed the first men on the moon).

One of the landmark books on goal programming was written by Ijiri (1965) where he developed the concept of pre-emptive priority factors, assigning different priority levels to disproportionate goals and variant weights for the goals at the identical priority level. In goal programming, there is an achievement function that minimizes the deviations from the entrenched goal targets within a set of constraints. These are also known as slack variables (in the simplex algorithm of linear programming), and are used as dummy variables.

Terms: Objectives, Goals and Constraints

Objectives are referred to the optimization of the measure of performance of a decision, such as profit maximization or cost minimization.

Goals state a target value, i.e., the minimum acceptable level of performance of any decision taken by the decision-maker.

Constraints are similar to goals, in terms of their mathematical formulation. However, while goals are implied as the right-hand side value to achieve a certain target value, it is desirable for the constraints to achieve the right-hand side value.

Goal Programming Model Formulation

Single Goal with Multiple Subgoals

An objective is the desired level of result by a decision-maker. This desired level of result (goal) may be underachieved, completely achieved, or overachieved within the given decision-making environment. The target level of any goal is determined by the relative managerial effort that is applied to an activity. In mathematical terms, one unit of applied effort towards activity x_j might contribute the amount a_{ij} towards the i^{th} goal. If this applied effort achieves its target value, the *ith* constraint would be denoted as:

$$\sum_{j=1}^{n} a_{ij}\, x_j = b_i$$

However, one of the key features of goal programming is its flexibility and non-binding implications to mathematical interpretation. Thus, to allow underachievement or overachievement in the goal, we may denote:

d_i^-= Negative deviation from i^{th} goal, i.e., below the target value.

d_i^+= Positive deviation from i^{th} goal, i.e., above the target value.

In the light of these notations, the i^{th} goal can be further written as:

$$\sum_{j=1}^{n} a_{ij}\, x_j + d_i^- - d_i^+ = b_i$$

$$\begin{bmatrix}\text{Value of the}\\ \text{objective}\end{bmatrix} + \begin{bmatrix}\text{Amount}\\ \text{below the}\\ \text{goal}\end{bmatrix} + \begin{bmatrix}\text{Amount}\\ \text{above the}\\ \text{goal}\end{bmatrix} = \text{Goal}$$

Here, i is equal to 1, 2,..., m.

It is important to note that it is not possible to achieve amount below the goal or above the goal simultaneously. In such a case, deviational variables of both the target values or goals $\left(d_i^- \text{ or } d_i^+\right)$ may be zero in the solution $\left(d_i^- \times d_i^+ = 0\right)$. Taking this at its optimal value, one may assume it to be of a positive value in the solution or must be kept at zero. The only significant point to be taken care of is that the goal deviational variables must be non-negative.

Note: Slack and surplus variables in the linear programming model are equivalent to the deviational variables in goal programming.

The surplus variable in linear programming and the deviational variable in goal programming, that is denoted as d_i^+, is done away from the objective function when there is a situation of overachievement. Similarly, in case of underachievement, d_i^- (known as slack variable or deviational variable) is removed from the objective function of goal programming. But, there is an exceptional situation in which both d_i^- and d_i^+ are included in the objective function. This happens only when there is an exact attainment of the goal and it is ranked as per the pre-emptive priority factor.

Example 1.17: A packaged food manufacture produces two kinds of products, chips and soda. The unit profit from a packet of chips is ` 80, and of a bottle of soda is ` 40. The goal of the plant manager is to earn a total profit of exactly ` 640 in the next week.

Model Formulation

We may interpret the profit goal in terms of subgoals, which are sales volume of chips and soda. Thereby, a goal programming model may be formulated as:

$$\text{Minimize, } z = d_i^- + d_i^+,$$

$$\text{subject to } 80x_1 + 40x_2 + d_i^- - d_i^+ = 640$$

$$x_1, x_2, d_i^-, d_i^+ \geq 0$$

where,

x_1 = Number of Packet of Chips Sold

x_2 = Number of Bottles of Soda Sold

d_i^- = Underachievement of the Profit Goal of ₹ 640

d_i^+ = Overachievement of the Profit Goal of ₹ 640

If the profit goal is not fully achieved, the slack in the profit goal will be expressed by d_i^- (negative deviational variable). Contrary to this situation, if the solution shows a profit in excess of ` 640, then both d_i^+ and d_i^- will be zero. Here it must

NOTES

be noted that d_i^- and d_i^+ are complementary to each other. So, if the profit goal of ` 640 is exactly achieved, both d_i^- and d_i^+ will be zero.

In the above example 1.17, there are an infinite number of combinations of x_1 and x_2 that would achieve the goal. The solution would be of any linear combination of x_1 and x_2 between the two points ($x_1 = 8, x_2 = 0$) or ($x_1 = 0, x_2 = 16$). This straight line is exactly the iso-profit function line when the total profit is ` 640.

Equally Ranked Multiple Goals

This model given below can be extended to handle cases of multiple goals. Let us suppose that there are no model constraints.

Taking the same example as example 1.17:

Example 1.18: Let us consider that the package food manufacturer now desires to achieve a weekly profit as close as to ` 640 as possible. He wants to achieve sales volume for chips and soda close to six and four respectively. We can formulate this effort as a goal programming model.

Model Formulation

$$\text{Minimize, } z = d_1^- + d_2^+ + d_3^- + d_i^+$$

$$\text{Subject to, } 80x_1 + 40x_2 + d_i^- - d_i^+ = 640$$

$$x_1 + d_2^- = 6$$

$$x_2 + d_3^- = 4$$

$$x_1, x_2, d_1^-, d_1^+ d_2^-, d_3^- \geq 0$$

The above equation expresses the profit goal and the sales goals.

Here, d_2^- and d_3^- represent the underachievements of sales volume for chips and soda. It should be noted that d_2^+ and d_3^+ are not included in the second and third constraints, since the sales goal are given as the maximum sales volume. The solution to this problem can be formulated by a simple examination of the problem: if $x_2 = 6$ and $x_2 = 4$, then all targets will be completely achieved.

Therefore, $\quad d_1^- = d_2^- = d_3^- = d_1^+ = 0$

Ranking and Weighting of Unequal Multiple Goals

In general terms, goal programming model is a linear representation in which the optimum attainment of objectives is sought within the given decision environment. It is this decision environment which determines the basic component of the model, like constraints, decision variables and the objective function. Since multiple and conflicting goals are usually not of equal importance, negative or positive deviations are not added. To achieve the goals as per the pre-emptive priority factor, p_1, p_2... and so on are assigned to deviational variables in the formulation of the objective function to be minimized. The p_s does not assume any numerical value, so they are simply a convenient way of indicating that one goal is comparatively more important than another. The relationship between various priority factors is based on priority ranking, such as $p_1 >> p_2 >> ... \; p_k >> p_{k+1}$ ..., where >> 'Denotes more important than'.

This further means, $p_j >> p_{j+1}$ ($j = 1, 2, ..., k$) where n is a large number. The priority ranking of any target value or goal cannot be improved by multiplying by

n. Thus, it is important to note that the deviational variables at a similar priority level must have the same unit of measurability. This can be well illustrated in the following example:

Example 1.19: An office furniture manufacturer produces two types of products: desks and chairs. For manufacturing a chair or a desk, the manufacturer requires one hour of production capacity in the plant (maximum production capacity is 50 hours per week). However, due to limited sales capacity, the maximum number of desks and chairs which could be sold are six and eight per week, respectively. The gross margin from the sale of a desk is · 90 and · 60 for a chair.

The manufacturer desires to determine the number of units of each desk and chair, which should be produced per week in consideration of the following set of goals:

Goal 1: Available production capacity should be utilized as much as possible but should not exceed 50 hours per week.

Goal 2: Sales of both the products (desks and chairs) produced per week should be as much as possible.

Goal 3: Overtime should not exceed 20 per cent of the available time.

Model Formulation

We can formulate this problem according to a goal programming model so that the manufacturer may achieve his goals.

Suppose x_1 and x_2 = number of units of desk and chair produced. The first goal pertains to production capacity attainment with a target set at 50 hours per week. This constraint can be expressed as:

$$x_1 + x_2 + d_1^- + d_1^+ = 50$$

here, d_1^- = Underutilization of Production Capacity

d_1^+ = Overutilization of Production Capacity.

If this goal is not achieved, then d_1^- would take on a positive value and d_1^+ would be zero.

The second goal pertains to maximization of sales volume with a target of 6 units of desks and 8 units of chairs per week. The sales constraints can be expressed as:

$$x_1 + d_2^- = 6 \qquad\qquad x_2 + d_3^- = 8$$

Thus, it is important to note that the sales goals are the maximum possible sales volume, d_2^+ and d_3^+ will not appear as these constraints. So, here the overachievement of sales goals is ruled out.

The third goal looks for the minimization of overtime hours as minimum as possible. The constraint is denoted as:

$$d_1^+ + d_4^- - d_4^+ = 0.2\ (50) = 10$$

Here,

d_4^- = Overtime less than 20 per cent of goal constraint

d_4^+ = Overtime more than 20 per cent of goal constraint

d_1^+ = Overtime beyond 50 hours

Now, we an formulate a model to express the above given problem as a goal programming model.

Minimize (total deviation) $Z = d_1^+ + d_2^- + d_3^- + d_4^+$ subject to the above mentioned constraints,

(*i*) Production Capacity Constraint

$$x_1 + x_2 + d_1^- - d_1^+ = 50$$

(*ii*) Sales Constraints

$$x_1 + d_2^- = 6$$
$$x_2 + d_3^- = 8$$

(*iii*) Overtime Constraint

$$d_1^+ + d_4^- - d_4^+ = 10, \text{ and}$$
$$x_1, x_2, d_1^-, d_2^-, d_3^-, d_4^-, d_4^+ \geq 0.$$

Check Your Progress

7. Explain the duality in linear programming.
8. Describe the formulation of dual problem.
9. Write first four steps of dual simplex algorithm.
10. What is parametric linear programming?
11. Define upper bound techniques.
12. Sate interior point algorithm.
13. What are the objectives, goals and constraints of linear goal programming?

1.11 ANSWERS CHECK YOUR PROGRESS

1. The concept of operations research came into existence in a military context during World War II, when military management wanted to arrive at decisions on optimal utilization of scarce military resources with a new approach to the systematic and scientific study of the operations of the system.
2. Operation Research (OR) is the scientific knowledge through interdisciplinary team effort for the purpose of determining the best utilization of limited resources.
3. Decision-making has always been very important in the business and industrial world, particularly with regard to the problems concerning production of commodities. Which commodity/commodities to produce, in what quantities and by which process or processes, are the main questions before a production manager. English economist Alfred Marshall pointed out that the businessman always studies his production function and his input prices and substitutes one input for another till his costs become the minimum possible. All this sort of substitution, in the opinion of Marshall, is being done by businessman's trained instinct rather than with formal

calculations. But now there does exist a method of formal calculations often termed as Linear Programming (LP).

4. The word 'Linear' means that the relationships are represented by straight lines, i.e., the relationships are of the form $y = a + bx$ and the word 'Programming' means taking decisions systematically.
5. The applications of LP are numerous and are increasing every day. LP is extensively used in solving resource allocation problems. Production planning and scheduling, transportation, sales and advertising, financial planning, portfolio analysis, corporate planning, etc., are some of its most fertile application areas. More specifically, LP has been successfully applied in the following fields:
    (i) Agricultural applications: LP can be applied in farm management problems as it relates to the allocation of resources such as acreage, labour, water supply or working capital in such a way that is maximizes net revenue.
    (ii) Contract awards: Evaluation of tenders by recourse to LP guarantees that the awards are made in the cheapest way.
    (iii) Industrial applications: Applications of LP in business and industry are of most diverse kind. Transportation problems concerning cost minimization can be solved by this technique. The technique can also be adopted in solving the problems of production (product-mix) and inventory control.
6. Simplex method is an iterative procedure for solving LPP in a finite number of steps. This method provides an algorithm which consists of moving from one vertex of the region of feasible solution to another in such a manner that the value of the objective function at the succeeding vertex is less or more as the case may be that at the previous vertex.
7. For every given linear programming problem, there is another intimately related linear programming problem referred to as its dual. The duality theorem states that 'For every maximization (or minimization) problem in linear programming, there is a unique similar problem of minimization (or maximization) involving the same date which describes the original problem'.
8. The following changes are used in formulating the dual problem:
    (i) Change the objective function of maximization in the primal into minimization in the dual, and vice versa.
    (ii) The variables in the primal should be equal to the constraints in the dual and vice versa.
    (iii) The cost coefficients $C_1, C_2 \ldots C_n$ in the objective function of the primal should be the RHS constant of the constraints in the dual and vice versa.
    (iv) In forming the constraints for the dual, we consider the transpose of the body matrix of the primal problem.
    (v) The variables in both the problems are non-negative.
    (vi) If the variable in the primal is unrestricted in sign, then the corresponding constraint in the dual will be an equation, and vice versa.

9. The dual simplex method is very similar to the regular simplex method, the only difference lies in the criterion used for selecting a variable to enter the basis and to leave the basis. In the dual simplex method, we first select the variable to leave the basis and then the variable to enter the basis. This method yields an optimal solution to the given LPP in a finite number of steps, provided no basis is repeated.

   The dual simplex method is used to solve problems that start as dual feasible (i.e., whose primal is optimal but infeasible). In this method, the solution starts as optimum, but infeasible, and remains infeasible until the true optimum is reached, at which point the solution becomes feasible.

10 Parametric linear programming investigates the effect of predetermined continuous variations of these coefficients on the optimal solution. It is simply an extension of sensitivity analysis and aims at finding the various basic solutions that become optimal, one after the other, as the coefficients of the problem change continuously.

11. An upper bound or majorant of a subset S of some preordered set $(K, \leq)$ is an element of *K* which is greater than or equal to every element of *S*. Dually, a lower bound or minorant of *S* is defined to be an element of *K* which is less than or equal to every element of *S*. A set with an upper (respectively, lower) bound is said to be bounded from above or majorized (respectively bounded from below or minorized) by that bound.

12. Interior Point Methods (also referred to as barrier methods or IPMs) are a certain class of algorithms that solve linear and nonlinear convex optimization problems.

    An interior point method was discovered by Soviet mathematician I. I. Dikin in 1967 and reinvented in the U.S. in the mid-1980s.

13. Objectives: The measure of performance of a decision, such as profit maximization or cost minimization.

    Goals: The minimum acceptable level of performance of any decision taken by the decision-maker.

    Constraints: In terms of their mathematical formulation. However, while goals are implied as the right-hand side value to achieve a certain target value, it is desirable for the constraints to achieve the right-hand side value.

1.12 SUMMARY

- Operations research was first coined in 1940 by J.F. McCloskey and F.N. Trefethen in a small town Bowdsey, the United Kingdom.
- This new science came into existence in a military context. During World War II, military management called on scientists from various disciplines and organized them into teams to assist in solving strategic and tactical problems, relating to air and land defence.
- OR is a scientific method of providing executive departments with a quantitative basis for decisions regarding the operations under their control.

- OR is a systematic method-oriented study of the basic structures, characteristics, functions and relationships of an organization to provide the executive with a sound, scientific and quantitative basis for decision-making.
- OR is a scientific approach to problem solving for executive management.
- Decision-making has always been very important in the business and industrial world, particularly with regard to the problems concerning production of commodities. Which commodity/commodities to produce, in what quantities and by which process or processes, are the main questions before a production manager.
- The word 'Linear' means that the relationships are represented by straight lines, i.e., the relationships are of the form $y = a + bx$ and the word 'Programming' means taking decisions systematically.
- The problem for which LP provides a solution may be stated to maximize or minimize for some dependent variable which is a function of several independent variables when the independent variables are subject to various restrictions.
- Agricultural applications: LP can be applied in farm management problems as it relates to the allocation of resources such as acreage, labour, water supply or working capital in such a way that is maximizes net revenue.
- Industrial applications: Applications of LP in business and industry are of most diverse kind. Transportation problems concerning cost minimization can be solved by this technique. The technique can also be adopted in solving the problems of production (product-mix) and inventory control.
- Simplex method is an iterative procedure for solving LPP in a finite number of steps. This method provides an algorithm which consists of moving from one vertex of the region of feasible solution to another in such a manner that the value of the objective function at the succeeding vertex is less or more as the case may be that at the previous vertex.
- For every given linear programming problem, there is another intimately related linear programming problem referred to as its dual. The duality theorem states that 'for every maximization (or minimization) problem in linear programming, there is a unique similar problem of minimization (or maximization) involving the same date which describes the original problem'.
- Every LPP (called the primal) is associated with another LPP (called its dual). Either of the problem can be considered as primal with the other as dual.
- If the variable in the primal is unrestricted in sign, then the corresponding constraint in the dual will be an equation, and vice versa.
- The dual simplex method is very similar to the regular simplex method, the only difference lies in the criterion used for selecting a variable to enter the basis and to leave the basis. In the dual simplex method, we first select the variable to leave the basis and then the variable to enter the basis. This method yields an optimal solution to the given LPP in a finite number of steps, provided no basis is repeated.

NOTES

- The dual simplex method is used to solve problems that start as dual feasible (i.e., whose primal is optimal but infeasible). In this method, the solution starts as optimum, but infeasible, and remains infeasible until the true optimum is reached, at which point the solution becomes feasible.
- Parametric linear programming investigates the effect of predetermined continuous variations of these coefficients on the optimal solution. It is simply an extension of sensitivity analysis and aims at finding the various basic solutions that become optimal, one after the other, as the coefficients of the problem change continuously.
- Interior Point Methods (also referred to as barrier methods or IPMs) are a certain class of algorithms that solve linear and nonlinear convex optimization problems.
- An interior point method was discovered by Soviet mathematician I. I. Dikin in 1967 and reinvented in the U.S. in the mid-1980s.
- Goal programming, as a concept, was first used by Charnes, Cooper and Ferguson in 1955. However, it was only in 1961 that the actual name appeared in a text introduced by Charnes and Cooper.

1.13 KEY TERMS

- **Operations research:** The application of scientific knowledge through interdisciplinary team effort for the purpose of determining the best utilization of limited resources.
- **Linear programming:** the word 'Linear' means that the relationships are represented by straight lines, i.e., the relationships are of the form $y = bx$ and the word 'Programming' means taking decisions systematically.
- **Simplex method:** Simplex method is an iterative procedure for solving LPP in a finite number of steps. This method provides an algorithm which consists of moving from one vertex of the region of feasible solution to another in such a manner that the value of the objective function at the succeeding vertex is less or more as the case may be that at the previous vertex.
- **Duality in LPP:** Every LPP (called the primal) is associated with another LPP (called dual). Either of the problem can be considered as primal with the other as dual.
- **Interior Point Algorithm:** Interior Point Methods (also referred to as barrier methods or IPMs) are a certain class of algorithms that solve linear and nonlinear convex optimization problems.

1.14 SELF-ASSESSMENT QUESTIONS AND EXERCISES

Short-Answer Questions

1. Where did the concept of operations research originate?

2. What is meant by proportionality in linear programming?
3. Mention two areas where linear programming finds application.
4. Write the simplex algorithm.
5. State the formulation of dual problem.
6. Define dual simplex method.
7. What do you mean by parametric linear programming?
8. What is upper bound techniques?
9. Write the properties of interior point algorithm.
10. What are the objectives, goals and constraints of linear goal programming?

Long-Answer Questions

1. What are the scope of operations research? Explain.
2. Describe the areas where linear programming can be used?
3. Analyse the basic concepts and notations in linear programming.
4. Explain the applications of linear programming.
5. Illustrate the simplex method with appropriate examples.
6. Elaborate on the simplex algorithm giving examples.
7. Discuss the duality in linear programming with help of examples.
8. Solve the following mixed integer programing problems using Gomory's cutting plane method.

   Min $Z = 10x_1 + 9x_2$

   Subject to constraints, $x_1 \leq 8$

   $$x_2 \leq 10$$
   $$5x_1 + 3x_2 \leq 45$$
   $x_1, x_2 \leq 0$ and x_1 is an integer.
9. Describe the parametric linear programming.
10. Discuss the upper bound techniques that are used in LPP.
11. Analyse the interior point algorithm. Give appropriate examples.
12. What do you understand by linear goal programming? Explain.

1.15 FURTHER READING

Kalavathy, S. 2002. *Operations Research*. New Delhi: Vikas Publishing House Pvt. Ltd.

Gupta, P.K. and D.S. Hira. 2002. *Introduction to Operations Research*. New Delhi: S. Chand And Company Limited.

Taha, H.A. 2006. *Operations Research: An Introduction*, 8th Edition. New Delhi: Prentice-Hall of India.

Jensen, Paul A. and Jonathan F. Bard. 2003. *Operations Research Models and Methods*. New York: John Wiley & Sons.

Arumugam, R. S. 2006. *Operations Research*. Palayamkottai (Tamil Nadu): New Gamma Publications.

Sundharesan, V., K. S. Ganapathy and K. Ganesan. 2017. *Resource Management Techniques (Operations Research)*. Chennai: A. R. Publications.

Gillett, Billy E. 2007. *Introduction to Operations Research*. New Delhi: Tata McGraw-Hill.

Ackoff, R. L. and M. W. Sasieni. 1968. *Fundamentals of Operations Research*. New York: John Wiley & Sons Inc.

Kothari, C. R. 1992. *An Introduction to Operational Research*. New Delhi: Vikas Publishing House Pvt. Ltd.

Sharma, J. K. 2001. *Operations Research: Theory and Applications*. New Delhi: Macmillan India Ltd.

Swaroop, Kanti, P.K. Gupta and Man Mohan. 2001. *Operations Research*. New Delhi: Sultan Chand & Sons.

UNIT 2 TRANSPORTATION AND ASSIGNMENT PROBLEMS

NOTES

Structure

2.0 INTRODUCTION

In mathematics, transportation problems deal with the objective of transporting various quantities of a single homogeneous commodity initially stored at various origins to different destinations in a way that keeps transportation cost at a minimum. You will learn about applications of the transportation problem and solution and rules to solve such problems. The solution of any transportation problem is obtained in two stages, initial solution and optimal solution. There are three methods of obtaining an initial solution which include North West Corner Rule, Least Cost Method and Vogel's Approximation Method (VAM). VAM is preferred since the solution obtained this way is very close to the optimal solution. The optimal solution of any transportation problem is a feasible solution that minimizes the total cost. An optimal solution is the second stage of a solution obtained by improving the initial solution.

An assignment problem presented in the form of n × n cost matrix of real numbers. For solving an assignment problem, the Hungarian method is used. A network is a graphic representation of logically and sequentially connected arrows and nodes, representing the activities and events, respectively of a project.

NOTES

An event is the beginning and end points of an activity and is represented by a node. Also learn how to construct a network after going through the rules for network construction and a set of tools that has proven to be consistently valuable to project managers. The tools are collectively known as the Project Evaluation and Review Technique (PERT) and the Critical Path Method (CPM). PERT was developed by the U.S. Navy and its consultants for the Polaris Missile Project, while the Critical Path Method was created by DuPont and the Remington Rand Corporation for the management of large chemical plants. The maximization in assignment problems and the job sequence problem. To solve a job sequence problem, the assignment problem has to be balanced. An assignment problem is balanced if the cost matrix is a square matrix, otherwise it is termed as unbalanced. To make an unbalanced assignment problem balanced, dummy rows or dummy columns are added with all entries as zeroes.

In this unit, you will learn about the transportation problem, assignment problem, network analysis, shortest path problem and maximum flow problem, minimum cost flow problem, network simplex method and project planning and control with PERT-CPM.

2.1 OBJECTIVES

After going through this unit, you will be able to:

- Analyse the transportation problem and assigment problem
- Explain the three methods of finding an initial solution
- State the method of doing network analysis for large projects
- Understand PERT and CPM for handling projects
- Explain shorter route and maximal flow problem
- Perform critical path and PERT calculations
- Understand project scheduling
- Solve an assignment problem using the Hungarian method

2.2 TRANSPORTATION PROBLEM

The Transportation Problem (TP) is one of the subclasses of LPP (Linear Programming Problem) in which the objective is to transport various quantities of a single homogeneous commodity that are initially stored at various origins to different destinations in such a way that the transportation cost is minimum. To achieve this objective we must know the amount and location of available supplies and the quantities demanded. In addition we must know the costs that result from transporting one unit of commodity from various origins to various destinations.

Elementary Transportation Problem

Consider a transportation problem with m origins (rows) and n destinations (columns). Let C_{ij} be the cost of transporting one unit of the product from the ith origin to jth destination, a_i the quantity of commodity available at origin i, b_j the quantity of commodity needed at destination j. X_{ij} is the quantity transported from *i*th origin to *j*th destination. This transportation problem can be stated in the following tabular form.

Destinations

Origins

	1	2	3	…	n	Capacity
1	C_{11} X_{11}	C_{12} X_{12}	C_{13} X_{13}	… …	C_{1n} X_{1n}	a_1
2	C_{21} X_{21}	C_{22} X_{22}	C_{23} X_{23}	… …	C_{2n} X_{2n}	a_2
3	C_{31} X_{31}	C_{32} X_{32}	C_{33}	… …	C_{3n} X_{3n}	a_3
m	C_{m1} X_{m1}	C_{m2} X_{m2}	C_{m3} X_{m3}	… …	C_{mn} X_{mn}	a_m
Demand	b_1	b_2	b_3	…	b_n	$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

The linear programming model representing the transportation problem is given by,

$$\text{Minimize } Z = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}X_{ij}$$

Subject to the constraints,

$$\sum_{j=1}^{n} X_{ij} = a_i \qquad i = 1, 2, ..., n \text{ (Row sum)}$$

$$\sum_{i=1}^{m} X_{ij} = b_j \qquad j = 1, 2, ..., n \text{ (Column sum)}$$

$$X_{ij} \geq 0 \quad \text{For all } i \text{ and } j$$

The given transportation problem is said to be balanced if,

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

i.e., the total supply is equal to the total demand.

Definitions

Feasible Solution: Any set of non-negative allocations (X_{ij}>0) which satisfies the row and column sum is called a feasible solution.

Basic Feasible Solution: A feasible solution is called a basic feasible solution if the number of non-negative allocations is equal to $m + n - 1$, where m is the number of rows and n the number of columns in a transportation table.

Non-Degenerate Basic Feasible Solution: Any feasible solution to a transportation problem containing m origins and n destinations is said to be non-degenerate, if it contains $m + n - 1$ occupied cells and each allocation is in independent positions.

The allocations are said to be in independent positions if it is impossible to form a closed path. Closed path means by allowing horizontal and vertical lines and when all the corner cells are occupied.

NOTES

The allocations in the following tables are not in independent positions.

	*	*
	*	*

*		*
*		*

	*	*	
	*		
	*	*	

The allocations in the following tables are in independent positions.

	*	
*	*	*
*		

*	*		
	*		*
		*	*

Degenerate Basic Feasible Solution: If a basic feasible solution contains less than $m + n - 1$ non-negative allocations it is said to be degenerate.

Transportation Algorithm

This algorithm can be used for minimizing the transportation cost for goods from *O* origins to *D* destinations and there may be *O**D* number of direct routes from *O* origins to *D* destinations. Problem is balanced when sum of supplies at *O* sources is equal to sum of demands at *D* destinations. If it is not so, then this problem is not balanced. There may be two such situations. Supply may be lesser than demand and in that case it is balanced by adding dummy supply node. If demand is lesser than supply then dummy demand node is added to make it a balanced problem. Thus, before starting to use this algorithm, problem should be made balanced, if it is not balanced.

Data is presented in tabular form. As a convention, origins are put on left side of the table with quantity to be supplied listed towards right side and demands are put on top with quantity of demand towards the bottom side. Unit cost of transportation is put at the top of every cell within a small box. Zero unit cost shows unshipped unit column in case supply is in excess of the demand. Similarly, a unit cost either penalty or zero shows shortage row in supplies that are lesser that demand.

The algorithm has two phases. In phase I this makes allocation for supplies on demands by making utilizing an approach of minimum unit cost for generating a feasible solution. This feasible solution may not be optimal. Optimization is done in second phase and in this phase checking is done for optimality conditions and improvement is done for reducing the cost if optimality conditions are not satisfied. This second phase adopts iterative steps and stops only when optimality conditions are satisfied. Once done, no further steps are required.

Basic and Non-Basic Cells

Basic cells are those that indicate positive values and non-basic cells have zero value for flow. According to transportation problem number of basic cells will be exactly $m + n - 1$.

Algorithm follows as below:

Step 0

Initialization: Before starting to solve the problem, it should be balanced. If not then make it balanced by 'Unshipped Supply' column in case demand is less than supply or by adding 'Shortage' row in case supply is less than the supply. Put zero for unit costs in the column for unshipped supply. Put either penalty costs or zero in a row that shows shortage.

Phase I: To Find Initial 'Feasible Solution'

Step 1: Locate cell with minimum cost having positive supply as well as demand, then make allocation in that cell having residuals as minimum.

Step 2: Reduce residual supply/demand as per allocations made above (Step 1). Do this till all demands are met and then proceed to Phase II. If all demands are not met go to Step 1.

Phase II: Optimal Solution

Carry out check for Optimality.

Step 3: For rows and columns find dual values, u_i for rows and v_j for columns. For this, set u_1 (first dual value) to 0 followed by solution of triangular dual equations one by one. These **dual equations** are to be applied for basic cells only as below as,

$$C_{ij} = u_i + v_j$$

Where, C_{ij} denotes unit cost, as given for that cell and either v_j or u_i is already known. Here, v_j denotes dual value of column j and u_i denotes that for row i. These are taken as already known. Other dual value is computed from equation,

$$C_{ij} = u_i + v_j$$

Thus, dual values of every cell can be computed by setting the first to zero and appropriate order is used for dual equations for basic cells.

Step 4: Optimality conditions are expressed as **reduced costs** in case of all non-basic cells as given below:

$$\Delta_{ij} = C_{ij} - (u_i + v_j)$$

Reduced cost (Δ_{ij}) in case of non-basic cells represent net change in unit cost resulting due to movement of cell *ij* to solve and adjusting around a cycle that has been created in this way for basic cells to find current solution. This is shown in Step 5. Hence if one Δ_{ij} is positive, then by use of this cell total transportation will increase, but when Δ_{ij} is negative for a cell, this will cause reduction in total transportation. If all reduced costs (Δ_{ij}) are positive, i.e., $\Delta_{ij} > 0$, ***optimality conditions*** are satisfied and no further improvement is possible and algorithm terminates at this point. But if at least one Δ_{ij} is negative, optimality conditions are not satisfied and there is possibility of reduction in costs. If optimality conditions are not satisfied algorithm continues as given in Step 5.

Adjustment for Reducing Cost

Step 5: Select a cell '*ij*' that is most negative for Δ_{ij}. This becomes entering variable. Put (+) sign for identifying it. To maintain constraints of balance in supply and demand, locate basic cells for *i*th row as well as *j*th column that compensates for increase in value in cell '*ij*'. Put negative (–) sign in these cells. This process should be continued to get one cycle in which (+) and (–) are marked. Such a cycle will unique and if 'Dead Ends' are encountered in such a process, make a back track and one amongst other alternatives are tried. A cycle has rows/column having non-basic cells for holding compensating (+) or (–) sign. This may require trial and error for to finding it.

Step 6: After determining every basic cell within this cycle, adjustment is obtained as minimum value in basic cells that are negative. This is known as adjustment amount and let it be called '**aa**'. Add this to every cell value that is positive and marked with (+) sign, and subsequently deduct this from cells having negative (–) sign and then drop those from the basis that becomes zero. In case two or more cells become zero from such adjustment, drop only one of these that have greatest C_{ij} value. This is necessary for maintaining basic cells having $m + n - 1$ number for computing dual values. Reduction in cost that is associated with such a change is found as product of reduced cost Δ_{ij} for incoming cell multiplied by cell value previously held by outgoing cell.

After finding this new solution, move to the third Step 3 for checking conditions of optimality. If optimality condition is satisfied, algorithm terminates.

Applications of Transportation Problem

The Travelling Salesman Problem

Assume that a salesman has to visit *n* cities. He wishes to start from a particular city, visits each city once and then returns to his starting point. His objective is to select the sequence in which the cities are visited in such a way that his total travelling time is minimized.

To visit 2 cities A and B, there is no choice. To visit 3 cities we have 2! possible routes. For 4 cities we have 3! possible routes. In general to visit *n* cities there are $(n - 1)!$ possible routes.

Mathematical Formulation

Let C_{ij} be the distance or time or cost of going from city *i* to city *j*. The decision variable X_{ij} be 1 if the salesman travels from city *i* to city *j* and otherwise 0.

The objective is to minimize the travelling time.

$$Z = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}\, X_{ij}$$

Subject to the constraints,

$$\sum_{j=1}^{n} X_{ij} = 1, i = 2, ..., n$$

$$\sum_{i=1}^{n} X_{ij} = 1, j = 2, ..., n$$

Subject to the additional constraint that X_{ij} is so chosen that no city is visited twice before all the cities are completely visited.

In particular, going directly from i to i is not permitted. Which means $C_{ij} = \infty$, when $i = j$.

In travelling salesman problem we cannot choose the element along the diagonal and this can be avoided by filling the diagonal with infinitely large elements.

The travelling salesman problem is very similar to the assignment problem except that in the former case, there is an additional restriction that X_{ij} is so chosen that no city is visited twice before the tour of all the cities is completed.

Treat the problem as an assignment problem and solve it using the same procedures. If the optimal solution of the assignment problem satisfies the additional constraint, then it is also an optimal solution of the given travelling salesman problem. If the solution to the assignment problem does not satisfy the additional restriction then after solving the problem by assignment technique we use the method of enumeration.

Example 2.1: A travelling salesman has to visit 5 cities. He wishes to start from a particular city, visit each city once and then return to his starting point. Cost of going from one city to another is shown below. You are required to find the least cost route.

To City

$$\textbf{From City}\quad \begin{array}{c|ccccc|} & A & B & C & D & E \\ \hline A & \infty & 4 & 10 & 14 & 2 \\ B & 12 & \infty & 6 & 10 & 4 \\ C & 16 & 14 & \infty & 8 & 14 \\ D & 24 & 8 & 12 & \infty & 10 \\ E & 2 & 6 & 4 & 16 & \infty \\ \hline \end{array}$$

Solution: First we solve this problem as an assignment problem.

Subtract the minimum element in each row from all the elements in its row.

$$\begin{array}{c|ccccc|} & A & B & C & D & E \\ \hline A & \infty & 2 & 8 & 12 & 0 \\ B & 8 & \infty & 2 & 6 & 0 \\ C & 8 & 6 & \infty & 0 & 6 \\ D & 16 & 0 & 4 & \infty & 2 \\ E & 0 & 4 & 2 & 14 & \infty \\ \hline \end{array}$$

Subtract the minimum element in each column from all the elements in its column.

NOTES

$$\begin{array}{c|ccccc|} & A & B & C & D & E \\ A & \infty & 2 & 8 & 12 & 0 \\ B & 8 & \infty & 2 & 6 & 0 \\ C & 8 & 6 & \infty & 0 & 6 \\ D & 16 & 0 & 4 & \infty & 2 \\ E & 0 & 4 & 2 & 14 & \infty \end{array}$$

We have the first modified matrix. Draw minimum number of lines to cover all zeros.

$$\begin{array}{c|ccccc|} & A & B & C & D & E \\ A & \infty & 2 & 8 & 12 & 0 \\ B & 8 & \infty & 2 & 6 & 0 \\ C & 8 & 6 & \infty & 0 & 6 \\ D & 16 & 0 & 4 & \infty & 2 \\ E & 0 & 4 & 2 & 14 & \infty \end{array}$$

Here $N = 4 < n = 5$. Subtract the smallest uncovered element from all the uncovered elements and add to the element which is in the point of intersection of lines. Hence, we get the second modified matrix.

$$\begin{array}{c|ccccc|} & A & B & C & D & E \\ A & \infty & 0 & 6 & 12 & 0 \\ B & 8 & \infty & 0 & 6 & 0 \\ C & 8 & 4 & \infty & 0 & 6 \\ D & 18 & 0 & 4 & \infty & 4 \\ E & 0 & 2 & 0 & 14 & \infty \end{array}$$

$N = 5 = n = 5 =$ Order of matrix. We make assignment.

Assignment

$$\begin{array}{c|ccccc|l} & A & B & C & D & E & \\ A & \infty & \xcancel{0} & 6 & 12 & \textcircled{0} & \text{A} \to \text{E} \\ B & 8 & \infty & \textcircled{0} & 6 & \xcancel{0} & \text{B} \to \text{C} \\ C & 8 & 4 & \infty & \textcircled{0} & 6 & \text{C} \to \text{D} \\ D & 18 & \textcircled{0} & 4 & \infty & 4 & \text{D} \to \text{B} \\ E & \textcircled{0} & 2 & \xcancel{0} & 14 & \infty & \text{E} \to \text{A} \\ & & & & & & \text{A} \to \text{E} \to \text{A} \end{array}$$

As the salesman should go from A to E and then come back to A without covering B, C, D which is contradicting the fact that no city is visited twice before all the cities are visitied.

Hence, we obtain the next best solution by bringing the next minimum non-zero element namely 4.

$$\begin{array}{c|ccccc|} & A & B & C & D & E \\ A & \infty & \textcircled{0} & 6 & 12 & 0 \\ B & 8 & \infty & \textcircled{0} & 6 & \xcancel{0} \\ C & 8 & 4 & \infty & \textcircled{0} & 6 \\ D & 18 & \xcancel{0} & 4 & \infty & \textcircled{4} \\ E & \textcircled{0} & 2 & \xcancel{0} & 14 & \infty \end{array}$$

$A \to B, B \to C, C \to D, D \to E, E \to A$

Since all the cities have been visited and no city is visited twice before completing the tour of all the cities, we have an optimal solution to the travelling salesman.

The least cost route is $A \to B \to C \to D \to E \to A$.

Total Cost = 4 + 6 + 8 + 10 + 2 = ₹ 30.

Example 2.2: A machine operator processes five types of items on his machine each week and must choose a sequence for them. The set-up cost per change depends on the items presently on the machine and the set-up to be made according to the following table.

To Item

$$\textbf{From Item}\ \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array}\begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \left[\begin{array}{ccccc} \infty & 4 & 7 & 3 & 4 \\ 4 & \infty & 6 & 3 & 4 \\ 7 & 6 & \infty & 7 & 5 \\ 3 & 3 & 7 & \infty & 7 \\ 4 & 4 & 5 & 7 & \infty \end{array}\right] \end{array}$$

If he processes each type of item once and only once in each week, how should he sequence the items on his machine in order to minimize the total set-up cost?

Solution: Reduce the cost matrix and make assignments in rows and columns having single row.

Modify the matrix by subtracting the least element from all the elements in its row and also in its column.

$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & \infty & 1 & 4 & 0 & 1 \\ B & 1 & \infty & 3 & 0 & 1 \\ C & 2 & 1 & \infty & 2 & 0 \\ D & 0 & 0 & 4 & \infty & 4 \\ E & 0 & 0 & 1 & 3 & \infty \end{array}$$

$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & \infty & 1 & 4 & 0 & 1 \\ B & 1 & \infty & 3 & 0 & 1 \\ C & 2 & 1 & \infty & 2 & 0 \\ D & 0 & 0 & 4 & \infty & 4 \\ E & 0 & 0 & 1 & 3 & \infty \end{array}$$

Here, $N = 4 < n = 5$, i.e., $N < n$.

Subtract the smallest uncovered element from all the uncovered elements and add to the element which is at the point of intersection of lines and get the reduced second modified matrix.

$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline & \infty & 0 & 3 & 0 & 1 \\ B & 0 & \infty & 2 & 0 & 1 \\ C & 1 & 0 & \infty & 1 & 0 \\ D & 0 & 0 & 4 & \infty & 5 \\ E & 0 & 0 & 1 & 4 & \infty \end{array}$$

Here, $N = 5 = n = 5$ = Order of matrix. We make the assignment.

Assignment

We make a solution by considering the next smallest non-zero element by considering 1.

NOTES

$$\begin{array}{c|ccccc|l} & A & B & C & D & E & \\ \hline A & \infty & \xcancel{0} & 3 & \xcancel{0} & \textcircled{1} & \mathrm{A} \to \mathrm{E} \\ B & \xcancel{0} & \infty & 2 & \textcircled{0} & 1 & \mathrm{B} \to \mathrm{D} \\ C & 1 & \textcircled{0} & \infty & 1 & \xcancel{0} & \mathrm{C} \to \mathrm{B} \\ D & \textcircled{0} & \xcancel{0} & 4 & \infty & 5 & \mathrm{D} \to \mathrm{A} \\ E & \xcancel{0} & \xcancel{0} & \textcircled{1} & 4 & \infty & \mathrm{E} \to \mathrm{C} \end{array}$$

$A \to E,\ E \to C,\ C \to B,\ B \to D,\ D \to A$

We get the solution $A \to E \to C \to B \to D \to A$.

This schedule provides the required solution in which each item is not processed once in a week.

i.e., $A \to E \to C \to B \to D \to A$.

The total set-up cost comes to ₹ 21.

2.2.1 Solutions of Transportation Problem

Optimal solution is a feasible solution (not necessarily basic) which minimizes the total cost.

The solution of a Transportation Problem (TP) can be obtained in two stages, namely initial solution and optimum solution.

Initial solution can be obtained by using any one of the three methods, viz.

(*i*) North West Corner Rule (NWCR)

(*ii*) Least Cost Method or Matrix Minima Method

(*iii*) Vogel's Approximation Method (VAM)

VAM is preferred over the other two methods, since the initial basic feasible solution obtained by this method is either optimal or very close to the optimal solution.

The cells in the transportation table can be classified as occupied cells and unoccupied cells. The allocated cells in the transportation table is called *occupied cells* and *empty cells* in a transportation table is called *unoccupied cells*.

The improved solution of the initial basic feasible solution is called optimal solution which is the second stage of solution, that can be obtained by MODI (MOdified DIstribution Method).

North West Corner Rule

Step 1: Starting with the cell at the upper left corner (North West) of the transportation matrix we allocate as much as possible so that either the capacity of the first row is exhausted or the destination requirement of the first column is satisfied, i.e., $X_{11} = \text{Min } (a_1, b_1)$.

Step 2: If $b_1 > a_1$, we move down vertically to the second row and make the second allocation of magnitude $X_{22} = \text{Min}(a_2\ b_1 - X_{11})$ in the cell (2, 1).

NOTES

If $b_1 < a_1$, move right horizontally to the second column and make the second allocation of magnitude $X_{12} = \text{Min}\ (a_1, X_{11} - b_1)$ in the cell (1, 2).

If $b_1 = a_1$, there is a tie for the second allocation. We make the second allocations of magnitude,

$$X_{12} = \text{Min}\ (a_1 - a_1, b_1) = 0 \text{ in the cell } (1, 2)$$

$$\text{or, } X_{21} = \text{Min}\ (a_2, b_1 - b_1) = 0 \text{ in the cell } (2,1)$$

Step 3: Repeat Steps 1 and 2 moving down towards the lower right corner of the transportation table until all the rim requirements are satisfied.

Example 2.3: Obtain the initial basic feasible solution of a transportation problem whose cost and rim requirement table is as follows:

Origin\Destination	D_1	D_2	D_3	*Supply*
O_1	2	7	4	5
O_2	3	3	1	8
O_3	5	4	7	7
O_4	1	6	2	14
Demand	7	9	18	34

Solution: Since $\Sigma a_i = 34 = \Sigma b_j$, there exists a feasible solution to the transportation problem. We obtain initial feasible solution as follows.

The first allocation is made in the cell (1, 1) the magnitude being,

$X_{11} = \text{Min}\ (5, 7) = 5.$

The second allocation is made in the cell (2, 1) and the magnitude of the allocation is given by $X_{21} = \text{Min}\ (8, 7 - 5) = 2.$

	D_1	D_2	D_3	*Supply*
O_1	2 ⑤	7	4	~~5~~ 0
O_2	3 ②	3 ⑥	1	~~8~~ ~~6~~ 0
O_3	5	4 ③	7 ④	~~7~~ ~~4~~ 0
O_4	1	6	2 ⑭	~~14~~ 0
Demand	~~7~~ ~~2~~ 0	~~9~~ ~~3~~ 0	~~18~~ ~~14~~ 0	34

The third allocation is made in the cell (2, 2) the magnitude $X_{22} = \text{Min}\ (8 - 2, 9) = 6.$

The magnitude of the fourth allocation is made in the cell (3, 2) given by $X_{32} = \text{Min}\ (7, 9 - 6) = 3.$

The fifth allocation is made in the cell (3, 3) with magnitude $X_{33} = \text{Min}\ (7 - 3, 14) = 4.$

The final allocation is made in the cell (4, 3) with magnitude X_{43} = Min (14, 18 – 4) = 14.

Hence, we get the initial basic feasible solution to the given TP and is given by,

$$X_{11} = 5;\ X_{21} = 2;\ X_{22} = 6;\ X_{32} = 3;\ X_{33} = 4;\ X_{43} = 14$$

$$\text{Total Cost } = 2 \times 5 + 3 \times 2 + 3 \times 6 + 3 \times 4 + 4 \times 7 + 2 \times 14$$

$$=10 + 6 + 18 + 12 + 28 + 28 = ₹\ 102$$

NOTES

Least Cost or Matrix Minima Method

Step 1: Determine the smallest cost in the cost matrix of the transportation table. Let it be C_{ij}. Allocate X_{ij} = Min (a_i, b_j) in the cell *(i, j)*.

Step 2: If $X_{ij} = a_i$ cross off the *i*th row of the transportation table and decrease b_j by a_i. Then go to Step 3.

If $X_{ij} = b_j$ cross off the *j*th column of the transportation table and decrease a_i by b_j. Go to Step 3.

If $X_{ij} = a_i = b_j$ cross off either the *i*th row or the *j*th column but not both.

Step 3: Repeat Steps 1 and 2 for the resulting reduced transportation table until all the rim requirements are satisfied. Whenever the minimum cost is not unique make an arbitrary choice among the minima.

Example 2.4: Obtain an initial feasible solution to the following TP using matrix minima method.

	D_1	D_2	D_3	D_4	*Supply*
O_1	1	2	3	4	6
O_2	4	3	2	0	8
O_3	0	2	2	1	10
Demand	4	6	8	6	24

Solution: Since $\Sigma a_i = \Sigma b_j = 24$, there exists a feasible solution to the TP using the steps in the least cost method, the first allocation is made in the cell (3, 1) the magnitude being $X_{31} = 4$. This satisfies the demand at the destination D_1 and we delete this column from the table as it is exhausted.

	D_1	D_2	D_3	D_4	*Supply*
O_1	1	2 ⑥	3	4	~~6~~ 0
O_2	4	3	2 ②	0 ⑥	~~8~~ ~~2~~
O_3	0 ④	2	2 ⑥	1	~~10~~ ~~6~~
Demand	~~4~~ 0	~~6~~ 0	~~8~~ ~~2~~ 0	~~6~~ 0	24

The second allocation is made in the cell (2, 4) with magnitude X_{24} = Min (6, 8) = 6. Since it satisfies the demand at the destination D_4, it is deleted from the table. From the reduced table, the third allocation is made in the cell (3, 3) with magnitude X_{33} = Min (8, 6) = 6. The next allocation is made in the cell (2, 3) with magnitude X_{23} of Min (2, 2) = 2. Finally, the allocation is made in the cell (1, 2) with magnitude X_{12} = Min (6, 6) = 6. Now, all the requirements have been satisfied and hence, the initial feasible solution is obtained.

The solution is given by,

$$X_{12} = 6; X_{23} = 2; X_{24} = 6; X_{31} = 4; X_{33} = 6$$

Since the total number of occupied cells = 5 < $m + n - 1$

We get a degenerate solution.

Total Cost = 6 × 2 + 2 × 2 + 6 × 0 + 4 × 0 + 6 × 2

= 12 + 4 + 12 = ₹ 28.

NOTES

Vogel's Approximation Method (VAM)

Vogel Approximation Method (VAM) is used to find the feasible solution for transportation of goods where the solution is either optimal or near to the optimal solution. Typically, this method is used to reduce the transportation costs by interpreting using a mathematical table the transportation costs from one place to another. In the table, the column represents the demand centres while the row represents the supply points. The following are the general steps used in VAM:

Step 1: Identify the minimum and next minimum numbers in a column and repeat the same for the row.

Step 2: The above step is repeated for all other columns and rows.

Step 3: Now, subtract the two numbers identified for each column and each row such that the difference is positive.

Step 4: Identify the maximum difference among all the rows and also among all the columns.

Step 5: Assign all the demand units for that minimum number in that column which has got the maximum difference (repeat the same for the row).

Step 6: Remove that column and row completely and repeat the above process until all the demand units are filled up completely.

Vogel's Approximation Method (VAM) also takes costs into account in allocation.

The steps involved in VAM method for finding the initial solution are as follows.

Step 1: Find the penalty cost, namely the difference between the smallest and next smallest costs in each row and column.

Step 2: Among the penalties as found in Step (1) choose the maximum penalty. If this maximum penalty is more than one (i.e., if there is a tie) choose any one arbitrarily.

Step 3: In the selected row or column as by Step (2) find out the cell having the least cost. Allocate to this cell as much as possible depending on the capacity and requirements.

Step 4: Delete the row or column which is fully exhausted. Again, compute the column and row penalties for the reduced transportation table and then go to Step (2). Repeat the procedure until all the rim requirements are satisfied.

Note: If the column is exhausted, then there is a change in row penalty and vice versa.

Example 2.5: Find the initial basic feasible solution for the following transportation problem using VAM.

		Destination				
		D_1	D_2	D_3	D_4	***Supply***
Origin	O_1	11	13	17	14	250
	O_2	16	18	14	10	300
	O_3	21	24	13	10	400
	Demand	200	225	275	250	950

Solution: Since $\Sigma a_i = \Sigma b_j = 950$, the problem is balanced and there exists a feasible solution to the problem.

First, we find the row and column penalty P_I as the difference between the least and the next least cost. The maximum penalty is 5. Choose the first column arbitrarily. In this column, choose the cell having the least cost name (1, 1). Allocate to this cell with minimum magnitude (i.e., Min (250, 200) = 200.) This exhausts the first column. Delete this column. Since a column is deleted, then there is a change in row penalty P_{II} and column penalty P_{II} remains the same. Continuing in this manner, we get the remaining allocations as given in the following table below.

I Allocation

	D_1	D_2	D_3	D_4	*Supply*	P_I
O_1	11 200	13	17	14	50 ~~250~~	2
O_2	16	18	14	10	300	4
O_3	21	24	13	10	400	3
Demand	~~200~~ 0	225	275	250		
P_I	5↑	5	1	0		

II Allocation

	D_2	D_3	D_4	*Supply*	P_{II}
O_1	13 50	17	14	~~50~~	1
O_2	18	14	10	300	4
O_3	24	13	10	400	3
Demand	~~225~~ 175	275	250		
P_{II}	5↑	1	0		

III Allocation

	D_2	D_3	D_4	*Supply*	P_{III}
O_2	18 175	14	10	~~300~~ 125	4
O_3	24	13	10	400	3
Demand	~~175~~ 0	275	250		
P_{III}	6↑	1	0		

IV Allocation

	D_3	D_4	*Supply*	P_{IV}
O_2	14	10 125	~~125~~ 0	4 ←
O_3	13	10	400	3
Demand	275	~~250~~ 125		
P_{IV}	1	0		

V Allocation

	D_3	D_4	*Supply*	P_V
O_3	13 275	10	~~400~~ 125	3
Demand	~~275~~ 0	125		
P_V	13↑	10		

VI Allocation

	D_4	*Supply*	P_{VI}
O_3	10 125	~~125~~ 0	10 ←
Demand	~~125~~ 0		
P_{VI}	10		

Finally, we arrive at the initial basic feasible solution which is shown in the following table.

	D_1	D_2	D_3	D_4	*Supply*
O_1	11 200	13 50	17	14	250
O_2	16	18 175	14	10 125	300
O_3	21	24	13 275	10 125	400
Demand	200	225	275	250	

There are 6 positive independent allocations which equals to $m + n - 1 = 3 + 4 - 1$. This ensures that the solution is a non-degenerate basic feasible solution.

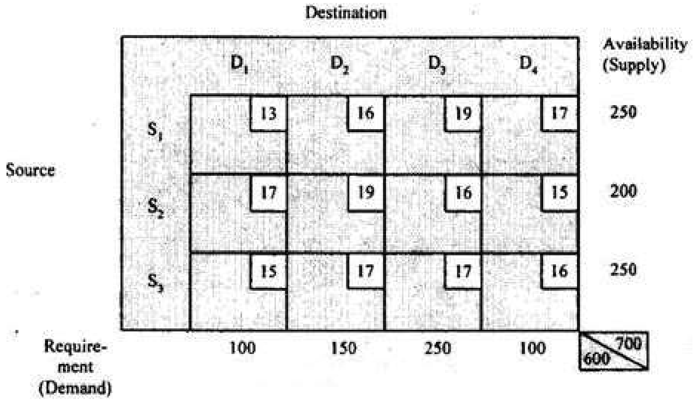
$\therefore$ Transportation Cost,

$= 11 \times 200 + 13 \times 50 + 18 \times 175 + 10 \times 125 + 13 \times 275 + 10 \times 125$

= ₹ 12,075.

Examples to Find Solution of Transportation Problem Using Vogel's Approximation Method or VAM

Example 2.6: An organization has four destinations (D_1, D_2, D_3 and D_4) and three sources (S_1, S_2 and S_3) for supply of goods. The transportation cost per unit is given below. The total availability is 700 units which exceeds the cumulative demand of 600 units. Find the optimal transportation scheme for this condition using the Vogel's Approximation Method or VAM.



		Destination				
		D_1	D_2	D_3	D_4	Availability (Supply)
Source	S_1	13	16	19	17	250
	S_2	17	19	16	15	200
	S_3	15	17	17	16	250
Requirement (Demand)		100	150	250	100	700 600

Solution: The solution is obtained as follows.

***Step 1*:** First check for balance of supply and demand

Supply = 250 + 200 + 250 = 700 units

Demand = 100 + 150 + 250 + 100 = 600 units

Decision Rule

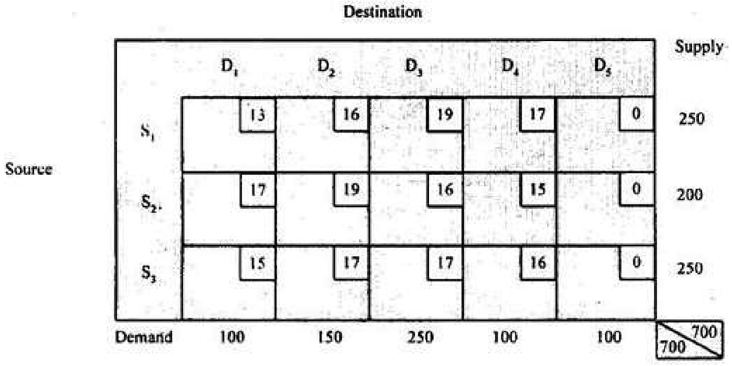
(*i*) If Supply = Demand then go to next step.

(*ii*) If Supply > Demand then add a 'Dummy Destination' with zero transportation cost.

(*iii*) If Supply < Demand then add a 'Dummy Source' with zero transportation cost.

In the given problem, **Supply > Demand**.

Hence, we add a 'Dummy Destination' say D_5 with zero transportation cost and balance demand which is difference in supply and demand (= 100 units). The initial transportation matrix is now formulated with transportation cost in each route. Each cell of the transportation matrix represents a possible route. In the following table, dummy column is introduced for balancing the supply and demand.



		Destination					
		D_1	D_2	D_3	D_4	D_5	Supply
Source	S_1	13	16	19	17	0	250
	S_2.	17	19	16	15	0	200
	S_3	15	17	17	16	0	250
Demand		100	150	250	100	100	700 700

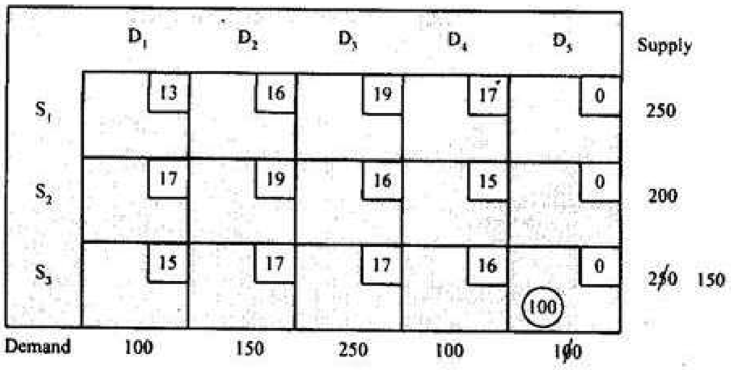
NOTES

***Step 2*:** (*i*) Decide the nature of problem, i.e., minimization of transportation cost.

(*ii*) Make initial assignment using the Vogel's approximation method.

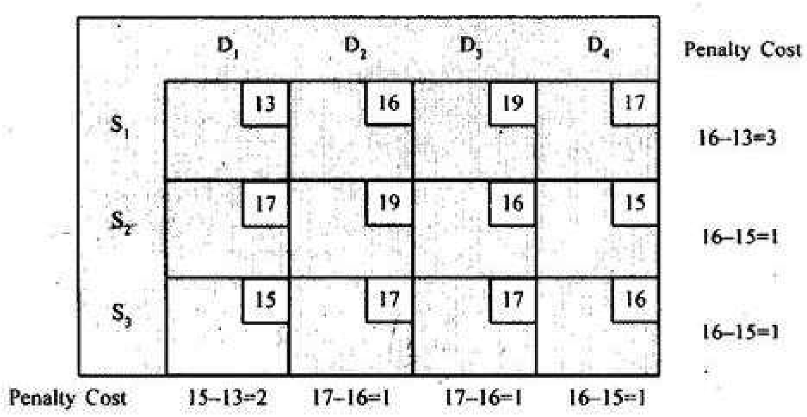
(*i*) Select the lowest transportation cost route in the initial matrix. For example, it is route S_1D_5, S_2D_5 and S_3D_5 in the given problem with zero transportation cost. Allocate the minimum of remaining balance of supply (in last column) and demand (in last row).

In this method, we calculate the difference between the two least-cost routes for each row and column. The difference is called as penalty cost for not using the least-cost route. Following table shows the first calculation of 'Penalty' cost in VAM.



	D_1	D_2	D_3	D_4	D_5	Supply
S_1	13	16	19	17	0	250
S_2	17	19	16	15	0	200
S_3	15	17	17	16	0 100	2~~5~~0 150
Demand	100	150	250	100	1~~0~~0	

Highest of all calculated penalty costs is for S_3 and S_2. Therefore, allocation is to be made in row of source S_3. The route or cell which should be selected should be the lowest cost of this row, i.e., the route S_3D_5. Hence, first allocation in Vogel's method is as follows.

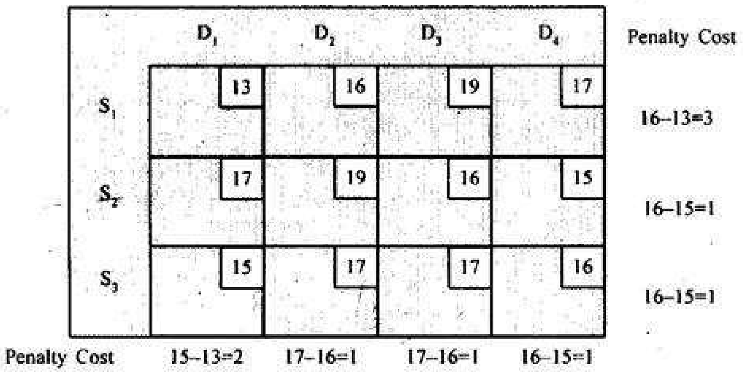


	D_1	D_2	D_3	D_4	Penalty Cost
S_1	13	16	19	17	16–13=3
S_2	17	19	16	15	16–15=1
S_3	15	17	17	16	16–15=1
Penalty Cost	15–13=2	17–16=1	17–16=1	16–15=1	

With the first allocation, destination D_5 is used. Leave out this column and develop the remaining matrix for calculating the penalty cost. We obtain the following matrix.

NOTES

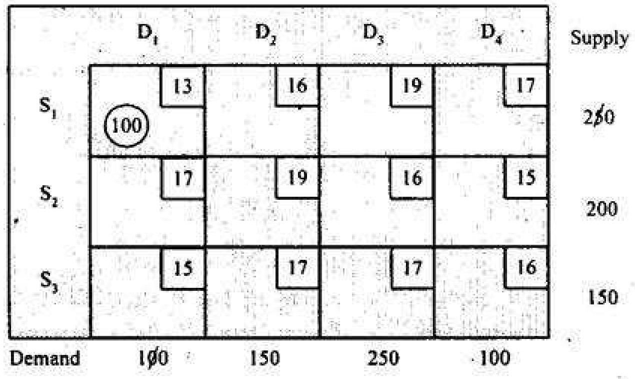
Now for this, source S_1 has highest penalty cost. For this row, the least cost route is S_1D_1. Hence, next assignment is due in this route:



	D_1	D_2	D_3	D_4	Penalty Cost
S_1	13	16	19	17	16–13=3
S_2	17	19	16	15	16–15=1
S_3	15	17	17	16	16–15=1
Penalty Cost	15–13=2	17–16=1	17–16=1	16–15=1	

Second Calculation of Penalty Cost in VAM

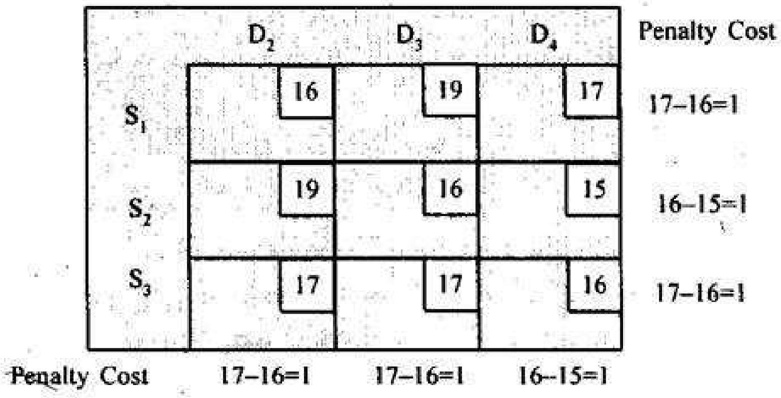
Second allocation in Vogel's method is obtained as follows:



	D_1	D_2	D_3	D_4	Supply
S_1	13 100	16	19	17	2~~5~~0 1~~5~~0
S_2	17	19	16	15	200
S_3	15	17	17	16	150
Demand	1~~0~~0	150	250	100	

After second allocation, since destination D_1 is used, leave this column and proceed for calculation of next penalty cost. Allocation is done in route S_1D_2. Since there is tie between all routes, break the tie by arbitrarily selecting any route, for example S_1D_2 in this case.

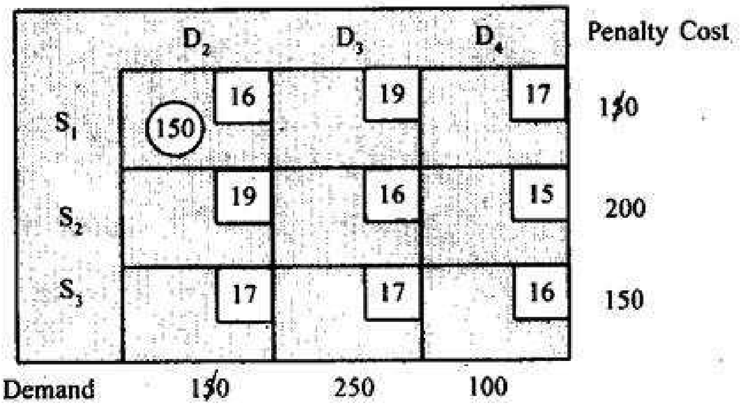
Third Calculation of Penalty Cost in VAM



	D_2	D_3	D_4	Penalty Cost
S_1	16	19	17	17–16=1
S_2	19	16	15	16–15=1
S_3	17	17	16	17–16=1
Penalty Cost	17–16=1	17–16=1	16–15=1	

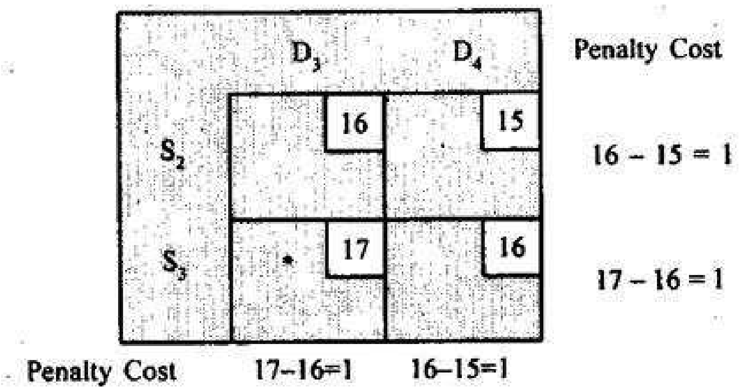
NOTES

Third Allocation in VAM



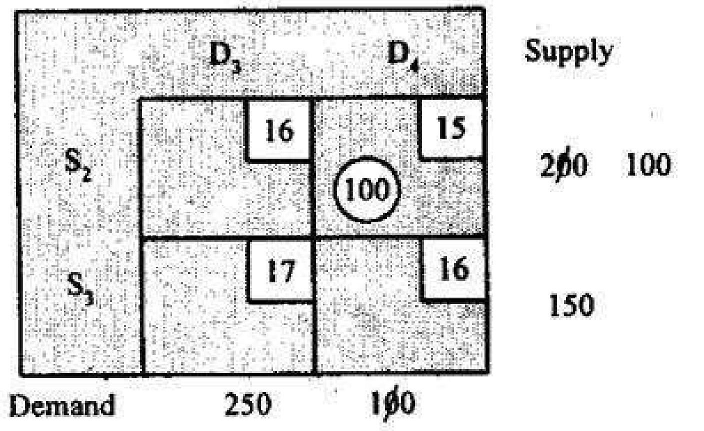
	D_2	D_3	D_4	Penalty Cost
S_1	150 16	19	17	1~~5~~0
S_2	19	16	15	200
S_3	17	17	16	150
Demand	1~~5~~0	250	100	

Fourth Calculation of Penalty Cost in VAM



	D_3	D_4	Penalty Cost
S_2	16	15	16 − 15 = 1
S_3	17	16	17 − 16 = 1
Penalty Cost	17−16=1	16−15=1	

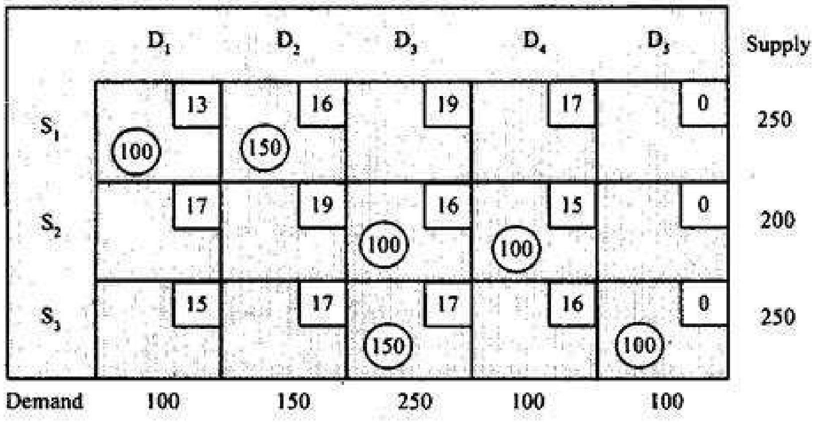
Fourth Allocation in VAM



	D_3	D_4	Supply
S_2	16	100 15	2~~0~~0 100
S_3	17	16	150
Demand	250	1~~0~~0	

With the fourth allocation, column D_4 is used. In the only left column D_3, the allocations of 100 units and 150 units are done in route S_2D_3 and S_4D_3, respectively. Thus, we obtain the following allocations using the Vogel's Approximation Method or VAM.

Final Allocation Through Vogel's Method



	D₁	D₂	D₃	D₄	D₅	Supply
S₁	13 100	16 150	19	17	0	250
S₂	17	19	16 100	15 100	0	200
S₃	15	17	17 150	16	0 100	250
Demand	100	150	250	100	100	

The initial cost for this allocation is:

(13 × 100 + 16 × 150 + 16 × 100 + 15 × 100 + 17 × 150 + 0 × 100) or equal to **₹ 9350**

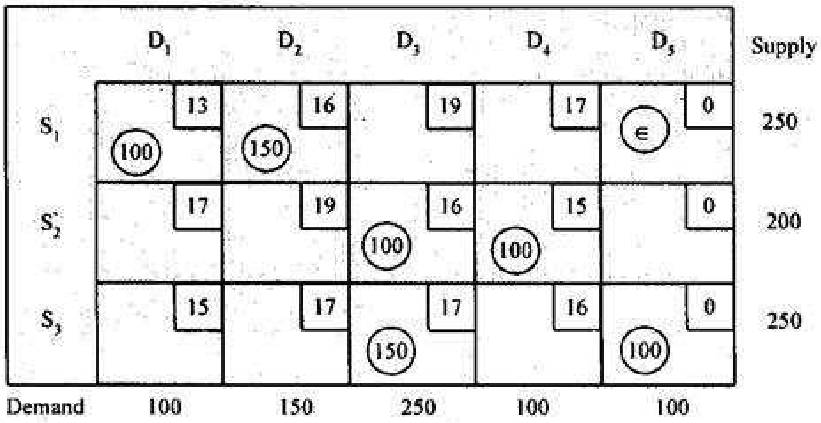
Step 3: Verify for degeneracy, $(m + n - 1) = 7$.

Number of filled cell = 6, which is one less than $(m + n + 1)$. Hence, go to Step 4 for removing the degeneracy.

Step 4: Now we allocate in the least cost unfilled cell. This cell is route S_1D_5 or S_2D_5. Select route S_1D_5. We obtain the following matrix after removing degeneracy.

NOTES

Final Allocation After Removing Degeneracy in Vogel's Method



	D₁	D₂	D₃	D₄	D₅	Supply
S₁	13 100	16 150	19	17	0 ∈	250
S₂	17	19	16 100	15 100	0	200
S₃	15	17	17 150	16	0 100	250
Demand	100	150	250	100	100	

Optimization of Initial Assignment

The initial feasible assignment done by using Vogel's approximation method does not guarantee optimal solution. Hence, next step is to check the optimality of the initial solution.

Step 5: Check the optimality of the initial solution. For this, calculate the opportunity cost of un-occupied routes.

First, we start with any row (or column). Select row 1, i.e., source S_1. For this row define row value, $u_1 = 0$. Now consider all filled routes of this row. For these routes, calculate column values v using following equation:

$$u_1 + v_1 = C_{ij} \text{ (For any filled route)}$$

Where $\quad u_1$ = Row Value

v_j = Column Value

C_{ij} = Unit Cost of Assigned Route

Once first set of column values is known, say in this case v_j is known, locate other routes of filled cells in these columns. Calculate next of u_i or v_j values using the above equation. In this method, for all rows and columns, u_i and v_j values are determined for a non-degenerate initial solution.

Step 6: Check the optimality.

Calculate the opportunity of non-allocated or unfilled routes. For this, use the following equation:

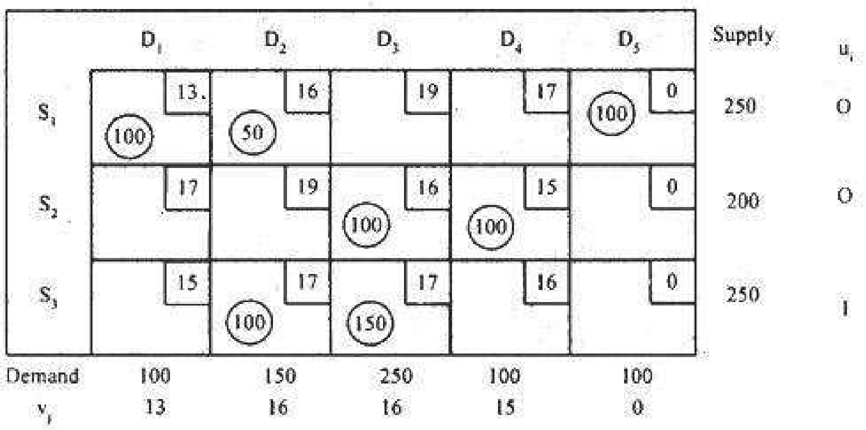
$$\textbf{Opportunity Unassigned Route} = \boldsymbol{u_i + v_j - C_{ij}}$$

Where, $\quad u_i$ = Row Value

v_j = Column Value

C_{ij} = Unit Cost of Unassigned Route

If the opportunity cost is negative for all unassigned routes, the initial solution is optimal. If in case any of the opportunity costs is positive, then go to next step.



	D_1	D_2	D_3	D_4	D_5	Supply	u_i
S_1	13. 100	16 50	19	17	0 100	250	0
S_2	17	19	16 100	15 100	0	200	0
S_3	15	17 100	17 150	16	0	250	1
Demand	100	150	250	100	100		
v_j	13	16	16	15	0		

***Step 7*: ** Make a loop of horizontal and vertical lines which joins some filled routes with the unfilled route, which has a positive opportunity cost. Note that all the corner points of the loop are either filled cells or positive opportunity cost unassigned cells.

For this, we start with row, S_1 and take $u_1 = 0$. Now S_1D_1, S_1D_2, and S_1D_5 are filled cells. Hence, for filled cells ($v_j = C_{ij} - u_i$).

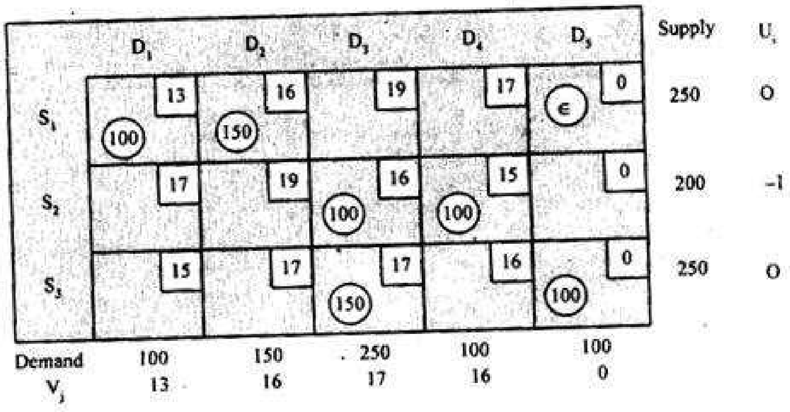
$$v_1 = 13 - 0 = 13$$

$$v_2 = 16 - 0 = 16$$

$$v_5 = 0 - 0 = 0$$

The optimality of Vogel's method's initial solution is as follows.

Calculation of u_i and v_j for Vogel Approximation Method's Initial Solutions



	D₁	D₂	D₃	D₄	D₅	Supply	Uᵢ
S₁	13 100	16 150	19	17	0 ε	250	0
S₂	17	19	16 100	15 100	0	200	–1
S₃	15	17	17 150	16	0 100	250	0
Demand	100	150	250	100	100		
Vⱼ	13	16	17	16	0		

Opportunity cost of above assignment using VAM is as follows:

Unassigned Route	Opportunity Cost ($u_i + v_j – C_{ij}$)
S_1D_3	0 + 17 – 19 = –2
S_1D_4	0 + 16 – 17 = –1
S_2D_1	–1 + 13 – 17 = –5
S_2D_2	–1 + 16 – 19 = –4
S_2D_5	–1 + 0 – 0 = –1
S_3D_1	0 + 13 – 15 = –2
S_3D_2	0 + 16 – 17 = –1
S_3D_4	0 + 16 – 16 = 0

Since all opportunity costs are negative or zero, the initial assignment of Vogel's solution is optimal with total cost of ₹ 9350.

Example 2.7: Distances between factory and its warehouses and demand at each warehouse are given in the following table. Calculate the values of penalty to all the rows and columns for the reduced transportation problem and repeat the same procedure till the entire requirement has been met. Solve this problem using Vogel's Approximation Method or VAM.

Transportation Table

Factory/Warehouse	W1	W2	W3	Supply
F1	16	22	14	200
F2	18	14	18	150
F3	8	14	16	100
Demand	**175**	**125**	**150**	

Solution: The solution is obtained as follows.

Step 1: Compute the penalty for each row and column of the transportation problems. The penalty for the first row is, (16 - 14) = 2. Similarly the values of penalty for the second and the third row are 4 and 6.

	W_1	**W_2**	**W_3**	**Supply**	**Penalty**
F_1	16	22	14	200	**2**
F_2	18	14	18	150	**4**
F_3	8	14	16	100	**6**
Demand	175	125	150		
	8	**0**	**2**		

Similarly, the values of penalty for the first, second and the third columns are 8, 0 and 2, respectively.

Step 2: Identify the row or column with the largest penalty value. In this case, the first column with a penalty value is 8.

Step 3: The cell with the least cost is chosen and the possible number of goods is assigned to that cell. Therefore, assign 100 to the cell (F_3, W_1).

Step 4: If the remaining row supply or column demand is zero, remove that row/column.

Now, the transportation problem can be reduced as illustrated in the following table:

	W_1	**W_2**	**W_3**	**Supply**	**Penalty**
F_1	16	22	14	200	**2**
F_2	18	14	18	150	**4**
Demand	**75**	**125**	**150**		
Penalty	**2**	**8**	**4**		

Step 5: The process is repeated for the reduced transportation problem till the entire supply at the factories is assigned to satisfy the demand at different warehouses.

Now, the W_2 column has the highest penalty, i.e., 8. Therefore, assign 125 units to the cell (F_2, W_2) since the cell has the least cost in the W_2 column.

Then the transportation problem can further be reduced as illustrated in the following table:

	W_1	**W_3**	**Supply**	**Penalty**
F_1	16	14	200	**2**
F_2	18	18	25	**0**
Demand	**75**	**150**		
Penalty	**2**	**4**		

Now, the W_3 column has the highest penalty, i.e., 4. Next assign 150 units to the cell (F_1, W_3) since the cell has the least cost. Then remove the W_3 column and the remaining units are assigned to the cells (F_1, W_1) and (F_2, W_1). Thus, 50 units are assigned to the cell (F_1, W_1) and 25 units to the cell (F_2, W_1).

Since the number of cells occupied is 5, i.e., (3+3−1), the solution obtained is a feasible solution. Thus, the cost obtained using VAM is:

(50 × 16) + (25 × 18) + (100 × 8) + (125 × 14) + (150 × 14) = ₹5,900.

2.3 ASSIGNMENT PROBLEM

The assignment problem is used to find the best possible assignment for the given situations.

Basics of Assignment Problem

The assignment problem is one of the fundamental combinatorial optimization problems. It helps to find a maximum weight identical in nature in a weighted bipartite graph. The assignment problem is also termed as a special case of transportation problem.

Suppose there are *n* jobs to be performed and *n* persons are available for doing these jobs. Assume that each person can do each job at a time, though with varying degrees of efficiency. Let C_{ij} be the cost if the *i*th person is assigned to the *j*th job. The solution to the problem is to find an assignment (which job should be assigned to which person on one-one basis) so that the total cost of performing all jobs is minimum. Problems of this kind are known as assignment problems.

The assignment problem can be stated in the form of $n \times n$ cost matrix $[C_{ij}]$ of real numbers as given in the following table:

$$\begin{array}{cc|ccccc|}
 & & & & \textit{\textbf{Jobs}} & & \\
 & & 1 & 2 & 3\ldots & j\ldots & n \\
 & 1 & C_{11} & C_{12} & C_{13}\cdots & C_{1j}\cdots & C_{1n} \\
 & 2 & C_{21} & C_{22} & C_{23}\cdots & C_{2j}\cdots & C_{2n} \\
 & 3 & C_{31} & C_{32} & C_{33}\cdots & C_{3j}\cdots & C_{3n} \\
\textit{\textbf{Persons}} & & & & & & \\
 & i & C_{i1} & C_{i2} & C_{i3}\cdots & C_{ij}\cdots & C_{in} \\
 & & \vdots & & & & \\
 & n & C_{n1} & C_{n2} & C_{n3}\cdots & C_{nj}\cdots & C_{nn}
\end{array}$$

Mathematical Formulation of the Assignment Problem

Mathematically, the assignment problem can be stated as follows:

$$\text{Minimize, } Z = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}x_{ij} \quad i = 1, 2, \ldots, n$$

$$j = 1, 2, \ldots, n$$

Subject to the restrictions:

NOTES

$$x_{ij} = \begin{cases} 1 \text{ if the } i\text{th person is assigned } j\text{th job} \\ 0 \text{ if not} \end{cases}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ (one job is done by the } i\text{th person)}$$

and $\sum_{i=1}^{n} x_{ij} = 1$ (only one person should be assigned the *j*th job)

Where x_{ij} denotes that the *j*th job is to be assigned to the *i*th person.

2.3.1 Job Sequence Problem

Job sequencing is basically the planning of the jobs in sequential manner and is an essential part of any work. Without proper planning and scheduling one can not achieve the desired output and profit. For sequencing a job, generally the two techniques are used termed as Priority Rules and Johnson's Rules. Priority rules give the guidelines for properly sequencing the job, where as Johnson's rule is used to minimize the completion time for a set of jobs to be done on two different machines. Using these rules one can assign jobs and maximize product and profit.

Basic Characteristics of Job Sequencing

1. Only one single job should be scheduled for a machine at a time.
2. Do not stop the process in between before completion.
3. New processing can be started after the completion of the previous processing.
4. Any job is scheduled for processing as per the order and due date requirements.
5. If the jobs are transferred from one machine to another due to some reason, then the time involved in transferring the jobs is considered negligible.

Priority Rules: These rules are used to get specific guidelines for job sequencing. The rules do not consider job setup cost and time while analysing processing times. In it job processing time and due dates are given importance because the due dates are fixed to give delivery in time to the customers. The rules are very useful for process-focussed amenities, for example health clinics, print shops and manufacturing industries. Hence, priority rules minimize the time for completing a job, sequences the jobs in the organization, checks if any job is late and maximizes resource utilization. The most popular priority rules are as follows:

- **First Come First Serve (FCFS):** The job to be processed first is the job that turned up first in the organization.
- **Earliest Due Date (EDD):** The job to be processed first is the job that has earliest due date.
- **Shortest Processing Time (SPT):** The job to be processed first and completed is the job that is shortest in nature; in other words the job can be processed in short time.
- **Longest Processing Time (LPT):** The job to be processed first is the job that is very important or of high priority though it can take longer processing time.

- **Critical Ratio (CR):** The job to be processed first is analysed on the basis of critical ratio, which is an index number calculated from time remaining until due date divided by the remaining work time.

Johnson's Rule: This rule is applied to minimize the completion time for a set of jobs that are to be processed on two different machines or at two consecutive work stations. The main objectives of the rules are,

- To minimize the processing time while sequencing a set of jobs on two different machines or work stations.
- To minimize the complete idle time on the processing machines.
- To minimize the flow time of the job, i.e., from the start of the first job until the completion of the last job.

Necessary Conditions for Johnson's Rules: The necessary conditions to efficiently complete the processing of the jobs are as follows:

- Knowledge about job time for each job at the specific work station.
- Job time must not depend on sequencing of jobs.
- All the jobs to follow the predefined work sequence.
- Avoid job priority.

Four Steps Johnson's Rule: The following are the important Four Steps in Johnson's rule:

Step 1: List all the jobs and the processing time of each machine to which these jobs are scheduled.

Step 2: Choose the job which has the shortest processing time. If the shortest time has been scheduled on the first machine or work station then the job is selected first for processing. In case the shortest time is scheduled on the second machine or work station then the job is processed at the end.

Step 3: After scheduling the job for processing go to Step 4.

Step 4: Repeat Step 2 again to schedule the processing of remaining jobs and fill the sequence columns towards the centre till all the jobs are scheduled.

The following example will help you to understand how the sequences are scheduled.

For example, there are five jobs to be done at a factory and each job must be processed through two work stations at two different machines, drill machine and lathe machine. Using Johnson's rule we can schedule the sequence of jobs.

The time (in hours) for processing each job is given in the following table:

Jobs	***Work Station 1 (Drill)***	***Work Station 2 (Lathe)***
A	5	2
B	3	6
C	9	4
D	12	8
E	8	14

Using the Steps of Johnson's rule, the job processing sequences are scheduled as follows:

Step 1: In the given table, the job with the shortest processing time is job **A**, in work station 2 (with a time of 2 hours). Because it is at the second work station, schedule **A** last.

				A

Step 2: Next shortest time is of job **B** (with a time of 3 hours). Because it is at the first work station, schedule it at first priority and eliminate it from the list.

B				**A**

Step 3: The next shortest time is of job **C** (with a time of 4 hours), but it is at the second work station. Therefore, place it at last before **A**.

B			**C**	**A**

Step 4: There is a tie between job **D** (with a time of 8 hours at work station 2) and job **E** (with a time of 8 hours at work station 1) for the shortest remaining job. Because job **E** is at the first work station, so place it first after job **B**. Then place job **D** in the last sequencing position. You will get the job sequence schedule as follows:

B	**E**	**D**	**C**	**A**

The final sequential times at both the work stations will be:

Jobs	***B***	***E***	***D***	***C***	***A***
Work Station 1 (Drill)	3	8	12	9	5
Work Station 2 (Lathe)	6	14	8	4	2

Check Your Progress

1. What is a transportation problem?
2. How are transportation problems presented?
3. Give the mathematical presentation of a transportation problem.
4. What is an assignment problem?
5. How is an assignment problem presented to find the minimum cost?

2.4 NETWORK ANALYSIS

Network scheduling is a technique used for planning, and scheduling large projects in the field of construction, maintenance, fabrication, purchasing computer system, etc. The technique is a method of minimizing the trouble spots such as production, delays and interruptions, by determining critical factors and coordinating various parts of the overall job.

There are two basic planning and control technique that utilize a network to complete a predetermined project or schedule. These are Programme Evaluation Review Technique (PERT) and Critical Path Method (CPM).

A project is defined as a combination of interrelated activities all of which must be executed in a certain order for its completion.

The work involved in a project can be divided into three phases corresponding to the management functions of planning, scheduling and control.

Planning: This phase involves setting the objectives of the project and the assumptions to be made. Also it involves the listing of tasks or jobs that must be performed to complete a project under consideration. In this phase, men, machines and materials required for the project, in addition to the estimates of costs and duration of the various activities of the project, are also determined.

Scheduling: This consists of laying the activities according to the precedence order and determining,

(*i*) The start and finish times for each activity.

(*ii*) The critical path on which the activities require special attention.

(*iii*) The slack and float for the non-critical paths.

Controlling: This phase is exercised after the planning and scheduling, which involves the following:

(*i*) Making periodical progress reports.

(*ii*) Reviewing the progress.

(*iii*) Analysing the status of the project.

(*iv*) Management decisions regarding updating, crashing and resource allocation.

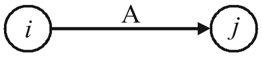
2.4.1 Basic Terms

To understand the network techniques one should be familiar with few basic terms of which both CPM and PERT are special applications.

Network: It is the graphic representation of logically and sequentially connected arrows and nodes representing activities and events of a project. Networks are also called arrow diagram.

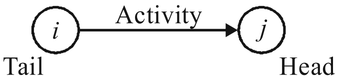
Activity: An activity represents some action and is a time consuming effort necessary to complete a particular part of the overall project. Thus, each and every activity has a point of time where it begins and a point where it ends.

It is represented in the network by an arrow,

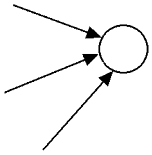


Here, *A* is called the activity.

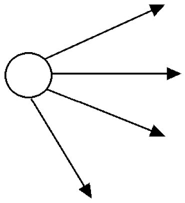
Event: The beginning and end points of an activity are called events or nodes. Event is a point in the time and does not consume any resource. It is represented by a numbered circle. The head event called the *j*th event has always a number higher than the tail event called the *i*th event.



Merge and Burst Events: It is not necessary for an event to be the ending event of only one activity but can be the ending event of two or more activities. Such event is defined as a merge event.



If the event happens to be the beginning event of two or more activities it is defined as a burst event.



Preceding, Succeeding and Concurrent Activities: Activities, which must be accomplished before a given event can occur are termed as *preceding activities.*

Activities, which cannot be accomplished until an event has occurred are termed as *succeeding activities.*

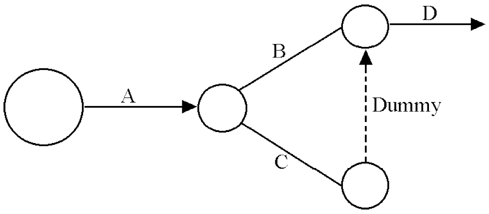
Activities, which can be accomplished concurrently are known as *concurrent activities.*

This classification is relative, which means that one activity can be preceding to a certain event, and the same activity can be succeeding to some other event or it may be a concurrent activity with one or more activities.

Dummy Activity: Certain activities, which neither consumes time nor resources but are used simply to represent a connection or a link between the events are known as dummies. It is shown in the network by a dotted line. The purpose of introducing dummy activity is as follows:

(*i*) To maintain uniqueness in the numbering system as every activity may have distinct set of events by which the activity can be identified.

(*ii*) To maintain a proper logic in the network.

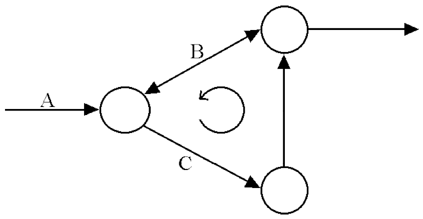


2.4.2 Common Errors

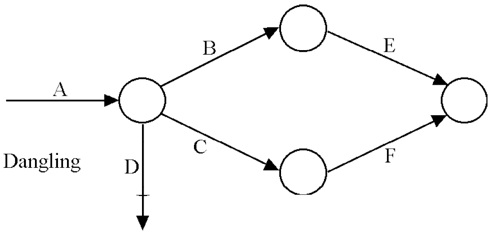
Following are the three common errors in a network construction:

Looping (Cycling): In a network diagram looping error is also known as cycling error. Drawing an endless loop in a network is known as error of looping. A loop can be formed if an activity were represented as going back in time.

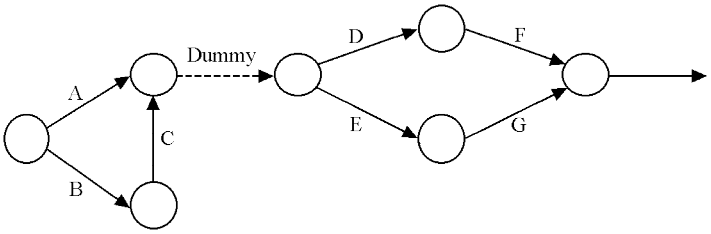
NOTES



Dangling: To disconnect an activity before the completion of all the activities in a network diagram is known as dangling.



Redundancy: If a dummy activity is the only activity emanating from an event and which can be eliminated is known as redundancy.



2.4.3 Rules of Network Construction

There are a number of rules in connection with the handling of events and activities of a project network that should be followed.

(*i*) Try to avoid arrows which cross each other.

(*ii*) Use straight arrows.

(*iii*) No event can occur until every activity preceding it has been completed.

(*iv*) An event cannot occur twice, i.e., there must be no loops.

(*v*) An activity succeeding an event cannot be started until that event has occurred.

(*vi*) Use arrows from left to right. Avoid mixing two directions. Vertical and standing arrows may be used if necessary.

(*vii*) Dummies should be introduced if it is extremely necessary.

(*viii*) The network has only one entry point called the start event and one point of emergence called the end or terminal event.

Numbering the Events (Fulkerson's Rule)

After the network is drawn in a logical sequence every event is assigned a number. The number sequence must reflect the flow of the network. In numbering the events the following rules should be observed:

(*i*) Event numbers should be unique.

(*ii*) Event numbering should be carried out on a sequential basis from left to right.

(*iii*) The initial event which has all outgoing arrows with no incoming arrow is numbered as 1.

(*iv*) Delete all arrows emerging from all the numbered events. This will create at least one new start event out of the preceding events.

(*v*) Number all new start events 2, 3 and so on. Repeat this process until all the terminal event without any successor activity is reached. Number the terminal node suitably.

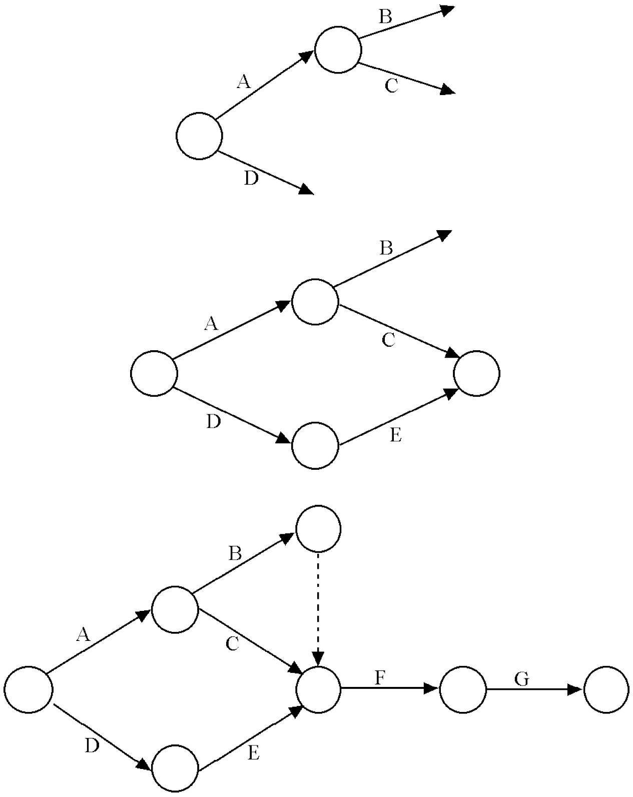
Note: The head of an arrow should always bear a number higher than the one assigned to the tail of the arrow.

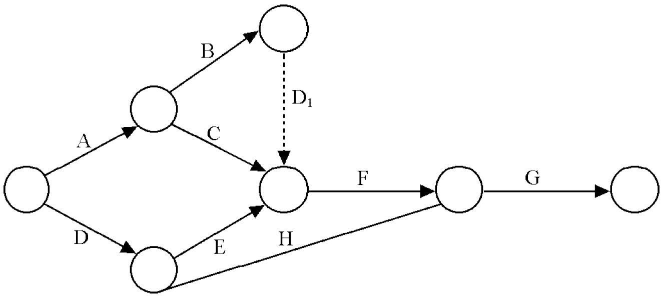
2.4.4 Construction of Network

Example 2.8: Construct a network for the project whose activities and their precedence relationships are as given below:

Activities	***A***	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***
Immediate Predecessor	–	A	A	–	D	B,C,E	F	D	G,H

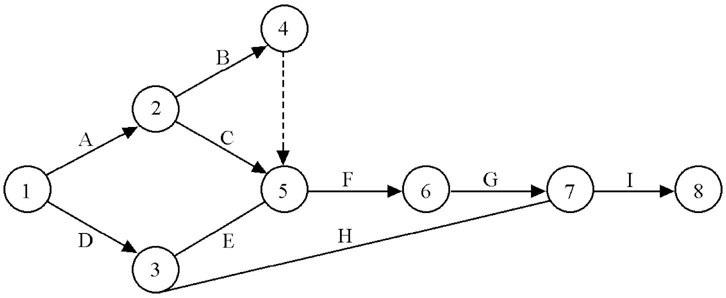
Solution: From the given constraints, it is clear that *A*, *D* are the starting activity and *I* the terminal activity. *B, C* are starting with the same event and are both the predecessors of the activity *F*. Also *E* has to be the predecessor of both *F* and *H*. Hence, we have to introduce a dummy activity.





D_1 is the dummy activity.

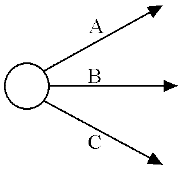
Finally we have the following network.

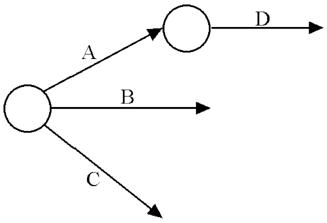


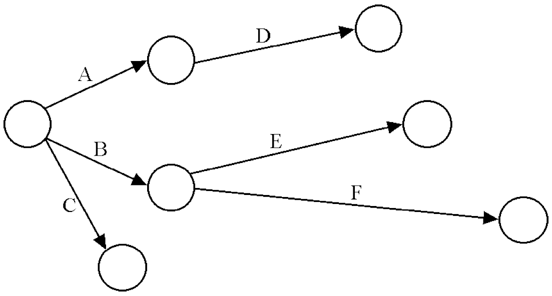
Example 2.9: Construct a network for each of the projects whose activities and their precedence relationships are given below.

Activity	***A***	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***	***J***	***K***
Predecessor	–	–	–	A	B	B	C	D	E	H,I	F,G

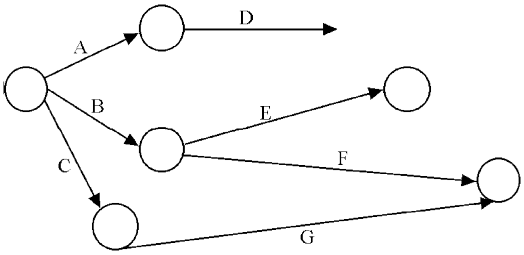
Solution: *A, B, C* are the concurrent activities as they start simultaneously. *B* becomes the predecessor of activity *E* and *F*. Since the activities *J, K* have two preceding activities, dummy may be introduced (if possible).

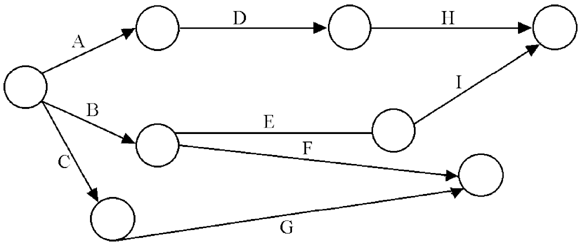




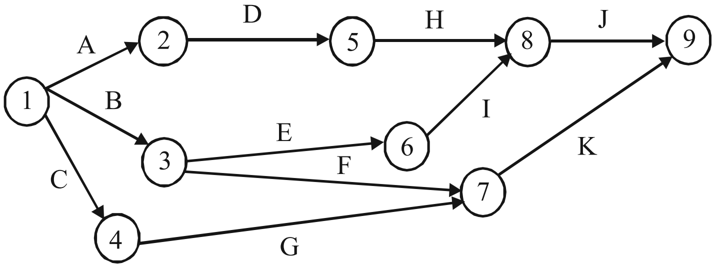


NOTES





Finally we have,



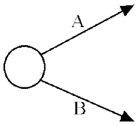
Example 2.10: Construct a network of the project whose activities are given as below.

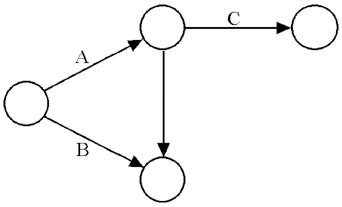
A<C, D, I; *B<G, F*; *D<G, F*; *F<H, K*; *G, H<J*; *I, J, K<E*

Solution: Given *A<C* which means that *C* cannot be started until *A* is completed, i.e., *A* is the preceding activity to *C*. The above constraints can be given in the following table:

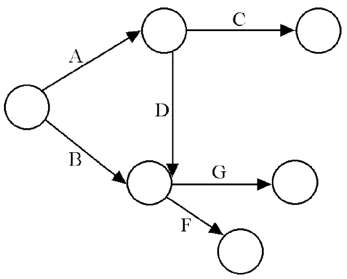
Activity	***A***	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***	***J***	***K***
Predecessor	–	–	A	A	I,J,K	B,D	B,D	F	A	G,H	F

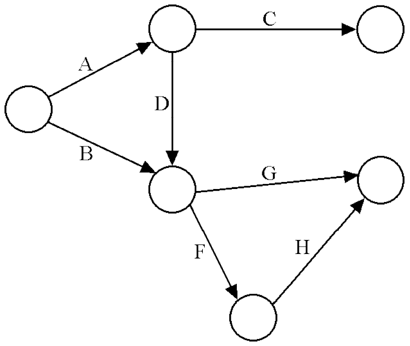
A, *B* are the starting activity, and *E* is the terminal activity.

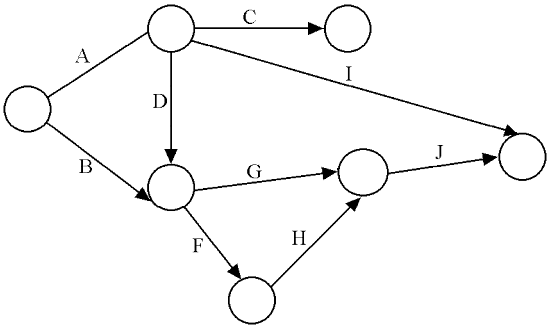




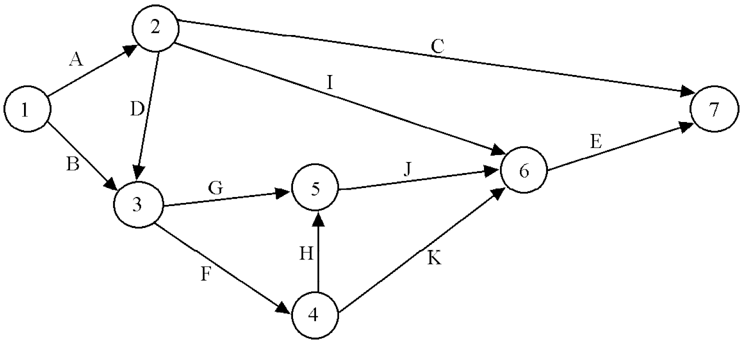
NOTES







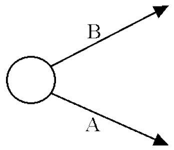
Finally we have,

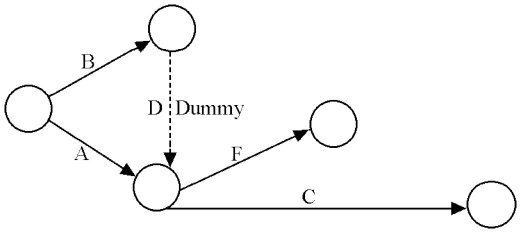


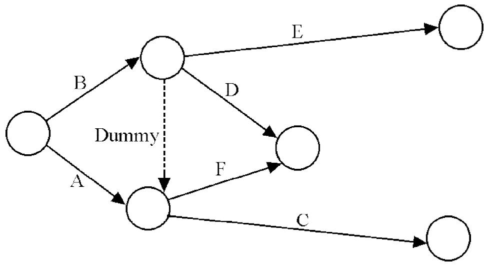
Example 2.11: Construct the network for the project whose activities and precedence relationship is given below. Show also the dummy activity.

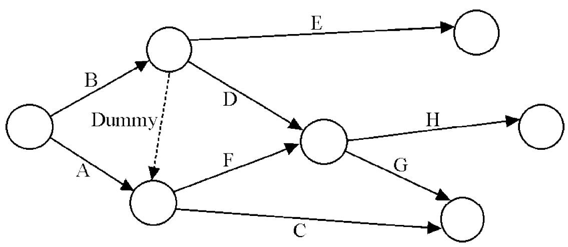
Activities	***A***	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***
Immediate Predecessor	–	–	A,B	B	B	A,B	F,D	F,D	C,G

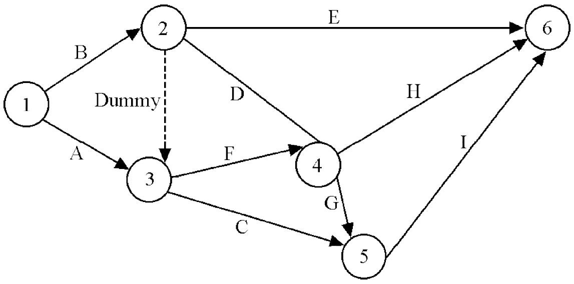
Solution: *A*, *B* are concurrent activities as they start simultaneously. *I* is the terminal activity. Since the activities *C* and *F* are coming from both the activities *A*, *B* we need to introduce a dummy activity.











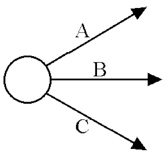
Example 2.12: Make a network of the project having activities and precedence relationship as given below:

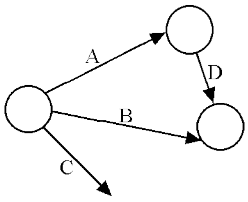
A, *B*, *C* can start simultaneously,

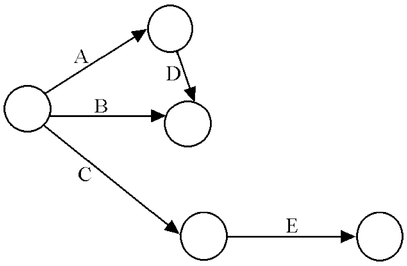
A<D, *I*; *B<G*, *F*; *D<G*, *F*; *C<E*; *E<H*, *K*; *F<H*, *K*; *G*, *H<J*

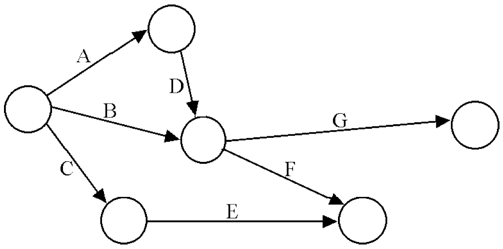
Solution: The above constraints can be formatted into a table.

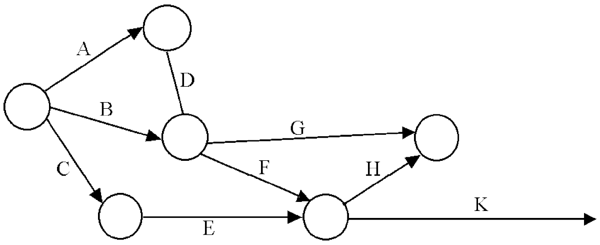
Activity	***A***	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***	***J***	***K***
Predecessor Activity	–	–	–	A	C	B,D	B,D	E,F	A	G,H	E,F

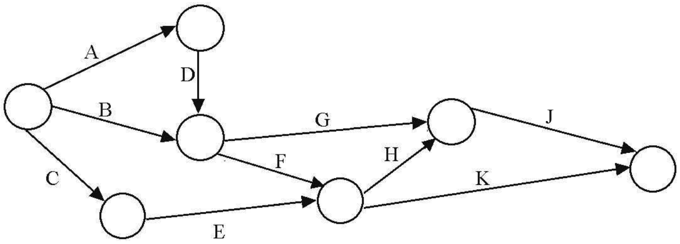


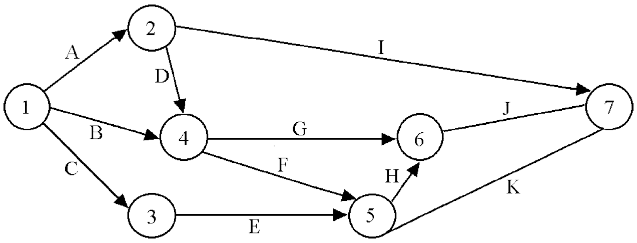












2.4.5 Time Analysis

Once the network of a project is constructed the time analysis of the network becomes essential for planning various activities of the project. An activity time is a forecast of the time an activity is expected to take from its starting point to its completion (under normal conditions).

We shall use the following notation for basic scheduling computations.

(i, j) = Activity (i, j) with Tail Event i and Head Event j

T_{ij} = Estimated Completion Time of Activity (i,j)

$(ES)_{ij}$ = Earliest Starting Time of Activity (i,j)

$(EF)_{ij}$ = Earliest Finishing Time of Activity (i,j)

$(LS)_{ij}$ = Latest Starting Time of Activity (i, j)

$(LF)_{ij}$ = Latest Finishing Time of Activity (i, j)

The basic scheduling computation can be put under the following three groups.

NOTES

Forward Pass Computations (for Earliest Event Time)

Before starting computations, the occurrence time of the initial network event is fixed. The forward pass computation yields the earliest start and the earliest finish time for each activity (i, j) and indirectly the earliest occurrence time for each event namely E_i. This consists of the following three steps:

Step 1: The computations begin from the start node and move towards the end node. Let zero be the starting time for the project.

Step 2: Earliest starting time $(ES)_{ij} = E_i$ is the earliest possible time when an activity can begin assuming that all of the predecessors are also started at their earliest starting time. Earliest finish time of activity (i, j) is the, Earliest starting time + Activity time

$$(EF)_{ij} = (ES)_{ij} + t_{ij}$$

Step 3: Earliest event time for event j is the maximum of the earliest finish time of all the activities ending at that event.

$$E_j = \underset{i}{Max}\left(E_j + t_{ij}\right)$$

The computed 'E' values are put over the respective rectangle ▭ representing each event.

Backward Pass Computations (for Latest Allowable Time)

The latest event time (L) indicates the time by which all activities entering into that event must be completed without delaying the completion of the project. These can be calculated by reversing the method of calculations used for the earliest event time. This is done in the following steps:

Step 1: For ending event assume $E = L$.

Step 2: Latest finish time for activity (i, j) is the target time for completing the project

$$(LF)_{ij} = L_j$$

Step 3: Latest starting time of the activity *(i,j)* = Latest completion time of (i, j), the activity time.

$$(LS)_{ij} = (LF)_{ij} - t_{ij}$$
$$= L_j - t_{ij}$$

Step 4: Latest event time for event i is the minimum of the latest start time of all activities originating from the event.

$$\boxed{L_i = \underset{j}{Min}\left(L_j - t_{ij}\right)}$$

The computed 'L' values are put over the respective triangle Δ representing each event.

Determination of Floats and Slack Times

Float is defined as the difference between the latest and the earliest activity time.

Slack is defined as the difference between the latest and the earliest event time.

Hence, the basic difference between the slack and the float is that slack is used for events only whereas float is used for activities.

There are mainly three kinds of floats as given below:

(i) Total Float: It refers to the amount of time by which the completion of an activity could be delayed beyond the earliest expected completion time without affecting the overall project duration time.

Mathematically, the total float of an activity (i, j) is the difference between the latest start time and the earliest start time of that activity.

Hence, the total float for an activity (i, j) denoted by $(TF)_{ij}$ is calculated by the formula,

$$(TF)_{ij} = \text{(Latest start – Earliest start) for activity } (i, j)$$

$$\boxed{\begin{aligned} &\text{i.e.,} (TF)_{ij} = (LS)_{ij} - (ES)_{ij} \\ &\text{or, } (TF)_{ij} = (L_j - E_i) - t_{ij} \end{aligned}}$$

Where E_i, L_j are the earliest time and latest time for the tail event i and head event j and t_{ij} is the normal time for the activity (i, j). This is the most important type of float as it concerns with the overall project duration.

(ii) Free Float: The time by which the completion of an activity can be delayed beyond the earliest finish time without affecting the earliest start of a subsequent succeeding activity.

Mathematically, the free float for activity (i, j) denoted by $(FF)_{ij}$ can be calculated by the formula,

$$\boxed{\begin{aligned} &FF_{ij} = (E_j - E_i) - t_{ij} \\ &FF_{ij} = \text{Total float – Head event slack} \end{aligned}}$$

Head Event Slack = $L_j - E_j$

This float is concerned with the commencement of subsequent activity.

The free float can take values from zero up to total float, but it cannot exceed total float. This float is very useful for rescheduling the activities with minimum disruption of earlier plans.

(iii) Independent Float: The amount of time by which the start of an activity can be delayed without affecting the earliest start time of any immediately following activities assuming that the preceding activity has finished at its latest finish time.

Mathematically, independent float of an activity (i, j) denoted by $(IF)_{ij}$ can be calculated by the formula,

NOTES

$$\boxed{\begin{array}{l}(IF)_{ij} = \left(E_j - L_i\right) - t_{ij} \\ \qquad \text{or} \\ (IF)_{ji} = \text{Free float} - \text{Tail event slack}\end{array}}$$

Where tail event slack is given by,

$$\text{Tail Event Slack } = L_i - E_i$$

The negative independent float is always taken as zero. This float is concerned with prior and subsequent activities.

$$(IF)_{ij} \leq (FF)_{ij} \leq (TF)_{ij}$$

Notes: 1. If the total float TF_{ij} for any activity (i, j) is zero, then those activities are called *critical activity*.

2. The float can be used to reduce project duration. While doing this, the float of not only that activity but that of other activities would also change.

Critical Activity: An activity is said to be critical if a delay in its start will cause a further delay in the completion of the entire project.

Critical Path: The sequence of critical activities in a network is called the *critical path.* It is the longest path in the network from the starting event to the ending event and defines the minimum time required to complete the project. In the network, it is denoted by double line. This path identifies all the critical activities of the project. Hence, for the activity (i, j) to lie on the critical path, following conditions must be satisfied.

(*i*) $ES_i = LF_i$

(*ii*) $ES_j = LF_j$

(*iii*) $ES_j - ES_i = LF_j - LF_i = t_{ij}$

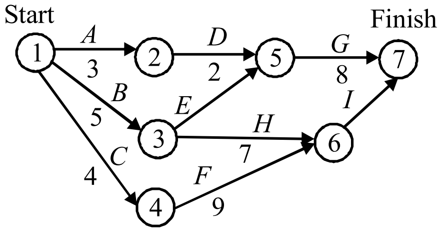
ES_i, ES_j, are the earliest start and finish time of the event *i* and *j*.

LF_i, LF_j are the latest start, finish time of the event *i* and *j*.

2.5 SHORTEST PATH PROBLEM AND MAXIMUM FLOW PROBLEM

The shortest path problem can be defined for graphs whether undirected, directed, or mixed. It is defined here for undirected graphs; for directed graphs the definition of path requires that consecutive vertices be connected by an appropriate directed edge.

Network Diagram



In the above diagram, each arrow represents an *activity* and each circle an event. Circle 1 represents the starting event and circle 7 represents the ending event. The names of the activities are generally stated just above the corresponding arrows. Thus *A* in the above diagram is the name of the activity represented by the arrow just drawn below it.

NOTES

- *Merge and Burst Events:* It may be pointed out that it is not necessary for an event to be the ending event of only one activity but an event can be the ending event of two or more activities in which case the said event is technically described as *merge event.* Similarly, if the event happens to be the beginning event of two or more activities it is technically called as the '*Burst Event*'.
- *Preceding, Succeeding and Concurrent Activities:* The activities can be classified as preceding activities; succeeding activities and the concurrent activities. Activities which most be accomplished before a given event can occur are termed as *preceding activities*; activities which cannot be accomplished until an event has occurred are termed as *succeeding activities* and activities which can be accomplished concurrently are known as *concurrent activities.* This classification is relative which means that one activity can be preceding to a certain event and the same activity can be succeeding to some other event or it may be a concurrent activity with one or more of the activities.
- *Dummy Activities:* Some times we use dummy activities in the preparation of network diagram. Such activities are to designate a precedence relationship and in the network diagram are shown as broken lines. They are characterized by their use of zero time and zero resource. Their main function is to help in assuring that the activities and events in a network diagram are in proper sequence.
- *Path and Critical Path:* A path is continuous chain of activities through a network which connects the first event to the last event. Critical path consists of the sequence of those events and connected activities that require the maximum time in the completion of the project. It is that path which takes the longest time. It is known as critical because it controls the completion date of the project. The length of this path determines the minimum time in which the project may be completed.
- *Critical Activities or Bottleneck Activities:* All the activities associated with the critical path are called as *critical* or *bottleneck activities*. Any delay in the completion of one or more of these activities will cause delay in the completion date of the project. Hence such activities require special attention of the project incharge.
- *Earlier Start Time or Est:* Est for an activity is the earliest possible time an activity can begin on the assumption that all activities preceding to it started at the earliest possible times.
- *Earliest Finish Time or Eft:* Eft is the sum of the earliest start time and the estimated time to perform the concerning activity.

- *Latest Finish Time or Lft:* Lft for an activity is the latest possible time an activity can finish without delaying the project beyond its dead line on the assumption that all the subsequent activities are performed as planned.
- *Latest Start Time (or Lst):* Lst for an activity is the difference between the latest finish time and the estimated time for the activity to be performed.
- *Float (Total, Interfering Independent and Free Floats):* Quite often the term float (in CPM terminology) is used in context of network analysis. Float may be understood as total float, interfering float, free float and independent float. *Total float is* the duration by which an activity can be delayed without delaying the project and can be worked out as either (*Lst-Est*) or (*Lft-Eft*). *Interfering float* is that part of the total float which causes a reduction in the float of the successor activity or activities. In other words, it is that portion of the activity float which cannot be consumed without affecting adversely the float of the succeeding activity or activities. It is worked out as a difference between the *Lft* of the activity and the *Est* of the following activity or zero whichever is larger. Interfering float is also known as the head event slack of an activity. *Free float* is that portion of the total float within which an activity can be manipulated without affecting the float of subsequent activities. It is worked out by subtracting the head event slack from the total float. The head event slack is its latest event time minus earliest event time or ($L_T – E_T$). *Independent float* is that portion of the total float within which an activity can be delayed for started without affecting float of the preceding activities. It is worked out by subtracting the tall event slack from the free float. If it obtains a negative value then it is taken as equal to zero. *Tail event* refers to the event where an activity say begins and *Head event* is the event where an activity comes to an end. If we have events (1) and (2) then (1) is the tail event and (2) is the head event of an activity *A*.

  Float may be positive or negative. Positive float indicates that the activities concerned have certain amount of spare time and can be delayed without effecting the project duration. On the other hand, negative float highlights the situation in which the activities concerned are short of time and unless their duration (to the extent of negative float) is reduced, completion of the project by the target time cannot be assured. Thus, negative float indicates the extent of criticality of the activities.
- *Slack:* The term slack is normally associated with events. It indicates the amount of latitude that is available for an event to occur. It is worked out as under:

Slack of an event = (Latest occurence time of the event) – (Earlier occurence time of the event) or simply slack of event = ($L_T – E_T$). Slack can be positive or negative depending upon whether the targetted date of completion is later or earlier than the earliest finish time of the task respectively.

When used for activities, the term slack should be used for activity slack (activity slack is synonymous to float). Since slack is associated with the events, each activity will have two slacks which includes the slack of its head event or the head slack and the slack of its tail event or the tail slack.

Preparation of the Network Arrow Diagram

We require the following information for each activity in the project for the preparation of the network diagram:

(*i*) The sequencing requirements for an activity must be known, i.e., the set of activities which must be completed prior to the beginning of each specific activity should be known.

(*ii*) An estimate of the time each activity will take should also be known.

Keeping all what has been stated above in view, the network diagram can easily be prepared. But the following rules of constructing network diagrams will have to be invariably adhered:

(*i*) Each activity is shown by an arrow only once in the network.

(*ii*) Network has to be developed on the basis of logical dependencies between various activities.

(*iii*) The length of arrows representing various activities have no significance; they only indicate the logical precedence.

(*iv*) Arrow direction shows the general progression in time.

(*v*) Events in the network are shown by numbers.

(*vi*) Activities are identified by the numbers of their starting and the ending events.

(*vii*) Parallel activities between two events without intervening events are not permitted. In such a situation dummy activities may have to be introduced.

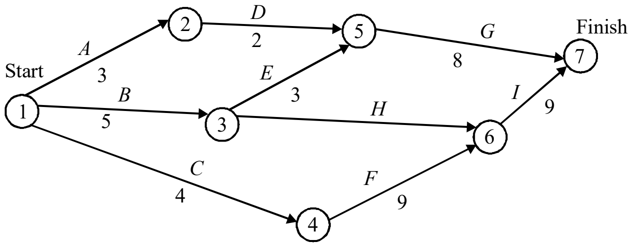
(*viii*) Looping is not permitted in a network. This means that if activity *A* precedes *B* and *B* precedes C, then C cannot precede *A*.

Now construct the network diagrams using the above stated rules.

Examples 2.13: Prepare a network arrow diagram for the following information.

Activity		*Name of the Activity*	*Pre-requisite Activity*	*Estimated Time (Weeks)*
Event	*Event*			
1	2	*A*	None	3
1	3	*B*	None	5
1	4	*C*	None	4
2	5	*D*	None	2
3	5	*E*	*A*	3
4	6	*F*	*B*	9
5	7	*G*	*C*	8
3	6	*H*	*D*	7
6	7	*I*	*E*	9

Solution: Draw the following network arrow diagram to solve the problem:



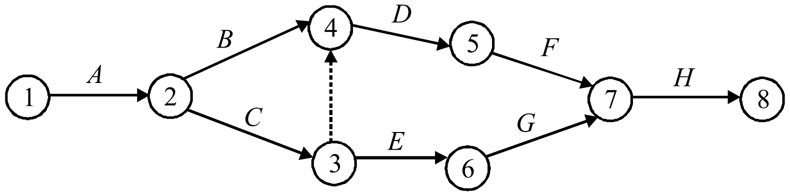
NOTES

The above is the required *network* digram for the given problem. Activity *A must* be completed before activity *D* can begin the arrow activities. The immediate preceding activity of activity *D* is activity *A* which means that activity *A* must be completed before activity *D* can begin. The arrow from circle 1 to circle 2 indicate that activity a must be completed before activity *D* can begin. Similarly activity *B* must be completed before activities *E* and/or *H*; *C* must be completed before *F* can begin; activities *H* and *F* must be completed before *I* can begin and activities *D* and *E* must be completed before activity *G* can begin. The estimated time for each activity has been placed just below the arrow representing that activity.

Example 2.14: Draw a network arrow diagram for the following information concerning some project:

Activity	*Predecessor Activity or Activities*
A	None
B	*A*
C	*A*
D	*B,C*
E	*C*
F	*D*
G	*E*
H	*F,G*

Solution: Draw the following network arrow diagram to solve the problem:



In this diagram, activity 3-4 is the dummy activity shown as a broken line. It is required because activities *B* and *C* both precede activity *D* but activity *C* alone precedes activity *E.*

2.5.1 Shortest-Route Problem: Dijkstra's Algorithm

Graphs are used to represent network of communication. A graph is depicted in diagrammatic form as a set of dots for the vertices, joined by lines or curves for the edges. If arrows are placed on one or both endpoints of the edges of a graph to indicate directedness, the graph is said to be directed. A graph structure can be extended by assigning a weight to each edge of the graph. Graphs with weights or weighted graphs are used to represent structures in which pairwise connections have some numerical values. For example if a graph represents a road network, the weights could represent the length of each road. A graph or directed graph together with a function which assigns a positive real number to each edge is known as a network.

Shortest route problem deals with ways of finding the minimum path distance form a selected node, say s which is source to a destination node d. Such problems arise in our real life and also in the field of computer science. For example, vertices of a weighted graph may represent cities and weights on edges represent costs of driving distances between pairs of cities connected by a direct road or rail link.

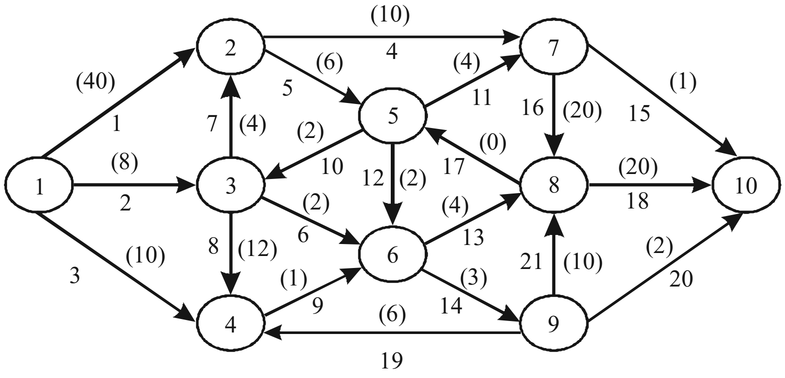
Edsger Dijkstra, a Dutch computer scientist, devised an algorithm to solve this problem and is known as **Dijkstra's algorithm**, on his name. It is an algorithm for graph search solving the single-source shortest path problem for a weighted graph having non-negative edge path costs, producing a tree that gives the shortest path. This algorithm is of great use in routing. The concept of 'shortest path first' finds extensive use in network routing protocols, like IS-IS and OSPF (Open Shortest Path First).

The Algorithm

We take a node as initial node or starting node. We select a destination X and find it's the distance from the **initial node.** This algorithm will assign some initial distance values and then will go step-by-step. These are as below:

1. Assigning every node, a distance value. For initial node, it is zero and for all other nodes, it is infinity.
2. Initially all unvisited nodes are marked. Initial node is set as current.
3. For current node, distance from the initial node to every unvisited neighbour is calculated. For example, if current node (A) has distance of 7, and an edge connecting it with another node (B) is 3, the distance to B through A will be 7 + 3 = 10. If it is less than the previously recorded distance which is infinity in the beginning as in step 1. The distance is overwritten.
4. When all neighbours of the current node are done with, it is marked as visited so that it is not visited node again and the distance recorded is final and minimal.

In the figure below problem of shortest path is presented. There are 10 nodes and starting node is node 1.



It is required to find the set of paths which is minimum from the source node to another node in the network. This problem of shortest path is a tree which is solved as below:

This has *m* number of nodes, starting with the source node alone, this procedure makes the number of iterations one less than the number of vertices, i.e., $m - 1$ for finding shortest path and construct a shortest path tree. In the above example, there are 10 number of nodes and will require 9 iterations.

Let *S* be the the set of nodes already visited. Nodes that are not solved are not in *S.* In each iteration, a number is assigned to each node. A node d_i denotes the length of minimum distance or shortest path from source node to i_{th} node.

After finishing the traversal d_i shows the shortest path to that node. algorithm assigns numbers *di* to each node in the network, where *di* is the length of the shortest path to node *i* from the source node. At the end of the algorithm πi is the length of the shortest path to node *i*. Let ***M*** be the set of all edges which is also called arcs.

In the beginning, at source node $S = \{s\}$ and $d_s = 0$.

Repeat until all nodes are in set *S.*

Find the edge, *p*(*i, j*), where *i* is the solved node and *j* is the unsolved node and arc moves from a node already solved to those not yet solved.

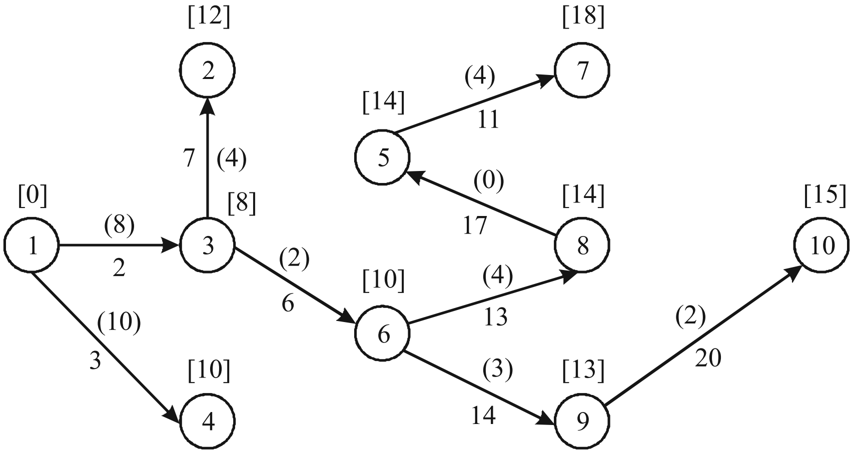
$p(i,j) = \text{arcmin}\{di' + cp' : p'(i', j') \in \boldsymbol{M}, i' \in \boldsymbol{S}, j' \in \boldsymbol{S}^c\}$

Add node *j* and arc *p* to the tree. Add node *j* to the solved set ***S***.

Let dj = di + cp.

In each iteration this algorithm computes the length of path, (which is path-length) from solved node to unsolved nodes. Node having the shortest length is included in the set of solved nodes. After the spanning tree is created, the process terminates.

The spanning tree obtained is shown below:



The numbers in bold, put under brackets indicate the value of length which is associated with nodes . For example shortest path of node 6 is 10. The numbers in the bracket for the nodes in S_c indicates the shortest path length to unsolved nodes passing through the solved nodes in the set *S.* After this the arc is selected with the smallest *dj* value for $\iota \in \boldsymbol{S}^c$. Hence, choice is made from minimum of 18, 22, 14, 13 and is expressed as min{18, 22, 14, 13}. So, node 9 and arc 14 are included in the spanning tree.

Tabular Presentation of this Algorithm

The algorithm creates a table of seven columns as shown in Table 2.1. Column 1 shows the value of *h* which shows the number of nodes in the set ***S***. Second column lists the members of the set ***S*** which contains solved nodes having minimum one arc connected to a node which is not yet traversed and called unsolved node. Third column shows the closest unsolved node for each node listed in column 2. Column 4 contain every *i,* which is the index of nodes listed in second column,

NOTES

third column lists *j* as the index of the node which is listed in third column. We set *d* as the index of the path or arc oining nodes *i* and *j* and carrying out computation for each case *dj' = di + pk*.

Fifth Column selects the least number from fourth column. Second column has nodes denoted by *i* and *j* is that for node in third column from where, this number was calculated. Fifth column contains node *j* and sixth column lists the length of the shortest path to the node which is added. This is the minimum and is obtained from fourth column. Seventh column 7 has the arc $p(i, j)$ Shortest path tree is formed by adding the node *j* and arc *p* and *j* is added to the set *S*.

Table 2.1 *Tabular Presentation of Dijkstra's Algorithm*

h	*Solved Nodes*	*Unsolved Node, Closest to Solved Node*	*Path Length to Unsolved Node*	*Node Added to the Set of Solved Node*	*Shortest Path*	*Arc Added to Tree*
1	1	3	8	3	8	2
2	1	4	10	4	10	3
	3	6	10			
3	1	2	40	6	10	6
	3	6	10			
	4	6	11			
4	1	2	40	2	12	7
	3	2	12			
	6	9	13			
5	2	5	18	9	13	14
	6	9	13			
6	2	5	18	8	14	13
	6	8	14			
	9	10	15			
7	2	5	18	5	14	17
	8	5	14			
	9	10	15			
8	2	7	22	10	15	20
	5	7	18			
	8	10	34			
	9	10	15			
9	2	7	22	7	18	11
	5	7	18			

2.5.2 Maximum Flow Problem and Spanning Tree

Flow is a network. The network may be a transportation network through which commodities flow or a pipeline network through which oil flows or any number of other possibilities. In each case, the problem is to determine a maximal flow. Maximizing the flow in a network is a problem that belongs to graph theory and operations research.

Network Models

A transport network is a simple, weighted, directed graph *G* satisfying the following:

1. There are two distinguished vertices *s* and *t* of *G*, called the source and sink of *G* respectively, i.e., in *G*, only one vertex with no incoming edge is called as source, *s* and only one vertex with no outgoing edge is called as sink, *t*.

NOTES

2. The weight C_{ij} of the directed edge (i,j) is a non-negative number, i.e, every edge is associated with non-negative real numbers called as capacity of that edge.
3. The underlying (undirected) graph is connected.

Notes:

1. If G is a network, we will denote the source by a and the sink by z.
2. Let G be a network, a and z be the source and sink of G, respectively. The other vertices are called as intermediate vertices.
3. In general, source has in-degree '0' and sink has out-degree '0'.
4. A flow in a network assigns a flow in each directed edge which does not exceed the capacity of the edge.
5. It is assumed that the flow into an intermediate vertex u is equal to the flow out of u.

Flow

Let G be a transport network. Let C_{ij} denote the capacity of the directed edge (i,j). A flow F in G assigns each directed edge (i,j) a non-negative number F_{ij} such that

i. $F_{ij} \leq C_{ij}$

ii. For each intermediate vertex j,

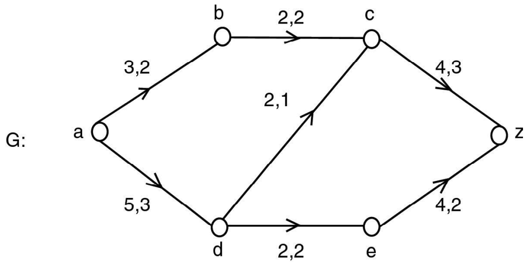
$$\sum_i F_{ij} = \sum_i F_{ji}$$

(Here $\sum_i F_{ij}$ is the flow into j and $\sum_i F_{ji}$ is the flow out of j)

Value of the Flow: Let F be the flow in a network G. The value $\sum_i F_{ai} = \sum_i F_{iz}$ is called the value of the flow F.

Note: Whenever the flow along an edge is equal to the capacity of that edge, then that edge is said to be saturated. Otherwise it is unsaturated.

Example 2.15: Find the flow of the following network:

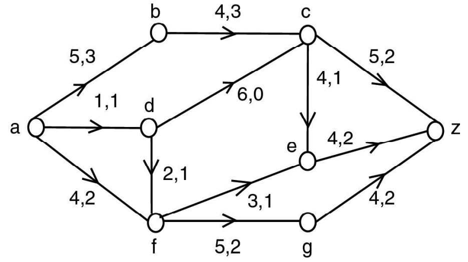


Solution: Here, the first quantity represents the capacity and the second quantity represents the flow.

The flow of the above network is 5.

Example 2.16: Find the flow for the following networks.

(*i*)



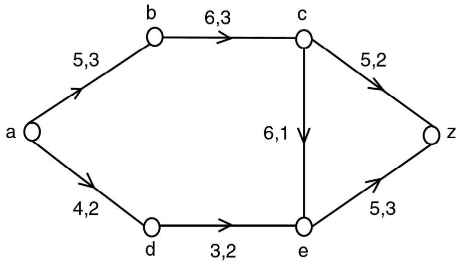
Solution: The flow out of source *a* is $F_{ab} + F_{ad} + F_{af} = 3 + 1 + 2 = 6$.

The flow into the sink *z* is $F_{gz} + F_{ez} + F_{cz} = 2 + 2 + 2 = 6$.

All other intermediate vertices are given by, $\sum_i F_{ij} = \sum_i F_{ji}$

$\therefore$ Flow of the network is 6.

(*ii*)



On similar grounds the flow is found to be 5.

Maximal Flow: A flow *F* in a network *G* is called a maximal flow if $|F| \geq |F'|$ for every flow F' in *G*.

Improve the Value of Flow F in a Network: Start with some initial flow and iteratively increase the value of flow until no improvement is possible in the given network.

We can take the initial flow to be the flow, when the flow in each edge is zero. To increase the value of a given flow, we have to determine a path from the source *a* to the sink *z* and increase the flow along this path. Let *P* be a path from *a* to *z* in the underlying graph of the network *G*. If an edge *e* in *P* is directed from v_i to v_{i+1}, we say e is properly oriented; otherwise *e* is improperly oriented.

If we can find a path *P* from *a* to *z* in which every edge in *P* is properly oriented and the flow in each edge is less than the capacity of the edge, then it is possible to increase the value of the flow.

Note: It is also possible to increase the flow in certain paths from *a* to *z* in which properly and improperly oriented edges are present.

Proof: Let *P* be a path from *a* to *z* and let *V* be a vertex in *P* different from *a* and *z*. Since *V* is an intermediate vertex in the network *N* (also in *P*), two edges e_1 and e_2 are incident on this vertex *V*. There are now four cases for the orientations of the edges e_1 and $e_{2.}$ They are as follows:

(*i*) When flow is increased in each edge by Δ the flow into *V* will be equal to the flow out of *V.*

(*ii*) When we increase the flow in e_2 by Δ, we must decrease the flow in e_1 by Δ so that the flow into *V* will still equal the flow out of *V*.

(*iii*) When we increase the flow in e_1 by Δ, we must decrease the flow in e_2 by Δ to achieve equal flow in and out of V.

(*iv*) When flow is decreased in both edges by Δ we achieve equal flow in and out of V.

NOTES

All the above cases are possible since $F_{ij} < C_{ij}$ for a properly oriented edge (i,j) and $0 < F_{ij}$ for an improperly oriented edge (i, j)

Result: Let P be a path from a to z in a network G.

(*i*) For a properly oriented edge (i,j) in P, $F_{ij} < C_{ij}$

(*ii*) For an improperly oriented edge (i,j) in P, $0 < F_{ij}$

Let Δ = mini x

Where x consists of the numbers $C_{ij} - F_{ij}$ for properly oriented edges (i, j) in P and F_{ij} for improperly oriented edge (i, j) in P.

$$\text{Define } F^*_{ij} = \begin{cases} F_{ij} & \text{if } \quad (i, j) \text{ is not in } P \\ F_{ij} + \Delta & \text{if } \quad (i, j) \text{ is properly oriented in } P \\ F_{ij} - \Delta & \text{if } \quad (i, j) \text{ is improperly oriented in } P \end{cases}$$

Then F^* is a flow whose value is greater by Δ than the value of F.

In every case, the resulting edge assignments give a flow F^*. Clearly the value of $F^* = F + \Delta$.

Labelling Procedure

Now we present an algorithm called as labelling procedure to find the maximal flow in a network.

We have a network G in which a is the source, z is the sink and C is the capacity. The capacity of each edge is a non-negative integer. The vertices of G are ordered $= v_0, v_1, ..., v_n = z$.

(1) Set $F_{ij} = 0$ for each edge (i,j). [initialization]

(2) Label vertex a by $(–, \infty)$. [labelling the source]

(3) If the sink z is labelled, go to Step 6.

(4) Choose the not yet examined, labelled vertex v_i with smallest index i. Stop if there is no such vertex;otherwise set $v = v_i$.[Go to adjacent vertices of the source]

(5) Let (α, Δ) be the label of v. Examine each edge of the form (v,w), (w,v) where w is unlabelled.

If $F_{vw} < C_{vw}$, label vertex w by $(v, \text{mini } (\Delta, C_{vw} - F_{vw}))$

if $F_{vw} = C_{vw}$, do not label w.

For an edge of the form (w,v)

if $F_{wv} > 0$, label vertex w by $(v, \text{mini } (\Delta, F_{wv}))$

if $F_{wv} = 0$, do not label w. Go to Step (3)

(6) Let (γ, Δ) be the label of z.

Let $w_0 = z$, $w_1 = \gamma$

If the label of $w_i = (\gamma^1, \ \Delta^1)$ put $w_i + 1 = \gamma^1$. Continue $w_k = a$. Now the path P: $a = w_k, w_{k-1}, ..., w_1, w_0 = z$. is the required path from a to z.

To change the flow of the edges in P find, if the edge e is properly oriented, then increase the flow in e by Δ. Otherwise decrease the flow in e by Δ. Remove all labels from vertices and go to Step 2.

NOTES

2.6 MINIMUM COST FLOW PROBLEM

Step 1 Determine the smallest cost in the cost matrix of the transportation table. Let it be C_{ij}. Allocate $x_{ij} = \min (a_i, b_j)$ in the cell (i, j)

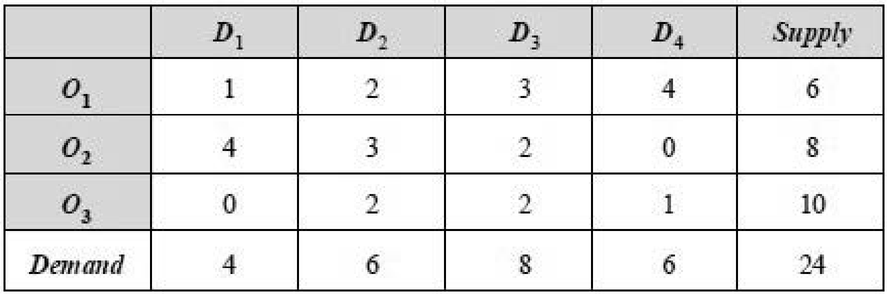
Step 2 If $x_{ij} = a_i$, cross off the i^{th} row of the transportation table and decrease b_j by a_i. Then go to step 3.

If $x_{ij} = b_j$, cross off the j^{th} column of the transportation table and decrease a_i by b_j. Go to step 3.

If $x_{ij} = a_i = b_j$, cross off either the i^{th} row or the j^{th} column but not both.

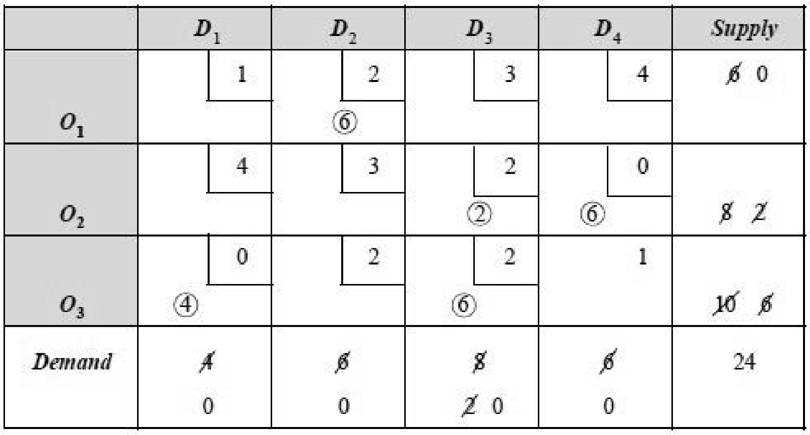
Step 3 Repeat steps 1 and 2 for the resulting reduced transportation table until all the rim requirements are satisfied. Whenever the minimum cost is not unique, make an arbitrary choice among the minima.

Example 2.17: Obtain an initial feasible solution to the following TP using the matrix minima method.



	D_1	D_2	D_3	D_4	*Supply*
O_1	1	2	3	4	6
O_2	4	3	2	0	8
O_3	0	2	2	1	10
Demand	4	6	8	6	24

Solution: Since $Sa_i = Sb_j = 24$, there exists a feasible solution to the TP. Using the steps in the least cost method, the first allocation is made in the cell (3, 1) the magnitude being $x_{31} = 4$. It satisfies the demand at the destination D_1 and we delete this column from the table as it is exhausted.



	D_1	D_2	D_3	D_4	*Supply*
O_1	1	2 ⑥	3	4	~~6~~ 0
O_2	4	3	2 ②	0 ⑥	~~8~~ ~~2~~
O_3	0 ④	2	2 ⑥	1	~~10~~ ~~6~~
Demand	~~4~~ 0	~~6~~ 0	~~8~~ ~~2~~ 0	~~6~~ 0	24

The second allocation is made in the cell (2, 4) with magnitude x_{24} = min (6, 8) = 6. Since it satisfies the demand at the destination D_4, it is deleted from the table. From the reduced table the third allocation is made in the cell (3, 3) with magnitude x_{33} = min (8, 6) = 6. The next allocation is made in the cell (2, 3) with magnitude x_{23} of min (2, 2) = 2. Finally the allocation is made in the cell (1, 2) with magnitude x_{12} = min (6, 6) = 6. Now all the rim requirements have been satisfied and hence, initial feasible solution is obtained.

The solution is given by,

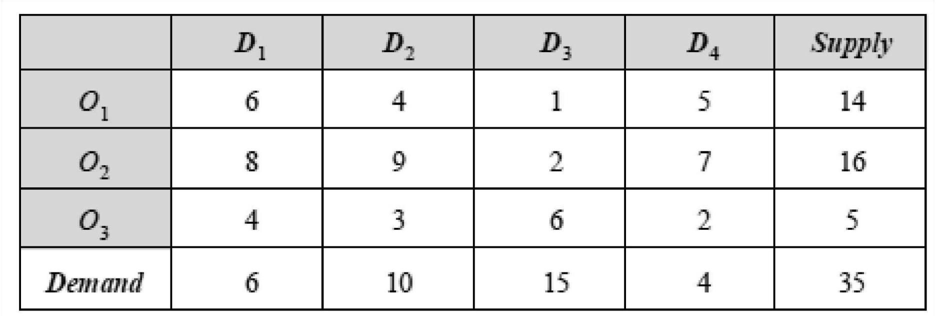
$$x_{12} = 6,\ x_{23} = 2,\ x_{24} = 6,\ x_{31} = 4,\ x_{33} = 6$$

Since the total number of occupied cells = 5 < $m + n + 1$.

We get a degenerate solution.

$$\text{Total cost} = (6 \times 2) + (2 \times 2) + (6 \times 0) + (4 \times 0) + (6 \times 2)$$
$$= 12 + 4 + 12 = ₹\ 28.$$

Example 2.18: Determine an initial basic feasible solution for the following TP, using least cost method.



	D_1	D_2	D_3	D_4	*Supply*
O_1	6	4	1	5	14
O_2	8	9	2	7	16
O_3	4	3	6	2	5
Demand	6	10	15	4	35

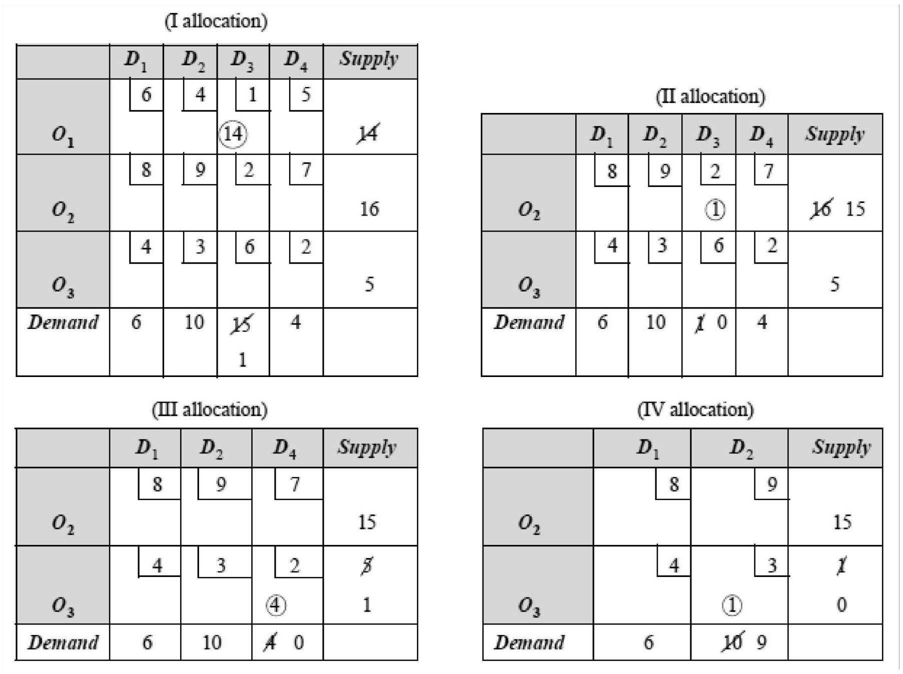
Solution: Since $\Sigma a_i = \Sigma b_j$, there exists a basic feasible solution. Using the steps in least cost method, we make the first allocation to the cell (1, 3) with magnitude x_{13} = min (14, 15) = 14 (as it is the cell having the least cost).

This allocation exhausts the first row supply. Hence, the first row is deleted. From the reduced table, the next allocation is made in the next least cost cell (2, 3) which is chosen arbitrarily with magnitude x_{23} = min (1, 16) = 1, which exhausts the 3rd column destination.

From the reduced table, the next least cost cell is (3, 4) to which allocation is made with magnitude min (4, 5) = 4. This exhausts the destination D_4 requirement, deleting the fourth column from the table. The next allocation is made in the cell (3, 2) with magnitude x_{32} = Min (1, 10) = 1, which exhausts the 3rd origin capacity. Hence, the 3rd row is exhausted. From the reduced table the next allocation is given to the cell (2,1) with magnitude x_{21} = min (6, 15) = 6. This exhausts the first column requirement. Hence, it is deleted from the table.

Finally the allocation is made to the cell (2, 2) with magnitude x_{22} = min (9, 9) = 9, which satisfies the rim requirement. These allocations are shown in the transportation table as follows:

NOTES



(I allocation)

	D_1	D_2	D_3	D_4	Supply
O_1	6	4	1 ⑭	5	~~14~~
O_2	8	9	2	7	16
O_3	4	3	6	2	5
Demand	6	10	~~15~~ 1	4	

(II allocation)

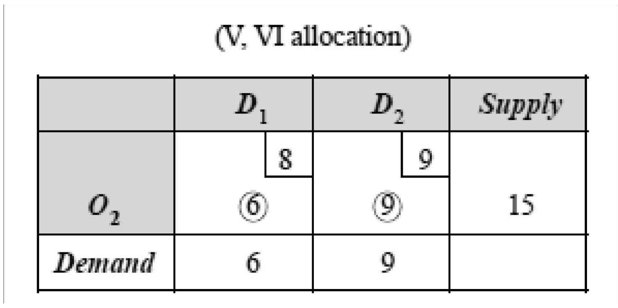
	D_1	D_2	D_3	D_4	Supply
O_2	8	9	2 ①	7	~~16~~ 15
O_3	4	3	6	2	5
Demand	6	10	~~1~~ 0	4	

(III allocation)

	D_1	D_2	D_4	Supply
O_2	8	9	7	15
O_3	4	3	2 ④	~~5~~ 1
Demand	6	10	~~4~~ 0	

(IV allocation)

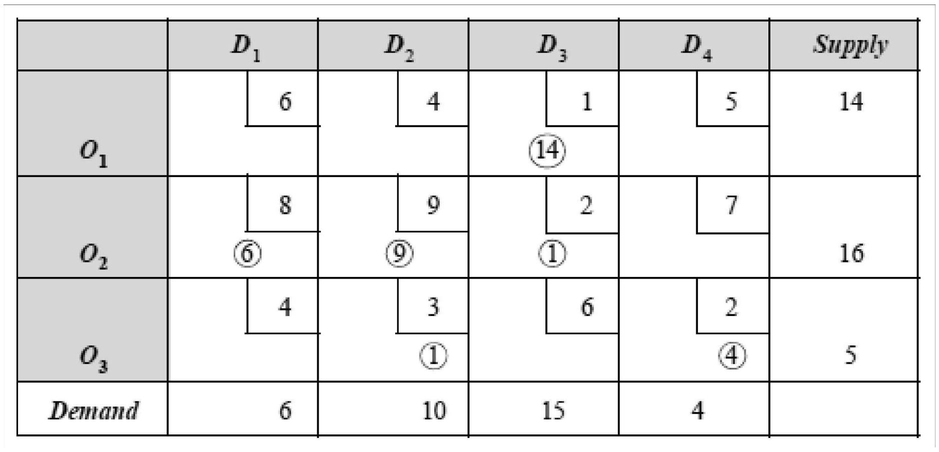
	D_1	D_2	Supply
O_2	8	9	15
O_3	4	3 ①	~~1~~ 0
Demand	6	~~10~~ 9	



(V, VI allocation)

	D_1	D_2	Supply
O_2	8 ⑥	9 ⑨	15
Demand	6	9	

The following table gives the initial basic feasible solution.



	D_1	D_2	D_3	D_4	Supply
O_1	6	4	1 ⑭	5	14
O_2	8 ⑥	9 ⑨	2 ①	7	16
O_3	4	3 ①	6	2 ④	5
Demand	6	10	15	4	

Solution is given by,

$x_{13} = 14$; $x_{21} = 6$; $x_{22} = 9$; $x_{23} = 1$; $x_{32} = 1$; $x_{34} = 4$

Transportation cost

$=(14 \times 1) + (6 \times 8) + (9 \times 9) + (1 \times 2) + (1 \times 3) + (4 \times 2)$

$=$ ₹ 156.

Network Simplex Method

The network simplex algorithm is used to solve minimum cost network flow problems. When some of the edges have capacities then a variable can be non-basic at either its lower bound or its upper bound.

We have a directed graph $G = (V,E)$. We assume G is connected. Each arc $(i,j) \in E$ has a cost per unit flow of c_{ij} and a capacity of u_{ij} (possibly infinite). Each node *i* ” *V* has a net supply b_i, so

$$b_i \begin{cases} > 0 & \text{if } i \text{ is a supply node} \\ = 0 & \text{if } i \text{ is a transshipment node} \\ < 0 & \text{if } i \text{ is a demand node.} \end{cases}$$

The problem is formulated as

$$\begin{array}{lrcll} \min_x & \sum_{(i,j)\in E} c_{ij}x_{ij} & & & \\ \text{s.t.} & \sum_{(i,j)\in E} x_{ij} - \sum_{(k,i)\in E} x_{ki} & = & b_i & \forall i \in V \\ & 0 \le x_{ij} & \le & u_{ij} & \forall (i,j) \in E. \end{array} \qquad (P)$$

The dual problem is

$$\begin{array}{llcl} \max_{y,z,w} & \sum_{i\in V} b_i y_i - \sum_{(i,j)\in E} u_{ij}w_{ij} & & \\ \text{s.t.} & y_i - y_j + z_{ij} - w_{ij} & = c_{ij} & \forall (i,j) \in E \\ & z_{ij}, w_{ij} & \ge 0 & \forall (i,j) \in E \end{array} \qquad (D)$$

If some x_{ij} is unbounded then the corresponding w_{ij} is set equal to zero. In a basic feasible solution, the set of basic variables constitutes a spanning tree in G. Let x be a basic feasible solution to (P). We use **complementary slackness** to find a dual solution (y,z,w). We must have:

$$\bar{y}_i - \bar{y}_j = c_{ij} \quad \text{if } \bar{x}_{ij} \text{ is basic.}$$

This underdetermined system can be solved easily: fix one component of y, and then the remaining components are determined by a chain reaction. Note that $z_{ij} = w_{ij} = 0$ for the basic x_{ij}.

We need to **check dual feasibility** for the nonbasic edges. Break into cases depending on which bound is active:

- If nonbasic $x_{ij} = 0$ then $w_{ij} = 0$ from complementary slackness, so $z_{ij} = c_{ij} - y_i + y_j$.
- If nonbasic $x_{ij} = u_{ij}$ then $z_{ij} = 0$ from complementary slackness, so $w_{ij} = -c_{ij} + y_i - y_j$.

If $w \ge 0$ and $z \ge 0$ then **x is optimal**.

Else, **perform a simplex iteration**: choose a nonbasic arc with $z_{ij} < 0$ or $w_{ij} < 0$ to enter the basis, construct a cycle using the incoming arc and the basic edges, adjust flow around the edges of the cycle, use the minimum ratio test to determine a basic variable to leave the basis, and update the basis and x.

2.7 PROJECT PLANNING AND CONTROL WITH PERT-CPM

NOTES

The network methods discussed so far may be termed as deterministic, since estimated activity times are assumed to be known with certainty. However, in research project or design of gear box of a new machine, various activities are based on judgement. It is difficult to obtain a reliable time estimate due to the changing technology. Time values are subject to chance variations. For such cases where the activities are non-deterministic in nature, PERT was developed. Hence, PERT is a probabilistic method where the activity times are represented by a probability distribution. This probability distribution of activity times is based upon three different time estimates made for each activity. These are as follows:

(*i*) Optimistic Time Estimate

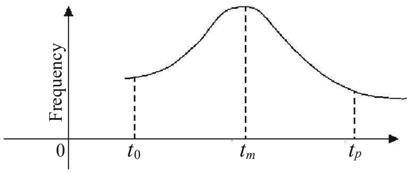
(*ii*) Most Likely Time Estimate

(*iii*) Pessimistic Time Estimate

Optimistic Time Estimate: It is the smallest time taken to complete the activity if everything goes on well. There is very little chance that activity can be done in time less than the optimistic time. It is denoted by t_0 or *a*.

Most Likely Time Estimate: It refers to the estimate of the normal time the activity would take. This assumes normal delays. It is the mode of the probability distribution. It is denoted by t_m or (*m*).

Pessimistic Time Estimate: It is the longest time that an activity would take if everything goes wrong. It is denoted by t_p or *b*. These three time values are shown in the following figure.



Time Distribution Curve

From these three time estimates, we have to calculate the expected time of an activity. It is given by the weighted average of the three time estimates,

$$\boxed{t_e = \frac{t_0 + 4t_m + t_p}{6}}$$

β distribution with weights of 1, 4, 1, for t_o, t_m and t_p estimates respectively.

Variance of the activity is given by,

$$\sigma^2 = \left[\frac{t_p - t_0}{6}\right]^2$$

The expected length (duration), denoted by T_c of the entire project is the length of the critical path, i.e., the sum of the t_c's of all the activities along the critical path.

The main objective in the analysis through PERT is to find the completion for a particular event within specified date T_S, given by $P(Z \leq D)$ where,

NOTES

$$D = \frac{\text{Due date} - \text{Expected date of completion}}{\sqrt{\text{Project variance}}}$$

Where Z stands for standard normal variable.

2.7.1 Project Scheduling

In order to include the cost factors in project scheduling, we must first define the cost duration relationships for various activities in the project. The total cost of any project comprises of direct and indirect costs.

Direct Cost: This cost is directly dependent upon the amount of resources available in the execution of individual activities, e.g., manpower loading, materials consumed, etc. *The direct cost increases if the activity duration is to be reduced.*

Indirect Cost: This cost is associated with overhead expenses such as managerial services, indirect supplies, general administration, etc. The indirect cost is computed on a per day, per week or per month basis. *The indirect cost decreases if the activity duration is to be reduced.*

Network diagram can be used to identify the activities whose duration should be shortened, so that the completion time of the project can be shortened in the most economic manner. The process of reducing the activity duration by putting on extra effort is called *crashing the activity.*

The crash time (T_C) represents the minimum activity duration time that is possible and any attempts to further crash would only raise the activity cost without reducing the time. The activity cost corresponding to the crash time is called the *crash cost* (C_C), which is the minimum direct cost required to achieve the crash performance time.

The *normal cost* (C_N) is equal to the absolute minimum of the direct cost required to perform an activity. The corresponding time duration taken by an activity is known as the *normal time* (T_N).

Cost Slope

The cost slope, indicating the increase in cost per unit reduction in time is defined as,

$$\text{Cost slope} = \frac{\text{Crash cost} - \text{Normal cost}}{\text{Normal time} - \text{Crash time}} = \frac{C_C - C_N}{T_N - T_C}$$

i.e., it represents the rate of increase in the cost of performing the activity per unit reduction in time and is called *cost/time trade off.* It varies from activity to activity. The total project cost is the sum total of the project's direct and indirect costs.

Time-Cost Optimization Algorithm

Following are the steps involved in project crashing.

Step 1 Find the normal critical path and identify the critical activities.

$$\text{Cost slope} = \frac{\text{Crash cost} - \text{Normal cost}}{\text{Normal time} - \text{Crash time}}$$

Step 2 Calculate the cost slope for the different activities by using the formula.

Step 3 Rank the activities. The activity whose cost slope in minimum is to be ranked 1, the next minimum as rank 2 and so on, i.e., the ranking takes place in ascending order of cost slope.

Step 4 By crashing the activities on the critical path, other paths also become critical and are called *parallel paths.*

In such cases, the project duration can be reduced by crashing activities simultaneously on the parallel critical path.

Step 5 Find the total cost of the project at each step.

Step 6 Continue the process until all the critical activities are fully crashed or no further crashing is possible.

In the case of indirect cost, the process of crashing is repeated until the total cost is minimum, beyond which it may increase.

This minimum cost is called the *optimum project cost* and the corresponding time, the *optimum project time.*

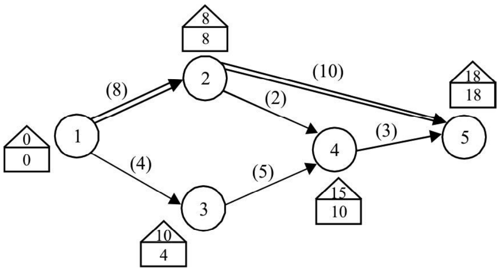
Example 2.19: Determine the optimum project duration and cost for the following data.

Activity	*Normal*		*Crash*	
	Time (days)	*Cost (₹)*	*Time (days)*	*Cost (₹)*
1–2	8	100	6	200
1–3	4	150	2	350
2–4	2	50	1	90
2–5	10	100	5	400
3–4	5	100	1	200
4–5	3	80	1	100

Indirect cost is ₹ 70 per day.

Solution: Since the overhead of the project is given, the cost is indirect. Hence, the project duration can be reduced with the total cost.

Making use of the normal time, we have the following network.



Critical path 1–2–5

Project normal duration = 18 days

Cost of the project = Normal cost of all the activities + Indirect cost

$= 580 + (70 \times 18) =$ ₹ 1,840.

We can reduce the project duration from 18 days, by crashing the activity on the critical path.

The cost slope and the number of days to be crashed are given in Table (A).

Table (A)

Activity	*Cost slope =* $\frac{C_C - C_N}{T_N - T_C}$
	$\frac{200-100}{8-6} =$
1–2	
1–3	50(2) (IV)
2–4	40(1) (III)
2–5	60(5) (V)
3–4	25(4) (II)
4–5	10(2) (I)

From the network, the other paths are given by,

1 – 2 – 5| ~~18~~ ~~17~~ ~~16~~ ~~15~~ ~~14~~ ~~13~~ ~~12~~ ~~11~~

1 – 2 – 4 – 5| ~~13~~ ~~12~~ 11

1 – 3 – 4 – 5| ~~12~~ 11

We form a table (B) to calculate the optimum project duration and its cost.

Table (B)

Normal duration	*Crash activity*	*Crash cost*	*Indirect cost*	*Total cost*
18	–	–	18 × 70 = 12601260 + 580 =	1840
17	1–2 (1)	1 × 50 = 50	17 × 70 = 1190	1820
16	1–2 (2)	50 + (1 × 50) = 100	16 × 70 = 1120	1800
15	2–5 (1)	100 + (60 × 1) = 160	15 × 70 = 1050	1790
14	2–5 (2)	160 + (1 × 60) = 220	14 × 70 = 980	1780
13	2–5 (3)	220 + (1 × 60) = 280	13 × 70 = 910	1770
12	2–5 (4)	280 + (1 × 60) = 340	12 × 70 = 840	1760
11	2–5 (1–2–5) 4–5 (1–3–4–5)	340 + (1 × 60) + (1 × 10) = 410	11 × 70 = 770	1760

We rank the activities in ascending order of cost slope as given in the above table. First we crash the activity 1–2. It is the activity lying on the critical path and with minimum cost slope. This crashing is shown in the other paths also. As the activity 1–2 can be crashed for two days, next we have the activity 2–5 on the critical path. After crashing to 12 days, we get the parallel path namely 1–2–5 and 1–3–4–5.

As there is no common activity between these 2 paths, we crash the activity 2–5 on the path 1–2–5 and 4–5 on the path 1–3–4–5 as it is the activity having the minimum rank. No more crashing is possible as all the activities in the path 1–2–5 are in crash time, even though there are activities available in other parallel paths.

Hence, the optimum project duration is 11 days with total cost of ₹ 1,760.

Example 2.20: The following table gives the activities of a construction project along with other relevant information.

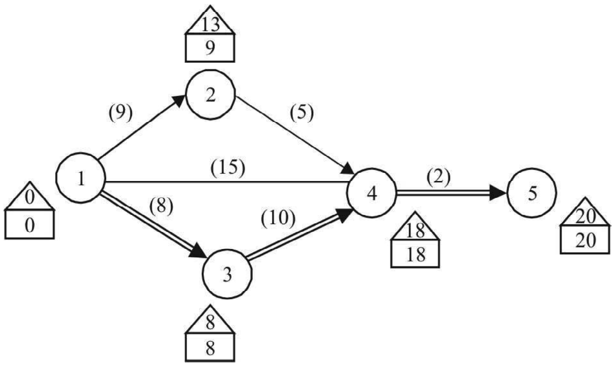
(*i*) What is the normal project length and the minimum project length?

(*ii*) Determine the minimum crashing costs of schedule, ranging from normal length down to and including the minimum length schedule.

Activity *i–j*	*Normal duration* *(days)*	*Crash duration* *(days)*	*Cost of crashing* *(₹per day)*
1–2	9	6	20
1–3	8	5	25
1–4	15	10	30
2–4	5	3	10
3–4	10	6	15
4–5	2	1	40

(*iii*) What is the optimal length schedule duration of each job for your solution? Overhead of the project is ₹ 60 per day.

Solution: Since the overhead cost is given, the cost is indirect. The project duration can be reduced by reducing the total cost associated with it.



The critical path comprises the activities 1–3, 3–4 and 4–5 with the normal duration as 20 days.

The total cost associated with the project is 20 × 60 = ₹ 1,200.

Activity	*Cost of crashing*
1–2	20 (3) III
1–3	25 (3) IV
1–4	30 (5) V
2–4	10 (2) I
3–4	15 (4) II
4–5	40 (1) VI

The cost slope is given in the data. We reduce the project duration by crashing the activities lying on the critical path.

The critical activity 3–4 has the minimum cost slope. It is ranked I, so this activity is crashed first for 4 days.

The other paths of the network are:

1 – 3 – 4 – 5| ~~20~~ ~~19~~ ~~18~~ ~~17~~ ~~16~~ ~~15~~ ~~14~~ 13

1 – 4 – 5| ~~17~~ ~~16~~ ~~15~~ ~~14~~ 13

1 – 2 – 4 – 5| ~~16~~ ~~15~~ ~~14~~ 13

Normal duration	*Crash activity*	*Crash cost*	*Overhead cost*	*Total cost*
20	–	–	20 × 60 = 1200	1200
19	3–4 (1)	1 × 15 = 15	19 × 60 = 1140	1155
18	3–4 (2)	15 + (1 × 15) = 30	18 × 60 = 1080	1110
17	3–4 (3)	30 + (1 × 15) = 45	17 × 60 = 1020	1065
16	4–5 (1–3–4–5–1–4–5)	45 + (1 × 40) = 85	16 × 60 = 960	1045
15	3–4 (1–3–4–5) 1–4 (1–4–5) 2–4 (1–2–4–5)	85 + 15 + 30 + 10 = 140	15 × 60 = 900	1040
14	1–3 (1–3–4–5) 1–4 (1–4–5) 2–4 (1–2–4–5)	140 + 25 + 30+ 10 = 205	14 × 60 = 840	1045
13	1–3 (1–3–4–5) 1–4 (1–4–5) 1–2 (1–2–4–5)	205 + 25 + 30 + 20 = 280	13 × 60 = 780	1060

After 17 days, we get parallel paths namely, 1–3–4–5 and 1–4–5. We crash the activity 4–5 for one day at the rate of ₹ 40 per day.

After 16 days, we get all the critical paths. As there is no common activity between them, we crash the activity 3–4 for the path 1–3–4–5, 1–4 for the path 1–4–5 and 2–4 for the path 1–2–4–5.

As the activity 3–4 in the path 1–3–4–5 is in crash time, we crash the activity 1–3 for 1 day at the rate of ₹ 25 per day. The optimum duration is 15 days as it gives the minimum cost of ₹ 1,040.

To find the minimum duration we crash further.

Normal duration	*Crash activity*	*Crash cost*	*Overhead cost*	*Total cost*
12	1–3 (1–3–4–5) 1–4 (1–4–5) 1–2 (1–2–4–5)	280 + 25 + 30 +20 = 355	12 × 60 = 720	1075

The minimum duration is 12 days and no more crashing is possible as all the activities in the path 1–3–4–5 are in crash time.

Optimum duration = 15 days with total cost associated as ₹ 1,040

Minimum duration = 12 days with total cost associated as ₹ 1,075.

***Note*:** From the above problem, we observe that the optimum duration and minimum duration are not the same.

Optimum duration refers to the duration that yields the minimum total cost, whereas minimum duration is the one in which no more crashing is possible beyond that duration.

Example 2.21: The table below provides costs and estimates for a seven-activity project.

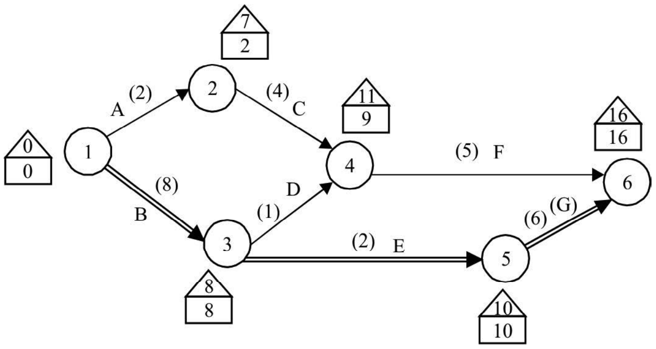
Activity	***Time estimate*** **(*weeks*)**		***Direct cost estimate*** **(*₹1,000*)**	
	Normal	***Crash***	***Normal***	***Crash***
A 1–2	2	1	10	15
B 1–3	8	5	15	21
C 2–4	4	3	20	24
D 3–4	1	1	7	7
E 3–5	2	1	8	15
F 4–6	5	3	10	16
G 5–6	6	2	12	36

(*i*) Draw the project network corresponding to normal time.

(*ii*) Determine the critical path, normal duration and cost of the project.

(*iii*) Crash the activities so that the project completion time reduces to 9 weeks.

Solution: As the problem involves *direct cost*, we expect that the project duration can be reduced with an increase in total cost. First we draw the network.



Critical path 1–3–5–6; Normal duration = 16 weeks

Total cost = ₹ 82,000

The calculations for cost slope and crashing number of days are shown in the table below:

Activity	*Slope (₹1000)*
1–2	5 (1) IV
1–3	2 (3) I
2–4	4 (1) III
3–4	0
3–5	7 (1) VI
4–6	3 (2) II
5–6	6 (4) V

The different paths of the network are,

1–3–5–6| ~~16~~ ~~13~~ ~~11~~ 9

1–3–4–6| ~~14~~ ~~11~~ 9

1–2–4–6| ~~11~~ 9

First Crashing: We crash the activity 1–3, as it is the critical activity with minimum rank. We crash it for 3 weeks at the rate of ₹ 2 (1000) per day.

Project duration reduced to 16 – 3 = 13 weeks

Total cost = 82 + (3 × 2) = 88 (₹ 1,000)

Second Crashing: Next, we crash the activity 5–6 for 2 weeks at the rate of ₹ 6 (1000) per week, as this activity has the next minimum rank.

Project duration reduced to 13 – 2 = 11 weeks

Total cost = 88 + (2 × 6) = 100 (₹ 1,000).

Third Crashing: After 11 days, we get all the paths that are critical. As there is no activity in common, we crash the activity 5–6 for 2 weeks in the path 1–3–5–6 and crash the activity 4–6 for 2 weeks, which is common to the paths 1–3–4–6 and 1–2–4–6.

Project duration reduces to 11–2 = 9 weeks.

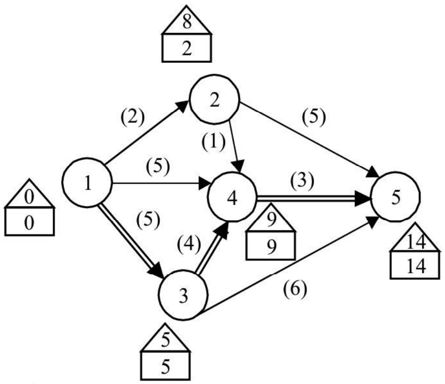
Total cost = 100 + 2 × 6 + 2 × 3 = 118 (₹ 1,000).

The project duration cannot be reduced beyond 9 weeks as all the activities are in crash time. Hence, the optimum duration is 9 weeks with the total cost associated as ₹ 118 (1000).

Example 2.22: The following time-cost table (time in weeks and cost in rupees) applies to a project. Use it to arrive at the network associated with completing the project in minimum time with minimum cost.

Activity	*Normal*		*Crash*	
	Time	*Cost*	*Time*	*Cost*
1–2	2	800	1	1400
1–3	5	1000	2	2000
1–4	5	1000	3	1800
2–4	1	500	1	500
2–5	5	1500	3	2100
3–4	4	2000	3	3000
3–5	6	1200	4	1600
4–5	5	900	3	1600

Solution: As the direct cost is given, we reduce the project duration by an increase in total cost. First we construct the network.



Critical path is given by, 1–3–4–5

Project duration = 14 weeks

Total cost associated = ₹ 8,900

We calculate the cost slope for each activity as given in the following table:

Activity	*Cost slope* = $\frac{C_C - C_N}{T_N - T_C}$
1–2	600 (1) VI
1–3	333.333 (3) III
1–4	400 (2) V
2–4	–
2–5	300 (2) II
3–4	1000 (1) VII
3–5	200 (2) I
4–5	350 (2) IV

The project duration is reduced by crashing the activities. First, the activity that lies on the critical path and ranks the minimum is crashed.

First Crashing: We crash the activity 1–3 for 3 weeks at the rate of ₹ 333.33 per week. Project duration reduced to 14 – 3 = 11 weeks.

$$\begin{aligned}\text{Total cost} &= 8900 + 3\ (333.33)\\ &= ₹\ 9899.99\\ &= ₹\ 9{,}900\end{aligned}$$

Second Crashing: Next, crash the activity 4–5 as this activity lies on the critical path with next minimum rank. Crash 4–5 for 2 weeks at the rate of ₹ 350 per week. Project duration is reduced to 11 – 2 = 9 weeks.

$$\text{Total cost} = 9900 + (2 \times 350) = ₹\ 10{,}600$$

Third Crashing: Next, we crash the activity 3–4 for one week, at the rate of ₹ 1,000 per week. Project duration reduced to 9 – 1 = 8 weeks.

$$\text{Total cost} = 10600 + 1000 = ₹\ 11{,}600$$

As all the activities on the path 1–3–4–5 are in crash time, no more crashing is possible beyond this.

∴ Optimum and minimum project duration is given by 8 weeks with total cost as ₹ 11,600.

2.7.2 PERT Procedure

Step 1: Draw the project network.

Step 2: Compute the expected duration of each activity using the formula.

$$t_e = \frac{t_0 + 4t_m + t_p}{6}$$

Also calculate the expected variance σ^2 of each activity.

i.e., $$\sigma^2 = \left(\frac{t_p - t_o}{6}\right)^2$$

Step 3: Compute the earliest start, earliest finish, latest start, latest finish and total float of each activity.

Step 4: Find the critical path and identify the critical activities.

Step 5: Compute the project length variance σ^2 which is the sum of the variance of all the critical activities and hence find the standard deviation of the project length σ.

Step 6: Calculate the standard normal variable $Z = \dfrac{T_s - T_e}{\sigma}$ where T_S is the scheduled time to complete the project.

T_e = Normal expected project length duration.

σ = Expected standard deviation of the project length.

Using the normal curve, we can estimate the probability of completing the project within a specified time.

Example 2.23: The following table shows the jobs of a network alongwith their time estimates.

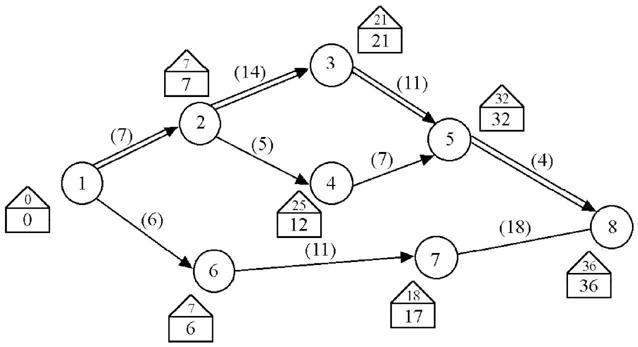
Job	*1 - 2*	*1 - 6*	*2 - 3*	*2 - 4*	*3 - 5*	*4 - 5*	*6 - 7*	*5 - 8*	*7 - 8*
$a(days)$	1	2	2	2	7	5	5	3	8
$m(days)$	7	5	14	5	10	5	8	3	17
$b(days)$	13	14	26	8	19	17	29	9	32

Here, *a* is optimistic time, *m* is most likely time and *b* is pessimistic time estimate.

Draw the project network and find the probability that the project is completed in 40 days.

Solution: First we calculate the expected time and standard deviation for each activity.

Activity	$t_e = \dfrac{t_o + 4t_m + t_p}{6}$	$\sigma^2 = \left(\dfrac{t_p - t_o}{6}\right)^2$
1–2	$\dfrac{1+4\times7+13}{6} = 7$	$\left(\dfrac{13-1}{6}\right)^2 = 4$
1–6	$\dfrac{2+4\times5+14}{6} = 6$	$\left(\dfrac{14-2}{6}\right)^2 = 4$
2–3	$\dfrac{2+4\times14+26}{6} = 14$	$\left(\dfrac{26-2}{6}\right)^2 = 16$
2–4	$\dfrac{2+5\times4+8}{6} = 5$	$\left(\dfrac{8-2}{6}\right)^2 = 1$
3–5	$\dfrac{7+4\times10+19}{6} = 11$	$\left(\dfrac{19-7}{6}\right)^2 = 4$
4–5	$\dfrac{5+5\times4+17}{6} = 7$	$\left(\dfrac{17-5}{6}\right)^2 = 4$
6–7	$\dfrac{5+8\times4+29}{6} = 11$	$\left(\dfrac{29-5}{6}\right)^2 = 16$
5–8	$\dfrac{3+3\times4+9}{6} = 4$	$\left(\dfrac{9-3}{6}\right)^2 = 1$
7–8	$\dfrac{8+4\times17+32}{6} = 18$	$\left(\dfrac{32-8}{6}\right)^2 = 16$



Expected project duration = 36 days.

Critical path 1 → 2 → 3 → 5 → 8

Project length variance = $\sigma^2 = 4 + 16 + 4 + 1$

$= 25$

$\sigma = 5$

Probability that the project will be completed in 40 days is given by,

$$P(Z \leq D)$$

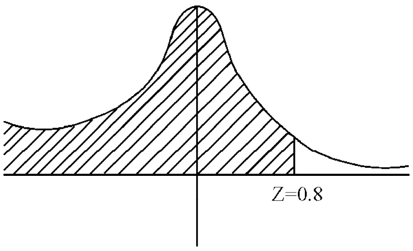
$$D = \frac{T_s - T_e}{\sigma} = \frac{40-36}{5} = \frac{4}{5} = 0.8$$

Area under the normal curve for $\Delta = 0.8$,

$$P(Z \leq 0.8)$$

$$= 0.5 + \phi\,(0.8) \quad [\phi(8) = 0.2881 \text{ (refer } Z - \text{table)}]$$

$$= 0.5 + 0.2881 = 0.7881 = 78.81\%$$



Conclusion: If the project is performed 100 times under the same conditions, there will be 78.81 occasions for this job to be completed in 40 days.

Example 2.24: A small project is composed of seven activities whose time estimates are listed in the table as follow.

	Estimated Duration (Weeks)		
Activity	*Optimistic*	*Most Likely*	*Pessimistic*
1–2	1	1	7
1–3	1	4	7
2–4	2	2	8
2–5	1	1	1
3–5	2	5	14
4–6	2	5	8
5–6	3	6	15

You are required to:

(a) Draw the project network.

(b) Find the expected duration and variance of each activity.

(c) Calculate the early and late occurrence for each event and the expected project length.

(d) Calculate the variance and standard deviations of project length.

(e) What is the probability that the project will be completed:

(*i*) 4 weeks earlier than expected.

(*ii*) Not more than 4 weeks later than expected.

(*iii*) If the project due date is 19 weeks, what is the probability of meeting the due date.

Solution: The expected time and variance of each activity is computed as shown in the table below:

Activity	*a*	*m*	*b*	$t_e = \dfrac{t_o + 4t_m + t_p}{6}$	$\sigma^2 = \left(\dfrac{t_p - t_o}{6}\right)^2$
1–2	1	1	7	2	1
1–3	1	4	7	4	1
1–4	2	2	8	3	1
2–5	1	1	1	1	0
3–5	2	5	14	6	4
4–6	2	5	8	5	1
5–6	3	6	15	7	4

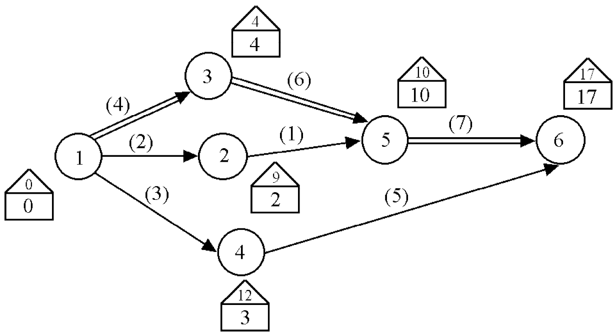
The earliest and the latest occurrence time for each is calculated as below:

$$E_1 = 0;\ E_2 = 0 + 2 = 2$$
$$E_3 = 0 + 4 = 4$$
$$E_4 = 0 + 3 = 3$$
$$E_5 = \text{Max}\ (2 + 1, 4 + 6) = 10$$
$$E_6 = \text{Max}\ (10 + 7, 3 + 5) = 17$$

To determine the latest expected time we start from E_6 being the last event and move backwards subtracting t_e from each activity. Hence, we have

$$L_6 = E_6 = 17$$
$$L_5 = L_6 - 7 = 17 - 7 = 10$$
$$L_4 = 17 - 5 = 12$$
$$L_3 = 10 - 6 = 4$$
$$L_2 = 10 - 1 = 9$$
$$L_1 = \text{Min}\ (9 - 2, 4 - 4, 12 - 3) = 0$$

Using the above information, we get the following network, where the critical path is shown by the double line arrow.



We observe the critical path of the above network as 1 → 3 → 5 → 6.

The expected project duration is 17 weeks, i.e., $T_e = 17$ weeks.

The variance of the project length is given by,

$$\sigma^2 = 1 + 4 + 4 = 9$$

Hence, $\sigma = 3$

(*i*) The probability of completing the project within 4 weeks earlier than expected is given by,

$$P(Z \le D) \text{ where } D = \frac{T_S - T_e}{\sigma}$$

$$D = \frac{\text{Due date} \; - \; \text{Expected date of completion}}{\sqrt{\text{Project variance}}}$$

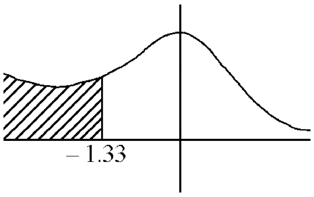
$$D = \frac{17-4-17}{3} = \frac{13-17}{3} = \frac{-4}{3}$$

$$= -1.33$$

$$\therefore P(Z \le -1.33) = 0.5 - \phi(1.33)$$

$$= 0.5 - 0.4082 \text{ (from the table)}$$

$$= 0.0918 = 9.18\%$$



Conclusion: If the project is performed 100 times under the same conditions, then there will be 9 occasions for this job to be completed in 4 weeks earlier than expected.

(*ii*) The probability of completing the project not more than 4 weeks later than expected is given by,

$P\ (Z \le D)$

Where, $D = \dfrac{T_s - T_e}{\sigma}$

Here, $T_s = 17\ + 4 = 21$

NOTES

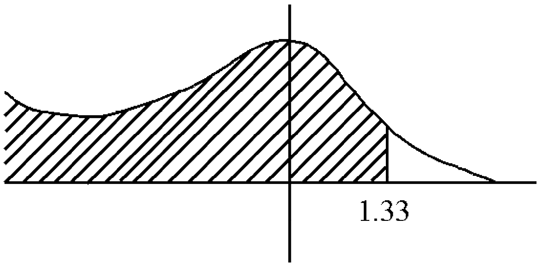
$$D = \frac{21-17}{3} = \frac{4}{3} = 1.33$$

$P\ (Z \le 1.33)$

$= 0.5 + \phi(1.33)$

$= 0.5 + 0.4082$ (from the table)

$= 0.9082 = 90.82\%$



Conclusion: If the project is performed 100 times under the same conditions, then there will be 90.82 occasions when this job will be completed not more than 4 weeks later than expected.

(*iii*) The probability of completing the project within 19 weeks, is given by,

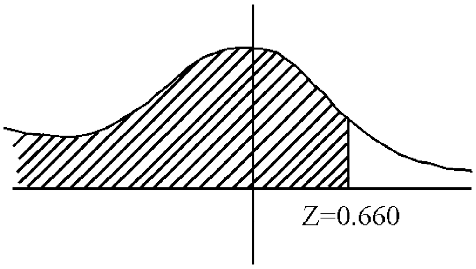
$$P(Z \le D) \text{ where, } D = \frac{19-17}{3} = \frac{2}{3} \qquad \left[\text{Since } T_S = 19\right]$$

$$= 0.666$$

$P\ (Z \le 0.666) = 0.5 + \phi\ (0.666)$

$= 0.5 + 0.2514$ (from the table)

$= 0.7514 = 75.14\%$



Conclusion: If the project is performed 100 times under the same conditions, then there will be 75.14 occasions for this job to be completed in 19 weeks.

Example 2.25: Consider the following project.

Activity	***Time Estimate in Weeks***			***Predecessor***
	t_o	t_m	t_p	
A	3	6	9	None
B	2	5	8	None
C	2	4	6	A
D	2	3	10	B
E	1	3	11	B
F	4	6	8	C,D
G	1	5	15	E

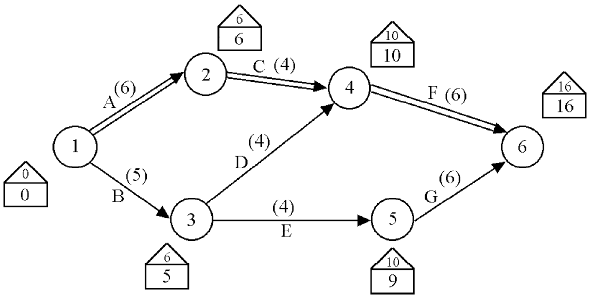
NOTES

Find the path and standard deviation. Also find the probability of completing the project by 18 weeks.

Solution: First we calculate the expected time and variance of each activity as in the following table:

Activity	t_o	t_m	t_p	$t_e = \frac{t_o + 4t_m + t_p}{6}$	$\sigma^2 = \left(\frac{t_p - t_o}{6}\right)^2$
A	3	6	9	$\frac{3+4\times6+9}{6} = 6$	$[(9 - 3)/6]^2 = 1$
B	2	5	8	$\frac{30}{6} = 5$	$[(8 - 2)/6]^2 = 1$
C	2	4	6	24/6 = 4	$[(6 - 2)/6]^2 = 0.444$
D	2	3	10	4	1.777
E	1	3	11	4	2.777
F	4	6	8	6	0.444
G	1	5	15	6	5.444

We construct the network with the help of predecessor relation given in the data.



Critical path is 1 → 2 → 4 → 6 or A → C → F

The project length = 16 weeks.

Project length variance $\sigma^2 = 1 + 0.444 + 0.444 = 1.888$

Standard deviation = $\sigma = 1.374$

The probability of completing the project in 18 weeks is given by:

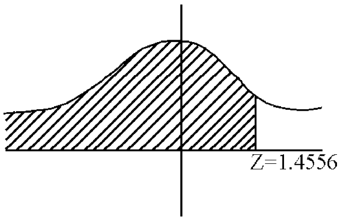
$$P(Z \leq D)$$

Where, $$D = \frac{T_S - T_e}{\sigma}$$

$T_s = 18$; $T_e = 16$; $\sigma = 1.374$

$$D = \frac{18-16}{1.374} = 1.4556$$

$$\begin{aligned} P(Z \leq D) &= P(Z \leq 1.4556) = 0.5 + \phi(1.4456) \\ &= 0.5 + 0.4265 \text{ (from table)} \\ &= 0.9265 = 92.65\% \end{aligned}$$



Conclusion: If the project is performed 100 times under the same conditions, then there will be 92.65 occasions when this job will be completed by 18 weeks.

Example 2.26: Assuming that the expected times are normally distributed, find the probability of meeting the schedule date as given for the network.

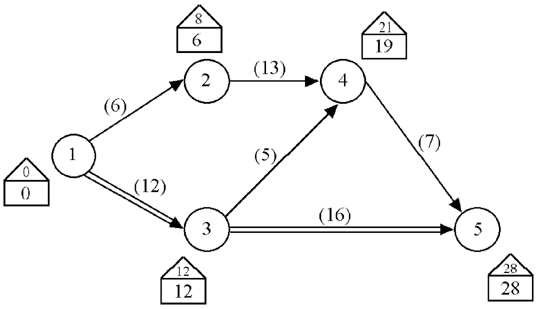
Activity		***Days***	
(i–j)	***Optimistic***	***Most Likely***	***Pessimistic***
	t_o	t_m	t_p
1–2	2	5	14
1–3	9	12	15
2–4	5	14	17
3–4	4	4	10
4–5	8	17	20
3–5	6	6	12

Scheduled project completion date is 30 days. Also, find the date on which the project manager can complete the project with a probability of 0.90.

Solution: The expected time t_e and variance for each activity is calculated in the following table:

Activity	$t_e = (t_o + 4t_m + t_p)/6$	$\sigma^2 = ((t_p - t_o)/6)^2$
1–2	6	4
1–3	12	1
2–4	13	4
3–4	5	1
3–5	16	4
4–5	7	1

To determine the critical path, the earliest expected time and the latest allowable time. First we draw the project network as follows:



The critical path is given by $1 \rightarrow 3 \rightarrow 5$ and the project duration is given by 28 days. Project length variance = $\sigma^2 = 1 + 4 = 5$. Standard deviation = $\sqrt{\sigma^2} = 2.236$.

NOTES

The probability of completing the project within 30 days is given by,

$$P\,(Z \le D), \text{ where } D = \frac{T_s - T_e}{\sigma} = \frac{30 - 28}{2.236} = 0.8944$$

$$P\,(Z \le 0.8944) = 0.5 + \phi\,(0.8944)$$
$$= 0.8133 = 81.33\%$$

Conclusion: If the project is performed 100 times under the same conditions, then there will be 81.33 occasions when the project will be completed in 30 days.

If the probability for the completion of the project is 0.90 then the corresponding value of $Z = 1.29$.

$$Z = \frac{T_s - T_e}{\sigma} = 1.29$$

i.e., $\frac{T_s - 28}{2.236} = 1.29$

$\therefore\ T_s = (1.29)\,(2.236) + 28$

$\therefore\ T_s = 30.88$ weeks

Check Your Progress

6. What are the management functions for the three phases of work involved in a project?
7. Define the term planning.
8. What is an activity?
9. What information is required for the preparation of the network diagram?
10. Write an application of Dijkstra's algorithm.
11. What is saturated edge?
12. Define crash time.
13. Name the three types of time estimates.

2.8 ANSWERS CHECK YOUR PROGRESS

1. Transportation problem deals with transportation of various quantities of a single homogeneous commodity initially stored at various origins to different destinations at the minimum cost.
2. Transportation problems are presented in tabular form with m rows (as origins) and *n* columns (as destinations). Every cell of this table has cost as C_{ij} of transporting one unit of product from *i*th origin to *j*th destination and X_{ij} which is the quantity transported from *i*th origin to *j*th destination.

NOTES

3. If there are m origins and *n* destinations and C_{ij} be the cost of transporting one unit of product from *i*th origin to *j*th destination and X_{ij} be the quantity transported from *i*th origin to *j*th destination and *Z* is the cost given then it can be mathematically represented as:

$$Z = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}\, X_{ij}$$

4. It is a problem of finding an assignment of jobs among persons so that total cost of performing all jobs is at minimum.
5. An assignment problem can be stated in form of $n \times n$ matrix $[C_{ij}]$ with *i* rows and *j* columns of real numbers. Here, C_{ij} is the cost when *i*th person is assigned *j*th job.
6. Management functions involved in three phases of work involved in a project are, planning, scheduling and controlling.
7. Planning is setting of objectives of the project by listing of tasks to be performed and resources available to complete the project.
8. An activity represents an action. It is an effort that consumes time that is needed to complete a part of the overall project.
9. We require the following information for each activity in the project for the preparation of the network diagram:
   (i) The sequencing requirements for an activity must be known, i.e., the set of activities which must be completed prior to the beginning of each specific activity should be known.
   (ii) An estimate of the time each activity will take should also be known.
10. Dijkstra's algorithm is of great use in routing. The concept of shortest path first finds extensive use in network routing protocols.
11. Whenever the flow along an edge is equal to the capacity of that edge, then that edge is said to be saturated.
12. The crash time represents the minimum activity duration time that is possible and any attempts to further crash would only raise the activity cost without reducing the time.
13. The probability distribution of activity times in PERT is based upon three different time estimates made for each activity, namely optimistic time estimate, most likely time estimate and pessimistic time estimate.

2.9 SUMMARY

- The transportation problem is one of the subclasses of LPP (Linear Programming Problem) in which the objective is to transport various quantities of a single homogeneous commodity that are initially stored at various origins to different destinations in such a way that the transportation cost is minimum.
- Any set of non-negative allocations ($X_{ij} > 0$) which satisfies the row and column sum is called a feasible solution.

- A feasible solution is called a basic feasible solution if the number of non-negative allocations is equal to $m + n - 1$, where m is the number of rows and n is the number of columns in a transportation table.
- Optimal solution is a feasible solution (not necessarily basic) which minimizes the total cost.
- Initial solution can be obtained by using any one of the three methods, viz. North West Corner Rule (NWCR), Least Cost Method or Matrix Minima Method and Vogel's Approximation Method (VAM).
- VAM is preferred over the other two methods since the initial basic feasible solution obtained by this method is either optimal or very close to the optimal solution.
- The assignment problem is one of the fundamental combinatorial optimization problems. It helps to find a maximum weight identical in nature to a weighted bipartite graph. The assignment problem is also termed as a special case of transportation problem.
- Job sequencing is basically the planning of the jobs in sequential manner and is an essential part of any work.
- Priority rules give the guidelines for properly sequencing the job whereas Johnson's rule is used to minimize the completion time for a set of jobs to be done on two different machines. Using these rules one can assign jobs and maximize product and profit.
- Any assignment problem is said to be unbalanced if the cost matrix is not a square matrix, i.e., the number of rows and columns are not equal. To make it balanced we add a dummy row or dummy column with all the entries as zero.
- Network scheduling, planning and control and technique used for planning and scheduling large projects in the field of construction, maintenance, fabrication, etc.
- Program Evaluation Review Technique (PERT) and Critical Path Method (CPM), which are two planning and control techniques for keeping a project schedule on track to complete it within the scheduled time.
- An activity is said to be critical if a delay in its start will cause a further delay in the completion of the entire project. The sequence of critical activities in a network is called the critical path.
- Shortest route problem deals with ways of finding the minimum path distance from a selected node which is source to a destination node.
- Flow is a network. The network may be a transportation network through which commodities flow or a pipeline network through which oil flows or any number of other possibilities.
- Critical path method involves the preparation of the network in the form of arrow diagram and its analysis to indicate the critical path.
- Network diagram can be used to identify the activities whose duration should be shortened, so that the completion time of the project can be shortened in the most economic manner.

- PERT is a probabilistic method where the activity times are represented by a probability distribution. This probability distribution of activity times is based upon three different time estimates made for each activity, namely optimistic time estimate, most likely time estimate and pessimistic time estimate.

2.10 KEY TERMS

- **Transportation problem:** A problem for transportation of various quantities of a single homogeneous commodity, initially stored at various origins to different destinations at the minimum cost.
- **Feasible solution:** A set of non-negative allocations where some quantity is transferred from an origin *i* to a destination *j*, ($X_{ij} > 0$) and satisfies the row and column sum is a feasible solution.
- **Basic feasible solution:** A feasible solution where number of non-negative allocations is equal to $m + n - 1$, where m is the number of rows and *n* is the number of columns in a transportation table.
- **Assignment problem:** A problem of finding an assignment of jobs among persons so that total cost of performing all jobs is at minimum.
- **Unbalanced assignment:** An assignment is unbalanced if the cost matrix is not a square matrix.
- **Network:** A graphic representation of logically connected activities and events where activities are presented as arrows and events as nodes.
- **Activity:** An activity represents an action. It is an effort that consumes time that is needed to complete a part of the overall project.
- **Critical path:** It is the path connecting all critical events of the project from start to the completion of the project.
- **PERT:** It stands for Program Evaluation Review Technique (PERT). It is a probabilistic method where activity times are represented by a probability distribution.
- **CPM:** It stands for Critical Path Method (CPM) and is based on determination of the critical path.

2.11 SELF-ASSESSMENT QUESTIONS AND EXERCISES

Short-Answer Questions

1. What is transportation problem?
2. How will you define the transportation algorithm?
3. List the merits and limitations of the North West Corner rule.
4. Vogel's Approximation Method results in the most economical initial basic feasible solution. How?

5. What is an initial basic feasible solution?
6. Write the first three steps of the Hungarian procedure for solving an assignment problem.
7. While solving a problem using the Hungarian method at what stage are you able to make an assignment? Explain in brief.
8. What do you mean by maximization in an assignment problem?
9. What is understood by a project?
10. Define dangling in a network. How can it be avoided?
11. Write two basic differences between PERT and CPM.
12. How are time estimates used in PERT and CPM?
13. Differentiate between float and slack.

Long-Answer Questions

1. Give the mathematical formulation of a transportation problem.
2. Write an algorithm to solve a transportation problem.
3. Define the terms feasible solution, basic solution, non-degenerate solution and optimal solution in a transportation problem.
4. Explain the following briefly with examples:
   (i) North West Corner Rule
   (ii) Least Cost Method
   (iii) Vogel's Approximation Method
5. Explain degeneracy in a transportation problem. Describe a method to resolve it.
6. What do you mean by an unbalanced transportation problem? Explain the process of converting an unbalanced transportation problem into a balanced one.
7. What do you understand by transportation model?
8. Write the steps for finding a solution using the Hungarian method.
9. The following table gives the activities and duration of a construction project.

Activity	***1 - 2***	***1 - 3***	***2 - 3***	***2 - 4***	***3 - 4***	***4 - 5***
Duration (days)	20	25	10	12	6	10

   (*i*) Draw the network for the project.
   (*ii*) Find the critical path.

[Ans. CPM: 1–2–3–4–5]

10. A small project consits of 11 activities *A*, *B*, *C*, ..., *K*. The precedence relationship *A*, *B* can start simultaneously. Given *A*<*C*, *D*, *I*; *B*<*G*, *F*; *D*<*G*, *F*; *F*<*H*, *K*; *G*, *H*<*J*; *I*, *J*, *K*<*E*. The duration of the activities are as follows.

Activity	***A***	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***	***J***	***K***
Duration (Days)	5	3	10	2	8	4	5	6	12	8	9

NOTES

Draw the network of the project. Summarise the CPM calculations in a tabular form computing total, and free floats of activities and hence determine the critical path.

[Ans. Critical path *A–D–F–H–J–E*
Project duration 33 days]

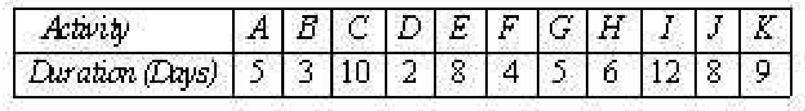
11. The following table gives the activities and duration of a construction project.

Activity	1–2	1–3	2–3	2–4	3–4	4–5
Duration (Days	20	25	10	12	6	10

(*i*) Draw the network for the project.

(*ii*) Find the critical path.

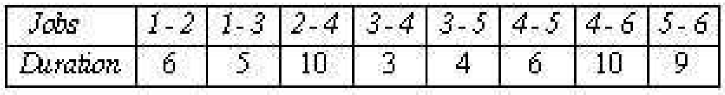
12. A small project consists of 11 activities *A*, *B*, *C*, ..., *K*. The precedence relationship *A*, *B* can start simultaneously. Given *A*<*C*, *D*, *I*; *B*<*G*, *F*; *D*<*G*, *F*; *F*<*H*, *K*; *G*, *H*<*J*; *I*, *J*, *K*<*E*. The duration of the activities are as follows.



Activity	A	B	C	D	E	F	G	H	I	J	K
Duration (Days)	5	3	10	2	8	4	5	6	12	8	9

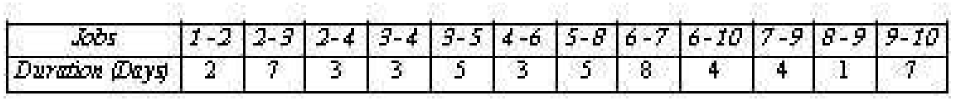
Draw the network of the project. Summarize the CPM calculations in a tabular form computing total, and free floats of activities and hence determine the critical path.

13. Draw the network and determine the critical path for the given data. Also calculate all the floats involved in CPM.



Jobs	1-2	1-3	2-4	3-4	3-5	4-5	4-6	5-6
Duration	6	5	10	3	4	6	10	9

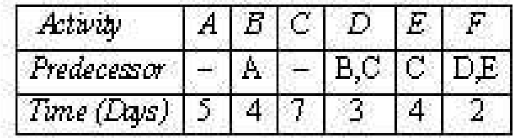
14. A small maintenance project consists of the following 12 jobs.



Jobs	1-2	2-3	2-4	3-4	3-5	4-6	5-8	6-7	6-10	7-9	8-9	9-10
Duration (Days)	2	7	3	3	5	3	5	8	4	4	1	7

Draw the arrow network of the project. Summarize CPM calculations in a tabular form calculating the three types of floats and hence determine the critical path.

15. Consider the following data for activities in a given project.



Activity	A	B	C	D	E	F
Predecessor	–	A	–	B,C	C	D,E
Time (Days)	5	4	7	3	4	2

Draw the arrow diagram for the project. Compute the earliest and the latest event times. What is the minimum project completion time? List the activities on the critical path.

2.12 FURTHER READING

NOTES

Kalavathy, S. 2002. *Operations Research*. New Delhi: Vikas Publishing House Pvt. Ltd.

Gupta, P.K. and D.S. Hira. 2002. *Introduction to Operations Research*. New Delhi: S. Chand And Company Limited.

Taha, H.A. 2006. *Operations Research: An Introduction*, 8th Edition. New Delhi: Prentice-Hall of India.

Jensen, Paul A. and Jonathan F. Bard. 2003. *Operations Research Models and Methods*. New York: John Wiley & Sons.

Arumugam, R. S. 2006. *Operations Research*. Palayamkottai (Tamil Nadu): New Gamma Publications.

Sundharesan, V., K. S. Ganapathy and K. Ganesan. 2017. *Resource Management Techniques (Operations Research)*. Chennai: A. R. Publications.

Gillett, Billy E. 2007. *Introduction to Operations Research*. New Delhi: Tata McGraw-Hill.

Ackoff, R. L. and M. W. Sasieni. 1968. *Fundamentals of Operations Research*. New York: John Wiley & Sons Inc.

Kothari, C. R. 1992. *An Introduction to Operational Research*. New Delhi: Vikas Publishing House Pvt. Ltd.

Sharma, J. K. 2001. *Operations Research: Theory and Applications*. New Delhi: Macmillan India Ltd.

Swaroop, Kanti, P.K. Gupta and Man Mohan. 2001. *Operations Research*. New Delhi: Sultan Chand & Sons.

UNIT 3 DYNAMIC PROGRAMMING

NOTES

Structure

3.0 INTRODUCTION

Dynamic programming is mathematical optimization method. The method was developed by Richard E. Bellman in the 1950s and has found applications in numerous fields, from aerospace engineering to economics.

Dynamic programming are problems in which a series of interrelated decisions are required. The objective in such problems happens to be to find a combination of decisions that will optimize some appropriate measure of effectiveness. For instance, it might be desirable to specify a series of production decisions that will minimize total costs or a series of marketing decisions that will maximize total revenues or a series of pricing decisions over a given time period with a view to is popularly described as the technique of dynamic programming.

Game theory is the study of mathematical models of strategic interaction among rational decision-makers. It has applications in all fields of social science, as well as in logic, systems science and computer science. Originally, it addressed zero-sum games, in which each participant's gains or losses are exactly balanced by those of the other participants. In the 21st century, game theory applies to a wide range of behavioural relations, and is now an umbrella term for the science of logical decision making in humans, animals, and computers. Modern game theory began with the idea of mixed-strategy equilibria in two-person zero-sum games and its proof by John von Neumann. The von Neumann's original proof used the Brouwer fixed-point theorem on continuous mappings into compact convex sets, which became a standard method in game theory and mathematical economics. His paper was followed by the 1944 book *Theory of Games and Economic Behaviour*, co-written with Oskar Morgenstern, which considered cooperative games of several players. The second edition of this book provided an axiomatic theory of expected utility, which allowed mathematical statisticians and economists to treat decision-making under uncertainty.

NOTES

In this unit, you will learn about the dynamic programming, game theory, two- person zero-sum games, games with mixed strategies and graphical solutions.

3.1 OBJECTIVES

After going through this unit, you will be able to:

- Understand dynamic problems
- Explain the concept and significance of game theory
- Describe the process of two-person zero-sum games
- Discuss the significance of mixed strategies
- Explain the graphic solution of 2 × n and m ×2 games

3.2 DYNAMIC PROGRAMMING

There are problems in which a series of interrelated decisions are required. The objective in such problems happens to be to find a combination of decisions that will optimize some appropriate measure of effectiveness. For instance, it might be desirable to specify a series of production decisions that will minimize total costs or a series of marketing decisions that will maximize total revenues or a series of pricing decisions over a given time period with a view to maximize the present value of profits or a series of sequential decisions pertaining to problems such as allocation, replacement, scheduling, routing, inventory, etc. Such decision problems are often solved through sequential decision theory or what is popularly described as the technique of dynamic programming.

A multistage decision system in which each decision and state variables can take only finite number of values which can be represented graphically by a **decision tree**.

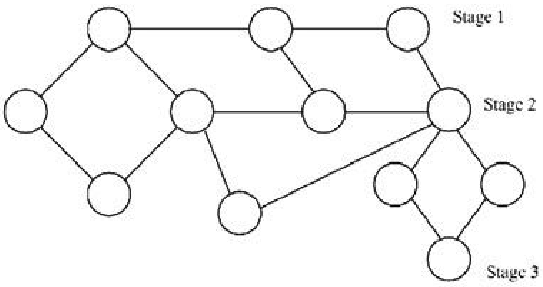


Fig 3.1

Circles represent nodes corresponding to stages, and lines between circles denote arcs, corresponding to decisions. The dynamic programming technique deals with such situations by dividing the given problem into sub-problems or stages.

Bellman's principle of optimality states that 'An optimal policy (a set of decisions) has the property that whatever the initial state and decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.'

The problem that does not satisfy the principle of optimality cannot be solved by the dynamic programming method.

Dynamic Programming Algorithm

The solution of a multistage problem by dynamic programming involves the following steps:

Step 1: Identify the decision variables and specify the objective function to be optimized under certain limitations, if any.

Step 2: Decompose the given problem into a number of smaller subproblems. Identify the state variable at each stage.

Step 3: Write down the general recursive relationship for computing the optimal policy. Decide whether forward or backward method is to be followed to solve the problem.

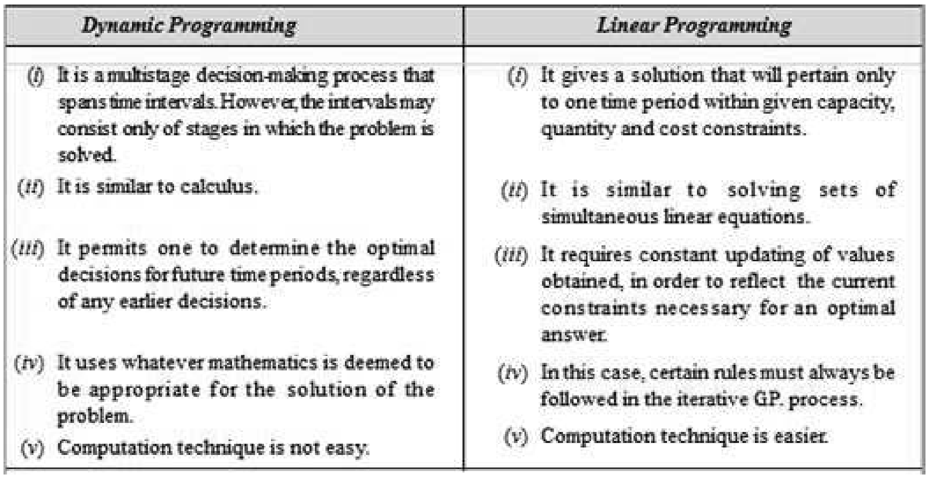
Step 4: Construct appropriate stages to show the required values of the return function at each stage.

Step 5: Determine the overall optimal policy or decisions and its value at each stage. There may be more than one such optimal policies.

Difference between Linear and Dynamic Programming

Table 3.2 shows the differences between Linear and Dynamic Programming.

Table 3.1 *Difference between Linear and Dynamic Programming*



Dynamic Programming	*Linear Programming*
(*i*) It is a multistage decision-making process that spans time intervals. However, the intervals may consist only of stages in which the problem is solved.	(*i*) It gives a solution that will pertain only to one time period within given capacity, quantity and cost constraints.
(*ii*) It is similar to calculus.	(*ii*) It is similar to solving sets of simultaneous linear equations.
(*iii*) It permits one to determine the optimal decisions for future time periods, regardless of any earlier decisions.	(*iii*) It requires constant updating of values obtained, in order to reflect the current constraints necessary for an optimal answer.
(*iv*) It uses whatever mathematics is deemed to be appropriate for the solution of the problem.	(*iv*) In this case, certain rules must always be followed in the iterative G.P. process.
(*v*) Computation technique is not easy.	(*v*) Computation technique is easier.

3.2.1 Deterministic and Probabilistic Dynamic Programming

This section elaborates upon the dynamic programming approach to deterministic problems, where the state at the next stage is completely determined by the state and policy decision at the current stage. The probabilistic case, where there is a probability distribution for what the next state will be, is discussed in the next section.

NOTES

Deterministic dynamic programming can be described diagrammatically as shown in Fig. 3.2. Thus, at stage *n* the process will be in some state s_n. Making policy decision x_n then moves the process to some state s_{n+1} at stage $n + 1$. The contribution thereafter to the objective function under an optimal policy has been previously calculated to be $f^*n_1(s_{n+1})$. The policy decision x_n also makes some contribution to the objective function. Combining these two quantities in an appropriate way provides $f_n(s_n, x_n)$, the contribution of stages *n* onward to the objective function. Optimizing with respect to x_n then gives $f_n^*(s_n)$ $f_n(s_n, x_n^*)$. After x_n^* and $f_n^*(s_n)$ are found for each possible value of s_n, the solution procedure is ready to move back one stage.

One way of categorizing deterministic dynamic programming problems is by the form of the objective function. For example, the objective might be to minimize the sum of the contributions from the individual stages (as for the stagecoach problem), or to maximize such a sum, or to minimize a product of such terms, and so on. Another categorization is in terms of the nature of the set of states for the respective stages. In particular, states s_n might be representable by a discrete state variable (as for the stagecoach problem) or by a continuous state variable, or perhaps a state vector (more than one variable) is required.

Several examples are presented to illustrate these various possibilities. More importantly, they illustrate that these apparently major differences are actually quite inconsequential (except in terms of computational difficulty) because the underlying basic structure shown in Fig. 3.2 always remains the same.

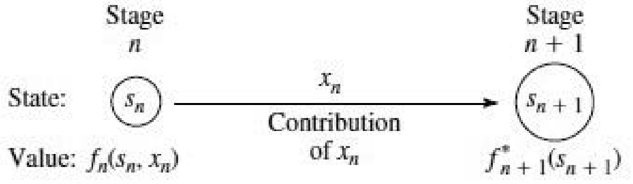


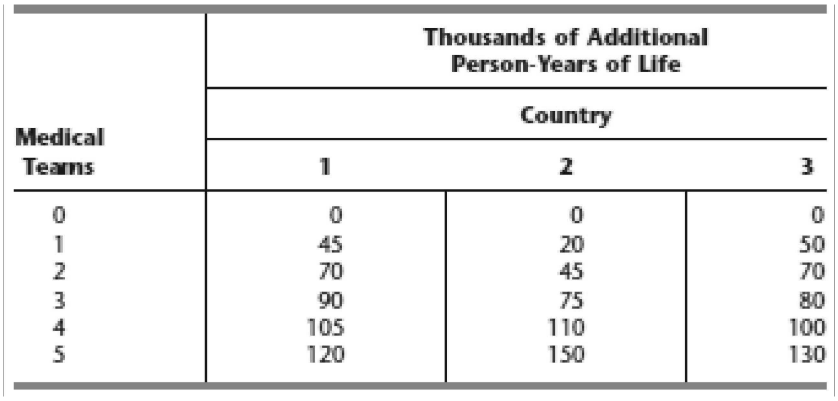
Fig.3.2 *The Basic Structure for Deterministic Dynamic Programming*

Example 3.1: Distributing Medical Teams to Countries

The World Health Council is devoted to improving health care in the underdeveloped countries of the world. It now has five medical teams available to allocate among three such countries to improve their medical care, health education, and training programs. Therefore, the council needs to determine how many teams (if any) to allocate to each of these countries to maximize the total effectiveness of the five teams. The teams must be kept intact, so the number allocated to each country must be an integer.

The measure of performance being used is additional person-years of life. (For a particular country, this measure equals the *increased life expectancy* in years times the country's population.) Table 3.2 gives the estimated additional person-years of life (in multiples of 1,000) for each country for each possible allocation of medical teams. Which allocation maximizes the measure of performance?

Table 3.2 *Data for the World Health Council Problem*



Medical Teams	Thousands of Additional Person-Years of Life		
	Country		
	1	2	3
0	0	0	0
1	45	20	50
2	70	45	70
3	90	75	80
4	105	110	100
5	120	150	130

NOTES

Formulation: This problem requires making three *interrelated decisions,* namely, how many medical teams to allocate to each of the three countries. Therefore, even though there is no fixed sequence, these three countries can be considered as the three stages in a dynamic programming formulation. The decision variables x_n (n = 1, 2, 3) are the number of teams to allocate to stage (country) n.

The identification of the states may not be readily apparent. To determine the states, we ask questions such as the following. What is it that changes from one stage to the next? Given that the decisions have been made at the previous stages, how can the status of the situation at the current stage be described? What information about the current state of affairs is necessary to determine the optimal policy hereafter? On these bases, an appropriate choice for the "state of the system" is

s_n number of medical teams still available for allocation to remaining countries (n, . . . , 3).

Thus, at stage 1 (country 1), where all three countries remain under consideration for allocations, $s_1 = 5$. However, at stage 2 or 3 (country 2 or 3), s_n is just 5 minus the number of teams allocated at preceding stages, so that the sequence of states is

$s_1 \;\; 5, s_2 \;\; 5 \;\; x_1, s_3 \;\; s_2 \;\; x_2.$

With the dynamic programming procedure of solving backward stage by stage, when we are solving at stage 2 or 3, we shall not yet have solved for the allocations at the preceding stages. Therefore, we shall consider every possible state we could be in at stage 2 or 3, namely, sn 0, 1, 2, 3, 4, or 5.

Figure 3.3 shows the states to be considered at each stage. The links (line segments) show the possible transitions in states from one stage to the next from making a feasible allocation of medical teams to the country involved. The numbers shown next to the links are the corresponding contributions to the measure of performance, where these numbers come from Table 3.2. From the perspective of this figure, the overall problem is to find the path from the initial state 5 (beginning stage 1) to the final state 0 (after stage 3) that maximizes the sum of the numbers along the path.

NOTES

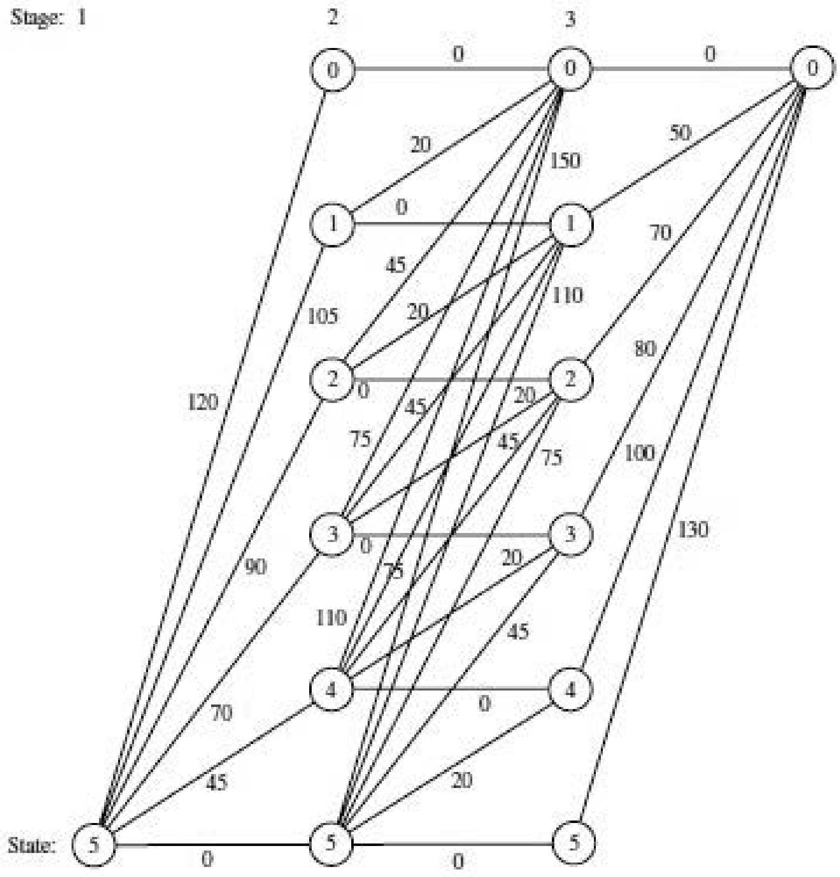


Fig. 3.3 Graphical Display of the World Health Council Problem, Showing the Possible States at Each Stage, the Possible Transitions in States, and the Corresponding Contributions to the Measure of Performance.

To state the overall problem mathematically, let $p_i(x_i)$ be the measure of performance from allocating *xi* medical teams to country *i*, as given in Table 3.2. Thus, the objective is to choose $x1, x_2, x_3$ so as to

$$\text{Maximize} \quad \sum_{i=1}^{3} p_i(x_i),$$

subject to

$$\sum_{i=1}^{3} x_i = 5,$$

and

x_i are nonnegative integers.

$$f_n(s_n, x_n) = p_n(x_n) + \max \sum_{i=n+1}^{3} p_i(x_i),$$

where the maximum is taken over $x_{n+1} \ldots, x_3$ such that

$$\sum_{i=n}^{3} x_i = s_n$$

and the x_i are nonnegative integers, for $n = 1, 2, 3$. In addition,

$$f_n^*(s_n) = \max_{x_n=0,1,\ldots,s_n} f_n(s_n, x_n)$$

NOTES

Therefore,

$$f_n(s_n, x_n) = p_n(x_n) + f^*_{n+1}(s_n - x_n)$$

(with f_4* defined to be zero). These basic relationships are summarized in Fig. 3.4.

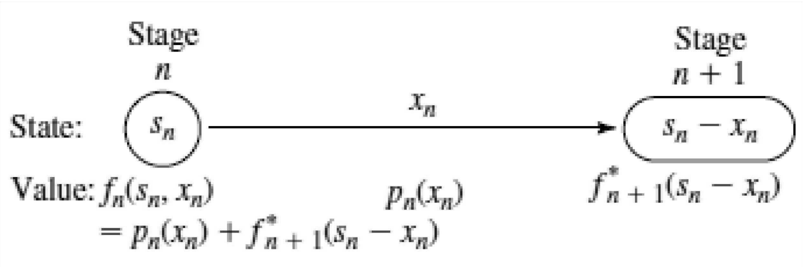


Fig. 3.4 *The Basic Structure for the World Health Council Problem.*

Consequently, the *recursive relationship* relating functions f_1*,f_2*, and f_3* for this problem is

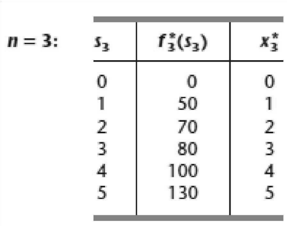
$$f^*_n(s_n) = \max_{x_n=0,1,\ldots,s_n} \{p_n(x_n) + f^*_{n+1}(s_n - x_n)\}, \qquad \text{for } n = 1, 2.$$

For the last stage ($n = 3$),

$$f^*_3(s_3) = \max_{x_3=0,1,\ldots,s_3} p_3(x_3).$$

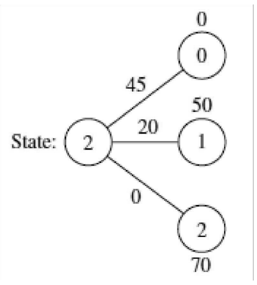
The resulting dynamic programming calculations are given next.

Solution Procedure: Beginning with the last stage (n =3), we note that the values of $p_3(x_3)$ are given in the last column of Table 3.2 and these values keep increasing as we move down the column. Therefore, with s_3 medical teams still available for allocation to country 3, the maximum of $p_3(x_3)$ is automatically achieved by allocating all s_3 teams; so x_3* s_3 and f_3*(s_3) $p_3(s_3)$, as shown in the following table.


n = 3:

s_3	$f^*_3(s_3)$	x^*_3
0	0	0
1	50	1
2	70	2
3	80	3
4	100	4
5	130	5

We now move backward to start from the next-to-last stage (n = 2). Here, finding x_2* requires calculating and comparing $f_2(s_2, x_2)$ for the alternative values of x_2, namely, x_2 ..., 0, 1, . . . , s_2. To illustrate, we depict this situation when $s_2 = 2$ graphically:



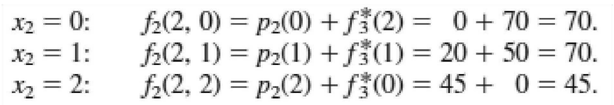
NOTES

This diagram corresponds to Fig. 3.4 except that all three possible states at stage 3 are shown. Thus, if $x_2 = 0$, the resulting state at stage 3 will be $s_2 - x_2 = 2 - 0 = 2$, whereas $x_2 = 1$ leads to state 1 and $x_2 = 2$ leads to state 0. The corresponding values of $p_2(x_2)$ from the country 2 column of Table 3.2 are shown along the links, and the values of $f_3^*(s_2 - x_2)$ from the $n = 3$ table are given next to the stage 3 nodes. The required calculations for this case of $s_2 = 2$ are summarized below.

Formula: $f_2(2, x_2) = p_2(x_2) + f_3^*(2 - x_2)$.

$p_2(x_2)$ is given in the country 2 column of Table 3.2.

$f_3^*(2 - x_2)$ is given in the $n = 3$ table (bottom of preceding page).



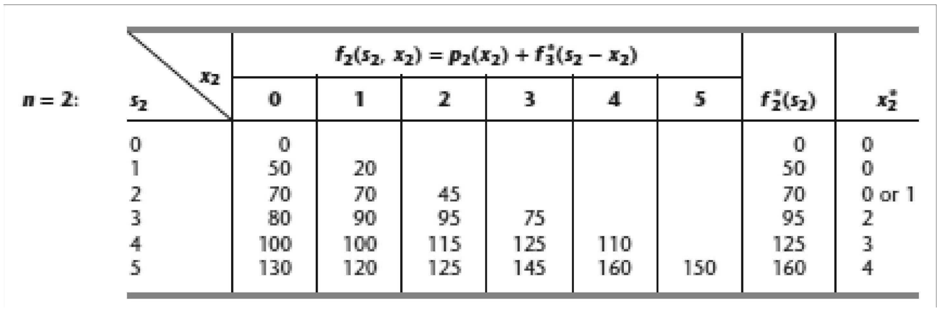
$x_2 = 0$: $f_2(2, 0) = p_2(0) + f_3^*(2) = 0 + 70 = 70.$

$x_2 = 1$: $f_2(2, 1) = p_2(1) + f_3^*(1) = 20 + 50 = 70.$

$x_2 = 2$: $f_2(2, 2) = p_2(2) + f_3^*(0) = 45 + 0 = 45.$

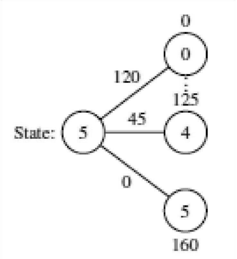
Because the objective is maximization, $x_2^* = 0$ or 1 with $f_2^*(2) = 70$.

Proceeding in a similar way with the other possible values of s_2 (try it) yields the following table.



$n = 2$: s_2 \ x_2	$f_2(s_2, x_2) = p_2(x_2) + f_3^*(s_2 - x_2)$							
s_2	0	1	2	3	4	5	$f_2^*(s_2)$	x_2^*
0	0						0	0
1	50	20					50	0
2	70	70	45				70	0 or 1
3	80	90	95	75			95	2
4	100	100	115	125	110		125	3
5	130	120	125	145	160	150	160	4

We now are ready to move backward to solve the original problem where we are starting from stage 1 ($n = 1$). In this case, the only state to be considered is the starting state of $s_1 = 5$, as depicted below.



Since allocating x_1 medical teams to country 1 leads to a state of $5 - x_1$ at stage 2, a choice of $x_1 = 0$ leads to the bottom node on the right, $x = 1$ leads to the next node up, and so forth up to the top node with $x_1 = 5$. The corresponding $p_1(x_1)$ values from Table 3.2 are shown next to the links. The numbers next to the nodes are obtained from the $f_2^*(s_2)$ column of the n = 2 table. As with $n = 2$, the calculation needed for each alternative value of the decision variable involves adding the corresponding link value and node value, as summarized below.

Formula: $f_1(5, x_1) = p_1(x_1) + f_2^*(5 - x_1)$.

$p_1(x_1)$ is given in the country 1 column of Table 3.2

f_2*$(5 - x_1)$ is given in the $n = 2$ table.

$x_1 = 0$: $f_1(5, 0) = p_1(0) + f_2^*(5) = 0 + 160 = 160.$
$x_1 = 1$: $f_1(5, 1) = p_1(1) + f_2^*(4) = 45 + 125 = 170.$
$\vdots$
$x_1 = 5$: $f_1(5, 5) = p_1(5) + f_2^*(0) = 120 + 0 = 120.$

The similar calculations for x_1 = 2, 3, 4 (try it) verify that x_1* = 1 with f_1*(5) = 170, as shown in the following table.

$n = 1$:

x_2 / s_1	$f_1(s_1, x_1) = p_1(x_1) + f_2^*(s_1 - x_1)$ 0	1	2	3	4	5	$f_1^*(s_1)$	x_1^*
5	160	170	165	160	155	120	170	1

Thus, the optimal solution has x_1* = 1, which makes s_2 = 5 – 1 = 4, so x_2* = 3, which makes s_3 = 4 – 3 = 1, so x_3* = 1. Since f_1*(5) = 170, this (1, 3, 1) allocation of medical teams to the three countries will yield an estimated total of 170,000 additional personyears of life, which is at least 5,000 more than for any other allocation.

These results of the dynamic programming analysis also are summarized in Fig. 3.5

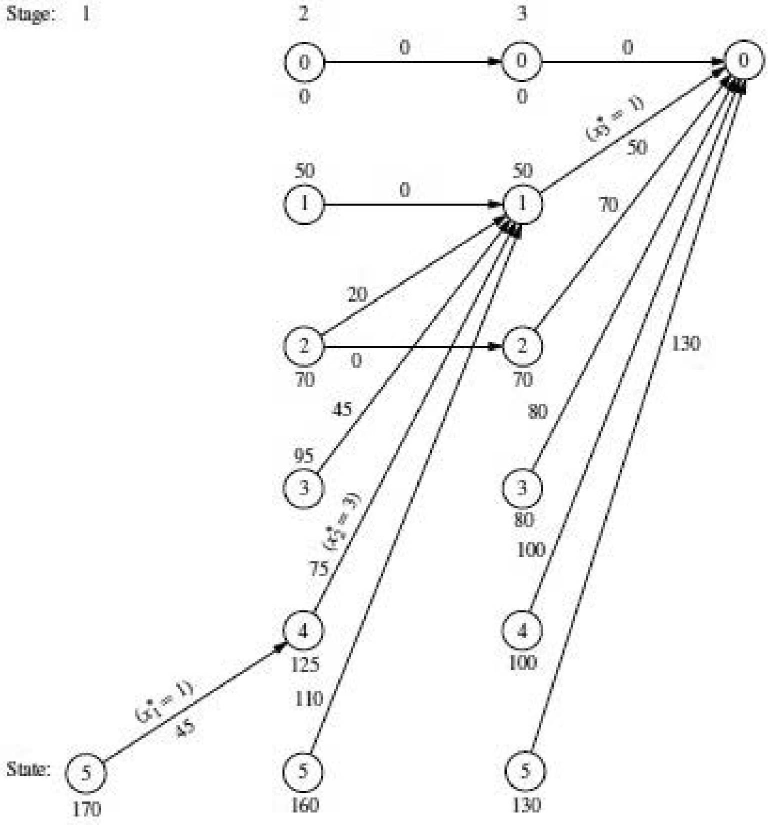


Fig. 3.5 *Graphical Display of the Dynamic Programming Solution of the World Health Council Problem. An Arrow from State s_n to State s_{n-1} Indicates that an Optimal Policy Decision from State s_n is to Allocate $(s_n - s_{n-1})$ Medical Teams to Country n. Allocating the Medical Teams in this way when following the Boldfaced Arrows from the Initial State to the Final State gives the Optimal Solution.*

NOTES

A Prevalent Problem Type—The Distribution of Effort Problem

The preceding example illustrates a particularly common type of dynamic programming problem called the distribution of effort problem. For this type of problem, there is just one kind of resource that is to be allocated to a number of activities. The objective is to determine how to distribute the effort (the resource) among the activities most effectively. For the World Health Council example, the resource involved is the medical teams, and the three activities are the health care work in the three countries.

Assumptions: There are some key differences between the distribution of effort problem and linear programming that help illuminate the general distinctions between dynamic programming and other areas of mathematical programming. One key difference is that the distribution of effort problem involves only one resource (one functional constraint), whereas linear programming can deal with thousands of resources.

On the other hand, the distribution of effort problem is far more general than linear programming in other ways. Consider the four assumptions of linear programming proportionality, additivity, divisibility, and certainty. Proportionality is routinely violated by nearly all dynamic programming problems, including distribution of effort problems (e.g., Table 3.2 violates proportionality). Divisibility also is often violated, as in Example 3.2, where the decision variables must be integers. In fact, dynamic programming calculations become more complex when divisibility does hold. Although we shall consider the distribution of effort problem only under the assumption of certainty, this is not necessary, and many other dynamic programming problems violate this assumption as well.

Of the four assumptions of linear programming, the *only* one needed by the distribution of effort problem (or other dynamic programming problems) is *additivity* (or its analog for functions involving a *product* of terms). This assumption is needed to satisfy the *principle of optimality* for dynamic programming.

Formulation: Because they always involve allocating one kind of resource to a number of activities, distribution of effort problems always have the following dynamic programming formulation (where the ordering of the activities is arbitrary):

Stage $n =$ activity n ($n = 1, 2, \dots, N$).

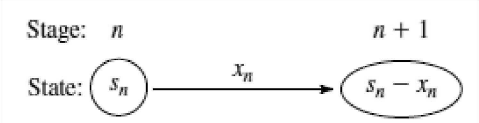
$x_n =$ amount of resource allocated to activity n.

State $s_n =$ amount of resource still available for allocation to remaining activities

($n, \dots, N$).

The reason for defining state *sn* in this way is that the amount of the resource still available for allocation is precisely the information about the current state of affairs (entering stage *n*) that is needed for making the allocation decisions for the remaining activities.

When the system starts at stage n in state s_n, the choice of x_n results in the next state at stage $n + 1$ being $s_{n+1} = s_n - x_n$, as depicted below:



Note how the structure of this diagram corresponds to the one shown in Fig. 3.4 for the World Health Council example of a distribution of effort problem. What will differ from one such example to the next is the rest of what is shown in Fig. 3.4, namely, the relationship between $f_n(s_n$, xn) and $f^*_{n+1}(s_n - x_n)$, and then the resulting recursive relationship between the f_n^* and f^*_{n+1} functions. These relationships depend on the particular objective function for the overall problem.

The structure of the next example is similar to the one for the World Health Council because it, too, is a distribution of effort problem. However, its recursive relationship differs in that its objective is to minimize a product of terms for the respective stages.

At first glance, this example may appear not to be a deterministic dynamic programming problem because probabilities are involved. However, it does indeed fit our definition because the state at the next stage is completely determined by the state and policy decision at the current stage.

Example 3.2: Distributing Scientists to Research Teams

A government space project is conducting research on a certain engineering problem that must be solved before people can fly safely to Mars. Three research teams are currently trying three different approaches for solving this problem. The estimate has been made that, under present circumstances, the probability that the respective teams—call them 1, 2, and 3—will not succeed is 0.40, 0.60, and 0.80, respectively. Thus, the current probability that all three teams will fail is (0.40)(0.60)(0.80) 0.192. Because the objective is to minimize the probability of failure, two more top scientists have been assigned to the project.

Table 3.3 gives the estimated probability that the respective teams will fail when 0, 1, or 2 additional scientists are added to that team. Only integer numbers of scientists are considered because each new scientist will need to devote full attention to one team. The problem is to determine how to allocate the two additional scientists to minimize the probability that all three teams will fail.

Formulation: In this case, scientists replace medical teams as the kind of resource involved, and research teams replace countries as the activities.

Therefore, instead of medical teams being allocated to countries, scientists are being allocated to research teams. The only basic difference between the two problems is in their objective functions.

With so few scientists and teams involved, this problem could be solved very easily by a process of exhaustive enumeration. However, the dynamic programming solution is presented for illustrative purposes.

In this case, stage n (n = 1, 2, 3) corresponds to research team n, and the state *sn* is the number of new scientists *still available* for allocation to the remaining teams. The decision variables *xn* (n = 1, 2, 3) are the number of additional scientists allocated to team n.

Let $p_i(x_i)$ denote the probability of failure for team i if it is assigned *xi* additional scientists, as given by Table 3.2. If we let denote multiplication, the government's objective is to choose x_1, x_2, x_3 so as to

NOTES

Minimize $\prod_{i=1}^{3} p_i(x_i) = p_1(x_1)p_2(x_2)p_3(x_3)$,

subject to

$$\sum_{i=1}^{3} x_i = 2$$

Table 3.3 *Data for the Government Space Project Problem*

New Scientists	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30

and

x_i are nonnegative integers.

Consequently, $f_n(s_n, x_n)$ for this problem is

$$f_n(s_n, x_n) = p_n(x_n) \cdot \min \prod_{i=n+1}^{3} p_i(x_i),$$

where the minimum is taken over $x_{n+1}, \ldots, x_3$ such that

$$\sum_{i=n}^{3} x_i = s_n$$

and

x_i are nonnegative integers,

for n = 1, 2, 3. Thus,

$$f_n^*(s_n) = \min_{x_n=0,1,\ldots,s_n} f_n(s_n, x_n),$$

where

$$f_n(s_n, x_n) = p_n(x_n) \cdot f_{n+1}^*(s_n - x_n)$$

(with f4* defined to be 1). Figure 3.6 summarizes these basic relationships.

Thus, the recursive relationship relating the f_1*, f_2*, and f_3* functions in this case is

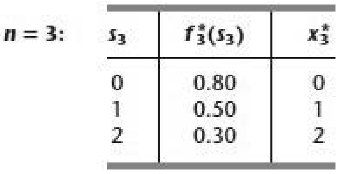
$$f_n^*(s_n) = \min_{x_n=0,1,\ldots,s_n} \{p_n(x_n) \cdot f_{n+1}^*(s_n - x_n)\}, \qquad \text{for } n = 1, 2,$$

and, when $n = 3$,

$$f_3^*(s_3) = \min_{x_3 = 0,1,\ldots,s_3} p_3(x_3).$$

NOTES

Solution Procedure: The resulting dynamic programming calculations are as follows:



$n = 3$:

s_3	$f_3^*(s_3)$	x_3^*
0	0.80	0
1	0.50	1
2	0.30	2

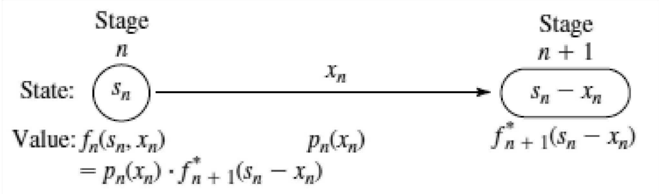
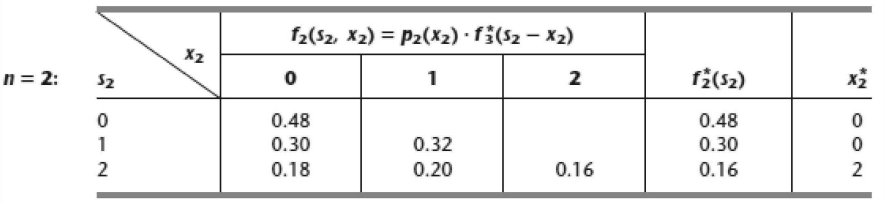
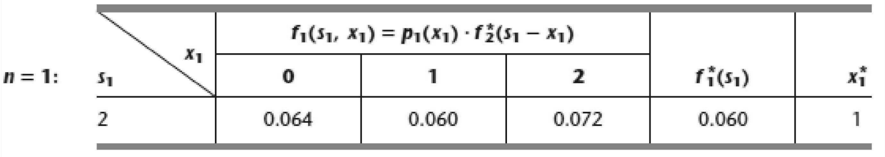


Fig. 3.6 *The Basic Structure for the Government Space Project Problem*



$n = 2$:

x_2 / s_2	$f_2(s_2, x_2) = p_2(x_2) \cdot f_3^*(s_2 - x_2)$			$f_2^*(s_2)$	x_2^*
	0	1	2		
0	0.48			0.48	0
1	0.30	0.32		0.30	0
2	0.18	0.20	0.16	0.16	2



$n = 1$:

x_1 / s_1	$f_1(s_1, x_1) = p_1(x_1) \cdot f_2^*(s_1 - x_1)$			$f_1^*(s_1)$	x_1^*
	0	1	2		
2	0.064	0.060	0.072	0.060	1

Therefore, the optimal solution must have $x_1* = 1$, which makes $s_2 = 2 - 1 = 1$, so that $x_2* = 0$, which makes $s_3 = 1 - 0 = 1$, so that $x_3* = 1$. Thus, teams 1 and 3 should each receive one additional scientist. The new probability that all three teams will fail would then be 0.060.

Probabilistic Dynamic Programming

Probabilistic dynamic programming differs from deterministic dynamic programming in that the state at the next stage is not completely determined by the state and policy decision at the current stage. Rather, there is a probability distribution for what the next state will be. However, this probability distribution still is completely determined by the state and policy decision at the current stage. The resulting basic structure for probabilistic dynamic programming is described diagrammatically in Fig. 3.7.

For the purposes of this diagram, we let S denote the number of possible states at stage $n + 1$ and label these states on the right side as 1, 2, . . . , S. The system goes to state i with probability p_i (i 1, 2, . . . , S) given state s_n and decision x_n at stage n. If the system goes to state i, C_i is the contribution of stage n to the objective function.

When Fig. 3.7 is expanded to include all the possible states and decisions at all the stages, it is sometimes referred to as a **decision tree.** If the decision tree is not too large, it provides a useful way of summarizing the various possibilities.

NOTES

Because of the probabilistic structure, the relationship between $f_n(s_n, x_n)$ and the $f*_{n+1}(s_{n+1})$ necessarily is somewhat more complicated than that for deterministic dynamic programming. The precise form of this relationship will depend upon the form of the overall objective function.

To illustrate, suppose that the objective is to minimize the expected sum of the contributions from the individual stages. In this case, $f_n(s_n, x_n)$ represents the minimum ex-pected sum from stage n onward, given that the state and policy decision at stage n are s_n and x_n, respectively. consequently.

$$f_n(s_n, x_n) = \sum_{i=1}^{S} p_i[C_i + f^*_{n+1}(i)],$$

with

$$f^*_{n+1}(i) = \min_{x_{n+1}} f_{n+1}(i, x_{n+1}),$$

where this minimization is taken over the feasible values of x_{n+1}.

Example 3.3 has this same form. Example 3.4 will illustrate another form.

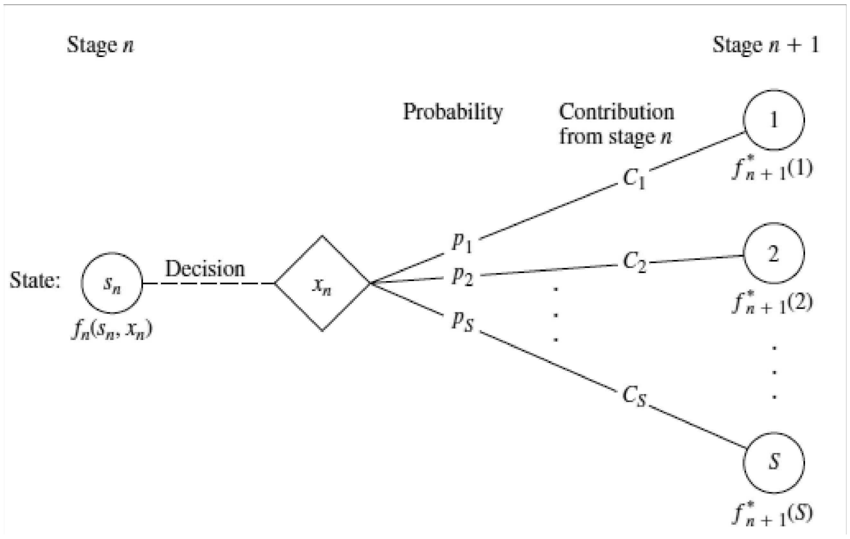


***Fig. 3.7** The Basic Structure for Probabilistic Dynamic Programming*

Example 3.3: Determining Reject Allowances

The Standard Manufacturing Company has received an order to supply one item of a particular type. However, the customer has specified such stringent quality requirements that the manufacturer may have to produce more than one item to obtain an item that is acceptable. The number of *extra* items produced in a production run is called the reject allowance. Including a reject allowance is common practice when producing for a custom order, and it seems advisable in this case.

The manufacturer estimates that each item of this type that is produced will be acceptable with probability1/2 and defective (without possibility for rework) with probability 1/2. Thus, the number of acceptable items produced in a lot of size L will have a binomial distribution; i.e., the probability of producing no acceptable items in such a lot is $\left(\frac{1}{2}\right)^L$.

Marginal production costs for this product are estimated to be ₹100 per item (even if defective), and excess items are worthless. In addition, a setup cost of ₹300 must be incurred whenever the production process is set up for this product, and a completely new setup at this same cost is required for each subsequent production run if a lengthy in spection procedure reveals that a completed lot has not yielded an acceptable item. The manufacturer has time to make no more than three production runs. If an acceptable item has not been obtained by the end of the third production run, the cost to the manufacturer in lost sales income and penalty costs will be ₹1,600.

The objective is to determine the policy regarding the lot size (1 + reject allowance) for the required production run(s) that minimizes total expected cost for the manufacturer.

Formulation: A dynamic programming formulation for this problem is

Stage n = production run n (n = 1, 2, 3),

x_n = lot size for stage n,

State sn = number of acceptable items still needed (1 or 0) at beginning of stage n.

Thus, at stage 1, state $s_1 = 1$. If at least one acceptable item is obtained subsequently, the state changes to sn = 0, after which no additional costs need to be incurred.

Because of the stated objective for the problem,

$f_n(s_n, x_n)$ = total expected cost for stages $n, \ldots, 3$ if system starts in state sn at stage n, immediate decision is xn, and optimal decisions are made thereafter,

$f_n^*(s_n) = \min f_n(s_n, x_n),$

$x_n = 0, 1, \ldots$

where $f_n^*(0) = 0$. Using ₹100 as the unit of money, the contribution to cost from stage n is $[K(x_n) + x_n]$ regardless of the next state, where $K(x_n)$ is a function of x_n such that

$$K(x_n) = \begin{cases} 0, & \text{if } x_n = 0 \\ 3, & \text{if } x_n > 0. \end{cases}$$

Therefore, for sn 1,

$$\begin{aligned} f_n(1, x_n) &= K(x_n) + x_n + \left(\frac{1}{2}\right)^{x_n} f_{n+1}^*(1) + \left[1 - \left(\frac{1}{2}\right)^{x_n}\right] f_{n+1}^*(0) \\ &= K(x_n) + x_n + \left(\frac{1}{2}\right)^{x_n} f_{n+1}^*(1) \end{aligned}$$

[where $f_4^*(1)$ is defined to be 16, the terminal cost if no acceptable items have been obtained]. A summary of these basic relationships is given in Fig. 3.8.

NOTES

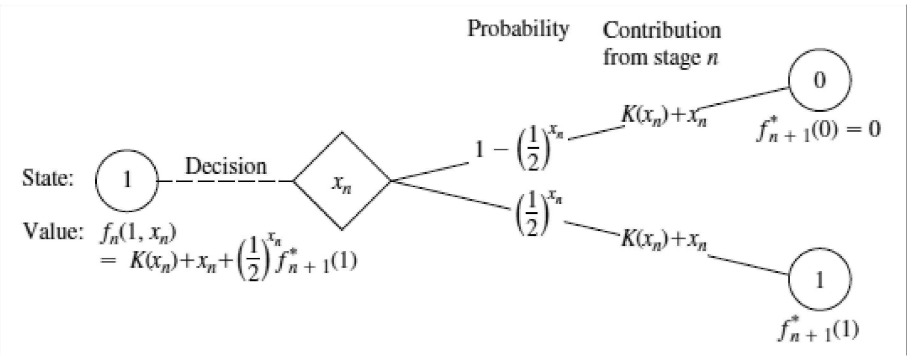

Probability
Contribution from stage n
State:
1
Decision
x_n
Value: $f_n(1, x_n)$
$= K(x_n)+x_n+\left(\frac{1}{2}\right)^{x_n} f^*_{n+1}(1)$
$1-\left(\frac{1}{2}\right)^{x_n}$
$K(x_n)+x_n$
0
$f^*_{n+1}(0)=0$
$\left(\frac{1}{2}\right)^{x_n}$
$K(x_n)+x_n$
1
$f^*_{n+1}(1)$

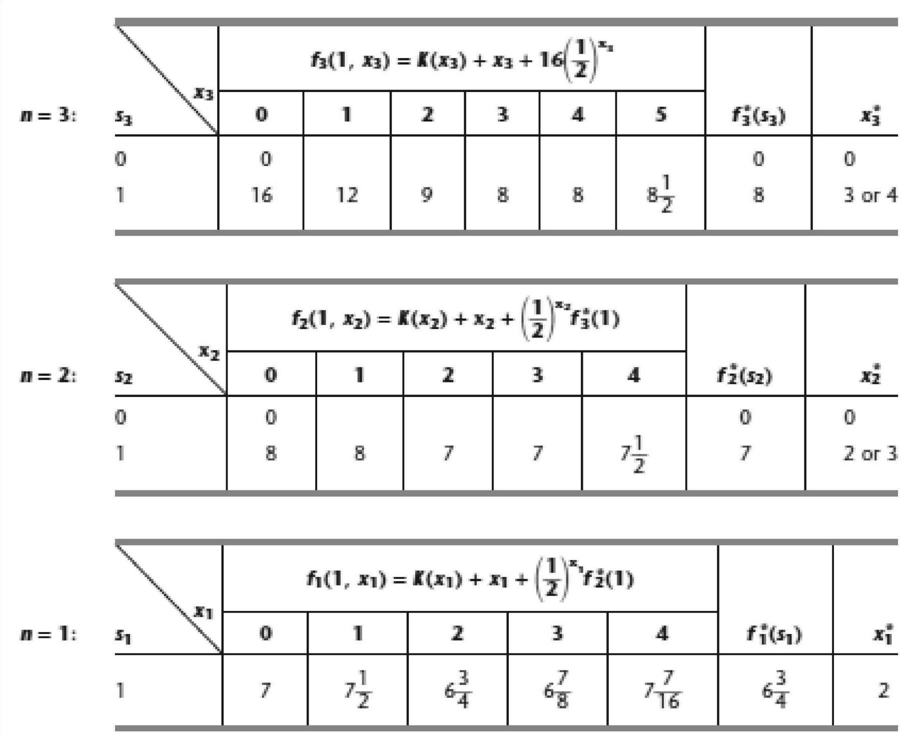
Fig. 3.8 *The Basic Structure for the Hit-and-Miss Manufacturing Co. Problem*

Consequently, the recursive relationship for the dynamic programming calculations is

$$f^*_n(1) = \min_{x_n=0,\,1,\,\ldots} \left\{K(x_n) + x_n + \left(\frac{1}{2}\right)^{x_n} f^*_{n+1}(1)\right\}$$

for $n = 1, 2, 3$.

Solution Procedure: The calculations using this recursive relationship are summarized as follows.



$n = 3$:

	$f_3(1, x_3) = K(x_3) + x_3 + 16\left(\frac{1}{2}\right)^{x_3}$							
s_3 \ x_3	0	1	2	3	4	5	$f^*_3(s_3)$	x^*_3
0	0						0	0
1	16	12	9	8	8	$8\frac{1}{2}$	8	3 or 4

$n = 2$:

	$f_2(1, x_2) = K(x_2) + x_2 + \left(\frac{1}{2}\right)^{x_2} f^*_3(1)$						
s_2 \ x_2	0	1	2	3	4	$f^*_2(s_2)$	x^*_2
0	0					0	0
1	8	8	7	7	$7\frac{1}{2}$	7	2 or 3

$n = 1$:

	$f_1(1, x_1) = K(x_1) + x_1 + \left(\frac{1}{2}\right)^{x_1} f^*_2(1)$						
s_1 \ x_1	0	1	2	3	4	$f^*_1(s_1)$	x^*_1
1	7	$7\frac{1}{2}$	$6\frac{3}{4}$	$6\frac{7}{8}$	$7\frac{7}{16}$	$6\frac{3}{4}$	2

Thus, the optimal policy is to produce two items on the first production run; if none is acceptable, then produce either two or three items on the second production run; if none is acceptable, then produce either three or four items on the third production run. The total expected cost for this policy is ₹675.

Example 3.4: Winning in Las Vegas

An enterprising young statistician believes that she has developed a system for winning a popular Las Vegas game. Her colleagues do not believe that her system works, so they have made a large bet with her that if she starts with three chips, she will not have at least five chips after three plays of the game. Each play of the game involves betting any desired number of available chips and then either winning or losing this number of chips. The statistician believes that her system will give her a probability of $\frac{2}{3}$ of winning a given play of the game.

Assuming the statistician is correct, we now use dynamic programming to determine her optimal policy regarding how many chips to bet (if any) at each of the three plays of the game. The decision at each play should take into account the results of earlier plays. The objective is to maximize the probability of winning her bet with her colleagues.

Formulation: The dynamic programming formulation for this problem is

Stage $n = n$th play of game ($n = 1, 2, 3$),

x_n = number of chips to bet at stage n,

State s_n = number of chips in hand to begin stage n.

This definition of the state is chosen because it provides the needed information about the current situation for making an optimal decision on how many chips to bet next.

Because the objective is to maximize the probability that the statistician will win her bet, the objective function to be maximized at each stage must be the probability of finishing the three plays with at least five chips. (Note that the value of ending with more than five chips is just the same as ending with exactly five, since the bet is won either way.) Therefore,

$f_n(s_n, x_n)$ = probability of finishing three plays with at least five chips, given that the statistician starts stage n in state s_n, makes immediate decision x_n, and makes optimal decisions thereafter,

$$f_n^*(s_n) = \max_{x_n 0,\ 1,\ \ldots,\ s_n} f_n(s_n, x_n).$$

The expression for $f_n(s_n, x_n)$ must reflect the fact that it may still be possible to accumulate five chips eventually even if the statistician should lose the next play. If she loses, the state at the next stage will be $s_n - x_n$, and the probability of finishing with at least five chips will then be $f^*_{n+1}(s_n - x_n)$. If she wins the next play instead, the state will become *sn xn*, and the corresponding probability will be $f^*_{n+1}(s_n = x_n)$. Because the assumed probability of winning a given play is $\frac{2}{3}$, it now follows that

$$f_n(s_n, x_n) = \frac{1}{3} f^*_{n+1}(s_n - x_n) + \frac{2}{3} f^*_{n+1}(s_n + x_n)$$

NOTES

[where $f_4{}^*(s_4)$ is defined to be 0 for $s_4 < 5$ and 1 for $s_4 \geq 5$]. Thus, there is no direct contribution to the objective function from stage n other than the effect of then being in the next state. These basic relationships are summarized in Fig. 3.9. Therefore, the recursive relationship for this problem is

$$f_n^*(s_n) = \max_{x_n=0,\,1,\,\ldots,\,s_n} \left\{\frac{1}{3} f_{n+1}^*(s_n - x_n) + \frac{2}{3} f_{n+1}^*(s_n + x_n)\right\},$$

for n = 1, 2, 3, with $f_4{}^*(s_4)$ as just defined.

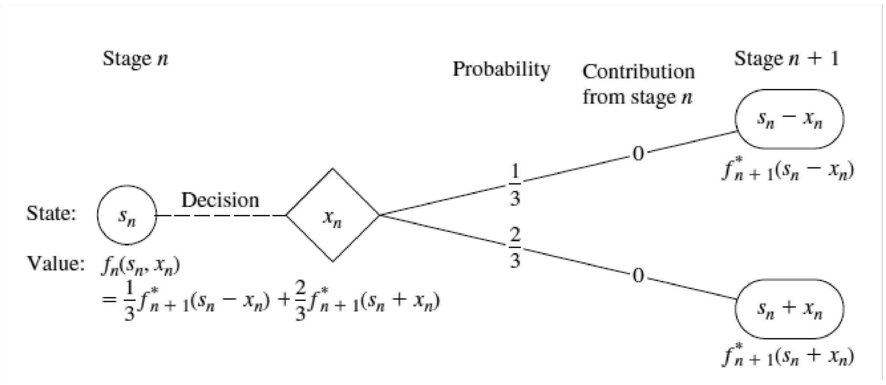
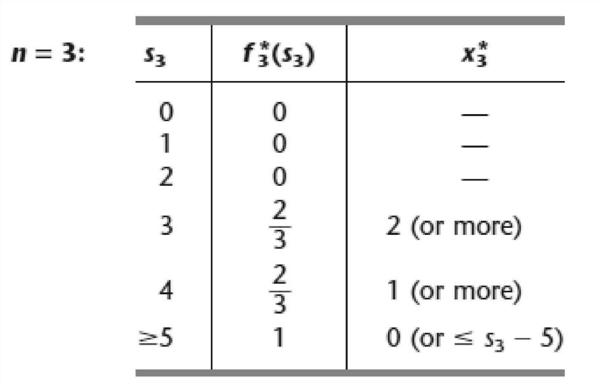


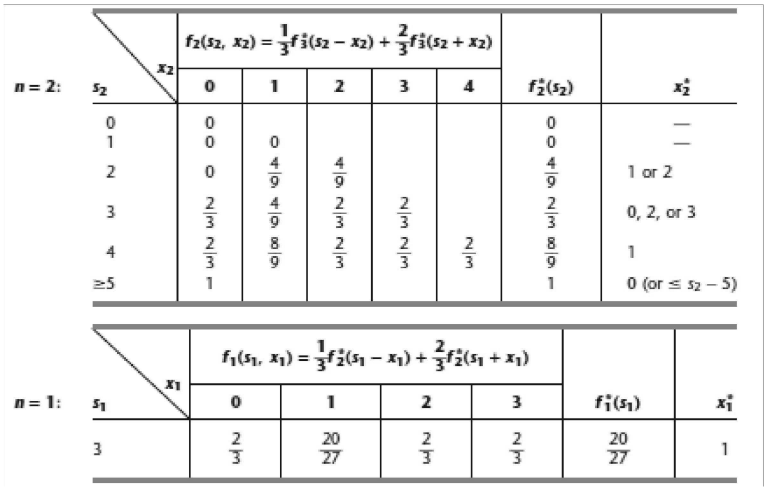
Fig. 3.9 The Basic Structure for the Las Vegas Problem

Solution Procedure: This recursive relationship leads to the following computational results.



$n = 3$:

s_3	$f_3^*(s_3)$	x_3^*
0	0	—
1	0	—
2	0	—
3	$\frac{2}{3}$	2 (or more)
4	$\frac{2}{3}$	1 (or more)
≥5	1	0 (or $\leq s_3 - 5$)



$n = 2$:

	$f_2(s_2, x_2) = \frac{1}{3}f_3^*(s_2 - x_2) + \frac{2}{3}f_3^*(s_2 + x_2)$						
s_2 \ x_2	0	1	2	3	4	$f_2^*(s_2)$	x_2^*
0	0					0	—
1	0	0				0	—
2	0	$\frac{4}{9}$	$\frac{4}{9}$			$\frac{4}{9}$	1 or 2
3	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{2}{3}$	$\frac{2}{3}$		$\frac{2}{3}$	0, 2, or 3
4	$\frac{2}{3}$	$\frac{8}{9}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{8}{9}$	1
≥5	1					1	0 (or $\leq s_2 - 5$)

$n = 1$:

	$f_1(s_1, x_1) = \frac{1}{3}f_2^*(s_1 - x_1) + \frac{2}{3}f_2^*(s_1 + x_1)$					
s_1 \ x_1	0	1	2	3	$f_1^*(s_1)$	x_1^*
3	$\frac{2}{3}$	$\frac{20}{27}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{20}{27}$	1

Therefore, the optimal policy is

$$x_1^* = 1 \begin{cases} \text{if win,} \quad x_2^* = 1 \begin{cases} \text{if win,} & x_3^* = 0 \\ \text{if lose,} & x_3^* = 2 \text{ or } 3. \end{cases} \\ \text{if lose,} \quad x_2^* = 1 \text{ or } 2 \begin{cases} \text{if win,} & x_3^* = \begin{cases} 2 \text{ or } 3 & (\text{for } x_2^* = 1) \\ 1, 2, 3, \text{ or } 4 & (\text{for } x_2^* = 2) \end{cases} \\ \text{if lose,} & \text{bet is lost} \end{cases} \end{cases}$$

This policy gives the statistician a probability of $\frac{20}{27}$ of winning her bet with her colleagues.

3.2.2 Dynamic Programming and the Principle of Optimality

Dynamic programming is the mathematical technique whose development is largely due to Richard E. Bellman. It is applicable to many types of problems where in a series of interrelated decisions (i.e., sequential decisions) are required. There is no single algorithm that can be used to solve all such problems, i.e., a separate algorithm is needed for each type of problem. Thus, dynamic programming is an approach involving the optimisation of multistage decision processes. Essentially, the technique of dynamic programming divides a given problem into stages or subproblems and then solves the subproblems sequentially (usually working backward from the natural end of the problem) until the initial problem is finally solved. The principle behind the operation of this technique is known as the principle of optimality. This principle, set forth by Bellman, states that 'An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.'[9] The principle of optimality is an important concept in context of subsequential decision theory.

Some Important Terms

We may mention the meaning of a few terms which are used when the dynamic programming technique is formalized. These terms are as follows:

(i) *Stage:* It refers to the particular decision we are facing. If we are to take 5 sequential decisions then we have 5 stages. Alternatively a stage can be understood as a subproblem. Thus, in case of five stages we have five subproblems of a given problem.

(ii) *State Variable:* It is a variable defining the current situation at any stage. Suppose we have a pricing problem, then the state variable would be the current price level at a given stage.

(ii) *Optimal Decision Rule:* This rule specifies which decision to make, as a function of state variable and the stage number.

(iv) *Optimal Policy:* An optimal policy is a set of optimal decision rules which guides one's decisions through all stages of the given problem. In fact, this constitutes the solution of the given problem optimizing the desired measure of effectiveness.

NOTES

Salient Features of Dynamic Progamming Approach

In applying the dynamic programming technique, it is necessary to divide the problem into a number of subproblems or decision stages. It is also necessary to describe the stage of the system by a state variable. Then each subproblem, working backward from the natural end of the problem, as stated earlier, should be solved in turn, i.e., a decision must be made at an earlier stage. The decision made at each stage influences the next. In fact the decision made at each stage must take into account its effect not only on the next stage, but also on the entire following sequence of stages. After each subproblem has been solved, the answer is recorded and the payoff (profit, cost, etc., as the case may be) from that stage on to the end of the problem is also recorded. Finally, the optimum overall payoff is ticked and the related decisions at several stages are noted. Such decisions constitute the optimal solution for a given problem. This technique of solving a given problem is often termed as recursive approach. Thus, dynamic programming provides a systematic procedure, whereby, starting with the last stage of the problem and working backward, one makes an optimal decision for each stage. When effectiveness of each stage is optimised in this fashion, the resulting sequence of decisions will given an optimal solution to the problem.

3.2.3 Problems in Dynamic Programming

The basic ideas used in dynamic programming approach or the sequential decision theory can be well understood through the application of this technique to different situations. The same is being done here for general understanding of the fundamentals involved concerning dynamic programming. In particular, we shall study about the following application of dynamic programming.

1. Travelling Salesman's Problem
2. Pricing Problem
3. Production Scheduling Inventory Problem
4. Allocation Problem

Through the following illustrations we explain dynamic programming approach applicable in different situations.

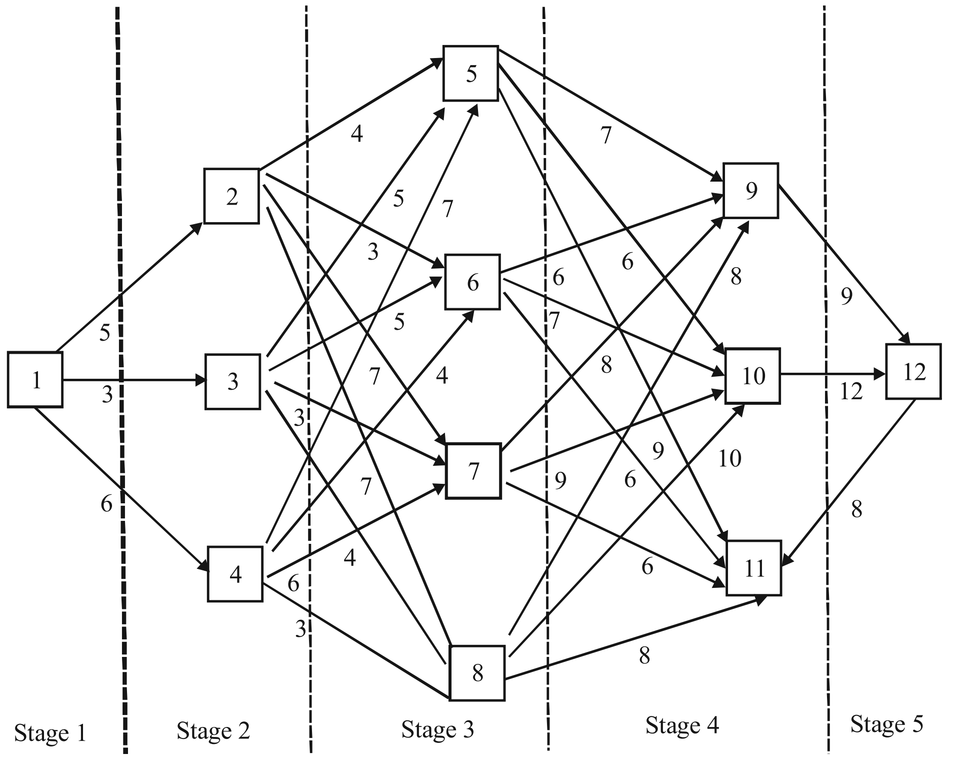
1. Travelling Salesman's Problem

Example 3.5: A manager of a certain company, based in city 12, is in city 1 to attend a series of business seminars. On his return he wants to reach city 12 adopting the least-cost route between city 1 and city 12. The cost from moving between cities is given below. (Only certain cities can be reached directly from a given city. For instance only cities 2, 3, 4 can be reached directly from city 1, cities 5, 6, 7, 8 can be reached directly from city 2, and so on.)

From city				*To city*				*(figures are in hundred ₹)*			
	2	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
1	5	3	6								
2				4	3	7	7				
3				5	5	3	4				
4				7	4	6	3				
5								7	6	9	
6								6	7	6	
7								8	9	6	
8								8	10	8	
9											9
10											12
11											8

What route should be taken by the manager from city 1 to city 12, so that the cost is the least possible?

Solution: The given problem can be represented in the form of a network as follows:



(Figures in squares represent the names of cities 1 to 12 and arrows indicate the route from one city to the other. The cost of the concerning route between the two cities is just written below the corresponding arrow).

The problem is to determine the least cost route from city 1 to city 12. The problem can be split into five stages (although decision-making stages are only 1 to 4) as has been shown in the network. We now solve it applying the dynamic programming approach, working it backward from the natural end of the problem.

Suppose the manager has arrived at stage 4 and further suppose for a moment that he is in city 9. What is the least cost route to city 12? Since there is only one route possible, the minimum cost route is from city 9 to city 12 at a cost of ₹ 900. However, suppose the manager finds himself in city 10 in stage 4, then the minimum cost route is from city 10 to city 12 at a cost of ₹ 1200. In case the manager is in city 11 in stage 4 then the minimum cost route is from city 11 to city 12 at a cost of ₹ 800. So, if the manager arrives at city 9 or 10 or 11 in stage 4 the best policy is to go to city 12 from any one of those destinations, since it is the only route for him then to reach city 12. As such he is in fact not to make any decision when he is in stage 4, i.e., all this is trivial, but a mention of it is necessary in order to establish the full dynamic programming algorithm for the given example. All this can be put in tabular form as follows:

Solution to Stage 4 (or Decision Stage 1, i.e., First Decision)

Leaving	*Going to*	*Total accumulated cost*
City 9	City 12	₹ 900
City 10	City 12	₹ 1200
City 11	City 12	₹ 800

Suppose at stage 3, the manager is in city 5. He can reach city 12 via any of the three cities stated above, viz., 9 or 10 or 11. If he decides to visit city 9, 10 or 11 his total cost will be ₹. 1600, 1700, 1800 respectively. Accordingly, his choice would be to go to city 9 and then to city 12. Similarly, if he is in city 6 in stage 3 he should next stop in city 11 and then to city 12 for a minimum total cost of ₹ 1400. If he is in city 7 then he should stop in city 11 and then should go to city 12 for a minimum total cost of ₹ 1400. However, if he happens to be in city 8 then he should next go to city 11 and then to city 12 for a minimum total cost of ₹ 1600. All this can be put in tabular form as shown in the following table.

Solution to Stage 3 (or Decision Stage 2, i.e., 2nd Decision)

Leaving	*Going to*	*Total accumulated cost*
5	9	₹ 1600
6	11	₹ 1400
7	11	₹ 1400
8	11	₹ 1600

Now suppose at stage 2, the manager is in city 2 from where he can go to city 5 or 6 or 7 or 8 and then to city 12. If he goes to city 12 via city 5 his total costs would be ₹ 20,000 (utilizing the solution reached at stage three as shown in the following table) but if he goes via city 6 his costs would be ₹ 17,000 and these will be respectively ₹ 21,000 and ₹ 23,000 if he decides to go via city 7 and city 8. Accordingly he must go via city 6. Instead the manager may be in city 3 in stage 2 then the total costs would be ₹ 2100 and ₹ 1900 and ₹ 2000 if he decides to go via city 5, 6, 7, and 8, respectively. Obviously he must decided to go via city 7. If he is in city 4 in stage 2 then the total costs would be ₹ 2300, ₹ 1800, ₹ 20,000 and ₹ 1900 if he decides to go via city 5, 6, 7 and 8 respectively. These results can be stated in tabular from as follows:

Solution to Stage 2 (or Decision Stage 3, i.e., 3rd Decision)

Leaving	*Going to*	*Total accumulated cost*
2	6	₹1700
3	7	₹1700
4	6	₹1800

On the basis of the above analysis we can now easily write the solution to stage 1 in tabular form as follows:

Solution to Stage 1 (or Decision Stage 4, i.e., 4th Decision)

Leaving	Going to	Total accumulated cost
1	2	₹2200
1	3	₹2000
1	4	₹2400

The above analysis shows that the manager should adopt the following route to reach city 12:

City 1 → Cify 3 → *City* 7 → *City* 11 → *City* 12

This will be the cheapest possible route with a minimum overall cost of ₹ 20,000. In other words, this is the least cost route from city 1 to city 12.

2. Pricing Problem

Example 3.6: *A* manager is interested in deciding a price policy for a new product to be adopted over the next 4 years. He is considering five different price levels, viz. ₹ 12, ₹ 14, ₹ 16, ₹ 18 and, 20 per unit. After evaluating potential price moves by his competitors, he has worked out the following payoff table which relates his price in a given year to the present value of the profit expected that year:

Payoff Table

Price level		*Year*		
(per unit)	1	2	3	4
₹12	3	9	3	7
₹14	2	1	2	2
₹16	7	4	8	1
₹18	9	2	6	4
₹20	5	5	3	1

Further suppose that the manager wishes to avoid making price changes of more than ₹ 2 from one year to the next, i.e., if the price in any given year is say ₹ 5 per unit then the manager can fix the price in the following year at most at ₹ 7 per unit in the upward direction or at ₹ 3 in the downward direction. Find the optimal price policy the manager should adopt over the next 4 years and compute the maximum expected profitability as well using the given information.

Solution: First of all, we divide the given problem into four subproblems or stages and these subproblems or stages represent the four decisions on which way to move the price level. The following table represents the same.

Payoff Table

	Year			
Price level (per unit)	*1*	*2*	*3*	*4*
₹ 12	3	9	3	7
₹ 14	2	1	2	2
₹ 16	7	4	8	1
₹ 18	9	2	6	4
₹ 20	5	5	3	1
	↑	↑	↑	↑
	Decision 1 or Stage 1	Decision 2 or Stage 2	Decision 3 or Stage 3	Decision 4 or Stage 4

Now applying backward induction, i.e., working backward, we first solve the last subproblem. For instance, the last subproblem is which way to move when you are at one of the price level in the 3rd year. This subproblem may be solved by evaluating each allowable move from each price level in this stage and choosing the move with the highest payoff. This can be done as follows:

Price level (per unit)	*Given pay offs*		*Possible price moves in year 3 and 4*		*Related Total payoffs*
	Year 3	*Year 4*			
₹ 12	3	7	₹ 12	₹ 12	3 + 7 = 10*
			₹ 12	₹ 14	3 + 2 = 5
₹ 14	2	2	₹ 14	₹ 12	2 + 7 = 9*
			₹ 14	₹ 14	2 + 2 = 4
			₹ 14	₹ 16	2 + 1 = 3
₹ 16	8	1	₹ 16	₹ 14	8 + 2 = 10
			₹ 16	₹ 16	8 + 1 = 9
			₹ 16	₹ 18	8 + 4 = 12*
₹ 18	6	4	₹ 18	₹ 16	6 + 1 = 7
			₹ 18	₹ 18	6 + 4 = 10*
			₹ 18	₹ 20	6 + 1 = 7
₹ 20	3	1	₹ 20	₹ 18	3 + 4 = 7*
			₹ 20	₹ 20	3 + 1 = 4

Similarly, we must solve the subproblem when we are at one of the price levels in year 2 and the same has been done as follows:

Price level (per unit)	*Given payoffs*		*Possible price moves in year 2, 3 and 4*			*Related Total payoffs*
	Year 2	*Year 3*				
₹12	9	3	₹12	₹12	₹12	9 + 3 + 7 = 19**
			₹12	₹14	₹12	9 + 2 + 7 = 18
			₹14	₹12	₹12	1 + 3 + 7 = 11
₹14	1	2	₹14	₹14	₹12	1 + 2 + 7 = 10
			₹14	₹16	₹18	1 + 8 + 4 = 13**
			₹16	₹14	₹12	4 + 2 + 7 = 13
₹16	4	8	₹16	₹16	₹18	4 + 8 + 4 = 16**
			₹16	₹18	₹18	4 + 6 + 4 = 14
			₹18	₹16	₹18	2 + 8 + 4 = 14**
₹18	2	6	₹18	₹18	₹18	2 + 6 + 4 = 12
			₹18	₹20	₹18	2 + 3 + 4 = 9
			₹20	₹18	₹18	5 + 6 + 4 = 15**
₹20	5	3	₹20	₹20	₹18	5 + 3 + 4 = 12

Similarly, we must solve the subproblem when we are at one of the price levels in year 1 and the same has been done as follows:

Price level (per unit)	*Given payoffs year*		*Possible price moves in year*				*Related total payoffs*
	1	*2*	*1*	*2*	*3*	*4*	
			₹12	₹12	₹12	₹12	3 + 9 + 3 + 7 = 22*
₹12	3	9	₹12	₹14	₹16	₹18	3 + 1 + 8 + 4 = 16
			₹14	₹12	₹12	₹12	2 + 9 + 3 + 7 = 21*
₹14	2	1	₹14	₹14	₹16	₹18	2 + 1 + 8 + 4 = 15
			₹14	₹16	₹16	₹18	2 + 4 + 8 + 4 = 18
			₹16	₹14	₹16	₹18	7 + 1 + 8 + 4 = 20
₹16	7	4	₹16	₹16	₹16	₹18	7 + 4 + 8 + 4 = 23*
			₹16	₹18	₹16	₹18	7 + 2 + 8 + 4 = 21
			₹18	₹16	₹16	₹18	9 + 4 + 8 + 4 = 25*
₹18	9	2	₹18	₹18	₹16	₹18	9 + 2 + 8 + 4 = 23
			₹18	₹20	₹18	₹18	9 + 5 + 6 + 4 = 24
			₹20	₹18	₹16	₹18	5 + 2 + 8 + 4 = 19
₹20	5	5	₹20	₹20	₹18	₹18	5 + 5 + 6 + 4 = 20*

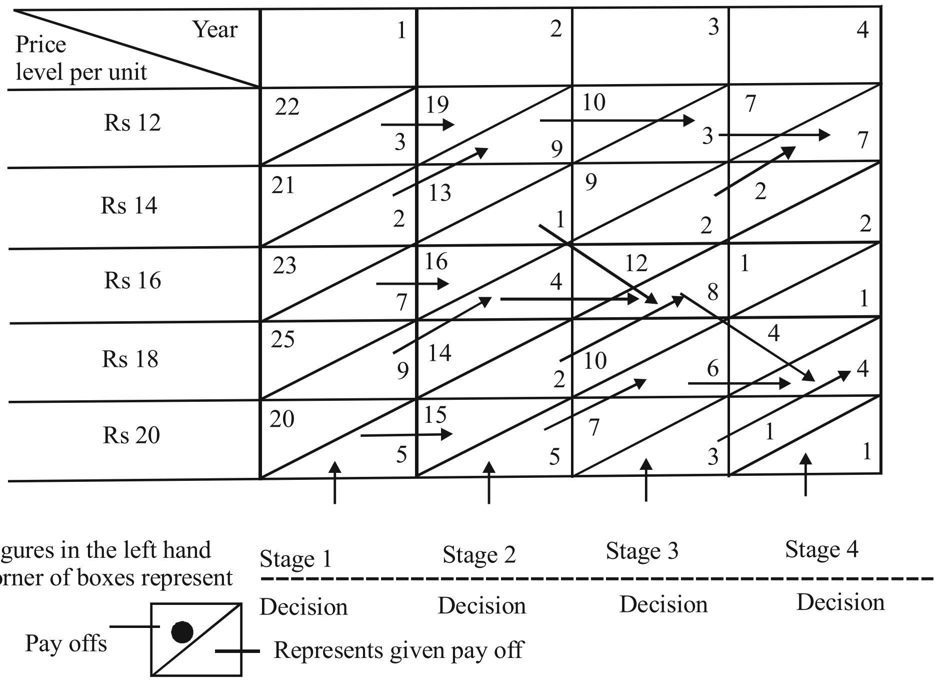
The analysis indicates that the highest possible total payoff is 25 and can be attained by adopting the following price policy by the manager.

NOTES

Year	*Price per unit to be charged*	*Payoff*
1	₹ 18	9
2	₹ 16	4
3	₹ 16	8
4	₹ 18	4
		Total 25 (Maximum possible payoff)

The recently stated optimum results can as well be summarised for the sake of convenience in the form of a payoff (given payoffs as well as accumulated optimum possible payoffs at each stage of decision-making) matrix as follows:

Arrows in the following table represent the sequence of decisions at various stages.



Price level per unit \ Year	1	2	3	4
Rs 12	22 3	19 9	10 3	7 7
Rs 14	21 2	13 1	9 2	2 2
Rs 16	23 7	16 4	12 8	1 1
Rs 18	25 9	14 2	10 6	4 4
Rs 20	20 5	15 5	7 3	1 1

Optimal pricing policy corresponds to sequence of decisions for which the accumulated payoff happens to be maximum possible. In the given case it is 25, corresponding to which pricing decision should be to charge ₹ 18 per unit in year 1, ₹ 16 in year 3 and ₹ 18 in year 4. The various arrows represent this fact in the matrix provided.

3. Production Scheduling—Inventory Problem

Example 3.7: A certain manufacturing plant is to provide 5 units of a certain item in period one and 8 units in period 2. It has no inventory at the beginning of period 1 and should not have any inventory at the end of period 2. The production costs for any period are rupees X^2, where X is the number of units produced during the period. The inventory carrying costs are rupees 5I, where I is the number of units carried in inventory from one period to the next. What production schedule should be followed to minimize total costs? Also work out the minimum possible total costs for solution to the given problem.

NOTES

Solution: To find the solution to the problem we start with period 2. Since the plant wishes to have no inventory at the end of this period, the production amounts can easily be determined as shown in the following table.

Table for period 2 solution

Beginning Inventory (Units)	*Optimal Production (Units)*	*Production cost (₹)*	+	*Inventory costs (₹)*	=	*Total costs (₹)*
0	8	64	+	0		64
1	7	49	+	5	=	54
2	6	36	+	10	=	46
3	5	25	+	15	=	40
4	4	16	+	20	=	36
5	3	9	+	25	=	34
6	2	4	+	30	=	34
7	1	1	+	35	=	36
8	0	0	+	40	=	40

(***Note:*** Since demand in period 2 is of 8 units and as such only enough is produced to meet this need depending on the level of beginning inventory which can be assumed to be anything from 0 to 8 units. Values of beginning inventory larger than 8 need not be considered in view of the fact that we need only 8 units in period two and want no inventory at the end of it.)

We now turn to period 1. Beginning inventory in period 1 is zero and we require 5 units of the item in period one and 8 units in period two (accordingly our requirement for both the periods is of 5 + 8 = 13 units) which can be produced in period 1 and the corresponding solution would be as follows:

Table for period 1 solution

Beginning	*Possible*			*Cost in period*		*Ending*		*Subse-*	*Total*
Inventory	*Production*					*one*		*Inventory*	*quent*
costs									
(*Units*)	(*Units*)	*Production costs*		*Inventory costs*		*Total costs*		(*units*)	*costs*
		(₹)		(₹)		(₹)		(₹)	(₹)
0	5	25	+	0	=	25	0	64	89
	6	36	+	0	=	36	1	54	90
	7	49	+	0	=	49	2	46	95
	8	64	+	0	=	64	3	40	104
	9	81	+	0	=	81	4	36	117
	10	100	+	0	=	100	5	34	134
	11	121	+	0	=	121	6	34	155
	12	144	+	0	=	144	7	36	180
	13	169	+	0	=	169	8	40	209

The above analysis indicates that the minimum possible total costs is ₹ 89 corresponding to which we have the following solution:

Produce 5 units in period 1
Produce 8 units in period 2

There will be zero inventory at the end of period 1 or at the beginning of period 2.

Alternatively, the given problem can be solved as under:

$$\text{Total costs } (TC) = X_1^2 + X_2^2 + 5I$$

where X_1^2 = Number of units produced in period one

X_2^2 = Number of units produced in period two

I = Number of units carried in inventory from one pe riod to the next

Thus, X_1^2 is the production costs of period one

X_1^2 is the production costs of period two

and $5I$ is the inventory carrying cost incurred during period two

$$I = (X_1 - 5) \quad \left\{\begin{array}{l}\text{Since inventory to be carried from one period} \\ \text{to the next = (production in period one) –} \\ \text{(requirement in period one)}\end{array}\right\}$$

and,

$$\begin{aligned}\therefore\ X_2 &= (8 - I) \\ &= [8 - (X_1 - 5)] \\ &= (13 - X_1)\end{aligned}$$

Putting these values in TC equation, we can rewrite it as follows:

$$TC = X_1^2 + (13 - X_1)^2 + 5(X_1 - 5)$$

or $$TC = 2X_1^2 - 21X_1 + 144$$

Since we want to find that value of X_1 for which TC is minimum possible we can work out $\frac{\partial T.C.}{\partial X_1}$ and then equating $\frac{\partial T.C.}{\partial X_1} = 0,$ we can find the desired value of X_1 The same has been done as follows:

$$\therefore \quad T.C = 2X_1^2 - 21X_1 + 144$$

$$\frac{\partial T.C.}{\partial X_1} = 4X_1 - 21$$

and taking $4X_1 - 21 = 0$

we have $X_1 = 5.25$

= 5 units approximately

Thus, we should produce 5 units of the item in period 1 and accordingly we should produce 8 units in period 2. There will thus be no inventory to be carried from one period to the other. Total cost for this solution would be:

$$TC = 2(5)^2 - 21(5) + 144 = 50 - 105 + 144 = ₹\ 89.$$

Alternatively TC can be worked out as follows:

$$\begin{aligned} TC &= X_1^2 + X_2^2 + 5I \\ &= (5)^2 + (8)^2 + 5(0) \\ &= 25 + 64 = ₹\ 89. \end{aligned}$$

4. Allocation Problem

Example 3.8: A sales manager of a big commercial organization has secured eight tickets for an all-star cricket match. He will keep one for himself and give the remaining seven away to various people four four different companies.

Number of		*Companies*		
Tickets	*A*	*B*	*C*	*D*
0	0	0	0	0
1	5	4	7	2
2	9	7	10	6
3	11	9	12	8
4	13	11	14	10
5	15	14	16	11
6	16	16	18	12
7	17	16	18	12

Suppose the above numbers represent the additional truckloads of orders he would receive over the coming year from a particular allocation of tickets. How many tickets should he distribute within each company so as to maximize his total additional orders? (Solve the problem applying dynamic programming approach.)

Solution: For applying dynamic programming approach to the given problem, let us assume that the four companies are just the four stages of this problem. Let us name company *A* as stage 1, *B* as stage 2, *C* as stage 3 and *D* as stage 4. The state of the problem at any given point is given by the number of tickets still to be allocated to a company.

We shall start with stage 4. The stage 4 solution is just a trivial one. It consists of the allocation of tickets in company *D* and will be stated as tabulated in the Table for Solution to Stage 4.

Table for Solution to Stage 4

Numbers of tickets to be allocated at stage i = Ti T_4	*Optimal numbers of tickets to be allocated to company D represented as*T_n	*Aggregate additional truckloads of orders: represented as*AA_0
0	0	0
1	1	2
2	2	6
3	3	8
4	4	io
5	5	11
6	6	12
7	6,7	12

It may be noted that there is a tie for T_D when $T_4 = 7$ since increasing the number of tickets at company *D* from 6 to 7 has no effect on the aggregate additional orders.

Now we move to stage 3 wherein Company C enters the solution. The objective now is to obtain an optimal allocation of tickets between companies *C* and *D*. The solution to stage 3 will be as shown in the following table.

NOTES

Table for Solution to Stage 3

No. of tickets to be allocated at stage 3	*Optimal numbers of tickets to be allocated* to company		*Aggregate additional truckloads* of orders
T_3	T_D	T_C	AA_0
0	0	0	0
1	0	1	7
2	0	2	10
3	2	1	13
4	2	2	16
5	2,3	3,2	18
6	3	3	20
7	3,4	4,3	22

The solution to stage 3 indicates that if there is only one ticket to be allocated, it should be allocated to company *C;* if there are four tickets to be allocated, then they should be allocated like this: 2 to company *D* and 2 to company *C* for then only the aggregate additional truckloads of orders would be the maximum possible. One can read all other allocations in the table in similar manner. If there are five tickets, then either, 2 to *D* and 3 to C or 3 to *D* and 2 to C may be allocated and the aggregate additional truckloads of orders, viz., 18 will be the same in both these cases, which will be maximum possible in case only 5 tickets are there for allocation. A similar type of situation can be noticed when 7 tickets are to be allocated.

We now turn to stage 2 wherein we shall think of allocating the tickets to company *B* on the one hand and companies *C* and *D* taken together on the other hand in accordance with their respective payoffs. (The payoff in case of company *B* is given in the question itself corresponding to different numbers of tickets to be allocated and the payoff for companies *C* and *D* will be those as worked out in the preceding table giving the solution to stage 3). If we want to optimize the allocation at this stage, the solution will be as follows:

Table for Solution to Stage 2

No. of tickets to be allocated at stage 2	*Optimal numbers of tickets to be allocated to company*		*Aggregate additional truckloads of orders*
T_2	$T_D + T_C$	T_M	AA_0
0	0	0	0
1	1	0	7
2	1	1	11
3	1,2	2,1	14
4	2	2	17
5	3	2	20
6	4	2	23
7	4,5	3,2	25

The above solution can be read in the same way as has been explained in case of solution to Stage 3. Finally, we can turn to stage 1 where in we shall have to decide the allocation of tickets between company *A* on the one hand and companies *B, C* and *D* taken together on the other in accordance with their respective payoffs for *B, C, D* as worked out in the preceding table and for *A* as given in the question itself. The solution will be as follows:

Table for Solution to Stage 1

No. of tickets to be allocated at stage 1	*Optimal numbers of tickets to be allocated to company*		*Aggregate additional truckloads of orders*
T_1	$T_D + T_C + T_B$	T_A	AA_0
0	0	0	0
1	1	0	7
2	1	1	12
3	1,2	2,1	16
4	2	2	20
5	3	2	23
6	4	2	26
7	5	2	29

The above analysis states that out of 7 tickets to be allocated 2 should be allocated to company *A* as per table giving solution to stage 1. Of the remaining 5 tickets 2 should be allocated to company *B* as per table giving solution to stage 2. Of the remaining three tickets, 1 should be allocated to company *C* and 2 should be allocated to company *D* as per table giving solution to stage 3. Thus, the optimal policy will be to allocate 2 tickets to company *A, 2* to company *B,* 1 to company *C* and 2 to company *D* corresponding to which aggregate additional truckloads of orders would be 9 + 7 + 7 + 6 = 29 which will be the maximum possible number of additional orders that the manager can obtain by adopting the said policy in respect of allocation of tickets among the four companies.

3.2.4 Dynamic Programming Under Uncertainty

All the illustrations presented so far are related to apply dynamic programming technique under certainty situations. It is also possible to bring problems where uncertainty exists, under the net of this technique. Example 3.5 will illustrate the use of dynamic programming under uncertainty situation.

Example 3.9: A manufacturing concern must purchase the raw material it requires within five weeks. If it purchases the raw material at the end of a week, then whatever price prevails in that week has to be paid for it. If it chooses to wait, then it must purchase the raw material in some future week, at whatever price prevails when it decides to purchase the raw material. If it has not purchased before week 5 then it will be forced to purchase the same at the end of week five, since the raw material is needed for production. The price of the raw material varies each week according to the probabilities in the following table.

Price	*Probability*
₹ 400	0.4
₹ 500	0.3
₹ 600	0.3

NOTES

Work out a purchasing policy which will minimize the expected cost of the said raw material.

Solution: This problem may be solved by applying dynamic programming approach as follows:

Let n = Week number (or the stage number), $n = 1, 2 \ldots 5$ in the given case

X_n = Price observed in week n (or the state variable)

$f_n(X,)$ = Minimum expected cost if price observed in week n is X_n and an optimal policy is adopted from week n to the natural end of the problem (or the return function)

Working backward, we consider that last week, viz., week 5 first of all for which

$$f_5\ (X_5) = X_5$$

Since the manufacturing concern must purchase the raw material in week, 5 if it has not purchased the same earlier.

Turning to week 4 we may state the following:

$$f_4 X_4 = \text{Min}[\overset{\text{Act}}{X_4}; \overset{\text{Wait}}{(400)\ (0.4) + (500)(0.3) + (600)(0.3)}]$$

That is, the manufacturing concern may either act and obtain the price X_4 or wait and obtain the expected value of the price in the last week. Let us rewrite the above equation concerning $f_4(X_4)$ in terms of (X_5), where

$$f_5(X_5) = X_5$$

$$f_4(X_4) = \text{Min}\ [\overset{\text{Act}}{X_4}; \overset{\text{Wait}}{f_5(400)\ (0.4) + f_5(500)(0.3) + f_5(600)(0.3)}]$$

Similarly, the situation in week 3 can be stated as follows:

$$f_3(X) = \text{Min}\ [\overset{\text{Act}}{X_3}; \overset{\text{Wait}}{f_4(400)\ (0.4) + f_4(500)(0.3) + f_4(600)(0.3)}]$$

Similarly, the situation for any other week can be stated in a general form as follows:

$$f(X) = \text{Min}\ [\overset{\text{Act}}{\text{X}_n}; \overset{\text{Wait}}{f_{n+1}(400)\ (0.4) + f_{n+1}(500)(0.3) + f_{n+1}(600)(0.3)}]$$

$$n = 1, 2, \ldots 4$$

These equations may be solved recursively starting with equation $f(X_5) = X_5$ and working backwards. *At week* 5 the material must be purchased if the same has not been purchased earlier. *At week* 4, equation concerning $f_4(X_4)$ stated above holds and to perform the minimisation involved in it we set the values of the two alternatives viz 'Act' or 'Wait' equal to each other and solve for the break-even value of X_4, represented as X_{4b} and it works out to:

$$X_{4b} = (400)(0.4) + (500)\ (0.3) + (600)\ (0.3)$$
$$= 160 + 150 + 80 = 490$$

If the price in week 4 is below ₹ 490, the purchase should be made immediately otherwise the purchase should be delayed. It is necessary to record the minimum expected cost at week 4:

$$f_4(X_4) = \begin{cases} 400 \text{ if } X_4 = 400 \\ 490 \text{ otherwise} \end{cases}$$

This states that the minimum expected cost as of week 4 is ₹ 400 if price is ₹ 400 or ₹ 490 if the price is either ₹ 500 or ₹ 600. The value of ₹ 490 represents the expected price in week 5.

NOTES

Now turning to week 3 we may say that

Act Wait

$$f_3(X_3) = \text{Min}\ [X_3;\ f_4(400)\ (0.4) + f_4(500)(0.3) + f_4(600)(0.3)]$$

Substituting the result of $f_4(X_4) = \begin{cases} 400 \text{ if } X_4 = 400 \\ 490 \text{ otherwise} \end{cases}$

worked out above in it we have,

Act Wait

$$f_3(X_3) = \text{Min}\ [X_3\ (400)\ (0.4) + (490)(0.3) + (490)\ (0.3)]$$

Again we find the break even value for X represented as X as follows:

$$X_{3t} = (400)\ (0.4) + (490)\ (0.3) + (490)\ (0.3)$$
$$= 160 + 147 + 147 = 454$$

If the price in week 3 is below ₹ 454 the purchase should immediately be made, but if the price is above ₹ 454, the purchase should be delayed. The minimum expected cost at week three is recorded as follows:

$$f_3(X_3) = \begin{cases} 400 \text{ if } X_3 = 400 \\ 490 \text{ otherwise} \end{cases}$$

Act Wait

$$f_2(X_2) = \text{Min}\ [X_2;\ (400)\ (0.4) + (454)(0.3) + (454)\ (0.3)]$$

Thus, $X_u = (400)\ (0.4) + (454)(0.3) + (454)(0.3)$

$$= 160 + 1362 + 136.2$$
$$= 43240$$

and accordingly we have,

$$f_2(X_2) = \begin{cases} 400 \text{ if } X_2 = 400 \\ 432.40 \text{ otherwise} \end{cases}$$

[432.40 otherwise turning to week 1 we may say that

Act Wait

$$f_2(X_2) = \text{Min}\ [X_2;\ (400)\ (0.4) + (432.40)(0.3) + (432.40)\ (0.3)]$$

Thus, $X_2 = (400)\ (0.4) + (432.40)(0.3) + (432.40)(0.3)$

$$= 160 + 129.72 + 129.72$$
$$= 419.44$$

and accordingly, we have,

$$f_1(X_1) = \begin{cases} 400 \text{ if } X_1 = 400 \\ 419.54 \text{ otherwise} \end{cases}$$

NOTES

The optimal policy for the manufacturing concern may be summarized as follows:

If the price in weeks, 1, 2, 3 or 4 is ₹ 400 then the firm must make the purchase immediately otherwise it must wait. If no purchase has been made by week 5, the purchase must be made at the price prevailing in week 5. If this policy is adopted the minimum expected cost is ₹ 400 if X equals ₹ 400; and ₹ 419.54 if X is greater than ₹ 400. Prior to knowing X, the minimum expected cost is:

$$= 400\ (0.4) + (419.54)(0.6)$$
$$= 160 + 251.724 = ₹\ 411.72$$

Check Your Progress

1. Define the principle of optimality.
2. What are the dynamic programming techniques?
3. Name the problems in dynamic programming.

3.3 GAME THEORY

Competition is a watchword of modern life. We say that a competitive situation exists if two or more individuals are making decisions in situation that involves conflicting interests and in which the outcome is controlled by the decisions of all parties concerned. We assme that in a competitive situation, each participant acts in a rational manner and tries to resolve the conflict of interests in his favour. It is in this context that game theory has developed. Professor John von Neumann and Oscar Morgenstern published their book entitled '***The Theory of Games and Economic Behaviour***' wherein they provided a new approach to many problems involving conflict situations—an approach now widely used in Economics, Business Administration, Sociology, Psychology and Political Science as well as in Military Training. Fundamentally, the theory of games attempts to provide an answer to the question: What may be considered a rational course of action for an individual confronted with a situation whose outcome depends not only upon his own actions but also upon the actions of others, Who in turn, are faced with a similar problem of choosing a rational course of action? In fact, the theory of games is simply the logic of rational decisions.

The term 'Game' represents a conflict between two or more parties. Game theory is really the 'Science of Conflict'. It is not concerned with finding an optimum or winning strategy for a particular conflict situation but it provides general rules concerning the logic that underlies strategic behaviour of all types.

Game theory applies to those competitive situations that are technically known as 'Competitive Games' or simply 'Games'. Situations, in order to be termed games, must possess the following properties:

(i) The number of competitors is finite.

(ii) There is a conflict of interests between the participants.

(iii) Each of the participants has available to him a finite list of possible courses of action, i.e., several choices of appropriate actions; this list being not necessarily the same for each competitor.

(iv) The rules governing these choices are specified and known to all the players; a play of the game results when each of the players chooses a single course of action from the list of courses available to him.

(v) The outcome of the game is affected by the choices made by all the players; the choices are made simultaneously so that no competitor knows his opponent's choice until he is already committed to his own.

(vi) The outcome for all the specific sets of choices by all the players is known in advance and numerically defined. The outcome of a play consists of the particular set of courses of action undertaken by the competitors. Each outcome determines a set of payments (+ve, –ve or zero), one to each competitor.

Illustration of a Game

When a competitive situation meets all the above stated criteria, we can call it a game. This can be made clear by an example of a simple game. Suppose there are two opponents *X* and *Y.* We can think of them as sitting across a table from each other and each with two buttons in front of him. We shall denote player *X*'*s* buttons , *m* and *n* and player *Y*'*s* buttons as *r* and *t;* thus each player has two choices open to him. We also presume a partition between them so that neither can see in advance which button his opponent is going to press. At a signal from a third party, each player presses one of his buttons. The results of each of the possible four combinations is known in advance to both of the players; the uncertainty inherent in the game arises from the fact that neither player knows what button his opponent will press next. Every time the third party signals, each player presses one of his buttons and the game thus continues. At the end of, say, a hundred 'Plays', the game is over and the points won by each of the players are totalled and the winner is determined. It is assumed that both players are of equal intelligence and that each actively attempts to win the game.

This sort of simple game can be illustrated in tabular form as follows:

A Simple Game in Tabular Form

		Player Y	
		Button *r*	Button *t*
Player *X*	Button *m*	*X* wins 2 points	*X* wins 3 points
	Button *n*	*Y* wins 3 points	*Y* wins 1 point

In the above table, the plays open to each of the opponents and the resulting gains or losses (called payoffs) have been shown. This game is biased against Player *Y* because if Player *X* presses button *m* for each play of the game, Player *Y* cannot win; in fact, Player *Y* faces a choice between losing 2 points on each play if he responds by pressing Button *r* or losing 3 points on each play if he responds by pressing Button *t*. Player *Y* will respond each time by pressing Button *r* since this represents his least loss alternative. Thus, Player *X* will win 2 points on each play of the game. We can similarly present a game biased against Player *X* wherein *X* has no chances winning and at best an minimize his losses like Player *Y* in the above stated case.

But all simple games cannot be said to have such simple solutions. This can be illustrated by the following example of a game:

A Simple Game in Tabular Form

		Player *Y*	
		Button *r*	Button *t*
Player *X*	Button *m*	*Y* wins 4 points	*X* wins 3 points
	Button *n*	*X* wins 1 point	*Y* wins 2 points

In the above case, it is not as simple as it was in the earlier example to determine the individual player's strategies. We can see from the game that Player *X* would not press his button *m*, hoping to win 3 points on each play simply because Player *Y* could counter with button *r* and win 4 points himself. Similarly, Player *Y* would not press button hoping to win 4 points on every play because Player *X* could counter with his button *n* and win 1 point himself. In such a situation, therefore, it is advantageous for the players to play each of their choices (buttons) a part of the time only. How to calculate the proportion of time to allot to each choice shall be discussed a little later.

Standard Conventions in Game Theory

In order to eliminate the necessity for written descriptions of the 'Payoffs' (as we have shown in the above two examples), a standard set of conventions has been established in game theory. It is the usual practice to omit a description like 'Player *X* wins two points' and replace it with integer 2. The positive algebraic sign which is assumed to accompany this number indicates that it is Player *X* who benefits from this payoff. Similarly, instead of saying 'Player *Y* wins three points' one simply indicates this with the value –3, the minus sign indicating that it is Player *Y* who benefits from this particular payoff.

Another standard convention that is usually followed is that Player *X* has choices between the rows and Player *Y* has choices between the columns.

Keeping these two conventions in view we can write the above stated two illustrations of games as follows:

Illustration 1

	Player *Y*	
Player *X*	2	3
	–3	–1

Illustration 2

	Player *Y*	
Player *X*	–4	3
	1	–2

A further convention in game theory is known as matrix notation. In this case, games are represented in the form of a matrix (a rectangular array of numbers deriving from matrix algebra). When games are expressed in this fashion, the resulting matrix is commonly known as a *payoff matrix*. The above stated two illustrations can be put in the form of payoff matrices as follows:

Illustration 1

Player Y

$$\text{Player } X \begin{bmatrix} 2 & 3 \\ -3 & -1 \end{bmatrix}$$

Illustration 2

Player Y

$$\text{Player } X \begin{bmatrix} -4 & 3 \\ 1 & -2 \end{bmatrix}$$

The term *strategy* is often talked about in game theory. It refers to the total pattern of choices employed by any player. It may be defined as a complete set of plans of action specifying precisely what the player will do under every possible future contingency that might occur during the play of the game. For example, if Player X chooses to play his first row half of the time and his second row half of the time, his strategy for the game is 1/2, 1/2. Thus, the strategy of a player is the decision rule he uses for making the choice from his list of courses of action. The strategy could be a pure or mixed one. When a player plays one row all the time (or one column all the time) he is said to be adopting a pure strategy. In a mixed strategy, Player X will play each of his rows a certain part of the time and Player Y will play each of his columns a certain part of the time. In business, a close analogy is when, for example, a manager follows a certain course of action A until an alternate course of action B appears to be more profitable. Later on, should Action A appear more attractive again, the manager switches back to it.

3.3.1 Two-Person Zero-Sum Games

In game theory, a zero-sum game is a mathematical representation of a situation in which each participant's gain or loss of utility is exactly balanced by the losses or gains of the utility of the other participants. Alternatively, we can say that the zero-sum games are the games in which one player's win is the other player's loss. If the total gains of the participants are added up and the total losses are subtracted, then they will sum to zero. Thus, for example, cutting a cake, where taking a larger piece reduces the amount of cake available for others as much as it increases the amount available for that taker, is a zero-sum game if all participants value each unit of cake equally.

In contrast, non-zero-sum describes a situation in which the interacting parties' aggregate gains and losses can be less than or more than zero. A zero-sum game is also called a strictly competitive game while non-zero-sum games can be either competitive or non-competitive. Zero-sum games are most often solved with the 'Minimax Theorem' which is closely related to linear programming duality or with Nash equilibrium. Many mathematicians have a reasoning prejudice towards seeing situations as zero-sum, known as zero-sum bias.

Definition: A two player game is called a **zero-sum game** if the sum of the payoffs to each player is constant for all possible outcomes of the game. More specifically, the terms or coordinates in each payoff vector must add up to the same value for each payoff vector. Such games are sometimes called **constant-sum games** as an alternative.

NOTES

The zero-sum property (if one gains, another loses) means that any result of a zero-sum situation is Pareto optimal. Generally, any game where all strategies are Pareto optimal is called a conflict game. Following is the example of generic zero-sum game:

	Choice 1	Choice 2
Choice 1	−A, A	B, −B
Choice 2	C, −C	−D, D

Zero-sum games are a specific example of constant-sum games where the sum of each outcome is always zero. Such games are distributive, not integrative; the pie cannot be enlarged by good negotiation.

Situations where participants can all gain or suffer together are referred to as non-zero-sum. Other non-zero-sum games are games in which the sum of gains and losses by the players are sometimes more or less than what they began with.

The idea of Pareto optimal payoff in a zero-sum game gives rise to a generalized relative selfish rationality standard, the punishing-the-opponent standard, where both players always seek to minimize the opponent's payoff at a favourable cost to himself rather to prefer more than less. The punishing-the-opponent standard can be used in both zero-sum games, for example warfare game and chess, and non-zero-sum games, for example pooling selection games.

Basic Concepts of Two-Person Zero-Sum Games

Following are the basic and significant concepts of simple two-person zero-sum games:

- A two-person game is characterized by the strategies of each player and the payoff matrix.
- The payoff matrix shows the gain (positive or negative) for player 1 that would result from each combination of strategies for the two players. Remember that the matrix for player 2 is the negative of the matrix for player 1 in a zero-sum game.
- The entries in the payoff matrix can be in any units as long as they represent the *utility* (or value) to the player.
- There are two key assumptions about the behaviour of the players. The first is that both players are *rational*. The second is that both players are *materialistic* meaning that they choose their strategies in their own interest.

3.3.2 Sum Games

In game theory, the concept 'Value of a Game' is considered very important. It refers to the average payoff per play of the game over an extended period of time. This can be explained by an example. Suppose the two games are as follows:

$$\begin{array}{cc} & \text{Player } Y \\ \text{Player } X & \begin{bmatrix} 3 & 4 \\ -6 & 2 \end{bmatrix} \end{array} \qquad \begin{array}{cc} & \text{Player } Y \\ \text{Player } X & \begin{bmatrix} -7 & 2 \\ -3 & -1 \end{bmatrix} \end{array}$$

Game One (With a Positive Value) — Game Two (With a Negative Value)

In game one, Player *X* would play his first row on each play of the game and *Y* would respond by playing his first column each time in order to minimize his losses. Since Player *X* wins three points on each play of the game, his average winnings per play will also be three as long as the game is played. The value of the game is thus three with an implicit positive algebraic sign which denotes that Player *X* wins the game. But in game two, Player *Y* plays his first column on each play of the game and Player *X* responds by playing his second row on each play of the game in order to minimize losses. As a result, *Y* wins 3 points on each play of the game. Since the average payoff per play is –3, the value of this game is –3, the minus sign indicating that *Y* is the winner.

But determining the value of the game is not always as simple as in the case of these two examples. In case the players determine that their best alternative is to play each row or each column a certain part of the time, calculating the value of the game becomes a bit more complex. We shall take them up a little later in this unit.

Now we shall see the process of determination of optimum strategies and the value of a game with the help of some illustrations.

Example 3.10: Determine the optimum strategies for the two players *X* and *Y* and find the value of the game from the following payoff matrix:

$$\begin{array}{cc} & \text{Player } Y \\ \text{Player } X & \begin{bmatrix} 3 & -1 & 4 & 2 \\ -1 & -3 & -7 & 0 \\ 4 & -6 & 2 & -9 \end{bmatrix} \end{array}$$

Solution: For determining the optimum strategies, the cautious approach is to assume the worst and act accordingly. If Player *X* plays with first row strategy, then Player *Y* will play with second column the win one point, otherwise he will lose 3, 4 and 2 if he plays with column, 13 and 4 respectively. If Player *X* plays with second row strategy, then the worst would happen to him only when Player *Y* plays with the third column because in that case *Y* would win 7 points. If Player *X* plays with third row strategy, then the worst he can expect is losing 9 points when *Y* plays with the fourth column. In this problem then, Player *X* should adopt first row strategy because only then his loss will be minimum. Thus, Player X can make the best of the situation by aiming at the highest of these minimal payoffs. This decision rule is known as 'Maximin Strategy'.

Looking from the perspective of Player *Y*, we can say that if *Y* plays with column first strategy the maximum he can lose is 4 points if Player *X* adopts the strategy of row three. If Player *Y* plays with column two strategy, there is no question of any loss whatever may be the strategy of Player *X*. In such a case player *X* will adopt the strategy of row one, for only then his loss will be minimum. If Player *Y* plays with column third strategy, the maximum he can lose is 4 points if *X* adopts the strategy of row one. If *Y* adopts the strategy of column four, he can lose at the most 2 points if *X* adopts the strategy of row one. In this problem then, *Y* should adopt the second column strategy and thereby ensure a victory of 1 point which is the maximum in the given case. Thus, Player *Y* can make the best of the situation by aiming at the lowest of these maximum payoffs (viz., 4, –1, 4, 2). Thus, he should seek the minimum among the maximum payoffs. This decision rule is known as 'Minimax Strategy'.

Thus, Player *Y* will play his second column on each play and Player *X* will respond by playing his first row on each play. In this way, *Y* will win 1 point and *X* will lose 1 point in each play. Hence, this is a two-person zero-sum game with a pure strategy. Since *Y* will win 1 point and X will lose 1 point in each play, the value of the game is –1. This payoff –1 is then a saddle point in the given game and can be marked (encircled) as under:

$$\begin{array}{cc} & \text{Player } Y \\ \text{Player } X & \begin{bmatrix} 3 & \enclose{circle}{-1} & 4 & 2 \\ -1 & -3 & -7 & 0 \\ 4 & -6 & 2 & -9 \end{bmatrix} \end{array}$$

3.3.3 Games With Mixed Strategies

In game theory, the strategy of a player in a game is a complete plan of action for any situation that may occur. This determines the complete behaviour of player and the player's strategy determines the action that the player will take at any stage of the game.

A strategy profile is also sometimes termed as strategy combination. It is a set of strategies for each player that specifies all actions in a game. It must include one and only one strategy for every player. Sometimes the strategy concept is by mistake confused with that of a move. A move is an action taken by a player at some point during the play of a game whereas a strategy is a complete algorithm for playing the game because it guides a player what to do for every possible situation throughout the game. A player's strategy set describes what strategies are available for playing the game. Strategies are of two types, pure and mixed. A pure strategy provides a complete definition of how a player will play a game. A mixed strategy is an assignment of a probability to each pure strategy. This allows for a player to randomly select a pure strategy. Since probabilities are continuous, there are infinitely many mixed strategies available to a player, even if their strategy set is finite. Certainly, a pure strategy can be considered as a degenerate case of a mixed strategy in which that specific pure strategy is selected with probability 1 and every other strategy with probability 0.

A totally mixed strategy is a mixed strategy in which the player assigns a strictly positive probability to every pure strategy. The totally mixed strategies are important for equilibrium refinement.

Consider the payoff matrix table of pure coordination game (Refer Table 3.4). Here one player chooses the row and the other chooses a column. The row player receives the first payoff, the column player the second. If row opts to play A with probability 1, i.e., play A for sure then the player is said to be playing a pure strategy. If column opts to flip a coin and play A if the coin lands heads and B if the coin lands tails then the player is said to be playing a mixed strategy and not a pure strategy.

Table 3.4 *Pure Coordination Game*

	A	*B*
A	1 , 1	0 , 0
B	0 , 0	1 , 1

Example 3.11: Find the optimum strategies and the value of the game from the following payoff matrix concerning two-person game:

$$\begin{array}{cc} & \text{Player } Y \\ \text{Player } X & \begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix} \end{array}$$

Solution: In the given game, there is no saddle point because there is no one value which is smallest value in its row and largest in its column. Therefore the players will resort to what is known as mixed strategy, i.e., player X will play each of his rows a certain portion of time and player Y will play each of his columns a certain part of the time. The question then is to determine what proportion of the time a player should spend on his respective rows and columns. This can be done by the use of algebraic method stated as follows.

Let Q equal the proportion of time player X spends playing the first row, then 1–Q must equal the time he spends playing his second row (because one equals the time available for play). Similarly, suppose player Y spends time R in playing first column and 1–R proportion of time he spends playing the second column. All this can be stated as under:

$$\begin{array}{ccc} & & \text{Player } Y \\ & & R \quad 1-R \\ \text{Player } X & \begin{array}{c} Q \\ 1-Q \end{array} & \begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix} \end{array}$$

Now, we must find out the values of Q and R. Let us analyse the situation from X's view point. He would like to devise a strategy that will maximize his winning (or minimize his losses) irrespective of what his opponent Y does. For this X would like to divide his play between his rows in such a manner that his expected winnings or losses when Y plays the first column will equal his expected winnings or losses when Y plays the second column. Expected winnings indicate the sum, overtime, of the payoffs multiplied by the probabilities that these payoffs will obtain.

NOTES

This can be calculated as shown below:

X's *EXPECTED WINNINGS*

	When Y Plays Column One	*When Y Plays Column Two*
X Plays Row One: *Q* of the Time	(1) (*Q*)	(4) (*Q*)
X Plays Row Two: *1–Q* of the Time.	5(1 – *Q*)	3(1 – *Q*)
X's Total Expected	*Q* + 5(1 – *Q*) Winnings	4*Q* + 3(1 – *Q*)

Equating the expected winnings of *X* when *Y* plays column one with when *Y* plays column two we can find the value of *Q* as follows:

$$
\begin{aligned}
& Q + 5(1 - Q) &&= 4Q + 3(1 - Q)\\
\text{or}\quad & Q + 5 - 5Q &&= 4Q + 3 - 3Q\\
\text{or}\quad & -5Q &&= -2\\
\text{or}\quad & Q &&= 2/5\\
\therefore\quad & 1 - Q &&= 3/5
\end{aligned}
$$

This means that player *X* should play his first row 2/5 of the time and his second row 3/5 of the time if he wants to maximize his expected winnings from the game.

On the similar basis expected losses of *Y* can be worked out as under:

Y's *EXPECTED LOSSES*

	Y Plays Column One R of the Time	*Y Plays Column Two 1—R of the Time*	*Y's Total Expected Losses*
When *X* Plays Row One	1(*R*)	4(1 – *R*)	*R* + 4(1– *R*)
When *X* Plays Row Two	5(*R*)	3(1 – *R*)	5*R* + 3(1 – *R*)

Equating the expected losses of *Y* when *X* plays row one with when *X* plays row two, we can find the value of *R* as follows:

$$
\begin{aligned}
& R + 4(1 - R) &&= 5R + 3(1 - R)\\
& R + 4 - 4\,R &&= 5R + 3 - 3R\\
\text{or}\quad & -5R &&= -1\\
\text{or}\quad & R &&= 1/5\\
\therefore\quad & 1 - R &&= 4/5
\end{aligned}
$$

This means that player *Y* should play his first column 1/5 of the time and his second column 4/5 of the time if he wants to minimize his expected losses in the game.

Now we can illustrate the original game with the appropriate strategies for each of the player as follows:

$$
\begin{array}{ccc}
 & & \textit{Player Y} \\
 & & \begin{array}{cc} 1/5 & 4/5 \end{array} \\
\textit{Player X} & \begin{array}{c} 2/5 \\ 3/5 \end{array} & \begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix}
\end{array}
$$

NOTES

Alternative Method or Short Cut Method for Finding the Above Strategies

Original game

$$\begin{array}{cc} & Y \\ X & \begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix} \end{array}$$

Step 1: Subtract the smaller payoff in each row from the larger one and the smaller payoff in each column from the larger one.

$$\begin{array}{cccc} & Y & & \\ X & \begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix} & \begin{array}{c} 3 \\ 2 \end{array} & \begin{array}{c} (\text{i.e., } 4-1=3) \\ (\text{i.e., } 5-3=2\ ) \end{array} \\ & \begin{array}{cc} 4 & 1 \\ (\text{i.e. } 5-1=4) & (\text{i.e. } 4-3=1) \end{array} & & \end{array}$$

Step 2: Interchange each of these pairs of subtracted numbers found in Step 1 above.

$$\begin{array}{ccc} & Y & \\ X & \begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix} & \begin{array}{c} 2 \\ 3 \end{array} \\ & \begin{array}{cc} 1 & 4 \end{array} & \end{array}$$

Step 3: Put each of the interchanged numbers over the sum of the pair of numbers.

$$\begin{array}{ccc} & Y & \\ X & \begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix} & \begin{array}{c} 2/(2+3) \\ 3/(2+3) \end{array} \\ & \begin{array}{cc} 1/(1+4) & 4/(1+4) \end{array} & \end{array}$$

Step 4: Simplify the fraction to obtain the proper proportions or the required strategies.

$$\begin{array}{ccc} & Y & \\ X & \begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix} & \begin{array}{c} 2/5 \\ 3/5 \end{array} \\ & \begin{array}{cc} 1/5 & 4/5 \end{array} & \end{array}$$

Now we determine the value of the game. Looking at the game from X's point of view we can argue as follows:

(*i*) During the 1/5 of the time Y plays column one, X wins 1 point 2/5 of the time (when X plays row one) and 5 points 3/5 of the time (when X plays row second).

(*ii*) During the 4/5 of the time Y plays column two, X wins 4 points 2/5 of the time (when X plays row one) and 3 points 3/5 of the time (when X plays row second).

Thus total expected winnings of player X are the sum of the above two statements as under:

$$\frac{1}{5}\left\{(1)\left(\frac{2}{5}\right)+(5)\left(\frac{3}{5}\right)\right\}+\frac{4}{5}\left\{(4)\left(\frac{2}{5}\right)+(3)\left(\frac{3}{5}\right)\right\}$$

$$= \left(\frac{1}{5}\right)\left(\frac{17}{5}\right) + \left(\frac{4}{5}\right)\left(\frac{17}{5}\right)$$

$$= \frac{17}{25} + \frac{68}{25}$$

$$= \frac{85}{25}$$

$$= \frac{17}{5}$$

Thus the value of the game is 17/5 which means that player *X* can expect to win an average payoff of 17/5 points for each play of the game if he adopts the strategy we have determined as stated above. If the value of the game determined above had a negative sign, it would simply signify that *Y* was the winner. The same result we can also get by looking at the game from *Y's* point of view doing similar calculations.

Pure and Mixed Strategies with Saddle Point

The saddle point in a payoff matrix is the one which is the smallest value in its row and the largest value in its column. The saddle point is also known as the equilibrium point in the theory of games. An element of a matrix that is simultaneously the minimum of the row in which it occurs and the maximum of the column in which it occurs is a saddle point of the matrix game. In a game having a saddle point, the optimum strategy for Player *X* is always to play the row containing a saddle point and for Player *Y* to play the column that contains a saddle point. The saddle point also gives the value of such a game. The saddle point in the payoff matrix of a game may or may not exist. If there is a saddle point, we can easily find out the optimum strategies and the value of the game by what is known as solution by saddle point without having to do too many calculations. But when the saddle point is not there, then we have to use algebraic methods for working out the solutions of the game problems.

Game Problems of Mixed Strategy

Example 3.12: Find the optimum strategies and the value of the game from the following payoff matrix concerning a two-person game:

Player *Y*

$$\text{Player } X \begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix}$$

Solution: In the given game, there is no saddle point because there is no one value which is smallest in its row and largest in its column. Therefore, the players have to resort to mixed strategy, i.e., Player *X* will play each of his rows a certain part of time and Player *Y* will play each of his columns a certain part of the time. The question then is to determine the proportion of time a player should spend on his respective rows and columns. This can be done by the use of the following algebraic method:

Let Q equal the proportion of time Player X spends playing the first row, then $1 - Q$ must equal the time he spends playing his second row (because one equals the time available for play). Similarly, suppose Player Y spends time R in playing the first column and $1 - R$ be the time he spends on playing the second column. All this can be stated as follows:

$$\text{Player } Y$$

$$\text{Player } X \quad \begin{array}{c} Q \\ 1-Q \end{array} \begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix}$$

Now, we must find the values of Q and R. Let us analyse the situation from X's view point. He would like to devise a strategy that would maximize his winning (or minimize his losses) irrespective of what his opponent Y does. For this, X would like to divide his play between his rows in such a manner that his expected winnings or losses, when Y plays the first column equal to his expected winnings or losses when Y plays the second column. (Expected winnings indicate the sum, overtime, of the payoffs that will be obtained multiplied by the probabilities that these payoffs will obtain). In our case, we can calculate the same as follows:

X's Expected Winnings

	When Y plays *Column one*	*When Y plays* *Column two*
X plays row one: Q of the time	$(1)(Q)$	$(4)(Q)$
X plays row two: $1 - Q$ of the time	$5(1- Q)$	$3(1 - Q)$
X's total expected winnings	$Q + 5(1 - Q)$	$4Q + 3(1 - Q)$

Equating the expected winnings of X when Y plays column one with when Y plays column two, we can find the value of Q as follows:

$$\begin{aligned} Q + 5(1 - Q) &= 4Q + 3(1 - Q) \\ Q + 5 - 5Q &= 4Q + 3 - 3Q \\ -5Q &= -2 \\ \text{Or} \qquad Q &= -2/5 \\ 1 - Q &= 3/5 \end{aligned}$$

This means that Player X should play his first row 2/5 of the time and his second row 3/5 of the time if he wants to maximize his expected winnings from the game.

Similarly, the losses of Y can be worked out as follows:

Y's Expected Losses

Equating the expected losses of Y when X plays row one with when X plays row two, we can find the value of R as follows:

$$\begin{aligned} R + 4(1 - R) &= 5R + 3(1 - R) \\ R + 4 - 4R &= 5R + 3 - 3R \end{aligned}$$

$$\text{Or} \qquad -5R = -1$$

$$\text{Or} \qquad R = 1/5$$

$$1 - R = 4/5$$

This means that Player Y should play his first column 1/5 of the time and his second column 4/5 of the time if he wants to minimize his expected losses in the game.

Now we can illustrate the original game with the appropriate strategies for each of the player as follows:

$$\text{Player } Y$$

$$\text{Player } X \begin{array}{c} {}^{2/5} \\ {}^{3/5} \end{array}\begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix}$$

An Alternative Method (or the Short-Cut Method) for finding the above strategies is as follows:

Original game

$$Y$$

$$X\begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix}$$

Step 1 Subtract the smaller payoff in each row from the larger one and the smaller payoff in each column from the larger one.

$$\begin{array}{l} \quad Y \\ X\begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix}\begin{array}{l} 3 \\ 2 \end{array} \end{array} \qquad \begin{array}{l} \text{(i.e., } 4 - 1 = 3) \\ \text{(i.e., } 5 - 3 = 2) \end{array}$$

$$\text{(i.e., } 5 - 1 = 4) \text{ (i.e., } 4 - 3 = 1)$$

Step 2 Interchange each of these pairs of subtracted numbers found in Step 1 above.

$$Y$$

$$X\begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix}\begin{array}{l} 2 \\ 3 \end{array}$$

$$1 \quad 4$$

Step 3 Put each of the interchanged numbers over the sum of the pair of the numbers.

$$Y$$

$$X\begin{bmatrix} 1 & 4 \\ 5 & 3 \end{bmatrix}\begin{array}{l} 2/(2+3) \\ 3/(2+3) \end{array}$$

$$1/(1+4) \quad 4/(1+4)$$

Step 4 Simplify the fraction to obtain the proper proportions or the required strategies.

$$\begin{array}{ccc} & Y & \\ X & \begin{bmatrix}1 & 4\\ 5 & 3\end{bmatrix} & \begin{matrix}2/5\\ 3/5\end{matrix} \\ & \begin{matrix}1/5 & 4/5\end{matrix} & \end{array}$$

NOTES

Now we determine the value of the game. Looking at the game from *X's* point of view we can argue that:

(i) During the 1/5 of the time that *Y* plays column one, *X* wins 1 point 2/5 of the time (when *X* plays row one) and 5 points 3/5 of the time (when *X* plays row second).

(ii) During the 4/5 of the time that *Y* plays column two, *X* wins 4 points 2/5 of the time (when *X* plays row one) and 3 points 3/5 of the time (when *X* plays row second).

Thus, the total expected winnings of player *X* are the sum of the above two statements, that is,

$$\frac{1}{5}\left\{(1)\left(\frac{2}{5}\right)+(5)\left(\frac{3}{5}\right)\right\}+\frac{4}{5}\left\{(4)\left(\frac{2}{5}\right)+(3)\left(\frac{3}{5}\right)\right\}$$

$$= \left(\frac{1}{5}\right)\left(\frac{17}{5}\right)+\left(\frac{4}{5}\right)\left(\frac{17}{5}\right)$$

$$= \frac{17}{25}+\frac{68}{25}$$

$$= \frac{85}{25}$$

$$= \frac{17}{5}$$

Thus, the value of the game is 17/5 which means that Player *X* can expect to win an average payoff of 17/5 points for each play of the game if he adopts the strategy we have determined above. If the value of the game determined above had a negative sign, it would simply signify that *Y* was the winner. The same result can also be achieved by looking at the game from *Y*'s point of view and carrying out similar calculations.

First Alternative Method (or Short-cut Method) for Determining the Value of a Game

Under this, we calculate:

X's expectations when *Y* plays column one,

X's expectations when *Y* plays column two,

Y's expectations when *X* plays row one

Y's expectations when *X* plays row 2

In all these four cases the answer remains the same. Hence, any one of these calculations is sufficient to determine the value of the game. The logic behind this approach is the same as that of determining the optimum strategies. (In the case of Player *X*, we determined a strategy that guaranteed *X* the same winnings, irrespective of his opponent's choice of columns). This can be illustrated as follows:

NOTES

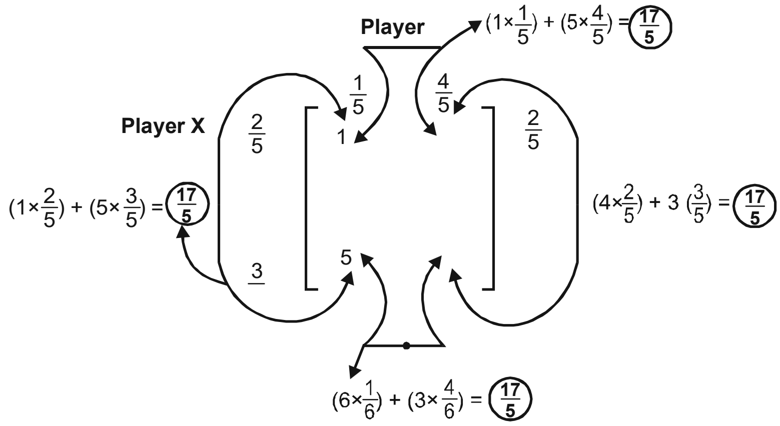


Fig. 3.10 Diagrammatic Form of Determining the Value of a Game

Second Alternative Method for Determining the Value of a Game Using the Probabilities of Each Payoff

In a simple 2 × 2 game without a saddle point, each player's strategy consists of two probabilities denoting the portion of the time he spends on each of his rows or columns. Since each player plays a random pattern, we can list the probabilities of each payoff (in our given question) as follows:

Payoff	Strategies which Produce this Payoff	Probability of this Payoff
1	Row 1 column 1	$\frac{2}{5}\times\frac{1}{5}=\frac{2}{25}$
4	Row 1 column 2	$\frac{2}{5}\times\frac{4}{5}=\frac{8}{25}$
5	Row 2 column 1	$\frac{3}{5}\times\frac{1}{5}=\frac{3}{25}$
3	Row 2 column 2	$\frac{3}{5}\times\frac{4}{5}=\frac{12}{25}$
		Sum = 1.0

The value of the game can be found out by taking the sum of the products of each of these payoffs and their respective probabilities.

For the given problem, this can be worked out as follows:

Payoff	Probability of the Payoff	Product of the first two Columns
1	$\frac{2}{25}$	$\frac{2}{25}$
4	$\frac{8}{25}$	$\frac{32}{25}$
5	$\frac{3}{25}$	$\frac{15}{25}$
3	$\frac{12}{25}$	$\frac{36}{25}$
		Total = $\frac{85}{25}$ $= \frac{17}{5}$ (Value of the Game)

NOTES

Mixed Strategy Problems by Arithmetic Method

By mixed strategy is meant a situation in which the course of action is selected with some fixed probability. As such, there is a probabilistic situation. In such a strategy, the objective of the players is to maximize the expected gain or minimize the expected loss by choosing among various pure strategies with fixed probabilities.

In mathematical terms, a mixed strategy for a player having two or more courses of action can be thought as a set S of n probabilities (whose sum is unity). Here, n is the number of pure strategies by the player. Let p_j be the probability of selecting strategy j, where $j = 1, 2, 3, \ldots\ldots, n$.

Then, $S = \{p_{1,} p_{2,} \ldots\ldots, p_n\}$; $p_1 + p_2 + \ldots\ldots + p_n = 1$ and $\forall j\ p_j \geq 0$.

There are cases when a pure strategy for a game may not exist. Hence, no saddle point exists. In such cases, both the players choose an optimal mixture of strategies to find an equilibrium point. The optimal mixed strategy may be determined for each player in this case by assigning the probability of it being chosen to each strategy. This is known as mixed strategy since this is a probabilistic combination of the available choices of strategy.

Value of the Game

When the mixed strategy is obtained, it has the least payoff that Player A can expect to gain and Player B can expect to lose. The expected payoff with arbitrary payoff matrix $[a_{ij}]$ of order $m \times n$ is given by $E(p, q) = \Sigma\Sigma\, p_i\, a_{ij}\, q_j =$ $\text{P}^{\text{T}}\,\text{A Q}$, where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$; $\text{P} = (p_{1,} p_{2, \ldots\ldots,} p_m)$ and Q $= (p_{1,} p_{2, \ldots\ldots,} p_n)$

Arithmetic Method of Solving Mixed Strategy Problems

This method is also known as the short-cut method. This is a simple method in which the optimal strategy is found for each player in a payoff matrix of order 2 × 2 with no saddle point. The followings are steps to be adopted in this method:

NOTES

1. Calculate the difference between the two values of the first row, neglecting the negative sign, and put it against the second row.
2. Calculate the difference between the two values of the second row, neglecting the negative sign, and put it against the first row.
3. Repeat the above two steps for the two columns also.

The values obtained by swapping the differences, as stated above, are the optimal relative frequencies for play for the strategies of both the players. These are then converted into probabilities by dividing each by their sum.

Note: This method cannot be used to solve a 2 × 2 game having a saddle point.

Example 3.13: The following is the payoff matrix of two competitor companies in terms of their advertising plan.

Company A	**Company B**	
	Normal Advertisement B_1	Special Advertisement B_2
Normal Advertisement A_1	12	15
Special Advertisement A_2	14	10

Suggest the optimal strategies for the two companies and the net outcome.

Solution: The payoff matrix has no saddle point and so mixed strategies and arithmetic methods can be used.

The solution reveals that Company *A* should adopt strategy A_1, 57% of the time and A_2, 43% of the time. In the same way, Company *B* should adopt strategy B_1, for 71% of the time and B_2, 29% of the time.

We can now calculate the expected gain for Company *A* as follows:

(i) 12 × 4/7 + 14 × 3/7 = 90/7 and Company *B* adopts B_1

(ii) 15 × 4/7 + 10 × 3/7 = 90/7 and Company *B* adopts B_2

We proceed as follows:

Company A	**Company B**			
	B_1	B_2		
A_1	12	15	14 – 10 = 4	$p(A_1) = 4/(4 + 3) = 4/7$
A_2	14	10	15 – 12 = 3	$p(A_2) = 3/(4 + 3) = 3/7$
	15 – 10 = 5	14 – 12 = 2		
	$p(B_1) = 5/(5 + 2) = 5/7$	$p(B_2) = 2/(5 + 2) = 2/7$		

The expected loss for Company *B* is,

(i) 12 × 5/7 + 15 × 2/7 = 90/7 and Company *A* adopts A_1

(ii) 14 × 5/7 + 10 × 2/7 = 90/7 and Company *A* adopts A_2

3.3.4 Graphical Solution

NOTES

The graphic method can only be used in games with no saddle point and having payoff $m \times n$ matrices where either m or n is two. The graphic method enables us to substitute a much simpler 2×2 matrix for the original $m \times 2$ or $2 \times n$ matrix.

We will apply the graphic short-cut by plotting on 2 different vertical axis, the 2 payoffs corresponding to each of the 5 columns. The payoff numbers in the first row are plotted on axis 1 and those in the second row on axis 2, which should be drawn at some distance away from the first axis but should be parallel to the first axis as shown in the following figure:

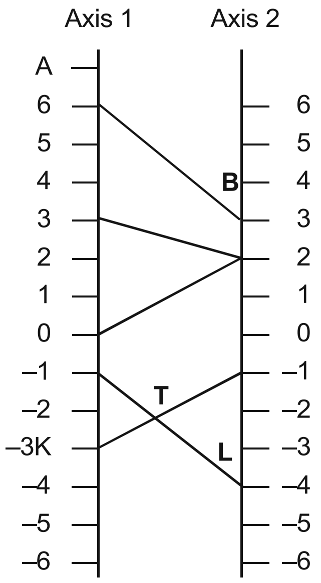


***Fig. 3.11** Graphic Method*

Thus, the 2 payoff numbers 6 and 3 in the first column are denoted respectively by point *A* on axis 1 and point *B* on axis 2. Line *AB* then denotes *Y*'s move of the first column. By plotting the payoff numbers of each of the remaining 4 columns on the 2 axes, we obtain five lines like the line *AB*, which correspond to the given 5 moves of *Y.*

If using a thick line we draw the segments which bound the figure from the bottom, namely, the segments AT and *LT* and mark the highest point (7) on this boundary, the two lines passing through it identify the two critical moves of *Y*, which combined with the two of *X,* yield the following 2×2 matrix:

$$X\begin{bmatrix} -1 & -3 \\ -4 & -1 \end{bmatrix}$$

The optimal strategies now can be determined in the way explained above.

Example 3.14: Determine the optimum strategies and the value of the game from the following payoff matrix concerning a 2 person 4×2 game.

$$\begin{array}{cc} & Y \\ X & \begin{bmatrix} -6 & -2 \\ -3 & -4 \\ 2 & -9 \\ -7 & -1 \end{bmatrix} \end{array}$$

Solution: There is no saddle point in the given game. This game also cannot be reduced by dominance because there is no row which is always preferred by *X* irrespective of what *Y* might do.

Player *X* can actually think of this 4 × 2 game as six sub-games, each of size 2 × 2 (because he can choose not to play any two of his rows if he so desires). These six sub-games can be listed as follows:

$$\begin{array}{cc} & Y \\ & \begin{array}{cc} 2/5 & 3/5 \end{array} \\ X \begin{array}{c} 1/5 \\ 4/5 \end{array} & \begin{bmatrix} -6 & -2 \\ -3 & -4 \end{bmatrix} \end{array} \leftarrow \textit{Sub game No.}\,(i)$$

Strategies have been noted in this matrix payoff.

The value of the game is $\left\{-\frac{6}{5}+\left(-\frac{12}{5}\right)\right\} = -\frac{18}{5} = -3.60$

$$\begin{array}{cc} & Y \\ & \begin{array}{cc} 7/15 & 8/15 \end{array} \\ X \begin{array}{c} 11/15 \\ 4/15 \end{array} & \begin{bmatrix} -6 & -2 \\ 2 & -9 \end{bmatrix} \end{array} \leftarrow \textit{Sub game No.}\,(ii)$$

Strategies have been noted in this matrix payoff.

The value of the game is $\left(\frac{11}{15}\times -6\right)+\left(\frac{4}{15}\times 2\right)=\left(-\frac{66}{15}\right)+\left(\frac{8}{15}\right)=-\frac{58}{15}=-3.87$

$$\begin{array}{cc} & Y \\ X & \begin{bmatrix} -6 & -2 \\ -7 & -1 \end{bmatrix} \end{array} \leftarrow \textit{Sub game No.}\,(iii)$$

There is a saddle point here. Hence, the value of this sub-game is –6.

$$\begin{array}{c} Y \\ \begin{bmatrix} -3 & -4 \\ 2 & -9 \end{bmatrix} \end{array} \leftarrow \textit{Sub game No.}\,(iv)$$

There is saddle point here. Hence value to this sub-game is –4.

$$\begin{array}{cccc} & & \multicolumn{2}{c}{Y} \\ & & \frac{3}{7} & \frac{4}{7} \\ X & \frac{6}{7} & -3 & -4 \\ & \frac{1}{7} & -7 & -1 \end{array} \leftarrow Sub\ game\ No.(v)$$

Strategies have been noted in the matrix payoff.

The value of the game is $\left(-\frac{18}{7}\right)+\left(-\frac{7}{7}\right)$

$= -\frac{25}{7} = -3.57$

$$\begin{array}{cccc} & & \multicolumn{2}{c}{Y} \\ & & \frac{8}{7} & \frac{9}{7} \\ X & \frac{6}{17} & 2 & -9 \\ & \frac{11}{17} & -7 & -1 \end{array} \leftarrow Sub\ game\ No.(vi)$$

Strategies have been noted in the matrix payoff.

The value of the game is,

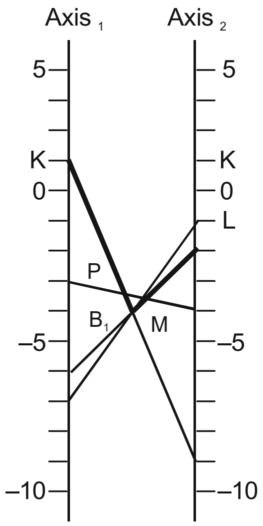
$\left(\frac{12}{17}\right)+\left(-\frac{77}{17}\right) = -\frac{65}{17} = -3.82$

If we look at the values of all these sub-games, we find that the values being negative, *Y* will win and *X* will lose in all the cases. Player *X*, who is making the choice, will naturally prefer the sub-game with the smallest negative value and this is sub-game No. (v) in our example. Thus, *X* will play a two-row mixed strategy between the second and fourth rows of the original game and he will expect to lose an average of 357 points per play of the game, which will be his minimum possible loss. Player *Y* will also adopt a strategy consisting of the same number of columns. In brief, sub-game No.(v)'s strategies will be adopted by players *X* and *Y* with a game value equal to –3.57 in the case of the given example.

Solution through Graphic Method

The above example can also be easily worked out with the help of the graphic method as follows.

The case of payoff matrices having only two columns but more than two rows is similar, except that in the diagram, we thicken the line segments which bound the figure from the top and take the lowest point on this boundary. The following diagram shows the payoff numbers from each row represented as points on two vertical axes, 1 and 2.



Thus, line B joins the first payoff number –6 and –2 of the first row. Similarly, other lines have been drawn representing payoff numbers in the 2nd, 3rd and 4th rows. The segments *KP, PM* and *ML* drawn in thick lines bound the figure from the top and their lowest intersection *M,* through which two lines pass, defining the following 2 × 2 matrix relevant for our purpose:

$$\begin{array}{cc} & Y \\ X & \begin{bmatrix} -3 & -4 \\ -1 & -1 \end{bmatrix} \end{array}$$

The optimal strategies can now be determined as usual.

3.3.5 Solution by Linear Programming

Consider the general LPP.

Max $Z = C_1\, x_1 + C_2 + x_2 + ... + C_n\, x_n$

Subject to the constraints,

$$a_{i1}\, x_1 + a_{i2}\, x_2 + ... + a_{in}\, x_n \leq b_i \text{ where, } i = 1, 2 \ ... \ m$$

and, $x_j \geq 0$ where $j = 1, 2 \ ... \ n$.

The problem can be formulated as a dynamic programming problem as follows.

Let the general LPP be considered as a multistage problem with each activity *j* ($j = 1, 2 ... m$) as an individual stage. Then this is an *n* stage problem and the decision variables are the level of activities x_j (≥ 0) at stage *j*. As x_j is continuous, each activity has an infinite number of alternatives within the feasible region.

We know that allocation problems are a particular type of LPP. These problems require the allocation of available resources to the activities. Each constraint represents the limitation of different resources and $b_1, b_2 \ ... \ b_m$ are the amounts of available resources. Since there are *m* resources, states must be represented by an *m*-dimensional vector, given by $(\beta_1, \beta_2 \ ... \ \beta_m)$.

Let $f_n\,(\beta_1, \beta_2 ... \beta_m)$ be the maximum value of the general LPP defined for stages $x_1, x_2 \ ... \ x_n$ for states

$(b_1, b_2 \ ... \ b_m)$ using forward recursive equation, that is,

f_j (β$_1$ β$_2$ ... β$_m$) = max ($C_j\, x_j + f_{j-1}$(β$_1$ − $a_{1j}\, x_2$, β$_2$ − $a_{2j}\, x_j$ ... β$_m$ − $a_{mj}\, x_j$)

$$0 \le x_j \le \beta$$

NOTES

The maximum value of β that x_j can assume is, β = $\text{Min}\left\{\frac{b_1}{a_{1j}}, \frac{b_2}{a_{2j}} \ldots \frac{b_m}{a_{mj}}\right\}$ because the minimum value satisfies the set of constraints simultaneously.

Example 3.15: Use dynamic programming to solve the following LPP,

Max $Z = 3x_1 + 5x_2$

Subject to the constraints, $x_1 \le 4$

$$x_2 \le 6,$$
$$3x_1 + 2x_2 \le 18$$
$$x_1, x_2 \ge 0$$

Solution: The problem consists of three resources and two decision variables. Hence the problem has two stages and three state variables.

Let B_{1j}, B_{2j}, B_{3j} be the state of the system at stage j and f_j (B_{1j}, B_{2j}, B_{3j}) be the optimal (maximum) value of the objective function for state j = 1, 2 given the state (B_{1j}, B_{2j}, B_{3j}). Using backward computation procedure, we have

$$f_2\,(B_{12}, B_{22}, B_{23}) = \text{Max}\,[5x_2]$$
$$0 \le x_2 \le P_{22}$$
$$0 \le 2x_2 \le P_{32}$$
$$= 5\ \text{Max}\,(x_2)$$
$$0 \le x_2 \le \beta\ _{22}$$
$$0 \le x_2 \le \frac{\beta_{32}}{2}$$

Since Max x_2, which satisfies $0 \le x_2 \le \beta_{22}$,

$$0 \le x_2 \le \frac{\beta_{32}}{2} \text{ is the minimum of } \left(\beta_{22}, \frac{\beta_{32}}{2}\right)$$

i.e., $$\text{Max}\,(x_2) = x_2{}^* = \text{Min}\left(\beta_{22}, \frac{\beta_{32}}{2}\right) \qquad (1)$$

∴ $$f_2\,(\beta_{12}, \beta_{22}, \beta_{32}) = 5\ \text{Min}\left(\beta_{22}, \frac{\beta_{32}}{2}\right) \qquad (2)$$

Also,

f_1 (β$_{11}$, β$_{21}$, β$_{31}$) = Max [$3x_1 + f_2\ (\beta_{11} - x_1,\ \beta_{21} - 0,\ \beta_{31} - 3x_1)$]

$$0 \le x_1 \le \beta_{11}$$
$$0 \le 3x_1 \le \beta_{31}$$

From f_2 (β$_{11}$ − x_1, β$_{21}$ − 0, β$_{31}$ − $3x_1$) = 5 Min $\left(\beta_{21}, \frac{\beta_{31} - 3x_1}{2}\right)$

NOTES

$$\therefore \quad f_1(\beta_{11}, \beta_{21}, \beta_{31}) = \text{Max}\left[3x_1 + 5\,\text{Min}\left(\beta_{21}, \frac{\beta_{31} - 3x_1}{2}\right)\right]$$

$$0 \le x_1 \le \beta_{11}$$

$$0 \le 3x_1 \le$$

Since it is a two-stage problem, at the first stage

$$\beta_{11} = 4, \beta_{21} = 6, \beta_{31} = 18$$

$$\therefore \quad f_1(\beta_{11}, \beta_{21}, \beta_{31}) = \text{Max}\left\{3x_1 + 5\,\text{Min}\left(6, \frac{18 - 3x_1}{2}\right)\right\}$$

$$0 \le x_1 \le 4$$

$$0 \le x_1 \le 6$$

$$= \text{Max}\left\{3x_1 + 5\,\text{Min}\left(6, \frac{18 - 3x_1}{2}\right)\right\}$$

$$0 \le x_1 \le 4$$

Now, $$\text{Min}\left(6, \frac{18-3x_1}{2}\right) = \begin{Bmatrix} 6 & \text{if } 0 \le x_1 \le 2 \\ \dfrac{18-3x_1}{2} & \text{if } 2 \le x_1 \le 4 \end{Bmatrix} \qquad (3)$$

$$0 \le x_1 \le 4$$

From (3)

$$f_1(\beta_{11}, \beta_{21}, \beta_{31}) = \text{Max}\begin{Bmatrix} 3x_1 + 5(6) & \text{if } 0 \le x_1 \le 2 \\ 3x_1 + 5\left(\dfrac{18-3x_1}{2}\right) & \text{if } 2 \le x_1 \le 4 \end{Bmatrix}$$

Since, Max of $3x_1 + 30$, $0 \le x_1 \le 2$ occurs at $x_1 = 2$ and

Max of $\frac{90-9x_1}{2}$, $2 \le x_1 \le 4$ also occurs at $x_1 = 2$

$$\therefore \quad f_1(\beta_{11}, \beta_{21}, \beta_{31}) = 3 \times 2 + 30 = 36$$

Now, $$x_2 = \text{Min}\left\{\beta_{21}, \frac{\beta_{31} - 3x_1}{2}\right\}$$

$$= \text{Min}\left\{6, \frac{18-3x_1}{2}\right\} = \text{Min}\,(6, 6) = 6$$

The optimal solution is, Max $Z = 36$, $x_1 = 2$, $x_2 = 6$.

Example 3.16: Use dynamic programming to solve the LPP

Max $Z = x_1 + 9x_2$

Subject to the constraints, $2x_1 + x_2 \le 25$

$$x_2 \le 11$$

$$x_1, x_2 \ge 0$$

Solution: The problem has two resources and two decision variables. The states of the equivalent dynamic programming are β_{1j}, β_{2j}, $j = 1, 2$.

$$f_2(\beta_{12}, \beta_{22}) = \text{Max}\,(9x_2)$$
$$0 < x_2 \leq 25$$
$$0 \leq x_2 \leq 11$$

i.e., $$f_2(\beta_{12}, \beta_{22}) = 9\,\text{Max}\,(x_2)$$
$$= 9\,\text{Max}\,(25, 11)$$

Since the Max of x_2 satisfying the conditions of $x_2 \leq 25, x_2 \leq 11$, is the Min of (25, 11)

$\therefore$ $$x_2^* = 11$$

Now, $$f_1(\beta_{11}, \beta_{21}) = \text{Max}\,[x_1 + f_2(\beta_{11} - 2x_1, \beta_{21} - 0)]$$
$$0 \leq x_1 \leq \frac{25}{2}$$

At this last stage, substitute $\beta_{11} = 25, \beta_{21} = 11$

$$f_1(25, 11) = \text{Max}\,[x_1 + 9\,\text{Min}\,(25 - 2x_1, 11)]$$
$$\text{Min}\,(25 - 2x_1, 11) = \{11, 0 \leq x_1 \leq 7\}$$
$$25 - 2x_1, 7 \leq x_1 \leq \frac{25}{2}$$

$\therefore$ $$x_1 + 9\,\text{Min}\,(25 - 2x_1, 11) = x_1 + 99, 0 \leq x_1 \leq 7$$
$$225 - 17x_1, 7 \leq x_1 \leq \frac{25}{2}$$

Since the maximum of both ($x_1 + 99, 225 - 17x_1$) occurs at $x_1 = 7$

$$f_1(25, 11) = 7 + 9\,\text{Min}\,(11, 11)$$
$$= 106 \text{ at } x_1^* = 7$$
$$x_1^* = \text{Min}\,(25 - 2x_1^*, 11) = \text{Min}\,(11, 11) = 11$$

Hence the optimum solution is,

$$x_1^* = 7, x_2^* = 11 \text{ and Max } Z = 106$$

Check Your Progress

4. What do you understand by the game theory?
5. How can we illustrate a game?
6. State the standard conventions in game theory.
7. What do you mean by the two-person zero-sum games?
8. What are the basic concepts of two-person zero-sum games?
9. What do you mean by the value of a game?
10. Mention the strategy used in game theory.
11. Define the term strategy combination.
12. What do you mean by the term mixed strategies?
13. What do you understand by the totally mixed strategy?

3.4 ANSWERS TO 'CHECK YOUR PROGRESS'

NOTES

1. The technique of dynamic programming divides a given problem into stages or subproblems and then solves the subproblems sequentially (usually working backwards from the natural end of the problem) until the initial problem is finally solved.
2. Stage: It refers to the particular decision we are facing. If we are to take 5 sequential decisions then we have 5 stages. Alternatively a stage can be understood as a subproblem. Thus, in case of five stages we have five subproblems of a given problem.

   State Variable: It is a variable defining the current situation at any stage. Suppose we have a pricing problem, then the state variable would be the current price level at a given stage.

   Optimal Decision Rule: This rule specifies which decision to make, as a function of state variable and the stage number.

   Optimal Policy: An optimal policy is a set of optimal decision rules which guides one's decisions through all stages of the given problem. In fact, this constitutes the solution of the given problem optimizing the desired measure of effectiveness.
3. (i) Travelling Salesman's Problem

   (ii) Pricing Problem

   (iii) Production Scheduling Inventory Problem

   (iv) Allocation Problem
4. The term 'Game' represents a conflict between two or more parties. Game theory is really the 'Science of Conflict'. It is not concerned with finding an optimum or winning strategy for a particular conflict situation but it provides general rules concerning the logic that underlies strategic behaviour of all types.

   Game theory applies to those competitive situations that are technically known as 'Competitive Games' or simply 'Games'.
5. Suppose there are two opponents X and Y. We can think of them as sitting across a table from each other and each with two buttons in front of him. We shall denote player X's buttons, m and n and player Y's buttons as r and t; thus each player has two choices open to him. We also presume a partition between them so that neither can see in advance which button his opponent is going to press. At a signal from a third party, each player presses one of his buttons.
6. In order to eliminate the necessity for written descriptions of the 'Payoffs' (as we have shown in the above two examples), a standard set of conventions has been established in game theory. It is the usual practice to omit a description like 'Player X wins two points' and replace it with integer 2. The positive algebraic sign which is assumed to accompany this number indicates that it is Player X who benefits from this payoff. Similarly, instead

of saying 'Player Y wins three points' one simply indicates this with the value 3, the minus sign indicating that it is Player Y who benefits from this particular payoff. Another standard convention that is usually followed is that Player X has choices between the rows and Player Y has choices between the columns.

7. In game theory, a zero-sum game is a mathematical representation of a Game Theory situation in which each participant's gain or loss of utility is exactly balanced by the losses or gains of the utility of the other participants. Alternatively, we can say that the zero-sum games are the games in which one player's win is the other player's loss.

8. Following are the basic and significant concepts of simple two-person zero-sum games:

   - A two-person game is characterized by the strategies of each player and the payoff matrix.
   - The payoff matrix shows the gain (positive or negative) for player 1 that would result from each combination of strategies for the two players. Remember that the matrix for player 2 is the negative of the matrix for player 1 in a zero-sum game.
   - The entries in the payoff matrix can be in any units as long as they represent the utility (or value) to the player.
   - There are two key assumptions about the behaviour of the players. The first is that both players are rational. The second is that both players are materialistic meaning that they choose their strategies in their own interest.

9. In game theory, the concept 'Value of a Game' is considered very important. It refers to the average payoff per play of the game over an extended period of time. But determining the value of the game is not always as simple as in the case of these two examples. In case the players determine that their best alternative is to play each row or each column a certain part of the time, calculating the value of the game becomes a bit more complex.

10. In game theory, the strategy of a player in a game is a complete plan of action for any situation that may occur. This determines the complete behaviour of player and the player's strategy determines the action that the player will take at any stage of the game.

11. A strategy profile is also sometimes termed as strategy combination. It is a set of strategies for each player that specifies all actions in a game. It must include one and only one strategy for every player. Sometimes the strategy concept is by mistake confused with that of a move. A move is an action taken by a player at some point during the play of a game whereas a strategy is a complete algorithm for playing the game because it guides a player what to do for every possible situation throughout the game. A player's strategy set describes what strategies are available for playing the game. Strategies are of two types, pure and mixed.

12. Mixed strategy is an assignment of a probability to each pure strategy. This allows for a player to randomly select a pure strategy. Since probabilities are continuous, there are infinitely many mixed strategies available to a player,

even if their strategy set is finite. Certainly, a pure strategy can be considered as a degenerate case of a mixed strategy in which that specific pure strategy is selected with probability 1 and every other strategy with probability 0.

13. A totally mixed strategy is a mixed strategy in which the player assigns a strictly positive probability to every pure strategy. The totally mixed strategies are important for equilibrium refinement.

3.5 SUMMARY

- Dynamic programming is the mathematical technique whose development is largely due to Richard E. Bellman.
- In applying the dynamic programming technique, it is necessary to divide the problem into a number of subproblems or decision stages. It is also necessary to describe the stage of the system by a state variable.
- The basic ideas used in dynamic programming approach or the sequential decision theory can be well understood through the application of this technique to different situations.
- The term 'Game' represents a conflict between two or more parties. Game theory is really the 'Science of Conflict'. It is not concerned with finding an optimum or winning strategy for a particular conflict situation but it provides general rules concerning the logic that underlies strategic behaviour of all types.

  Game theory applies to those competitive situations that are technically known as 'Competitive Games' or simply 'Games'.
- Suppose there are two opponents X and Y. We can think of them as sitting across a table from each other and each with two buttons in front of him. We shall denote player X's buttons, m and n and player Y's buttons as r and t; thus, each player has two choices open to him. We also presume a partition between them so that neither can see in advance which button his opponent is going to press. At a signal from a third party, each player presses one of his buttons.
- In two-person games, the players may have many possible choices open to them for each play of the game but the number of player's remains only two. But games can also involve many people as active participants, each with his own set of choices for each play of the game. A game of three players can be named as three-person game. Thus, in case of more than two persons, the game is generally named an n person game.
- Zero-sum and non-zero-sum games. A zero-sum game is one in which the sum of the payments to all the competitors is zero for every possible outcome of the game. In other words, in such a game the sum of the points won equals the sum of the points lost, i.e., one player wins at the expense of the other (others).
- Games of perfect information and games of imperfect information. Whatever strategy is adopted by either player, if the same can also be discovered by his competitor, then such games are known as games of perfect information. In games of imperfect information, neither player knows the entire situation

and must be guided in part by guess work as to what the real situation is.

- Games with finite, i.e., limited number of moves (or plays) and games with unlimited number of moves. Games with finite number of moves are those where the number of moves is limited to a fixed magnitude before play begins. If the game could continue over an extended period of time and no limit is put on the number of moves, it is referred to as a game with an unlimited number of moves.
- 2 × 2 two-person games and 2 × *m* and *m* × 2 games. Two-person zero-sum games with only two choices open to each player are denoted as 2 × 2 two person games but games in which one of the players has more than two choices of rows or columns and in which the other player has exactly two choices are referred to as *m* × 2 or 2 × *m* games respectively.
- In game theory, a zero-sum game is a mathematical representation of a situation in which each participant's gain or loss of utility is exactly balanced by the losses or gains of the utility of the other participants. Alternatively, we can say that the zero-sum games are the games in which one player's win is the other player's loss.
- A two-player game is called a zero-sum game if the sum of the payoffs to each player is constant for all possible outcomes of the game. More specifically, the terms or coordinates in each payoff vector must add up to the same value for each payoff vector. Such games are sometimes called constant-sum games as an alternative.
- A strategy profile is also sometimes termed as strategy combination. It is a set of strategies for each player that specifies all actions in a game. It must include one and only one strategy for every player. Sometimes the strategy concept is by mistake confused with that of a move.
- Mixed strategy is an assignment of a probability to each pure strategy. This allows for a player to randomly select a pure strategy. Since probabilities are continuous, there are infinitely many mixed strategies available to a player, even if their strategy set is finite. Certainly, a pure strategy can be considered as a degenerate case of a mixed strategy in which that specific pure strategy is selected with probability 1 and every other strategy with probability 0.
- A totally mixed strategy is a mixed strategy in which the player assigns a strictly positive probability to every pure strategy. The totally mixed strategies are important for equilibrium refinement.

3.6 KEY TERMS

- **Dynamic programming:** A mathematical technique which is applicable to many types of problems where a series of interrelated decisions are required.
- **Game theory:** Game theory is the 'Science of Conflicts'.
- **Game:** A conflict between two or more parties.
- **Zero-sum and non-zero games:** A zero-sum game is one in which the sum of the payments to all the competitors is zero for every possible outcome of the game.

- **Mixed strategies:** Mixed strategy is an assignment of a probability to each pure strategy.
- **Totally mixed strategy:** A totally mixed strategy is a mixed strategy in which the player assigns a strictly positive probability to every pure strategy.
- **Graphic method:** The graphic method can only be used in games with no saddle point and having payoff m x n matrices where either m or n is two.

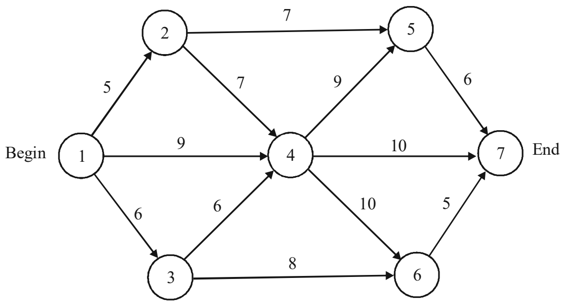
3.7 SELF-ASSESSMENT QUESTIONS AND EXERCISES

Short-Answer Questions

1. State the differences between linear and dynamic programming.
2. What different terms are often talked about in context of dynamic programming?
3. What do you understand by dynamic programming and principle of optimality developed by Richard E. Bellman?
4. Write a short note on recursive approach method to solve dynamic programming problems.
5. What is game theory?
6. Define the illustration of a game.
7. State the standard conventions that are used in game theory.
8. What are the basic concepts of two-person zero-sum games?
9. Define about the value of game.
10. What are mixed strategies?
11. Define the meaning of the term ‘Game’.
12. Write the properties of game.
13. State about the constant-sum games.

Long-Answer Questions

1. What do you mean by deterministic and probabilistic dynamic programming? Explain.
2. Use the dynamic programming approach to find (i) The maximum path (ii) The minimum path through network:



NOTES

3. Seven units of capital can be invested in four activities with the return from each activity given in the following table. Find the allocation of capital to each activity that will maximize the total return.

   What will be the total return if the available capital is reduced by 1 unit?

Q	$g^1(Q)$	$g^3(Q)$	$f(Q)$	$g^4(Q)$
0	0	0	0	0
1	2	3	2	1
2	4	5	3	3
3	6	7	4	5
4	7	9	5	6
5	8	10	5	7
6	9	11	5	8
7	9	12	5	8

4. An investor has ₹ 1,00,000 to invest some time in the next five months. Each month a new investment opportunity arises, the return each investment promises is a random variable distributed according to the probabilities given in following table:

Return on investment	*Probability*
20 per cent	0.5
60 per cent	0.3
75 per cent	0.2

   At the beginning of each month an investment opportunity is presented with some actual return based on the probabilities in the table provided. The investor must either take that opportunity or reject it. Once rejected, the opportunity is withdrawn. If the investor has not invested his money by the fifth month, he must invest it at the return available in the fifth month. What his expected return would be if he wants to make an optimal decision?

5. Given are the following transition matrices:

	To		
From	2	3	4
1	₹ 2	₹ 4	₹ 4

	To		
From	5	6	7
2	₹ 7	₹ 4	₹ 6
3	₹ 3	₹ 2	₹ 4
4	₹ 4	₹ 1	₹ 5

	To	
From	8	9
5	₹ 1	₹ 4
6	₹ 6	₹ 3
7	₹ 3	₹ 3

	To
From	10
8	₹ 3
9	₹ 4

   (a) Determine the least-cost route from 1 to 10.

   (b) Determine the highest-cost route from 1 to 10.

6. Analyse the maximin-minimax principal. Give appropriate examples.

7. Discuss briefly about the standard conventions that are used in game theory.

8. Explain the terms used in game and its types along with examples.

NOTES

9. Use graphic methods in solving the following games:

$$(a)\quad \begin{array}{c} \\ A \end{array}\overset{B}{\begin{pmatrix} 3 & -3 & 4 \\ -1 & 1 & -3 \end{pmatrix}}$$

$$(b)\quad \begin{array}{c} \\ X \end{array}\overset{Y}{\begin{pmatrix} 2 & 5 \\ 4 & 1 \end{pmatrix}}$$

10. The following matrix represents the payoff to P_1 in a rectangular game between two persons P_1 and P_2.

$$P_1 \overset{P_2}{\begin{bmatrix} 8 & 15 & -4 & -2 \\ 19 & 15 & 17 & 16 \\ 0 & 20 & 15 & 5 \end{bmatrix}}$$

By the notion of dominance, reduce the game to a 2 × 4 game and solve it graphically.

11. Solve the following game:

Player B

$$\begin{array}{c|cccc} & I & II & III & IV \\ \hline I & 3 & 2 & 4 & 0 \\ II & 3 & 4 & 2 & 4 \\ III & 4 & 2 & 4 & 0 \\ IV & 0 & 4 & 0 & 8 \end{array}$$

3.8 FURTHER READING

Kalavathy, S. 2002. *Operations Research*. New Delhi: Vikas Publishing House Pvt. Ltd.

Gupta, P.K. and D.S. Hira. 2002. *Introduction to Operations Research*. New Delhi: S. Chand And Company Limited.

Taha, H.A. 2006. *Operations Research: An Introduction*, 8th Edition. New Delhi: Prentice-Hall of India.

Jensen, Paul A. and Jonathan F. Bard. 2003. *Operations Research Models and Methods*. New York: John Wiley & Sons.

Arumugam, R. S. 2006. *Operations Research*. Palayamkottai (Tamil Nadu): New Gamma Publications.

Sundharesan, V., K. S. Ganapathy and K. Ganesan. 2017. *Resource Management Techniques (Operations Research)*. Chennai: A. R. Publications.

Gillett, Billy E. 2007. *Introduction to Operations Research*. New Delhi: Tata McGraw-Hill.

NOTES

Ackoff, R. L. and M. W. Sasieni. 1968. *Fundamentals of Operations Research*. New York: John Wiley & Sons Inc.

Kothari, C. R. 1992. *An Introduction to Operational Research*. New Delhi: Vikas Publishing House Pvt. Ltd.

Sharma, J. K. 2001. *Operations Research: Theory and Applications*. New Delhi: Macmillan India Ltd.

Swaroop, Kanti, P.K. Gupta and Man Mohan. 2001. *Operations Research*. New Delhi: Sultan Chand & Sons.

UNIT 4 INTEGER PROGRAMMING

NOTES

Structure

4.0 INTRODUCTION

An Integer Programming Problem (IPP) is a mathematical optimization or feasibility program in which some or all of the variables are restricted to be integers. In many settings the term refers to Integer Linear Programming (ILP), in which the objective function and the constraints (other than the integer constraints) are linear. Integer programming is NP-complete. In particular, the special case of 0-1 integer linear programming, in which unknowns are binary, and only the restrictions must be satisfied, is one of Karp's 21 NP-complete problems. If some decision variables are not discrete the problem is known as a mixed-integer programming problem. This is a type of linear programming in which all or some variables are constrained to assume non-negative integer values. Problems related to such programming are known as integer programming problems. If all variables assume integer values, then it is called Pure Integer Programming Problem (Pure IPP). But in the optimal solution, if this restriction of integer is not on all and only few can assume non-integer solution, then it is a case of mixed integer programming problem. If these integers are limited to either of the two values 0 or 1, then these are known as '0 – 1 Programming Problems' or 'Standard Discrete Programming Problem'. Nature of problems where there is decision making, depends on either to do or not to do. The choice is between two outcomes, in which one outcome has to be selected. Integer programming problem is applied in business and industries. Problems like traveling salesman problem, assignment problem and transportation problems are part of integer programming where the decision variables are either 0 or 1. If x_{ij} represents an activity from i to j, then $x_{ij} = 1$, which specifies that the activity is performed, and when $x_{ij} = 0$ then activity is not performed.

Replacement theory is concerned with the prediction of replacement costs and determination of the most economic replacement policy. The problems of replacement are encountered in case of both men and machines. The replacement theory is an equally important aspect of OR. In case of items whose efficiency go on decreasing according to their age, we have to spend more money on account of increased operating cost, increased repair cost, etc. In such cases, the replacement of an old item with a new one is the only alternative to prevent such increased expenses. Thus, it becomes necessary to determine the age at which replacement is more economical rather than to continue the operations at increased cost. Replacement theory is equally important in context of those equipment's which fail completely and instantaneously. As such, the study of replacement theory is the study of the technique of formulating the appropriate replacement policy in situations that arise when some items need replacement for one reason or the other. Sequencing models determine an appropriate order (sequence) for a series of jobs to be done on a finite number of service facilities in some pre-assigned order, so as to optimize the total cost (time) involved. The algorithm, which is used to optimize the total elapsed time for processing n jobs through two machines is called 'Johnson's Algorithm'.

In this unit, you will learn about the integer programming and cutting plan techniques, simulation problem, replacement problems and sequencing.

4.1 OBJECTIVES

After going through this unit, you will be able to:

- Discuss the significance of Integer Programming
- Analyse principles of Integer Programming Problem (IPP) in real life situations
- Solve problems on IPP using the cutting plane method branch and bound or search methods
- Explain the simulation and problems of implementation
- State the reasons for replacement
- Explain the maintenance and reliability problems
- Describe the basic concept of sequencing models

4.2 INTEGER PROGRAMMING AND CUTTING PLAN TECHNIQUES

A linear programming problem in which all or some of the decision variables are constrained to assume non-negative integer values is called an *Integer Programming Problem* (IPP).

In a *Linear Programming Problem* (LPP) if all variables are required to take integer values then it is called the *Pure* (*all*) *Integer Programming Problem* (Pure IPP).

If only some of the variables in the optimal solution of a LPP are restricted to assume non-negative integer values, while the remaining variables are free to take any non-negative values, then it is called a *Mixed Integer Programming Problem* (Mixed IPP).

Further, if all the variables in the optimal solution are allowed to take values 0 or 1, then the problem is called the 0–1 *Programming Problem* or *Standard Discrete Programming Problem.*

The general integer programming problem is given by,

Max $Z = CX$

Subject to constraints,

$A\,X \le B$

$X \ge 0$ Where some or all variables are integers.

NOTES

Importance of Integer Programming Problems

In IPP, all the decision variables are allowed to take any non-negative real values as it is quite possible and appropriate to have fractional values in many situations. There are several frequently occurring circumstances in business and industries that lead to planning models involving integer-valued variables. For example, in production, manufacturing is frequently scheduled in terms of batches, lots or runs. In allocation of goods, a shipment must involve a discrete number of trucks or aircrafts. In such cases the fractional values of variables like 13/3 may be meaningless in the context of the actual decision problem.

This is the main reason why integer programming is so important for marginal decisions.

Applications of Integer Programming

Integer programming is applied in business and industries. All assignment and transportation problems are integer programming problems, because in the assignment and travelling salesmen problem all the decision variables are either zero or one.

i.e., $x_{ij} = 0$ or 1

Other examples include capital budgeting and production scheduling problems. In fact, any situation involving decisions of the type 'Either to do a job or not' can be viewed as an IPP. In all such situations,

$x_{ij} = 1$, if the jth activity is performed.

$x_{ij} = 0$, if the jth activity is not performed.

In addition, allocation problems involving the allocation of men or machines give rise to IPP, since such commodities can be assigned only in integers and not in fractions.

***Note**:* If the non-integer variable is rounded off, then it violates the feasibility and there is no guarantee that the rounded off solution will be optimal. Due to these difficulties, there is a need for developing a systematic and efficient procedure for obtaining the exact optimal integer solution to such problems.

4.2.1 Methods of Integer Programming Problem

Two methods are used to solve IPP:

(*i*) Gomory's Cutting Plane Method

(*ii*) Branch and Bound Method or Search Method

Gomory's Cutting Plane Method

A systematic procedure for solving pure IPP was first developed by R.E. Gomory in 1956, which he later used to deal with the more complicated cases of mixed integer programming problems. This method consists of first solving the IPP as an ordinary LPP by ignoring the restriction of integer values and then introducing a new constraint to the problem such that the new set of feasible solution includes all the original feasible integer solutions, but does not include the optimum non-integer solution initially found. This new constraint is called 'Fractional Cut' or 'Gomorian Constraint'. Then the revised problem is solved using the simplex method, till an optimum integer solution is obtained.

Branch and Bound Method or Search Method

This is an enumeration method in which all feasible integer points are enumerated. This is the widely used search method based on Branch and Bound technique. It was developed in 1960 by A.H. Land and A.G. Doig. This method is applicable to both pure and mixed IPP. It first divides the feasible region into smaller subsets that eliminate parts containing no feasible integer solution.

Gomory's Fractional Cut Algorithm or Cutting Plane Method for Pure (All) IPP

Step 1: Convert the minimization IPP into an equivalent maximization LPP. Ignore the integrality condition.

Step 2: Introduce slack and/or surplus variables if necessary to convert the given LPP in its standard form and obtain the optimum solution of the given LPP by using simplex method.

Step 3: Test the integrality of the optimum solution.

(*i*) If all $x_{Bi} \geq 0$ and are integers, then an optimum integer solution is obtained.

(*ii*) If all $x_{Bi} \geq 0$ and at least one x_{Bi} is not an integer, then go to the next step.

Step 4: Rewrite each x_{Bi} as $x_{Bi} = [x_{Bi}] + f_i$, where x_{Bi} is the integral part of x_{Bi} and f_i is the positive fractional part of x_{Bi}, $0 < f_i < 1$.

Choose the largest fraction of x_{Bi}, i.e., choose Max (f_i). If there is a tie, then select arbitrarily. Let Max $(f_i) = f_k$, corresponding to x_{Bk} (the *k*th row is called the 'Source Row').

Step 5: Express each negative fraction, if any, in the source row of the optimum simplex table as the sum of a negative integer and a non-negative fraction.

Step 6: Find the fractional cut constraint or Gomorian constraint.

From the source row, $$\sum_{j=1}^{n} a_{kj} x_j = x_{Bi}$$

i.e., $$\sum_{j=1}^{n} ([a_{kj}] + f_{kj})\, x_j = [x_{Bk}] + f_k$$

The equation is in the form, $$\sum_{j=1}^{n} f_{kj} x_j \geq f_K \Rightarrow -\sum_{j=1}^{n} f_{kj}\, x_j \leq -f_k$$

Or, $$-\sum_{j=1}^{n} f_{kj}x_j + G_1 = -f_k$$

Where, G_1 is the Gomorian slack.

NOTES

Step 7: Add the fractional cut constraint obtained in Step (6) at the bottom of the simplex table obtained in Step (2). Find the new feasible optimum solution using dual simplex method.

Step 8: Go to Step (3) and repeat the procedure until an optimum integer solution is obtained.

Example 4.1: Find the optimum integer solution to the following IPP.

Max $Z = x_1 + x_2$

Subject to constraints, $3x_1 + 2x_2 \leq 5$

$x_2 \leq 2$

$x_1, x_2 \geq 0$ and are integers.

Solution: After introducing the non-negative slack variables $S_1, S_2 \geq 0$, the standard form of the IPP becomes,

Max $Z = x_1 + x_2 + 0S_1 + 0S_2$

Subject to constrains, $3x_1 + 2x_2 + S_1 = 5$

$0x_1 + x_2 + S_2 = 2$

$x_1, x_2, S_1, S_2 \geq 0$

Ignoring the integrality condition, solve the problem by simplex method. The initial basic feasible solution is given by putting $x_1 = 0$ and $x_2 = 0$.

Hence,

$$S_1 = 5 \text{ and } S_2 = 2$$

		C_j	1	1	0	0	
C_B	B	x_B	x_1	x_2	S_1	S_2	$Min \frac{x_B}{x_i}$
← 0	S_1	5	3	2	1	0	5/3←
0	S_2	2	0	1	0	1	—
	Z_j	0	0	0	0	0	
	$Z_j - C_j$		–1↑	–1	0	0	
1	x_1	5/3	1	2/3	1/3	0	5/2 = 2.5
← 0	S_2	2	0	1	0	1	2/1 = 2←
	Z_j	5/3	1	2/3	1/3	0	
	$Z_j - C_j$		0	–1/3↑	1/3	0	
1	x_1	1/3	1	0	1/3	–2/3	
1	x_2	2	0	1	0	1	
	Z_j	7/3	1	1	1/3	1/3	
	$Z_j - C_j$		0	0	1/3	1/3	

Since all $Z_j - C_j \geq 0$, an optimum solution is obtained, which is given by:

Max $Z = 7/3$, $x_1 = 1/3$, $x_2 = 2$

To obtain an optimum integer solution, we have to add a fractional cut constraint in the optimum simplex table.

Since $x_B = 1/3$, the source row is the first row.

Expressing the negative fraction –2/3 as a sum of negative integer and positive fraction, we get $-2/3 = -1 + 1/3$

Since x_1 is the source row, we have,

$$1/3 = x_1 + 1/3\ S_1 - 2/3\ S_2$$

i.e.,
$$1/3 = x_1 + 1/3S_1 + (-1 + 1/3)\ S_2$$

The fractional cut Gomorian constraint is given by,

$$1/3S_1 + 1/3S_2 \geq 1/3$$

$$\Rightarrow \quad -1/3S_1 - 1/3S_2 \leq -1/3$$

$$\Rightarrow \quad -1/3S_1 - 1/3\ S_2 + G_1 = -1/3$$

Where, G_1 is the Gomorian slack. Add this fractional cut constraint at the bottom of the above optimal simplex table.

We apply dual simplex method. Since $G_1 = -1/3$, hence G_1 leaves the basis. To find the entering variable we find,

$$\text{Max}\left\{\frac{Z_j - C_j}{a_{ij}},\ a_{ij} < 0\right\} = \text{Max}\left\{\frac{1/3}{-1/3}, \frac{1/3}{-1/3}\right\}$$

$$\text{Max}\ \{-1, -1\} = -1$$

We choose S_1 as the entering variable arbitrarily.

		C_j	1	1	0	0	0
C_B	B	x_B	x_1	x_2	S_1	S_2	G_1
1	x_1	1/3	1	0	1/3	–2/3	0
1	x_2	2	0	1	0	1	0
0	G_1	–1/3	0	0	–1/3	–1/3	1←
	Z_j	7/3	1	1	1/3	1/3	0
	$Z_j - C_j$		0	0	1/3↑	1/3	0
1	x_1	0	1	0	0	–1	1
1	x_2	2	0	1	0	1	1
0	S_1	1	0	0	1	1	–3
	Z_j	2	1	1	0	0	1
	$Z_j - C_j$		0	0	0	0	1

Since all $Z_j - C_j \geq 0$ and all $x_{Bi} \geq 0$, we obtain an optimal feasible integer solution.

∴ The optimum integer solution is,

Max $Z = 2$, $x_1 = 0$, $x_2 = 2$.

Example 4.2: Find an optimum integer solution to the following IPP.

Max $Z = x_1 + 2x_2$

Subject to contraints,

$$2x_2 \leq 7$$
$$x_1 + x_2 \leq 7$$
$$2x_1 \leq 11$$
$$x_1,\ x_2 \geq 0 \text{ and are integers.}$$

Solution: Introducing slack variables $S_1, S_2, S_3 \geq 0$, we get,

Max $Z = x_1 + 2x_2 + 0S_1 + 0S_2 + 0S_3$

Subject to constraints,

$$2x_2 + S_1 = 7$$
$$x_1 + x_2 + S_2 = 7$$
$$2x_1 + S_3 = 11$$

Ignoring the integrality condition, we get the optimum solution of the given IPP with initial basic feasible solution obtained by putting $x_1 = 0$ and $x_2 = 0$ as $S_1 = 7$, $S_2 = 7$ and $S_3 = 11$.

		C_j	1	2	0	0	0	
C_B	B	x_B	x_1	x_2	S_1	S_2	S_3	$Min \frac{x_B}{x_i}$
0	S_1	7	0	2	1	0	0	7/2 = 3.5←
0	S_2	7	1	1	0	1	0	7/1 = 7
0	S_3	11	2	0	0	0	1	—
	Z_j	0	0	0	0	0	0	
	$Z_j - C_j$		–1	–2↑	0	0	0	
2	x_2	7/2	0	1	1/2	0	0	—
←0	S_2	7/2	1	0	–1/2	1	0	7/2 = 3.5
0	S_3	11	2	0	0	0	1	11/2 = 5.5
	Z_j	7	0	2	1	0	0	
	$Z_j - C_j$		–1↑	0	1	0	0	
2	x_2	7/2	0	1	1/2	0	0	
1	x_1	7/2	1	0	–1/2	1	0	
0	S_3	4	0	0	1	–2	1	
	Z_j	21/2	1	2	1/2	1	0	
	$Z_j - C_j$		0	0	1/2	1	0	

Since all $Z_j - C_j \geq 0$, an optimum solution is obtained which is given by,

$$\text{Max } Z = \frac{21}{2},\ x_1 = \frac{7}{2},\ x_2 = \frac{7}{2}$$

Since the optimum solution obtained above is not an integer, we now select a constraint corresponding to,

$$\text{Max }(fi) = \text{Max }(f_1,\ f_2,\ f_3)$$

$$x_1 = 7/2 = 3 + 1/2$$

$$x_2 = 7/2 = 3 + 1/2$$

$$S_3 = 4 = 4 + 0$$

$$\therefore \qquad \text{Max }(f_i) = \text{Max }\left(\frac{1}{2}, \frac{1}{2}, 0\right) = 1/2$$

Since the Max fraction is same for both x_1 and x_2 rows, we choose x_1 row as the source row arbitrarily. From this row we have,

$$7/2 = x_1 + 0x_2 - 1/2\ S_1 + 1S_2 + 0S_3$$

On expressing the negative fraction as a sum of negative integer and a positive fraction, we have,

$$3 + 1/2 = x_1 + 0x_2 + (-1 + 1/2)\ S_1 + 1S_2 + 0S_3$$

$\therefore$ The Gomorian constraint is given by,

$$1/2\ S_1 \geq 1/2$$

i.e.,

$$-1/2\ S_1 \leq -1/2 \Rightarrow -1/2\ S_1 + G_1 = -1/2$$

Here, G_1 is the Gomorian slack. Adding this new constraint at the bottom of the above optimal simplex table, we get a new table.

We apply dual simplex method. Since $G_1 = -1/2$, G_1 leaves the basis. Entering variable is given by,

$$\text{Max}\left\{\frac{Z_j - C_j}{a_{ij}}, a_{ij} < 0\right\} = \text{Max}\left\{\frac{1/2}{-1/2}\right\}$$

This gives the non-basic variable S_1 to enter into the basis. Drop G_1 and introduce S_1.

		C_j	1	2	0	0	0	0
C_B	B	x_B	x_1	x_2	S_1	S_2	S_3	G_1
2	x_2	7/2	0	1	1/2	0	0	0
1	x_1	7/2	1	0	–1/2	1	0	0
0	S_3	4	0	0	1	–2	1	0
0	G_1	–1/2	0	0	–1/2	0	0	1←
	Z_j	21/2	1	2	1/2	1	0	0
	$Z_j - C_j$		0	0	1/2 ↑	1	0	0
2	x_2	3	0	1	0	0	0	1
1	x_1	4	1	0	0	1	0	–1
0	S_3	3	0	0	0	–2	1	2
0	S_1	1	0	0	1	0	0	–2
	Z_j	10	1	2	0	1	0	1
	$Z_j - C_j$		0	0	0	1	0	1

Since all $Z_j - C_j \geq 0$, an optimum solution has been obtained in integers. Hence, the integer optimum solution is given by,

$$\text{Max } Z = 10,\ x_1 = 4,\ x_2 = 3$$

NOTES

Example 4.3: Solve the following integer programming problem.

Max $Z = 2x_1 + 20x_2 - 10x_3$

Subject to constraints, $2x_1 + 20x_2 + 4x_3 \leq 15$

$6x_1 + 20x_2 + 4x_3 = 20$

$x_1, x_2, x_3 \geq 0$ and are integers.

Solution: Introducing slack variable $S_1 \geq 0$ and an artificial variable $A_1 \geq 0$, the initial basic feasible solution becomes $S_1 = 15$, $A_1 = 20$ by putting $x_1 = x_2 = x_5 = 0$ Ignoring the integer condition, solve the problem using simplex method.

Max $Z = 2x_1 + 20x_2 - 10x_3 + 0S_1 - MA_1$

Subject to constraints, $2x_1 + 20x_2 + 4x_3 + S_1 = 15$

$6x_1 + 20x_2 + 4x_3 + A_1 \quad = 20$

$x_1, x_2, x_3, S_1, A_1 \quad \geq 0$

		C_j	2	20	–10	0	$-M$	
C_B	B	x_B	x_1	x_2	x_3	S_1	A_1	$Min\ \frac{x_B}{x_i}$
0	S_1	15	2	20	4	1	0	15/20 = 3/4 ←
$-M$	A_1	20	6	20	4	0	1	20/20 = 1
	Z_j	$-20M$	$-6M$	$-20M$	$-4M$	0	$-M$	
	$Z_j - C_j$		$-6M - 2$	$-20M - 20$	$-4M + 10$	0	0	
				↑				
20	x_2	3/4	1/10	1	$\frac{1}{5}$	1/20	0	$\frac{3}{4} \times 10 = 7.5$
$-M$	A_1	5	4	0	0	–1	1	$\frac{5}{4} = 1.25$
	Z_j	$15 - 5M$	$2 - 4M$	20	4	$1 + M$	$-M$	
	$Z_j - C_j$		$-4\ M$	0	14	$M\ + 1$	0	
			↑					
20	x_2	5/8	0	1		1/5	3/40	—
2	x_1	5/4	1	0		0	–1/4	—
	Z_j	15	2	20		4	1	—
	$Z_j - C_j$		0	0		14	1	

Since all $Z_j - C_j \geq 0$, an optimum solution is obtained which is given by,

$$\text{Max } Z = 15,\ x_1 = 5/4,\ x_2 = 5/8,\ x_3 = 0$$

NOTES

Since the optimum solution obtained above is not an integer hence select a constraint corresponding to,

$$\text{Max }(f_i) = \text{ Max }(f_1, f_2)$$

$$x_1 = \frac{5}{4} \text{ and } x_2 = \frac{5}{8}$$

$$\therefore \qquad \text{Max }(f_i) = \text{ Max }\left(\frac{5}{4}, \frac{5}{8}\right) = \frac{5}{8}$$

$$\therefore \qquad \text{Max }(f_1, f_2) = \text{Max }(5/4,\ 5/8) = 5/8$$

$\therefore$ The source row is the first row, namely x_2 row. From this source row we have,

$$5/8 = 0x_1 + 1x_2 + (1/5)x_3 + (3/40)S_1$$

The fractional cut constraint is given by,

$$(1/5)x_3 + (3/40)S_1 \geq 5/8$$

$$(-1/5)x_3 - (3/40)S_1 \leq -5/8 \Rightarrow -(1/5)x_3 - (3/40)S_1 + G_1 = -5/8$$

Here, G_1 is the Gomorian slack.

Adding this additional constraint in the optimum simplex table, we obtain the new table as given below.

We apply dual simplex method. Since $G_1 = -5/8$, hence G_1 leaves the basis.

Also, $\text{Max}\left\{\frac{Z_j - C_j}{a_{ij}}, a_{ij} < 0\right\} = \text{Max}\left\{\frac{14}{-1/5}\ \frac{1}{-3/40}\right\} = \text{Max} -\frac{40}{3}$

This gives the non-basic variable S_1, which enters the basis.

		C_j	2	20	–10	0	0
C_B	B	x_B	x_1	x_2	x_3	S_1	G_1
20	x_2	$\frac{5}{8}$	0	1	1/5	3/40	0
2	x_1	$\frac{5}{4}$	1	0	0	–1/4	0
0	G_1	$\frac{-5}{8}$	0	0	–1/5	–3/40	1←
	Z_j	15	2	20	4	1	0
	$Z_j - C_j$		0	0	14	1 ↑	0
20	x_2	0	0	1	0	0	1
2	x_1	10/3	1	0	2/3	0	–10/3
0	S_1	25/3	0	0	8/3	1	–40/3
	Z_j	20/3	2	20	4/3	0	40/3
	$Z_j - C_j$		0	0	34/3	0	40/3

Again, since the solution is non-integer, we add one more fractional cut constraint.

$$\therefore \quad x_1 = \frac{10}{3} = 3 + \frac{1}{3},\ x_2 = 0 \quad \text{and} \quad S_1 = \frac{25}{3} = 8 + \frac{1}{3}$$

$$\therefore \qquad \text{Max}\{f_i\} = \text{Max}\ (\frac{1}{3},\ 0,\ 1/3)$$

Since, the Max fraction is same for both the rows x_1 and S_1, we choose S_1 arbitrarily.

$\therefore$ From the source row we have,

$$25/3 = 0x_1 + 0x_2 + (8/3)x_3 + 1S_1 - (40/3)G_1$$

Expressing the negative fraction as the sum of negative integer and positive fraction we have,

$$(8 + 1/3) = 0x_1 + 0x_2 + (2 + 2/3)x_3 + 1S_1 + (-14 + 2/3)G_1$$

The corresponding fractional cut is given by,

$$-2/3x_3 - 2/3\ G_1 + G_2 = -1/3.$$

Add this second Gomorian constraint at the bottom of the above simplex table and apply dual simplex method.

Since $G_2 = -1/3$, hence G_2 leaves the basis. Also,

$$\text{Max}\left(\frac{Z_j - C_j}{a_{ij}},\ a_{ij} < 0\right) = \text{Max}\left(\frac{34/3}{-2/3}, \frac{40/3}{-2/3} -\right) = -17$$

This gives the non-basic variable x_3 which enters the basis. Using dual simplex method, introduce x_3 and drop G_2.

		C_j	2	20	–10	0	0	0
C_B	B	x_B	x_1	x_2	x_3	S_1	G_1	G_2
20	x_2	0	0	1	0	0	1	0
2	x_1	10/3	1	0	2/3	0	–10/3	0
0	S_1	25/3	0	0	8/3	1	–40/3	0
0	G_2	–1/3	0	0	–2/3	0	–2/3	1←
	Z_j	20/3	2	20	4/3	0	40/3	0
	$Z_j - C_j$		0	0	34/3↑	0	40/3	0
20	x_2	0	0	1	0	0	1	0
2	x_1	3	1	0	0	0	– 4	1
0	S_1	7	0	0	0	1	– 16	4
–10	x_3	1/2	0	0	1	0	1	–3/2
	Z_j	1	2	20	–10	0	2	17
	$Z_j - C_j$		0	0	0	0	2	17

Since the solution is still a non-integer, a third fractional cut is required. It is given from the source row (x_3 row) as,

$$\frac{1}{2} \le \left(-2 + \frac{1}{2}\right) G_2$$

Or, $$-\frac{1}{2} \ge -\frac{1}{2}\, G_2$$

NOTES

Or, $$-1/2 = -1/2\; G_2 + G_3$$

Insert this additional constraint at the bottom of the table. The modified simplex table is shown below.

Using dual simplex method, we drop G_3 and introduce G_2.

		C_j	2	20	–10	0	0	0	0
C_B	B	x_B	x_1	x_2	x_3	S_1	G_1	G_2	G_3
20	x_2	0	0	1	0	0	1	0	0
2	x_1	3	1	0	0	0	–4	1	0
0	S_1	7	0	0	0	1	–16	4	0
–10	x_3	1/2	0	0	1	0	1	–3/2	0
0	G_3	–1/2	0	0	0	0	0	–1/2	1←
	Z_j	1	2	20	–10	0	2	17	0
	$Z_j - C_j$		0	0	0	0	2	17↑	0
20	x_2	0	0	1	0	0	1	0	0
2	x_1	2	1	0	0	0	–4	0	2
0	S_1	3	0	0	0	1	–16	0	8
–10	x_3	2	0	0	1	0	1	0	–3
0	G_2	1	0	0	0	0	0	1	–2
	Z_j	–16	2	20	–10	0	2	0	34
	$Z_j - C_j$		0	0	0	0	2	0	34

Since all $Z_j - C_j \ge 0$ and also the variables are integers, the optimum integer solution is obtained and given by $x_1 = 2$, $x_2 = 0$, $x_3 = 2$ and Max $Z = -16$.

Example 4.4: Solve the integer programming problem.

Max $Z = 7x_1 + 9x_2$

Subject to constraints, $-x_1 + 3x_2 \le 6$

$7x_1 + x_2 \le 35$

$x_1, x_2 \ge 0$ are integers.

Solution: Now ignoring the integer conditions, and introducing slack variables $S_1, S_2 \ge 0$, we get the standard form of IPP as,

Max $Z = 7x_1 + 9x_2 + 0S_1 + 0S_2$

Subject to constraints, $-x_1 + 3x_2 + S_1 = 6$

$7x_1 + x_2 + S_2 = 35$

$x_1, x_2, S_1, S_2 \ge 0$

The given IPP is solved using simplex method.

		C_j	7	9	0	0	
C_B	B	x_B	x_1	x_2	S_1	S_2	$Min \frac{x_B}{x_i}$
0	S_1	6	–1	3	1	0	6/3 = 2←
0	S_2	35	7	1	0	1	35/1 = 35
	Z_j	0	0	0	0	0	
	$Z_j - C_j$		–7	– 9↑	0	0	
9	x_2	2	–1/3	1	1/3	0	–
0	S_2	33	22/3	0	–1/3	1	$33\times\frac{3}{22}$ ←
	Z_j	18	–3	9	3	0	
	$Z_j - C_j$		–10↑	0	3	0	
9	x_2	7/2	0	1	7/22	1/22	
7	x_1	9/2	1	0	–1/22	3/22	
	Z_j	63	7	9	28/11	15/11	
	$Z_j - C_j$		0	0	28/11	15/11	

NOTES

Since all $Z_j - C_j \geq 0$, an optimum solution is obtained as,

$$x_1 = \frac{9}{2},\ x_2 = \frac{7}{2} \text{ and Max } Z = 63.$$

Since the optimum solution obtained above is not an integer solution, we select a constraint corresponding to,

$$x_{B1} = \frac{7}{2} = 3 + \frac{1}{2},\ x_{B2} = \frac{9}{2} = 4 + \frac{1}{2}$$

$$\text{Max }(f_i) = \text{Max }(f_1, f_2)$$

$$= \text{Max}\left\{\frac{1}{2}, \frac{1}{2}\right\}$$

Since both the equations have the same value of f_i, either one of the two equations can be used. Let us consider the x_2 row as source row.

From x_2 row we have,

$$3 + \frac{1}{2} = 0x_1 + x_2 + \frac{7}{22}S_1 + \frac{1}{22}S_2$$

There is no negative fraction.

The Gomorian constraint is given by,

$$\frac{7}{22}S_1 + \frac{1}{22}S_2 \geq \frac{1}{2}$$

i.e.,
$$-\frac{7}{22}S_1 - \frac{1}{22}S_2 \leq -\frac{1}{2}$$

$\Rightarrow$
$$-\frac{7}{22}S_1 - \frac{1}{22}S_2 + G_1 = -\frac{1}{2}$$

NOTES

Here, G_1 is the Gomorian slack. Adding this new constraint at the bottom of the above optimal simplex table, we have the new table.

We apply dual simplex method. Since $G_1 = -1/2$, hence G_1 leaves the basis. Also,

$$\text{Max}\left\{\frac{Z_j - C_j}{a_{ij}}, a_{ij} < 0\right\} = \text{Max}\left\{\frac{\frac{28}{11}}{-\frac{7}{22}}, \frac{\frac{15}{11}}{-\frac{1}{22}}\right\}$$

$$= \text{Max } (-8, -30) = -8$$

This gives the non-basic variable S_1 to enter into the basis.

Applying dual simplex method drop G_1 and introduce S_1.

		C_j	7	9	0	0	0
C_B	B	x_B	x_1	x_2	S_1	S_2	G_1
9	x_2	7/2	0	1	7/22	1/22	0
7	x_1	9/2	1	0	–1/22	3/22	0
0	G_1	–1/2	0	0	–7/22	–1/22	1←
	Z_j	63	7	9	28/11	15/11	0
	$Z_j - C_j$		0	0	28/11↑	15/11	0
9	x_2	3	0	1	0	0	1
7	x_1	32/7	1	0	0	1/7	–1/7
0	S_1	11/7	0	0	1	+1/7	–22/7
	Z_j	59	7	9	0	1	8
	$Z_j - C_j$		0	0	0	1	8

The optimal solution obtained by dual simplex method as above is still a non-integer. Thus a new Gomory's constraint is to be reconsidered.

Here, $x_2 = 3$, $x_1 = \frac{32}{7} = 4 + \frac{4}{7}$ and $S_1 = \frac{11}{7} = 1 + \frac{4}{7}$

$$\therefore \qquad \text{Max } (f_i) = \text{Max}\left\{-, \frac{4}{7}, \frac{4}{7}\right\} = \frac{4}{7}$$

Choose the x_1 row as source row arbitrarily as both the fraction values are the same. From the source row we have,

$$\frac{4}{7} = 1x_1 + 0x_2 + 0S_1 + \frac{1}{7}S_2 + \frac{6}{7}G_1$$

There is no negative fraction in the source row.

The Gomory's constraint is given by,

$$\frac{1}{7}S_2 + \frac{6}{7}G_1 \geq \frac{4}{7}, \quad \text{i.e.,} \quad -\frac{1}{7}S_2 - \frac{6}{7}G_1 + G_2 = -\frac{4}{7}$$

Here, G_2 is the Gomorian slack. Adding this constraint in the above simplex table, we get a modified table.

We again apply the dual simplex method.

Since $G_2 = -\frac{4}{7}$, hence G_2 leaves the basis. Also,

$$\text{Max}\left\{\frac{Z_j - C_j}{a_{ij}},\ a_{ij} < 0\right\} = \text{Max}\left\{\frac{1}{-\frac{1}{7}}, \frac{8}{-\frac{6}{7}}\right\}$$

$$= \text{Max}\left(-7, -\frac{28}{3}\right) = -7$$

This gives the non-basic variable S_2 to enter into the basis.

		C_j	7	9	0	0	0	0
C_B	B	x_B	x_1	x_2	S_1	S_2	G_1	G_2
9	x_2	3	0	1	0	0	1	0
7	x_1	32/7	1	0	0	1/7	–1/7	0
0	S_1	11/7	0	0	1	1/7	–22/7	0←
0	G_2	–4/7	0	0	0	–1/7	–6/7	1
	Z_j	59	7	9	0	1	8	0
	$Z_j - C_j$		0	0	0	1↑	8	0
9	x_2	3	0	1	0	0	1	0
7	x_1	4	1	0	0	0	–1	1
0	S_1	1	0	0	1	0	–4	1
0	S_2	4	0	0	0	1	6	–7
	Z_j	55	7	9	0	0	2	7
	$Z_j - C_j$		0	0	0	0	2	7

Since all $Z_j - C_j \geq 0$ and also the solution is an integer, we obtain an optimum integer solution given by $x_1 = 4, x_2 = 3$ and Max $Z = 55$.

Mixed Integer Programming Problem

In mixed IPP only some of the variables are restricted to integer values, while the other variables may take integer or other real values.

Mixed Integer Cutting Plane Procedure: The iterative procedure for the solution of mixed integer programming problem is as follows.

Step 1: Reformulate the given IPP into a standard maximization LPP form and then determine an optimum solution using simplex method.

Step 2: Test the integrality of the optimum solution.

(*i*) If all $x_{Bi} \geq 0$ ($i = 1, 2,... m$) and are integers, then the current solution is an optimum one.

(*ii*) If all $x_{Bi} \geq 0$ ($i = 1, 2,... m$) but the integer restricted variables are not integers, then go to the next step.

Step 3: Choose the largest fraction among those x_{Bi}, which are restricted to integers. Let it be $x_{Bk} = f_k$ (assume),

Step 4: Find the fractional cut constraints from the source row, namely *k*th row.

From the source row,

$$\sum_{j=1}^{n} a_{kj}\, k_j = x_{Bk}$$

i.e., $$\sum_{j=1}^{n}\left(\left[a_{kj}\right]+f_{ki}\right)r_j = [x_{Bk}] + f_k$$

NOTES

In the form $$\sum_{j=1}^{n} f_{ki}\, x_j \geq f_k$$

i.e., $$\sum_{j\in j^+} f_{kj}\, x_j + \left(\frac{f_k}{f_{k-1}}\right)\sum_{j\in j^-} f_{kj}\, x_j \geq f_k$$

$$-\sum_{j\in j^+} f_{kj}\, x_j - \left(\frac{f_k}{f_{k-1}}\right)\sum_{j\in j^-} f_{kj}\, x_j \leq -f_k$$

$$-\sum_{j\in j^+} f_{kj}\, x_j - \left(\frac{f_k}{f_{k-1}}\right)\sum_{j\in j^-} f_{kj}\, x_j + G_k = -f_k$$

Here, G_k is Gomorian slack.

$$j^+ = [\, j/f_{kj} \geq 0\,]$$

$$j^- = [\, j/f_{kj} < 0\,]$$

Step 5: Add this cutting plane generated in Step (4) at the bottom of the optimum simplex table obtained in Step (1). Find the new optimum solution using dual simplex method.

Step 6: Go to Step (2) and repeat the procedure until all $x_{Bi} \geq 0$ ($i = 1, 2,\ldots m$) and all restricted variables are integers.

Example 4.5: Find the optimum integer solution of the following IPP.

Max $Z = x_1 + x_2$

Subject to constraints, $3x_1 + 2x_2 \leq 5$

$x_2 \leq 2$

$x_1 + x_2 \geq 0$ and x_1 is an integer.

Solution: Introducing slack variables $S_1, S_2 \geq 0$ the standard form of IPP becames,

Max $Z = x_1 + x_2 + 0S_1 + 0S_2$

Subject to constraints, $3x_1 + 2x_2 + S_1 = 5$

$x_2 + S_2 = 2$

$x_1, x_2, S_1, S_2 \geq 0$

Initial basic feasible solution,

$$S_1 = 5, S_2 = 2$$

Ignore the integer condition and solve the problem using simplex method to obtain optimum solution.

		C_j	1	1	0	0	
C_B	B	x_B	x_1	x_2	S_1	S_2	*Min* $\frac{x_B}{x_i}$
0	S_1	5	3	2	1	0	5/3←
0	S_2	2	0	1	0	1	—

	Z_j	0	0	0	0	0	
	$Z_j - C_j$		–1↑	–1	0	0	
1	x_1	5/3	1	2/3	1/3	0	5/2
0	S_2	2	0	1	0	1	5/2←
	Z_j	5/3	1	2/3	1/3	0	
	$Z_j - C_j$		0	–1/3↑	1/3	0	
1	x_1	1/3	1	0	1/3	–2/3	
1	x_2	2	0	1	0	1	
	Z_j	7/3	1	1	1/3	1/3	
	$Z_j - C_j$		0	0	1/3	1/3	

Since all $Z_j - C_j \geq 0$, the current basic feasible solution is optimum. But x_1 is non-integer. From the source row (first row) we have,

$$1/3 = x_1 + 0\ x_2 + 1/3\ S_1 - 2/3\ S_2$$

The Gomorian constraint is given by,

$$\frac{1}{3}S_1 + \left(\frac{\frac{1}{3}}{\frac{1}{3}-1}\right)\left(\frac{-2}{3}\right)S_2 \geq \frac{1}{3}$$

$$\frac{1}{3}S_1 + \frac{1}{3}S_2 \geq \frac{1}{3} \quad \Rightarrow \quad -\frac{1}{3}S_1 - \frac{1}{3}S_2 \leq -\frac{1}{3}$$

$$\frac{-1}{3}S_1 - \frac{1}{3}S_2 + G_1 = -\frac{1}{3}$$

Hence, G_1 is the Gomorian slack.

Adding this Gomorian constraint at the bottom of the above simplex table, we have the following equation unsing the dual simplex method. Since hence $G_1 = -1/3 < 0$, G_1 leaves the basis. Also,

$$\text{Max}\left\{\frac{Z_j - C_j}{a_{ij}}, a_{ij} < 0\right\}$$

$$\text{Max}\left\{\frac{\frac{1}{3}}{\frac{-1}{3}}, \frac{\frac{1}{3}}{\frac{-1}{3}}\right\} = \text{Max}\ (-1, -1) = -1$$

As this corresponds to both S_1 and S_2, we choose S_1 arbitrarily as the entering variable.

Drop G_1 and introduce S_1.

		C_j	1	1	0	0	0
C_B	B	x_B	x_1	x_2	S_1	S_2	G_1
1	x_1	1/3	1	0	1/3	–2/3	0
1	x_2	2	0	1	0	1	0
0	G_1	–1/3	0	0	–1/3	–1/3	1←
	Z_j	7/3	1	1	1/3	1/3	0
	$Z_j - C_j$		0	0	1/3↑	1/3	0

1	x_1	0	1	0	0	– 1	1
1	x_2	2	0	1	0	1	0
0	S_2	1	0	0	1	1	– 3
	Z_j	2	1	1	0	0	1
	$Z_j - C_j$		0	0	0	0	1

Since all $Z_j - C_j \geq 0$ and all $x_{Bi} \geq 0$, the current solution is feasible and optimal.

The required optimal integer solution is given by,

$$x_1 = 0, x_2 = 2 \text{ and Max } Z = 2.$$

Example 4.6: Find the optimum integer solution of the given IPP.

Max $Z = 4x_1 + 6x_2 + 2x_3$

Subject to constraints,
$$4x_1 - 4x_2 \leq 5$$
$$-x_1 + 6x_2 \leq 5$$
$$-x_1 + x_2 + x_3 \leq 5$$
$$x_1, x_2, x_3 \geq 0 \text{ and } x_1, x_3 \text{ are integers.}$$

Solution: Introducing slack variables $S_1, S_2, S_3 \geq 0$ the standard form of IPP becomes,

Max $Z = 4x_1 + 6x_2 + 2x_3 + 0S_1 + 0S_2 + 0S_3$

Subject to constraints,
$$4x_1 - 4x_2 + S_1 = 5$$
$$-x_1 + 6x_2 + S_2 = 5$$
$$-x_1 + x_2 + x_3 + S_3 = 5$$

The initial basic feasible solution is given by $S_1 = 5$, $S_2 = 5$ and $S_3 = 5$. Ignoring the integer condition, the optimum solution of given IPP is obtained by the simplex method.

		C_j	4	6	2	0	0	0	
C_B	B	x_B	x_1	x_2	x_3	S_1	S_2	S_3	***Min*** x_B/x_i
0	S_1	5	4	–4	0	1	0	0	—
0	S_2	5	–1	6	0	0	1	0	5/6←
0	S_3	5	–1	1	1	0	0	1	5/1
	Z_j	0	0	0	0	0	0	0	
	$Z_j - C_j$		–4	–6↑	–2	0	0	0	
0	S_1	25/3	10/3	0	0	1	2/3	0	25/10←
6	x_2	5/6	–1/6	1	0	0	1/6	0	—
0	S_3	25/6	–5/6	0	1	0	–1/6	1	—
	Z_j	5	–1	6	0	0	1	0	
	$Z_j - C_j$		–5↑	0	–2	0	1	0	
4	x_1	5/2	1	0	0	3/10	1/5	0	—
6	x_2	5/4	0	1	0	1/20	1/5	0	
0	S_3	25/4	0	0	1	1/4	0	1	25 4/1←
	Z_j	35/2	4	6	0	3/2	2	0	
	$Z_j - C_j$		0	0	–2↑	3/2	2	0	

4	x_1	5/2	1	0	0	3/10	1/5	0
6	x_2	5/4	0	1	0	1/20	1/5	0
2	x_3	25/4	0	0	1	1/4	0	1
	Z_j	30	4	6	2	2	2	0
	$Z_j - C_j$		0	0	0	2	2	0

Since all $Z_j - C_j \geq 0$, the solution is optimum. But the integer constrained variables x_1 and x_3 are non-integer.

$$\therefore \qquad x_1 = 5/2 = 2 + 1/2, \; x_2 = \frac{5}{4} = 1 + \frac{1}{4} \text{ and}$$

$$x_3 = 25/4 = 6 + 1/4$$

Max (f_1, f_2, f_3) = Max (1/2, 1/4, 1/4) = 1/2

From the first row we have,

$$(2 + 1/2\ ) = x_1 + 0x_2 + 0x_3 + (3/10)\ S_1 + (1/5)\ S_2$$

The Gomorian constraint is given by,

$$3/10\ S_1 + 1/5\ S_2 \geq 1/2$$

$$-3/10\ S_1 - 1/5\ S_2 \leq -1/2$$

i.e., $-3/10\ S_1 - 1/5\ S_2 + G_1 = -\ 1/2$, where G_1 is the Gomorian slack. Introduce this new constraint at the bottom of the above simplex table using dual simplex method since $G_1 = -\ 1/2 < 0$, G_1 leaves the basis. Also,

$$\text{Max}\left\{\frac{Z_j - C_j}{a_{ij}}, a_{ij} < 0\right\} = \text{Max}\left\{\frac{2}{\frac{-3}{10}}, \frac{2}{\frac{-1}{5}}\right\} = \text{Max}\left\{\frac{-20}{3}, -10\right\} = \frac{-20}{3}$$

This Gives the non-basic variable S_1, to enter into the basis.

		C_j	4	6	2	0	0	0	0
C_B	B	x_B	x_1	x_2	x_3	S_1	S_2	S_3	G_1
4	x_1	5/2	1	0	0	3/10	1/5	0	0
6	x_2	5/4	0	1	0	1/20	1/5	0	0
2	x_3	25/4	0	0	1	1/4	0	1	0
0	G_1	–1/2	0	0	0	–3/10	–1/5	0	1←
	Z_j	30	4	6	2	2	2	2	0
	$Z_j - C_j$		0	0	0	2↑	2	2	0
4	x_1	2	1	0	0	0	0	0	1
6	x_2	7/6	0	1	0	0	1/6	0	1/6
2	x_3	35/6	0	0	1	0	–1/6	1	5/6
0	S_1	5/3	0	0	0	1	2/3	0	–10/3
	Z_j	80/3	4	6	2	0	2/3	2	20/3
	$Z_j - C_j$		0	0	0	0	2/3	2	20/3

Since all $Z_j - C_j \geq 0$, the solution is optimum and also the integer restricted variable $x_2 = 7/6, x_3 = 35/6$, and $S_1 = 5/3$ is not an integer, therefore, we add another Gomorian constraint,

$\therefore \quad x_2 = 7/6 = 1 + 1/6, x_3 = 35/6 = 5 + 5/6$, and $S_1 = 5/3 = 1+1/3$

$$\text{Max } (f_1, f_2, f_3) = \text{Max } \left(\frac{1}{6}, \frac{5}{6}, \frac{1}{3}\right) = 5/6$$

Therefore, the source row is the third row.

From this row we have,

$$5 + \frac{5}{6} = 0x_1 + 0x_2 + x_3 + 0S_1 - \frac{1}{6}S_2 + S_3 + \frac{5}{6}G_1$$

The Gomorian constraint is given by,

$$\left(\frac{\frac{5}{6}}{\frac{5}{6} - 1}\right)\left(\frac{-1}{6}\right)S_2 + \frac{5}{6}G_1 \geq \frac{5}{6}$$

$$\Rightarrow \qquad \frac{5}{6}S_2 + \frac{5}{6}G_1 \geq \frac{5}{6}$$

$$\Rightarrow \qquad \frac{-5}{6}S_2 - \frac{5}{6}G_1 + G_2 \leq \frac{-5}{6}$$

Here, G_2 is the Gomorian slack.

Add this second cutting plane constraint at the bottom of the above optimum simplex table.

Use dual simplex method. Because $G_2 = -5/6 < 0$, hence G_2 leaves the basis.

Also, $\text{Max}\left\{\frac{Z_j - C_j}{a_{ij}}\ a_{ij} < 0\right\} = \text{Max}\left\{\frac{\frac{2}{3}}{\frac{-5}{6}}, \frac{\frac{20}{3}}{\frac{-5}{6}}\right\} = \text{Max}\left\{\frac{-4}{5}, -8\right\} = -\frac{4}{5},$

Here, corresponds to S_2.

Drop G_2 and introduce S_2.

		C_j	4	6	2	0	0	0	0	0
C_B	B	x_B	x_1	x_2	x_3	S_1	S_2	S_3	G_1	G_2
4	x_1	2	1	0	0	0	0	0	1	0
6	x_2	7/6	0	1	0	0	1/6	0	1/6	0
2	x_3	35/6	0	0	1	0	–1/6	1	5/6	0
0	S_1	5/3	0	0	0	1	2/3	0	–10/3	0
0	G_2	–5/6	0	0	0	0	–5/6	0	–5/6	1←
	Z_j	80/3	4	6	2	2/3	2/3	2	20/3	0
	$Z_j - C_j$		0	0	0	2/3	↑2/3	2	20/3	0
4	x_1	2	1	0	0	0	0	0	1	0
6	x_2	1	0	1	0	0	0	0	0	1/5
2	x_3	6	0	0	1	0	0	1	1	–1/5
0	S_1	1	0	0	0	1	0	0	–4	4/5
0	S_2	1	0	0	0	0	1	0	1	–6/5
	Z_j	26	4	6	2	0	0	2	6	4/5
	$Z_j - C_j$		0	0	0	0	0	2	6	4/5

Since all $Z_j - C_j \geq 0$ and also all the restricted variables x_1, x_2 and x_3 are integers, an optimum integer solution is obtained.

The optimum integer solution is,

$x_1 = 2, x_2 = 1, x_3 = 6$ and Max $Z = 26$.

4.2.2 Branch and Bound Method

This method is applicable to both, pure as well as mixed IPP. Some times a few or all the variables of an IPP are constrained by their upper or lower bounds. The most general method for the solution of such constrained optimization problems is called 'Branch and Bound Method'.

This method first divides the feasible region into smaller subsets and then examines each of them successively until a feasible solution that gives an optimal value of objective function is obtained.

Let the given IPP be,

Max $Z = CX$

Subject to constraints, $AX \leq b$

$X \geq 0$ and are integers.

In this method, we first solve the problem by ignoring the integrality condition.

(*i*) If the solution is in integers, the current solution is optimum for the given IPP.

(*ii*) If the solution is not in integers, say one of the variable X_r is not an integer, then $x_r^* < x_r < x^*_{r+1}$ where x^*_r, x^*_{r+1} are consecutive non-negative integers.

Hence, any feasible integer value of x_r must satisfy one of the two conditions.

$$x_r \leq x_r^* \text{ or } x_r \geq x^*_{r+1}$$

These two conditions are mutually exclusive (both cannot be true simultaneously). By adding these two conditions separately to the given IPP, we form different sub-problems.

Sub-problem 1		Sub-problem 2	
Max $Z = CX$		Max $Z = CX$	
Subject to constraints,	$AX \leq b$	Subject to constraints,	$AX \leq b$
	$x_r \leq x^*_r$		$x_r \geq x^*_{r+1}$
	$x \geq 0.$		$x \geq 0.$

Thus, we have branched or partitioned the original problem into two sub-problems. Each of these sub-problems is then solved separately as IPP.

If any sub-problem yields an optimum integer solution, it is not further branched. But if any sub-problem yields a non-integer solution, it is further branched into two sub-problems. This branching process is continued until each problem terminates with either an integer optimal solution or there is an evidence that it cannot yield a better solution. The integer-valued solution among all the sub-problems, which gives the most optimal value of the objective function is then selected as the optimum solution.

Note: For minimization problem, the procedure is the same except that upper bounds are used. The sub-problem is said to be fathomed and is dropped from further consideration if it yields a value of the objective function lower than that of the best available integer solution and it is useless to explore the problem any further.

4.3 SIMULATION PROBLEMS

Monte Carlo simulation is often used by modern management when it cannot use other techniques. There are many industrial problems which defy mathematical solutions. The reason is that either they are too complicated or that the data cannot be expressed in mathematical terms. In such cases, it is still possible to reach valid conclusions by using the Monte Carlo technique. A considerable help is thus obtained at practically no cost in taking decisions concerning the functioning of a business system. The data and conclusions can be obtained through simulation of an actual operation on the basis of its own past working. It paves the way for predicting the changes in its behaviour and the result is evaluated from innovations that we want to introduce.

By using a fresh series of random numbers at the appropriate junctures we can also examine the reactions of the simulated model just as if the same alterations had actually been made in the system itself. Monte Carlo simulation, therefore, provides a tool of knowing in advance whether or not the expense to be incurred or the investment to be made in making the changes envisaged. Through this technique, you can introduce the innovations on a piece of paper, examine their effects and then may decide to adopt or not to adopt such innovations in the functioning of real system. The usefulness of simulation lies in the fact that it allows us to experiment with a model of the system rather than the actual system; in case we are convinced about the results of our experiments we can put the same into practice. Thus the effect of the actual decisions are tested in advance through the technique of simulation by resorting to the study of the model representing the real life situation or the system.

The main purpose of simulation in management is to provide feedback, which is vital for the learning process. It creates an atmosphere in which managers play a dynamic role by enriching their experience through involvement in reckoning with actual conditions through experimentation on paper. The technique permits trying out several alternatives as the entire production for service process can be worked out on paper, without dislocating the system in any way. Thus, Monte Carlo technique transforms the manager from a blind-folded driver of an automobile, reacting to instructions of a fellow passenger to one who can see fairly, clearly, where he is going.

Introduction to Simulation

Simulation is a representation of reality through the use of a model or other device which will react in the same manner as reality under a given set of conditions.

Simulation is also defined as the use of a system model that has the designed characteristics of reality in order to produce the essence of actual operation.

Types of Simulation

Simulation is mainly of the following two types:

(i) Analog (Environmental) Simulation

(ii) Computer (System) Simulation or Digital Simulation

Some examples of simulation models are given as follows:

(i) Testing an aircraft model in a wind tunnel

(ii) Children cycling park with various signals and crossing—to model a traffic system

(iii) Planetorium

To determine the behaviour of a real system in true environments a number of experiments are performed on simulated models either in the laboratories or in the computer itself.

Table 4.1 is an example of simulation worksheet.

Table 4.1 *Simulation Worksheet for Simulating Sizes to Locate the Number of Misfits*

Assembly S.N.	*Shafts Random^1 Numbers (from Tippett tables)*	*Random^2 Normal Deviate (z)*	*Simulating^3 Size $X = \mu+z(\sigma)$*	*Rings Random Numbers (from Tippett tables)*	*Random Normal Deviate (z)*	*Simulating Size $X = \mu+z(\sigma)$*
1	2952	0.82	0.980 + 0.82(0.01) = 0.9882	3992	1.28	1.0 + 1.28(0.02) = 1.0256
2	3170	0.91	0.980 + 0.91(0.01) = 0.9891	4167	1.38	1.0 + 1.38(0.02) = 1.0276
3	7203	– 0.59	0.980 – 0.59(0.01) = 0.9741	1300	0.33	1.0 + 0.33(0.02) = 1.0066
4	3408	1.00	0.980 + 1.0(0.01) = 0.9900	3563	1.06	1.0 + 1.06(0.02) = 1.0212
5	0560	0.14	0.980 + 0.14(0.01) = 0.981.4	1112	0.28	1.0 + 0.28(0.02) = 1.0056
6	6641	– 0.42	0.980 – 0.42(0.01) = 0.9758	9792	– 2.04	1.0 – 2.04(0.02) = 0.9592
7	5624	– 0.16	0.980 – 0.16(0.01) = 0.9784	9525	– 1.67	1.0 – 1.67(0.02) = 0.9666
8	5356	– 0.09	0.980 – 0.09(0.01) = 0.9791	2693	0.74	1.0 + 0.74(0.02) = 1.0148
9	2769	0.76	0.980 + 0.76(0.01) = 0.9876	6107	– 0.28	1.0 – 28(0.02) = 0.9944
10	5246	– 0.06	0.980 – 0.06(0.01) = 0.9794	9025	– 1.29	1.0 – 1.29(0.02) = 0.9742

Random Variable

The random variable is a real valued function defined over a sample space associated with the outcome of a conceptual chance experiment. Random variables are classified according to their probability density function.

(i) ***Random Number:*** It refers to a uniform random variable or a numerical value assigned to a random variable following uniform probability density function. In other words, it is a number in a sequence of numbers whose probability of occurence is the same as that of any other number in that sequence.

(ii) ***Pseudorandom Numbers:*** Random numbers are called Pseudorandom numbers when they are generated by some deterministic process but have already qualified the predetermined statistical test for randomness.

Areas of Application of Monte Carlo Simulation

Monte Carlo simulation has been applied to a wide diversity of problems ranging from queuing process, inventory problem, risk analysis concerning a major capital investment such as the introduction of a new product, expansion of the capacity, and many other problems. Budgeting is another area where simulation can be very useful. In fact, the system of flexible budgeting is an exercise in simulation. Simulation can as well be used for preparing the master budget through functional budgets.

Over and above, the greatest contribution of simulation is in the analysis of complex systems. Many real-world problems involve systems made up of many components parts that are interrelated. The system may be dynamic and changing over time and may involve probabilistic or uncertain events. Simulation is the only technique for quantitative analysis of such problems.

Monte Carlo Method

Monte Carlo methods are basically the algorithms used in the computation of result to be calculated from repeated random sampling. These methods help in computerized calculations because these can perform repeated computation using random or pseudo-random numbers. It is also used when it is not feasible to compute correct result with a deterministic algorithm. Monte Carlo simulation methods are used to study systems having degrees of freedom and in the situations when there is significant ambiguity in inputs for example, calculating risk factor in a business.

Various simulation models, based on the principle of similitude (such as model of aeroplanes initiating flight conditions in a wind tunnel) have been in use for a long time. However, Monte Carlo simulation is a recent operations research innovation. The novelty lies in making use of pure chance to contact a simulated version of the process under analysis, in exactly the same way as pure chance operates the original system under working conditions. Only models under uncertainty can be evaluated using Monte Carlo technique.

'Monte Carlo' is the code name given by John von Neumann and S.M. Ulam to the technique of solving problems though it is too expensive for experimental solutions and too complicated for analytical treatment.

Monte Carlo method is not one single method. It involves various widely-used classes of approaches to follow a specific model. Using it the following can be done:

- Define a domain with feasible inputs.
- Randomly generate inputs from the domain.

- Perform deterministic computation with the inputs.
- Combine the results of the personal computations into the final result.

NOTES

Monte Carlo methods are used to solve various mathematical problems based on sampling experiments in statistics using the sequences of random numbers for simulation and are termed as statistical simulation methods. Thus, Monte Carlo method is not one single method, but it is a collection of various methods and is basically used to perform similar procedure. Some of these methods are discussed here with the help of solved examples.

The Monte Carlo simulation technique can as well be used to solve probabilistic problems. Suppose, we are to evaluate the probability *P* that a tank will be knocked out by either a first or second shot from an antitank gun assumed to posses a constant kill probability of 1/2. The probability analysis will say that the chance of tank being knocked out by either a first or second shot from an antitank gun is 1/2+1/2 (1–1/2) =3/4. However, we can also work out this probability by simulating each round of the antitank gun by the flip of a coin through Monte Carlo simulation technique.

Since the probability of a 'Head' is the same as that of a kill, we may call it a hit when the coin turns up a head and otherwise a miss. If we flip the coin a large number of times, the value of *P* may be calculated by merely counting the number of times a head turns up at least in two successive throws and then dividing this number by the total pairs of throws of the coin. Monte Carlo method in this simple case is indeed a poor substitute for the theoretical probability analysis. However, many real-life systems are so complicated that even the well defined probability analysis very often fails but such situations can be handled by Monte Carlo simulation, particularly the Monte Carlo technique that provides the simplest possible solutions for queuing problems. Problems of corporate planning, inventory control, capital investment, consumer behaviour and quality control can also be handled through simulation.

Monte Carlo simulation uses random number tables to reproduce on paper the operation of any given system under its own working conditions. This technique is used to solve problems that depend upon probability where formulation of mathematical model is not possible. It involves first, the determining of the probability distribution of the concerned variables and then sampling from this distribution by means of random numbers to obtain data. It may, however, be emphasized here that the probability distributions to be used should closely resemble the real world situation.

One should always remember that simulation is not a perfect substitute but rather an alternative procedure for evaluating a model. Analytical solution produces the optimal answer to a given problem, while Monte Carlo simulation yields a solution which should be very close to the optimal but not necessarily the exact correct solution. Monte Carlo Simulation solution converges to the optimal solution as the number of simulated trials goes to infinity.

NOTES

Check Your Progress

1. Define integer programming.
2. State the importance of integer programming problems.
3. State the applications of integer programming.
4. What is Gomory's cutting method
5. What is branch and bound method?
6. Define the mixed integer programming problem.
7. Explain random number.
8. What is pseudorandom number?
9. Define simulation. Why is it used?
10. State the Monte Carlo technique.

4.4 REPLACEMENT PROBLEMS

The replacement of certain items is to be done usually for one of the following four reasons:

(i) Due to new developments, the current equipment has become technologically obsolete.

(ii) The current equipment has become unusable, i.e., it has failed and does not work at all. For instance, the electric light bulb has failed and as such must be replaced. This is a case of sudden failure, but the complete failure of an item like a machine may be a gradual one.

(iii) The current equipment has deteriorated on account of its long use over time and as such does not function efficiently. In other words, it requires expensive maintenance.

(iv) The current equipment is expected to fail shortly, the probability of which increases over time. As such it is considered economically advantageous to replace equipment in anticipation of costly failure. Such replacement is later known as 'Preventive Replacement'.

The above stated reasons indicate that the need for replacement arises in different situations and hence, different replacement strategies have to be evolved to suit the concerning situations. At times we may wait to replace an item till it fails, but in some other situations we may replace an item in anticipation of costly failure. Then we may as well decide to replace an item by the same item or by a different type of item. In case of deteriorating item, the problem is to determine at which time it is profitable to replace the item. Whatever the case may be, the ultimate objective is to decide the best replacement strategy to enable the concerned business unit to maximize its profit.

Replacement Models

Replacement models generally fall into two categories depending upon the life pattern of the equipment under study. There are models which deal in replacing equipments that deteriorate with age or usage. Choice between an existing asset and its potential replacement is based on analysis directed to reduce the differences in future receipts and disbursements, to an equivalent basis for comparing. Then there are models which help in establishing replacement policy or strategy for those equipment which break down or fail completely or are expected to fail shortly. Such models require the use of probabilistic concepts and the failure data. In addition to these two categories, one can as well conceive of models of replacement concerning those items that become out of date due to new technological developments, but such models are usually complicated models and as such beyond our scope. We only describe in the pages that follow the replacement models in respect of items (i) That deteriorate with time and (ii) That breakdown completely or are expected to breakdown shortly.

Replacement of a Deteriorating Item

Most of the items particularly machines, equipments, etc., deteriorate with time resulting in the following:

(i) Increased expenditures for operating costs, i.e., costs involved in running the equipment along with repairs and other maintenance costs concerning the said equipment.

(ii) Decreases in the productivity of the equipment which can as well be treated as a cost. For instance, a certain machine produces 24,000 items per year when new, but only 20,000 items in its second year of use. If the item sells for ₹ 2 and the material and sales costs amount to ₹ 1.50, then the cost of productivity loss is 4000 × ₹ 0.50 = ₹ 2000. Such costs tend to increase over time at an increasing rate.

(iii) Decreases in the value of the equipment, i.e., the resale or salvage value decreases (or what is altenatively known as increases in the real capital cost of the equipment, viz., depreciation overtime).

Thus, we have three types of relevant costs, viz., the operating costs, cost of productivity loss and the cost of capital in terms of depreciation. The first two types of costs can be grouped together since they possess the same functional characteristic, i.e., they tend to increase over time at an increasing rate, but the third type of cost in the form of average depreciation gradually diminishes over time. As a result, the average cost curve of the concerning equipment initially moves downward with time, reaching its lowest point, beyond which it moves upwards at an increasing speed. The time with reference to the lowest point in the average cost curve happens to be the optimal replacement time of the concerning equipment. All this can be shown graphically, illustrated in Figure 4.1.

NOTES

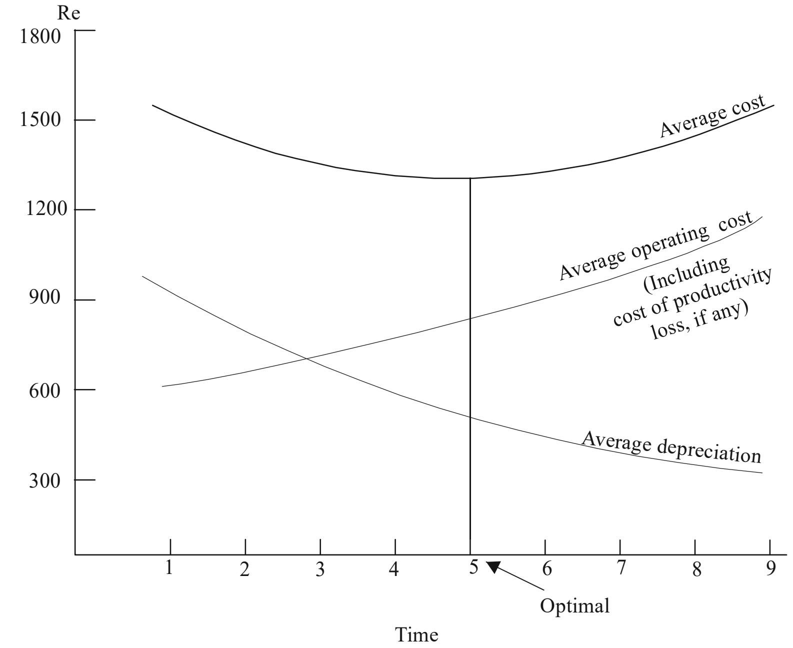


Fig. 4.1 *Optimal Replacement of Equipment that Deteriorates with Time*

The concerning replacement model can be developed as follows:

If, C = Capital cost of an equipment when new

$S(t)$ = The resale (or the salvage) value of the equipment after t years hence;

$f(t)$ = Operating or maintenance cost (including cost of productivity loss, if any) of the equipment at time t.

Hence, the total maintenance cost incurred on the equipment during n years

$$\int_0^n f(t)\,dt$$

Then the total cost (T) incurred on the equipment during n years can be worked out as follows:

$$T = C - S(t) + \int_0^n f(t)\,dt$$

and the average annual cost *(T_A)* incurred on the machine per year during n years is equal to:

$$T_A = \frac{1}{n}\left[C - (t) + \int_0^n f(t)dt\right]$$

Hence to determine the optimal period for replacing the equipment, we would find that value of n in the above equation which minimizes the value of T_t. For this purpose we must differentiate the expression for T_A with respect to *n,* i.e.; we should work out $\frac{\partial TA}{\partial_n}$ and then taking $\frac{\partial TA}{\partial_n} = 0$ we can find out the minimum possible value of T_A and then accordingly can state the value of n—the optimal period for replacing the equipment. The model stated above presumes the following:

(i) The capital equipment is needed for an indefinite period.

(ii) At the time of purchase the depreciation and maintenance costs can be accurately estimated.

(iii) The value of money remains the same during the period. It should as well be remembered that if changes take place in the repurchase cost of the equipment, then this fact can alter the optimal replacement period. For instance, an incentive such as the investment tax credit for new capital equipment has the effect of decreasing depreciation and thus reducing the replacement period whilst an increase in the price of the new equipment would have the reverse effect.

Let us take some concept examples to evolve optimal strategy of replacement of an item that deteriorates with age or usage.

Example 4.7: Suppose a certain manufacturing company uses a machine that costs ₹ 3000 when new. The following table gives the estimates of the expected operating costs per year and the salvage value of the machine.

Year	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Operating costs (₹)	600	700	800	900	1000	1200	1500	2000
Salvage value (₹)	2000	1333	1000	750	500	300	300	200

It is presumed that the machine is needed over an indefinite period in the future. Using the given information, determine the best age at which to replace the machine. If the optimum replacement policy is followed by the company, then what will be the average yearly cost of owning and operating the machine?

Solution: To solve this question, let us work out the average cost per year

Age at t or replacement (in Yrs)	1	2	3	4	5	*6*	*7*	*8*
$\int_0^n f(t)$ or Total operating Cost (₹)	600	1300	2100	3000	4000	5200	6700	8700
C – S (t) or Depreciation (₹)	1000	1667	2000	2250	2500	2700	2700	2800
$T = C - S(t) + \int_0^n f(t)\,dt$ or Total cost (₹)	1600	2967	4100	5250	6500	7900	9400	11503
T_A or average cost (₹)	1600	1483	1367	1312	1300	1317	1343	1438

The table shows that the lowest average cost per year is achieved by replacing the machine at the end of five years. As such the best age for the replacement of machine is at the end of every fifth year. If this optimum replacement policy is adopted by the company then the average yearly cost of owning and operating for this period is ₹ 1300.

NOTES

The above model (Example 4.7) assumes that we have use of the equipment indefinitely, the but if we requite the same only for definite time period, then we will have to make certain adjustments in our model. Particularly in working out the average cost, we will calculate, it over the total fixed (or definite) time period and then we will select the replacement period which minimizes the average cost over the time period the equipment is required. This is illustrated in Example 4.8.

Example 4.8: We again take the data as given in Example 4.7 and further presume that we shall need the machine only for the next 8 years. If we presently have a one-year old machine, then what should out replacement policy be?

Solution: First of all, we prepare the following table in which all costs are from age one on.

Age at replacement (in year)		*2*	*3*	*4*	*5*	*6*	*7*	*8*
Total operating costs	(₹)	700	1500	2400	3400	4600	6100	8100
Depreciation	(₹)	667	1000	1250	1500	1700	1700	1800
Total cost	(₹)	1367	2500	3650	4900	6300	7800	9900

In addition to the costs during the remaining life of our present machine, we shall as well have the cost of a new machine for the balance of the next 8 years. For instance, if we replace the present machine at age 5, we shall have to buy a new machine for the last 4 years at a cost of ₹ 5250 (refer the table). Thus over the 8 years the total cost would be ₹ 5250 + ₹. 4900 = ₹ 10150. If we examine and look into all other possible replacement ages of the present machine we obtain the following table.

Age of replacing present machine (in years)		*2*	*3*	*4*	*5*	*6*	*7*	*8*
Cost per present machine	(₹)	1367	2500	3650	4900	6300	7800	9900
Cost per new machine	(₹)	9400	7900	6500	5250	4100	2967	1600
Total cost	(₹)	10767	10400	10150	10150	10400	10767	11500

Thus we may replace our present machine either at age 4 or at age 5 because the total cost for either of these ages is the same and happens to be minimum possible and buy a new machine for the last 5 years (if present machine is replaced at age 4) or for the last 4 years (if present machine is replaced at age 5).

Discounting Costs Considerations

The analysis presented so far assumes that the value of money remains the same overtime. However, this may not always be the case. In real life situation, the value of money generally changes with time. If we presume that money does carry a rate of interest, say r per year, then a rupee invested now will be worth $(1+ r)$ a year hence, $(1+ r)^2$ two years hence, and $(1+ r)^{\mathrm{n}}$ in n years, time. This, in other words, means that if one has to make a payment of one rupee in n years' time, it is equivalent to a payment of $(l + r)^{-n}$ rupee today.

We now explain the determination of an optimal replacement policy assuming that the value of money does change overtime. For obtaining such a policy we shall assume first of all that the machine is replaced after every *n* years, taking $n = 1, 2, ...$ accordingly and we shall calculate the present worth of each period as follows:

Assume that the new machine costs *C* and that the salvage value at the end of year *n* is S_n and let the running cost in year *n* be R_B assumed to be payable at the start of the year and if we replaced the machine at the end of year *n* then the present value of all the costs (or total discounted costs including purchase price) is

$$= C - \left\{(S_n)(1+r)^{-n}\right\} + \sum_{n=1}^{n=\propto}(R_n)(1+r)^{-n+1}$$

and then dividing this by corresponding weight W_n (where $W_n = \sum_{n=1}^{n=\propto}(1+r)^{-n+1}$

we get $\overline{X}_w$, the weighted average cost for year *n* and then looking at $\overline{X}_w$'s for each of several *n* years beginning from year 1, 2.........∞ we select the lowest possible $\overline{X}_w$ and corresponding to it the year *n* is considered the optimal year when the machine should be replaced. Consider Example 4.3 to understand this better.

Example 4.9: A truck is priced at ₹ 60,000 and running costs are estimated ₹ 6,000 for each of the first four year, increasing by ₹ 2,000 per year in the 5th and subsequent years. It may be assumed that the running costs are payable at the start of the year. If money is worth 10 per cent per year, when should the truck be replaced? (Assume that the truck once purchased will be sold for scrap at a negligible price.)

Solution: Since the money is worth 10 per cent per year, the discount rate is given by $(1+10\%)^{-1} = 1/1+0.10 = 0.9091$. Using the table giving PWF values (given in Appendix at the end of the book) we tabulate figures say for 12 years *m* to workout the optimal replacement period for the given problem. The computation Table 4.1 clearly shows that the lowest possible $\overline{X}_w$ is attained at ₹ 17,99,823 in 9 years time. Accordingly, it is better to replace the truck after 9th year. We can alternatively arive at the same conclusion as under

In Table 4.2 we see that 16,000 < 17,998.23 < 18,000.

Now, since the running cost of 9th year is ₹ 16,000 and that of 10th year is ₹ 18,000 and since 17,998.23 < 18,000 (or 18,000 > 17,998.23) it is better to replace the truck after the 9th year.

- **Replacement of an Item that Break Down Completely or is Expected to Break Down Shortly**

Sometimes the item that deteriorates with age or usage does not result in increased operating costs or in decreased productivity. This fact applies, For instance, in case of an electric light bulb which either shines or does not shine. The effect of continued usage is simply to increase the probability that it will burn out. An appropriate policy for such items may be to replace them when they break down completely. Technically we can call it Individual Replacement policy under which an item is replaced immediately after its failure. However, in many cases the failure

Table 4.2 Computation of Optimal Replacement Policy

Year, n Let $n = k = 12$ for our purpose	1	2	3	4	5	6	7	8	9	10	11	12
Running Cost(Start of year, R_n p.w.f. $(1 + r)^n$	6000 1.000	6000 0.9091	6000 0.8264	6000 0.7513	8000 0.6830	10000 0.6209	12000 0.5645	14000 0.5132	16000 0.4665	18000 0.4241	20000 0.3855	22000 0.3505
Discounted running Costs, $(R_n)\ (1+ r)^{-n+1}$	6000	5455	4958	4508	5464	6209	6774	7185	7464	7634	7710	7711
Total discounted running Costs over years, $\sum_{N=1}^{n = k = 12} (R_n)\ (1+ r)^{-n+1}$	6000	11455	16413	20921	26385	32594	39368	46553	54017	61651	69361	77072
Salvage value Year end, S_n Discounted salvage Value, $(S_n)\ (1+ r)^{-n}$	— —	— —	— —	— —	— —	— —	— —	— —	— —	— —	— —	— —
Total discounted costs (including purchase price),= (Purchase price)– Discounted (salvage value + (Total discounted running Cost over years) Or $C - [(S_n)(1+ r)^{-n} + \sum_{N=1}^{n=k=12} (R_n)\ (1+ r)^{-n+1}$	66000	71455	76413	80921	86385	92594	99368	106553	114017	121651	129361	137072
Weights W_n $W_n = \sum_{N=1}^{n=k=12} (R_n)\ (1+ r)^{-n+1}$	1.0000	1.9091	2.7355	3.4868	4.1698	4.7907	5.3552	5.8684	6.3349	6.7590	7.1445	7.4950
Weighted average cost X_w = Row 8/Row 9	66000.00	37428.63	27933.83	23207.81	20716.82	19327.86	18555.42	18157.08	17998.23	17998.37	18106.38	18288.46

of the item may result in complete break down of the system resulting in substantial cost involved in loss in production idle labour, wastage and other damages apart from the actual cost of replacing the item. Take the case of an electrical equipment fitted in an aeroplane. The equipment may just cost a few hundred rupees to replace, but if it fails when the plane is in the air, then it may involve substantial costs, even many lives at times. Similarly, in industrial equipment, failures may cause loss of production and may result in damaged or faulty products. In some cases, failures may involve safety risks to workers. To avoid the costs of sudden failures, we try to estimate when such failures are likely to occur and try to replace the item before it actually fails. This sort of replacement is often described as Preventive Replacement, or replacement in anticipation of failure. Preventive replacement is based on the premiss that the probability of failing increases with age or usage and when such probability becomes high, it makes sense to replace the item during non-production hours before it fails to avoid the cost of failure.

The general model for an optimal preventive replacement policy can be developed as under: Let us take

C_R = Cost of replacement

C_F = Cost of failure (inclusive of the cost of replacement)

p_t = Probability that an item will fail in time period t, if it has not failed earlier

P_t = Probability that an item will not fail during or before time period t.

It is worked out as:

$P_t = (P_{t-1})\,(1-p_t)$

The optimal replacement policy is to replace the item every T time period if it has not failed earlier, so that expected cost per time period is minimized. As such we must find such replacement period. For this purpose we must first work out the expected cost for any selected replacement time period. If T happens to be our replacement period, then the maximum life of any item is T because it will be replaced in the said period irrespective of the fact that it has failed (in which case it will have to be replaced earlier) or not failed. Thus, the expected or average lift of the item, U_T would be worked out as under:

$$U_T = 1 + \sum_{t=1}^{T-1} P_t$$

If the expected life of the item is U_T under replacement period T, in any given fixed period of time, say k, we would use $\frac{k}{U_r}$ units of the said item. (Remember $\frac{k}{U_T}$ represents the numbers of the concerning item to be used in the overall given period in the long run.) Then we would, considering the required probability, expect $\left[(P_{T-1})\left(\dfrac{k}{U_T}\right)\right]$ units out of $\dfrac{k}{U_T}$ units of an item to be replaced

before failure and the remaining $\left[(1-P_{T-1})\left(\dfrac{k}{U_T}\right)\right]$ units to fail prior to replacement. Taking into consideration the C_R and C_F costs, the total expected cost for the period k would be as follows:

Total expected cost for the period k will be:

$$\left\{(P_{T-1})\left(\frac{k}{U_T}\right)(C_R)\right\}+\left\{1-P_{T-1}\left(\frac{k}{U_T}\right)(C_F)\right\}$$

Dividing the above by the number of periods, we obtain the average expected cost per period for replacement policy T_t

$$AC(T)=\frac{(P_{T-1})(C_R)+\left(1-P_{T-1}\right)(C_F)}{U_T}$$

Hence that value of T which minimizes the average cost as per the above equation is the optimal replacement policy.

The above equation gives us the expected average cost for a preventive replacement policy of time T. In order to find out the optimal replacement period T, we must calculate, with the help of the said equation, the average cost for each possible T and must select that T corresponding to which the expected average cost stands minimized.

To illustrate the stated procedure let us have a look at a few examples.

Example 4.10: A certain small engineering firm has made an intensive study concerning lathe bits. A bit costs ₹ 6 to replace but if it breaks down on a job, the production loss, is estimated to be ₹ 29. The data concerning the life of a bit in the past are given.

Jobs	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Probability of bits *breaking on job*	.01	.03	.09	.13	.25	35	.95

Using the given information state, after how many jobs should the engineering firm replace a bit that has not broken?

Solution: In the question we have been given the following:

(i) The cost of replacement of the item, C_R = ₹ 6

(ii) The cost of failure of an item, C_P = ₹35
(i.e., Production loss on failure plus replacement cost)

(iii) The probability that an item will fail in time period t or what we denote symbolically as p_t.

As we have to find the optimal preventive replacement policy, we develop the following table for the calculation of average expected cost for each possible replacement period T on the basis of the given information:

Jobs	*1*	*2*	*3*	*4*	*5*	*6*	*7*
P_t	0.01	0.03	0.09	0.13	0.25	0.55	0.95
P_7	0.990	0.960	0.874	0.760	0.570	0.257	0.013
U_T	1	1.990	2.950	3.824	4.584	5.154	5.411
$\{(P_{t-1})(C_R)+$ $(1-P_{T-1})(C_F)\}$	6	6.29	7.16	9.65	12.96	18.47	27.55
$AC(T)$	6	3.16	2.43	2.52	2.83	3.58	5.09

NOTES

Note: In the above table p_t has been taken as given in the question for different jobs:

P_t has been worked out as: $P_t = [(P_{t-1})(1-p_t)]$

U_T has been worked out as: $U_T = 1 + \sum_{t=1}^{T-1} P_t$

$AC(T)$ has been worked out as:

$$AC(T) = \frac{(P_{T-1})(C_R) + \left(1 - P_{T-1}\right)(C_F)}{U_T}$$

In the table we find that the minimum average expected cost is reached in time period (i.e, job) 3. Thus, the optimal replacement policy is to replace the lathe bit that has been in use for three jobs. In other words, the firm should replace a bit that has not broken after every three jobs.

Group Replacemests

Group replacement is often talked about in context of a system that contains a large number of identical low-cost items that are increasingly liable to failure with age. Under a group replacement policy, all the items in the system are replaced at worked out regular intervals irrespective of the fact that items have failed or have not failed, subject to the provision that if any item fails before the optimal replacement period, it may be individually replaced. If the failed items are replaced say at the end of the week of failure and all items are replaced say every tenth week, then such replacement constitutes an example of 'Group Replacement'. Thus, in group replacement all items are replaced, including some which (being replacement) may be almost new. Such a policy is considered advantageous when the item (which is one of the many identical items of a system) value is so small that the cost of maintaining records of individual ages cannot be justified economically. We can easily see the application of such a policy in replacing street light bulbs; it is certainly a costly affair to bring a truck and crew to the burnt-out bulb for replacing it, but once the crew is on the street for replacing all the bulbs at the fixed interval, the replacement cost per bulb is reduced significantly.

Example 4.11: The following mortality rates have been observed for a certain type of light bulbs.

Week	*1*	*2*	*3*	*4*	*5*
Per cent failing by					
end of week	10	25*	50	80	100

There are 1000 bulbs in use and it costs ₹ 2 to replace an individual bulb which has burnt out. If all bulbs were replaced simultaneously it would cost 50 paise per bulb. It is proposed to replace all bulbs at fixed intervals, whether or not

NOTES

they have burnt out and to continue replacing burnt out bulbs as they fail. At what intervals should all the bulbs be replaced? At what group replacement price per bulb would a policy of strictly individual replacement become preferable to the adopted policy?

Solution: Denoting p_i as the probability that a light bulb fails during the *i*th week of its life when put to use, we have

Week		1	2	3	4
p_t	0.10	0.15	0.25	0.30	0.20
(=1,2,3,4,5	(p_1)	(p_2)	(p_3)	(p_4)	(p_5)

and denoting N_i as the number of replacements made at the end of the *i*th week, we have

$$\begin{aligned}
N_0 &= \text{Number of light bulbs in the beginning} = 1000 \\
N_1 &= \text{Number of light bulbs being replaced by the end of the first week} \\
&= N_0 \cdot p_1 \\
&= (1000)\ (.10) = 100 \\
N_2 &= \text{Number of light bulbs being replaced by the end of the second week} \\
&= N_0 \cdot p_2 + N_1 \cdot p_1 \\
&= (1000)\ (.15) + (100)\ (.10) \\
&= 160
\end{aligned}$$

and similarly we can work out N_v N_A and N_5 as follows:

$$\begin{aligned}
N_3 &= N_0 . p_3 + N_1 . p_2 = N_2 . p_1 \\
&= (1000)\ (25) + (100)\ (.15) + (160)\ (.10) \\
&= 281 \\
N_4 &= N_0 . p_4 + N_1 .p_3 + N_2 . p_2 + N_3\ p_1 \\
&= (1000)\ (.30) + (100)\ (.25) + (160)\ (.15) + (281)\ (.10) \\
&= 377 \\
N_5 &= N_0 . p_5 + N_1 . p_4 + N_2 . p_3 + N_3 p_2 + N_4 . p_1 \\
&= (1000)\ (.20) + (100)\ (.30) + (160)\ (.25) + (281)\ (.15) + (377) \\
(.10) &= 350.
\end{aligned}$$

Since the replacement of all the bulbs simultaneously costs 50 paise per bulb and the replacement of an individual bulb on failure costs ₹ 2, the average cost of replacement of all the bulbs simultaneously can be worked out as shown.

End of week	*I*	*2*	*3*	*4*	*5*
Total Cost of Group Replacement (₹)*	700	1020	1582	2336	3036
Average cost of group replacement per week (₹)	700	510	527	584	607

The table indicates that the average cost of group replacement is minimum in the 2nd week and as such all the bulbs should be replaced after every two weeks. In other words, it is optimal to have a group replacement after every 2nd week in the given case.

For different weeks the total cost of group replacement has been calculated as follows:

At the end of 1st week, the total cost =

(1000 × .50) + (100 X 2) = ₹ 700

At the end of 2nd week, the total cost =

(1000 × .50) + (100 + 160)(2) = ₹ 1020

At the end of 3rd week, the total cost =

(1000 × .50) + [100 +160 + 281)(2)]= ₹ 1582

At the end of 4th week, the total cost =

(100 × .50) + [(100 + 160 + 281 + 377)(2)] = ₹ 2336

At the end of the 5th week, the total cost =

(100 × .50) + [(100 + 160 + 281 + 377 + 350) (2)] = ₹ 3036

To answer the other part of the question, we must first work out the cost of individual replacement, i.e, the cost of an item immediately when it fails and the same has been worked out as follows:

The expected life of each light bulb in the given case = $\sum_{i=1}^{5} ip_i$

$$= (1) (.10) + (2) (.15)+(3)(.25) + (4) (.30) +5) (.20)$$

Hence, average number of failures per week = $\dfrac{N_0}{\sum_{i=1}^{5} ip_i}$

$$= \frac{1000}{3.35}$$

and the cost of individual replacement per week

= (Average number of failures per week) X

(Cost of replacing an individual bulb)

$$= \frac{1000}{3.35} \times ₹ 2 = ₹ 597$$

Now we know from the above calculations that the group replacement after one week costs ₹ 700 whereas the individual replacement after a week costs ₹ 597 only, and hence the individual replacement is preferable in comparison to group replacement if the latter is to take place after every one week.

4.5 SEQUENCING

Sequencing models determine an appropriate order (sequence) for a series of jobs to be done on a finite number of service facilities in some pre-assigned order, so as to optimize the total cost (time) involved.

NOTES

Definition

Suppose there are n jobs $(1, 2, ..., n)$, each of which has to be processed one at a time at m machines (A, B, C, ...). The order of processing each job through each machine is given. The problem is to find a sequence among $(n!)^m$ number of all possible sequences for processing the jobs so that the total elapsed time for all the jobs will be minimum.

Terminology and Notations

The following terminologies are to be used.

Number of Machines: It means the service facilities through which a job must pass before it is completed.

Processing Order: It refers to the order in which various machines are required for completing the job.

Processing Time: It means the time required by each job on each machine.

Idle Time on a Machine: This is the time for which a machine remains idle during the total elapsed time. The notation x_{ij} is used to denote the idle time of a machine j between the end of the $(i-1)$th job and the start of the ith job.

Total Elapsed Time: This is the time between starting the first job and completing the last job, which also includes the idle time, if present.

No Passing Rule: It means, passing is not allowed, i.e., maintaining the same order of jobs over each machine. If each of N-jobs is to be processed through 2 machines M_1 and M_2 in the order M_1 M_2, then this rule will mean that each job will go to machine M_1 first and then to M_2. If a job is finished on M_1, it goes directly to machine M_2 if it is free, otherwise it starts a waiting line or joins the end of the waiting line, if one already exists. Jobs that form a waiting line are processed on machine M_2 when it becomes free.

Principal Assumptions

(*i*) No machine can process more than one operation at a time.

(*ii*) Each operation once started must be performed till completion.

(*iii*) Each operation must be completed before starting any other operation.

(*iv*) Time intervals for processing are independent of the order in which operations are performed.

(*v*) There is only one machine of each type.

(*vi*) A job is processed as soon as possible, subject to the ordering requirements.

(*vii*) All jobs are known and are ready for processing, before the period under consideration begins.

(*viii*) The time required to transfer jobs between machines is negligible.

Job Sequence Problems

Job sequencing is basically the planning of the jobs in sequential manner and is an essential part of any work. Without proper planning and scheduling one can not achieve the desired output and profit. For sequencing a job, generally the two techniques are used termed as Priority Rules and Johnson's Rules. Priority rules give the guidelines for properly sequencing the job, where as Johnson's rule is used to minimize the completion time for a set of jobs to be done on two different machines. Using these rules one can assign jobs and maximize product and profit.

Basic Characteristics of Job Sequencing

1. Only one single job should be scheduled for a machine at a time.
2. Do not stop the process in between before completion.
3. New processing can be started after the completion of the previous processing.
4. Any job is scheduled for processing as per the order and due date requirements.
5. If the jobs are transferred from one machine to another due to some reason, then the time involved in transferring the jobs is considered negligible.

Priority Rules: These rules are used to get specific guidelines for job sequencing. The rules do not consider job setup cost and time while analysing processing times. In it job processing time and due dates are given importance because the due dates are fixed to give delivery in time to the customers. The rules are very useful for process-focussed amenities, for example health clinics, print shops and manufacturing industries. Hence, priority rules minimize the time for completing a job, sequences the jobs in the organization, checks if any job is late and maximizes resource utilization. The most popular priority rules are as follows:

- **First Come First Serve (FCFS):** The job to be processed first is the job that turned up first in the organization.
- **Earliest Due Date (EDD):** The job to be processed first is the job that has earliest due date.
- **Shortest Processing Time (SPT):** The job to be processed first and completed is the job that is shortest in nature; in other words the job can be processed in short time.
- **Longest Processing Time (LPT):** The job to be processed first is the job that is very important or of high priority though it can take longer processing time.
- **Critical Ratio (CR):** The job to be processed first is analysed on the basis of critical ratio, which is an index number calculated from time remaining until due date divided by the remaining work time.

Johnson's Rule: This rule is applied to minimize the completion time for a set of jobs that are to be processed on two different machines or at two consecutive work stations. The main objectives of the rules are,

- To minimize the processing time while sequencing a set of jobs on two different machines or work stations.
- To minimize the complete idle time on the processing machines.
- To minimize the flow time of the job, i.e., from the start of the first job until the completion of the last job.

Necessary Conditions for Johnson's Rules: The necessary conditions to efficiently complete the processing of the jobs are as follows:

- Knowledge about job time for each job at the specific work station.
- Job time must not depend on sequencing of jobs.
- All the jobs to follow the predefined work sequence.
- Avoid job priority.

Four Steps Johnson's Rule: The following are the important four steps in Johnson's rule:

Step 1: List all the jobs and the processing time of each machine to which these jobs are scheduled.

Step 2: Choose the job which has the shortest processing time. If the shortest time has been scheduled on the first machine or work station then the job is selected first for processing. In case the shortest time is scheduled on the second machine or work station then the job is processed at the end.

Step 3: After scheduling the job for processing go to Step 4.

Step 4: Repeat Step 2 again to schedule the processing of remaining jobs and fill the sequence columns towards the centre till all the jobs are scheduled.

The following example will help you to understand how the sequences are scheduled.

For example, there are five jobs to be done at a factory and each job must be processed through two work stations at two different machines, drill machine and lathe machine. Using Johnson's rule we can schedule the sequence of jobs.

The time (in hours) for processing each job is given in the following table:

Jobs	***Work Station 1 (Drill)***	***Work Station 2 (Lathe)***
A	5	2
B	3	6
C	9	4
D	12	8
E	8	14

Using the Steps of Johnson's Rule, the job processing sequences are scheduled as follows:

Step 1: In the given table, the job with the shortest processing time is job **A**, in work station 2 (with a time of 2 hours). Because it is at the second work station, schedule **A** last.

				A

Step 2: Next shortest time is of job **B** (with a time of 3 hours). Because it is at the first work station, schedule it at first priority and eliminate it from the list.

B				**A**

Step 3: The next shortest time is of job **C** (with a time of 4 hours), but it is at the second work station. Therefore, place it at last before **A**.

B			**C**	**A**

Step 4: There is a tie between job **D** (with a time of 8 hours at work station 2) and job **E** (with a time of 8 hours at work station 1) for the shortest remaining job. Because job **E** is at the first work station, so place it first after job **B**. Then place job **D** in the last sequencing position. You will get the job sequence schedule as follows:

B	E	D	C	A

The final sequential times at both the work stations will be:

Jobs	*B*	*E*	*D*	*C*	*A*
Work Station 1 (Drill)	3	8	12	9	5
Work Station 2 (Lathe)	6	14	8	4	2

Check Your Progress

11. What are the two categories of replacement models?
12. What are the three types of relevant costs?
13. What is the basis of preventive replacement?
14. Define group replacement.
15. What is basic concept of sequencing models?
16. Give the definition of sequencing models.
17. State the principal assumptions of sequencing.
18. Define the job sequence problem.
19. What are basic characteristics of job sequencing?

4.6 ANSWERS CHECK YOUR PROGRESS

1. A linear programming problem in which all or some of the decision variables are constrained to assume non-negative integer values is called an Integer Programming Problem (IPP). In a Linear Programming Problem (LPP) if all variables are required to take integer values, then it is called the Pure (all) Integer Programming Problem (Pure IPP).
2. In IPP, all the decision variables are allowed to take any non-negative real values as it is quite possible and appropriate to have fractional values in many situations. There are several frequently occurring circumstances in business and industries that lead to planning models involving integer-valued variables. For example, in production, manufacturing is frequently scheduled in terms of batches, lots or runs. In allocation of goods, a shipment must involve a discrete number of trucks or aircrafts. In such cases the fractional values of variables like 13/3 may be meaningless in the context of the actual decision problem.
3. Integer programming is applied in business and industries. All assignment and transportation problems are integer programming problems, because

NOTES

in the assignment and travelling salesmen problem all the decision variables are either zero or one.

i.e., $x_{ij} = 0$ or other examples include capital budgeting and production scheduling problems. In fact, any situation involving decisions of the type 'Either to do a job or not' can be viewed as an IPP. In all such situations,

$x_{ij} = 1$, if the jth activity is performed.

$x_{ij} = 0$, if the jth activity is not performed.

In addition, allocation problems involving the allocation of men or machines give rise to IPP, since such commodities can be assigned only in integers and not in fractions.

4. A systematic procedure for solving pure IPP was first developed by R.E. Gomory in 1956, which he later used to deal with the more complicated cases of mixed integer programming problems. This method consists of first solving the IPP as an ordinary LPP by ignoring the restriction of integer values and then introducing a new constraint to the problem such that the new set of feasible solution includes all the original feasible integer solutions, but does not include the optimum non-integer solution initially found. This new constraint is called 'Fractional Cut' or 'Gomorian Constraint'. Then the revised problem is solved using the simplex method, till an optimum integer solution is obtained.
5. This is an enumeration method in which all feasible integer points are enumerated. This is the widely used search method based on branch and bound technique. It was developed in 1960 by A.H. Land and A.G. Doig. This method is applicable to both pure and mixed IPP. It first divides the feasible region into smaller subsets that eliminate parts containing no feasible integer solution.
6. In mixed IPP only some of the variables are restricted to integer values, while the other variables may take integer or other real values.

   Mixed integer cutting plane procedure: The iterative procedure for the solution of mixed integer programming problem is as follows.
7. Random number is a number whose probability of occurrence is the same as that of any other number in the collection.
8. Random numbers are called pseudorandom numbers when they are generated by some deterministic process and they qualify the predetermined statistical test for randomness.
9. The representation of reality in some physical form or in some form of mathematical equations may be called as simulation, i.e., simulation is imitation of reality. This is used because one is satisfied with suboptimal results for decision-making and also when representation by a mathematical model is beyond the capabilities of the analyst.
10. It is a simulation technique in which statistical distribution functions are created by using a series of random numbers. This is generally used to solve problems which cannot be adequately represented by the mathematical models.

11. The two categories are:

    (i) Models that deteriorate with time and (ii) Models that break down completely or are expected to breakdown shortly.

12. We have three types of relevant costs, viz., the operating costs, cost of productivity loss and the cost of capital in terms of depreciation.

13. Preventive replacement is based on the premise that the probability of failing increases with age or usage and when such probability becomes high, it makes sense to replace the item, during non-production hours before it fails to avoid the cost of failure.

14. In group replacement all items are replaced including some which (being replacement) may be almost new.

15. Sequencing models determine an appropriate order (sequence) for a series of jobs to be done on a finite number of service facilities in some pre-assigned order, so as to optimize the total cost (time) involved.

16. Suppose there are n jobs $(1, 2, ..., n)$, each of which has to be processed one at a time at m machines $(A, B, C, ...)$. The order of processing each job through each machine is given. The problem is to find a sequence among $(n!)^m$ number of all possible sequences for processing the jobs so that the total elapsed time for all the jobs will be minimum.

17. (i) No machine can process more than one operation at a time.

    (ii) Each operation once started must be performed till completion.

    (iii) Each operation once started must be performed till completion.

    (iv) Time intervals for processing are independent of the order in which operations are performed.

    (v) There is only one machine of each type.

    (vi) There is only one machine of each type.

    (vii) All jobs are known and are ready for processing, before the period under consideration begins.

    (viii) The time required to transfer jobs between machines is negligible.

18. Job sequencing is basically the planning of the jobs in sequential manner and is an essential part of any work. Without proper planning and scheduling, one cannot achieve the desired output and profit. For sequencing a job, generally the two techniques are used termed as Priority Rules and Johnson's Rules. Priority rules give the guidelines for properly sequencing the job, whereas Johnson's rule is used to minimize the completion time for a set of jobs to be done on two different machines. Using these rules one can assign jobs and maximize product and profit.

19. (i) Only one single job should be scheduled for a machine at a time.

    (ii) Only one single job should be scheduled for a machine at a time.

    (iii) New processing can be started after the completion of the previous processing.

    (iv) Any job is scheduled for processing as per the order and due date requirements.

    (v) If the jobs are transferred from one machine to another due to some reason, then the time involved in transferring the jobs is considered negligible.

4.7 SUMMARY

NOTES

- A linear programming problem in which all or some of the decision variables are constrained to assume non-negative integer values is called an Integer Programming Problem (IPP).
- In a Linear Programming Problem (LPP) if all variables are required to take integer values, then it is called the Pure (all) Integer Programming Problem (Pure IPP).
- In a Linear Programming Problem (LPP) if all variables are required to take integer values, then it is called the Pure (all) Integer Programming Problem (Pure IPP).
- A systematic procedure for solving pure IPP was first developed by R.E. Gomory in 1956, which he later used to deal with the more complicated cases of mixed integer programming problems.
- This is an enumeration method in which all feasible integer points are enumerated. This is the widely used search method based on Branch and Bound technique. It was developed in 1960 by A.H. Land and A.G. Doig.
- Monte Carlo simulation is often used by modern management when it cannot use other techniques.
- By using a fresh series of random numbers at the appropriate junctures we can also examine the reactions of the simulated model just as if the same alterations had actually been made in the system itself.
- The main purpose of simulation in management is to provide feedback, which is vital for the learning process.
- Simulation is a representation of reality through the use of a model or other device which will react in the same manner as reality under a given set of conditions.
- Simulation is also defined as the use of a system model that has the designed characteristics of reality in order to produce the essence of actual operation.
- Replacement models generally fall into two categories depending upon the life pattern of the equipment under study. There are models which deal in replacing equipment that deteriorate with age or usage.
- The optimal replacement policy is to replace the item every T time period if it has not failed earlier, so that expected cost per time period is minimized.
- Sequencing models determine an appropriate order (sequence) for a series of jobs to be done on a finite number of service facilities in some pre-assigned order, so as to optimize the total cost (time) involved.
- Suppose there are n jobs (1, 2, ..., n), each of which has to be processed one at a time at m machines (A, B, C, ...). The order of processing each job through each machine is given. The problem is to find a sequence among $(n!)^m$ number of all possible sequences for processing the jobs so that the total elapsed time for all the jobs will be minimum.

- Number of Machines: It means the service facilities through which a job must pass before it is completed.
- Processing Order: It refers to the order in which various machines are required for completing the job.
- Idle Time on a Machine: This is the time for which a machine remains idle during the total elapsed time. The notation x_{ij} is used to denote the idle time of a machine j between the end of the $(i-1)$th job and the start of the ith job.
- Job sequencing is basically the planning of the jobs in sequential manner and is an essential part of any work. Without proper planning and scheduling one cannot achieve the desired output and profit. For sequencing a job, generally the two techniques are used termed as Priority Rules and Johnson's Rules.

4.8 KEY TERMS

- **Integer Programming Problem (IPP):** A problem in which all or some variables are constrained to assume non-negative integer values.
- **Pure IPP:** In an Pure IPP (Integer programming problem) when all variables are constraint to non-negative integer values.
- **Mixed IPP:** In this type of IPP, some variables are allowed to assume nonnegative non-integer values.
- **Branch and bound method:** An enumeration method in which a feasible region is divided into smaller subsets and each of these is examined successively to find a feasible solution. Initially, it is solved ignoring integer constraints. Then, solution so obtained is divided into two disjoint subsets of two consecutive non-negative integers. This method is used both for pure and mixed integer programming problems.
- **Simulation:** A representation of reality using a model or other device that reacts in the same manner as reality under a given set of conditions. It is an imitation of a reality.
- **Random number:** It refers to a number assigned to a random variable following uniform probability density function.
- **Preventive replacement:** Replacement of equipment to avoid the cost of sudden failures by estimating when such failures may occur. It can also be called as replacement in anticipation of failure.
- **Sequencing models:** Sequencing models determine an appropriate order (sequence) for a series of jobs to be done on a finite number of services facilities in some pre-assigned order, so as to optimize the total cost (time) involved.
- **Number of machines:** It means the service facilities through which a job Sequencing Problem must pass before it is completed.

4.9 SELF-ASSESSMENT QUESTIONS AND EXERCISES

NOTES

Short-Answer Questions

1. How an IPP is different from a linear programming problem?
2. When is simulation used? What is done in a simulation?
3. Differentiate between a random number and a pseudorandom number.
4. Who gave the code name Monte Carlo?
5. What is replacement? What factors cause a replacement policy to be adopted by a modern business unit? Give suitable examples.
6. Write a short note on replacement models.
7. What is no passing rule in a sequencing algorithm?
8. State the principal assumptions made while dealing with a sequencing problem.
9. What is sequencing problem?

Long-Answer Questions

1. Find the optimum integer solution of the following pure integer programming problems.

   Max $Z = 4x_1 + 3x_2$

   Subject to constraints,
   $$x_1 + 2x_2 \leq 4$$
   $$2x_1 + x_2 \leq 6$$
   $$x_1,\ x_2 \geq 0 \text{ and are integers.}$$

2. Find the optimum integer solution of the following pure integer programming problems.

   Max $Z = 3x_1 + 4x_2$

   Subject to constraints,
   $$3x_1 + 2x_2 \leq 8$$
   $$x_1 + 4x_2 \geq 10$$
   $$x_1,\ x_2 \geq 0 \text{ and are integers.}$$

3. Explain simulation and its types.
4. What is a random variable? Explain why it is used.
5. Explain the methodology of the Monte Carlo simulation techniques.
6. Describe the role of simulation in management process.
7. Discuss Monte Carlo simulation with reference to queuing theory, inventory control and production line.
8. Explain the importance of preventive replacement in the context of replacement strategies.
9. Illustrate the graphical method to solve two jobs on machines with given technological ordering for each job. What are the limitations of the method?

4.10 FURTHER READING

NOTES

Kalavathy, S. 2002. *Operations Research*. New Delhi: Vikas Publishing House Pvt. Ltd.

Gupta, P.K. and D.S. Hira. 2002. *Introduction to Operations Research*. New Delhi: S. Chand And Company Limited.

Taha, H.A. 2006. *Operations Research: An Introduction*, 8th Edition. New Delhi: Prentice-Hall of India.

Jensen, Paul A. and Jonathan F. Bard. 2003. *Operations Research Models and Methods*. New York: John Wiley & Sons.

Arumugam, R. S. 2006. *Operations Research*. Palayamkottai (Tamil Nadu): New Gamma Publications.

Sundharesan, V., K. S. Ganapathy and K. Ganesan. 2017. *Resource Management Techniques (Operations Research)*. Chennai: A. R. Publications.

Gillett, Billy E. 2007. *Introduction to Operations Research*. New Delhi: Tata McGraw-Hill.

Ackoff, R. L. and M. W. Sasieni. 1968. *Fundamentals of Operations Research*. New York: John Wiley & Sons Inc.

Kothari, C. R. 1992. *An Introduction to Operational Research*. New Delhi: Vikas Publishing House Pvt. Ltd.

Sharma, J. K. 2001. *Operations Research: Theory and Applications*. New Delhi: Macmillan India Ltd.

Swaroop, Kanti, P.K. Gupta and Man Mohan. 2001. *Operations Research*. New Delhi: Sultan Chand & Sons.

UNIT 5 NONLINEAR PROGRAMMING

Structure

5.0 INTRODUCTION

In mathematics optimization, the numerical optimization of nonlinear multivariable objective functions in general requires effective and reliable techniques. Because these issues necessitate an iterative solution technique, trial and error becomes impracticable when there are more than three or four variables. Because a generic nonlinear function's behaviour is unpredictable, robustness (the capacity to obtain a solution) is required. There may be relative maxima or minima, saddle points, regions of convexity or concavity, and so on. In some areas, the optimization method may advance very slowly toward the optimal solution, necessitating a significant amount of computer time. Fortunately, we have a lot of experience evaluating nonlinear programming techniques for unconstrained functions, so we can assess many ways for optimizing these functions. The Fibonacci sequence has the property that a number is the sum of its two predecessors. Therefore the sequence can be computed by repeated addition. The ratio of two consecutive numbers approaches the Golden ratio, 1.618...,. Binary search works by dividing the seek area in equal parts (1:1). Fibonacci search can divide it into parts approaching 1:1.618 while using the simpler operations.

The Karush–Kuhn–Tucker (KKT) conditions, also known as the Kuhn–Tucker conditions, are first derivative tests (sometimes called first-order necessary conditions) for a solution in nonlinear programming to be optimal, provided that some regularity conditions are satisfied. Quadratic Programming (QP) is the process of solving certain mathematical optimization problems involving quadratic functions.

Specifically, one seeks to optimize (minimize or maximize) a multivariate quadratic function subject to linear constraints on the variables. Quadratic programming is a type of nonlinear programming.

NOTES

The Nonlinear Programming (NLP) is the process of solving an optimization problem where some of the constraints or the objective function are nonlinear. An optimization problem is one of calculation of the extrema (maxima, minima or stationary points) of an objective function over a set of unknown real variables and conditional to the satisfaction of a system of equalities and inequalities, collectively termed constraints. It is the sub-field of mathematical optimization that deals with problems that are not linear.

In this unit, you will learn about the one and multivariable unconstrained optimization, Kuhn-Tucker conditions for constrained optimization, quadratic programming and types of nonlinear programming problems.

5.1 OBJECTIVES

After going through this unit, you will be able to:

- Discuss the one and multivariable unconstrained optimization
- Explain the Kuhn-Tucker conditions for constrained optimization
- State quadratic programming
- Describe the separable programming
- Discuss the convex and non-convex programming

5.2 ONE AND MULTIVARIABLE UNCONSTRAINED OPTIMIZATION

The methods used to find the maximum/minimum of function $f(x)$ depending on whether $f(x)$ is a single variable of $f(x)$, $x = (x_1, x_2 \ldots x_n)$ has several variables are one dimensional

1. Search Method (1-D).
2. Direct Search Methods (Interal halving)
3. Fibonacci Sequence Method
4. Golden Section Search Method

The 1-D search method is also known as *univariate search method*, where variables in x are varied one at a time and the iterations are stopped, if successive values of $f(x)$ yield the same value. It is also called the *step method*.

In Fibonacci sequence method, we specify the limit of accuracy and calculate the end points of the intervals using the Fibonacci sequence.

A slightly modified form of Fibonacci sequence method is the Golden section search method, where the distance of the points of uncertainty intervals are determined using a standard formula.

5.2.1 Fibonacci Search

NOTES

Fibonacci search is a univariate search technique that can be used to find the maximum (or minimum) of an arbitrary unimodal, univariate objective function. This method is a sequential search technique that successfully reduces the interval in which the maximum (or minimum) of an arbitrary nonlinear function must lie. To apply this technique, the assumption of unimodality must be invoked or the technique may locate a stationary point or completely fail. This method gives an optimal solution in the minimax case in a sequence of N functional evaluations.

Definition: The interval of uncertainty is defined as the interval in which the optimum solution is known to exist.

Consider the following successive relationship which generates an infinite series of numbers

$$x_n = x_{n-1} + x_{n-1}, \qquad n = 2, 3...$$

Define $x_0 = 0$ and $x_1 = 1$

The above equation generates a series of numbers that are known as *Fibonacci numbers* given in Table 5.1. The Fibonacci sequence is defined as

Table 5.1 *Fibonacci Numbers*

$$F_N = F_{N-1} + F_{N-2}, N > 1$$

Identifier	*Sequence*	*Fibonacci Number*
F_0	0	1
F_1	1	1
F_2	2	2
F_3	3	3
F_4	4	5
F_5	5	8
F_6	6	13
F_7	7	21
F_8	8	34
F_9	9	55
F_{10}	10	89
F_{11}	11	144

Interval Length of Fibonacci Method

Define the initial interval of search to be of Length L_0. This interval is critical interval of uncertainty. This interval must lie between points A and B. To reduce the initial interval of uncertainty to some finite length L_N using exactly N functional evaluations.

L_0 = Length of the interval of uncertainty after n functional evaluation.

X_n = Value of the variable X after N functional evaluations.

f_n = Value of the objective function using X_n, $n = 1, 2,\ldots,N$

$\in$ = The Minimum separation allowed between any points over the interval L_0.

It represents the resolution that can be obtained between the points X_n and X_{n-1}. The calculation of this parameter ε is essential as the optimum solution is based on the elimination of the regions in which the optimum solution can't lie.

Steps Involved in Fibonacci Sequence Method

Step 1: Define the end points of the search A and B.

Step 2: Define the number of functional evaluations N, that are to be used in the search.

Step 3: Define the minimum resolution parameter $\in$.

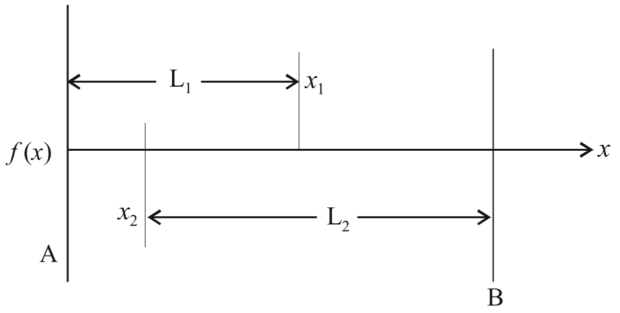
Step 4: Define the initial interval and first interval of uncertainty as $(B - A)$.

Step 5: Define the second interval of uncertainty as follows:

$$L_2 = \frac{1}{F_n}\left(L_0 F_{N-1} + \in (-1)^n\right)$$

where F_N and F_{N-1} are Fibonacci numbers.

Step 6: Locate the first two functional evaluations as the two symmetric points x_1 and x_2 defined as follows: $x_1 = A + L_2$, $x_2 = B - L_1$.



Step 7: Calculate $f(x_1)$ and $f(x_2)$ and eliminate the interval in which the optimum cannot lie.

Step 8: Use the relationship $L_n = L_{n-2} - L_{n-1}$ to locate subsequent points of evaluation within the remaining interval of uncertainty.

Repeat steps 7 and 8 until N fractional evaluations have been executed. The final solution can be either an average of the two points evaluations (x_N and x_{N-1}) or the best (max/min) functional evaluation.

Example 5.1: Find the minimum of $f(x) = x^2 - 2x$ by Fibonacci method. Take interval $0 \leq x \leq 1.5$ and $\in = 0.25$.

Solution: Fibonacci sequence is defined as

$$F_2 = F_0 + F_1, F_3 = F_1 + F_2 \text{ and so on.}$$

$$\{F_0, F_1, F_2, F_3, F_4, F_5\} = \{1, 1, 2, 3, 5, 8\}$$

Step 1: Find Fibonacci number F_n using

$$\in = \frac{1}{F_N} \leq 0.25 \Rightarrow N \geq 4.$$

Take $N = 4$

Step 2: End points of the interval are (0, 1.5)

$\therefore\ A = 0, B = 1.5;\ L_0 = B - A = 1.5 - 0 = 1.5$

Step 3: Calculate $x_1 = \mathrm{A} + L_2,\ x_2 = B - L_2$

$$\begin{aligned} L_2 &= \frac{1}{F_N}\left(L_0F_{N-1} + \in(-1)^N\right) \\ &= \frac{1}{F_N}\left(1.5F_3 + \in(-1)^4\right) \\ &= \frac{1}{5}\left(1.5(3) + 0.25\right) \\ &= 0.95 \end{aligned}$$

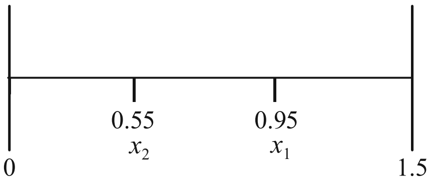
$$x_1 = 0 + 0.95 = 0.95$$

$$x_2 = 1.5 - 0.95 = 0.55$$

Step 4: Plot point x_2 in the figure at a distance 0.55 from (0) and x_4 at a distance 0.95 from the end points.

$$f(x_1) = f(0.95) = -0.9975$$

$$f(x_2) = f(0.55) = -0.7975$$



Step 5: Since $f(x_2) > f(x_1)$ *i.e.* $f(0.55) > f(0.95)$

Optimum lies in the interval (0.55, 1.5)

Step 6: Take $A = 0.55,\ B = 1.5$

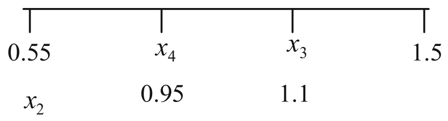
Redetermine a new point $x_3 = A + L_3$

$$x_4 = B - L_3$$

$$L_3 = L_2 - L_1 = 1.5 - 0.95 = 0.55$$

$$x_3 = 0.55 + 0.55 = 1.1$$

$$x_4 = 1.5 - 0.55 = 0.95$$



Step 7: Find $f(0.95) = -\,0.9975$ since $f(1.1) > f(0.95)$

$f(1.1) = -0.99$

Also $f(0.55) > f(0.95)$

Reject the interval (1.1, 1.5).

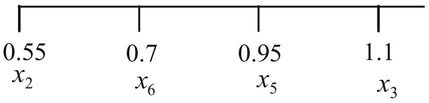
The new interval of uncertainty is (0.55, 1.1)

$$A = 0.55,\ B = 1.1$$

$$L_4 = L_2 - L_3 = 0.95 - 0.55 = 0.4$$

$$x_5 = 0.55 + 0.4 = 0.95$$

$$x_6 = 1.1 - 0.4 = 0.7$$



$$f(0.95) = -\,0.9975$$

$$f(0.7) = -\,0.91$$

$$f(0.7) > f(0.95)$$

$$f(1.1) > f(0.95)$$

$$f(0.7) > f(0.95)$$

New interval of uncertainty is (0.95, 1.1.)

$$L_5 = L_3 - L_4$$

$$= 0.55 - 0.4 = 0.15 < 0.25\ (\in)$$

The following table shows the progression through three functional evaluations.

Functional Evaluations(n)	*Interval of uncertainty*	x_{n-1}	$F(x_{n-1})$	x_n	$F(x_n)$
1	$0.95 \geq x \geq 0$	0.55	–0.7975	0.95	–0.9975
2	$0.55 \geq x \geq 0.95$	0.95	–0.9975	1.1	–0.99
3	$0.7 \geq x \geq 0.95$	0.7	–0.91	0.95	–0.9975

At the 3rd functional evaluation, the interval of uncertainty is established as $I_3 = 0.95$.

The best optimal solution is $\frac{1.1 + 0.95}{2} = 1.025$

$$f(1.025) = -0.999375 = -1.0$$

Example 5.2: Maximize the function $f(x) = -3x^2 + 21.6x + 1.0$ with a minimum resolution of 0.50 over six functional evaluations. The optimal value of $f(x)$ is assumed to be in the range $25 \geq x \geq 0$.

Solution

Given $\in = 0.50$

End points $A = 0$

$$N = 6$$

$$B = 25$$

$$L_{10} = L_1 = B - A = 25$$

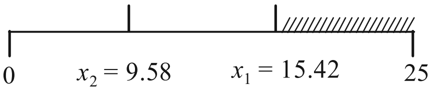
$$L_2 = \frac{1}{F_N}\left(L_1 F_{N-1} + \in(-1)^N\right)$$

$$L_2 = \frac{1}{F_6}\left(L_1 F_5 + \in(-1)^6\right)$$

$$= \frac{1}{13}\left(25(8) + 0.5\right) = 15.4231$$

$$
\begin{aligned}
L_2 &= 15.4231\\
x_1 &= A + L_2 = 15.4231\\
x_2 &= B - L_2 = 9.5769\\
f(x_1) &= f(15.4231)\\
&= -3\left(15.4231\right)^2 + 21.6\left(15.4231\right) + 1.0 = -\,379.477\\
f(x_2) &= f(9.5769)\\
&= -3\left(9.5769\right)^2 + 21.6\left(9.5769\right) + 1.0 = -\,67.233
\end{aligned}
$$

Since $f(x_2) > f(x_1)$, the region of uncertainty is 0 to 15.4321, i.e., region to the right of $x_1 = 15.42$ is eliminated.



$$
\begin{aligned}
L_0 &= 25\\
L_1 &= 25\\
L_2 &= 15.4231
\end{aligned}
$$

Here end points are $A = 0,\ \ B = 15.42$

$$
\begin{aligned}
L_3 &= L_1 - L_2 = 25 - 15.4231 = 9.5769\\
x_3 &= A + L_3 = 0 + 9.5769\ = 9.5769\\
x_4 &= \mathrm{B} - L_3 = 15.42 - 9.5769 = 5.8462\\
f(x_3) &= f(9.5769) = -\,67.233\\
f(x_4) &= f(5.8462) = 24.744\\
f(x_4) &> f(x_3) \text{ i.e., } f(5.8462) > f(9.5769)
\end{aligned}
$$

the region of uncertainty is 0 to 9.5769.

i.e., region to the right of $x_3 = 9.5769$ is eliminated.

The following table shows the progression through the six functional evaluations.

Fibonacci Search

Functional Evaluations	*Interval of uncertainty*	x_{n-1}	$F(x_{n-1})$	x_n	$F(x_n)$
1	$0 \le x \le 2.5$	0	1	25	−1938.8
2	$0 \le x \le 15.4231$	9.5769	–67.233	15.4231	–379.477
3	$0 \le x \le 9.5769$	5.8462	24.744	9.5769	–67.233
4	$0 \le x \le 5.8462$	3.731	39.83	5.8962	24.744
5	$2.115 \le x \le 5.8462$	2.115	32.26	3.731	39.83
6	$2.115 \le x \le 4.2304$	3.731	39.83	4.2304	38.688

At the sixth functional evaluation, the interval of uncertainty is established as $I_6 = 2.115$.

The optimal solution is $x_5 = 3.731$.

$$f(x_5) = 39.83$$

$$\in = 4.2304 - 3.731 = 0.4994 < 0.5$$

Exercise: Find the minimum of $f(x) = x^3 - 3x - 5$, $0 \leq x \leq 1.2$ and level of uncertainty is $x = 0.35$.

5.2.2 Golden Section Search

In performing a Fibonacci search, a prior specification of the resolution factor ($\in$) and the number of experiments to be performed (N) are required. In search methods to have a proper functioning, the successive experiments will gradually reduce the interval of uncertainty, i.e., the final interval of uncertainty will converge to zero as the number of functional evaluation increases to infinity provided that $\in$ is small.

$$\lim_{\substack{N\to\infty \\ \in\to 0}} \{L_2\} = \lim_{\substack{N\to\infty \\ \in\to 0}} \left\{\frac{1}{F_N}\left(L_0 F_{N-1} + \in\left(-1\right)^N\right)\right\}$$

$$= L_0 \left\{\frac{F_{N-1}}{F_N}\right\}$$

In the above limit, the ratio of $\frac{F_{N-1}}{F_N}$ goes to 0.618 which is known as *golden ratio or golden section*. The modified version of Fibonacci method with this golden ratio is known as *golden section search.* In comparison to the Fibonacci method, the golden section search is less efficient as it is derived from Fibonacci method. In golden section search method, the minimax principle is lost for early searches as the solution in each search is neither dependent upon resolution considerations nor on the number of functional evaluations. In practice, the golden section search is often used because it requires less information to implement each search and is by construction self-starting.

The general procedure for this method is given below:

Step 1: Define the initial interval of uncertainty as $L_0 = B - A$, and A and B are the end points of the search.

Step 2: Determine the first two functional evaluations at points x_1 and x_2 defined by

$$x_1 = A + 0.618\ (B - A)$$

$$x_2 = B - 0.618\ (B - A)$$

Step 3: Eliminate the appropriate region in which the optimum cannot lie.

Step 4: Determine the region of uncertainty defined by

$$L_{j+1} = L_{j-1} - L_j \quad j = 2, 3\ldots\ldots$$

Where,

$$L_0 = B - A$$

$$L_1 = B - A$$

$$L_2 = x_1 - A \text{ or } L_2 = B - x_2$$

depending upon the region eliminated in step 3.

Step 5: Establish a new functional evaluation using the result of step 4. Evaluate $f(x)$ at this point, and then go to step 3. Repeat this procedure until a specified convergence criteria is satisfied.

Example 5.3: Minimize $f(x) = x^4 - 15x^3 + 72x^2 - 1135x$.

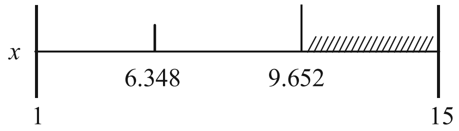
Terminate the search when $|f(x_n) - f(x_{n-1})| \leq 0.50$.

The initial range of x is $1 \leq x \leq 15$.

Solution

Given $A = 1$, $B = 15$

$$L_0 = B - A = 14$$
$$x_1 = 1 + 0.618(14) = 9.652$$
$$x_2 = 15 - 0.618(14) = 6.348$$



$$f(x_1) = 595.70, \quad f(x_2) = -168.82$$

Since $f(x_1) > f(x_2)$ the region to the right of $x = 9.652$ can be eliminated and the interval of uncertainty after two functional evaluations is given by $1 \leq x \leq 9.652$.

The following table shows the progression of the golden section search method.

Finctional Evaluations (n)	*X_{n-1} (right)*	*$f(x_{n-1})$*	*X_n (left)*	*$f(x_n)$*	*Interval of uncertainty*	*Length*
2	9.652	595.70	6.348	–168.82	$1 \leq x \leq 9.652$	8.652
3	6.346	–168.80	4.304	–100.06	$4.304 \leq x \leq 9.652$	5.348
4	7.609	–114.64	6.346	–168.80	$4.304 \leq x \leq 7.609$	3.305
5	6.346	–168.80	5.566	–147.61	$5.566 \leq x \leq 7.609$	2.043
6	6.828	–166.42	6.346	–168.80	$5.566 \leq x \leq 6.828$	1.262
7	6.346	–168.80	6.048	–163.25	$6.048 \leq x \leq 6.828$	0.780
8	6.530	–169.83	6.346	–168.80	$6.346 \leq x \leq 6.828$	0.482
9	6.643	–169.34	6.530	–169.83	$6.346 \leq x \leq 6.643$	0.297

$$|f(x_9) - f(x_8)| = -169.34 - (-169.83)$$
$$= 0.49$$

Since termination criterion are satisfied, the golden section search will stop at this point.

The optimum solution is given by $x = 6.643$, $f(x) = -169.34$

Exercise: Find the minimum of $x^2 - 2x$ in (1, 2) within an interval of uncertainty $0.13 < 0$ where L_0 is the original interval of uncertainty. Solve by Golden section search method.

Check Your Progress

1. What is one multivariable unconstrained optimization search method?
2. Define Fibonacci search method.
3. What do you mean by the golden section search method?

5.3 KUHN-TUCKER CONDITIONS FOR CONSTRAINED OPTIMIZATION

NOTES

The necessary and sufficient conditions for a local optimum of the general Nonlinear programming problem, with both equality and inequality constraints is called Kuhn-Tucker conditions.

Consider the following general form of NLP in which the objective function is to be maximized with all the constraints of ≤ type.

Maximize Z = $f(x_1, x_2, \ldots x_n)$

Subject to, $G(x_1, x_2, \ldots x_j \ldots x_n) \le b_i$, $i = 1, 2, \ldots, m$

$$j = 1, 2, \ldots n$$

$$x_j \ge 0$$

The modified form of the above problem is

Max $\qquad Z = f(x_1, x_2 \ldots x_n)$

Subject to, $g_i(x_1, x_2 \ldots x_n) \le 0,\ i = 1, 2 \ldots n$

$$x_j \ge 0,\ j = 1, 2, \ldots n$$

Where $\quad g_i(x_1, x_2 \ldots x_n) = G(x_1, x_2 \ldots x_n) - b_i$.

By adding slack variable to the constraints, the above problem can be modified as

$$\text{Maximize } Z = f(x_1,\ x_2, \ldots x_n)$$

Subject to $g_i\left(x_1,\ x_2 \ldots x_n\right) + s_i^2 = 0, \qquad i = 1, 2, \ldots m \qquad x_j \ge 0, \qquad j = 1, 2, \ldots n$

where s_i^2 are slack variables added to the constraint.

This problem consists of $n + m$ variables and m constraints. Let L be the Lagrangian function and λ_i be the Lagrangian multiplies of the i^{th} constraints.

Then the Lagrangian function is given as

$$L(x, s, \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i \left[g_i(x) + s_i^2 \right]$$

where $\lambda = (\lambda_1, \lambda_2, \ldots \lambda_m)$ is the vector of the Lagrangian multiplier.

The necessary conditions for an extreme point to be local optimum (maxima or minima) can be obtained by solving the frequency equations.

$$\frac{\partial L}{\partial x_j} = \frac{\partial f(x)}{\partial x_j} - \sum_{i=1}^{M} \lambda_i \frac{\partial g_i(x)}{\partial x_j} = 0 \qquad j = 1, 2, \ldots n$$

$$\frac{\partial L}{\partial \lambda_i} = -\left(g_i(x) + s_i^2 \right) = 0 \qquad i = 1, 2, \ldots m$$

$$\frac{\partial L}{\partial s_i} = -2x_i \lambda_i = 0 \qquad i = 1, 2, \ldots m$$

The equation $\frac{\partial L}{\partial \lambda_i} = 0$ gives back the original set of constraints

$g_i\left(x\right) + s_i^2 = 0.$

NOTES

The equation $\frac{\partial L}{\partial s_i} = 0$ provides the set of rules $-2\lambda_i x_i = 0$ or $\lambda_i s_i = 0$ for finding the unconstrained optimum. The condition $\lambda_i s_i = 0$ implies that either $\lambda_i = 0$ or $s_i = 0$.

If $s_i = 0$ and $\lambda_i > 0$ then the equation $\frac{\partial L}{\partial \lambda_i} = 0$ gives $g_i(x) = 0$.

This means either $\lambda_i = 0$ or $g_i(x) = 0$ which in turn can be written as

$$\lambda_i g_i(x) = 0$$

Since $s_i^2 \geq 0 \Rightarrow g_i(x) \geq 0$, the equation $\lambda_i g_i(x) = 0$

$\Rightarrow$ when $g_i(x) < 0, \lambda_i = 0$ and when $g_i(x) = 0, \lambda_i > 0$. Also, if $\lambda_i = 0, s_i^2 > 0$ then the i^{th} constraint is inactive. This constraint will not change the optimum value of Z^n because

$$\lambda = \frac{\partial z}{\partial b_j} = 0 \text{ and hence can be discarded.}$$

To maintain the relation of λ_i, Kuhn-Tucker has established the following necessary conditions.

i. $\lambda_i \geq 0 \; i = 1, 2, ...m$

ii. $\frac{\partial L}{\partial x_j} = 0 \quad (j = 1, 2, ...n)$

iii. $\lambda_i g_i(x_1, x_2, \ldots, x_n) = 0 \quad i = 1, 2, \ldots m$

iv. $g_i(x_1, x_2, \ldots, x_n) = 0 \quad i = 1, 2, \ldots m$

Note:

For a minimization problem, with concave objective function and with all ≥ type constraints (concave type constraints), the value of λ_i should be ≥ 0. If L is concave, in the case of maximization problem and convex in the case of minimization problem. The different possibilities of λ_i are given below.

(i) For maximization objective function ≤ type constraints $\lambda_i \geq 0 \quad i = 1, 2 \ldots m$

(ii) For maximization objective function, with ≥ type constraints $\lambda_i \leq 0 \quad i = 1, 2 \ldots m$

(iii) For maximization objective function with = type constraints λ_i is unrestricted in sign,

$i = 1, 2, \ldots$ m

NOTES

(iv) For minimization objective function with ≤ type of constraints $\lambda_i \leq 0 \qquad i = 1, 2, ...m$

(v) For minimization objective function and with ≥ type constraints $\lambda_i \geq 0 \qquad i = 1, 2, ...m$

(vi) For minimization objective function and with = constraints, λ_i is unrestricted in sign, $i = 1, 2 \ldots m$.

Table 5.2 Summary of Kuhn-Tucker Conditions

	$F(x)$	$g(x)$	λ_i
Maximum	Concave	Convex Concave Linear	≥ 0 ≤ 0 No restriction
Minimum	Convex	Convex Concave Linear	≤ 0 ≥ 0 No restriction

If $f(x)$ and $g(x)$ are indefinite then K.T. conditions are necessary but not sufficient for stationary points.

Example 5.4: Solve the following NLP using Kuhn-Tucker conditions.

Maximize $\quad Z = x_1^2 - x_1x_2 - 2x_2^2$

Subject to, $\quad 4x_1 + 2x_2 \leq 24$

$5x_1 + 10x_2 \leq 20$

$x_1, x_2 \geq 0$

Solution: The Lagrangian function

$$\lambda(x,\ \lambda) = x_1^2 - x_1x_2 - 2x_2^2 - \lambda_1(4x_1 + 2x_2 - 24) - \lambda_2(5x_1 + 10x_2 - 20)$$

The Kuhn-Tucker conditions are

$$\lambda_1 \geq 0, \quad \lambda_2 \geq 0 \tag{5.1}$$

$$\frac{\partial L}{\partial x_1} = 2x_1 + x_2 - 4\lambda_1 - 5\lambda_2 = 0 \tag{5.2}$$

$$\frac{\partial L}{\partial x_2} = x_1 - 4x_2 - 2\lambda_1 - 10\lambda_2 = 0 \tag{5.3}$$

$$\lambda_1(4x_1 + 2x_2 - 24) = 0 \tag{5.4}$$

$$\lambda_2(5x_1 + 10x_2 - 30) = 0 \tag{5.5}$$

$$4x_1 + 2x_2 - 24 \leq 0 \tag{5.6}$$

$$5x_1 + 10x_2 - 30 \le 0 \qquad (5.7)$$

From Equation (5.4) if $\lambda_1 = 0, x_1$ and x_2 must be equal too which is not true.

$$4x_1 + 2x_2 - 24 = 0 \qquad (5.8)$$

NOTES

Similarly from Equation (5.5) if $\lambda_2 = 0, x_1$ and x_2 must be equal too which is not true.

$$5x_1 + 10x_2 - 30 = 0 \qquad (5.9)$$

Solving Equations (5.8) and (5.9) Equations, (5.2) and (5.3), we get the optimum solution

$$x_1 = 6, \quad x_2 = 0, \quad \lambda_1 = 0, \quad \lambda_2 = 0, \quad Z_{\max} = 36$$

Example 5.5: Determine x_1 and x_2 so as to

Maximize $\quad Z = 12x_2 + 21x_1 + 2x_1x_2 - 2x_1^2 - 2x_2^2$

Subject to the constraints

$(i)\ x_2 \le 8,\ (ii)\ x_1 + x_2 \le 10, x_1, x_2 \ge 0$

Solution

$$f(x_1, x_2) = 12x_1 + 21x_2 + 2x_1x_2 - 2x_1^2 - 2x_2^2$$

$$g_1(x_1, x_2) = x_2 - 8 \le 0$$

$$g_2(x_1, x_2) = x_1 + x_2 - 10 \le 0$$

Define Lagrangian function as

$$\lambda(x, s, \lambda) = f(x) - \lambda_1(g_1(x) + s_1^2) - \lambda_2(g_2(x) + s_2^2)$$

The Kuhn-Tucker necessary conditions can be stated as:

(i) $$\frac{\partial f}{\partial x_j} - \sum_{i=1}^{2} \lambda_i \frac{\partial g_i}{\partial x_i}, = 0 \quad j = 1, 2$$

$$\Rightarrow \quad 12 + 2x_2 - 4x_1 - \lambda_2 = 0$$

$$21 + 2x_1 - 4x_2 - \lambda_1 - \lambda_2 = 0$$

(ii) $$\lambda_i g_i(x) = 0 \quad j = 1, 2$$

$$\lambda_1(x_2 - 8) = 0$$

$$\lambda_2(x_1 + x_2 - 10) = 0$$

(iii) $$g_i(x) \le 0$$

$$\Rightarrow \quad x_2 - 8 \le 0, \quad x_1 + x_2 - 10 \le 0$$

(iv) $$\lambda_i \ge 0 \quad i = 1, 2$$

There are 4 cases.

Case I

If, $\quad \lambda_1 = 0, 1 = 0$, then from condition (i)

$$12+2x_2-4x_1 = 0$$

$$21+2x_1-4x_2 = 0$$

NOTES

Solving these equations, we get $x_1 = {}^{15}\!/_2, x_2 = 9$. This solution violates condition (iii) and therefore it should be discarded.

Case II

$\lambda_1 \neq 0$, $\lambda_2 = 0$, then from condition (ii)

we have $x_2 - 8 = 0$ or $x_2 = 8$

$$x_1 + x_2 - 10 = 0 \text{ or } x_1 = 2$$

Substituting these values in conditions (i) we get

$\lambda_1 = -27$ and $\lambda_2 = 20$. However this solution violates the condition (iv) and therefore may be discarded.

Case III

$\lambda_1 \neq 0$, $\lambda_2 = 0$, then from conditions (ii) and (i) we have

$$x_1 + x_2 = 10$$

$$2x_1 - 4x_2 = -12 + \lambda_1$$

Solving these equations we get $x_1 = 2, x_2 = 8$ and $\lambda_1 = -16$.

This solution violates the condition (iv) and therefore may be discarded.

Case IV

$\lambda_1 = \lambda_2 \neq 0$, then from conditions (i) and (ii)

$$2x_1 - 4x_2 = -12 + \lambda_2$$

$$2x_1 - 4x_2 = -21 + \lambda_2$$

$$x_1 + x_2 = 0.$$

Solving these equations, we get $x_1 = {}^{17}\!/_4, x_2 = {}^{23}\!/_4, \lambda_2 = {}^{13}\!/_4$. This solution is the optimum solution as it does not violate any of the Kuhn-Tucker conditions.

$$x_1 = {}^{17}\!/_4, x_2 = {}^{23}\!/_4, \lambda_1 = 0, \lambda_2 = {}^{13}\!/_4$$

$$\text{Max Z} = 1894/16.$$

Example 5.6: Determine x_1, x_2, x_3 so as to

Maximize $Z = -x_1^2 - x_2^2 - x_3^2 + 4x_1 + 6x_2$

Subject to the constraints,

$$x_1 + x_2 \leq 2$$

$$2x_1 + 3x_2 \leq 12$$

$$x_1, x_2 \geq 0$$

Solution: Given,

$$f(x) = -x_1^2 - x_2^2 - x_3^2 + 4x_1 + 6x_2$$

$$g_1(x) = x_1 + x_2 - 2$$

$$g_2(x) = 2x_1 + 3x_2 - 12$$

Define Lagrangian function $L(x, \lambda, S)$

$$= f(x) - \lambda_1\left(g_1(x) + s_1^2\right) - \lambda_2\left(g_2(x) + s_2^2\right)$$

Where s_1, s_2 are slack variables and λ_1, λ_2 are Lagrangian multipliers.

The Kuhn-Tucker conditions are given by:

(1) $\dfrac{\partial f}{\partial x_j} - \sum_{i=1}^{2} \lambda_i \dfrac{\partial g_i}{\partial x_j} = 0 \quad j = 1,2,3$

(i) $-2x_1 + 4 = \lambda_1 + 2\lambda_2$

(ii) $-2x_2 + 6 = \lambda_1 + 3\lambda_2$

(iii) $-2x_2 = 0$

(2) $\lambda_i g_i(x) = 0$

gives

(i) $\lambda_1\left(x_1 + x_2 - 2\right) = 0$

(ii) $\lambda_2\left(2x_1 + 3x_2 - 12\right) = 0$

(3) $g_i(x) \le 0$

(i) $x_1 + x_2 - 2 \le 0$

(ii) $2x_1 + 3x_2 - 12 \le 0$

(4) $\lambda_1 \ge 0 \; \lambda_2 \ge 0$

We have four different cases.

Case I: $\lambda_1 = 0, \lambda_2 = 0$ (i), (ii), (iii) of (1) yield $x_1 = 2, x_2 = 3, x_3 = 0$. This solution violates the inequalities of (3).

Case II: $\lambda_1 = 0, \lambda_2 \neq 0$. In this case (ii) of (2) will give $2x_1 + 3x_2 = 12$ and (i) and (ii) of (1) gives

$-2x_1 + 4 = 2\lambda_2, -2x_2 + 6 = 3\lambda_2$. The solution to these simultaneous equation gives

$x_1 = {}^{2}\!/_{13}, x_2 = {}^{3}\!/_{13}, \lambda_2 = {}^{24}\!/_{13} > 0$ also (iii) of (1) gives $x_3 = 0$. However this solution violates (i) of (3). So this solution is discarded.

Case III: . $\lambda_1 \neq 0, \lambda_2 \neq 0$. In this case (2) (i) and (ii) gives $x_1 + x_2 = 2$ and $2x_1 + 3x_2 = 12$. These equations give $x_1 = -6, x_2 = 8$. Then (1) (i), (ii) and (iii) yield $x_3 = 0, \lambda_1 = 68, \lambda_2 = -26$.

Since $\lambda_2 = -26$ violates the condition (4) so this solution is also discarded.

NOTES

Case IV: $\lambda_1 \neq 0, \lambda_2 = 0$. In this case (2) (i) gives $x_1 + x_2 = 0$. Along with (1) (i) and (ii) $-2x_1 + 4 = \lambda_1 + 2\lambda_2$ $-2x_2 + 6 = \lambda_1 + 3\lambda_2$ give $x_1 = {}^1\!/_2, x_2 = {}^3\!/_2$ and $\lambda_1 = 3 > 0$. Further from (3) (iii) $x_3 = 0$. This solution does not violate any of the Kuhn-Tucker conditions.

Hence the optimum solution is $x_1 = \frac{1}{2}, x_2 = \frac{3}{2}, x_3 = 0, \lambda_1 = 3, \lambda_2 = 0$. Max $Z = \frac{17}{2}$.

Exercise

1. Solve the following Nonlinear programming problem using Kuhn-Tucker conditions.

Maximize $Z = x_1^2 + x_1\, x_2 - 2x_2^2$

Subject to, $4x_1 + 2x_2 \leq 24$

$x_1, x_2 \geq 0$

(**Ans:** $x_1 = 6,\ x_2 = 0, \lambda_1 = 3, Z = 36$)

2. Solve the following Nonlinear programming problem using Kuhn-Tucker conditions.

Maximize $Z = 8x_1 + 10x_2 - x_1^2 - x_2^2$

Subject to, $3x_1 + 2x_2 \leq 6$

$x_1, x_2 \geq 0$

(**Ans:** $x_1 = \frac{4}{3}, x_2 = \frac{33}{13}$ Max $Z = 21.3$)

5.3.1 Quadratic Programming

Quadratic Programming (QP) is a Nonlinear Programming (NLP). It is an optimization problem of special type in which a quadratic function of many variables is optimized (maximized or minimized), subject to constraints of linear nature on these variables, using mathematical approach. In this type of programming problem objective function is Nonlinear but constraints are linear. Mathematical representation of QP is given as below:

$$\text{Optimize (Max or Min) Z} = \sum_{j=1}^{n} c_j x_j + \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} x_j d_{jk} x_k$$

Subject to constraints,

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_{i,} \quad i = 1, 2, ..., n \qquad \text{...(5.10)}$$

and $$x_j \geq 0,\ j = 1, 2, ..., n \qquad \text{...(5.11)}$$

Using matrix notations, same can be written as,

$$\text{Optimize (Max or Min) } Z = CX + \frac{1}{2}X^T DX$$

Subject to constraints,

$$AX \leq b \qquad \ldots(5.12)$$

and $$x \geq 0 \qquad \ldots(5.13)$$

Where, X^T shows transpose of matrix X. Constraint $AX \leq b$ shows that each entry of the vector AX is less than or equal to the corresponding entry of the vector b.

$X = (x_1, x_2, ..., x_n)^T$; $C = (c_1, c_2, ..., c_n)$; $b = (b_1, b_2, ..., b_m)^T$

$D = [d_{jk}]$ is an $n \times n$ symmetric matrix, i.e., $d_{jk} = d_{kj}$; $A = [a_{ij}]$ is an $m \times n$ matrix

The matrix D is symmetric as well as positive-definite (i.e., the quadratic term $X^T DX$ in x is positive for all values of x but $x \neq 0$ in case it is minimization type and it is negative-definite (i.e., $x^T Dx < 0$ for all values of x but $x \neq 0$ in case the problem is of the maximization type. Objective function of the quadratic programming problem is strictly convex in x for minimization and concave for maximization. If D is null matrix, then QP problem reduces to the standard LP problem.

5.3.2 Karush–Kuhn–Tucker Conditions

Karush–Kuhn–Tucker conditions are also known as KKT conditions or as Kuhn-Tucker conditions. As a mathematical approach to such problems, conditions that are necessary for finding solution to a NLP that is optimal, if some conditions of regularity are satisfied. This method is a generalization of Lagrange's multipliers applied to inequality constraints. These conditions have been named on its contributors W. Karush, H.W. Kuhn, and A.W. Tucker. Necessary and sufficient KKT conditions for finding optimal solution for maximizing quadratic function with linear constraints are derived as below:

Step 1: Introducing slack variables s_i^2 and r_j^2 to constraints Equations (5.12) and (5.13), the problem becomes

$$\text{Max } f(x) = \sum_{j=1}^{n} c_j x_j + \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} x_j d_{jk} x_k$$

Subject to constraints

$$\sum_{j=1}^{n} a_{ij} x_j + s_i^2 = b_i; \quad i = 1, 2, ..., m$$

$$-x_j + r_j^2 = 0; \quad j = 1, 2, ..., n$$

Step 2: Forming the Lagrange function as follows:

$$L\ (x, S, r, \lambda, \mu) = f(x) - \sum_{j=1}^{n} \lambda_i \left\{a_{ij} x_j + S_i^2 - b_i\right\} - \sum_{j=1}^{n} \mu_j \left\{-x_j + r_j^2\right\}$$

Step 3: Differentiate $L(x, S, r, \lambda, \mu)$ partially with respect to x, S, r, λ and μ. Then equating these derivatives to zero we get the required necessary KKT conditions. That is,

NOTES

(i) $C-\frac{1}{2}(2X^T D)-\lambda A+\mu=0$

or, $c_j-\sum_{k=1}^{n} x_k d_{jk}-\sum_{i=1}^{m}\lambda_i a_{ij}+\mu_j=0; \quad j=1,2,\ldots,n$

(*ii*) $-2\lambda S = 0$ or $\lambda_i S_i^2 = 0$

or, $\lambda_i\left\{\sum_{j=1}^{n} a_{ij}x_j-b_i\right\}=0; \quad i=1,2,\ldots,m$

(*iii*) $-2\mu r = 0$ or $\mu_j r_j = 0; \quad j = 1, 2, ..., n$

$\mu_j x_j = 0, \; j = 1, 2, ..., n$

(*iv*) $AX + S^2 - b = 0$; i.e., $AX \le b$

or, $\sum_{j=1}^{n} a_{ij}x_j \le b_i, \; i=1,2,\ldots,m$

(*v*) $x + r^2 = 0$, i.e., $x \ge 0$
or, $x_j \ge 0, \quad j = 1, 2, ..., n$

(*vi*) $\lambda_i, \mu_j, x_j, S_i, r_j \ge 0$

Conditions (*i*) and (*iv*) are linear constraints involving $2(n + m)$ variables. The condition $\mu_j x_j = \lambda_i S_i = 0$ implies that both x_j and μ_j as well as S_i and λ_i cannot be basic variables at a time in a non-degenerate basic feasible solution. The conditions $\mu_j x_j = 0$ and $\lambda_i S_i = 0$ are also known as *complementary slackness conditions*.

5.3.3 Wolfe's Modified Simplex Method

This method for solution of a Quadratic Programming Problem (QPP) is Summarized below:

Step 1: Introducing artificial variables A_j ($j = 1, 2, ..., n$) in condition (*i*) we get,

$$c_j-\sum_{k=1}^{n} x_k d_{jk}-\sum_{i=1}^{m}\lambda_i a_{ij}+\mu_j+A_j=0$$

To get initial basic feasible solution, we shall have $x_j = 0$, $\mu_j = 0$, $A_j = -c_j$ and $S_i^2 = b_i$. However, for any real problem, this solution would be desirable if and only if $A_j = 0$ for all j.

Step 2: Use Phase I of the simplex method for checking for feasibility of the constraints $AX \le b$. In case feasible solution does not exist, terminate. Else, get an initial basic feasible solution for Phase II. To get desired feasible solution solve the following problem:

Minimize $Z = \sum_{j=1}^{n} A_j$

NOTES

Subject to constraints,

$$\sum_{k=1}^{n} x_k d_{jk} + \sum_{i=1}^{m} \lambda_i a_{ij} - \mu_j + A_j = -c_j;\ \ j = 1, 2, ..., n$$

$$\sum_{j=1}^{n} a_{ij} x_j + S_i^2 = b_i;\ \ i = 1, 2, ..., m$$

and $\lambda_i, x_j, \mu_j, S_i, A_j \geq 0$ for all i and j

$$\left.\begin{aligned} \lambda_i S_i &= 0 \\ \mu_j x_j &= 0 \end{aligned}\right\} \text{Complementary slackness conditions}$$

Thus, to find a variable that enters the basis at each iteration, the complementary slackness conditions have to be satisfied.

There are $2(m+n)$ variables in this problem with $(m+n)$ linear constraints and $(m+n)$ complementary slackness conditions.

Step 3: Next, use Phase II of the simplex method to find optimal solution of this problem of Step 2. The solution obtained this way will also be an optimal solution of QPP

Example 5.7: Use Wolfe's method to solve the given QPP:

Maximize $Z = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$

Subject to constraints,

$$x_1 + 2x_2 \leq 2$$

and $x_1, x_2 \geq 0$

Solution: Consider non-negativity conditions $x_1, x_2 \geq 0$ as inequality constraints. Add slack variables to all inequality constraints to convert them into equations. The standard form of QP problem changes to,

Maximize $Z = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$

Subject to constraints,

$$x_1 + 2x_2 + S_1^2 = 2$$

$$-x_1 + r_1^2 = 0$$

$$-x_2 + r_2^2 = 0$$

and $x_1, x_2, S_1, r_1, r_2 \geq 0$

To get necessary conditions, we construct the Lagrange function as follows:

$L(x_1, x_2, S_1, \lambda_1, \mu_1, \mu_2, r_1, r_2) = (4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2) - \lambda_1 (x_1 + 2x_2 + S_1^2 - 2)$

$$-\mu_1 (-x_1 + r_1^2) - \mu_2 (-x_2 + r_2^2)$$

NOTES

The necessary and sufficient conditions for maximum value of L and hence of Z are found from:

$$\frac{\partial L}{\partial x_1} = 4 - 4x_1 - 2x_2 - \lambda_1 + \mu_1 = 0 \; ; \qquad \frac{\partial L}{\partial x_2} = 6 - 2x_1 - 4x_2 - 2\lambda_1 + \mu_2 = 0$$

$$\frac{\partial L}{\partial \lambda_1} = x_1 + 2x_2 + S_1^2 - 2 = 0 \qquad ; \qquad \frac{\partial L}{\partial s_1} = 2\lambda_1 S_1 = 0$$

$$\frac{\partial L}{\partial \mu_1} = -x_1 + r_1^2 = 0 \qquad ; \qquad \frac{\partial L}{\partial \mu_2} = -x_2 + r_2^2 = 0$$

$$\frac{\partial L}{\partial r_1} = 2\mu_1 r_1 = 0 \qquad ; \qquad \frac{\partial L}{\partial r_2} = 2\mu_2 r_2 = 0$$

After simplification, we get

$$4x_1 + 2x_2 + \lambda_1 - \mu_1 = 4; \qquad 2x_1 + 4x_2 + 2\lambda_1 - \mu_2 = 6$$

$$x_1 + 2x_2 + S_1^2 = 2$$

$$\left.\begin{aligned} \lambda_1 S_1 &= 0 \\ \mu_1 x_1 = \mu_2 x_2 &= 0 \end{aligned}\right\} \text{(Complementary conditions)}$$

and $\quad x_1, x_2, \lambda_1, \mu_1, \mu_2, S_1 \geq 0$

Introduce artificial variables A_1 and A_2 in the first two constraints respectively. Modified LP problem changes to,

Minimize $Z^* = A_1 + A_2$

Subject to constraints,

$$\begin{aligned} 4x_1 + 2x_2 + \lambda_1 - \mu_1 + A_1 &= 4 \\ 2x_1 + 4x_2 + 2\lambda_1 - \mu_1 + A_2 &= 6 \\ x_1 + 2x_2 + S_1^2 &= 2 \end{aligned}$$

and $\quad x_1, x_2, \lambda_1, \mu_1, \mu_2, A_1, A_2 \geq 0$

The initial basic feasible solution to this LP problem is shown below:

		C_j	0	0	0	0	0	0	–M	–M	
C_B	B	x_B	x_1	x_2	λ_1	μ_1	μ_2	s_1	A_1	A_2	Min x_B/x_i
$-M$	A_1	4	4	2	1	–1	0	0	1	0	4/4 = 1←
$-M$	A_2	6	2	4	2	0	–1	0	0	1	6/2 = 3
0	s_1	2	1	2	0	0	0	1	0	0	2/1 = 2
	Z^*_j	$-10M$	$-6M$	$-6M$	$-3M$	M	$-M$	$-M$	M	0	
	$Z^*_j - C_j$		$-6M$↑	$-6M$	$-3M$	M	M	0	0	0	

0	x_1	1	1	1/2	1/4	–1/4	0	0	–	0	1/1/2 = 2
–M	A_2	4	0	3	3/2	1/2	–1	0	–	1	4/3
0	s_1	1	0	3/2	–1/4	1/4	0	1	–	0	1/3/2 = 2/3←
	Z^*_j	$-4M$	0	$-3M$	$-3/2M$	$-1/2M$	M	0	$-M$		
	$Z^*_j - C_j$		0	$-3M$↑	$-3/2M$	$-1/2M$	M	0	$-M$		
0	x_1	2/3	1	0	1/3	–1/3	0	–1/3	–	0	
$-M$	A_2	2	0	0	2	0	–1	–2	–	1	
0	x_2	2/3	0	1	–1/46	1/6	0	2/3	–	0	
	Z^*_j	$-2M$	0	0	$-2M$	0	M	$2M$	–	$-M$	
	$Z^*_j - C_j$		0	0	$-2M$↑	0	M	$2M$	–	0	
0	x_1	1/3	1	0	0	–1/3	1/6	0	–	–	
0	λ_1	1	0	0	1	0	–1/2	–1	–	–	
0	x_2	5/6	0	1	0	1/6	–1/12	1/2	–	–	
$Z_j - C_j$	Z^*_j	0	0	0	0	0	0	0	–	–	
	$Z^*_j - C_j$	0	0	0	0	0	0	0	–	–	

Since all $Z^*_j - C_j = 0$, hence an optimal solution for Phase I is reached. The optimal solution is:

$$x_1 = 1/3,\ x_2 = 5/6,\ \lambda_1 = 1,\ \lambda_2 = 0,\ \mu_1 = \mu_2 = 0,\ S_1 = 0$$

This solution also satisfies the complementary conditions: $\lambda_1 S_1 = 0$; $\mu_1 x_1 = \mu_2 x_2 = 0$ and the restriction on the signs of Lagrange multipliers, λ_1, μ_1 and μ_2.

Further, as Z* = 0, implies that the current solution is also feasible. Thus, the maximum value of objective function is,

$$\text{Max } Z = 4x_1 + 6x_2 - 2\,x_1^2 - 2x_1x_2 - 2x_2^2$$

$$= 4(1/3) + 6(5/6) - 2(1/3)^2 - 2(1/3)\,(5/6) - 2(5/6)^2 = 25/6$$

Example 5.8: Use Wolfe's method to solve the QPP, given below:

$$\text{Maximize } Z = 2x_1 + x_2 - x_1^2$$

Subject to constraints,

$$2x_1 + 3x_2 \le 6$$

$$2x_1 + x_2 \le 4$$

and $$x_1,\ x_2 \ge 0$$

Solution: Considering non-negativity conditions $x_1, x_2 \ge 0$ as inequality constraints. With the addition of slack variables to all inequalities to convert these to equations. After adding slack variables it take the form,

$$\text{Maximize } Z = 2x_1 + x_2 - x_1^2$$

Subject to constraints,

$$2x_1 + 3x_2 + S_1^2 = 6$$

$$2x_1 + x_2 + S_2^2 = 4$$

$$-x_1 + r_1^2 = 0$$

NOTES

$$-x_2 + r_2^2 = 0$$

Forming the Lagrange function as follows:

$$L\,(x_1, x_2, S_1, S_2, \lambda_1, \lambda_2, \mu_1, \mu_2, r_1, r_2) = (2x_1 + x_2 - x_1^2) - \lambda_1\,(2x_1 + 3x_2 + S_1^2 - 6)$$
$$- \lambda_2\,(2x_1 + x_2 - S_2^2 - 4) - \mu_1\,(-x_1 + r_1^2) - \mu_2\,(-x_2 + r_2^2)$$

The necessary and sufficient conditions for maximum of L and hence of Z are

$$\frac{\partial L}{\partial x_1} = 2 - 2x_1 - 2\lambda_1 - 2\lambda_2 + \mu_1 = 0 \quad ; \quad \frac{\partial L}{\partial x_2} = 1 - 3\lambda_1 - \lambda_2 + \mu_2 = 0$$

$$\frac{\partial L}{\partial s_1} = -2\lambda_1 S_1 = 0 \quad ; \quad \frac{\partial L}{\partial s_2} = -2\lambda_2 S_2 = 0$$

$$\frac{\partial L}{\partial r_1} = -2\mu_1 r_1 = 0 \quad ; \quad \frac{\partial L}{\partial r_2} = -2\mu_2 r_2 = 0$$

$$\frac{\partial L}{\partial \lambda_1} = 2x_1 + 3x_2 + S_1^2 - 6 = 0 \quad ; \quad \frac{\partial L}{\partial \lambda_2} = 2x_1 + x_2 + S_2^2 - 4 = 0$$

$$\frac{\partial L}{\partial \mu_1} = -x_1 + r_1^2 = 0 \quad ; \quad \frac{\partial L}{\partial \mu_2} = -x_2 + r_2^2 = 0$$

After simplifying these conditions, we get

$$2x_1 + 2\lambda_1 + 2\lambda_2 - \mu_1 = 2; \quad 3\lambda_1 + \lambda_2 - \mu_2 = 1$$

$$2x_1 + 3x_2 + S_1^2 = 6; \qquad 2x_1 + x_2 + S_2^2 = 4$$

$$\lambda_1 S_1 = \lambda_2 S_2 = 0; \qquad \mu_1 x_1 = \mu_2 x_2 = 0$$

and$x_1, x_2, \lambda_1, \lambda_2, \mu_1, \mu_2, S_1, S_2 \geq 0$

Introduce the artificial variables A_1 and A_2 in the first two constraints respectively. Then the modified QP problem becomes,

Minimize $Z^* = A_1 + A_2$

Subject to constraints,

$$2x_1 + 2\lambda_1 + 2\lambda_2 - \mu_1 + A_1 = 2; \quad 3\lambda_1 + \lambda_2 - \mu_2 + A_2 = 1$$

$$2x_1 + 3x_2 + s_1^2 = 6; \qquad 2x_1 + x_2 + s_2^2 = 4$$

and $x_1, x_2, s_1, s_2, A_1, A_2, \mu_1, \mu_2 \geq 0$

$$\lambda_1 s_1 = \lambda_2 s_2 = 0;$$
$$\mu_1 x_1 = \mu_2 x_2 = 0$$

The initial basic feasible solution to this QP (Quadractic Programming) problem is shown as follows.

		C_j	0	0	0	0	0	0	0	0	−M	−M	*Min* x_B/x_i
C_B	B	x_B	x_1	x_2	λ_1	λ_2	μ_1	μ_2	S_1	S_2	A_1	A_2	
$-M$	A_1	2	2	0	2	2	–1	0	0	0	1	0	2/2=1←
$-M$	A_2	1	0	0	3	1	0	–1	0	0	0	1	—
0	s_1	6	2	3	0	0	0	0	1	0	0	0	6/2=3
0	s_2	4	2	1	0	0	0	0	0	1	0	0	4/2=2
	Z^*_j	$-3M$	$-2M$	0	$-5M$	$-3M$	M	M	0	0	0	0	–M/–M
	$Z^*_j - C_j$		$-2M\uparrow$	0	$-5M$	$-3M$	M	M	0	0	0	0	
0	x_1	1	1	0	1	1	–1/2	0	0	0	0	$1x_2$	1/1 = 1
$-M$	A_2	1	0	0	3	1	0	–1	0	0	1	1/3→	1/3←
0	s_1	4	0	3	–2	–2	1	0	1	0	0	—	—
0	s_2	2	0	1	–2	–2	1	0	0	1	0	—	—
	Z^*_j	$-M$	0	0	$-3M$	$-M$	0	M	0	0	$-M$		
	$Z^*_j - C_j$		0	0	$-3M\uparrow$	$-M$	0	M	0	0	0		
0	x_1	2/3	1	0	0	2/3	–1/2	1/3	0	0	–	–	
0	λ_1	1/3	0	0	1	1/3	0	–1/3	0	0	–	–	
0	s_1	14/3	0	3	0	–4/3	1	–2/3	1	0	–	–	
0	s_2	8/3	0	1	0	–4/3	1	–2/3	0	1	–	–	
	Z^*_j	0	0	0	0	0	0	0	0	0	–	–	—
	$Z^*_j - C_j$	0	0	0	0	0	0	0	0	0	–	–	—

Sinc $Z^*_j - C_j = 0$, therefore an optimal solution for Phase I is reached. The optimal solution is:

$x_1 = 2/3$, $x_2 = 0$, $\lambda_1 = 1/3$, $\lambda_2 = 0$, $\mu_1, = \mu_2 = 0$, $S_1 = 14/3$ and $S_2 = 8/3$

This solution also satisfies complementary slackness conditions:

$\lambda_1 S_1 = \lambda_2 S_2 = 0$; $\mu_1 x_1 = \mu_2 x_2 = 0$ and the restriction on the signs of Lagrange multipliers: λ_1, λ_2, μ_1 and μ_2. Since, $Z^* = 0$, the current solution is also feasible. The maximum value of the objective function of the given quadratic problem is:

$$\text{Max } Z = 2x_1 + x_2 - x_1^2 = 2 \times \frac{2}{3} + 0 - \left(\frac{2}{3}\right)^2 = \frac{4}{3} - \frac{4}{9} = \frac{12-4}{9} = \frac{8}{9}$$

5.3.4 Beale's Method

This method uses results to solve based on calculus quadratic programming problem instead of Kuhn-Tucker conditions. General QPP is of the form,

Minimize $Z = CX + 1/2S^T DX$ …(5.14)

Subject to constraints, …(5.15)

$$AX = b$$

and $x \geq 0$ …(5.16)

Where $x \in E^n$, $b \in E^m$, $c \in E^n$, D is a symmetric $n \times n$ matrix and A is an $m \times n$ matrix.

Beale's method starts by making partitions of n variables in QP problem into basic and non-basic variables at every iteration of the solution process, and expressing basic variables and objective function in terms of non-basic variables. Let b be any $m \times m$ non-singular matrix containing columns of A corresponding to the basic variables $x_B \in E^m$ and let N be an $m \times (n-m)$ matrix which containing of A corresponding to non-basic variables $x_N \in E^{n-m}$. Then, Equation (5.15) may be written as:

$$[B, N]\begin{bmatrix} X_B \\ X_N \end{bmatrix} = b$$

$$BX_B + NX_N = b \quad \text{or} \quad XB = B^{-1} b - \mathrm{B}^{-1} NX_N$$

or, $$x_{B_i} = y_{i0} - \sum_{j=1}^{n-m} y_{ij} x_{N_j}\,; \; i = 1, 2, ..., m \qquad \ldots(5.17)$$

Where, $y_{i0} = (y_{10}, y_{20}, ..., y_{m0})^T = B^{-1}b$ and $y_{ij} = B^{-1}N$

For the current basic feasible solution $x_{N_j} = 0$ ($j = 1, 2, ..., n - m$), we have $x_{N_j} = y_{i0}$, ($i = 1, 2, ..., m$). Assuming that $y_{i0} \geq 0$.

The objective function Equation (5.14) in terms of X_B and X_N can be written as,

$$Z = [C_B, C_N]\begin{bmatrix} X_B \\ X_N \end{bmatrix} + \frac{1}{2}\begin{bmatrix} X_B^T, X_N^T \end{bmatrix}\begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}\begin{bmatrix} X_B \\ X_N \end{bmatrix}$$

Expressing Z in terms of the remaining $(n-m)$ non-basic variables X_N only and carrying out, we get,

$$Z = Z_0 + \alpha X_N + X_N^T G X_N \qquad \ldots(5.18)$$

Where, Z_0 = Value of objective function Z when $X_N = 0$ and $x_{B_i} = y_{i0}$

G = Symmetric matrix of order $(n-m) \times (n-m)$

$\alpha = \alpha_1, \alpha_2, ..., \alpha_{n-m}$ (constant)

Step 1: Evaluate the partial derivatives of Z with respect to non-basic variables $X_N = 0$. Thus from Equation (5.18) we get,

$$\frac{\partial Z}{\partial x_{N_j}} = \alpha_j + 2\sum_{k=1}^{n-m} g_{jk} x_{Nk}\,; \; j = 1, 2, ..., n-m \qquad \ldots(5.19)$$

Step 2: See the nature of $\left|\frac{\partial Z}{\partial x_{N_j}}\right|_{x_{N=0}} = \alpha_j; \quad k = 1, 2, ..., n - m$

(*i*) If $\alpha_j < 0$, for all j, then the current solution is also an optimal solution

(*ii*) But if at least one $\alpha_j > 0$, then one of the non-basic variables which is currently at zero level corresponding to largest positive value of α_j, will be selected to enter into the basis.

Step 3: $\left|\dfrac{\partial Z}{\partial x_N}\right|_{x_{N=0}} = \alpha_r$ (maximum), then choose non-basic variable x_r for entering into the basis. For this it will be profitable to go on increasing its value from zero to a points, till either

(*i*) Any one of the present basic variables becomes negative.

(*ii*) $\partial Z / \partial x_{N_j}$ reduces to zero and is about to become negative.

Step 4: To maintain feasibility of the solution we have to consider only that value of non-basic variable x_r, say β_1, which has positive coefficient. In this case, the first basic variable for leaving the basis has to satisfy the usual minimum ratio rule used in simplex method. This is given by,

$$\beta_1 = \begin{cases} \text{Min}\left\{\dfrac{y_{i0}}{y_{ij}}; y_{ij} > 0\right\} \\ \infty \qquad ; y_{ij} \le 0, \quad j = 1, 2, ..., n-m \end{cases} \qquad \ldots(5.20)$$

Where, $y_{i0} = x_{B_l}$

Since it is not desirable to increase the value of the non-basic variable x_r beyond the point where $\partial Z / \partial x_{N_j}$ becomes zero, the critical value of x_r say β_2, where $\partial Z / \partial x_{N_j}$ is zero is given as,

$$\beta_2 = \begin{cases} \dfrac{|\alpha_j|}{2g_{jj}}; & g_{jj} > 0 \\ \infty \; ; & g_{jj} \le 0 \end{cases}$$

Where, g_{jj} is the element of matrix G.

Value of non-basic variable x_r is found by taking either β_1 or β_2, whichever is minimum.

Thus, $\qquad x_r = \text{Min}\ \{\beta_1, \beta_2\}$

If $\beta_1 = \beta_2 = \infty$, the value of x_r may be increased indefinitely without violating either of the conditions (*i*) and (*ii*) of Step (3) and the QP problem must have an unbounded solution. Further,

(*i*) If entering variable x_r is increased up to β_1 only and at least one basic variable is reduced to zero, then a new basic feasible solution may be obtained using simplex method. But if by entering x_r into the basis two or more basic

NOTES

variables are reduced to zero, then the new solution so obtained will be degenerate and thus cycling may occur.

(*ii*) If entering variable increases to β_2 ($< \infty$), then we may get more than m variables at positive level at any iteration. This stage comes when the new (non-basic) feasible solution occur at $\partial Z / \partial x_{N_j} = 0$. At this stage we define a new unrestricted variable u_j given as,

$$u_j = \frac{\partial Z}{\partial x_r} = \alpha_j + 2\sum_{k=1}^{n-m} g_{jk} x_{Nk}$$

The variable u_j is also known as free variable. Clearly, now there are $m + 1$ non-zero variables and $m + 1$ constraints. These variables form a basic feasible solution for the new set of constraints:

$$AX = b$$

$$u_j = -2\sum_{k=1}^{n-m} g_{jk} x_{Nk} = \alpha_j$$

The variable u_j is introduced in the set of constraints only for the purpose of computations having zero value at the next basic feasible solution. Now variables X_B and u_j are treated as basic variables. The new set of constraints is again expressed in terms of non-basic variables to get the new basic feasible solution.

Step 5: Go to Step (1) and repeat the entire procedure to get a new basic feasible solution until no further improvement in the objective function obtained by making any permitted changes in one of the non-basic variables. The permitted changes here include increase in every variable and decrease in free variables. The procedure terminates when

$$\frac{\partial Z}{\partial x_{N_j}} \begin{cases} \leq 0, \text{ if } x_{N_j} \text{ is a restricted (non-negative) variable} \\ = 0, \text{ if } x_{N_j} \text{ is a free variable} \end{cases} \qquad \ldots(5.21)$$

Necessary conditions Equation (5.21) for terminating the procedure are also sufficient for a global minimum if D is positive semi-definite or positive definite.

Notes:
1. While evaluating $\partial Z / \partial u_j$, both increase and decrease showed be checked, as u_j is unrestricted in sign.
2. If at any iteration a free variable becomes a basic variable and is non-zero, then drop the new constraint that contains it. This should be done because it is a free variable and therefore, is neither chosen to leave the basis nor appears in the selection of leaving variable.

Example 5.9: Use Beale's method to solve the given QPP:

Maximize $Z = 2x_1 + 3x_2 - 2x_2^2$

Subject to constraints,

$$x_1 + 4x_2 \leq 4$$
$$x_1 + x_2 \leq 2$$

and $\quad x_1, x_2 \geq 0$

Solution: After introducing slack variables s_1 and s_2, the given constraints can be written as:

$$x_1 + 4x_2 + s_1 = 4$$
$$x_1 + x_2 + s_2 = 2$$
$$x_1, x_2, s_1, s_2 \geq 0$$

Making S_1 and S_2 basic variables in initial solution and expressing these in terms of non-basic variables x_1 and x_2 as follows:

$$S_1 = 4 + 1\,(-x_1) + 4\,(-x_2)$$
$$S_2 = 2 + 1\,(-x_1) + 1\,(-x_2)$$

We choose the initial basic feasible solution: $x_1 = x_2 = 0$; $S_1 = 4$ and $S_2 = 2$. This solution is shown in Table 5.3.

***Table 5.3** Initial Solution*

B	x_B	x_1	x_2	S_1	S_2
s_1	*4*	*1*	*4*	*1*	*0*
s_2	*2*	*1*	*1*	*0*	*1*

The initial value of the objective function at this solution Z = 0.

Also $X_B = (s_1, s_2) = (4, 2)$ and $X_N = (x_1, x_2) = (0, 0)$. Expressing Z in terms of non-basic variables x_1 and x_2, we get,

$$Z = 2x_1 + 3x_2 - 2x_2^2$$

$$\frac{\partial Z}{\partial x_1} = 2, \quad \frac{\partial Z}{\partial x_2} = 3 - 4x_2$$

At the curent basic feasible solution evaluate these partial derivatives of Z with respect to $X_N = 0$, i.e., $x_1 = x_2 = 0$

$$\left.\frac{\partial Z}{\partial x_1}\right|_{\substack{x_1=0\\x_2=0}} = 2 \quad \text{and} \quad \left.\frac{\partial Z}{\partial x_2}\right|_{\substack{x_1=0\\x_2=0}} = 3$$

Here $\alpha_1 = 2$ and $\alpha_2 = 3$. Both of these are positive; therefore we choose x_2 (due to most positive value of α_2) to enter into the basis to improve the value of the objective function. Using Table 5.3 the critical value β_1 of x_2 is given by:

Case 1: Largest value of x_2 without deriving any basic variable S_1 and S_2 to zero. Since,

$$S_1 = 4 - x_1 - 4x_2$$
$$S_2 = 2 - x_1 - x_2$$

Therefore, $\beta_1 = \text{Min}\,(4/4, 2/1) = 1$, (corresponding to y_{22})

Case 2: The partial derivative $\partial Z / \partial x_2$ becomes zero at $x_2 = 3/4$ ($x_1 = 0$).

Therefore, $$\beta_2 = \frac{|\alpha_2|}{2g_{22}} = \frac{|3|}{2(2)} = \frac{3}{4}$$

NOTES

The new value of the entering variable x_2 is given by,

$$x_2 = \text{Min } (\beta_1, \beta_2) = (1, 3/4) = 3/4$$

This value of x_2 corresponds to β_2; therefore Case (2) applies and neither of the current basic variables becomes zero. Consequently, we introduce a free variable u_1 and the new constraint,

$$u_1 = \frac{\partial Z}{\partial x_2} = 3 - 4x_2 \quad \text{or} \quad 4x_2 + u_1 = 3$$

This is shown in Table 5.2.

It may be noted from Table 5.4 that $X_B = (s_1, s_2, u_1)$ and $X_N = (x_1, x_2)$

Table 5.4 *Introducing a Free Variable u_1*

B	x_B	x_1	x_2	S_1	S_2	u_1
S_1	4	1	4	1	0	0
S_2	2	1	1	0	1	0
u_1	3	0	4	0	0	1

Now introduce x_2 into the basis and remove u_1 from the basis in Table 5.4. The new solution is shown in Table 5.5.

Table 5.5 *Introducing x_2 in the Basis*

B	x_B	x_1	x_2	S_1	S_2	u_1
S_1	1	1	0	1	0	1
S_2	5/4	1	0	0	1	1/4
x_2	3/4	0	1	0	0	–1/4

Now we have,

$$X_B = (s_1, s_2, x_2) = (1, 5/4, 3/4)$$
$$X_N = (x_1, u_1) = (0, 0)$$

Expressing basic variables x_2, s_1 and s_2 in terms of non-basic variables x_1 and u_1 as follows:

$$x_2 = \frac{3}{4} - \frac{1}{4}u_1; \;\; S_1 = 1 - x_1 + u_1; \;\; S_2 = \frac{5}{4} - x_1 + \frac{1}{4}u_1$$

Also eliminating the basic variable x_2 from the objective function and expressing it in terms of non-basic variables x_1 and u_1, we get,

$$Z = 2x_1 + 3\left(\frac{3}{4} - \frac{u_1}{4}\right) - 2\left(\frac{3}{4} - \frac{u_1}{4}\right)^2 = \frac{9}{8} + 2x_1 - \frac{u_1^2}{8}$$

Computing the partial derivatives of Z with respect to x_1 and u_1, we have,

$$\frac{\partial Z}{\partial x_1} = 2; \quad \frac{\partial Z}{\partial u_1} = -\frac{u_1}{4}$$

NOTES

At the current solution, we get,

$$\left.\frac{\partial Z}{\partial x_1}\right|_{\substack{u_1=0\\x_1=0}} = 2 \quad \text{and} \quad \left.\frac{\partial Z}{\partial u_1}\right|_{\substack{u_1=0\\x_1=0}} = 0$$

Here $\alpha_1 = 2$ and $\alpha_2 = 0$. Choosing x_1 to enter with the basis. Using Table 5.5, the critical value β_1 of x_1 is given by,

Case 1: Largest value of x_1 without deriving any basic variable S_1, S_2 and x_2 to zero. Since

$$x_2 = \frac{3}{4} - \frac{1}{4}u_1;\ S_1 = 1 - x_1 + u_1;\ S_2 = \frac{5}{4} - x_1 + \frac{1}{4}u_1$$

Therefore, $$\beta_1 = \text{Min}\left\{\frac{1}{1}, \frac{(5/4)}{1}\right\} = 1$$

Case 2: Since partial derivative $\partial Z / \partial x_1$ is non-zero, therefore $\beta_2 = 2$

Thus, the new value of the entering variable x_1 is: $x_1 = \text{Min}\ \{\beta_1, \beta_2\} = 1$. This value of x_1 corresponds to β_1, therefore Case (1) applies and the new optimal solution is shown in Table 5.6.

Table 5.6 *The New Optimal Solution*

B	x_B	x_1	x_2	S_1	S_2	u_1
x_1	***1***	***1***	***0***	***1***	***0***	***–1***
S_2	***1/4***	***0***	***0***	***–1***	***1***	***–3/4***
x_2	***3/4***	***0***	***1***	***0***	***0***	***–1/4***

Now we have $X_B = (x_1, S_2, x_2) = (1, 1/4, 3/4)$ and $X_N = (S_1, u_1) = (0, 0)$

Expressing basic variables x_1, x_2 and s_2 in terms of non-basic variables S_1 and u_1 as follows:

$$x_1 = 1 - S_1 + u_1;\ S_2 = \frac{1}{4} + S_1 - \frac{3}{4}u_1;\ x_2 = \frac{3}{4} - \frac{1}{4}u_1$$

Also expressing objective function Z in terms of non-basic variables s_1 and u_1, we get,

$$Z = \frac{25}{8} + 2u_1 - 2s_1 - \frac{1}{8}u_1^2$$

Computing partial derivatives of Z with respect to s_1 and u_1, we have

$$\frac{\partial Z}{\partial s_1} = -2;\ \frac{\partial Z}{\partial u_1} = -2 - \frac{1}{4}u_1$$

NOTES

But at the current solution, we have

$$\left.\frac{\partial Z}{\partial s_1}\right|_{\substack{s_1=0\\ x_1=0}} = -2 \quad \text{and} \quad \left.\frac{\partial Z}{\partial u_1}\right|_{\substack{s_1=0\\ u_1=0}} = -2$$

Since both $\alpha_j < (j = 1, 2)$, the optimal solution is: $x_1 = 1, x_2 = 3/4$ and Max Z = 25/8

5.4 TYPES OF NONLINEAR PROGRAMMING PROBLEMS

There is a wide variety of Nonlinear Programming Problems (NLPP). Some of the most important types are briefly introduced here. Unlike the linear programming, no standard algorithm like the simplex method can be employed to solve the Nonlinear programming problems. Many different algorithms have been developed to solve the different class of problems. It is beyond the scope of this book to cover all the types of N.L.P.P. and their solutions.

Unconstrained Optimization

In case of unconstrained optimization problems, there are no constraints, but only the objective function. The problem is simply to

$$\text{Maximize / minimize } f(X)$$

over all values of $X = (x_1, x_2,...,xn)$. The necessary condition for a particular solution to be optimal is

$$\frac{\partial f}{\partial x_j} = 0 \text{ at } x = X^* \text{ for } j = 1, 2,..., n,$$

where $f(X)$ is differentiable. This condition is also sufficient condition for maximization when $f(X)$ is a concave function and for minimization when $f(X)$ is a convex function. Thus the solution can be obtained by solving n equations obtained by setting the n partial derivatives equal to zero. For a Nonlinear function, these equations are often Nonlinear, and it becomes impossible to solve these equations analytically. Some search procedures have been developed by researchers to solve such problems.

Linearly Constrained Optimization

As the name indicates, in this type of NLPP, all the constraints $g^i(X)$ are of linear type, while the objective function $f(X)$ is Nonlinear. Since only one Nonlinear function is to be handled, the problem becomes comparatively simple to solve. A number of special algorithms have been developed, by extending the simplex method, to handle the Nonlinear objective function.

5.4.1 Separable Programming

Nonlinear programming (NLP) is an extension of Linear Programming (LP). The principal differences between NLP and LP is that in NLP, the variables which are either in objective function and/or in the constraints occur in higher powers such as (x^2) or in the multiplication form such as ($x_1, x_2 \dots$) with a greater number of variables, i.e., Nonlinear.

Generally, if the objective function $f(x)$ is a function of a single variable say x, the problem of finding maximum/minimum of $f(x)$ is elementary. If x is a function of two variables (x_1, x_2), we can find maximum/minimum for $f(x)$. However, if $x = (x_1, x_2, \dots x_n)$ then for finding maximum/minimum NLP techniques such as Wolfe's method, Beale's method, and Separable programming are used.

NOTES

If the objective function $f(x)$ is not continuous or differentiable, then search methods, whose only requirement is $f(x)$ should be compatible are used to find the extreme. Each of these problems requires different solution procedure.

In this chapter a set of NLP like the following are explained:

- Nonlinear Programming Problem of General Nature
- Quadratic Programming Problem
- Separable Programming Problem

Separable Programming Problem

Separable programming is a special case of convex programming. One additional assumption made here is that all the $f(X)$ and g^i(X) are separable functions.

A function is called separable, when it can be expressed as a sum of subfunctions where each subfunction is a function of one variable only. For example, if $f(X)$ is a separable function, it can be expressed as

$$f(X) = \sum_{i=1}^{n} f_i(x_i),$$

where each $f_i(x_i)$ is a function comprising of terms involving only x_i. Consider the two-variable objective function

$$f(x_1, x_2) = 4x_1 + 9x_1^2 - x_2^2,$$

which can be expressed as

$$f(x_1, x_2) = (4x_1 - x_1^2) + (9x_2 - x_2^2)$$

$$= f(x_1) + f(x_2),$$

where $f(x_1) = 4x_1 - x_1^2,$

and $f(x_2) = 9x_2 - x_2^2$

The separable programming problems are solved by methods which are extensions of the simplex method. They are applicable only if the objective function as well as all the constraints are separable.

5.4.2 Convex and Non-Convex Programming

Convex Programming

Convex programming covers special cases of various types of NLPP, where the objective function $f(X)$ is a concave function and each of the constraints g^i(X) is a convex function. These assumptions ensure that a local maximum is also a global maximum.

NOTES

Non-Convex Programming

All NLPP that do not satisfy the assumptions of convex programming, fall in this category of non-convex programming. There is no algorithm that can result into global optimal solution of such problems. Some algorithms attempt to find the local minima in case the problem does not deviate much from the assumptions of convex programming.

Some specific types of non-convex programming problems can be solved by employing specific methods. Two important types of NLPP under this category are Geometric programming and Fractional Programming.

Lagrange Multiplier Method

This method is used to optimize a continuous and differentiable function subject to equality constraints.

Consider the Nonlinear programming problem

Optimize $Z = f(x)$

Subject to, $G_i\ (x_1, x_2, \dots x_n) = b_i, i = 1, 2, \dots m.$

A modified form of the above model is shown below

Optimize $Z = f(x_1, x_2, \dots x_n)$

Subject to, $g_i\ (x_1, x_2, \dots x_n) = 0, i = 1, 2, \dots m.$

$x_j \geq 0,\ j = 1, 2, \dots n.$

Where, $g_i\ (x_1, x_2, \dots x_n) = G_i\ (x_1, x_2, \dots x_n) - b_i, i = 1, 2, \dots m.$

This problem consists of n variables with m constraints. Multiply each constraint with an unknown variable $\lambda_i\ (i = 1, 2, \dots m)$ and subtract each from the objective function $f(x)$ to be optimized. The new objective function is

$$L\ (x, \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i g_i(x),\ x = (x_1, x_2, \dots x_n)^T$$

Where $m < n$. The function $L\ (x, \lambda)$ is called the *Lagrange function.*

Steps of Lagrange Method

Step 1: Form the Lagrange function as

$$L(x, \lambda) = f(x_1, x_2, \dots x_n) - \sum_{i=1}^{n} \lambda_i g_i(x_1, x_2, \dots x_n).$$

Step 2: Find the partial derivatives of L with respect to x_j and λ of $L\ (x, \lambda)$ and equal to zero given by

$$\frac{\partial L}{\partial x_j} = 0, \quad j = 1, 2, \dots n$$

$$\frac{\partial L}{\partial \lambda_i} = 0, \quad i = 1, 2, \dots m$$

So, the system consists of $n + m$ unknown variables with $n + m$ simultaneous equations of the first order partial derivatives.

Step 3: Find the solution of the system of equation $(n + m)$ obtained from step 2.

Step 4: Form the Bordered Hessian matrix $[H^B]$ of size $n + m$.

Table 5.7 *General form of the Bordered Hessian Matrix*

	1	2	.. I ..	m	1 2 .. j .. n
1	0	0		0	Coefficients of $\frac{\partial L}{\partial \lambda_i} = 0, \quad i = 1, 2, ... m$ Written row wise
2	0	0	0	0	
:	:				
i	0	0	0	0	
:	:				
m	0	0	0	0	
1 2 : j : n	Coefficients of $\frac{\partial L}{\partial \lambda_i} = 0, \quad i = 1, 2, ... m$ written columnwise				$\frac{\partial L}{\partial x_i}\frac{\partial L}{\partial x_j}$ for $i = 1, 2, \dots m$ $j = 1, 2, \dots n$

Step 5: In this step using the sufficient conditions for maxima and minima, the testing for the stationary point $(x_1, x_2, \dots x_n)$ to be optimal solution is given as:

(i) Find the last $(n - 1)$ principal minors of the Bordered Hessian matrix.

(ii) The stationary point will give the maximum objective function value, if the sign of each of the last $(n - m)$ principal minor determinants of the Bordered Hessian matrix is same as that of $(-1)^{m+1}$, ending with the $(2m + 1)^{th}$ principal minor determinant.

(iii) The stationary point will give the minimum objective function value, if the sign of each of the last $(n - m)$ principal minor determinants of the bordered Hessian matrix is same as that of $(-1)^m$ ending with the $(2m + 1)^{th}$ principal minor determinant.

Example 5.10: Solve the Nonlinear programming problem.

Optimize $\quad Z = 4x_1^2 + 2x_2^2 + x_3^2 - 4x_1x_2$

Subject to, $\quad x_1 + x_2 + x_3 = 15,\ 2x_1 - x_2 + 2x_3 = 20;\ x_1, x_2, x_3 \geq 0$

Solution

$$f(x) = 4x_1^2 + 2x_2^2 + x_3^2 - 4x_1x_2$$
$$g_1(x) = x_1 + x_2 + x_3 - 15$$
$$g_2(x) = 2x_1 - x_2 + 2x_3 - 20$$

Construct the Lagrangian function

$$L(x, \lambda) = f(x) - \lambda_1 g_1(x) - \lambda_2 g_2(x) = (4x_1^2 + 2x_2^2 + x_3^2 - 4x_1x_2) - \lambda_1(x_1 + x_2 + x_3 - 15) - \lambda_2 (2x_1 - x_2 + 2x_3 - 20)$$

Using necessary conditions

$$\frac{\partial L}{\partial x_1} = 0 \Rightarrow 8x_1 - 4x_2 - \lambda_1 - 2\lambda_2 = 0$$

$$\frac{\partial L}{\partial x_2} = 0 \Rightarrow 4x_2 - 4x_1 - \lambda_1 + \lambda_2 = 0$$

$$\frac{\partial L}{\partial x_3} = 0 \Rightarrow 2x_3 - \lambda_1 - 2\lambda_3 = 0$$

$$\frac{\partial L}{\partial \lambda_1} = 0 \Rightarrow -(x_1 + x_2 + x_3 - 15) = 0$$

$$\frac{\partial L}{\partial \lambda_2} = 0 \Rightarrow -(2x_1 - x_2 + 2x_3 - 20) = 0$$

Solving these simultaneous equations, we get

$(x_1, x_2, x_3) = \left(\frac{33}{9}, \frac{10}{3}, 8\right)$ and $\lambda_1 = \frac{40}{9}, \lambda_2 = \frac{52}{9}$.

The bordered Hessian matrix at this solution is given by

$$\left[H^B\right] = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & -1 & 2 \\ 1 & 2 & 8 & -4 & 0 \\ 1 & -1 & -4 & 4 & 0 \\ 1 & 2 & 0 & 0 & 2 \end{bmatrix}$$

$$n = 3, m = 2$$

$$\therefore \quad n - m = 1, 2m + 1 = 5$$

$$|H^B| = 72 > 0.$$

The sign of this value $|H^B|$ is same as that of $(-1)^{1+1}$. Hence the solution (x_1, x_2, x_3) corresponds to the maximum point.

$\therefore$ The optimal results are presented as $(x_1, x_2, x_3) = \left(\frac{33}{9}, \frac{10}{3}, 8\right)$

$$Z_{\text{Max}} = \frac{7380}{81} = 91.1$$

Example 5.11: Solve the following problem by using the method of Lagrangian multiplier.

Minimize $\qquad Z = x_1^2 + x_2^2 + x_3^2$

Subject to the constraints

(i) $x_1 + x_2 + 3\,x_3 = 2$

(ii) $5x_1 + 2x_2 + x_3 = 5$ and $x_1, x_2 \geq 0$.

Solution

$$f(x) = x_1^2 + x_2^2 + x_3^2$$
$$g_1(x) = x_1 + x_2 + 3x_3 - 2$$
$$g_2(x) = 5x_1 + 2x_2 + x_3 - 5$$

Construct the Lagrangian function

$$L(x, T\lambda) = f(x) - \lambda_1 g_1(x) - \lambda_2 g_2(x)$$
$$= x_1^2 + x_2^2 + x_3^2 - \lambda_1 (x_1 + x_2 + 3\,x_3 - 2) -$$
$$\lambda_2(5\,x_1 + 2\,x_2 + x_3 - 5)$$

Using the necessary condition

$$\frac{\partial L}{\partial x_1} = 0 \Rightarrow 2x_1 - \lambda_1 - 5\lambda_2 = 0$$

$$\frac{\partial L}{\partial x_2} = 0 \Rightarrow 2x_2 - \lambda_1 - 2\lambda_2 = 0$$

$$\frac{\partial L}{\partial x_3} = 0 \Rightarrow 2x_3 - 3\lambda_1 - \lambda_2 = 0$$

$$\frac{\partial L}{\partial \lambda_1} = 0 \Rightarrow -\left(x_1 + x_2 + 3x_3 - 2\right) = 0$$

$$\frac{\partial L}{\partial \lambda_2} = 0 \Rightarrow -\left(5x_1 + 2x_2 + x_3 - 5\right) = 0$$

Solving these simultaneous equations, we get the solution

$$\left(x_1 x_2 x_3\right) = \left(\frac{37}{46}, \frac{16}{46}, \frac{13}{46}\right) \text{and} \left(\lambda_1 \lambda_2\right) = \left(\frac{3}{23}, \frac{7}{23}\right)$$

$$Min\ Z = \frac{897}{1058}$$

Apply sufficient condition to check for the minimum of Z for which form the bordered Hessian matrix

$$[H^B] = \begin{bmatrix} 0 & 0 & 1 & 1 & 3 \\ 0 & 0 & 5 & 2 & 1 \\ 1 & 5 & 2 & 0 & 0 \\ 2 & 2 & 0 & 2 & 0 \\ 3 & 1 & 0 & 0 & 2 \end{bmatrix}$$

Since $m = 2, n = 3, n - m = 1, 2m + 1 = 5.$

One minor of H^B of order 5 is $H^B \mid H^B| = 460 > 0$.

The sign of this value $|H^B|$ is same as that of $(-1)^m = (-1)^2 = 1$.

$\therefore$ The extreme point (x_1, x_2, x_3) corresponds to minimum of Z.

Optimum solution in Min $Z = \frac{897}{1058}$.

Exercise 5.12: Obtain the solution of the following problems by using the method of Lagrangian multipliers.

1. Min $Z = -2x_1^2 + 5x_1x_2 - 4x_2^2 + 18x_2$

   Subject to, $x_1 + x_2 = 7, \; x_1, x_2 \geq 0$

Solution: $x_1 = 4.95$

$x_2 = 2.045$

Min $Z = 10.5$

2. Maximize $Z = x_1^2 + 2x_2^2 + x_3^2$

   Subject to, $2x_1 + x_2 + 2x_3 = 30$

   $x_1, x_2, x_3 \geq 0$

Solution: $\left(\frac{120}{17}, \frac{30}{17}, \frac{120}{17}\right)$

Max $Z = 105.88$

3. Minimize $Z = 4x_1^2 + 2x_2^2 + x_3^2 - 4x_1x_2$

   Subject to,

   $x_1 + x_2 + x_3 = 15$

   $2x_1 - x_2 + 2x_3 = 20$

   $x_1, x_2, x_3 \geq 0$

Solution: $\left(x_1 = \frac{11}{3}, x_2 = \frac{10}{3}, x_3 = 8\right)$

Max $Z = 91.111.$

Check Your Progress

4. Give the Kuhn-Tucker conditions for constrained optimization.
5. What is quadratic programming?
6. State Karush-Kuhn-Tucker condition.
7. What do you understand by the convex programming?

5.5 ANSWERS TO 'CHECK YOUR PROGRESS'

1. The methods used to find the maximum/minimum of function $f(x)$ depending on whether $f(x)$ is a single variable of $f(x)$, $x = (x_1, x_2 \ldots x_n)$ has several variables are one dimensional
   (i) Search Method (1-D).
   (ii) Direct Search Methods (Interal halving)
   (iii) Fibonacci Sequence Method
   (iv) Golden Section Search Method

2. Fibonacci search is a univariate search technique that can be used to find the maximum (or minimum) of an arbitrary unimodal, univariate objective function. This method is a sequential search technique that successfully reduces the interval in which the maximum (or minimum) of an arbitrary nonlinear function must lie.
3. A prior specification of the resolution factor (∈) and the number of experiments to be performed (N) are required. In search methods to have a proper functioning, the successive experiments will gradually reduce the interval of uncertainty, i.e., the final interval of uncertainty will converge to zero as the number of functional evaluation increases to infinity provided that ∈ is small.

$$\lim_{\substack{N\to\infty \\ \in\to 0}} \{L_2\} = \lim_{\substack{N\to\infty \\ \in\to 0}} \left\{\frac{1}{F_N}\left(L_0 F_{N-1} + \in(-1)^N\right)\right\}$$

$$= L_0\left\{\frac{F_{N-1}}{F_N}\right\}$$

In the above limit, the ratio of $\frac{F_{N-1}}{F_N}$ goes to 0.618 which is known as golden ratio or golden section.

4. The necessary and sufficient conditions for a local optimum of the general Nonlinear programming problem, with both equality and inequality constraints is called Kuhn-Tucker conditions.
5. Quadratic Programming (QP) is a Nonlinear Programming (NLP). It is an optimization problem of special type in which a quadratic function of many variables is optimized (maximized or minimized), subject to constraints of linear nature on these variables, using mathematical approach. In this type of programming problem objective function is Nonlinear but constraints are linear.
6. Karush–Kuhn–Tucker conditions are also known as KKT conditions or as Kuhn-Tucker conditions. As a mathematical approach to such problems, conditions that are necessary for finding solution to a NLP that is optimal, if some conditions of regularity are satisfied. This method is a generalization of Lagrange's multipliers applied to inequality constraints. These conditions have been named on its contributors W. Karush, H.W. Kuhn, and A.W. Tucker.
7. Convex programming covers special cases of various types of NLPP, where the objective function $f(X)$ is a concave function and each of the constraints g^i(X) is a convex function. These assumptions ensure that a local maximum is also a global maximum.

5.6 SUMMARY

- One and multivariable unconstrained optimization methods is used to find the maximum/minimum of function $f(x)$ depending on whether $f(x)$ is a single variable of $f(x)$, $x = (x_1, x_2 \ldots x_n)$ has several variables are one dimensional

(i) Search Method (1-D).

(ii) Direct Search Methods (Interal Halving)

(iii) Fibonacci Sequence Method

(iv) Golden Section Search Method

- Fibonacci search is a univariate search technique that can be used to find the maximum (or minimum) of an arbitrary unimodal, univariate objective function. This method is a sequential search technique that successfully reduces the interval in which the maximum (or minimum) of an arbitrary nonlinear function must lie.
- A Fibonacci search, a prior specification of the resolution factor ($\in$) and the number of experiments to be performed (N) are required. In search methods to have a proper functioning, the successive experiments will gradually reduce the interval of uncertainty, i.e., the final interval of uncertainty will converge to zero as the number of functional evaluation increases to infinity provided that $\in$ is small.
- The necessary and sufficient conditions for a local optimum of the general Nonlinear programming problem, with both equality and inequality constraints is called Kuhn-Tucker conditions.
- In quadratic programming type of programming problem objective function is Nonlinear but constraints are linear.
- Karush–Kuhn–Tucker conditions are also known as KKT conditions or as Kuhn-Tucker conditions. As a mathematical approach to such problems, conditions that are necessary for finding solution to a NLP that is optimal, if some conditions of regularity are satisfied. This method is a generalization of Lagrange's multipliers applied to inequality constraints. These conditions have been named on its contributors W. Karush, H.W. Kuhn, and A.W. Tucker.
- Introducing artificial variables A_j ($j = 1, 2, ..., n$) in condition (i) we get,

$$c_j - \sum_{k=1}^{n} x_k d_{jk} - \sum_{i=1}^{m} \lambda_i a_{ij} + \mu_j + A_j = 0$$

- Beal's method uses results to solve based on calculus quadratic programming problem instead of Kuhn-Tucker conditions.
- Beale's method starts by making partitions of n variables in QP problem into basic and non-basic variables at every iteration of the solution process, and expressing basic variables and objective function in terms of non-basic variables.
- If at any iteration a free variable becomes a basic variable and is non-zero, then drop the new constraint that contains it. This should be done because it is a free variable and therefore, is neither chosen to leave the basis nor appears in the selection of leaving variable.
- The linear programming, no standard algorithm like the simplex method can be employed to solve the Nonlinear programming problems.

- Convex programming covers special cases of various types of NLPP, where the objective function $f(X)$ is a concave function and each of the constraints g^i(X) is a convex function. These assumptions ensure that a local maximum is also a global maximum.
- Separable programming is a special case of convex programming. One additional assumption made here is that all the $f(X)$ and g^i(X) are separable functions.
- A function is called separable, when it can be expressed as a sum of subfunctions where each subfunction is a function of one variable only. For example, if $f(X)$ is a separable function, it can be expressed as

$$f(X) = \sum_{i=1}^{n} f_i(x_i),$$

- All NLPP that do not satisfy the assumptions of convex programming, fall in this category of non-convex programming. There is no algorithm that can result into global optimal solution of such problems. Some algorithms attempt to find the local minima in case the problem does not deviate much from the assumptions of convex programming.

5.7 KEY TERMS

- **Fibonacci search:** Fibonacci search is a univariate search technique that can be used to find the maximum (or minimum) of an arbitrary unimodal, univariate objective function. This method is a sequential search technique that successfully reduces the interval in which the maximum (or minimum) of an arbitrary nonlinear function must lie.
- **Kuhn-Tucker conditions:** The necessary and sufficient conditions for a local optimum of the general Nonlinear programming problem, with both equality and inequality constraints is called Kuhn-Tucker conditions.
- **Quadratic programming:** Quadratic Programming (QP) is a Nonlinear Programming (NLP). It is an optimization problem of special type in which a quadratic function of many variables is optimized (maximized or minimized), subject to constraints of linear nature on these variables, using mathematical approach. In this type of programming problem objective function is Nonlinear but constraints are linear.
- **Beale's method:** Beale's method starts by making partitions of n variables in QP problem into basic and non-basic variables at every iteration of the solution process, and expressing basic variables and objective function in terms of non-basic variables.
- **Convex programming:** Convex programming covers special cases of various types of NLPP, where the objective function $f(X)$ is a concave function and each of the constraints g^i(X) is a convex function. These assumptions ensure that a local maximum is also a global maximum.

- **Separable programming:** Separable programming is a special case of convex programming. One additional assumption made here is that all the $f(X)$ and $g^i(X)$ are separable functions.

NOTES

5.8 SELF-ASSESSMENT QUESTIONS AND EXERCISES

Short-Answer Questions

1. What do you mean by the one and multivariable unconstrained optimization?
2. Write the short note on Kuhn-Tucker condition.
3. What is quadratic programming?
4. Mention the Beale's method.
5. What is the separable programming?
6. Differentiate between convex programming and non-convex programming.

Long-Answer Questions

2. Describe the Kuhn-Tucker condition for constrained optimization.
3. Explain the quadratic programming by giving an example.
4. Analyse the Karush-Kuhn-Tucker conditions. Give appropriate examples.
5. What is a Beale's method? Explain why it is used.
6. Discuss the types of Nonlinear programming problems.

5.9 FURTHER READING

Kalavathy, S. 2002. *Operations Research*. New Delhi: Vikas Publishing House Pvt. Ltd.

Gupta, P.K. and D.S. Hira. 2002. *Introduction to Operations Research*. New Delhi: S. Chand And Company Limited.

Taha, H.A. 2006. *Operations Research: An Introduction*, 8th Edition. New Delhi: Prentice-Hall of India.

Jensen, Paul A. and Jonathan F. Bard. 2003. *Operations Research Models and Methods*. New York: John Wiley & Sons.

Arumugam, R. S. 2006. *Operations Research*. Palayamkottai (Tamil Nadu): New Gamma Publications.

Sundharesan, V., K. S. Ganapathy and K. Ganesan. 2017. *Resource Management Techniques (Operations Research)*. Chennai: A. R. Publications.

Gillett, Billy E. 2007. *Introduction to Operations Research*. New Delhi: Tata McGraw-Hill.

NOTES

Ackoff, R. L. and M. W. Sasieni. 1968. *Fundamentals of Operations Research*. New York: John Wiley & Sons Inc.

Kothari, C. R. 1992. *An Introduction to Operational Research*. New Delhi: Vikas Publishing House Pvt. Ltd.

Sharma, J. K. 2001. *Operations Research: Theory and Applications*. New Delhi: Macmillan India Ltd.

Swaroop, Kanti, P.K. Gupta and Man Mohan. 2001. *Operations Research*. New Delhi: Sultan Chand & Sons.