Module- I

Overview of Operations Research

Meaning

Operations research is a scientific approach to problem solving for executive decision making which requires the formulation of mathematical, economic and statistical models for decision and control problems to deal with situations arising out of risk and uncertainty. In fact, decision and control problem in any organization are more often related to certain daily operations such as inventory control, production scheduling, manpower planning and distribution, and maintenance.

Definition

1. According to Operations Research Society of America (ORSA), it is a tool which is concerned with the design and operation of the man-machine system scientifically, usually under conditions requiring the optimum allocation of limited resources.
2. As per the operations Research Society of Great Britain, operations research is the application of the scientific methods to complex problems arising in the direction and management of large systems of men, machines, materials and money in industry, business and government.

Concept

The origin and development of operations research can be studied under the following classification:

1. Pre-World War II developments
2. Developments during World War II
3. Post-World War II developments
4. Computer era
5. Inclusion of uncertainly models.

Per-World War II developments

Many of the techniques of today's operations research have been actually developed and used even before the term 'operations research' was coined. Some of the techniques are: inventory control, queueing theory, and statistical quality controls.

In 1915, Ford Harris developed a simple EOQ (economic order quantity) model to optimize the total cost of inventory system, which was eventually analyzed in 1934 by R.H Wilson. Around the same time (1916), A.K. Erlang, a Danish telephone engineer, was responsible for many of the early theoretical developments in the area of queueing theory.

In the early 1900s, routine quality checks conducted by inspectors were not found to be satisfactory for some companies. The problem was analyzed in the inspection engineering department of Western Electric's Bell Laboratory by Shewhart who ultimately designed control charts in 1924. These are called as the first Shewhart control charts. During the period 1925-26, the Western Electric Company defined various terminologies associated with acceptance sampling of quality control that was used as a tool for controlling attributes of raw materials/ components/ finished products. The terminologies include consumer's risk producer's risk, probability of acceptance, operating characteristics (OS) curve, lot tolerance percent defective (LTPD), double sampling plan, type I error, type II error and so on. In 1925, Dodge introduced the basic concept of sampling inspection. Ten years later, Pearson developed the British Standard Institution Number 600, entitled ' Application of statistical method to international standardization and quality control'. In 1939, H. Roming presented his work on variable sampling plan in his Ph.D dissertation.

Developments during World War II

During the World War II, the effective utilization of scarce resources was the top-most concern of the military in Britain. So, in Britain scientists from different fields were jointly directed to do research on military operations for improving its effectiveness with the limited resources. Later on, this scientific and interdisciplinary approach became an important problems-solving aspect of operations research methodologies.

Post –World War II developments

After the World II, the industries in America and Britain concentrated in applying the operations research methodologies to industrial problems for maximizing the profitability with limited resources.

In 1947, Dantzig, developed **Simplex Method** to solve liner programming problem. Thereafter the operations Research Society of America, and the Institute of Management Science were founded in 1952 and 1953, respectively.

Computer era

Many of the operations research techniques involve complex computations and hence they take longer time for providing solutions to real life problems. The developments of high spend digital computers made it possible to successfully apply some of the operations research techniques to large size problems. The developments of recent interactive computers make the job of solving large even more simple because of human intervention towards sensitivity analysis.

Inclusion of uncertainty models

The use of probability theory and statistics to tackle undeterministic situation made the operations research techniques more realistic.

MODELS IN OR

Diverse items such as a map, a multiple activity chart, an autobiography, PERT network, break-even equation, balance sheet, etc. are all models.

CLASSIFICATION SCHEMES OF MODELS

The various schemes by which models can be classified are

1. By degree of abstraction
2. By function
3. By structure
4. By nature of the environment
5. By the extent of generality
6. By the time horizon

TYPES OF MATHEMATICAL MODELS

Many OR models have been developed and applied to problems in business and industry.

Some of these models are:

1. Mathematical techniques
2. Statistical techniques
3. Inventory models
4. Allocation models
5. Sequencing models
6. Project scheduling by PERT and CPM
7. Routing models
8. Competitive models
9. Queening models
10. Simulation techniques
11. Decision theory
12. Replacement models
13. Reliability theory
14. Markov analysis
15. Advanced OR models
16. Combined methods

1. **Mathematical Techniques**
   Mathematical techniques most commonly employed are: differential equations, linear difference equation, integral equations, operator theory, vector and matrix theory.

2. **Statistical Techniques**
   Some of the most commonly applied techniques come from probability theory and statistics. Since in the world we live, the course of future events cannot be predicated with absolute certainty, probabilities are associated with these events to analyse the uncertainties and supply data with reasonable accuracy for decision-making.
3. **Inventory Models**
   Inventory models deal with idle resources such as men, machines, money and materials.

These models are concerned with two decisions:

(i.) How much to order (produce or purchase) to replenish the inventory of an item and
(ii.) When to order so as to minimize the total cost.

4. **Allocation Models**
   Allocation models are used to solve problems in which (a) there a number of jobs to be performed and there are alternative ways of doing them and (b) resources of facilities are limited.
   In such situations, the objective is to allot the resources to the jobs in such a way as to optimize the overall effectiveness (i.e. minimize the total cost or maximize the total profit). This is called mathematical programming. When the constraints are expressed as linear equalities/ inequalities, this is called linear programming.
5. **Sequencing Models**
   These are applicable in situations in which the effectiveness measure (time, cost or distance) is a function of the order or sequence of performing a series of jobs (takes). The selection of the appropriate order in which waiting customers may be served is called sequencing, in these problems, generally there are n jobs to be performed, where each job requires processing on some or all of m different machines.
6. **Project Scheduling by PERT and CPM**
   In a large and complex project involving a n umber of interrelated activities, requiring a number of men, machines and materials, it is not possible for the management to make and execute and optimum scheduled just by intuition. Managements are thus, always on the lookout for some methods and techniques which may help in planning, scheduling and controlling the project. PERT and CPM are two of many project management techniques used for these purposes.
7. **Routing Models**
   Routing problems in networks are the problems which are related to sequencing. There are two important routing problems:
   (a) The travelling salesman problem
   (b) The minimal path problem

In the travelling salesman problem, these are a number of cities a salesman has to visit. The distance (or time or cost) between every pair or cities is known. The salesman is to start from his home city, visit each city only once and return to his home city, and the problem is to find shortest route in distance (or time or cost).

8. **Competitive Models**
   These models are used when two or more individuals or organizations with conflicting objectives try to make decisions. In such situations a decision made by one decision- maker affects the decision made one or more of the remaining decision- maker. Competitive models are applicable to wide enemies planning war tactics, firms struggling to maintain their market shares, situations of collective bargaining etc.
9. **Queuing Models**
   Queuing models involve the arrival of units to be serviced at one or more service facilities. These units (or customers) may be trucks arriving a loading station, customers entering a department store, patients arriving a doctor's clinic, ships arriving a port, letters arriving a port, letters arriving a typist's desk, etc. the arriving units may from one line and get serviced, as in a doctor's clinic. This will occur when the system has a single service channel.
10. **Simulation Techniques**
   Sometimes it may be very risk, cumbersome or time consuming to conduct real study or experiment to know more about a situation or problem. Also sometimes due to a large number experiment to know more about a situation or problem. Also sometimes due to a large number of variables or large number of interrelationship between variables, and the complexity of relationship, it may not be possible to develop an analytical mode representing the situation. Even if a model is constructed, it may not be possible to solve it. Simulation may be helpful in such situation.
11. **Decision Theory**
   Decision making is an everyday process; it is a major part of a manager's job. As wrong decisions can be disastrous for the companies, the importance of right decisions cannot be overemphasized. Decisions may be classified into two categories: tactical decisions and strategic decision. Tactical decision is those that effect the organization in the short run.
12. **Replacement Models**
   The replacement theory deals with industrial, military and other equipment that gets worn with usage and time and thereby functions with decreasing efficiency. For example, a machine requires higher operating cost, a transport vehicle requires more and more maintenance cost, a railway timetable becomes more and more out of data with time. The ever increasing repair and maintenance cost necessitates the replacement.
13. **Reliability Theory**

Reliability is the capability of an equipment to work without any break down or failure for specified period of time under given environmental conditions. The equipment may be simple a device like a switch, a fan, an electric heater or large and complex one such as a computer, a radar or an aero plane. Reliability of the complex equipment depends upon the reliability of its components. It is essential that equipment should well perform the function for which it is designed.

14. Markov Analysis

Markov analysis for decision-making is used in situations where various states are defined. The probability of going from one state to another is known and depends on the present state but independent of how that state was reached. Morkov analysis is used to determine the long-run probability of being in a particular state (Steady State probability). This steady state probability is used in decision-making.

Dynamic Programming

Introduction

Dynamic programming is a special kind of optimization technique which subdivides an original problem into as many number of sub problems as the variables, solves each sub problems individually and then obtains the solution of the original problem by integrating the solutions of the sub problems.

Application of Dynamic Programming

The dynamic programming can be applied to many real-life situations. A sample list of applications of the dynamic programming is given below.

1. Capital budgeting problem
2. Reliability improvement problem
3. State-coach problem (shortest- path problem)
4. Caro loading problem
5. Minimizing total tardiness in single machine scheduling problem
6. Optimal subdividing problem
7. Linear programming problem

They are discussed in the following sections:

Capital Budgeting Problem

A capital budgeting problem is a problem in which a given amount of capital is allocated to set of plant by selecting the most promising alternative for each selected plants such that the total revenue of the organization is maximized.

Example: An organization is planning to diversify its business with a maximum outlay of Rs. 5 cores. It has identified three different locations to install plants. The organization can invest in one or more of these plants subjects to the availability of the fund. The different possible alternative and their investment (in cores of rupees) and present worth of returns during the useful life (in cores of rupees) of each of these plants are summarized in Table. The first row of Table has zero cost and zero return for all the plants. Hence, it is known as do-nothing alternative. Find the optimal allocation of the capital to different plants which will maximize the corresponding sum of the present worth of returns.

Table

Alternative	Plant-1 Cost	Plant-1 Return	Plant -2 Cost	Plant -2 Return	Plant -2 Cost	Plant -2 Return
1	0	0	0	0	0	0
2	1	15	2	14	1	3
3	2	18	3	18	2	7
4	4	28	4	21	-	-

Solution: Maximum capital amount, c = Rs. 5 cores. Each plant is treated as a stage. So, the number of states is equal to 3. The plants 1, 2 and 3 are defined as stage 1, stage 2 and stage 3, respectively. So the forward recursive function is used for the problem.

State 1. The recursive function for a given combination of the state variable, x_1 and alternative, m_1 in the first stage is presented below. The corresponding returns are summarized in Table -2. For each value of the state variable, the best return and the corresponding alternative are presented in the last two columns, respectively.

$$f_1(x_1) = R\ (m_1)$$

Table- 2 Calculations for State 1 (Plant 1)

State Variable	Alternative m_1 1		2		3		4			
	C	R	C	R	C	R	C	R		
x_1	0	0	1	15	2	18	4	28	$f_1^*(x_1)$	m_1^*
0	0		-		-		-		0	1
1	0		15		-		-		15	2
2	0		15		18		-		18	3
3	0		15		18		28		18	3
4	0		15		18		28		28	4
5	0		15		18		28		28	4

C= cost, R= return

Stage 2. The recursive function $f_2(x_2)$ for a given combination of the state variable, x_2 and alternative m_2, in the second stage is given by

$$\boldsymbol{f_2(x_2) = R\ (m_2) = f_1[x_2\text{-}C(m_2)]}$$

The corresponding returns are summarized in Table 3. For each value of the state variable, the best return and the corresponding alternative(s) are presented in the last two columns, respectively.

Table 3. Calculations for Stage 2 (Plant 2)

	Alternative m_2					
State Variable	1	2	3	4		
	C R	C R	C R	C R		
x_2	0 0	2 14	3 18	4 21	$f_2(x_2)^*$	m_2^*
0	0	-	-	-	0	1
1	0+15=15	-	-	-	15	1
2	0+18=18	14+0= 14	-	-	18	1
3	0+18=18	14+15= 29	18+0= 18	-	29	2
4	0+28=28	14+18= 32	18+15= 33	21+0= 21	33	3
5	0+28=28	14+18= 32	18+18= 36	21+15= 36	36	3 and 4

Stage 3. The recursive function *f*3(x3) for different combinations of the state variable, x3 and alternative, m3 in the third stage is:

$$f_3(x_3) = R\,(m_3) + f_2\,[x_3\text{-}C(m_3)]$$

the corresponding returns are summarized in Table 4. For each value of the state variable, the best return and the corresponding alternative (s) are presented in the last two columns, respectively

Table- 4 Calculations for Stage 3 (Plant 3)

	Alternative m_3							
State Variable	1		2		3			
	C	R	C	R	C	R		
x_3	0	0	1	3	2	7	$f_3(x_3)^*$	m_3^*
5	0+36=36		3+33=36		7+29=33			

The final results of the original problem is traced as in Table 5. From this table, one can visualize the fact that the original problem has four alternate optimal solutions.

Table 5 Final Result

Stage-3		Stage 2		Stage 1		Optimal alternatives stage		
C*	m_3*	C*	m_2*	C*	m_1*	1	2	3
5	1	5-0= 5	3 4	5-3=2 5-4=1	3 2	3 - 3 - 1		
5	2	5-1= 4	3	4-3= 1	2	2 - 4 - 1		
5	3	5-2= 3	2	3-2= 1	2	2 - 2 - 3		

Integer Programming

Introduction

In linear programming problems, decision variables are non-negative values which are restricted to be zero or more than zero. This demonstrates one of the basic properties of linear programming, namely, continuity, which means that fractional values of the decision variables are possible in the solution of a linear programming model. For problems like, product mix problem, nutrition problems, the assumption of continuity may be valid for some applications.

Some items cannot be produced in fractions, like 101.5 cranes, 3.2 bucket wheel excavators, etc. if e round off the production volume of any one of such products of the nearest integer value, the corresponding solution would be different from the optimal solution based on the assumption of continuity. The difference would be significant if the profit per unit of each of the products in the product mix problem is very high. Hence, there is a need for integer programming methods to overcome this difficulty.

Example: consider the following production planning situation of a company manufacturing mixes. The company plans to manufacture two types of mixies. The selling prices for these mixies are: model A costs Rs. 1,750 per unit and Model B costs Rs. 2,000 per unit. Daily production volume of each type of these mixies is constrained by available man hours and available machine hours. the production specifications for the given problem situation are presented in Table- 1

Table

Resource requirement/ unit			
Resource	Model A	Model B	Availability (in hours)
Man hours	4	8	32
Machine hours	6	4	36

Fine the optimal production plan for the above problem. (Note: The values of availabilities of the resources are assumed as small numbers just to demonstrate the problem on a graph conveniently. In reality, these values will be high.)

Solutions: Let x_1 and x_2 be the respective number of Model A and Model B to be manufactured. AB integer linear programming model for the above problem is represented as given below:

$$\text{Maximize } Z = 175x_1 + 2000x_2$$

Subject to

$$4X_1 + 8X_2 \leq 32$$

$$6X_1 + 4X_2 \leq 36$$

$$X_1 \text{ and } X_2 \leq 0 \text{ and integers}$$

The optimal linear programming solution of the above problem is given below

$$X_1 = 5.0,\ X_2 = 1.5,\ Z \text{ (optimum)} = \text{Rs. } 11{,}750$$

In this solution, the value of X_1 is an integer and that of X_2 is a non-integer. But the solution of this problem will be meaningful only when the values of all decision variables are integer. A simple approach may be to round off the value of X_2 to the previous integer value of 1 to maintain feasibility. After rounding off the value of X_2, the values of the decision variable, X_1 and X_2 become 5 and 1, respectively. The corresponding total profit is Rs. 10,750 which is less than that of the optimum value of the linear programming solution.

The optimum integer solution for the given problem is as follows which is better than the rounded off solution of the linear programming problem.

$$X_1 = 4,\ X_2 = 2 \text{ and } Z \text{ (optimum)} = \text{Rs. } 11{,}000$$

THE CUTTING-PLANE ALGORITHM

An algorithm for solving fractional (pure integer) and mixed integer programming problems has been developed by Ralph E. Gomory.

Fractional (pure integer) algorithm

Step 1: First, relax the integer requirements.

Step 2: Solve the resulting LP Problem using simplex method.

Step 3: If all the basic variables (or the required variables) have integer values, optimality of The integer programming problem is reached. So, go to step7; otherwise go to step 4.

Step 4: Examine the constraints corresponding to the current optimal solution. Also, let m be the number of constraints, n be the number of variables (including slack, surplus and artificial variables), b_i be the right-hand side value of the *i*th constraint, and a_{ij} be the technological coefficients (matrix of left-hand side constants of the constraints). Then, the constraint equations are summarized as follows.

For each basic variable with non integer solution $\sum_{j=1}^{n} a_{ij}X_j = b_i, \quad i = 1, 2, 3, \ldots, m$ in the current optimal table, find the fractional part, f_i. Therefore, $b_i = [b_i] + f_i$ where $[b_i]$ is the integer part of bi and f_i is the frictional part of bi.

Step 5: choose the largest fraction among various fi' s; i.e. Max (fi). Threat the constraint Corresponding to the maximum fraction as the sources row. Let the corresponding sources row be as follows.

$$b_i = X_i + \sum_{i=1} a_{ij}w_j \quad \text{or} \quad X_i = b_i - \sum_{i=1} a_{ij}w_j$$

Where variables X_i (i=1,2,3,……,m) represent basic variables and variables w_j (j= 1,2,3,…..,n) are the non-basic variables. This kind of assumption is for convenience only.

Some example of diving b_i and a_{ij} into integer and fractional parts are shown as in table 2

Table Examples of Integer and Fractional Parts

b_i or a_{ij}	$[b_i]$ or $[a_{ij}]$ (Truncated integer)	$f_i = b_i - [b_i]$ $f_{ij} = a_{ij} - [a_{ij}]$
$2\frac{2}{3}$	2	$\frac{2}{3}$
$-3\frac{1}{4}$	−4	$\frac{3}{4}$
−5	−5	0
$-\frac{3}{7}$	−1	$\frac{4}{7}$

Based on the sources equation, develop an additional constraint (Gomory's constraint or fractional cut) as shown below:

$$S_i = \sum_{j=1}^{n} f_{ij}w_j - f_i \quad \text{or} \quad -f_i = S_i - \sum_{j=1}^{n} f_{ij}w_j$$

Where S_i is non-negative slack variable and also an integer.

Appendix 6: Append the fractional cut as the last row in the latest optimal table and proceed further using dual simplex method, and find the new optimum solution. If this new optimum solution is integer then go to step 7; otherwise go to step 4.

Step 7: Print the integer solution [X_i s and Z values]

Example: Find the optimum integer solution to the following linear programming problem.

Maximum $Z = 5X_1 + 8X_2$

Subject to $X_1 + 2X_2 \leq 8$

$4X_1 + X_2 \leq 10$

$X_1, X_2 \geq 0$ and integers

Solution: The canonical from of the above problem is as follows:

Maximum $Z = 5X_1 + 8X_2$

Subject to $X_1 + 2X_2 + S_1 = 8$

$4X_1 + X_2 + S_2 = 10$

X_1, X_2, S_1 and $S_2 \geq 0$ and integers

The initial table is shown in Table

Table-1 Initial Table

	C_j	5	8	0	0		
CB_i	Basic variable	X_1	X_2	S_1	S_2	Solution	Ratio
0	S_1	1	$\boxed{2}$	1	0	8	8/2 = 4*
0	S_2	4	1	0	1	10	10/1 = 10
	Z_j	0	0	0	0	0	
	$C_j - Z_j$	5	8*	0	0		

From the above table, the entering variable is X_2. Since its C_j- Z_j value is the maximum positive value. The minimum ratio is 4 and the corresponding variable is S_1. Therefore, S_1 leaves the basis. The resulting table is shown as in below Table.

Table-2 Iteration 1

CB_i	C_j / Basic variable	5 X_1	8 X_2	0 S_1	0 S_2	Solution	Ratio
8	X_2	1/2	1	1/2	0	4	8
0	S_2	$\boxed{7/2}$	0	−1/2	1	6	12/7*
	Z_j	4	8	4	0	32	
	$C_j - Z_j$	1*	0	−4	0		

In table 2, the maximum positive value for $C_j - z_j$ is 1. The corresponding variable is X_1. Therefore X_1 enters the basis. The minimum ratio is for the S_2 row. Therefore, S_2 leaves the basis. The resulting table is shown as in Table 3.

Table-3 Iteration-2 (Optimal Table)

CB_i	C_j / Basic variable	5 X_1	8 X_2	0 S_1	0 S_2	Solution
8	X_2	0	1	4/7	−1/7	22/7
5	X_1	1	0	−1/7	2/7	12/7
	Z_j	5	8	27/7	2/7	236/7
	$C_j - Z_j$	0	0	−27/7	−2/7	

In Table 3, all the values in the criterion row (Cj – Zj row) are) or negative. Hence, optimality for linear programming is reached. The results are as follows:

$$X_1 = \frac{12}{7}, \qquad X_2 = \frac{22}{7}, \qquad Z(\text{optimum}) = \frac{236}{7}$$

Since the values of the decision variables X1 and X2 are not integers, the solution is not optimal. So, to obtain integer solution for the given problem further steps are carried out.

The integer and fractional parts of the basic variables are summarized in Table 4

Table-4 Summary of Integer and Fractional parts

Basic variable in the optimal table	b_i	$[b_i] + f_i$
X_1	12/7	1 + (5/7)
X_2	22/7	3 + (1/7)

The fractional part, F1 is the maximum. So, select the row X1 as the source row for developing the first cut.

$$\frac{12}{7} = X_1 - \frac{1}{7}S_1 + \frac{2}{7}S_2$$

or

$$1 + \frac{5}{7} = X_1 + \left(-1 + \frac{6}{7}\right)S_1 + \left(0 + \frac{2}{7}\right)S_2$$

The corresponding fractional cut is

$$-\frac{5}{7} = S_3 - \frac{6}{7}S_1 - \frac{2}{7}S_2$$

This cut is appended to Table 3 as reproduced in Table 5 and further solved using dual simplex method.

Table- 5 Table after Appending Cut 1

	C_j	5	8	0	0	0	
CB_i	Basic variable	X_1	X_2	S_1	S_2	S_3	Solution
8	X_2	0	1	4/7	–1/7	0	22/7
5	X_1	1	0	–1/7	2/7	0	12/7
0	S_3	0	0	–6/7	$\boxed{-2/7}$	1	–5/7*
	Z_j	5	8	27/7	2/7	0	236/7
	$C_j - Z_j$	0	0	–27/7	–2/7*	0	

Only the third row (containing S3) has a negative solution value. Therefore, S3 leaves the basis. The entering variable is determined based on Table 5.

Table -6 Determination of Entering Variable

Variable	X_1	X_2	S_1	S_2	S_3
$-(C_j - Z_j)$	0	0	27/7	2/7	0
Row S_3	0	0	–6/7	–2/7	1
Ratio (absolute value)	–	–	9/2	1	–

The smallest absolute ratio is 1 and the corresponding variable is S2. So, the variable S2 enters the basis. The resultant values are shown in Table -7

Table-7 Table after Pivot Operation

CB_i	C_j / Basic variable	5 X_1	8 X_2	0 S_1	0 S_2	0 S_3	Solution
8	X_2	0	1	1	0	–1/2	7/2
5	X_1	1	0	–1	0	1	1
0	S_2	0	0	3	1	–7/2	5/2
	Z_j	5	8	3	0	1	33
	$C_j - Z_j$	0	0	–3	0	–1	

The solution is still not integer. So, develop a fractional cut. The basic variables, X2 and S2 are no integers. The fractional parts of both of them are ½. The constraint X2 is selected randomly as source row for developing the nest cut.

$$\frac{7}{2} = X_2 + S_1 - \frac{1}{2}S_3$$

Therefore,

$$3 + \frac{1}{2} = (1 + 0)\ X_2 + (1 + 0)\ S_1 + \left(-1 + \frac{1}{2}\right)S_3$$

or

Therefore, the corresponding fractional cut is

$$-\frac{1}{2} = S_4 - \frac{1}{2}S_3$$

Append this constraint at the end of Table 7 as shown in Table 8.

CB_i	C_j / Basic variable	5 X_1	8 X_2	0 S_1	0 S_2	0 S_3	0 S_4	Solution
8	X_2	0	1	1	0	–1/2	0	7/2
5	X_1	1	0	–1	0	1	0	1
0	S_2	0	0	3	1	–7/2	0	5/2
0	S_4	0	0	0	0	$\boxed{-1/2}$	1	–1/2*
	Z_j	5	8	3	0	1	0	33
	$C_j - Z_j$	0	0	–3	0	–1*	0	

Only row S4 has a negative value the solution column. Therefore, S4 leaves the basis. The entering variable is determined based on Table 9.

Table-9 Determination of Entering Variable

Variable	X_1	X_2	S_1	S_2	S_3	S_4
$-(C_j - Z_j)$	0	0	3	0	1	0
Row S_4	0	0	0	0	−1/2	1
Ratio (absolute value)	–	–	–	–	2	–

The smallest absolute ration (only ratio) is 2 and the corresponding variable is S3. So, the variable S3 enters the basis. The resultant values are shown as in Table 10

Table 10 Table after Pivot Operation

	C_j	5	8	0	0	0	0	
CB_i	Basic variable	X_1	X_2	S_1	S_2	S_3	S_4	Solution
8	X_2	0	1	1	0	0	−1	4
5	X_1	1	0	−1	0	0	2	0
0	S_2	0	0	3	1	0	−7	6
0	S_3	0	0	0	0	1	−2	1
	Z_j	5	8	3	0	0	2	32
	$C_j - Z_j$	0	0	−3	0	0	−2	

In Table 10, the values of all the basic variables are integers. So, the optimality is reached and the corresponding result are summarized as follows: X_1= 0, X_2= 4 Z (Optimum) = 32

BRANCH AND BOUND TECHNIQUE

If the member of decision variables in a n integer programming problem is only two, a branch-and –bound technique can be used to find its solution graphically. Various terminologies of branch-and-bound technique are explained as under:

Branching. If the solution to the linear programming problem contains non-integer values for some or all decision variable, then the solution space is reduced by introducing constraints with respect or any one of these decision variables. If the value of a decision variable X_1 is 2.5, then two more problems will be created by suing each of the following constraints.

$$X_1 \leq 2 \text{ and } X_1 \geq 3$$

Lower bound. This is a limit to define a lower value for the objective function at each and every node. The lower bound at a node is the value of the objective function corresponding to the truncated values (integer parts) of the decision variables of the problem in that node.

Upper bound. This is a limit to define an upper value for the objective function at each and every node. The upper bound at a node is the value of the objective function corresponding to the linear programming solution in that node.

Example. Solve the following integer programming problem using branch-and bound technique.

$$\text{Maximize } Z = 10X_1 + 20X_2$$

Subject to $6X_1 + 8X_2 \leq 48$

$X_1 + 3X_2 \leq 12$

$X_1, X_2 \geq 0$ and integers

Solution The introduction of the non-negative constraint $X_1 \geq 0$ and $X_2 \geq 0$ will eliminate the second, third and fourth quadrants of the X1X2 plane.

Now, from the first constraint in equation from

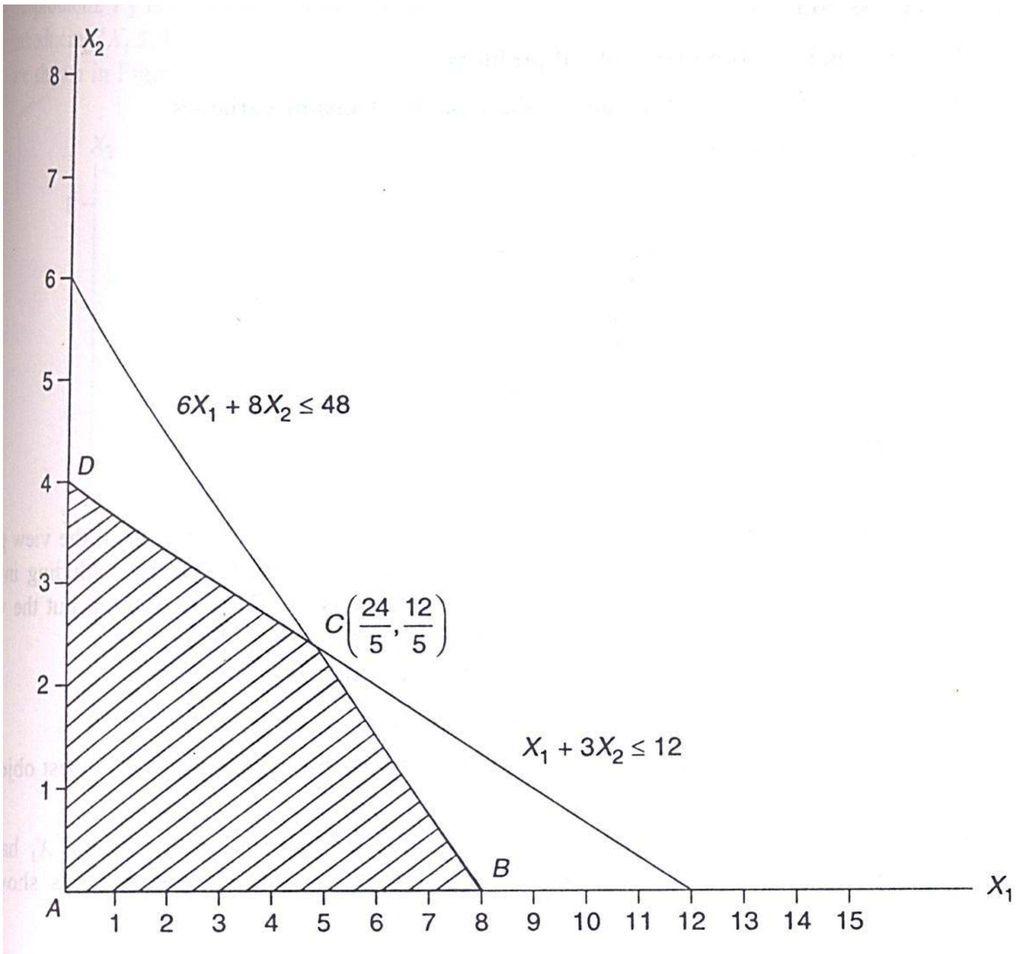
$$6X_1 + 8X_2 = 48$$

We get X2= 6, when X1= 0; and X1 = 8, when X2 = 0. Similarly from the second constraint in equation from

$$X_1 + 3X_2 = 12$$

We have X_2 =4 , when X_1 = 0; and X_1= 12, when X_2 = 0.

Now, plot the constraints 1 and 2 as shown in Figure



The closed polygon ABCD is the feasible region. The objective function value at each of the corner points of the closed polygon is computed as follows by substituting is coordinates in the objective function:

Z (A) = 10 x 0 + 20 x 0 = 0

Z (B) = 10 x 8+20 x 0= 80

Z (C) = 10 X $\frac{24}{5}$ + 20 x $\frac{12}{5}$ = 96

Z (D) = 10 x 0 + 20 x 4 + 80

Since, the type of the objective function is maximization; the solution corresponding to the maximum Z values is to be selected as the optimum solution. The Z values is maximum for the corner point C. hence, the corresponding solution of the continues linear programming problem is presented below.

$$X_1 = \frac{24}{5},\ X_2 = \frac{12}{5},\ Z\text{ (optimum)} = 96$$

These are jointly shown as problem P1 in figure 2. The notations for different types of lower bound are defined as follows:

Z_U = Upper bound = Z(Optimum) of LP problem

Z_L = Lower bound w.r.t. the truncated values of the decision variables

Z_B = Current best lower bound

Maximize $Z = 10X_1 = 20X_2$
Subject to
$6X_1 + 8X_2 \leq 48$
$X_1 + 3X_2 \leq 12$
X_1 and $X_2 \geq 0$ and integers

$X_1 = 24/5$
$X_2 = 12/5$
$Z_U = 96$
$Z_L = 80$
$Z_B = \infty$

Since both the values of X1 and X2 are not integers, the solution in not optimum from the view point of the given problem. So, the problem is to be modified into two problems by including integer constraints one by one. The lower bound of the solution of P1 is 80. This is nothing but the value of the objective function for the truncated values of the decision variables.

The rule for selecting of the variable for branching is explained as follows:

1. Selected the variable which has the highest fractional part.
2. If there is a tie, then break the tie by choosing the variable which has the highest objective function coefficient.

In the continuous solution of the given linear programming problem P1, the variable X1 has the highest fractional part (4/5). Hence, this variable is selected for further branching as shown in below.

Maximize $Z = 10X_1 = 20X_2$
Subject to
$6X_1 + 8X_2 \leq 48$

$X_1 = 24/5$
$X_2 = 12/5$

$X_1 = 5$
$X_2 = 9/4$
$Z_U = 95$

Maximize $Z = 10X_1 = 20X_2$
Subject to
$6X_1 + 8X_2 \leq 48$
$X_1 + 3X_2 \leq 12$

Maximize $Z = 10X_1 = 20X_2$
Subject to
$6X_1 + 8X_2 \leq 48$
$X_1 + 3X_2 \leq 12$

$X_1 = 4$
$X_2 = 8/3$
$Z_U =$

Branching From P_1

The problem P_2 and P_3 are generated by adding an additional constraint. The sub problem, P_2 is created by introducing '$X_1 \geq 5$' in problem. P_1 and the problem, P_3 is created by introducing '$X_1 \leq 4$' in problem P_1. The corresponding effects in slicing the non-integer feasible region are shown in Figures 1 and 2, respectively. The solution for each of the sub problems, P_2 and P_3

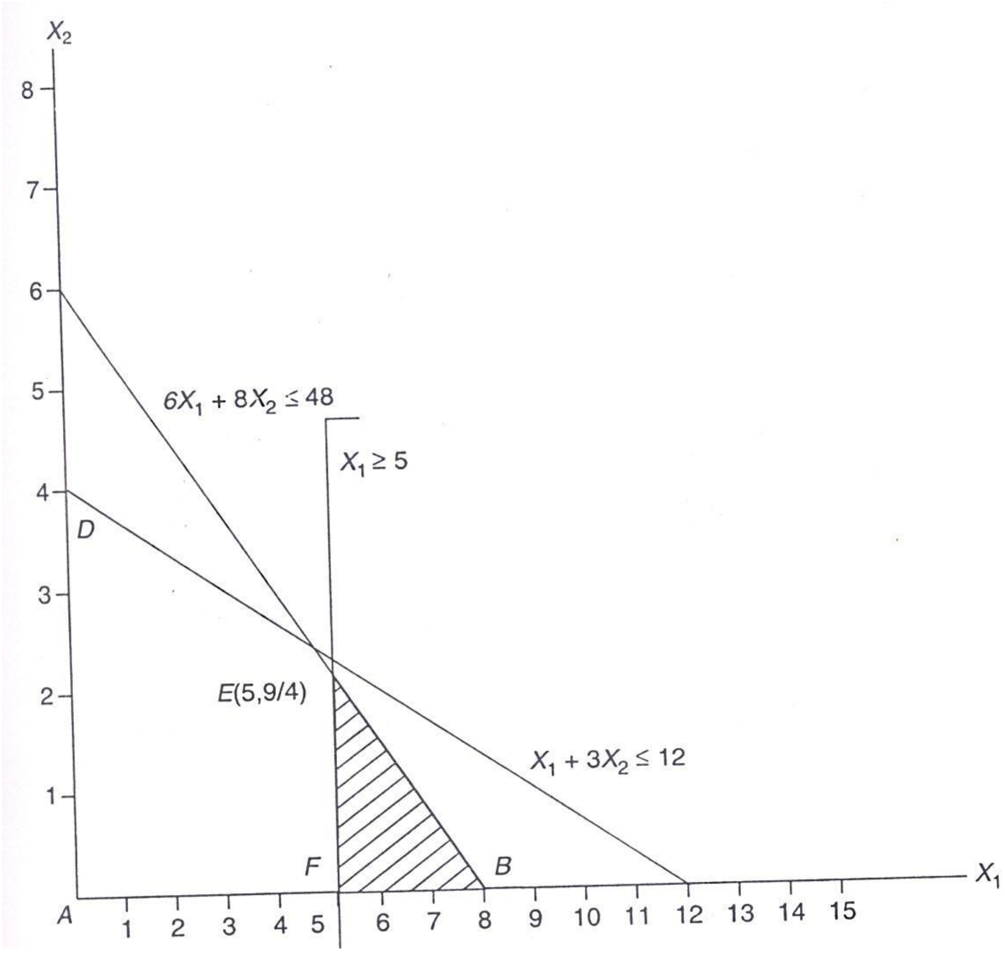


Figure 1 Feasible region of P_2 after introducing $X_1 \geq 5$ to P_1

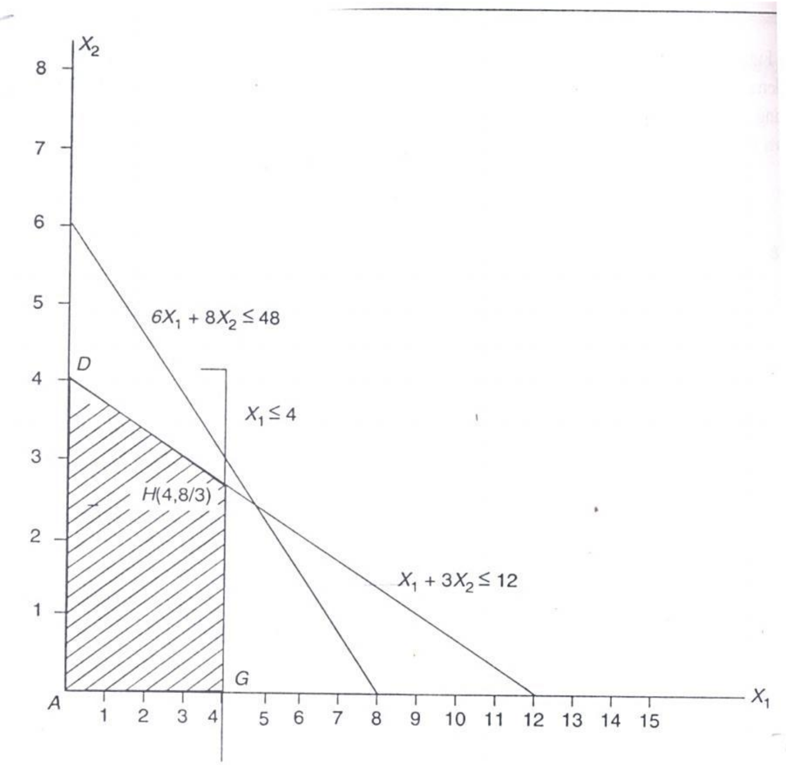


Figure- 2 Feasible region of P3 after introducing X1 $\leq$ 4 to P1.

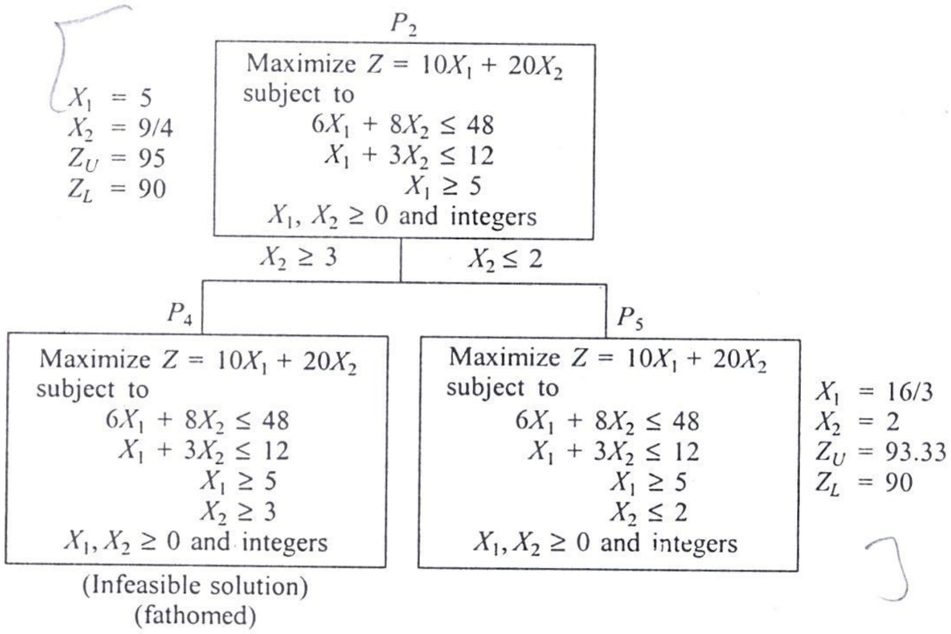


Figure 3 Branching From P_2

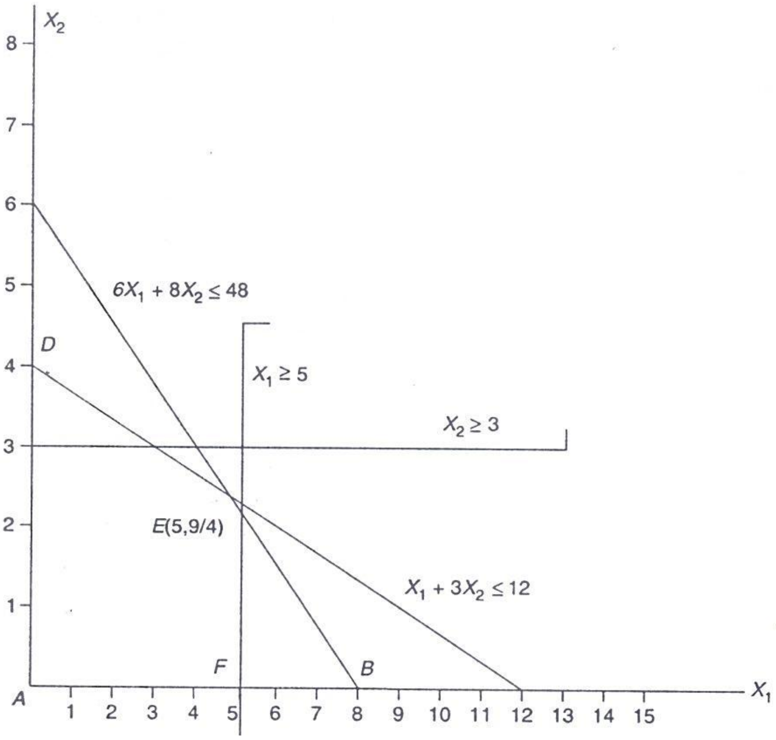


Figure 4 Infeasible region of P_4 after introducing $X_2 \geq 3$ to P_2.

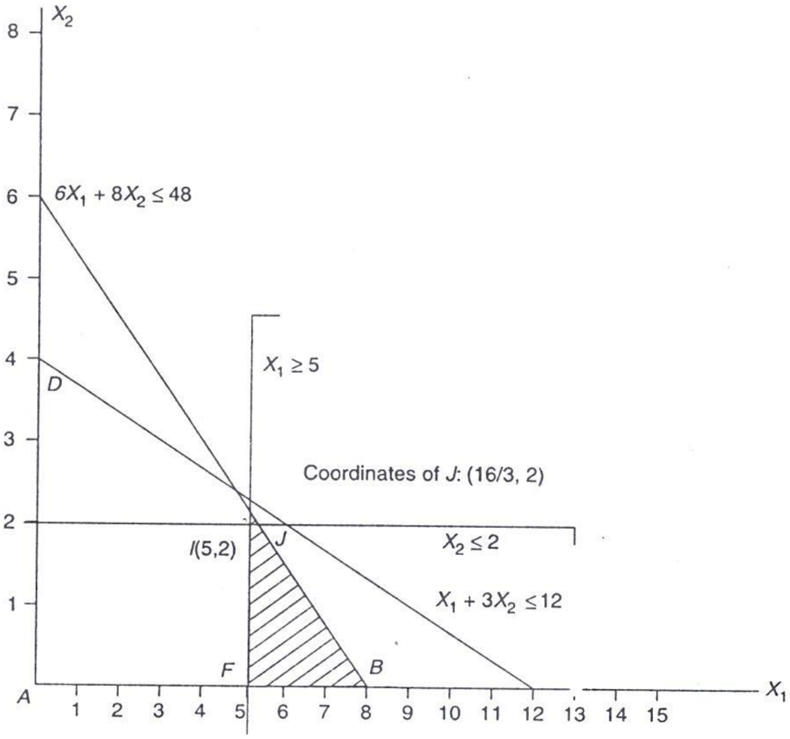


Figure 5 Infeasible region of P_4 after introducing $X_2 \geq 3$ to P_2

The further branching should be done from the node, P5 as shown Figure 6

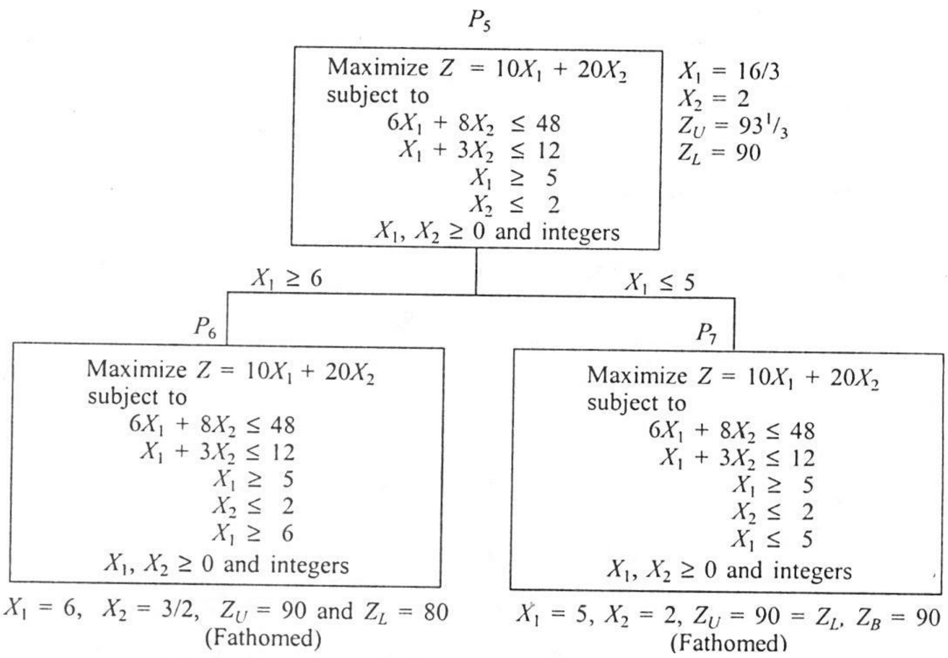


Figure 6 Branching from P_5

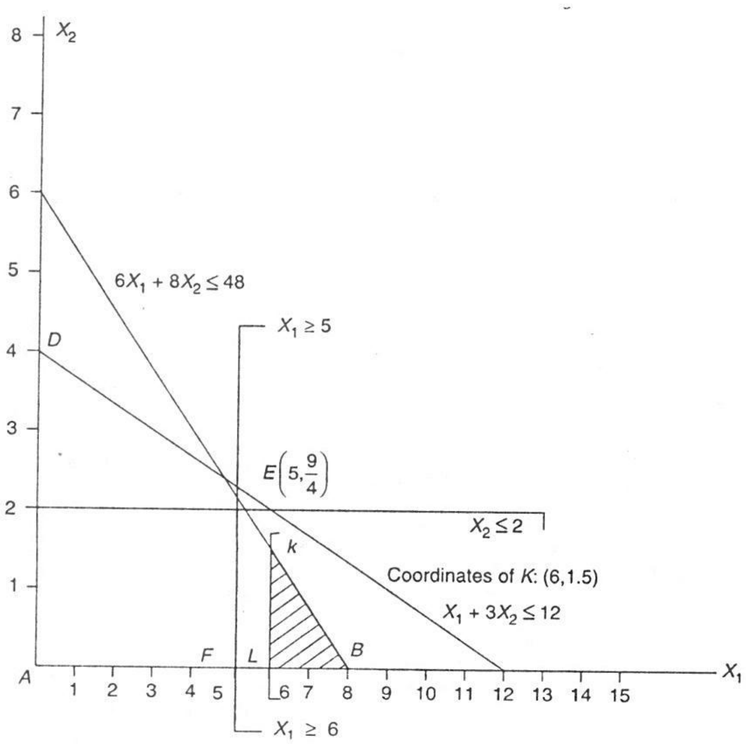


Figure 7 feasible region of P_6 after introducing $X1 \geq 6$ to P_5

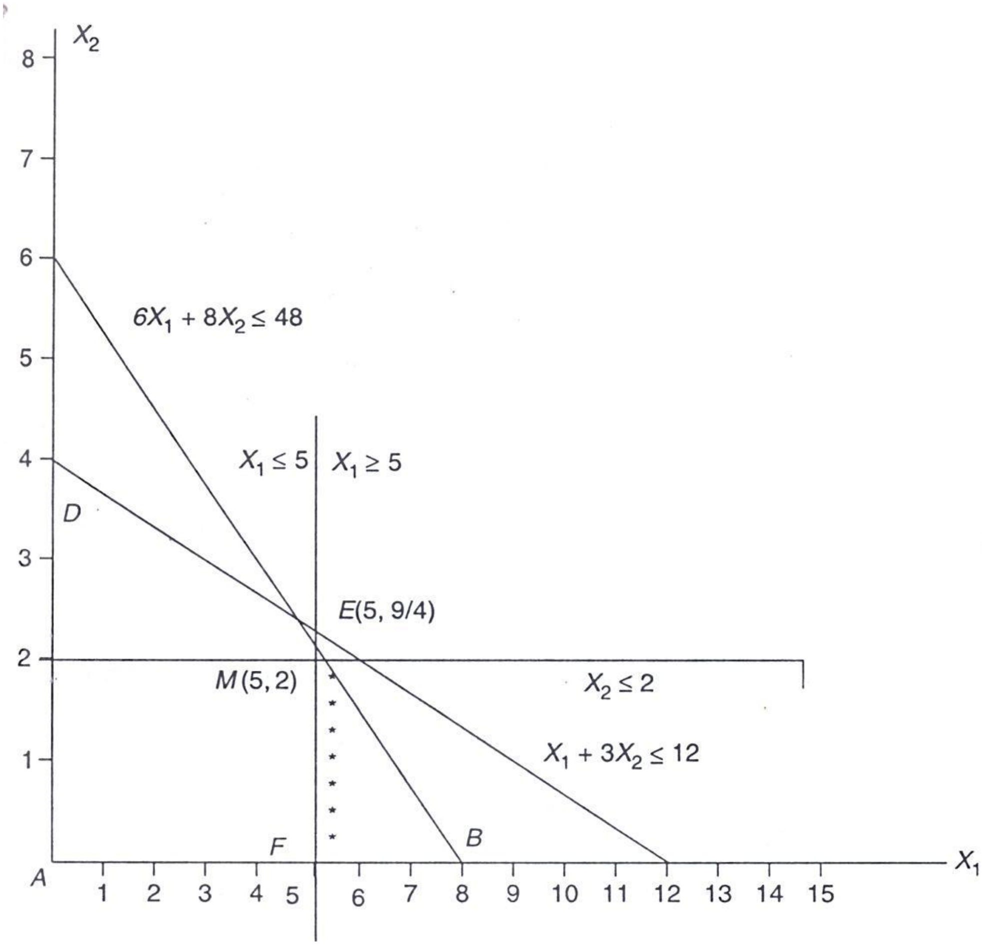


Figure 8 Feasible region of P_1 after introducing $X_1 \leq 5$ to P_5.

[Feasible region is the vertical line from M (5,2) to F (5,0) indicated By *S.]

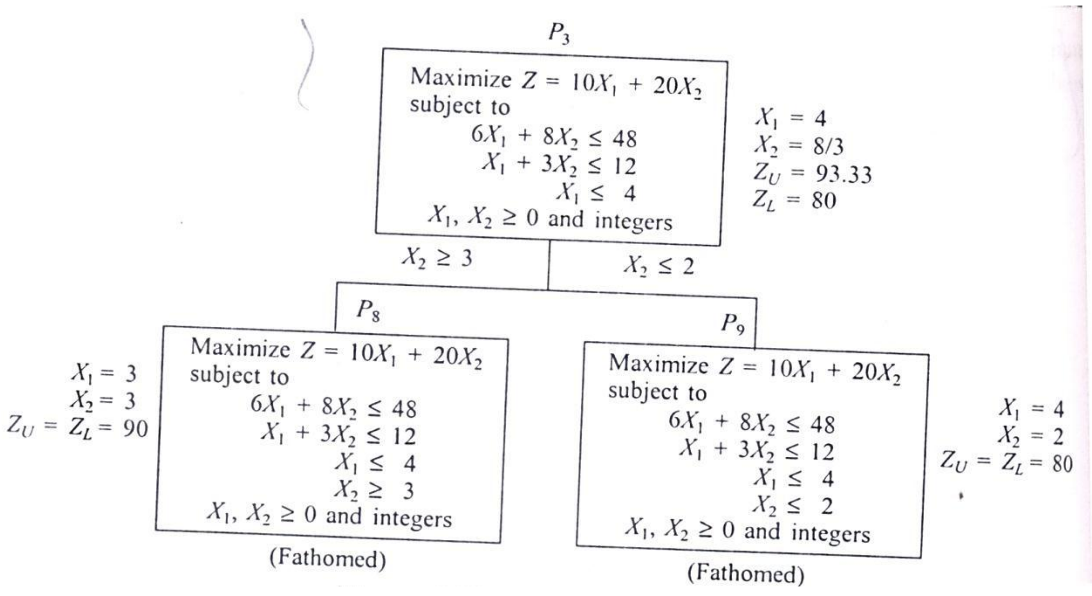


Figure 9 Branching From P_3

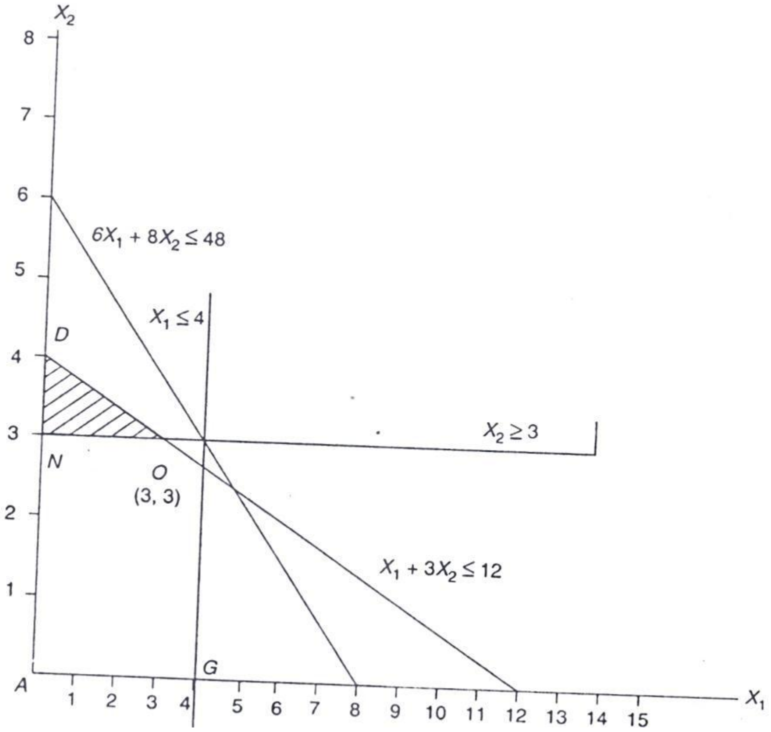


Figure 10 Feasible region of P_8 after introducing $X_2 \geq 3$ to P_3

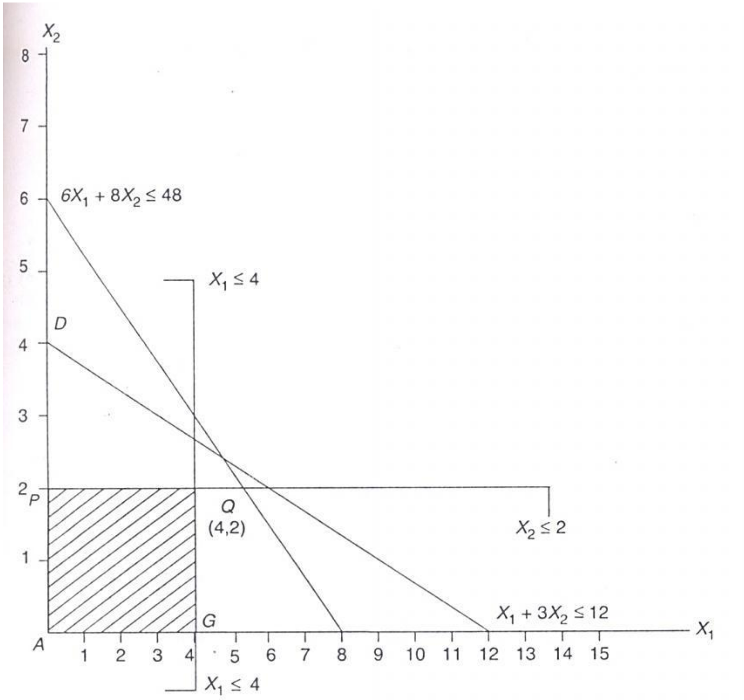


Figure 11 Feasible region of P_9 after introducing $X_2 \leq 2$ to P_3.

Now, all the terminal nodes are fathomed. The feasible fathomed node with the current best lower is P7. Hence, its solution is treated as the optimal solution as listed below. A complete branching tree is shown.

Module- II
SCHEDULING

Scheduling n Jobs on Parallel Identical Processors to Minimize Weighted Mean Flow Time

Example: to illustrate the procedure to minimize weighted mean flow time under single machine scheduling with parallel processors, consider the following example.

Job (j)	1	2	3	4	5	6	7	8	9
t_j	5	21	16	6	25	19	20	10	6
w_j	3	2	4	2	4	3	1	2	1
t_j/w_j	1.67	10.5	4	3	6.25	6.33	20	5	6

in the above table, the last row is computed mainly to represent the weighted processing time. Obtain a schedule which minimizes the weighted mean flow time is the number of parallel identical processors is three.

Solutions.

Step 1. From a priority list using weighed longest processing time (WLPT)

WLPT = {7,2,6,5,9,8,3,4,1}

Step 2.

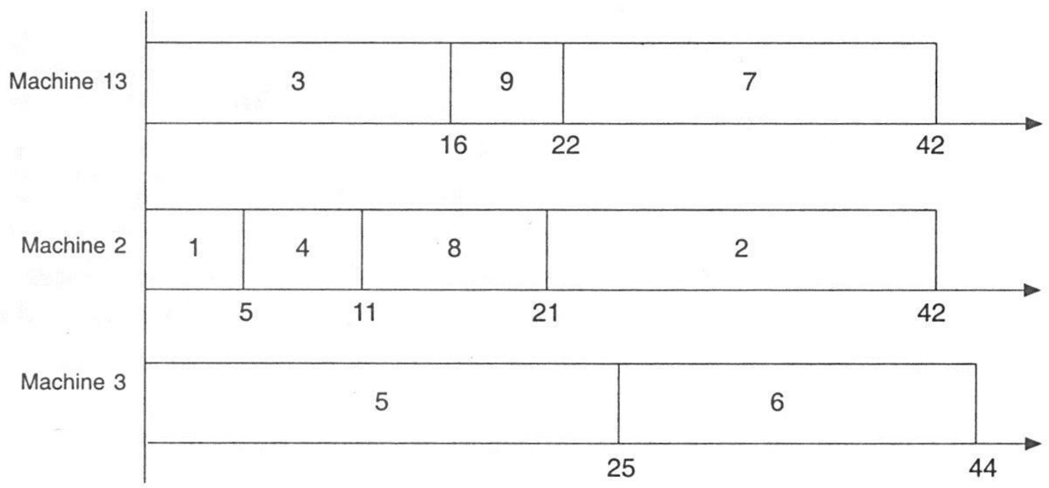
WLPT = {7, 2, 6, 5, 9, 8, 3, 4, 1}
Times = [20, 21, 19, 25, 6, 10, 16, 6, 5]

Stage	Processing Commitments Machine 1	Machine 2	Machine 3	Job Assigned	Machine Assigned
1	0	0	0	7	1
2	20	0	0	2	2
3	20	21	0	6	3
4	20	21	19	5	3
5	20	21	44	9	1
6	26	21	44	8	2
7	26	31	44	3	1
8	42	31	44	4	2
9	42	37	44	1	2
10	42	42	44	–	–

Step 3: apply WSPT sequencing to each machine (i.e. weighted shortest processing time).

	Machine 1	Machine 2	Machine 3
Jobs assigned	7 9 3	2 8 4 1	6 5
t_j/w_j	20 6 4	10.5 5 3 1.67	6.33 6.25
WSPT sequencing of jobs	3 – 9 – 7	1 – 4 – 8 – 2	5 – 6

So, finally schedule the jobs on each machine as per the WESPT sequencing are shown above. The above details are shown in the form a diagram in below.



Flow Shop Scheduling

In flow shop scheduling problem, there are n jobs; each requires processing on m different machines. The order in which the machines are required to process a job is called process sequence of that job. The process sequences of all the jobs are the same. But the processing times for various jobs on a machine may differ. If an operation is absent in a job, then the processing time of the operating of that job is assumed as zero.

The flow-shop scheduling problem can be characterized as given below:

1. A set of multiple-operation jobs is available for processing at time zero (Each job requires m operations and each operation requires a different machine).
2. Set-up times for the operations are sequence independent, and are included in processing times.
3. Job descriptors are known in advance.
4. M different machines are continuously available.
5. Each individual operation of jobs is processed till its completion without break.

JOHNSON'S PROBLEM

Example: Consider the following two machines and six jobs flow shop scheduling problem. Using Johnson's algorithm, obtain the optimal sequence which will minimize the makespan.

Job i	Machine 1	Machine 2
1	5	4
2	2	3
3	13	14
4	10	1
5	8	9
6	12	11

Solution: The working of the algorithm is summarized in the form of a table which is shown below.

Stage	Unscheduled Jobs	Minimum t_{ik}	Assignment	Partial Sequence
1	1, 2, 3, 4, 5, 6	t_{42}	4 = [6]	X X X X X 4
2	1, 2, 3, 5, 6	t_{21}	2 = [1]	2 X X X X 4
3	1, 3, 5, 6	t_{12}	1 = [5]	2 X X X 1 4
4	3, 5, 6	t_{51}	5 = [2]	2 5 X X 1 4
5	3, 6	t_{62}	6 = [4]	2 5 X 6 1 4
6	3	t_{31}	3 = [3]	2 5 3 6 1 4

The optimal sequence is 2-5-3-6-1-4

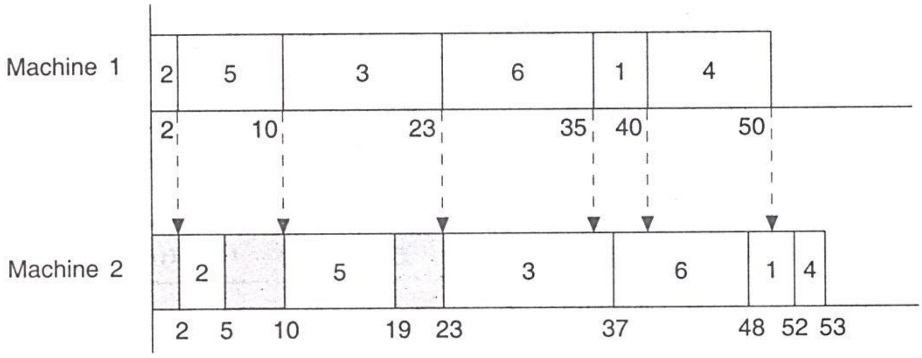
The makespan is determined as shown below:

In the following table

Time-in on Machine 2= max [Machine 1 time-out of the current job, Machine 2 time out- of the previous job]

	Processing Time				
	Machine 1		Machine 2		
Job	Time-in	Time-out	Time-in	Time-out	Idle Time on Machine 2
2	0	2	2	5	2
5	2	10	10	19	5
3	10	23	23	37	4
6	23	35	37	48	0
1	35	40	48	52	0
4	40	50	52	53	0

The makespan for this scheduled is 53. The makespan can also be obtained using Gantt chart which is shown in below.



N represents idle time

CDS HEURISTC

For large size problems, it would be difficult to get optimum solution in finite time, since the flow shop scheduling is a combinatorial this means the time complexity function of flow shop problem is exponential in nature. Hence, we have to use efficient heuristics for large size problem.

CDS (Campbell, Dudek and Smith) heuristic is one such heuristic used for flow shop scheduling the CDS heuristic corresponds to multistage use of Johnson's rule applied to a new problem formed from the original with processing times.

Find the makespan using the CDS heuristic for the following flow shop problem.

Job j	t_{j1}	t_{j2}	t_{j3}	t_{j4}
1	4	3	7	8
2	3	7	2	5
3	1	2	4	7
4	3	4	3	2

Solution

Stage 1

Job (j)	Machine 1 t_{j1}	Machine 2 t_{j2}
1	4	8
2	3	5
3	1	7
4	3	2

The optimal sequence for the above problem is shown below:

3-2-1-4

The makespan calculation for the above schedule is shown below.

	Processing Time (in hour)							
	Machine 1		Machine 2		Machine 3		Machine 4	
Job	In	Out	In	Out	In	Out	In	Out
3	0	1	1	3	3	7	7	14
2	1	4	4	11	11	13	14	19
1	4	8	11	14	14	21	21	29
4	8	11	14	18	21	24	29	31

Makespan of this problem = 31

Stage 2

Job (j)	Machine 1 t_{j1}	Machine 2 t_{j2}
1	7	15
2	10	7
3	3	11
4	7	5

After applying Johnson's algorithm to the above problem, we get sequence, 3-1-2-4. The makespan calculation is summarized in the following table.

	Processing Time (in hour)							
	Machine 1		Machine 2		Machine 3		Machine 4	
Job	In	Out	In	Out	In	Out	In	Out
3	0	1	1	3	3	7	7	14
1	1	5	5	8	8	15	15	23
2	5	8	8	15	15	17	23	28
4	8	11	15	19	19	22	28	30

The makespan for the sequence 3-1-2-4 is 30.

Stage 3

	Processing Time	
Job j	Machine 1 t_{j1}	Machine 2 t_{j2}
1	14	18
2	12	14
3	7	13
4	10	9

The application of Johnson's algorithm to the above data yields the sequence 3-2-1-4.

The determination of the corresponding makespan is shown below.

	Processing Time (in hour)							
	Machine 1		Machine 2		Machine 3		Machine 4	
Job	In	Out	In	Out	In	Out	In	Out
3	0	1	1	3	3	7	7	14
2	1	4	4	11	11	13	14	19
1	4	8	11	14	14	21	21	29
4	8	11	14	18	21	24	29	31

The makespan for the sequence 3-2-1-4 is 31.

Summary

Stage	Sequence	Makespan
1	3–2–1–4	31
2	3–1–2–4	30
3	3–2–1–4	31

The best sequence is 3-1-2-4 which has the makespan of 30.

THE TRAVELING SALESMAN PROBLEM (Shortest Cyclic Route Models)

There are a number of cities a salesman must visit. The distance (or time or cost) between every pair of cities is known. He starts from his home, city passes through each city once and only once and return to his home city. The problem is to find the routes shortest in distance (or time or cost).

If the distance (or time or cost) between every pair of cities is independent of direction of travel, the problem is said to be symmetrical.

Mathematical Statement of the Travelling Salesman Problem

Mathematically, the problem may be started as follows:

If C_{ij} is the cost of going from city i to city j and X_{ij} = 1, if the salesman goes directly from city i to j and zero otherwise, then the problem is to find X_{ij} which minimize

$$Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij},$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1,$$

$$\sum_{i=1}^{n} x_{ij} = 1,$$

$X_{ij=}$ 0 or 1; I = 1,2,…………,n; j = 1,2,……….., n,

Example

A salesman wants to visit cities 1,2,3 and 4. He does not want to visit any city twice before completing the tour of all the cities and wishes to return to his home city, the starting station. Cost of going from one city to another in rupees is given in table 4.180. find the least cost route.

Table 1

		1	2	3	4
From City	1	0	30	80	50
	2	40	0	140	30
	3	40	50	0	20
	4	70	80	130	0

Solution. This Traveling salesman problem can be first solved as assignment problem. If the optimal assignment table also satisfies the additional constraint that on city is to be visit twice before completing the tour of all the cities, it is also the optimal solution to the given travelling salesman problem. If it does not, it can be adjusted.

As going from city 1-1, 2-2, etc is not allowed, assign a large penalty $C_{ii}= \infty$ to these cells in table 1. The resulting table will have all diagonal elements ∞, subtract the smallest element of each row from all the elements of the row and if, necessary, the smallest element of each column from all elements of the column. This yields table 2 and table 3.

Table 2

	1	2	3	4
1	∞	0	50	20
2	10	∞	110	0
3	20	30	∞	0
4	0	10	60	∞

Reduced matric with zero in every row

→

Table 3

	1	2	3	4
1	∞	0	0	20
2	10	∞	60	0
3	20	30	∞	0
4	0	10	10	∞

Reduced matric with zero in every row and column

Check if optimal assignment can be made in the current reduced matrix. This is shown in table 4

Table 4

	1	2	3	4	
1	∞	$\boxed{0}$	$\cancel{0}$	20	
2	10	∞	60	$\boxed{0}$	✓
3	20	30	∞	$\cancel{0}$	✓
4	$\boxed{0}$	10	10	∞	
				✓	

First basic feasible solution

Table 5

	1	2	3	4	
1	∞	$\boxed{0}$	$\cancel{0}$	30	
2	$\cancel{0}$	∞	50	$\cancel{0}$	✓
3	10	20	∞	$\boxed{0}$	✓
4	$\boxed{0}$	10	10	∞	✓
	✓			✓	

Second basic feasible solution

Table 6

	1	2	3	4
1	∞	$\cancel{0}$	$\boxed{0}$	40
2	$\boxed{0}$	∞	40	$\cancel{0}$
3	10	10	∞	$\boxed{0}$
4	$\cancel{0}$	$\boxed{0}$	$\cancel{0}$	∞

Optimal solution

Table 6 provides an optimal solution to the assignment problem. According to it the salesman should visit city 1- 3- 4 -2 -1 and this involves a cost of Rs. (80+ 20 +80 + 40) = Rs. 220. This solution also satisfies the additional constraint of the travelling salesman problem. Thus the least cost route is 1- 3- 4- 2- 1 at a cost of Rs. 220.

Example

Product 1, 2, 3, 4 and 5 are to be processed on a machine. The set-up costs in rupees per change depend upon the product presently on the machine and the set-up to be made and are given by the following data;

$C_{12} = 16$, $C_{13} = 4$, $C_{14} = 12$, $C_{23} = 6$, $C_{34} = 5$, $C_{25} = 8$, $C_{35} = 6$, $C_{45} = 20$; $C_{ji,}$ $C_{ij} = \infty$ for all values of i and j not given in the data. Find the optimum sequence of products in order to minimize the total setup cost.

Solution. Determination of optimum order of processing the products so that the set-up costs are minimum is a travelling salesman problem. The set-up (changeover) costs between the products are analogous to distance between the cities. Each product must be produced only once and the production must return to the first product.

We first begin to solve this problem as an assignment problem. The given data is expressed in the form of a table 1

Table 1

	1	2	3	4	5
1	∞	16	4	12	∞
2	16	∞	6	∞	8
3	4	6	∞	5	6
4	12	∞	5	∞	20
5	∞	8	6	20	∞

Hungarian method or reduced matrix method will be used to obtain optimal assignment.

This method consists of the following steps:

Step I

Prepare a Square Matrix: This step is not necessary in this example

Step II

Reduce the Matrix: Proceeding as in example 4.6-2, we get table 2

Table 2

	1	2	3	4	5
1	∞	12	0	8	∞
2	10	∞	0	∞	2
3	0	2	∞	1	2
4	7	∞	0	∞	15
5	∞	2	0	14	∞

Matrix after sub step 1 (contains zero in each row)

After sub step 2 we get the following matrix:

Table 3

	1	2	3	4	5
1	∞	10	0	7	∞
2	10	∞	0	∞	0
3	0	0	∞	0	0
4	7	∞	0	∞	13
5	∞	0	0	13	∞

Matrix after sub step 2 (Contains zero in each row and in each column)
initial feasible solution

Step III

Check if optimal Assignment can be made in the current Feasible Solution or not proceeding as in example 4.6-2 we get

Table 4

	1	2	3	4	5	
1	∞	10	$\boxed{0}$	7	∞	✓
2	10	∞	~~0~~	∞	$\boxed{0}$	
3	$\boxed{0}$	~~0~~	∞	~~0~~	~~0~~	
4	7	∞	~~0~~	∞	13	✓
5	∞	$\boxed{0}$	~~0~~	13	∞	
			✓			

As the minimum number of lines crossing all zeros is 4 i.e., less than 5, optimal assignment cannot be made in the current feasible solution.

Step IV

Proceeding as in example 4.6-2 we get

Table 5

	1	2	3	4	5
1	∞	3	0	0	∞
2	10	∞	7	∞	0
3	0	0	∞	0	0
4	0	∞	0	∞	6
5	∞	0	7	13	∞

Step V

Check if Optimal Assignment can be made in the current Feasible Solution or not

Table 6

	1	2	3	4	5
1	∞	3	$\boxed{0}$	~~0~~	∞
2	10	∞	7	∞	$\boxed{0}$
3	~~0~~	~~0~~	∞	$\boxed{0}$	~~0~~
4	$\boxed{0}$	∞	~~0~~	∞	6
5	∞	$\boxed{0}$	7	13	∞

As there is no row or column without assignment, optimal assignment is possible in the current solution. Table 6, however provides an optimal solution to the assignment problem but not to the given travelling salesman problem (sequencing problem) as it gives 1- 3, 3-4, 4-1,2- 5 and 5-2 as the solution which means that the products should be processed in the order 1-3-4-1, without processing the product 2 and 5. This violates the additional constraint that wach product must be processed only once and only after having processed all the products, the production should return to product 1. So we try to find the 'next best' solution that also satisfies this additional constraint.

The nest minimum (non- zero) elements is 3 in cell (1, 2). We make assignment to entry 3 in this cell instead of zero assignment in cell (1, 3). Accordingly, zero assignment in cell (5, 2) is changed to assignment in cell (5, 3) with entry 7. This gives table 7.

Table 7

	1	2	3	4	5
1	∞	$\boxed{3}$	$\cancel{0}$	$\cancel{0}$	∞
2	10	∞	7	∞	$\boxed{0}$
3	$\cancel{0}$	$\cancel{0}$	∞	$\boxed{0}$	$\cancel{0}$
4	$\boxed{0}$	∞	$\cancel{0}$	∞	6
5	∞	$\cancel{0}$	$\boxed{7}$	13	∞

The resulting feasible solution 1-2, 2-5, 5-3, 3-4, 4-1 is also the optimal solution. Thus the optimal sequence for the processing of products is 1-2-5-3-4-1 and it involves a cost of Rs. (16 + 8 + 5 + 12+ 6) = Rs. 47.

TRANSPORTATION PROBLEM

Transportation problem is a special kind of linear programming problem in which goods are transported from a set of sources to a set of destinations subject to the supply and demand of the source and destination, respectively, such that the total cost of transportation is minimized.

MATHEMATICAL MODEL FOR TRANSPORTATION PROBLEM

In this section, a linear programming model for the transportation problem is presented.

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

		Destination (j)						
		1	2	⋯	j	⋯	n	Supply
	1	c_{11}	c_{12}	⋯	c_{1j}	⋯	c_{1n}	a_1
	2	c_{21}	c_{22}	⋯	c_{2j}	⋯	c_{2n}	a_2
	⋮	⋮	⋮		⋮		⋮	⋮
Source (i)	i	c_{i1}	c_{i2}	⋯	c_{ij}	⋯	c_{in}	a_i
	⋮	⋮	⋮		⋮		⋮	⋮
	m	c_{m1}	c_{m2}	⋯	c_{mj}	⋯	c_{mn}	a_m
Demand		b_1	b_2	⋯	b_j	⋯	b_n	

subject to

$$\sum_{j=1}^{n} X_{ij} \le a_i, \qquad i = 1, 2, 3, \ldots, m$$

and

$$\sum_{i=1}^{m} X_{ij} \ge b_j, \qquad j = 1, 2, 3, \ldots, n$$

where

$$X_{ij} \ge 0, \qquad i = 1, 2, 3, \ldots, m \quad \text{and} \quad j = 1, 2, 3, \ldots, n$$

TYPES OF TRANSPORTATION PROBLEM

1. **Balanced Transportation Problem**
   If the sum of the supplies of all the sources is equal to the sum of the demands of all the destinations, then the problem is termed as balanced transportation problem. This may be represented by the relation.

$$\sum_{i=1}^{m} ai = \sum_{j=1}^{n} bj$$

2. **Unbalanced Transportation Problem**

Of the sun of the supplies of all the sources is not equal to the sum of the demands of all the destination, then the problem is termed as unbalanced transportation problem. That means, for any unbalanced transportation problem we have

$$\sum_{i=1}^{m} ai \neq \sum_{j=1}^{n} bj$$

NORTHWEST CORNER CELL METHOD

Example: consider the following transportation problem involving three sources and four destinations. The cell entries represent the cost of transportation per unit.

Destination

	1	2	3	4	Supply
Source 1	3	1	7	4	300
Source 2	2	6	5	9	400
Source 3	8	3	3	2	500
Demand	250	350	400	200	1200

By northwest corner cell method. The supply and the demand values corresponding to the northwest corner cell (1, 1) are 300 and 250, respectively. The minimum of these values is 250. Hence, allocate 250 units to the cell (1,1) and subtract the same from the supply and demand values of the cell (1,1). Now the supply to destination 1 is fully met. Hence this column is deleted and the resultant data is shown in Table 2.

Table 1 Data of the Given problem with initial Allocation

Destination

	1	2	3	4	Supply
Source 1	250 3	1	7	4	~~300~~ 50
Source 2	2	6	5	9	400
Source 3	8	3	3	2	500
Demand	~~250~~ 0	350	400	200	

In Table- 2 the supply and the demand values corresponding to the northwest corner cell (1, 2) are 50 and 350, respectively. The minimum of these values is 50. Hence, we should allocated 50 units to the cell (1, 2) and subtract the same from the supply and demand values of the cell (1, 2)

Table 2 Result after Deleting Column 1

Source \ Destination	2	3	4	Supply
1	$\boxed{50}$ 1	7	4	~~50~~ 0
2	6	5	9	400
3	3	3	2	500
Demand	~~350~~ 300	400	200	

In this process, the supply of the source 1 is fully exhausted. Hence, this row is deleted and the resultant data is shown in Table 3

Table 3 Result after Deleting Row 1

Source \ Destination	2	3	4	Supply
2	$\boxed{300}$ 6	5	9	~~400~~ 100
3	3	3	2	500
Demand	~~300~~ 0	400	200	

In table 3 the supply and the demand values corresponding to the northwest corner cell (2, 2) are 400 and 300, respectively. The minimum o these values is 300. Hence, allocates 300 units to the cell (2, 2) and subtract the same from the supply and demand values of the cell (2, 2).

In this process, the demand of the destination 2 is fully satisfied. Hence, after deleting this column, the resultant data is shown in Table 4

Table 4 Result after Deleting Column 2

Source \ Destination	3	4	Supply
2	$\boxed{100}$ 5	9	~~100~~ 0
3	3	2	500
Demand	~~400~~ 300	200	

In table 4, the supply and the demand values corresponding to the northwest corner cell (2, 3) are 100 and 400, respectively. The minimum of these values is 100. Hence, we should allocate 100 units to the cell (2, 30 and subtract the same from the supply and demand values of the cell (2, 3).

In this process, the supply of the sources 2 is fully exhausted. Hence, this row is deleted and the resultant data is shown in Table 5.

Table 5 Result after Deleting Row 2

		Destination 3	Destination 4	Supply
Source	3	$\boxed{300}$ 3	$\boxed{200}$ 2	~~500~~ 0
Demand		~~300~~ 0	~~200~~ 0	

In table 5, only source is left out. Hence, the demands of the destinations 3 and 4 need to be matched with the supply of the source 3. The initial basic feasible solution for the given problem using the northwest corner cell method is shown in Table 6.

Table 6 Initial Basic Feasible Solution Using Northwest Corner Cell Method.

		Destination 1	Destination 2	Destination 3	Destination 4	Supply
	1	$\boxed{250}$ 3	$\boxed{50}$ 1	7	4	300
Source	2	2	$\boxed{300}$ 6	$\boxed{100}$ 5	9	400
	3	8	3	$\boxed{300}$ 3	$\boxed{200}$ 2	500
Demand		250	350	400	200	

The total cost of the solution is Rs. 4400. The total cost, is calculated by adding the priducts of the transportation cost per unit in each and every basic cell and the corresponding number of units allocated to it. A basic cell is one which has a positive allocation. Thus,

Total cost = 3 x 250 + 1 x 50 + 6 x 300 + 5 x 100 + 3 x 300 + 2 x 200 = Rs. 4400.

QUEUEING MODEL

In many real-word applications such as railways and airlines reservation counters, bank counters, gasoline stations etc. incoming customers become part of the respective queueing system. In fact waiting for service has become an integral part of our daily life. Albeit at a considerable cost most of the times. However, the adverse impact of the queueing up phenomena can be brought down to a minimum by applying various queueing models.

(M/M/1) : (GD/∞/∞) Model

The parameters of this model are given as follows:

i. Arrival rate follows poisson distribution.
ii. Service rate follows poision distribution.
iii. Number of servers is one.
iv. Service discipline is general discipline.
v. Maximum number of customers permitted in the system is infinite.
vi. Size of the calling source is finite.

The steady-state formula to obtain the probability of having n customers in the system P_n and the formulas for P_o, L_s, L_q ,W_s and W_q are presented below:

$$P_n = (1- \emptyset)\, \emptyset^{n}, \; n= 0,1,2,3,\ldots\ldots\ldots, \infty \text{ where }, \emptyset= \frac{\delta}{\mu} < 1$$

$$L_s = \frac{\emptyset}{1-\emptyset}$$

$$L_q = L_s - \frac{\emptyset}{\mu} = \frac{\emptyset\ 2}{1-\emptyset}$$

$$W_s = \frac{Ls}{\delta} = \frac{1}{(1-\emptyset)\mu} = \frac{1}{\mu-\delta}$$

$$W_q = \frac{Lq}{\delta} = \frac{\emptyset}{\mu-\delta}$$

Example: The arrival rate of customers at a banking counter follows poisson distribution with a mean of 45 per hour. The service rate of the counter clerk also follows poisson distribution with a mean of 60 per hour.

a) What is the probability of having 0 customer in the system (P_o)?
b) What is the probability of having 5 customers in the system (P_5)?
c) What is the probability of having 10 customers in the system (p_{10})?
d) Find L_s, L_q and W_q.

Solution We have follows data:

Arrival rate, δ = 45 per hour

Service rate, μ = 60 per hour

Utilization factor, $\emptyset = \frac{\delta}{\mu} = \frac{45}{60} = 0.75$

(a) $p_0 = 1 - \phi = 1 - 0.75 = 0.25$

(b) $p_5 = (1 - \phi)\phi^5 = (1 - 0.75)0.75^5 = 0.0593$

(c) $p_{10} = (1 - \phi)\phi^{10} = (1 - 0.75)0.75^{10} = 0.0141$

(d) $L_s = \frac{\phi}{1 - \phi} = \frac{0.75}{1 - 0.75} = 3$ customers

$$L_q = \frac{\phi^2}{1 - \phi} = \frac{0.75^2}{1 - 0.75} = 2.25 \text{ customers}$$

$$W_s = \frac{1}{\mu - \delta} = \frac{1}{60 - 45} = 0.067 \text{ hour}$$

$$W_q = \frac{\phi}{\mu - \delta} = \frac{0.75}{60 - 45} = 0.05 \text{ hour.}$$

(M/M/C): (GD/∞/∞) Model

The parameters of this models are as follows.

I. Arrival rate follows Poisson Distribution
II. Service rate follows Poisson distribution.
III. Number of service is C.
IV. Service discipline is general discipline
V. Maximum number of customers permitted in the system is infinite.
VI. Size of the calling source is infinite.

The steady-state formula to obtain the probability of having n customers in the system P_n, and the formula for P_o, L_s, L_q, W_q and W_s are presented below.

$$p_n = \frac{\phi^n}{n!} p_0, \qquad 0 \le n \le C$$

$$= \frac{\phi^n}{C^{n-C} C!} p_0, \qquad n > C$$

$$\frac{\phi}{C} < 1 \quad \text{or} \quad \frac{\delta}{\mu C} < 1$$

Where,

$$p_0 = \left\{ \sum_{n=0}^{C-1} \frac{\phi^n}{n!} + \frac{\phi^C}{C![1 - (\phi/C)]} \right\}^{-1}$$

$$L_q = \frac{\phi^{C+1}}{(C-1)!\,(C-\phi)^2}\, p_0 = \frac{C\phi p_C}{(C-\phi)^2}$$

$$L_s = L_q + \phi$$

$$W_q = \frac{L_q}{\delta}$$

$$W_s = W_q + \frac{1}{\mu}$$

Now the formula for P_o and L_q under special conditions are:

$$p_0 \approx 1 - \phi, \quad L_q \approx \frac{\phi^{C+1}}{C^2}, \qquad \text{where } \phi \ll 1$$

and

$$p_0 \approx \frac{(C-\phi)(C-1)!}{C^C} \quad \text{and} \quad L_q = \frac{\phi}{C-\phi}, \qquad \text{where } \frac{\phi}{C} \approx 1.$$

Example: at a central warehouse, vehicles arrive at the rate of 18 per hour and the arrival rate follow's Poisson distribution. The unloading time of the vehicles follows exponential distribution and the unloading rate is 6 vehicles per hour. There are 4 unloading crews. Find the following.

(a) P_o and P_3
(b) L_q, L_s, W_q and W_s

Solution we have

Arrival rate, δ = 18 per hour

Unloading rate, μ = 16 per hour

Number of unloading crews, C = 4

And

(a) Therefore, P_o is computed as:

$$p_0 = \left\{\sum_{n=0}^{C-1} \frac{\phi^n}{n!} + \frac{\phi^C}{C![1-(\phi/C)]}\right\}^{-1}$$

$$= \left\{\sum_{n=0}^{3} \frac{3^n}{n!} + \frac{3^4}{4![1-(3/4)]}\right\}^{-1}$$

$$= \left\{\frac{3^0}{0!} + \frac{3^1}{1!} + \frac{3^2}{2!} + \frac{3^3}{3!} + \frac{3^4}{4![1-(3/4)]}\right\}^{-1}$$

$$= 0.0377$$

Now, to compute P_3, we have

$$p_n = \frac{\phi^n}{n!} p_0, \; 0 \le n \le C$$

Therefore,

$$p_3 = \frac{\phi^3}{3!} p_0 = \frac{3^3}{6} \times 0.0377 = 0.1697$$

(b) L_q, L_s, W_q and W_s are computed as under:

$$L_q = \frac{\phi^{C+1}}{(C-1)! \times (C-\phi)^2} p_0 = \frac{3^5}{3! \times 1} \times 0.0377 = 1.53 \approx 2 \text{ vehicles}$$

$$L_s = L_q + \phi = 1.53 + 3 = 4.53 \approx 5 \text{ vehicles}$$

$$W_q = \frac{L_q}{\delta} = \frac{1.53}{18} = 0.085 \text{ hour} = 5.1 \text{ minutes}$$

$$W_s = W_q + \frac{1}{\mu} = 0.085 + \frac{1}{6} = 0.252 \text{ hour} = 15.12 \text{ minutes.}$$

7

(M/M/1):)GD/N/∞ Model S

The parameters of this model are defined below:

I. Arrival rate follows Poisson distribution
II. Service rate follows Poisson distribution.
III. Number of servers is one.
IV. Service discipline is general discipline.
V. Maximum number of customers permitted in the system is N

VI. Size of the calling source is infinite.

The steady- state formula to obtain the probability of having n customers in the system P_n, and the formulas for P_o, L_s, L_q, W_s and W_q are presented below:

$$p_n = \frac{1-\phi}{1-\phi^{n+1}}\phi^n, \qquad \phi \neq 1 \text{ and } n = 0, 1, 2, 3, \ldots, N$$

$$= \frac{1}{N+1}, \qquad \phi = 1$$

$$L_s = \frac{\phi\left[1-(N+1)\phi^N + N\phi^{N+1}\right]}{(1-\phi)(1-\phi^{N+1})}, \qquad \phi \neq 1$$

$$= \frac{N}{2}, \qquad \phi = 1$$

$$\delta_{\text{eff}} = \delta(1-p_N) = \mu(L_s - L_q)$$

$$L_q = L_s - \frac{\delta_{\text{eff}}}{\mu} = L_s - \frac{\delta(1-p_N)}{\mu}$$

$$W_q = \frac{L_q}{\delta_{\text{eff}}} = \frac{L_q}{\delta(1-p_N)}$$

$$W_s = W_q + \frac{1}{\mu} = \frac{L_s}{\delta_{\text{eff}}} = \frac{L_s}{\delta(1-p_N)}.$$

Example cars arrive at a drive-in restaurant with a mean arrival rate of 24 cars per hour and the service rate of the cars is 20 cars per hours. The arrival rate and the service rate follow Poisson distribution. The number of parking space for cars is only 4. Find the standard results of this system.

Solution here

Arrival rate, δ = 24 cars per hour

Service rate, μ = 20 cars per hour

N= 4

$\emptyset = \frac{\delta}{\mu} = \frac{24}{20} = 1.2$

Therefore, we get

$$L_s = \frac{\phi\left[1 - (N+1)\,\phi^N + N\phi^{N+1}\right]}{(1-\phi)(1-\phi^{N+1})} = \frac{1.2\left[1 - (4+1)1.2^4 + 4 \times 1.2^5\right]}{(1-1.2)(1-1.2^5)} = 2.36 \text{ cars}$$

and

$$p_N = \frac{1-\phi}{1-\phi^{N+1}}\,\phi^N = \frac{1-1.2}{1-1.2^5} \times 1.2^4 = 0.2787$$

The other results are:

$$\delta_{\text{eff}} = \delta(1 - p_N) = 24(1 - 0.2787) = 17.3112 \text{ per hour}$$

$$L_q = L_s - \frac{\delta_{\text{eff}}}{\mu} = 2.36 - \frac{17.3112}{20} = 1.494 \text{ cars}$$

$$W_q = \frac{L_q}{\delta_{\text{eff}}} = \frac{1.494}{17.3112} = 0.0863 \text{ hour} = 5.2 \text{ minutes}$$

$$W_s = \frac{L_s}{\delta_{\text{eff}}} = \frac{2.36}{17.3112} = 0.1363 \text{ hour} = 8.2 \text{ minutes.}$$

Module -III

Bin Packing — Portfolio optimization

INTRODUCTION

The Bin-Packing Problem (BPP) can be described, using the terminology of knapsack problems, as follows. Given n items and n knapsacks (or bins), with

W_j = weight of item j,

c = capacity of each bin,

Assign each item to one bin so that the total weight of the items in each bin does not exceed c and the number of bins used is a minimum. A possible mathematical formulation of the problem is

$$\text{minimize} \quad z = \sum_{i=1}^{n} y_i$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_{ij} \le c y_i, \quad i \in N = \{1, \dots, n\},$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j \in N,$$

$$y_i = 0 \text{ or } 1, \quad i \in N,$$

$$x_{ij} = 0 \text{ or } 1, \quad i \in N, j \in N,$$

where

$$y_i = \begin{cases} 1 & \text{if bin } i \text{ is used;} \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if item } j \text{ is assigned to bin } i; \\ 0 & \text{otherwise.} \end{cases}$$

We will suppose, as is usual, that the weights W_j are positive integers. Hence, without loss of generality, we will also assume that

$$c \quad \text{is a positive integer,}$$

$$w_j \leq c \quad \text{for } j \in N.$$

If assumption (8.6) is violated, C can be replaced by [c]. If an item violates assumption (8.7), then the instance is trivially infeasible. There is no easy way, instead, of transforming an instance so as to handle negative weights.

For the sake of simplicity we will also assume that, in any feasible solution, the lowest indexed bins are used, i.e. $y_i \geq y_i+1$ for i= 1,…., n -1.

Almost the totality of the literature on BPP is concerned with approximate algorithms and their performance. A thorough analysis of such results would require a separate book (the brilliant survey by Coffman, Garey and Johnson 1984), to which the reader is referred, includes a bibliography of more than one hundred references, and new results continue to appear in the literature). In Section 8.2 we briefly summarize the classical results on approximate algorithms. The remainder of the chapter is devoted to lowerbounds (Section 8.3), reduction procedures (Section 8.4) and exact algorithms (Section 8.5), on which very little can be found in the literature. Computational experiments are reported in Section 8.6

Example

Consider the instance of BPP defined by

n =9,

(W_j) = (70, 60, 50, 33, 33, 33, 11,7, 3),

c = 100

An optimal solution requires 4 bins for item sets {1, 7, 8, 9}, {2,4},{3,5} and

{6},respectively.

From (8.14),

Li = [300/100] =3.

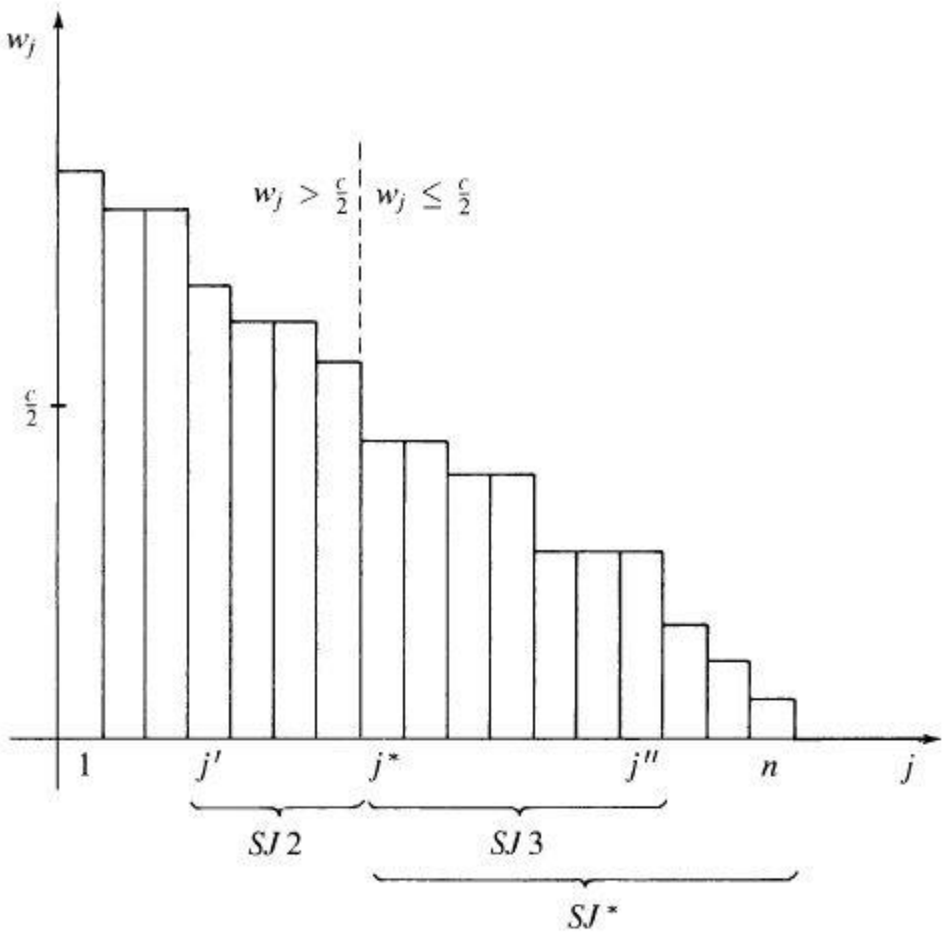
In order to determine L_2 we compute, using (8.19) and Corollary 8.4,

L(50) = 2 +0+ max (0, [50 -0)/ 100]) = 3

L (33)= 1+ 1 +max (0, [(149- 40)/100])= 4;

Since at this point we have 1+ 1+ [170-40)/100] = 4, the computation can be terminated with L_2 = 4.

The following procedure efficiently computes L_2. It is assumed that, on input, the items are sorted according to (8.11) and Wn ≤\ C/2. If (W_n > C/2 then, trivially, L_2 = n = z.) Figure 8.1 illustrates the meaning of the main variables of the procedure.



Example

Consider the instance of BPP defined by

n = 14,

(W_j) = (99, 94, 79, 64, 50, 46, 43, 37, 32, 19,18,7, 6, 3),

c = 100.

The first execution of MTRP gives

j =l: k =0, F = {1};

j =2: k =0, f = {1}

j = 2 : k = 1, j^* = 13, F = {2. 13},

and F = ∅ for j ≥3. Hence

z =2; (bj) = (1,2,0,0, 0, 0, 0, 0, 0, 0, 0, 0,2,0);

Executing L2 for the residual instance we get L_2 = 4, so

L3=6.

Item 14 is now removed and MTRP is applied to item set {3, 4,..., 12}, producing (indices refer to the original instance)

Exact algorithms

$$j = 3 : k = 1, j^* = 10, F = \{3, 10\};$$
$$j = 4 : k = 2, j^* = 9, j_a = 11, j_b = 12, F = \{4, 9\};$$
$$j = 5 : k = 2, j^* = 6, j_a = 7, j_b = 12, F = \emptyset;$$
$$j = 6 : k = 2, j^* = 5, j_a = 7, j_b = 12, F = \{6, 5\};$$
$$j = 7 : k = 2, j^* = 8, j_a = 8, j_b = 11, F = \{7, 8, 11\};$$
$$j = 12 : k = 0, F = \{12\};$$

Numbering the new bins with 3, 4,..., 7 we thus obtain

z = 7; (bj) = A, 2, 3, 4, 5, 5, 6, 6, 4, 3, 6, 7, 2, -);

Hence L2 = 0 (since Tf = 0) and the execution terminates with L3 = 7. Noting now that the eliminated item 14 can be assigned, for example to bin 4, we conclude that all reductions are valid for the original instance. The solution obtained (with ft 14 = 4) is also optimal, since all items are assigned

Example

Consider the instance of MTP defined by

n = 10;

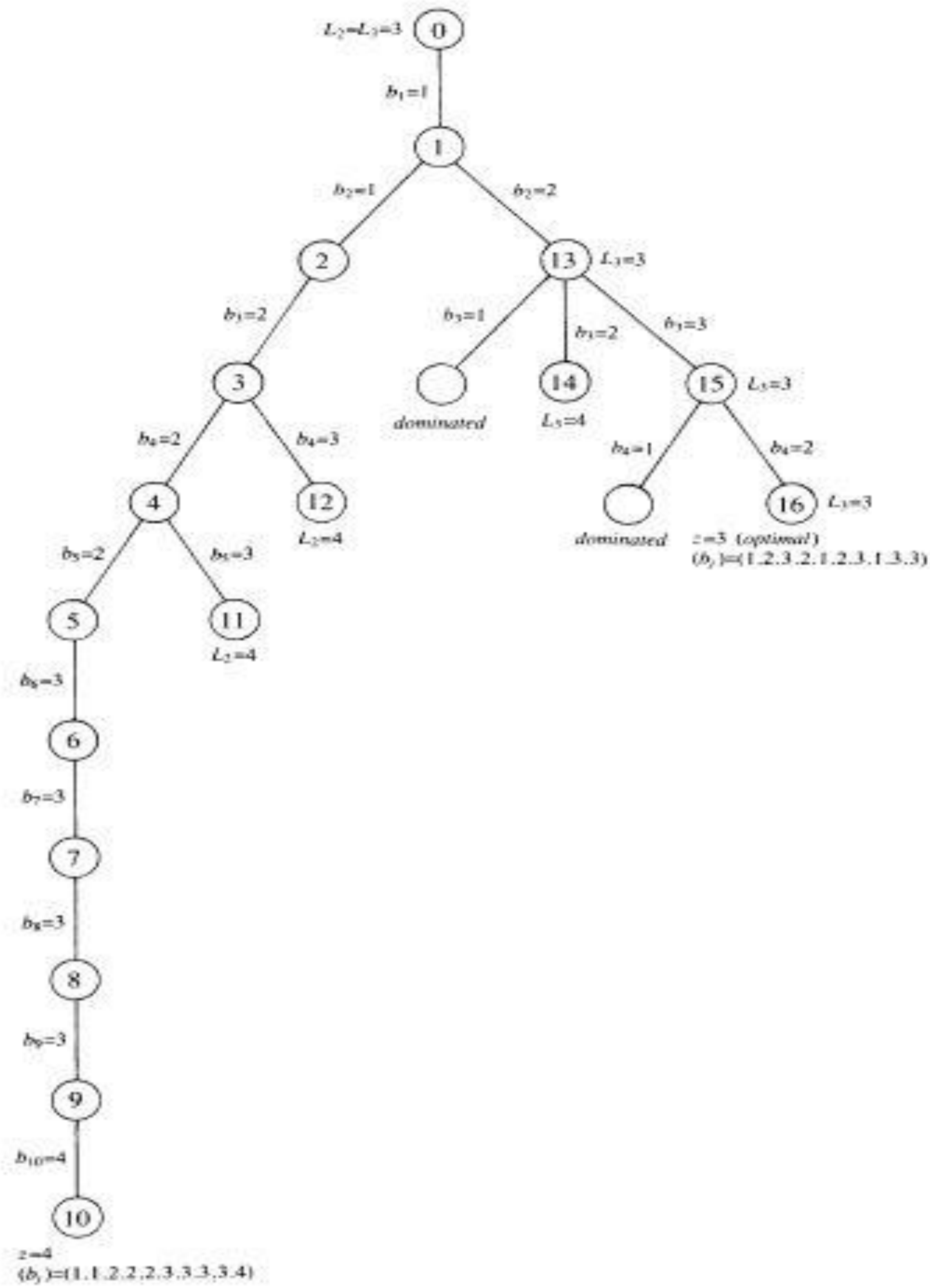
(w_y) = (49, 41, 34, 33, 29, 26, 26, 22, 20, 19);

c = 100.

We define a feasible solution through vector (b_j), with

b_j = bin to which item j is assigned (j = 1, ...n);

Figure 8.2 gives the decision-tree produced by algorithm MTP. Initially, all lower bound computations give the value 3, while approximate algorithm FFD gives the first feasible solution



z =4,

(b_j) = (1, 1, 2, 2, 2, 3, 3, 3, 3, 4),

Corresponding to decision-nodes 1-10. No second son is generated by nodes 5-9, since this would produce a solution of value 4 or more. Nodes 11 and 12 are fathomed by lower bound L_j. The first son of node 2 initializes bin 2, so no further son is generated. The first son of node 13 is dominated by node 2, since in both situations no further item can be assigned to bin 1; for the same reason node 2 dominates the first son of node 15.Node 14 is fathomed by lower bound L_j. At node 16, procedure MTRP (called by L_3) is applied to problem.

n = 9,

(W_j)= (74, 49, 34, 29, 26, 26, 22, 20, 19),

c = 100,

and optimally assigns to bin 2 the first and fifth of these items (corresponding to items 2, 4 and 6 of the original instance). Then, by executing the approximate algorithm FFD for the reduced instance.

(w_j) = (-, -, -, -, 29,-, 26,22,20,19),

(C_i) = (51,0, 66, 100, 100, ...),

where r, denotes the residual capacity of bin /, we obtain

(b_j) = (_ _ _ _ i,_ 3, 1,3, 3),

hence an overall solution of value 3, i.e. optimal.

The FORTRAN implementation of algorithm MTP is included in the present volume.

KUHN- TUCKER CONDITIONS

Consider the following general form of nonlinear programming problem which is having a maximization objective function with all less than or equal to type constraints.

$$\text{Maximize } Z = f(X_1, X_2,\ldots\ldots, X_j,\ldots\ldots X_n)$$

Subject to $G_i\,(X_1, X_2,\ldots\ldots, X_j \ldots.., X_n) \leq b_i$ $\quad i= 1,2,\ldots..,m$

$$X_j \geq 0, \quad j= 1,2,3\ldots\ldots,n$$

A modified form of the above model is

$$Z = f(X_1, X_2,\ldots\ldots, X_j,\ldots\ldots X_n)$$

Subject to

$$g_i\,(X_1, X_2,\ldots\ldots, X_j \ldots.., X_n) \leq 0 \qquad i= 1,2,\ldots..,m$$

$$X_j \geq 0, \quad j= 1,2,3\ldots\ldots,n$$

Where

$$g_i\,(X_1, X_2,\ldots\ldots, X_j \ldots.., X_n) = G_i\,(X_1, X_2,\ldots\ldots, X_j \ldots.., X_n) - b_i$$

Again the above models is modified as:

$$\text{Maximize } Z = f(X_1, X_2,\ldots\ldots, X_j,\ldots\ldots X_n)$$

Subject to

$$g_i\,(X_1, X_2,\ldots\ldots, X_j \ldots.., X_n) + S^2_i = 0, \; i= 1,2,\ldots..m$$

$$Xj \geq 0, \; j = 1,2,3,\ldots\ldots n$$

Where S^2_i is a complementary slack of the ith constraint. This model consists of n+ m variable and m constraint. Let be the Lagrangean function, and $\emptyset_i$ be the Lagrangean multiplier of the ith constraint. Then

$$L[(X_1, X_2,\ldots, X_n), (S_1, S_2,\ldots, S_m), (\phi_1, \phi_2,\ldots, \phi_m)]$$
$$= f(X_1, X_2,\ldots, X_j,\ldots, X_n) - \sum_{i=1}^{m} \phi_i[g_i(X_1, X_2,\ldots, X_j,\ldots, X_n) + S_i^2]$$

For the above maximization problem with concave objective function and with all less than or equal to type constraints (convex type constraints), the value of $\emptyset_i$ should be greater than or equal to 0. To maintain this relation of ϕ_i Khun- Tucker has established the following necessary conditions:

(a) $\phi_i \geq 0,\ i = 1, 2, 3, \ldots, m$

(b) $\dfrac{\delta L}{\delta X_j} = 0, \quad j = 1, 2, 3, \ldots, n$ (first partial derivatives)

(c) $\phi_i g_i(X_1, X_2, \ldots, X_n) = 0, \quad i = 1, 2, 3, \ldots, m$

(d) $g_i(X_1, X_2, \ldots, X_n) \leq 0, \quad i \leq 1, 2, 3, \ldots, m$

Example Solve the following nonlinear programming problem using Khun- Tucker conditions:

$$\text{Maximize } Z = 3X_1^2 + 14X_1X_2 - 8X_2^2$$

subject to

$$3X_1 + 6X_2 \leq 72$$
$$X_1 \text{ and } X_2 \geq 0$$

Solution The given problem is modified as:

$$\text{Maximize } Z = 3X_1^2 + 14X_1X_2 - 8X_2^2$$

subject to

$$3X_1 + 6X_2 - 72 \leq 0$$
$$X_1 \text{ and } X_2 \geq 0$$

Then, we have the Lagrangean function:

$$L = 3X_1^2 + 14X_1X_2 - 8X_2^2 - \phi(3X_1 + 6X_2 - 72)$$

The four sets of Kuhn–Tucker conditions are as given below:

(a) $\phi \geq 0$ (17.16)

(b) $\dfrac{\delta L}{\delta X_1} = 6X_1 + 14X_2 - 3\phi = 0, \quad \text{or} \quad 6X_1 + 14X_2 = 3\phi$ (17.17)

$\dfrac{\delta L}{\delta X_2} = 14X_1 - 16X_2 - 6\phi = 0, \quad \text{or} \quad 14X_1 - 16X_2 = 6\phi$ (17.18)

(c) $\phi(3X_1 + 6X_2 - 72) = 0$ (17.19)

(d) $3X_1 + 6X_2 - 72 \leq 0$ (17.20)

From Eqs. (17.17) and (17.18), we get

$$-2X_1 + 44X_2 = 0 \tag{17.21}$$

From Eq. (17.19), if ϕ is equated to 0, X_1 and X_2 should be equal to 0, which is not true. Therefore,

$$3X_1 + 6X_2 - 72 = 0, \quad \text{or} \quad 3X_1 + 6X_2 = 72 \tag{17.22}$$

By solving Eqs. (17.21) and (17.22), we get

$$X_1^* = 22, \qquad X_2^* = 1 \qquad Z(\text{maximum}) = 1752$$

Example Solve the following nonlinear programming problem using Khun- Tucker conditions.

$$\text{Maximize } Z = X_1^2 + X_1X_2 - 2X_2^2$$

subject to

$$4X_1 + 2X_2 \leq 24$$
$$5X_1 + 10X_2 \leq 30$$
$$X_1, X_2 \geq 0$$

Solution The given problem is modified as:

$$\text{Maximize } Z = X_1^2 + X_1X_2 - 2X_2^2$$

subject to

$$4X_1 + 2X_2 - 24 \leq 0$$
$$5X_1 + 10X_2 - 30 \leq 0$$
$$X_1, X_2 \geq 0$$

The Lagrangean function of this model is:

$$L = X_1^2 + X_1X_2 - 2X_2^2 - \phi_1 (4X_1 + 2X_2 - 24) - \phi_2 (5X_1 + 10X_2 - 30) \quad (17.23)$$

The four sets of Kuhn-Tucker conditions are as given below:

(a) $\phi_1 \geq 0$ (17.24)

$\phi_2 \geq 0$ (17.25)

(b) $\delta L/\delta X_1 = 2X_1 + X_2 - 4\phi_1 - 5\phi_2 = 0$ (17.26)

$\delta L/\delta X_2 = X_1 - 4X_2 - 2\phi_1 - 10\phi_2 = 0$ (17.27)

(c) $\phi_1(4X_1 + 2X_2 - 24) = 0$ (17.28)

$\phi_2(5X_1 + 10X_2 - 30) = 0$ (17.29)

(d) $4X_1 + 2X_2 - 24 \leq 0$ (17.30)

$5X_1 + 10X_2 - 30 \leq 0$ (17.31)

From equation (17.28), if ϕ_1 is equated to 0, X_1 and X_2 must be equal to 0, which is not true. Therefore,

$$4X_1 + 2X_2 - 24 = 0 \quad (17.32)$$

From equation (17.29), if ϕ_2 is equated to 0, X_2 and X_3 must be equal to 0, which is not true. Therefore,

$$5X_1 + 10X_2 - 30 = 0 \quad (17.33)$$

Now, solving equations (17.26), (17.27), (17.32) and (17.33) gives the following solution:

$X_1^* = 6$, $X_2^* = 0$, $\phi_1^* = 3$, $\phi_2^* = 0$, and

$Z^*(\text{maximum}) = 36$

WOLFE'S METHOD

$$\begin{aligned}\text{Maximize } Z &= c_1X_1 + c_2X_2 + \cdots + c_nX_n \\ &\quad + c_{n+1}X_1^2 + c_{n+2}X_2^2 + \cdots + c_{2n}X_n^2 \\ &\quad + c_{2n+1}X_1X_2 + c_{2n+2}X_1X_3 + \cdots + c_{2n+n_{c_2}}X_{n-1}X_n\end{aligned}$$

subject to

$$a_{i1}X_1 + a_{i2}X_2 + \cdots + a_{in}X_n \le b_i, \qquad i = 1, 2,\ldots, m$$

$$X_1, X_2,\ldots, X_n \ge 0$$

The above quadratic programming model may be presented in short, follows:

$$\text{Maximize } Z = CX + X^T DX$$

Subject to

$$AX \leq B, \text{ i= 1,2,3,……, m}$$

$$X \geq 0$$

Where

$$C = [c_1, c_2,\ldots, c_n]$$

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \qquad D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i1} & \cdots & a_{in} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \qquad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_m \end{bmatrix}$$

In matric [D], djj represents the coefficient of the term X^2_j in the objective function and djj (when I is not equal to j) represents the coefficient of the term Xi Xj in the objective function.

The above model is further modified by balancing the constrains as shown below.

$$\text{Maximize } Z = CX + X^T DX$$

Subject to

$$AX + S_i^2 = B, \quad i = 1, 2, 3, \ldots, m \qquad \text{(Type I constraints)}$$

$$X_j \geq 0, \quad j = 1, 2, 3, \ldots, n; \qquad -X_j \leq 0, \quad j = 1, 2, 3, \ldots, n; \qquad \text{(Type II constraints)}$$

Let $\emptyset_i$ be the Lagrangean multiplier of the *i*th constraint in type I set of constraints, μ be the Lagrangean multiplier associate with the *j*th constraint in type II set of constraints, and S_i^2 be the complementary slack of the constraint I of the I. application of Kuhn- Tucker conditions to this model results into the following system of constraints.

$$\left|\begin{array}{c|c|c|c} -2D & A^T & -I & 0 \\ \hline A & 0 & 0 & I \end{array}\right| \begin{bmatrix} X \\ Q \\ M \\ S \end{bmatrix} = \begin{bmatrix} C^T \\ B \end{bmatrix}$$

$$\mu_j X_j = 0, \qquad j = 1, 2, 3, \ldots, n$$

$$\phi_i S_i = 0, \qquad i = 1, 2, 3, \ldots, m$$

where, I is an identity matrix, and

$$Q = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_i \\ \vdots \\ \phi_m \end{bmatrix} \qquad M = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_j \\ \vdots \\ \mu_n \end{bmatrix} \qquad S = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_i \\ \vdots \\ S_m \end{bmatrix},$$

Procedure to solve quadratic programming problem

Step 1: The system of constraint is to be formed first.

Step 2: Next, add Rj as the artificial variable for the jth constraint of the first n constraints, since each of these first n constraints has no basic variable in it.

Step 3: A minimization type objective function is then formed by summing the artificial variables.

Step 4: Solve the model consisting of the above objective function and system of constraints using the two- phase simplex method.

$$\text{Maximize } Z = 6X_1 + 3X_2 - 2X_1^2 - 3X_2^2 - 4X_1X_2$$

subject to

$$X_1 + X_2 \leq 1$$
$$2X_1 + 3X_2 \leq 4$$
$$X_1 \text{ and } X_2 \geq 0$$

Solution The given problem is written as follows:

$$\begin{aligned}\text{Maximize } Z &= c_1X_1 + c_2X_2 + c_3X_1^2 + c_4X_2^2 + c_5X_1X_2 \\ &= 6X_1 + 3X_2 - 2X_1^2 - 3X_2^2 - 4X_1X_2\end{aligned}$$

subject to

$$X_1 + X_2 \leq 1$$
$$2X_1 + 3X_2 \leq 4$$
$$-X_1 \leq 0$$
$$-X_2 \leq 0$$

All the matrices that are required to generate the system of constraints except matrix D can be obtained from the given problem easily. The guideline for obtaining the Matrix D is as follows:

$$D = \begin{bmatrix} c_3 & \frac{c_5}{2} \\ \frac{c_5}{2} & c_4 \end{bmatrix} = \begin{bmatrix} -2 & -2 \\ -2 & -3 \end{bmatrix}$$

The model of the given problem is:

$$\text{Maximize } Z = [6 \quad 3] \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + [X_1 \quad X_2] \begin{bmatrix} -2 & -2 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

subject to

$$\begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \quad \text{where } -X_1 \leq 0, \quad -X_2 \leq 0$$

Let $\emptyset_i$ and $\emptyset_2$ are the Lagrangean multiples associated with Constraint 1 and Constraint 2 of the system of constraints, respectively. μ_1 and μ_2 are the Lagrangean multiplier associated with constraints 3 and 4 of the system of constraints, respectively. Application of ?Khun-Tucker conditions to this model resulting into the following system of constraints.

$$\left[\begin{array}{cc|cc|cc|cc} 4 & 4 & 1 & 2 & -1 & 0 & 0 & 0 \\ 4 & 6 & 1 & 3 & 0 & -1 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & 3 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} X_1 \\ X_2 \\ \phi_1 \\ \phi_2 \\ \mu_1 \\ \mu_2 \\ S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \\ 1 \\ 4 \end{bmatrix}$$

$$\mu_1 X_1 = \mu_2 X_2 = 0$$
$$\phi_1 S_1 = \phi_2 S_2 = 0$$

The above system of equations is written as follows:

$$\begin{aligned} 4X_1 + 4X_2 + \phi_1 + 2\phi_2 - \mu_1 &= 6 \\ 4X_1 + 6X_2 + \phi_1 + 3\phi_2 - \mu_2 &= 3 \\ X_1 + X_2 + S_1 &= 1 \\ 2X_1 + 3X_2 + S_2 &= 4 \end{aligned}$$

$$\mu_1 X_1 = \mu_2 X_2 = 0$$
$$\phi_1 S_1 = \phi_2 S_2 = 0$$

The model for phase I of the two- phase simplex method is presented below with a minimization objective function irrespective of the type of the objective function in the original problem

Minimize Z= R_1 + R_2

subject to

$$\begin{aligned} 4X_1 + 4X_2 + \phi_1 + 2\phi_2 - \mu_1 + R_1 &= 6 \\ 4X_1 + 6X_2 + \phi_1 + 3\phi_2 - \mu_2 + R_2 &= 3 \\ X_1 + X_2 + S_1 &= 1 \\ 2X_1 + 3X_2 + S_2 &= 4 \end{aligned}$$

$$\mu_1 X_1 = 0$$
$$\mu_2 X_2 = 0$$
$$\phi_1 S_1 = 0$$
$$\phi_2 S_2 = 0$$
$$X_1, X_2, R_1, R_2, \phi_1, \phi_2, \mu_1, \mu_2, S_1 \text{ and } S_2 \geq 0$$

Note: The artificial variables R1, R2 are included in the constraints (17.44) and (17.45), respectively to have a basic variable in each of them. Table 1 represents the initial table of phase I of the two phase simplex method.

Table 1

CB_i	C_j / Basic variable	X_1	X_2	ϕ_1	ϕ_2	μ_1	μ_2	R_1	R_2	S_1	S_2	Solution
	C_j	0	0	0	0	0	0	1	1	0	0	
1	R_1	4	4	1	2	−1	0	1	0	0	0	6
1	R_2	4	$\boxed{6}$	1	3	0	−1	0	1	0	0	3*
0	S_1	1	1	0	0	0	0	0	0	1	0	1
0	S_2	2	3	0	0	0	0	0	0	0	1	4
	Z_j	8	10	2	5	−1	−1	1	1	0	0	9
	$C_j - Z_j$	−8	−10*	−2	−5	1	1	0	0	0	0	

In table 1 the non-basic variables X_2 has the maximum negative criterion value. Since, μ_2 is not in the basis, X_2 is the entering variable. [In this iteration, both μ_2 and X2 should not be present in the basis as per the constraint (17.49)]. The corresponding leaving variable is R_2. As per this combination of entering variable and leaving variable, pivot operations are shown in table 2

Table 2

CB_i	C_j / Basic variable	X_1	X_2	ϕ_1	ϕ_2	μ_1	μ_2	R_1	R_2	S_1	S_2	Solution
	C_j	0	0	0	0	0	0	1	1	0	0	
1	R_1	4/3	0	1/3	0	−1	2/3	1	−2/3	0	0	4
0	X_2	$\boxed{2/3}$	1	1/6	1/2	0	−1/6	0	1/6	0	0	1/2*
0	S_1	1/3	0	−1/6	−1/2	0	1/6	0	−1/6	1	0	1/2
0	S_2	0	0	−1/2	−3/2	0	1/2	0	−1/2	0	1	5/2
	Z_j	4/3	0	1/3	0	−1	2/3	1	−2/3	0	0	4
	$C_j - Z_j$	−4/3*	0	−1/3	0	1	−2/3	0	5/3	0	0	

In the table 2, the non-basic variable X_1 has the maximum negative criterion value. Since μ_1 is not the basis, X_1 is the entering variable. [in this iteration, as per the constraint (17.48), both μ_1 and X1 should not be present in the basis. The corresponding leaving variable is X2. As per this combination of entering variable and leaving variable pivot operations are shown in Table 3.

Table 3.

CB_i	C_j	0	0	0	0	0	0	1	1	0	0	Solution
	Basic variable	X_1	X_2	ϕ_1	ϕ_2	μ_1	μ_2	R_1	R_2	S_1	S_2	
1	R_1	0	−2	0	−1	−1	1	1	−1	0	0	3
0	X_1	1	3/2	1/4	3/4	0	−1/4	0	1/4	0	0	3/4
0	S_1	0	−1/2	−1/4	−3/4	0	$\boxed{1/4}$	0	−1/4	1	0	1/4*
0	S_2	0	0	−1/2	−3/2	0	1/2	0	−1/2	0	1	5/2
	Z_j	0	−2	0	−1	−1	1	1	−1	0	0	3
	$C_j - Z_j$	0	2	0	1	1	−1*	0	2	0	0	

In table 4, the non -basic, variable μ_2 has the maximum negative criterion value. In this iteration, as per constant (17.49), both X_2 and μ_2 should not be present in the basis. Since X_2 is not in basis, μ_2 is the entering variable. The corresponding leaving variable is S_1. As per this combination of entering variable and leaving variable, pivot operation are shown in table 5.

Table 5.

CB_i	C_j	0	0	0	0	0	0	1	1	0	0	Solution
	Basic variable	X_1	X_2	ϕ_1	ϕ_2	μ_1	μ_2	R_1	R_2	S_1	S_2	
1	R_1	0	0	$\boxed{1}$	2	−1	0	1	0	−4	0	2*
0	X_1	1	1	0	0	0	0	0	0	1	0	1
0	μ_2	0	−2	−1	−3	0	1	0	−1	4	0	1
0	S_2	0	1	0	0	0	0	0	0	−2	1	2
	Z_j	0	0	1	2	−1	0	1	0	−4	0	2
	$C_j - Z_j$	0	0	−1*	−2	1	0	0	1	4	0	

In table 5, the non-basic variable $\emptyset_2$ has the maximum negative criterion value. Since S_2 is in the basis $\emptyset_2$ cannot enter the basis. So, the non-basis variable $\emptyset_1$ which has the nest highest negative criterion row values is considered as the entering variable. Since S_1 is not in the basis, $\emptyset_1$ is the entering variable. The corresponding leaving variable is R_1. As per this combination of entering variable and leaving, pivot operation are shown in Table 6.

CB_i	C_j / Basic variable	0 / X_1	0 / X_2	0 / ϕ_1	0 / ϕ_2	0 / μ_1	0 / μ_2	1 / R_1	1 / R_2	0 / S_1	0 / S_2	Solution
0	ϕ_1	0	0	1	2	−1	0	1	0	−4	0	2
0	X_1	1	1	0	0	0	0	0	0	1	0	1
0	μ_2	0	−2	0	−1	−1	1	1	−1	0	0	3
0	S_2	0	1	0	0	0	0	0	0	−2	1	2
	Z_j	0	0	0	0	0	0	0	0	0	0	0
	$C_j - Z_j$	0	0	0	0	0	0	1	1	0	0	

Since the values in the criterion row of Table 17.16 are all nonnegative, the optimality is reached. The optimal solution of the problem is: $X_1^* = 1$, $X_2^* = 0$ and the corresponding maximum value of the objective function is 4.