Here, any one of the three messages m_2, m_3, m_1, or m_1 m_4 could be represented by the same binary sequence 0 1 0. Hence, the sequence 0 1 0 can not be decoded accurately. Such type of ambiguities should be removed.

Thus, the unique decipherability may be defined as follows :

Definition. *A code is said to be uniquely decipherable (separable) if every finite sequence of code characters correspond to at most one message.*

For example,

m_1	0
m_2	1 0
m_3	1 1 0
m_4	1 1 1

Here, encoding procedure establishes a one-to-one correspondence between messages and their code words, without the necessity of having any space between successive messages.

If it is written as

$$0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,1\,1\,0\,1\,0\,0\,0\,1\,1\,0\,1\,0\,0\,1\,0\,0\,1\,0\,0,$$

then this message will be uniquely decoded into

$$m_1\; m_1\; m_1\; m_1\; m_1\; m_2\; m_1\; m_2\; m_1\; m_1\; m_3\; m_2\; m_1\; m_1\; m_3\; m_2\; m_1\; m_2\; m_1\; m_2\; m_1$$

26.18. SHANNON-FANO ENCODING PROCEDURE

This method of encoding is directed towards constructing reasonably efficient separable binary codes for sources without memory. Let $[X]$ be the ensemble of the message to be transmitted and $[P]$ be their corresponding probabilities, *i.e.*

$$[X] = [x_1, x_2, \dots, x_n]$$
$$[P] = [p_1, p_2, \dots, p_n]$$

Now, a sequence c_k of binary numbers of unspecified length n_k can be associated to each message x_k such that—

(i) No sequence of employed binary numbers c_k can be obtained from each other by adding more binary terms to shorter sequence.

(ii) The transmission of the encoded message is 'reasonably' efficient, *i.e.* 1 and 0 appear independently and with (almost) equal probabilities.

The Shannon-Fano encoding procedure can be explained by solving the following examples.

Illustrative Examples

Example 12. *Apply Shannon's encoding procedure to the following message ensemble :*

$$[X] = [m_1, m_2, m_3, m_4]$$
$$[P] = [0.4, 0.3, 0.2, 0.1]$$

Solution.

Message	*Probability*	*Encoded message*	*Length*
m_1	0.4	0	1
—	—	—	—
m_2	0.3	1 0	2
m_3	0.2	1 1 0	3
m_4	0.1	1 1 1	3
			Average length 1.9

Step 1. Messages are first written in order of non-increasing probabilities. Then the set is partitioned into two most equiprobable subsets $\{S_1\}$ and $\{S_2\}$. Zero is assigned to each message in one subset and 1 to each of the remaining messages.

Step. 2. The same procedure should be repeated for subsets of $\{S_1\}$ and $\{S_2\}$. In this example, the subset $\{S_1\} = \{m_1\}$ cannot be partitioned further. But, the subset $S_2 = \{m_2, m_3, m_4\}$ can be partitioned as $S_{21} = \{m_2\}$ and $S_{22} = \{m_3, m_4\}$. So, assign 0 to message m_2 and 1 to each of the messages m_3 and m_4.

Step 3. The procedure is continued till each subset contains only one message.

The entropy of the source is

$$H = -[0.4 \log 0.4 + 0.3 \log 0.3 + 0.2 \log 0.2 + 0.1 \log 0.1] = 1.9 \text{ bits/message}$$

The expected length is

$$\bar{L} = \Sigma\, p\{m_i\}\, n_i = 0.4 \times 1 + 0.3 \times 2 + 0.2 \times 3 + 0.1 \times 3 \qquad = 1.9. \text{ bits/symbol.}$$

Example 13. *A source without memory has six characters with following probabilities :*

A	B	C	D	E	F
1/3	1/4	1/8	1/8	1/12	1/12

Devise an encoding procedure with the prefix property giving minimum possible average length for the transmission over a binary noiseless channel.

What is the average length of the encoded message ?

Solution. Proceeding as in *Example* 12, we obtain the solution as follows :

Characters	Prob. (p)	Code	Code length (l)	$p \times l$
A	S_1 [1/3, 1/4]	00	2	2/3
B	S_2 [1/8, 1/8 (S_{21}); 1/12, 1/12 (S_{22})]	10	2	1/2
C		100	3	3/8
D		101	3	3/8
E		110	3	1/4
F		111	3	1/4

Average length $(\bar{L}) = \Sigma p \times l = 2/3 + 1/2 + 3/8 + 3/8 + 1/4 = 29/12$ bits/symbol. **Ans.**

26.19. A NOISELESS CODING THEOREM

Theorem 26.7. *Necessary and sufficient condition for the existence of an irreducible noiseless encoding procedure with specified word length* $[n_1, n_2, \ldots, n_N]$ *is a set of positive integers* $[n_1, n_2, \ldots, n_N]$ *that can be found such that*

$$\sum_{i=1}^{N} D^{-n_i} \le 1 \qquad \text{...(26.63)}$$

where D is the number of symbols in encoding alphabet.

Proof. ***Condition is Necessary*** :

Obviously, two encoded messages x_i and x_k can have the same length, *i.e.* $n_i = n_k$

Let W_i be the number of encoded messages of length n_i. But, number of encoded messages with only one letter cannot be larger than D. Therefore,

$$W_1 \le D. \qquad \text{...(26.64)}$$

Also, number of encoded messages of length 2, because of coding restriction, cannot be larger than $(D - W_1)\, D$. Hence

$$W_2 \le (D - W_1)\, D = D^2 - W_1 D \qquad \text{...(26.65)}$$

Likewise,

$$W_3 \le [(D - W_1)\, D - W_2]\, D = D^3 - W_1 D^2 - W_2 D \qquad \text{...(26.66)}$$

Finally, if m is the maximum length of encoded words, it is concluded that

$$W_m \le D^m - W_1 D^{m-1} - W_2 D^{m-2} - \ldots - W_{m-1} D \qquad \text{...(26.67)}$$

Now, dividing both sides of this inequality by D^m,

$$W_m D^{-m} \le 1 - W_1 D^{-1} - W_2 D^{-2} - \ldots - W_{m-1} D^{-(m-1)}$$

or

$$W_1 D^{-1} + W_2 D^{-2} + \ldots + W_{m-1} D^{-(m-1)} + W_m D^{-m} \le 1$$

or

$$\sum_{i=1}^{m} W_i D^{-i} \le 1 \qquad \text{...(26.68)}$$

where m is the maximum length of any message.

Now, this inequality can be written as

$$W_1 D^{-1} + W_2 D^{-2} + \ldots + W_m D^{-m} \le 1 \qquad \ldots(26.69)$$

or

$$\left[\frac{1}{D}+\frac{1}{D}+\ldots + W_1 \text{ times}\right]+\left[\frac{1}{D^2}+\frac{1}{D^2}+\ldots + W_2 \text{ times}\right]+\ldots\ldots+\left[\frac{1}{D^m}+\frac{1}{D_m}+\ldots\ W_m \text{ times}\right]$$

Each term in the bracket of eqn (26.69) corresponds to a specified message length such as in the first bracket, W_1 message is of length 1, in second bracket W_2 message is of length 2, and so on.

Hence the total number of messages are

$$W_1 + W_2 + \ldots + W_m = N \qquad \ldots(26.70)$$

Terms in W_k correspond to encoded messages of length k.

Consider later terms as ΣD^{-n_i} when the summation takes place over all those terms with $n_i = k$. Hence by a simple re-arrangement of terms, it can be equivalently written as

$$\sum_{j=1}^{m} W_j D^{-j} = \sum_{i=1}^{N} D^{-n_i} \qquad \ldots(26.71)$$

Therefore,

$$\sum_{j=1}^{m} W_j D^{-j} = \sum_{i=1}^{N} D^{-n_i} \le 1 \qquad \ldots(26.72)$$

The desired set of positive integers $[n_1, n_2, \ldots, n_N]$ satisfy the inequality (26.63).

***Condition is sufficient*:**

We have to show that the condition

$$\sum_{i=1}^{m} W_i D^{-i} = W_1 D^{-1} + W_2 D^{-2} + \ldots + W_m D^{-m} \le 1 \qquad \ldots(26.73)$$

is sufficient for the existence of desired codes.

Since the terms $W_1 D^{-1}, W_2 D^{-2}, \ldots, W_m D^{-m}$ are all positive, each term must be less than 1. Thus, it can be concluded that

$$W_1 D^{-1} \le 1 \text{ or } W_1 \le D, \qquad \ldots(26.74)$$

and

$$W_1 D^{-1} + W_2 D^{-2} \le 1 \text{ or } W_2 \le D(D - W_1) \qquad \ldots(26.75)$$

and so on. Since these are the conditions we have to satisfy in order to guarantee that no encoded message can be obtained from any other source by the addition of a sequence of letters of the encoding alphabet.

As an application of this theorem, let D be a binary set, *i.e.* $A = [a_1, a_2]$, then the encoding theorem requires that

$$\sum_{i=1}^{N} 2^{n_i} \le 1 \qquad \ldots(26.76)$$

As an application of the foregoing, consider the existence of a separable code book having N words of equal length n. The noiseless coding theorem suggests that such codes exist if

$$\sum_{i=1}^{N} D^{-n_i} \le 1, \text{ where } n_1 = n_2 = \ldots = n_N = n. \qquad \ldots(26.77)$$

or

$$D^{-n} + D^{-n} + \ldots + N \text{ times} \le 1 \quad \text{or} \quad ND^{-n} \le 1$$

or

$$\log N + (-n)\log D \le 0 \quad \text{or} \quad \log N \le n \log D. \qquad \ldots(26.78)$$

This relation between N, n and D guarantees the existence of desired codes.

This completes the proof of the theorem.

Illustrative Examples

Example 14. *There are 12 coins, all of equal weight except one which may be lighter or heavier. Using concepts of information theory, show that it is possible to determine which coin is the odd and indicate whether it is lighter or heavier.*

Solution. The principle '*maximum information is received when the events are equally likely*' can be used here to seek the information about the odd coin.

Take an ordinary weighing balance. Assuming complete ignorance about the identity of the odd coin, and whether it is lighter or heavier, one has to identify 24 equally likely possibilities of placing a coin on either pan. Obviously, this will require $\log_2 24$ bits of information.

At each weighing, try to generate the maximum possible amount of information.

For one weighing, let

p_L = probability that balance tips to *left*

p_R = probability that balance tips to *right*

p = probability that balance does not tip to any side

Thus, the information generated in this weighing is given by

$$H = -p_L \log p_L - p_R \log p_R - p \log p$$

H will be maximum if probabilities are equal. From this, one can conclude that weighing should be done in such a way that tipping to *left, balancing,* and tipping to *right* are equally probable events.

Put n coins in each of the left and right pans, and $12 - 2n$ are weighted. Therefore,

$$p = (12 - 2n)/12, p_L = p_R = m/12,$$

and hence $p = p_L = p_R = 1/3$, when $n = 4$. Thus, it is possible to divide 12 coins into 3 groups, say G_1, G_2, G_3 consisting of four coins each. Then, place two of them, say G_1 and G_2, in different pans of the balance. Now, two cases will arise:

Case I. Pans balance each other.

Case II. If pans do not balance, observe which one is heavier.

In the first case, odd coins lie in the third group (G_3). In the second case, remove one of the groups, say G_1, from the pans and set G_2 in place of G_1. If pans now balance each other, odd coins lie in the first group (G_1), otherwise in the second group (G_2). Also, observations noted in this case and in the second case earlier, reveal whether the odd coin is heavier or lighter.

Up to this stage, it is decided which one of the groups G_1, G_2, G_3 contains the odd coin. Denote this group by $G = \{c_1, c_2, c_3, c_4\}$ where the letter 'c' denotes the coin.

Now, put two coins, say c_1 and c_2 of group G in different pans of the balance. Again, two cases will arise:

Case I′. If pans balance each other, then the odd coin is either c_3 or c_4.

Case II′ If pans do not balance, then the odd coin is either c_1 or c_2.

In either case, replace one of the coins, say c_1 by c_3. If pans now balance, the Case I′ will decide that c_4 is the odd coin whereas Case II′ will decide that c_1 is the odd coin. On the other hand if pans do not balance each other, then the Case I′ decides that c_3 is the odd coin whereas Case II′ establishes that c_2 is the odd coin.

Finally, the coin being lighter or heavier is an immediate consequence observed after weighing.

Illustrative Example

Example 15. *Suppose we are given n coins which look quite alike, but of which some are false. The false coins have smaller weight than the genuine ones. The weights* α *and* β ($\beta < \alpha$) *of both the genuine and false coins are known. A scale is given by means of which any number less than* ($< n$) *of coins can be weighted together. Thus, we select an arbitrary subset of the coins and put them together on the scale; then the scale shows us the total weight of these coins. Find the lower bound of the minimal number* $a(n)$ *of weighings by means of which the genuine and false coins can be separated.*

Solution. Since the subset of the coins consisting of the false coin may be any of the 2^n subsets of the set of all coins, the amount of information needed is $\log_2 (2^n) = n$.

On the other hand, if we put $k \le n$ coins on the balance, the number of false coins among them may have the values 0, 1, ..., k and thus the amount of information given by each weighing cannot exceed, *i.e.*

$$\log_2 (k + 1) \le \log_2 (n + 1).$$

Hence, r weighings can give us at most $r \log_2 (n + 1)$ bits, and thus to obtain the necessary amount of information is (n bits) it is necessary that $r \log_2 (n + 1) < n$. That is,

$$a(n) \ge n/\log_2 (n + 1).$$

SELF-EXAMINATION QUESTIONS

1. Write a critical essay on information theory emphasizing the basic concepts ?
2. Define entropy function and establish its formal requirements.
3. (a) Define the different entropies for a two part communication system and calculate them for a discrete channel with independent input-output.

   (b) Give a measure for mutual information $I(x, y)$ and show that
   $$I(x, y) = H(x) + H(y) - H(x, y)$$
4. Show that the entropy function is maximum when mutually exclusive events are equiprobable. Show also that the partitioning of events into sub-events cannot decrease the entorpy of the system. **[Delhi (OR). 92]**
5. Give a brief account of memoryless schemes.
6. Show that all possible sets of binary codes with the prefix property for encoding the message ensemble (m_1, m_2, m_3) in words not more than three digits long.
7. Let S be the discrete sources without memory with a communication entropy $H(x)$ and a noiseless channel with capacity C bits per message. Show that it is possible to encode the output S so that, if hte encoded messages are transmitted through the channel, the rate of transmission of information approaches C per symbol as closely as desirable.
   $$\text{with } \sum_{i=1}^{n} p_i = \sum_{i=1}^{n} q_i .$$
8. An alphabet consists of four letters A, B, C, D with respective probabilities of transmission 1/3, 1/4, 1/5, 1/6. Find the average amount of inforamtion associated with the transmission of a letter.

EXAMINATION PROBLEMS

1. Evaluate the entropy associated with the following probability distribution :

Event :	A	B	C	D
Probability :	1/2	1/4	1/8	1/8

2. Let X be a discrete random variable taking vlaues $x_1, x_2, \dots, x_n$ with probability $P(X = x_k) = p_k$, $k = 1, 2, \dots, n$; $p_k \geq 0$, $\Sigma p_k = 1$. Define the entropy $H(p_1, p_2, \dots, p_n)$ of the probability distribution to X and prove that
   $$H(p_1, p_2, \dots, p_n) = H(p_1, p_2, \dots, p_{n-2}, p_{n-1}, p_n) + (p_{n-1} + p_n)\, H\left(\frac{p_{n-1}}{p_n + p_{n-1}}, \frac{p_n}{p_n + p_{n-1}}\right)$$
3. If H denotes the entropy function, then prove that
   $$H(p_1, p_2, \dots, p_N, q_1, q_2, \dots, q_m) = H(p_1, p_2, \dots, p_n) + p_n\, H\left(\frac{q_1}{p_n}, \frac{q_2}{p_n}, \dots, \frac{q_m}{p_n}\right)$$
   where $p_n = \sum_{k=1}^{m} q_k$. Verify the formula, defining additivity of entropies, for events A, B, C with probabilities 1/5, 4/15, 8/15 respectively.
4. The two finite probability schemes are given by $(p_1, p_1, \dots, p_n)$ and $(q_1, q_2, \dots, q_n)$. Show that
   $$-\sum_{k=1}^{n} p_k \log q_k \leq -\sum_{k=1}^{n} p_k \log p_k$$
   with equelity if and only it $p_i = q_i$, for all i

   [**Hint.** Since $\log x \leq x - 1$ if $x = 1$, therefore $\log (q_i/p_i) \geq (q_i/p_i) - 1$ if $p_i = q_i$, $x_i = q_i/p_i$.

   Thus, $\sum_{i=1}^{n} p_i \log (q_i/p_i) \leq \sum_{i=1}^{n} p_i (q_i/p_i - 1) = 0$ for $p_i = q_i$, *i.e.* $\sum_{i=1}^{n} p_i \log q_i \leq \sum_{i=1}^{n} p_i \log p_i$

   or $$-\sum_{i=1}^{n} p_i \log q_i \geq -\sum_{i=1}^{n} p_i \log p_i \,]$$
5. Evaluate the average uncertainty associated with the sample space of disjoint events A, B, C, where
   $$P(A) = 1/5,\ P(B) = 4/15,\ P(C) = 8/15.$$
6. Apply *Shannon-Fano* encoding procedure to the following set of messages :

   [m_1 m_2 m_3 m_4 m_5 m_6 m_7 m_8]

   [1/4 1/4 1/8 1/8 1/16 1/16 1/16 1/16]

   Also, determine the entropy (H) of the original source and average length $(\bar{L})$ of the encoded meassage.

   [Ans.

m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
00	01	100	101	1100	1101	1110	111

   $H = 2.75$ bits, $\bar{L} = 2.75$]
7. Apply Shannon-Fano encoding procedure to the following message :

[X] = x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
[P] = 0.49	0.14	0.14	0.07	0.07	0.04	0.02	0.02	0.01

[**Ans.**

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
0	100	101	1100	1101	1101	11110	111111

$\bar{L} = 3$ bits/symbol, $H = 1.60 + 2 \log 5 - (1.40) \log 7$]

8. Write a short note on entropy. Show that the entropy of the following events is $2 - (1/2)^{n-2}$

Event (x_i) :	x_1	x_2	...	x_i	...	x_{n-1}	x_n
Prob. $p(x_i)$:	1/2	1/4	...	$1/2^i$	...	$1/2^{n-1}$	$1/2^n$

[**Hint.** Here we have

$$p_i = \frac{1}{2^i},\ i = 1, 2, \dots n-1; \text{ and } p_n = \frac{1}{2^{n-1}} \text{ such that } \sum_{i=1}^{n} p_i = 1.$$

The entropy function H is defined as

$$H(p_i, p_2, \dots, p_n) = -\sum_{i=1}^{n} p_i \log p_i = -\sum_{i=1}^{n-1} p_i \log p_i - p_n \log p_n$$

$$= -\sum_{i=1}^{n-1} \left(\frac{1}{2^i}\right) \log \left(\frac{1}{2^i}\right) - \left(\frac{1}{2^{n-1}}\right) \log \left(\frac{1}{2^{n-1}}\right)$$

$$= -\sum_{i=1}^{n-1} \left(\frac{1}{2^i}\right) \log \left(\frac{1}{2^i}\right) + \left(\frac{1}{2^{n-1}}\right) \log_2 (2^{n-1}) = \sum_{i=1}^{n-1} i \left(\frac{1}{2^i}\right) + (n-1) \left(\frac{1}{2^{n-1}}\right) \quad [\because \log_2 2 = 1]$$

$$= \left\{\frac{1}{2} + \frac{2}{2^2} + \frac{3}{2^3} + \dots + \frac{n-1}{2^{n-1}}\right\} + \frac{n-1}{2^{n-1}} \quad \dots \text{(i)}$$

or $$\frac{1}{2} H(p_1, p_2, \dots, p_n) = \left\{\frac{1}{2^2} + \frac{2}{2^3} + \dots + \frac{n-1}{2^n}\right\} + \frac{n-1}{2^n} \quad \dots \text{(ii)}$$

Subtracting (ii) from (i), we get

$$H(p_1, p_2, \dots p_n) - \frac{1}{2} H(p_1\ p_2, \dots, p_n) = \left(\frac{1}{2} + \frac{1}{2^2} + \dots + \frac{1}{2^{n-1}}\right) + \left(\frac{n-1}{2^{n-1}} - \frac{2(n-1)}{2^n}\right)$$

or $$\frac{1}{2} H(p_1, p_2, \dots, p_n) = \left(\frac{1}{2} + \frac{1}{2^2} + \dots + \frac{1}{2^{n-1}}\right) = 1 - \left(\frac{1}{2}\right)^{n-1}$$

or $$H(p_1, p_2, \dots, p_n) = 2 - \left(\frac{1}{2}\right)^{n-2}$$

9. Prove that $H(p_1, p_2, \dots, p_n) \leq \log_2 n$, and equality holds if and only if $p_k = 1/n$, $k = 1, 2, 3, \dots, n$.

UNIT 5

NONLINEAR AND DYNAMIC PROGRAMMING

CONTAINING :

27

CLASSICAL OPTIMIZATION TECHNIQUES
(Lagrangian Method & Kuhn-Tucker Conditions)

27.1. INTRODUCTION

In this chapter, we shall concern ourselves with the classical theory of optimization. This theory deals with the use of differential calculus to determine the points of *maxima* and *minima* for both *unconstrained* and *constrained* continuous functions. Although, in general, the classical optimization techniques are not suitable for obtaining numerical solutions except for relatively simple problems, the underlying theory gives the basis for devising most of the *non-linear programming* algorithms.

We have introduced such topics in this chapter which include : the development of *necessary* and *sufficient conditions* for locating the extreme points for *unconstrained* problems, the treatment of the constrained problems using the *Lagrangian* methods, and the development of the *Kuhn-Tucker conditions* for the general problem with inequality constraints.

27.2. UNCONSTRAINED PROBLEMS OF MAXIMA AND MINIMA

We shall discuss the problem of determining the extreme points (the points of maxima and minima) of an unconstrained type of continuous function.

Mathematically, a function $f(x)$ has a maximum at a point x_0 if for $|h|$ sufficiently small

$$f(x_0 + h) - f(x_0) < 0.$$

Similarly, a function $f(x)$ has a minimum at a point x_0 if

$$f(x_0 + h) - f(x_0) > 0.$$

For example, *Fig.* 27.1. illustrates a continuous function $f(x)$ defined on the interval (a, b). The points x_1, x_2, x_3, x_4 and x_6 (*not* x_5) represent all the points of maxima and minima (called the stationary or critical points) of $f(x)$. These include x_1, x_3 and x_6 as the points of maxima, and x_2 and x_4 as the points of minima.

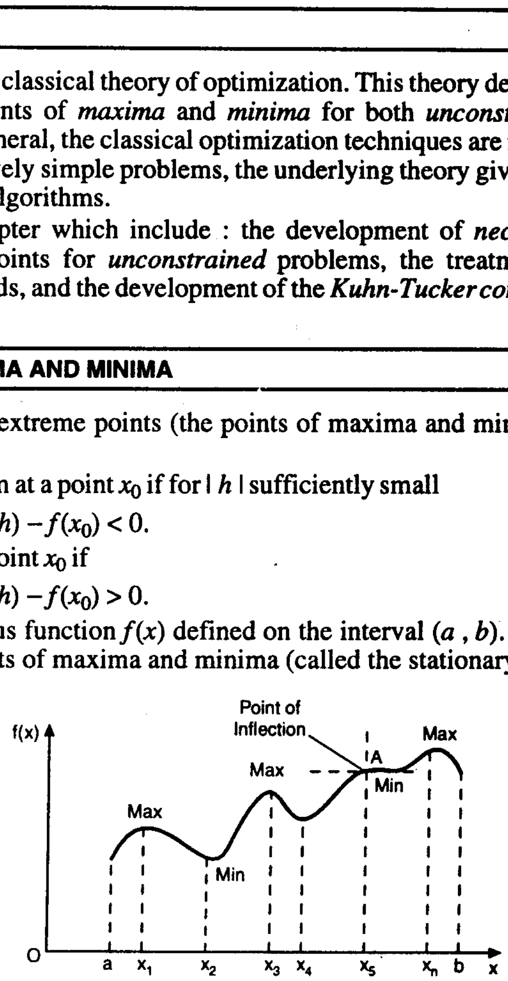

Fig. 27.1

Global (absolute) maximum :

Since $f(x_6) = \max. \{f(x_1), f(x_3), f(x_6)\}$, $f(x_6)$ is called a *global* or *absolute* maximum.

Local (relative) maxima :

On the other hand, $f(x_1)$ and $f(x_3)$ are called *local* or *relative* maxima.

Similarly, $f(x_4)$ is a local minimum while $f(x_2)$ is a global minimum.

Further, it should be noted that the point A corresponding to $f(x_5)$ is called the *point of inflection*.

Now, in the following section, it will be shown how extreme points can be determined for the general case of an n-variable function $f(\mathbf{x})$, $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$. *Theorem 27.1* gives the necessary conditions for the existence of an extreme point and *Theorem 27.2* proves the sufficiency conditions. We shall assume throughout this section that both the first and the second partial derivatives of $f(\mathbf{x})$ are continuous. The proof of the following theorems will be accomplished through the use of *Taylor's* theorem.

27.2-1. Some Important Theorems

Theorem 27.1. *A necessary condition for a continuous function $f(\mathbf{x})$ with continuous first and second partial derivatives to have an extreme point at $\mathbf{x_0}$ is that each first partial derivative of $f(\mathbf{x})$, evaluated at $\mathbf{x_0}$,* vanish, that is

$$\nabla f(\mathbf{x_0}) = 0$$

where $\nabla \equiv \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)$ *is the gradient vector.*

Proof. By *Taylor's* theorem, for $0 < \theta < 1$,

$$f(\mathbf{x_0} + \mathbf{h}) - f(\mathbf{x_0}) = \nabla f(\mathbf{x_0})\,\mathbf{h} + \tfrac{1}{2}\,\mathbf{h}' H \mathbf{h} \,\big|_{\mathbf{x_0} + \theta\mathbf{h}} \qquad \ldots(27.1)$$

where $\mathbf{h} = (h_1, h_2, \ldots, h_j, \ldots, h_n)'$ and $|\,h_j\,|$ is small enough for all $j = 1, 2, \ldots, n$.

For small $|\,h_j\,|$, the remainder term $\tfrac{1}{2}\,(\mathbf{h}' H \mathbf{h})$ is of order h_j^2 and hence it will tend to zero as $h_j \to 0$. Thus,

$$f(\mathbf{x_0} + \mathbf{h}) - f(\mathbf{x_0}) = \nabla f(\mathbf{x_0})\,\mathbf{h} + O\,(h_j^2) \qquad \ldots(27.2)$$

$$\cong \nabla f(\mathbf{x_0})\,\mathbf{h} \quad \cong \left[h_1 \frac{\partial f(\mathbf{x})}{\partial x_1} + h_2 \frac{\partial f(\mathbf{x})}{\partial x_2} + \ldots + h_p \frac{\partial f(\mathbf{x})}{\partial x_p} + \ldots + h_n \frac{\partial f(\mathbf{x})}{\partial x_n} \right]_{\mathbf{x} = \mathbf{x_0}}$$

Suppose that $\mathbf{x_0}$ is an extreme point. Now we shall prove the theorem by contradiction.

If possible, let us suppose that one of the partial derivatives, say pth, does not vanish, *i.e.* $\partial f(\mathbf{x_0})/\partial x_p \neq 0$. Then (27.2) becomes

$$f(\mathbf{x_0} + \mathbf{h}) - f(\mathbf{x_0}) = h_p \frac{\partial f(\mathbf{x_0})}{\partial x_p} \qquad \ldots(27.3)$$

Since $\frac{\partial f(\mathbf{x_0})}{\partial x_p} \neq 0$, either $\frac{\partial f(\mathbf{x_0})}{\partial x_p} < 0$ or $\frac{\partial f(\mathbf{x_0})}{\partial x_p} > 0$.

Now, suppose $\frac{\partial f(\mathbf{x_0})}{\partial x_p} > 0$. Then the *L.H.S.* of (27.3) will have the same sign as h_p, that is,

(i) $f(\mathbf{x_0} + \mathbf{h}) - f(\mathbf{x_0}) > 0$ when $h_p > 0$, and *(ii)* $f(\mathbf{x_0} + \mathbf{h}) - f(\mathbf{x_0}) < 0$ when $h_p < 0$.

This contradicts the assumption that $\mathbf{x_0}$ is an extreme point. The argument when $[\partial f(\mathbf{x_0})/\partial x_p] < 0$ is similar to the given above. Thus, we may conclude that when any of the partial derivatives are not identically equal to zero at $\mathbf{x_0}$, the point $\mathbf{x_0}$ is not an extreme point. Thus, it follows that for $\mathbf{x_0}$ to be an extreme point, it is necessary that

$$\nabla f(\mathbf{x_0}) = 0. \qquad \ldots(27.4)$$

This completes the proof of the theorem.

The condition (27.4) says that the partial derivatives of $f(\mathbf{x})$ with respect to x_p $(p = 1, 2, \ldots, n)$ must vanish at the extreme points $\mathbf{x_0}$.

Further, if we have the functions of one variable (say y) only, the above condition will reduce to

$$f'(y_0) = 0 \quad \text{or} \quad \left(\frac{\partial f}{\partial y} \right)_{y = y_0} = 0.$$

It is also important to note that the above conditions are also satisfied for the cases other than extreme point. These include, for example, *inflection* and *saddle* points. Consequently, the given conditions are necessary but not sufficient for determining the extreme points. Thus it is more reasonable to call the points obtained from the solution of

$$\nabla f(\mathbf{x}) = 0$$

as *stationary points.*

In the next theorem, we shall derive the sufficiency conditions for $\mathbf{x_0}$ to be an extreme point.

Theorem 27.2. *A sufficient condition for a stationary point* $\mathbf{x_0}$ *to be an extreme point is that the Hessian matrix H evaluated at* $\mathbf{x_0}$ *is,*

(i) negative-definite when $\mathbf{x_0}$ *is a maximum point, and*

(ii) positive-definite when $\mathbf{x_0}$ *is minimum point.*

Proof. By *Taylor's* theorem, for $0 < \theta < 1$, we have

$$f(\mathbf{x_0}+\mathbf{h}) - f(\mathbf{x_0}) = \nabla f(\mathbf{x_0})\,\mathbf{h} + \tfrac{1}{2}\,\mathbf{h}'H\mathbf{h}\,\big|_{\mathbf{x_0}+\theta\mathbf{h}}.$$

Since $\mathbf{x_0}$ is a stationary point, then by preceding theorem

$$\nabla f(\mathbf{x_0}) = 0.$$

Thus, $$f(\mathbf{x_0}+\mathbf{h}) - f(\mathbf{x_0}) = \tfrac{1}{2}\,\mathbf{h}'H\mathbf{h}\,\big|_{\mathbf{x_0}+\theta\mathbf{h}}$$

Let $\mathbf{x_0}$ be a maximum point, then by definition

$$f(\mathbf{x_0}+\mathbf{h}) < f(\mathbf{x_0})$$

for all non-null $\mathbf{h}$. This implies that for $\mathbf{x_0}$ to be a maximum.

$$\tfrac{1}{2}\,\mathbf{h}'H\mathbf{h}\,\big|_{\mathbf{x_0}+\theta\mathbf{h}} < 0 \quad\text{or}\quad \mathbf{h}'H\mathbf{h}\,\big|_{\mathbf{x_0}+\theta\mathbf{h}} < 0 \qquad ...(27.5)$$

Writing the quadratic form $\mathbf{h}'H\mathbf{h}$ in expanded form, we have

$$\sum_{i=1}^{n}\sum_{j=1}^{n} h_i h_j \frac{\partial^2 f(\mathbf{x})}{\partial x_i\,\partial x_j}\bigg|_{\mathbf{x}=\mathbf{x_0}+\theta\mathbf{h}} < 0.$$

However, since the second partial derivative $\dfrac{\partial^2 f(\mathbf{x})}{\partial x_i\,\partial x_j}$ is continuous in the neighbourhood of $\mathbf{x_0}$,

$$\frac{\partial^2 f(\mathbf{x})}{\partial x_i\,\partial x_j}\bigg|_{\mathbf{x}=\mathbf{x_0}} \text{ will have the same sign as } \frac{\partial^2 f(\mathbf{x})}{\partial x_i\,\partial x_j}\bigg|_{\mathbf{x}=\mathbf{x_0}+\theta\mathbf{h}}.$$

Consequently, $\mathbf{h}'H\mathbf{h}$ must yield the same sign when evaluated at both $\mathbf{x_0}$ and $\mathbf{x_0}+\theta\mathbf{h}$. Thus, from (27.5), we have

$$\mathbf{h}'H\mathbf{h}\,\big|_{\mathbf{x}=\mathbf{x_0}} < 0.$$

Since $\mathbf{h}'H\mathbf{h}\,\big|_{\mathbf{x}=\mathbf{x_0}}$ defines a quadratic form, this expression (and hence $\mathbf{h}'H\mathbf{h}\,\big|_{\mathbf{x}=\mathbf{x_0}+\theta\mathbf{h}}$) is negative if, and only if, the Hessian matrix H is negative-definite at $\mathbf{x_0}$. This completes the proof for maximization case.

A similar proof can be established for the minimization case to show that the corresponding Hessian matrix H is positive definite at $\mathbf{x_0}$.

27.2–2. Illustrative Example

Example 1. *Find the maximum or minimum of the function*

$$f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 - 4x_1 - 8x_2 - 12x_3 + 56.$$

Solution. Applying the necessary condition

$$\nabla f(\mathbf{x_0}) = 0 \quad\text{or}\quad \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3}\right) f(\mathbf{x}) = (0, 0, 0),$$

this gives

$$\frac{\partial f}{\partial x_1} = 2x_1 - 4 = 0,\quad \frac{\partial f}{\partial x_2} = 2x_2 - 8 = 0,\quad \frac{\partial f}{\partial x_3} = 2x_3 - 12 = 0.$$

The solution of these simultaneous equations is given by $\mathbf{x_0} = (2, 4, 6)$ which is the only point that satisfies the necessary conditions.

Now, by checking the sufficiency condition, we must determine whether this point is a maximum or minimum.

The *Hessian matrix*, evaluated at (2, 4, 6), is given by

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_2} & \frac{\partial^2 f}{\partial x_3^2} \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

The principal minor determinants of H :

$$|2|, \begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix}, \text{ and } \begin{vmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{vmatrix}$$

have the values 2, 4 and 8, respectively. Thus, each of the principal minor determinants is positive. Hence H is positive-definite. Therefore, the point (2, 4, 6) yields a minimum of $f(\mathbf{x})$.

Corollary. *If the Hessian of a function $f(\mathbf{x})$ is indefinite when evaluated at the point $\mathbf{x}_0$, where the necessary conditions are satisfied, then the point $\mathbf{x}_0$ is not an extreme point.*

The proof is easy. So it is left as an exercise for the readers.

27.2-3. How to Determine Sufficient Conditions when H is Semi-definite

First, we shall consider the case for single variable functions. The sufficiency condition established by *Therem* 27.2 reduces to the following cases. Given y_0 is a stationary point, then considering the *Hessian matrix* with one element,

(i) $f''(y_0) < 0$ is a sufficient condition for maximum. *(ii)* $f''(y_0) > 0$ is a sufficient condition for minimum.

It must be noted that in the single variable function, if $f''(y_0)$ vanishes, the higher order derivatives must be investigated, and then we reach the desired conclusion by applying the result of the following theorem :

Theorem 27.3. *Given a function $f(y)$, if at a stationary point y_0 the first $(n-1)$ derivatives vanishes and $f^{(n)}(y) \neq 0$, then at $y = y_0$, $f(y)$ has :*

(i) an inflection point if n is odd, and (ii) maximum if $f^{(n)}(y_0) < 0$ and a minimum if $f^{(n)}(y_0) > 0$.

The proof of this theorem is given in undergraduate calculus.

For example, we consider two functions (i) $f(y) = y^4$, (ii) $g(y) = y^3$.

For $f(y) = y^4$, we have $f'(y) = 4y^3 = 0$, which gives $y_0 = 0$ as stationary point.

Now $$f''(0) = f^{(3)}(0) = 0.$$

But $f^4(0) = 24 > 0$, hence $y_0 = 0$ is a minimum point [see ***Fig. 27.2 (i)***].

For $g(y) = y^3$, $g'(y) = 3y^2 = 0$.

This also gives $y_0 = 0$ as a stationary point. Since $g^{(3)}(0) = 6 \neq 0$, hence $y_0 = 0$ is an *inflection point.*

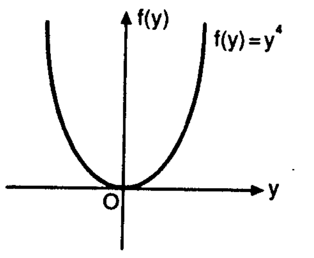

Fig. 27.2 *(i)*

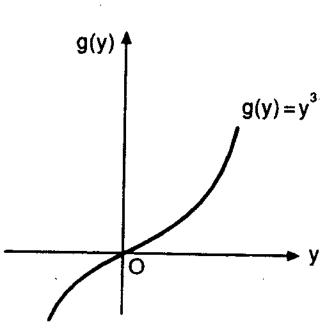

Fig. 27.2 *(ii)*

We now return to the case of functions of several variables. Many attempts have been made to develop sufficient conditions for extreme points in several variable case. One attempt was made by *Lagrange* as an extension of arguments of ***Theorem* 27.3** for one-dimensional case.

He argued that if the second order terms are semi-definite, the third order terms must vanish and fourth order terms would give the required information. If the fourth order terms are semi-definite, we must then investigate higher order terms. However, *Peano* developed a counter example to this argument. *Peano's* counter example is given below.

Example 2. *Consider* $f(\mathbf{x}) = (x_2^2 - x_1)^2 = x_1^2 - 2x_1x_2^2 + x_2^4$.

Let us apply *Lagrange's* argument. The necessary conditions are

$$\frac{\partial f}{\partial x_1} = 2\,(x_1 - x_2^2) = 0, \quad \frac{\partial f}{\partial x_2} = 4x_2^3 - 4x_1x_2 = 0.$$

The necessary conditions are satisfied along the curve $x_1 = x_2^2$. The second order terms are

$$\frac{\partial^2 f}{\partial x_1^2} = 2\,, \quad \frac{\partial^2 f}{\partial x_1\,\partial x_2} = -\,4x_2, \quad \frac{\partial^2 f}{\partial x_2^2} = 12x_2^2 - 4x_1 = 8x_2^2 \quad (\text{since } x_1 = x_2^2)$$

The *Hessian* ***H***, is given by

$$\begin{bmatrix} 2 & -4x_2 \\ -4x_2 & 8x_2^2 \end{bmatrix}$$

which can be easily seen to be positive semi-definite for any x_2. *Lagrange* argued that the third partial derivatives must all be zero since the *Hessian matrix*, *H*, is semi-definite. But, one of the third order terms

$$\frac{\partial^3 f}{\partial x_1\,\partial x_2^2} = -\,4.$$

Since this term does not vanish, *Lagrange* would argue that $x_1 = x_2^2$ is not a minimum. However, we can see that $f(x)$ cannot be negative, but is zero only when $x_1 = x_2^2$, and is positive for any other value of x_1 and x_2. We should note that the solution obtained is not a proper minimum.

When *Lagrange's* argument was shown to be erroneous, *Serret* modified the argument. *Serret* concluded that we should investigate higher-order terms only for those values of h_i for which the quadratic form $\mathbf{h}'\,H\,\mathbf{h}$ is zero. However, *Peano's* second counter example also proved this conclusion false.

Peano's Second Counter Example : Consider

$$f(\mathbf{x}) = (x_2^2 - x_1)\,(x_2^2 - 2x_1) = 2x_1^2 - 3x_1x_2^2 + x_2^4$$

$$\frac{\partial f}{\partial x_1} = -\,3x_2^2 + 4x_1 = 0, \quad \frac{\partial f}{\partial x_2} = 4x_2^3 - 6x_1x_2 = 0.$$

The point $(x_1 = 0\,, x_2 = 0)$ satisfies the necessary conditions shown above. The *Hessian* evaluated at $x_1 = 0\,, x_2 = 0$ is given by

$$H = \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix}$$

which is *negative semi-definite*.

Let us now expand $f(x_1\,, x_2)$ in a *Taylor's* series about $(x_1 = 0\,, x_2 = 0)$, we obtain

$$f(x_1 + h\,, x_2 + k) = \frac{1}{2\,!}\,h^2\,\frac{\partial^2 f(x_1\,, x_2)}{\partial x_1^2}\bigg|_{0,0} + hk\,\frac{\partial^2 f(x_1\,, x_2)}{\partial x_1\,\partial x_2}\bigg|_{0,0} + \frac{1}{2\,!}\,k^2\,\frac{\partial^2 f(x_1\,, x_2)}{\partial x_2^2}\bigg|_{0,0} + \ldots$$

The second order term, $2h^2$, is positive for all h and k except $h = 0$. Following *Serret's* argument, we look at the third order terms, but only for the value of h where the second order term becomes zero, *i.e.*, $h = 0$.

The third order terms are given by

$$\frac{1}{3\,!}\,(-\,18hk^2)$$

which becomes zero when $h = 0$. Now investigating the fourth order terms we obtain

$$f(x_1 + h, x_2 + k) - f(x_1, x_2) \mid_{0,0} = h^2 + 3hk^2 + k^4 + \dots$$

Applying *Serret's* argument, we see that when $h = 0$ (that is, when second order terms vanish), the form is *positive-definite* and consequently, the point (0, 0) is a minimum. However, we note that for $x_1 < x_2^2 < 2x_1$ the function is negative, and for $x_2^2 < x_1$ or $x_2^2 > 2x_1$ the function is positive, and consequently the point (0, 0) is a saddle point, since it decreases for all x_2 in the range $x_1 < x_2^2 < 2x_1$ and increases in the range $x_2^2 < x_1$ and $x_2^2 > 2x_1$. Thus, *Serret's* argument is proved erroneous.

Finally, this problem was resolved by *Scheffer.* To apply his arguments, we first need to break the problem into several one-dimensional optimization problems. Rewriting the *Taylor's* series

$$f(\mathbf{x}_0 + \mathbf{h}) - f(\mathbf{x}_0) = G_n(\mathbf{x}_0) + R_n(\mathbf{x}_0 + \theta\mathbf{h}, \mathbf{x}_0).$$

Suppose $\mathbf{x}_0$ is a stationary point. We select one element of the vector **h**, say h_r, to be constant, and allow all other elements of **h** to vary so that $| h_i | \le h_r$. The minimum of $G_n(\mathbf{x}_0)$ with respect to these h_i is determined over the $(n-1)$ dimensional space. This minimum is called $G_n(\mathbf{x}_0)^r$. This procedure is repeated for each $h_i, i = 1, 2, \dots, r, \dots, n$.

Then we shall conclude the following results :

(i) If min $G_n(\mathbf{x}_0)^r$ is positive, then $\mathbf{x}_0$ is a minimum, because $f(\mathbf{x}_0 + \mathbf{h}) - f(\mathbf{x}_0) > 0$.

(ii) If some $G_n(\mathbf{x}_0)^r$ are positive and others are negative, then $\mathbf{x}_0$ is not an extreme point.

(iii) If min $G_n(\mathbf{x}_0)^r$ is zero, then n is increased by one and the entire process is repeated as above.

Note. For full discussion of above material, the interested students are advised to see *"Theory of Maxima and Minima"* by ***H. Hancock**, New York; Dover,* 1950.

27.3. CONSTRAINED PROBLEMS OF MAXIMA AND MINIMA

In this section, we shall deal with the problem of optimization of continuous functions when side conditions or constraints are placed on the variables. These constraints may be in the form of equation or inequality. We shall discuss the case of equality constraints in ***Section* 27.4** and the other case of inequality constraints in ***Section* 27.5.**

We may point out the need for this discussion by the variety of systems limited by constraints. The amount of stock in an inventory system is limited by the size of storage houses. The flow rate of fluid in a series pipe system is limited by the capacity of the smallest link of pipe. There are many other systems too where constraints must be considered.

27.4. CONSTRAINTS IN THE FORM OF EQUATIONS : LAGRANGIAN METHOD

In this section, we shall discuss the *Lagrange's Multipliers Method* which provides a necessary condition for an optimum when constraints are equations. This is a particular case of the more general problem with inequality constraints which we shall discuss in the next section. The development of this method will made initially for a function of two variables. Later, we shall generalize the arguments for any number of variables.

Suppose that it is desired to find an optimum of a differentiable function $\mathbf{f}(x, y)$ whose variables are subject to a constraint $\mathbf{g}(x, y) = 0$ where **g** is also differentiable. If such an optimum occurs at a point (x_0, y_0) at which at least one of the partial derivatives $\partial\mathbf{g}/\partial x$ or $\partial\mathbf{g}/\partial y$ does not vanish, then we can proceed as follows :

Near (x_0, y_0), the equation of the curve $\mathbf{g}(x, y) = 0$ can be written in the form $y = \mathbf{h}(x)$.

Since **g** vanishes along the curve, we have

$$\frac{d}{dx}[\mathbf{g}(x, \mathbf{h}(x))] = \frac{\partial\mathbf{g}}{dx} + \frac{\partial\mathbf{g}}{\partial y}\cdot\frac{d\mathbf{h}}{dx} = 0 \text{ at } (x_0, y_0), \qquad \dots(27.6)$$

and since (x_0, y_0) gives the constrained optimum value, we also have

$$\frac{d}{dx}[\mathbf{f}(x, \mathbf{h}(\mathbf{x}))] = \frac{\partial\mathbf{f}}{\partial x} + \frac{\partial\mathbf{f}}{\partial y}\cdot\frac{d\mathbf{h}}{dx} = 0 \text{ at } (x_0, y_0). \qquad \dots(27.7)$$

Suppose $\frac{\partial g}{\partial y} \neq 0$ at (x_0, y_0), we can define a parameter λ by

$$\frac{\partial f}{\partial y} - \lambda \frac{\partial g}{\partial y} = 0 \text{ at } (x_0, y_0).$$

(The utility of taking –ve sign before λ will be clear in ***Chapter* 29**).

If equation (27.6) is multiplied by λ and the result is subtracted from equation (27.7), we obtain

$$\frac{\partial f}{\partial x} - \lambda \frac{\partial g}{\partial x} = 0 \text{ at } (x_0, y_0).$$

Thus, the equations

$$\frac{\partial f}{\partial x} - \lambda \frac{\partial g}{\partial x} = 0 \quad \text{and} \quad \frac{\partial f}{\partial y} - \lambda \frac{\partial g}{\partial y} = 0 \qquad \text{...(27.8)}$$

hold at (x_0, y_0).

If we set

$$L(x, y, \lambda) = f(x, y) - \lambda g(x, y). \qquad \text{...(27.9)}$$

the equations (27.8) can be written as

$$\frac{\partial L}{\partial x} = 0 \qquad \text{...(27.10)}$$

$$\frac{\partial L}{\partial y} = 0 \qquad \text{...(27.11)}$$

and the original constraint $g(x, y) = 0$ is just

$$\frac{\partial L}{\partial \lambda} = 0. \qquad \text{...(27.12)}$$

In other words, necessary conditions for an *unconstrained* optimum of L (namely, the vanishing of three partial derivatives of L) are also necessary conditions for a *constrained* optimum of $f(x, y)$ (under the assumption that by $\partial g/\partial x$ and $\partial g/\partial y$ do not vanish at the point in question). The function L defined in (27.9) is called the *Lagrangian function* and λ is called the *Lagrangian multiplier.*

We now proceed to generalize these arguments to find an optimum of a differentiable function of n variables subject to m constraints.

27.4–1. Generalized Lagrangian Method to *n*-Dimensional Case

The arguments developed in the above section can be readily generalized as follows :

Suppose we wish to find an optimum of a differentiable function

$$z = f(\mathbf{x}), \mathbf{x} = (x_1, x_2, \dots, x_n) \in R^n,$$

whose variables are subject to the m $(\leq n)$ constraints

$$g_i(\mathbf{x}) = 0, i = 1, 2, \dots, m, \text{ and } \mathbf{x} \geq 0,$$

where the g_i's are also differentiable. We form the *Lagrangian* function

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) \qquad \text{...(27.13)}$$

involving the *Lagrangian* multipliers $\bar{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)$.

Then the necessary conditions for an unconstrained optimum of L (namely, the vanishing of *L's* first partial derivative) are also necessary conditions for a constrained optimum of $f(\mathbf{x})$, provided that the matrix of partial derivatives $\partial g_i/\partial x_j$ has rank m at the point in question.

These necessary conditions for a max. (or min.) of $f(\mathbf{x})$ are the system of $m + n$ equations :

$$\left.\begin{aligned} \frac{\partial L}{\partial x_j} &= \frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_j} = 0, j = 1, 2, \dots, n \\ \frac{\partial L}{\partial \lambda_i} &= -g_i = 0, i = 1, 2, \dots, m \end{aligned}\right\} \qquad \text{...(27.14)}$$

which we can then solve (at least theoretically) for $m + n$ unknowns $x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_m$. We are really interested in obtaining $x_1, x_2, \dots, x_n$.

These necessary conditions also become sufficient for a maximum (minimum) of the objective function if the objective function is concave (convex) and the side constraints are equality ones.

27.4–2. Illustrative Examples

Example 3. *Obtain the set of necessary conditions for the non-lienar programming problem :*

$$\text{Maximize } \mathbf{z} = x_1^2 + 3x_2^2 + 5x_3^2,$$

subject to the constraints : $x_1 + x_2 + 3x_3 = 2$, $5x_1 + 2x_2 + x_3 = 5$, *and* $x_1, x_2, x_3 \geq 0$.

Solution. In this problem, we are given that

$$\mathbf{x} = (x_1, x_2, x_3), f(\mathbf{x}) = x_1^2 + 3x_2^2 + 5x_3^2,$$

$$g_1(\mathbf{x}) = x_1 + x_2 + 3x_3 - 2 = 0, \; g_2(\mathbf{x}) = 5x_1 + 2x_2 + x_3 - 5 = 0.$$

We construct the *Lagrangian* function for maximizing $f(\mathbf{x})$,

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda_1 g_1(\mathbf{x}) - \lambda_2 g_2(\mathbf{x}).$$

This gives the following necessary conditions :

$$\frac{\partial L}{\partial x_1} = 2x_1 - \lambda_1 - 5\lambda_2 = 0, \; \frac{\partial L}{\partial x_2} = 6x_2 - \lambda_1 - 2\lambda_2 = 0, \; \frac{\partial L}{\partial x_3} = 10x_3 - 3\lambda_1 - \lambda_3 = 0,$$

$$\frac{\partial L}{\partial \lambda_1} = -(x_1 + x_2 + 3x_3 - 2) = 0, \; \frac{\partial L}{\partial \lambda_2} = -(5x_1 + 2x_2 + x_3 - 5) = 0.$$

Example 4. *Obtain the necessary and sufficient conditions for the optimum solution of the following non-linear programming problem :*

$$\text{Min. } \mathbf{z} = f(x_1, x_2) = 3e^{2x_1+1} + 2e^{x_2+5}$$

subject to the constraints : $x_1 + x_2 = 7$ *and* $x_1, x_2 \geq 0$.

Solution. Let us have a new differentiable *Lagrangian* function $L(x_1, x_2, \lambda)$ defined by

$$L(x_1, x_2, \lambda) = f(x_1, x_2) - \lambda(x_1 + x_2 - 7) = 3e^{2x_1+1} + 2e^{x_2+5} - \lambda(x_1 + x_2 - 7)$$

where λ is the *Lagrangian* multiplier.

Since the objective function $\mathbf{z} = f(x_1, x_2)$ is convex and the side constraint is an equality, the necessary and sufficient conditions for the minimum of $f(x_1, x_2)$ are given by

$$\frac{\partial L}{\partial x_1} = 6e^{2x_1+1} - \lambda = 0 \text{ or } \lambda = 6e^{2x_1+1}, \; \frac{\partial L}{\partial x_2} = 2e^{x_2+5} - \lambda = 0 \text{ or } \lambda = 2e^{x_2+5}$$

$$\frac{\partial L}{\partial \lambda} = -(x_1 + x_2 - 7) = 0 \text{ or } x_1 + x_2 = 7$$

From these, we have

$$6e^{2x_1+1} = 2e^{x_2+5} = 2e^{7-x_1+5} \; (\because x_2 = 7 - x_1)$$

or $$\log 3 + 2x_1 + 1 = 7 - x_1 + 5$$

or $$x_1 = \tfrac{1}{3}[11 - \log 3], \; x_2 = 7 - \tfrac{1}{3}(11 - \log 3).$$

Example 5. *Find the dimensions of a rectangular parallelopiped with largest volume whose sides are parallel to the coordinate planes, to be inscribed in the ellipsoid*

$$\mathbf{g}(x, y, z) \equiv (x^2/a^2) + (y^2/b^2) + (z^2/c^2) - 1 = 0 \qquad \text{...(27.15)}$$

Solution. Let the dimensions of a rectangular parallelopiped be x, y and z. Its volume is then given by

$$\mathbf{v}(x, y, z) = xyz. \qquad \text{...(27.16)}$$

Forming the Lagrangian function L, we have

$$L(x, y, z, \lambda) = \mathbf{v}(x, y, z) - \lambda \mathbf{g}(x, y, z) \qquad \text{...(27.17)}$$

Now, differentiating (27.17) with respect to each variable and setting the results equal to zero, we obtain

$$\equiv \frac{\partial L}{\partial x} = yz - \frac{2\lambda x}{a^2} = 0, \frac{\partial L}{\partial y} = xz - \frac{2\lambda y}{b^2} = 0, \frac{\partial L}{\partial z} = xy - \frac{2\lambda z}{c^2} = 0, \qquad \text{...(27.18)}$$

Multiplying first three equations of system (27.18) by x, y, z respectively, adding, and then making use of the last equation, we obtain $3\mathbf{v}(x, y, z) - 2\lambda = 0$. Thus, $\lambda = \tfrac{3}{2}\mathbf{v}(\mathbf{x}, y, z)$.

Now, with this value of λ substituted in first three equations respectively, we have

$$x = a/\sqrt{3}\ ,\ y = b/\sqrt{3}\ ,\ z = c/\sqrt{3} \qquad \text{...(27.19)}$$

which is the required answer.

Notes. (i) The results of *Example* 5 also hold good for the special case of the sphere obtained by putting $a = b = c = 1$.

(ii) As practical applications, such a problem can be formulated for a modern auditoriam with a hemisphere for an outer structure :for example, if for ventilation reasons it is desired to have a parallelopiped for the inside.

Example 6. *A positive quantity b is to be divided into n parts in such a way that the product of n parts is to be a maximum. Use Lagrange's multiplier technique to obtain the optimal sub-division.* [Also see page 5.79]

Solution. Let b be divided into n parts $x_1, x_2, \dots, x_n$, so that we have to maximize the function

$$y = x_1 x_2 \dots x_n \qquad \text{...(27.20)}$$

subject to the constraints

$$x_1 + x_2 + \dots + x_n = b,\ \ x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0. \qquad \text{...(27.21)}$$

Now forming the *Lagrangian* function L we have

$$L(x_1, x_2, \dots, x_n, \lambda) = x_1 x_2 \dots x_n - \lambda\,[b - (x_1 + x_2 + \dots + x_n)] \qquad \text{...(27.22)}$$

Now, differentiating (27.22) with respect to each variable and setting the results equal to zero, we get

$$\frac{\partial L}{\partial x_1} = x_2 x_3 \dots x_n - \lambda\,(0 - 1) = 0\ ,\ \frac{\partial L}{\partial x_2} = x_1 x_3 \dots x_n - \lambda\,(0-1) = 0, \dots\dots,$$

$$\text{and} \quad \frac{\partial L}{\partial x_n} = x_1 x_2 \dots x_{n-1} - \lambda\,(0-1) = 0,\ \frac{\partial L}{\partial \lambda} = 0 - 1\,(b - x_1 - x_2 - \dots - x_n) = 0. \qquad \text{...(27.23)}$$

Multiplying first n equations of the system (27.23) by $x_1, x_2, \dots, x_n$ respectively, adding and then making use of last equation, we obtain

$$n\,(x_1 . x_2 \dots x_n) + \lambda\,(x_1 + x_2 + \dots + x_n) = 0$$

or

$$\lambda = -\,n\,(x_1 x_2 \dots x_n)/b \qquad (\because x_1 + x_2 + \dots + x_n = b)$$

Now, substituting this value of λ in the first n equations respectively, we obtain

$$x_1 = x_2 = x_3 = \dots = x_n = b/n$$

giving

$$y = (b/n)\,(b/n) \dots n \text{ times} = (b/n)^n.$$

These values of $x_1, x_2, \dots, x_n$ satisfy all the constraints and gives a value for y larger than the value when any of $x_1, x_2, \dots, x_n$ is zero, so together constitute the optimal subdivision of b.

Remark : In Lagrangian multiplier method, sometimes it becomes difficult to solve the system of n simultaneous equations. This difficulty can be removed by using dynamic programming approach based on the *"Bellman's principle of optimality"*. Dynamic programming technique converts one problem involving n variables into n sub-problems, each in one variable. The solution is obtained in an orderly manner by starting from one stage to the next and is completed after the final stage is reached. The detailed discussion of this technique is given in the last chapter.

27.4–3. Sufficient Conditions for Maximum (Minimum) of Objective Function (with single equality constraint)

In case the concavity (convexity) of the objective function is not known, the method of *Lagrange* multipliers can be generalized to obtain a set of sufficient conditions for a maximum (minimum) of the objective function

Let us consider the non-linear programming problem involving n variables and single constraint.

$$\text{Max. (or Min.) } \mathbf{z} = f(\mathbf{x})\ ,\ \mathbf{x} \in R^n$$

subject to the conditions : $g(\mathbf{x}) = 0,\ \mathbf{x} \geq 0.$

Let the *Lagrangian* function be : $L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x})$.

The necessary conditions for a stationary point to be a maximum or minimum are :

$$\frac{\partial L}{\partial x_j} = \frac{\partial f}{\partial x_j} - \lambda \frac{\partial g}{\partial x_j} = 0\ (j = 1, 2, \dots, n), \ \text{ and } \ \frac{\partial L}{\partial \lambda} = -\,g(\mathbf{x}) = 0\,.$$

The value of λ is obtained by $\lambda = \dfrac{\partial f/\partial x_j}{\partial g/\partial x_j}$ for $j = 1, 2, \dots, n$

The sufficient conditions for a maximum or minimum need the computation of $(n-1)$ principal minors of the determinant for each stationary point, as given below :

$$\begin{vmatrix} 0 & \frac{\partial g}{\partial x_1} & \frac{\partial g}{\partial x_2} & \cdots & \frac{\partial g}{\partial x_n} \\ \frac{\partial g}{\partial x_1} & \frac{\partial^2 f}{\partial x_1^2} - \lambda \frac{\partial^2 g}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} - \lambda \frac{\partial^2 g}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} - \lambda \frac{\partial^2 g}{\partial x_1 \partial x_n} \\ \frac{\partial g}{\partial x_2} & \frac{\partial^2 g}{\partial x_2 \partial x_1} - \lambda \frac{\partial^2 g}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} - \lambda \frac{\partial^2 g}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} - \lambda \frac{\partial^2 g}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{\partial g}{\partial x_n} & \frac{\partial^2 f}{\partial x_n \partial x_1} - \lambda \frac{\partial^2 g}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} - \lambda \frac{\partial^2 g}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} - \lambda \frac{\partial^2 g}{\partial x_n^2} \end{vmatrix} = \Delta_{n+1} \text{ (say)}$$

If $\Delta_3 < 0, \Delta_4 < 0, \Delta_5 > 0, \ldots$, the signs are *alternately* positive and negative, the stationary point is a local maximum.

If $\Delta_3 < 0, \Delta_4 < 0, \Delta_5 < 0, \ldots, \Delta_{n+1} < 0$, the sign being always negative, the stationary point is a local minimum.

Example 7. *Solve the non-linear programming problem :*

$$Min.\ \mathbf{z} = 2x_1^2 - 24x_1 + 2x_2^2 - 8x_2 + 2x_3^2 - 12x_3 + 200$$

subject to the constraints :

$$x_1 + x_2 + x_3 = 11, \quad and \quad x_1, x_2, x_3 \geq 0.$$ **[Agra 99, 98]**

Solution. The *Lagrangian* function can be formulated as follows :

$$L(x_1, x_2, x_3, \lambda) = 2x_1^2 - 24x_1 + 2x_2^2 - 8x_2 + 2x_3^2 - 12x_3 + 200 - \lambda(x_1 + x_2 + x_3 - 11).$$

The necessary conditions for the stationary point are :

$$\frac{\partial L}{\partial x_1} = 4x_1 - 24 - \lambda = 0, \quad \frac{\partial L}{\partial x_2} = 4x_2 - 8 - \lambda = 0, \quad \frac{\partial L}{\partial x_3} = 4x_3 - 12 - \lambda = 0,$$

$$\frac{\partial L}{\partial \lambda} = -(x_1 + x_2 + x_3 - 11) = 0.$$

By solving these simultaneous equations, we get the stationary point

$$\mathbf{x}^* = (x_1, x_2, x_3) = (6, 2, 3); \lambda = 0.$$

The sufficient condition for the stationary point to be a minimum is that the minors Δ_3 and Δ_4 must be both negative. To verify this, we have

$$\Delta_3 = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 4 & 0 \\ 1 & 0 & 4 \end{vmatrix} = -8 \quad \text{and} \quad \Delta_4 = \begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 4 & 0 & 0 \\ 1 & 0 & 4 & 0 \\ 1 & 0 & 0 & 4 \end{vmatrix} = -48.$$

Thus, $\mathbf{x}^* = (6, 2, 3)$ is the solution to the given NLPP.

Q. 1. Examine $\mathbf{z} = 6x_1x_2$ for maxima and minima under the requirement $2x_1 + x_2 = 10$.

2. What happens when the problem becomes that of maximizing $\mathbf{z} = 6x_1x_2 - 10x_3$ under the constraint equation $3x_1 + x_2 + 3x_3 = 10$.

27.4–4. Sufficient Conditions for Max. (Min.) of Objective Function (with more than one equality constraints)

Let us now consider the NLPP involving more than one constraint.

The problem is : Optimize $\mathbf{z} = f(\mathbf{x})$, $\mathbf{x} \in R^n$ subject to the constraints :

$$g_i(\mathbf{x}) = 0, i = 1, 2, \ldots, m, \text{ and } \mathbf{x} \geq 0.$$

In order to optimize $\mathbf{z} = f(\mathbf{x})$, the *Lagrangian* function

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) \quad (m < n)$$

contains the m *Lagrangian* multipliers $\bar{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_m)$. It may be verified that the equations :

$$\frac{\partial L}{\partial x_j} = 0 \quad \text{for } j = 1, 2, \ldots, n$$

$$\frac{\partial L}{\partial \lambda_i} = 0 \quad \text{for } i = 1, 2, \ldots, m$$

provide the necessary conditions for stationary points of $f(\mathbf{x})$. So the optimization of $f(\mathbf{x})$ subject to $g(\mathbf{x}) = 0$ is equivalent to the optimization of $L(\mathbf{x}, \lambda)$. The *Lagrange* multiplier method for a stationary point of $f(\mathbf{x})$ to be a maxima or minima is staed here without proof.

For this, we assume that the function $L(\mathbf{x}, \lambda)$, $f(\mathbf{x})$ and $g(\mathbf{x})$ all possess partial derivatives of first and second order with respect to the decision variables.

Let $$M = \left| \frac{\partial^2 L(\mathbf{x}, \lambda)}{\partial x_i \, \partial x_j} \right|_{n \times n} \quad \text{for all } i \text{ and } j.$$

be the matrix of second order partial derivatives of $L(\mathbf{x}, \lambda)$ w.r.t. decision variables,

$$V = \left[\frac{\partial g_i(\mathbf{x})}{\partial x_j} \right]_{m \times n}$$

where $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$.

Now define the square matrix $$H_B = \begin{bmatrix} O & V \\ V^T & M \end{bmatrix}_{(m+n) \times (m+n)}$$

where $\mathbf{O}$ is an $m \times m$ null matrix. The matrix H_B is called the *bordered Hessian matrix*. Then, the sufficient conditions for maximum and minimum can be stated as below.

Sufficient Conditions for Maximum and Minimum :

Let $(\mathbf{x}^*, \lambda^*)$ be the stationary point for the *Lagrangian* function $L(\mathbf{x}, \lambda)$, and H_B^* be the value of corresponding bordered *Hessian* matrix computed at this stationary point. Then,

(i) $\mathbf{x}^*$ is a maximum point, if starting with principal minor of order $(m + 1)$, the last $(n - m)$ principal minors of H_B^* form an alternating sign pattern starting with $(-1)^{m+n}$; and

(ii) $\mathbf{x}^*$ is a minimum point, if starting with the principal minor of order $(2m + 1)$; the last $(n - m)$ principal minors of H_B^* have the sign of $(-1)^m$.

Note. It may be found that the above conditions are only *sufficient* for identifying an extreme point, but not *necessary*. In other words, a stationary point may be an extreme point without satisfying the above conditions.

Example 8. *Solve the non-linear programming problem :*

$$\text{Optimize } \mathbf{z} = 4x_1^2 + 2x_2^2 + x_3^2 - 4x_1x_2$$

subject to the constraints : $x_1 + x_2 + x_3 = 15, 2x_1 - x_2 + 2x_3 = 20,$ *and* $x_1, x_2, x_3 \geq 0$.

Solution. We are given that $f(x) = 4x_1^2 + 2x_2^2 + x_3^2 - 4x_1x_2$ subject to the constraints :

$$g_1(\mathbf{x}) = x_1 + x_2 + x_3 - 15, \quad g_2(\mathbf{x}) = 2x_1 - x_2 + 2x_3 - 20.$$

The *Lagrangian* function is given by

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda_1 g_1(\mathbf{x}) - \lambda_2 g_2(\mathbf{x})$$
$$= (4x_1^2 + 2x_2^2 + x_3^2 - 4x_1x_2) - \lambda_1 (x_1 + x_2 + x_3 - 15) - \lambda_2 (2x_1 - x_2 + 2x_3 - 20).$$

The stationary point $(\mathbf{x}^*, \lambda^*)$ can be obtained by the following necessary conditions :

$$\frac{\partial L}{\partial x_1} = 8x_1 - 4x_2 - \lambda_1 - 2\lambda_2 = 0, \ \ldots\text{(i)} \quad \frac{\partial L}{\partial x_2} = 4x_2 - 4x_1 - \lambda_1 + \lambda_2 = 0 \ \ldots\text{(ii)}, \quad \frac{\partial L}{\partial x_3} = 2x_3 - \lambda_1 - 2\lambda_2 = 0, \ \ldots\text{(iii)}$$

$$\frac{\partial L}{\partial \lambda_1} = -(x_1 + x_2 + x_3 - 15) = 0, \ldots\text{(iv)} \quad \text{and} \quad \frac{\partial L}{\partial \lambda_2} = -(2x_1 - x_2 + 2x_3 - 20) = 0. \quad \ldots\text{(v)}$$

Adding (i) and (ii), $x_1 = (2\lambda_1 + \lambda_2)/4$. From (i) $x_2 = 3\lambda_1/4$. From (iii) $x_3 = \dfrac{\lambda_1 + 2x_2}{2}$

Putting the values in (iv) & (v), we get $7\lambda_1 + 5\lambda_2 = 60$ and $5\lambda_1 + 10\lambda_2 = 80$
Solving these eqns, $\lambda_1 = 40/9, \lambda_2 = 52/9$.

$$\therefore \quad x_1 = \frac{33}{9}, \; x_2 = \frac{10}{3}, \; x_3 = 8.$$

$$\mathbf{x}^* = (x_1, x_2, x_3) = (33/9, 10/3, 8), \quad \text{and} \quad \lambda^* = (\lambda_1, \lambda_2) = (40/9, 52/9).$$

For this stationary point (x^*, λ^*), the bordered *Hessian* matrix is given by

$$H_B^* = \begin{bmatrix} 0 & 0 & : & 1 & 1 & 1 \\ 0 & 0 & : & 2 & -1 & 2 \\ \cdots & \cdots & : & \cdots & \cdots & \cdots \\ 1 & 2 & : & 8 & -4 & 0 \\ 1 & -1 & : & -4 & 4 & 0 \\ 1 & 2 & : & 0 & 0 & 2 \end{bmatrix}$$

Since $m = 2$ and $n = 3$ here, so $n - m = 1, 2m + 1 = 5$. This means that we only need to check the determinant of H_B^* and it must have the positive sign (*i.e.*, the sign of $(-1)^2$)

Now, since $| H_B^* | = 72$ which is positive, $\mathbf{x}^*$ is a minimum point.

Q. 1. Minimize $\mathbf{z} = x_1^2 + x_2^2 + x_3^2$, subject to the constraints : $4x_1 + x_2^2 + 2x_3 = 14$, and $x_1, x_2, x_3 \geq 0$.
[Ans. $x_1 = 0.81, x_2 = 0.35, x_3 = 0.28$; min $\mathbf{z} = 0.857$**]**

2. Minimize $\mathbf{z} = 2x_1^2 + x_2^2 + 3x_3^2 + 10x_1 + 8x_2 + 6x_3 - 100$, subject to the constraints : $x_1 + x_2 + x_3 = 20$, and $x_1, x_2, x_3 \geq 0$.
[Ans. $x_1 = 5, x_2 = 11, x_3 = 4$; min. $\mathbf{z} = 281$**]** **[Agra 99]**

27.5. CONSTRAINTS IN THE FORM OF INEQUALITIES
(Kuhn-Tucker Necessary and Sufficient Conditions)

This section is concerned with developing the necessary and sufficient conditions for identifying the stationary points of the general inequality constrained optimization problems. These conditions are called the *Kuhn-Tucker Conditions,* after the men who developed them. The development is mainly based on *Lagrangian* method. These conditions are sufficient under certain limitations which will be stated in the following section.

Theorem 27.4. (Kuhn-Tucker Necessary Conditions). *Given the problem to maximize*

$$f(\mathbf{x}), \; \mathbf{x} = (x_1, x_2, \ldots, x_n)$$

subject to m number of inequality constraints

$$g_i(\mathbf{x}) \leq b_i, \; i = 1, 2, \ldots, m \tag{27.24}$$

including the non-negativity constraints $\mathbf{x} \geq 0$ *which are written as* $-\mathbf{x} \leq 0$, *the necessary conditions for a local maxima (or stationary point(s)) at* $\bar{\mathbf{x}}$ *are*

(i) $\dfrac{\partial L(\bar{\mathbf{x}}, \bar{\lambda}, \bar{\mathbf{s}})}{\partial x_j} = 0, j = 1, 2, \ldots, n$ *(ii)* $\bar{\lambda}_i [g_i(\bar{\mathbf{x}}) - b_i] = 0$ *(iii)* $g_i(\bar{\mathbf{x}}) \leq b_i$ *(iv)* $\bar{\lambda}_i \geq 0, \; i = 1, 2, \ldots, m.$

Proof. In the given problem, each of the inequality constraints can be converted into equations by adding the appropriate *non-negative* slack variables. Thus, to satisfy the non-negativity condition, if we add a non-negative slack variable s_i^2 to the ith constraint $g_i(\mathbf{x}) \leq b_i$, we obtain*

$$g_i(\mathbf{x}) + s_i^2 = b_i, \; i = 1, 2, \ldots, m$$

subtracting b_i gives

$$G_i(\mathbf{x}, s_i) = g_i(\mathbf{x}) + s_i^2 - b_i = 0, \; i = 1, 2, \ldots, m \quad \ldots(27.25)$$

Now, our problem becomes in the following form for application of *Lagrangian* method given in the preceding section :

$$\text{Max. } f(\mathbf{x})$$

subject to equality constraint

$$G_i(\mathbf{x}, s_i) = 0, \; i = 1, 2, \ldots, m \qquad \ldots(27.26)$$

In order to obtain all stationary points, we first form the *Lagrangian* function given by

$$L(\mathbf{x}, \lambda, s) = f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i G_i(\mathbf{x}, s_i) = f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i \, [g_i(\mathbf{x}) + s_i^2 - b_i]. \qquad \ldots(27.27)$$

$$V = \left\| \left| \frac{\partial g_i}{\partial x_j} \right| \right\|_{m \times n} = \begin{bmatrix} \partial g_1/\partial x_1 & \partial g_1/\partial x_2 & \partial g_1/\partial x_3 \\ \partial g_2/\partial x_1 & \partial g_2/\partial x_2 & \partial g_2/\partial x_3 \end{bmatrix}$$

Then the stationary points are obtained by solving the equations (obtained by equating to zero the partial derivatives of (27.27) w.r.t. x_j, λ_i, s_i, respectively, ($j = 1, 2, \ldots, n$; $i = 1, 2, \ldots, m$).

$$\frac{\partial L(\cdot)}{\partial x_j} = 0 = \frac{\partial f(\mathbf{x})}{\partial x_j} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i(\mathbf{x})}{\partial x_j}, \; j = 1, 2, \ldots, n \qquad \ldots(27.28)$$

$$\frac{\partial L(\cdot)}{\partial \lambda_i} = 0 = G_i(\mathbf{x}, s_i) = g_i(\mathbf{x}) + s_i^2 - b_i, \; i = 1, 2, \ldots, m \qquad \ldots(27.29)$$

$$\frac{\partial L(\cdot)}{\partial s_i} = 0 = -2\lambda_i s_i, \qquad i = 1, 2, \ldots, m. \qquad \ldots(27.30)$$

Multiplying the last equation (27.30) by ½ s_i, we get

$$\lambda_i s_i^2 = 0 \qquad \ldots(27.31)$$

We now solve $G_i(\mathbf{x}, s_i) = 0$ for

$$s_i^2 = b_i - g_i(\mathbf{x}) \qquad \ldots(27.32)$$

Substituting the value of s_i^2 from (27.32) in (27.31), we get

$$\lambda_i \, [b_i - g_i(\mathbf{x})] = 0, \; i = 1, 2, \ldots, m \qquad \ldots(27.33)$$

Thus, the equations (27.28), (27.33) and constraint (27.24) satisfied by the stationary point $\bar{\mathbf{x}}_0 = (\bar{\mathbf{x}}, \bar{\lambda}, \bar{\mathbf{s}})$ proves the necessary conditions *(i)*, *(ii)* and *(iii)* respectively.

We now proceed to prove the final (*i.e.* fourth) requirement $\bar{\lambda}_i \geq 0$, $i = 1, 2, \ldots, m$.

Since $\bar{\lambda}_i$ measures the rate of variation of f with respect to b_i, we have

$$\frac{\partial f(\bar{\mathbf{x}})}{\partial b_i} = +\bar{\lambda}_i.$$ (see its proof in the ***Appendix*** on page 1115)

From equation (27.30), we know that either $\bar{\lambda}_i = 0$, or $\bar{s}_i = 0$, or both vanish at the optimal condition. Let us investigate the case when $\bar{s}_i \neq 0$. This implies that the constraint is satisfied as strict inequality at $\bar{\mathbf{x}}$ and, consequently, if we relaxed the constraint (make b_i larger) the extreme point will not be affected. Therefore, the change in the optimal value of the objective function with changes in b_i will be zero, *i.e.*

$$\frac{\partial f(\bar{\mathbf{x}})}{\partial b_i} = +\bar{\lambda}_i = 0.$$

Now, suppose that $\bar{\lambda}_i \neq 0$. This means that the slack variable $\bar{s}_i$ vanishes. Thus $g_i(\bar{\mathbf{x}}) = b_i$.

If possible, let us suppose $\bar{\lambda}_i < 0$. Then, $\partial f(\bar{\mathbf{x}})/\partial b_i < 0$.

This implies that as b_i is increased, the objective function decreases. However, as b_i increases, more space become feasible and the optimal value of the objective function $f(\bar{\mathbf{x}})$ clearly cannot decrease. This contradicts our assumption. Hence, at an optimal solution $\bar{\lambda}_i \geq 0$.

Similarly, for the case of minimization, as b_i increases, $f(\bar{\mathbf{x}})$ cannot increase which implies $\bar{\lambda}_i \leq 0$. It must be noted that if the constraints are equations, that is, $g_1(\bar{\mathbf{x}}) = b_i$, then $\bar{\lambda}_i$ becomes unrestricted in sign.

Theorem 27.5. (*Kuhn-Tucker Sufficient Conditions*). *The Kuhn-Tucker conditions which are necessary by preceding Theorem 27.4 are also sufficient if $f(\mathbf{x})$ is concave and the feasible space is convex, i.e. if $f(\mathbf{x})$ is strictly concave and $g_i(\mathbf{x})$, $i = 1, 2, \ldots m$ are convex.*

Proof. Let us suppose that $f(\mathbf{x})$ and $g_i(\mathbf{x})$ satisfy the condition given in the statement of the theorem. Then,

$$L(\mathbf{x}, \lambda, \mathbf{s}) = f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i [g_i(\mathbf{x}) + s_i^2 - b_i]^*.$$

If $\lambda_i \geq 0$, then $-\lambda_i g_i(\mathbf{x})$ is concave if $g_i(x)$ is convex.

Hence, $$f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x})$$

is strictly concave. Since $\lambda_i s_i^2 = 0$ and $\lambda_i b_i$ is constant, if

$$f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x})$$

is concave, $L(\mathbf{x}, \lambda, \mathbf{s})$ is concave. We have shown that a necessary condition for $f(\mathbf{x})$ to be maximum at $\bar{\mathbf{x}}$ is that $L(\bar{\mathbf{x}}, \bar{\lambda}, \bar{\mathbf{s}})$ has a stationary point at $\bar{\mathbf{x}}$. However, if $L(\bar{\mathbf{x}}, \bar{\lambda}, \bar{\mathbf{s}})$ is strictly concave, its derivative must vanish at one point only, Consequently, this point must be the local maximum. Hence, the *Kuhn-Tucker* conditions *(i), (ii), (iii)* and *(iv)* are also sufficient for *an absolute (global)* maximum of $f(\mathbf{x})$ at $\bar{\mathbf{x}}$.

By a similar argument, it can be proved that for the minimization problem, the *Kuhn-Tucker* conditions are sufficient provided $f(\mathbf{x})$ and $g_i(\mathbf{x})$ for all i are convex.

Important Remarks :

(1) From above two theorems, we conclude that *Kuhn-Tucker conditions :*

(i) $\dfrac{\partial L(\bar{\mathbf{x}}, \bar{\lambda}, \bar{\mathbf{s}})}{\partial x_j} = 0, j = 1, 2, \ldots, n$

$$\left.\begin{array}{ll} (ii) & \lambda_i [g_i(\bar{\mathbf{x}}) - b_i] = 0 \\ (iii) & g_i(\bar{\mathbf{x}}) \leq b_i \\ (iv) & \bar{\lambda}_i \geq 0 \end{array}\right\} i = 1, 2, \ldots, m$$

are necessary as well as sufficient for an *absolute* (or *global)* maximum of $f(\mathbf{x})$ at $\bar{\mathbf{x}}$.

(2) It can be easily verified that these conditions are applicable to minimization case with the exception that λ must be ≤ 0.

(3) It must be noted in both the maximization and the minimization cases, that the *Lagrange* multipliers corresponding to *equality* constraints must be unrestricted in sign.

Q. 1. Discuss the economic interpretation of *Lagrangian Multipliers,* the duality theory, and derive the *Kuhn-Tucker* conditions for the non-linear programming problem :

Max. $z = f(\mathbf{x})$, subject to the constraints : $g_i(\mathbf{x}) \leq b_i$, $i = 1, 2, \ldots, m$.

2. State and prove *Kuhn-Tucker* necesary and sufficient conditions in non-linear programming. **[I.A.S. (Maths) 88, 86]**

27.5–1. Illustrative Example

Example 9. *Write the Kuhn-Tucker conditions for the following minimization problem :* **[I.A.S. (Main) 93]**

$$\textit{Minimize. } f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2, \textit{ subject to}$$

$$\begin{aligned} g_1(\mathbf{x}) &= 2x_1 + x_2 &\leq 5, \\ g_2(\mathbf{x}) &= x_1 + x_3 &\leq 2, \\ g_3(\mathbf{x}) &= -x_1 &\leq -1, \\ g_4(\mathbf{x}) &= -x_2 &\leq -2, \\ g_5(\mathbf{x}) &= -x_3 &\leq 0. \end{aligned}$$

Solution. Since this is a minimization problem, then $\lambda_i \leq 0$. The *Kuhn-Tucker* conditions are thus given by

$$(i)\quad (2x_1, 2x_2, 2x_3) + (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} = 0.$$

---

*Here s_i is squared to ensure that it is non-negative. Had we not squared it we would require $s_i \geq 0$ as side constraint also.

(ii) $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) \geq 0$.

(iii) $\lambda_1(g_1 - 5) = \lambda_2(g_2 - 2) = \lambda_3(g_3 + 1) = \lambda_4(g_4 + 2) = \lambda_5 g_5 = 0$.

(iv) $g(\mathbf{x}) \leq 0$, where $g(\mathbf{x}) = g_i(\mathbf{x}) - b_i$, $i = 1, 2, \ldots, 5$.

These conditions can be simplified to the following form :

$$\begin{array}{llllll} \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 & \leq 0 & \lambda_1(2x_1 + x_2 - 5) & = 0 & \lambda_5 x_3 & = 0 \\ 2x_1 + 2\lambda_1 + \lambda_2 - \lambda_3 & = 0 & \lambda_2(x_1 + x_3 - 2) & = 0 & 2x_1 + x_2 & \leq 5 \\ 2x_2 + \lambda_1 - \lambda_4 & = 0 & \lambda_3(1 - x_1) & = 0 & x_1 + x_3 & \leq 2 \\ 2x_3 + \lambda_2 - \lambda_5 & = 0 & \lambda_4(2 - x_2) & = 0 & x_1 \geq 1, x_2 \geq 2, & x_3 \geq 0 \end{array}$$

Solving above equations, we get $x_1 = 1, x_2 = 2, x_3 = 0, \lambda_1 = \lambda_2 = \lambda_3 = 2, \lambda_4 = 4, \lambda_5 = 0$.

Since the function $f(\mathbf{x})$ is convex and the solution space $g(\mathbf{x}) \leq 0$ is also convex, then $L(\mathbf{x}, \lambda, \mathbf{s})$ must be convex and the resulting stationary point will give the (global) constrained minimum. The given example shows, however, that it is difficult in general to solve the resulting conditions explicitly. That is why this procedure is not suitable for numerical computations. However, the importance of the *Kuhn-Tucker* conditions will come in quadratic and geometric programming algorithms to be discussed in the following chapters.

Example 10. *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 : $x_1 + x_2 \leq 2$, $\quad 2x_1 + 3x_2 \leq 12$, *and* $x_1, x_2 \geq 0$. **[I.A.S. (Main) 92]**

Solution. Here $f(\mathbf{x}) = -x_1^2 - x_2^2 - x_3^2 + 4x_1 + 6x_2$, $\mathbf{x} \in R^n$

$$g_1(\mathbf{x}) = x_1 + x_2 - 2, \quad g_2(\mathbf{x}) = 2x_1 + 3x_2 - 12.$$

First we decide about the concavity-convexity of $f(\mathbf{x})$. For this we compute the bordered *Hessian* matrix

$$H_B = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{bmatrix}, \quad \begin{array}{r} n = 3 \\ m = 2 \\ n - m = 1 \end{array} \quad \therefore \quad |H_B| = -8 < 0.$$

The objective function $f(\mathbf{x})$ is concave if the principal minors of matrix H_B alternate in sign, starting with the negative sign. If the principal minors are positive, the objective function is convex. So in this case $f(\mathbf{x})$ is *concave.*

Clearly, $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are *convex* in $\mathbf{x}$. Thus the *Kuhn-Tucker conditions* will be the necessary and sufficient conditions for a maximum. These conditions are obtained by partial derivatives of *Lagrangian function* :

$$L(\mathbf{x}, \lambda, \mathbf{s}) = f(\mathbf{x}) - \lambda_1 [g_1(\mathbf{x}) + s_1^2] - \lambda_2 [g_2(\mathbf{x}) + s_2^2]$$

where $\mathbf{s} = (s_1, s_2)$, $\lambda = (\lambda_1, \lambda_2)$, and s_1, s_2 being slack variables, and λ_1, λ_2 are *Lagrangian* multipliers.

The *Kuhn-Tucker conditions* are given by

(a) *(i)* $-2x_1 + 4 = \lambda_1 + 2\lambda_2$, *(ii)* $-2x_2 + 6 = \lambda_1 + 3\lambda_2$, *(iii)* $-2x_3 = 0$.

(b) *(i)* $\lambda_1(x_1 + x_2 - 2) = 0$, *(ii)* $\lambda_2(2x_1 + 3x_2 - 12) = 0$.

(c) *(i)* $x_1 + x_2 - 2 \leq 0$, *(ii)* $2x_1 + 3x_2 - 12 \leq 0$.

(d) $\lambda_1 \geq 0, \lambda_2 \geq 0$.

Now, four different cases may arise :

Case 1. **($\lambda_1 = 0$, and $\lambda_2 = 0$).** In this case, the system **(a)** of equations give : $x_1 = 2, x_2 = 3, x_3 = 0$. However, this solution violates both the inequalities of **(c)** given above.

Case 2. **($\lambda_1 = 0, \lambda_2 \neq 0$).** In this case, **(b)** give $2x_1 + 3x_2 = 12$ and **(a)** *(i)* and *(ii)* give $-2x_1 + 4 = 2\lambda_2$, $-2x_2 + 6 = 3\lambda_2$. The solution of these simultaneous equations gives $x_1 = 24/13$, $x_2 = 36/13$, $\lambda_2 = 2/13 > 0$; also equation **(a)** *(iii)* gives $x_3 = 0$. However, this solution violates **(c)** *(i)*. So this solution is discarded.

Case 3. **($\lambda_1 \neq 0, \lambda_2 \neq 0$).** In this case, **(b)** *(i)* and *(ii)* give $x_1 + x_2 = 2$ and $2x_1 + 3x_2 = 12$. These equations give $x_1 = -6$ and $x_2 = 8$. Thus, **(a)** *(i)*, *(ii)* and *(iii)* yield $x_3 = 0, \lambda_1 = 68, \lambda_2 = -26$. Since $\lambda_2 = -26$ violates the condition **(d)**, so this solution is also discarded.

Case 4. **($\lambda_1 \neq 0, \lambda_2 = 0$).** In this case, **(b)** *(i)* gives $x_1 + x_2 = 2$. This together with **(a)** *(i)* and *(ii)* give $x_1 = 1/2$, $x_2 = 3/2$, $\lambda = 3 > 0$. Further from **(a)** *(iii)* $x_3 = 0$. This solution does not violate any of the *Kuhn-Tucker conditions.*

Hence the optimum (maximum) solution to the given problem is

$$x_1 = 1/2, x_2 = 3/2, x_3 = 0 \text{ with } \lambda_1 = 3, \lambda_2 = 0,$$

the maximum value of the objective function is $\mathbf{z}^* = 17/2$.

EXAMINATION PROBLEMS

1. Verify that the function $f(x_1, x_2, x_3) = 2x_1x_2x_3 - 4x_1x_3 + x_1^2 + x_2^2 + x_3^2 - 2x_1 - 4x_2 + 4x_3$, has the stationary points (0, 3, 1), (2, 1, 1), (2, 3, –1), (0, 1, –1), and (1, 2, 0). Use the sufficiency condition to check for the extreme points.
2. Write the *Kuhn-Tucker* necessary conditions for the following problems :
   (*i*) Max. $f(\mathbf{x}) = x_1^3 - x_2^2 + x_1x_3^2$, subject to $x_1 + x_2^2 + x_3 = 5, 5x_1^2 - x_2^2 - x_3 \geq 2, x_1, x_2, x_3 \geq 0$.
   (*ii*) Min. $f(\mathbf{x}) = x_1^4 + x_2^2 + 5x_1x_2x_3$, subject to $x_1^2 - x_2^2 - x_3^3 \leq 10, x_1^3 + x_2^2 + 4x_3^2 \geq 20, x_1, x_2, x_3 \geq 0$.
3. Solve the following non-linear programming problems, using the method of *Lagrangian* multipliers :
   (*i*) Max. $\mathbf{z} = 6x_1 + 8x_2 - x_1^2 - x_1^2 - x_2^2$,
   subject to the constraints :
   $4x_1 + 3x_2 = 16,\ 3x_1 + 5x_2 = 15$, and $x_1, x_2 \geq 0$
   [Ans. $x_1 = 35/11, x_2 = 12/11, \mathbf{z}^* = 16.5$]
   (*ii*) Min. $\mathbf{z} = x_1^2 + x_2^2 + x_3^2$
   subject to the constraints :
   $x_1 + x_2 + 3x_3 = 2, 5x_1 + 2x_2 + x_3 = 5, x_1, x_2, x_3 \geq 0$.
   [Ans. $x_1 = 0.81, x_2 = 0.35, x_3 = 0.28, \mathbf{z}^* = 0.86$]
4. Use the *Kuhn-Tucker* conditions to solve the following non-linear programming problems :
   (*i*) Max $\mathbf{z} = 2x_1^2 + 12x_1x_2 - 7x_2^2$
   subject to the constraints :
   $2x_1 + 5x_2 \leq 98$, and $x_1, x_2 \geq 0$.
   [Ans. $x_1 = 44, x_2 = 2, \mathbf{z}^* = 4900$]
   (*ii*) Max. $\mathbf{z} = 8x_1 + 10x_2 - x_1^2 - x_2^2$
   subject to the constraints :
   $3x_1 + 2x_2 \leq 6$, and $x_1 \geq 0, x_2 \geq 0$. **[I.A.S. (Main) 91]**
   [Ans. $x_1 = 4/13, x_2 = 33/13, \mathbf{z}^* = 21.3$]
   (*iii*) Max. $\mathbf{z} = 2x_1 - x_1^2 + x_2$,
   subject to the constraints :
   $2x_1 + 3x_2 \leq 6, 2x_1 + x_2 \leq 4, x_1, x_2 \geq 0$
   [Delhi (Stat.) 96]
   [Ans. $x_1 = 2/3, x_2 = 14/9, \mathbf{z}^* = 22/9$]
   (*iv*) Max. $\mathbf{z} = 7x_1^2 + 6x_1 + 5x_2$,
   subject to the constraints :
   $x_1 + 2x_2 \leq 10, x_1 - 3x_2 \leq 9, x_1, x_2 \geq 0$.
   [Ans. $x_1 = 48/5, x_2 = 1/5, \mathbf{z}^* = 703.5$]
   (*v*) Max. $\mathbf{z} = 7x_1^2 - 6x_1 + 5x_2^2$
   subject to the constraints :
   $x_1 + 2x_2 \leq 10, x_1 - 3x_2 \leq 9, x_1, x_2 \geq 0$
   [Ans. $x_1 = 48/5, x_2 = 1/5$]
   (*vi*) Max. $\mathbf{z} = 6x_1^2 + 5x_2^2$,
   subject to the constraints :
   $x_1 + 5x_2 \geq 3, x_1, x_2 \geq 0$.
   [Ans. $x_1 = 3/31, x_2 = 18/3, \mathbf{z}^* = 54/31$]
   (*vii*) Max. $\mathbf{z} = 2x_1^2 + 12x_1x_2 - 7x_2^2$, subject to the constraints : $2x_1 + 5x_2 \leq 98, x_1, x_2 \geq 0$.
   [Ans. $x_1 = 0, x_2 = 0$, *max.* $\mathbf{z} = 0$]
5. Define convex programming problem. What is the *Lagrangian* function associated with it ? Solve the non-linear programming problem ?
   Min. $\mathbf{z} = -\log x_1 - \log x_2$, subejct to the constraints : $x_1 + x_2 \leq 2,\ x_1, x_2 \geq 0$.
   [Ans. $x_1 = 1, x_2 = 1$, min $\mathbf{z} = 0$]
6. Solve the following NLPP : Max $\mathbf{z} = 8x_1^2 + 2x_2^2$, subejct to the constraints : $x_1^2 + x_2^2 \leq 9, x_1 \leq 2$, and $x_1, x_2 \geq 0$.
7. A manufacturing concern produces two products, say *A* and *B*. The costs of production for these two products are displayed in following table :

	Number of units produced	Cost of production in Rupees
Product A	x_1	$60 + 1.2x_1 + 0.001x_1^2$
Product B	x_2	$40 + 2x_2 + 0.001x_2^2$

Because of the limited available resources the concern has to bear within the resrictions $2x_1 + 3x_2 \leq 2500$ and $x_1 + 2x_2 \leq 1500$. Using *Kuhn-Tucker conditions* method, determine the optimal level of production of *A* and *B* by the concern.

27.6. SADDLE POINT PROBLEMS

In ***Games Theory,*** the *saddle point* of a payoff matrix was defined. Let $\{ v_{ij} \}$ be the payoff matrix for a *two-person zero-sum game.* If v_{i*j} denote the payoff maxima at i^* over the rows and v_{ij*} denote the payoff minima at j^* over the columns, then

$$a_{i*j} \geq a_{i*j*} \geq a_{ij*} \text{ (by } \textit{Theorem } 19.1 \text{ on page 623)}$$

Let f be a real valued function of several variables. Also let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{u} = (u_1, u_2, \ldots, u_m)$, then for these variables the function f is denoted by $f(\mathbf{x}, \mathbf{u})$, $\mathbf{x} \in R^n$, $\mathbf{u} \in R^m$.

Now the saddle point of functions can be defined as follows :

Definition 1 ***(Saddle Point)*** **:** Let $f(\mathbf{x}, \mathbf{u})$ be a function of $\mathbf{x} \in R^n$ and $\mathbf{u} \in R^m$. The function $f(\mathbf{x}, \mathbf{u})$ is said to have a saddle point at $(\mathbf{x}^*, \mathbf{u}^*)$, if and only if,

$$f(\mathbf{x}^*, \mathbf{u}) \geq f(\mathbf{x}^*, \mathbf{u}^*) \geq f(\mathbf{x}, \mathbf{u}^*).$$

Definition 2 ***(Saddle Value Problem)*** **:** Let $\mathbf{x} \in R^n$, $\mathbf{u} \in R^m$. The problem of determining the saddle point value $f(\mathbf{x}^*, \mathbf{u}^*)$ under the constraints $\mathbf{x} \geq 0$ and $\mathbf{u} \geq 0$, is called a *Saddle Value Problem.*

For simplicity, we introduce the notation $f^* = f(\mathbf{x}^*, \mathbf{u}^*)$.

Assuming that $f(\mathbf{x}, \mathbf{u})$ is differentiable partially w.r.t. $\mathbf{x}$ and $\mathbf{u}$, we define the following partial derivatives of $f(\mathbf{x}, \mathbf{u})$ as the column vectors,

$$f_{\mathbf{x}^*} = \left[\frac{\partial f^*}{\partial x_1}, \ldots, \frac{\partial f^*}{\partial x_n}\right], \quad f_{\mathbf{u}^*} = \left[\frac{\partial f^*}{\partial u_1}, \ldots, \frac{\partial f^*}{\partial u_m}\right]$$

where the superscripts (*) indicate the partial derivatives obtained at $(\mathbf{x}^*, \mathbf{u}^*)$.

Theorem 27.6. ***(Necessary Conditions for Non-negative Saddle Point).*** *Let* $\mathbf{x} \in R^n$, $\mathbf{u} \in R^m$ *and f be a function of x and* $\mathbf{u}$. *For a point* $(\mathbf{x}^*, \mathbf{u}^*)$ *to be a non-negative saddle point of* $f(\mathbf{x}, \mathbf{u})$ *it is necessary that* (*i*) $f_{\mathbf{x}^*} \leq 0, f_{\mathbf{x}^*}^T \mathbf{x}^* = 0$, (*ii*) $f_{\mathbf{u}^*} \geq 0, f_{\mathbf{u}^*}^T \mathbf{u}^* = 0$, *for* $\mathbf{x}^* \geq 0$, $\mathbf{u}^* \geq 0$.

Proof. Let $(\mathbf{x}^*, \mathbf{u}^*)$ be a saddle point of $f(\mathbf{x}, \mathbf{u})$ with $\mathbf{x}^* \geq 0$, $\mathbf{u}^* \geq 0$.

Now this theorem can be proved in two parts.

Part 1. We may recall that $f(\mathbf{x}, \mathbf{u}^*)$ is maximized by $\mathbf{x}^*$ at the saddle point. This means that

$$\frac{\partial f^*}{\partial x_j} \leq 0 \text{ (at the saddle point for each } x_j^* \in \mathbf{x}^*)$$

For, if $\partial f^*/\partial x_j > 0$, we may increase the value of that x_j and hence that of f. Further, we may observe that, if $\partial f^*/\partial x_j < 0$, we would prefer a lower value of x_j in order to increase f. Obviously, we would stop decreasing x_j only when the lower limit, say zero, were reached. Thus

(*i*) if $\frac{\partial f^*}{\partial x_j} < 0$, then $x_j^* = 0$, (*ii*) if $\frac{\partial f^*}{\partial x_j} = 0$, then $x_j^* \geq 0$

Since x_j^* was selected arbitrarily, we have

$$\frac{\partial f^*}{\partial x_j} \leq 0 \text{ for } j = 1, 2, \ldots, n,$$

and

$$\frac{\partial f^*}{\partial x_j} < 0 \Rightarrow x_j^* = 0, \quad \frac{\partial f^*}{\partial x_j} = 0 \Rightarrow x_j^* \geq 0.$$

$$f_{\mathbf{x}^*} \leq 0 \text{ and } f_{\mathbf{x}^*}^T \mathbf{x}^* = 0.$$

Part 2. Now we recall that $f(\mathbf{x}^*, \mathbf{u})$ is minimized by $\mathbf{u}^*$ at the saddle point, *i.e.* for each $u_i^* \in \mathbf{u}^*$, $\partial f^*/\partial u_i \geq 0$ at the saddle point. For, if $\partial f^*/\partial u_i < 0$, we may decrease the value of that u_i and hence that of f, which is not possible. When $\partial f^*/\partial u_i > 0$, we may prefer even a lower value of u_i in order to lower f. Obviously, we would stop decreasing u_i only if the lower limit, say zero, were reached. Thus, $\partial f^*/\partial u_i \geq 0$, and

(*i*) $\partial f^*/\partial u_i > 0 \Rightarrow u_i^* = 0$, (*ii*) $\partial f^*/\partial u_i = 0 \Rightarrow u_i^* > 0$.

Since u_i was arbitrary, this holds for all $i = 1, 2, \ldots, m$.

Hence, $f_{\mathbf{u}^*} \geq 0$ and $f_{\mathbf{u}^*}^T \mathbf{u}^* = 0$.

Thus the theorem is completely proved.

Theorem 27.7. ***(Sufficient Condition for Non-negative Saddle Points).*** *Let* $\mathbf{x} \in R^n$, $\mathbf{u} \in R^m$ *and f be function of* **x** *and* **u**. *Then, for a point* $(\mathbf{x}^*, \mathbf{u}^*)$ *to be a non-negative saddle point of* $f(\mathbf{x}, \mathbf{u})$ *it is sufficient that :*

(a) $f_{\mathbf{x}^*} \le 0, f_{\mathbf{x}^*}^T \mathbf{x}^* = 0$ for $\mathbf{x}^* \ge 0$ *(b)* $f_{\mathbf{u}^*} \ge 0, f_{\mathbf{u}^*}^T \mathbf{u}^* = 0$ for $\mathbf{u}^* \ge 0$

(c) $f(\mathbf{x}, \mathbf{u}^*) \le f(\mathbf{x}^*, \mathbf{u}^*) + f_{\mathbf{x}^*}^T (\mathbf{x} - \mathbf{x}^*)$ *(d)* $f(\mathbf{x}^*, \mathbf{u}) \ge f(\mathbf{x}^*, \mathbf{u}^*) + f_{\mathbf{u}^*}^T (\mathbf{u} - \mathbf{u}^*)$.

Proof. Let us suppose that the given conditions (*a*) to (*d*) are satisfied for a point $(\mathbf{x}^*, \mathbf{u}^*)$ to be a non-negative saddle point.

From conditions (c) and (*a*), we have

$$f(\mathbf{x}, \mathbf{u}^*) \le f(\mathbf{x}^*, \mathbf{u}^*) + f_{\mathbf{x}^*}^T \mathbf{x}^*, \text{ for } \mathbf{x}^* \ge 0 \qquad \text{...(27.34)}$$

Further, since $f_{\mathbf{x}^*}^T \mathbf{x}^* = 0, f_{\mathbf{x}^*} \le 0 \Rightarrow f_{\mathbf{x}^*}^T \mathbf{x}^* \le 0$ for $\mathbf{x} \ge 0$.

Thus (27.34) reduces to

$$f(\mathbf{x}, \mathbf{u}^*) \le f(\mathbf{x}^*, \mathbf{u}^*) \text{ for } \mathbf{x}^* \ge 0, \mathbf{x} \ge 0 \qquad \text{...(27.35)}$$

Similarly, *(b)* and *(d)* give

$$f(\mathbf{x}^*, \mathbf{u}) \ge f(\mathbf{x}^*, \mathbf{u}^*) \text{ for } \mathbf{u}^* \ge 0, \mathbf{u} \ge 0 \qquad \text{...(27.36)}$$

Now combining (27.35) and (27.36), we get

$$f(\mathbf{x}, \mathbf{u}^*) \le f(\mathbf{x}^*, \mathbf{u}^*) \le f(\mathbf{x}^*, \mathbf{u})$$

for $\mathbf{x}^*, \mathbf{u}^*, \mathbf{x}, \mathbf{u} \ge \mathbf{0}$.

This proves that $(\mathbf{x}^*, \mathbf{u}^*)$ is a non-negative saddle point of $f(\mathbf{x}, \mathbf{u})$.

Thus, the theorem is completely proved.

27.7. NON-LINEAR PROGRAMMING PROBLEM AND SADDLE POINTS

Let the standard NLPP be :

$$\text{Max. } \mathbf{x} = f(\mathbf{x}), \mathbf{x} \in R^n \qquad \text{...(27.37)}$$

subject to the constraints :

$$g_i(\mathbf{x}) = 0 \ (i = 1, 2, \dots, m), \mathbf{x} \ge 0.$$

We can formulate the associated *Lagrangian* function

$$L(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) - \sum_{i=1}^{m} u_i \, g_i(\mathbf{x}) \qquad \text{...(27.38)}$$

where $\mathbf{u} \in R^m$ are the *Lagrange* multipliers.

Thus the problem of maximizing $f(\mathbf{x})$ is equivalent to that of maximum $L(\mathbf{x}, \mathbf{u})$.

Now we shall be able to establish the relationship between the maximization of $L(\mathbf{x}, \mathbf{u})$ and the saddle function of **x** and **u**. For this, we shall assume that $f(\mathbf{x})$ and all constraint functions $g_i(\mathbf{x})$ possess the partial derivatives.

Theorem 27.8. ***(Necessary Conditions for Saddle Point Correspondence).*** *For* $\mathbf{x}^* \in R^n$ *to be a solution to NLPP (27.37), it is necesary that* $\mathbf{x}^*$ *and some* $\mathbf{u}^*$ *satisfy the conditions of* ***Theorem 27.6*** *for* $f(\mathbf{x}, \mathbf{u}) = L(\mathbf{x}, \mathbf{u})$, *where* $\mathbf{u}^* \in R^m$ *are the Lagrange multipliers as given in (27.38).*

Proof. Let $\mathbf{x}^*$ be a solution to NLPP (27.37). Then, this solution must satisfy the *Kuhn-Tucker* conditions :

(a) $\dfrac{\partial f(\mathbf{x})}{\partial x_j^*} = \sum_{i=1}^{m} u_i^* \dfrac{\partial g_i(\mathbf{x})}{\partial x_j^*} \ (j = 1, 2, \dots, n)$ *(b)* $u_i^* \, g_i(\mathbf{x}) = 0 \ (i = 1, 2, \dots, m)$

(c) $g_i(\mathbf{x}) \le 0 \ (i = 1, 2, \dots, m)$ *(d)* $u_i^* \ge 0 \ (i = 1, 2, \dots, m)$

Now from *(a)*, we may write

$$\frac{\partial f(\mathbf{x})}{\partial x_j^*} - \sum_{i=1}^{m} u_i^* \frac{\partial g_i(\mathbf{x})}{\partial x_j^*} \le 0 \text{ for } j = 1, 2, \dots, n \text{ and}$$

$$\left(\frac{\partial f(\mathbf{x})}{\partial x_j^*} - \sum_{i=1}^{m} u_i^* \frac{\partial g_i(\mathbf{x})}{\partial x_j^*} \right) x_j^* = 0, \text{ for } j = 1, 2, \dots, n \, ; i = 1, 2, \dots, m.$$

or
$$\left[\frac{\partial f(\mathbf{x})}{\partial x_j^*} - \sum_{i=1}^{m} u_i^* \frac{\partial g_i(\mathbf{x})}{\partial x_j^*},\ j = 1, \dots, n\right] \le 0 \qquad \dots(27.39)$$

and
$$\left[\frac{\partial f(\mathbf{x})}{\partial x_j^*} - \sum_{i=1}^{m} u_i^* \frac{\partial g_i(\mathbf{x})}{\partial x_j^*},\ j = 1, \dots, n\right]^T \mathbf{x}^* = 0.$$

Obviously, $\mathbf{x}^* \ge 0$ is a solution to (27.37)

From above condition *(c)* we have.

(i) $-g_i(\mathbf{x}) \ge 0$ for $i = 1, 2, \dots, m$, and from *(b)* we have

(ii) $g_i(\mathbf{x})\, u_i^* = 0$ for $i = 1, 2, \dots, m$.

Also, *(d)* above permits us to write

(iii) $\mathbf{u}^* \ge 0$.

Combining (i), (ii) and (iii), we may write

$$[-g_i(\mathbf{x}),\ i = 1, 2, \dots, m] \ge 0 \text{ and } [-g_i(\mathbf{x}),\ i = 1, \dots, m]^T \mathbf{u}^* = 0 \qquad \dots(27.40)$$

Now, if $f(\mathbf{x}^*, \mathbf{u}^*) \equiv L(\mathbf{u}^*, \mathbf{x}^*)$, then from (27.38) we have

$$\frac{\partial f}{\partial x_j^*} = \frac{\partial f(\mathbf{x})}{\partial x_j^*} - \sum_{i=1}^{m} u_i^* \frac{\partial g_i(\mathbf{x})}{\partial x_j^*} \text{ for } j = 1, 2, \dots, n \quad \text{and} \quad \frac{\partial f}{\partial u_i^*} = -g_i(\mathbf{x}).$$

Thus, (27.39) and (27.40) may be written in the modified form as

$$f_{\mathbf{x}^*}^T \le 0 \text{ and } f_{\mathbf{x}^*}^T \mathbf{x}^* = 0 \text{ for } \mathbf{x}^* \ge 0;\ f_{\mathbf{u}^*}^T \ge 0 \text{ and } f_{\mathbf{u}^*}^T \mathbf{u}^* = 0 \text{ for } \mathbf{u}^* \ge 0.$$

These are essentially the conditions of ***Theorem 27.6.***

Thus the theorem is completely proved.

Theorem 27.9. ***(Sufficient Conditions for Saddle Point Correspondence).*** *Given the NLPP of maximizing* $z = f(\mathbf{x})$, $\mathbf{x} \in R^n$, *subject to the constraints* $g_i(\mathbf{x}) = 0$ *and* $\mathbf{x} \ge 0$. *For* $\mathbf{x}^*$ *to be a solution to this NLPP, it is sufficient that* $\mathbf{x}^*$ *and* $\mathbf{u}^*$ *satisfy the conditions of* ***Theorem 27.7*** *when* $f(\mathbf{x}, \mathbf{u}) = L(\mathbf{x}, \mathbf{u})$, *where* $\mathbf{u} \in R^m$.

Proof. The *Lagrangian* function can be constructed as

$$L(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) - \sum_{i=1}^{m} u_i^* g_i(\mathbf{x}),\ \mathbf{u} \in R^m.$$

Let $\quad L(\mathbf{x}, \mathbf{u}) \equiv f(\mathbf{x}, \mathbf{u})$ for $\mathbf{x} \in R^n$, $\mathbf{u} \in R^m$. That is, $f(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) - \sum_{i=1}^{m} u_i\, g_i(\mathbf{x})$.

We assume that the following conditions of ***Theorem 27.7*** are satisfied :

(a) $f_{\mathbf{x}^*} \le 0, f_{\mathbf{x}^*}^T \mathbf{x}^* = 0$ for $\mathbf{x}^* \ge 0$, *(b)* $f_{\mathbf{u}^*} \ge 0, f_{\mathbf{u}^*}^T \mathbf{u}^* = 0$ for $\mathbf{u}^* > 0$

(c) $f(\mathbf{x}, \mathbf{u}^*) \le f(\mathbf{x}^*, \mathbf{u}^*) + f_{\mathbf{x}^*}^T(\mathbf{x} - \mathbf{x}^*)$, and *(d)* $f(\mathbf{x}^*, \mathbf{u}) \ge f(\mathbf{x}^*, \mathbf{u}^*) + f_{\mathbf{u}^*}^T(\mathbf{u} - \mathbf{u}^*)$.

If we are able to show that $\mathbf{x}^*$ maximizes $f(\mathbf{x}, \mathbf{u}^*)$ for $\mathbf{x}^* \ge 0$, then our theorem will be proved.

It follows from the conditions *(a)* to *(d)* above that

$$f(\mathbf{x}, \mathbf{u}^*) \le f(\mathbf{x}^*, \mathbf{u}^*) \le f(\mathbf{x}^*, \mathbf{u}) \text{ for } \mathbf{x}^* \ge 0,\ \mathbf{u}^* \ge 0.$$

This shows that $\mathbf{x}^*$ maximizes $f(\mathbf{x}, \mathbf{u}^*)$ for $\mathbf{x}^* \ge 0$ and hence maximizes $L(\mathbf{x}, \mathbf{u}^*)$ for $\mathbf{x}^* \ge 0$. Consequently, $\mathbf{x}^*$ is the optimum solution to the given NLPP.

Thus the theorem is completely proved.

EXAMINATION PROBLEMS

1. If $F(\mathbf{x}, \mathbf{y})$ has a non-negative saddle point $(\mathbf{x}_0, \mathbf{y}_0)$, prove that $\mathbf{x}_0$ is a solution of the following convex programming problem : Min. $f(\mathbf{x})$, $\mathbf{x} \in R^n$ subject to the constraints : $g_i(\mathbf{x}) \le 0$ $(1 \le i \le m)$, $\mathbf{x} \ge 0$ where $f(\mathbf{x})$, $g_i(\mathbf{x})$ are convex functions.
2. Let $\mathbf{x}^*$ be a local minimum to the NLPP :
   Min. $f(\mathbf{x})$, subject to the constraints : $g_i(\mathbf{x}) \ge 0$, $i = 1, 2, \dots, m$; $h_j(\mathbf{x}) \ge 0$, $j = 1, 2, \dots, n$; $\mathbf{x} \ge 0$.

Show that a necessary condition for a differentiable function f to have an unconstrained local minima or maxima at $\mathbf{x}^*$ is $\nabla f(\mathbf{x}^*) = 0$.

3. Show that the optimum value of the objective function of the NLPP :

$$\text{Min } \mathbf{z} = \sum_{j=1}^{n} (c_j/x_j)\text{, subject to the constraints : } \sum_{j=1}^{n} \alpha_j x_j = b,\ x_j \geq 0\ (j = 1, 2, \ldots, n)$$

is given by $$x^* = \frac{1}{b}\left[\sum_{j=1}^{m} \sqrt{\alpha_j c_j}\right]^2,$$ where α_j, c_j and b are positive constraints.

4. Formulate the *Kuhn-Tucker* necessary conditins for the NLPP : Max. $\mathbf{z} = f(\mathbf{x})$, subject to the constraints :
$g_i(\mathbf{x}) \geq 0$, $g_j(\mathbf{x}) \geq 0$, $g_k(\mathbf{x}) \geq 0$ $i = 1, 2, \ldots, n_1$, $j = n_1 + 1, \ldots, n_2$, $k = 1, 2, \ldots, n_3$, and $\mathbf{x} \geq 0$.

5. Write the *Kuhn-Tucker* conditions for the following problem :
Minimize $f(\vec{X}) = x_1^2 + x_2^2 + x_3^2$, subject to $2x_1 + x_2 - x_3 \leq 0$, $1 - x_1 \leq 0$, $2 - x_2 \leq 0$, $-x_3 \leq 0$.
Also solve the problem. **[I.A.S. (Maths.) 82]**

6. Explain what is meant by *Tucker-Kuhn* conditions. **[I.A.S. (Maths.) 88]**

✷✷✷

28

NON-LINEAR PROGRAMMING PROBLEM
(Formulation and Graphical Method)

28.1. INTRODUCTION

In chapter 1 of *'**Linear Programming and The Theory of Games**'*, we have introduced the linear programming problem which can be reviewed as

$$\textit{Maximize } \mathbf{z} = \sum_{j=1}^{n} c_j x_j \qquad \text{...(28.1)}$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, 2, \ldots, m \qquad \text{...(28.2)}$$

$$\text{and} \qquad x_j \ge 0 \quad \text{for } j = 1, 2, \ldots, n. \qquad \text{...(28.3)}$$

The term *'non-linear programming'* usually refers to the problem in which the objective function (28.1) becomes *non-linear*, or one or more of the constraint inequalities (28.2) have *non-linear* relationship or both. In actual practice, such situation occurs if a purely linear relationship may not exist in the profit or cost function when the production levels vary. For example, production costs and revenues vary non-linearly with the scale of operations.

28.2. PRACTICAL SITUATIONS OF NON-LINEARITIES

The situations in which non-linearities are built into the programming models are :

(1) **Gasoline blending.** In the model of blending gasoline from so-called refinary raw stocks usually contains non-linear constraints relating to each blend, octane relating, since this quality characteristic varies non-linearly with the amount of *tetraethyl* lead added to the mix.

(2) **Sales revenue.** In marketing, we usually observe that—the lower a product's price, the greater the sales quantity. Therefore, sales revenue does not vary proportionately with price. Consequently, this phenomenon reflects the objective function to be non-linear. For example, let $S(p)$ represent the sales quantity as a function of price p, then $p\,S(p)$ is the associated sales revenue. If the sales quantity function is linear, say $S(p) = ap + b$, over the range of interest for p, then the sales revenue component in the objective function is quadratic $(ap^2 + bp)$, where p is the decision variable.

(3) **Portfolio selection problem.** Let x_j represent the proportion of available funds to be allocated to *security j*. Assume that a_j is the actual (random) gain per unit invested in *security j*, and α_j is the associated expected gain. Further suppose that we stipulate b to be the lowest acceptable expected gain per unit invested in the entire portfolio.

The consideration of risk is introduced by means of the objective function involving a quadratic form :

$$\text{Min } \mathbf{z} = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j, \qquad \text{...(28.4)}$$

where $\qquad \sigma_{ij} \equiv E[a_i - \alpha_i] \cdot [a_j - \alpha_j] \qquad$ ...(28.5)

represents the covariance of gain between *securities i* and *j*, subject to the constraints :

$$\sum_{j=1}^{n} x_j = 1 \text{ and } x_j \geq 0 \text{ for } j = 1, 2, \ldots, n \quad \ldots(28.6)$$

$$\sum_{j=1}^{n} \alpha_j x_j \geq b, \quad \ldots(28.7)$$

where the left-hand side of (28.7) represents the expected gain per unit invested, because the expected value of a sum ***equals*** the sum of expected values.

There may be additional constraints on the composition of the portfolio, several time periods, and other measures of risk.

(4) Safety-Stock Inventory Levels. Safety-stocks are usually maintained to accomodate weekly fluctuations in sales. One approach used to solve such a multiperiod model is to let the safety stock level for an item be a function of both, its forecasted sales quantity and the fraction of capacity utilization implied by this forecast. For example, let c be the weekly capacity available to produce an item, s the item's forecasted average weekly sales, and ns the item's safety stock level, where n denotes the number of week's sales depending on the capacity utilization factor s/c. To explain this, suppose management has established the formula for n to be $n = m + f(s/c)$; then the resultant safety-stock level is a quadratic function $[ms + (f/c)\, s^2]$ of the items forecasted average weekly sales. Such level may appear in many of the planning model's constraints as well as in the objective function.

28.3. FORMULATION OF NON-LINEAR PROGRAMMING PROBLEMS

To explain the method of formulation of non-linear programming problems, we consider the following example.

Example 1. (Production Allocation Problem). *A manufacturing company produces two products : Radios and TV sets. The sales-price relationship for these two products are given below :*

Products	*Quantity Demanded*	*Unit Price*
Radios	1500—5p	p
TV Sets	3800—10q	q

The total cost functions for these two products are given by $200x + 0.1x^2$ *and* $300y + 0.1y^2$ *respectively. The production takes place on two assembly lines. Radio sets are assembled on Assembly line I and TV sets are assembled on Assembly line II. Because of the limitations of the assembly line capacities, the daily production is limited to no more than 80 radio sets and 60 TV sets. The production of both types of products requires electronic components. The production of each of these sets require five units and six units of electronic equipment respectively. The electronic components are supplied by another manufacturer, and the supply is limited to 600 units per day. The company has 160 employees, i.e. the labour supply amounts to 460 man-days. The production of one unit of radio set requires 1 man-day of labour, whereas 2 man-days of labour are required for a TV set. How many units of radio and TV sets should the company produce in order to maximize the total profit ? Formulate the problem as non-linear programming problem.*

Formulation. Let x and y quantities of radio sets and TV sets be manufactured by the firm, respectively.

As given in the problem,

$$x = 1500 - 5p \text{ or } p = 300 - 0.2x \quad \ldots(28.8)$$

$$y = 3800 - 10q \text{ or } q = 380 - 0.1y \quad \ldots(28.9)$$

If the total production cost of amounts x and y is denoted by c_1 and c_2, respectively; then it is also given that

$$c_1 = 200x + 0.1x^2 \quad \ldots(28.10)$$

$$c_2 = 300y + 0.1y^2 \quad \ldots(28.11)$$

Thus, the revenue on radio sets becomes px and on TV sets qy. Therefore, the total revenue R is given by

$$R = px + qy. \quad \ldots(28.12)$$

Substituting the values of p and q from equations (28.8) and (28.9) in (28.12), we get

$$R = (300 - 0.2x)\, x + (380 - 0.1y)\, y \quad \text{or} \quad R = 300x - 0.2x^2 + 380y - 0.1y^2$$

Now the total profit P is obtained by subtracting the total cost $(c_1 + c_2)$ from the total revenue R.

Therefore,

$$P = R - (c_1 + c_2) \quad \text{or} \quad P = (300x - 0.2x^2 + 380y - 0.1y^2) - (200x + 0.1x^2 + 300y + 0.1y^2)$$

or
$$P = 100x - 0.3x^2 + 80y - 0.2y^2. \qquad ...(28.13)$$

Thus, we observe that the objective function obtained above is non-linear.

In this problem, the available resources affect the production. Since more than 80 radio sets cannot be assembled on assembly line I and 60 *TV* sets on assembly line II per day, we have the constraints :

$$x \le 80 \text{ and } y \le 60$$

Another constraint of daily requirement of the electronic components is $5x + 6y \le 600$.

Also, the number of available employees is restricted to 160 man-days. Therefore, we have one more constraint : $x + 2y \le 160$.

Since the production of negative quantities has no meaning, we must have the non-negativity restrictions : $x \ge 0, y \ge 0$.

Thus, finally, the complete formulation of the problem becomes :

$$\text{Max. } P = 100x + 80y - 0.3x^2 - 0.2y^2, \text{ subject to the restrictions :}$$
$$5x + 6y \le 600, \quad x + 2y \le 160, \quad x \le 80, y \le 60, \text{ and } x, y \ge 0.$$

Because of the non-linearity of the objective function, the problem is of non-linear programming category.

EXAMINATION PROBLEMS

1. A company manufactures two products *A* and *B*. It takes 30 minutes to process one unit of product *A* and 15 minutes for each unit of *B* and the maximum machine time available is 35 hours per week. Products *A* and *B* requie 2 kgs and 3 kgs of raw material per unit respectively. The available quantity of raw material is envisaged (considered) to be 180 kgs per week.

   The products *A* and *B* which have unlimited marked potential sell for Rs. 200 and Rs. 500 per unit respectively. If the manufacturing costs for products *A* and *B* are $2x^2$ and $3y^2$ respectively, find how much of each product should be produced per week, where

   x = quantity of product *A* to be produced, y = quantity of product *B* to be produced.

   [**Ans.** Max. $\mathbf{z} = (200 - 2x^2) + (500 - 2y^2)$; subject to $0.5x + 0.25y \le 35$, $2x + 3y \le 180$; and $x \ge 0$, $y \ge 0$]

2. The total profit of a restaurant was found to depend mostly on the amount of money spent on advertising and the quality of the preparation of the food (measured in terms of the salaries paid to the chefs). In fact the manager of the restaurant found that if he pays his chefs x Rs. per hour and spends y Rs. a week on advertising, the restaurant's weekly profit (in Rupees) will be

   $$\mathbf{z} = 412x + 806y - x^2 - y^2 - xy.$$

   What hourly wages should the manager pay his chefs and how much should he spend on advertising so as to maximize the restaurant's profit ?

28.4. GENERAL NON-LINEAR PROGRAMMING PROBLEM

The mathematical formulation of *general non-linear programming problem* may be expressed as follows :

$$\textit{Max. } (\text{or } \textit{Min.})\ \mathbf{z} = C(x_1, x_2, \ldots, x_n), \text{ subject to the constraints :}$$

$$a_1(x_1, x_2, \ldots, x_n)\ \{\le, = \text{ or } \ge\}\ b_1$$
$$a_2(x_1, x_2, \ldots, x_n)\ \{\le, = \text{ or } \ge\}\ b_2$$
$$\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots$$
$$a_m(x_1, x_2, \ldots, x_n)\ \{\le, = \text{ or } \ge\}\ b_m,$$

and
$$x_j \ge 0, j = 1, 2, \ldots, n,$$

where either $C(x_1, x_2, \ldots, x_n)$ or some $a_i(x_1, x_2, \ldots, x_n)$, $i = 1, .., m$; or both are non-linear.

In matrix notation, the general non-linear programming problem may be written as follows :

$$\textit{Max. } (\text{or } \textit{Min.})\ \mathbf{z} = C(\mathbf{x}), \text{ subject to the constraints :}$$
$$a_i(\mathbf{x})\ \{\le, = \text{ or } \ge\}\ b_i, i = 1, 2, \ldots, m$$
$$\text{and} \quad \mathbf{x} \ge 0,$$

where either $C(\mathbf{x})$ or some $a_i(\mathbf{x})$ or both are non-linear in $\mathbf{x}$.

28.5. CANONICAL FORM OF NON-LINEAR PROGRAMMING PROBLEM

The canonical form of non-linear programming problem can be viewed as follows :

$$Max.\ \mathbf{z} = C(x_1, x_2, \dots, x_n),\ subject\ to$$
$$a_i(x_1, x_2, \dots, x_n) \le 0,\ i = 1, 2, \dots, m$$
$$x_j \ge 0, j = 1, 2, \dots, n,$$

where at least one of the functions $C(x_1, x_2, \dots, x_n)$ and $a_i(x_1, x_2, \dots, x_n)$, $i = 1, 2, \dots, m$, is non-linear.

In matrix form, it may be defined as : Max. $\mathbf{z} = C(\mathbf{x})$, subject to $\mathbf{a}(\mathbf{x}) \le 0$.

The non-negativity conditions $\mathbf{x} \ge 0$ are summed to be the part of the given set of constraints. It is further assumed that at least one of the functions $C(\mathbf{x})$ and $\mathbf{a}(\mathbf{x})$ is non-linear. Furthermore, for the purpose of presentation, these functions are assumed to be continuously differentiable.

Unlike linear programming, no general algorithms are available for dealing with non-linear models. The reason for this is mainly the irregular behaviour of the non-linear functions. Although, a large number of algorithms have been developed for the solution of non-linear programming problem, even then there is a need of developing a more efficient solution procedure.

In the present and subsequent chapters we shall discuss some of the elementary type of solution techniques.

Q. 1. Give a formulation of the general Mathematical Programming Problem and obtain the linear programming as a special case of the same.
2. What is a non-linear programming problem ?

28.6. GRAPHICAL SOLUTION

In a linear programming problem, the optimal solution was usually obtained at one of the corner (extreme) points of the convex region generated by the constraints and the objective function of the problem. But, it is not necessary to find the solution at a corner or edge of the feasible region of non-linear programming problem. The following numerical examples will make the method clear.

Example 2. (***Linear objective function, Non-linear constraints***) *Solve graphically the following problem :*

$$\text{Max. } \mathbf{z} = 2x_1 + 3x_2,\ subject\ to\ x_1^2 + x_2^2 \le 20,\ x_1x_2 \le 8,\ and\ x_1, x_2 \ge 0.$$

Verify the Kuhn-Tucker conditions hold for the maxima you obtain. **[Banasthali (M.Sc.) 93]**

Solution. Let $\mathbf{Ox_1}$ and $\mathbf{Ox_2}$ be the set of rectangular cartesian coordinate axes in the plane of the paper. Obviously, the feasible region will lie in the first quadrant only, because $x_1 \ge 0$, $x_2 \ge 0$.

Now we plot the curves $x_1^2 + x_2^2 = 20$ and $x_1x_2 = 8$.

We observe that $x_1^2 + x_2^2 = 20$ represents a circle of radius $\sqrt{(20)}$ with its centre at the origin; and $x_1x_2 = 8$ represents a rectangular hyperbola whose asymptotes are the coordinate axes. Solving the equations $x_1^2 + x_2^2 = 20$ and $x_1x_2 = 8$, we find the coordinates of the intersection of these two curves as $B(4, 2)$ and $D(4, 2)$ in Fig. 28.1.

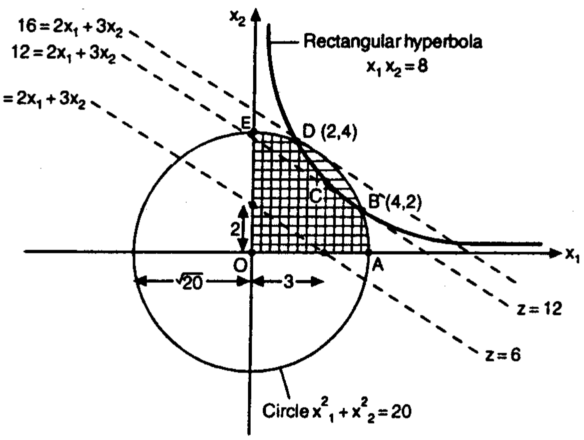

Fig. 28.1

As shown in the above figure, the points (x_1, x_2) lying in the first quadrant shaded by the horizontal lines satisfy the constraints $x_1^2 + x_2^2 \le 20$, $x_1 \ge 0$, $x_2 \ge 0$; while the points (x_1, x_2) lying in the area shaded by vertical lines satisfy the constraints $x_1x_2 = 8, x_1 \geq 0, x_2 \geq 0$. Thus the desired solution point (x_1, x_2) may be somewhere in the *non-convex feasible region OABCDE* shaded by both the horizontal and vertical lines.

Now we are in search of such a point (x_1, x_2) in the feasible region which maximize the objective function $z = 2x_1 + 3x_2$ and lines in the convex part of the region. The desired point can be immediat- ely found by moving parallel to the objec- tive line $2x_1 + 3x_2 = c$ for some constant $z = c$. For example, we go on moving parallel to the objective line $2x_1 + 3x_2 = 6$ (for $c = 6$, say) away from the origin so long as the line $c = 2x_1 + 3x_2$ touches the extreme boundary of the feasible region. In this problem, boundary point $D(2, 4)$ gives the maximum value of z. Hence the graphical solution of the problem is finally obtained as $x_1 = 2, x_2 = 4$, max. $z = 16$.

Verification of Kuhn-Tucker Conditions :

We can also verify that the optimum solution obtained above satisfies the *Kuhn-Tucker* conditions.

Here we are given that

$$f(\mathbf{x}) = 2x_1 + 3x_2, \quad g_1(\mathbf{x}) = x_1x_2 - 8, \quad g_2(\mathbf{x}) = x_1^2 + x_2^2 - 20,$$

and the problem is that of maximizing $f(\mathbf{x})$ subject to the constraints $g_1(\mathbf{x}) \leq 0, g_2(\mathbf{x}) \leq 0$, and $\mathbf{x} \geq 0$. The *Kuhn-Tucker conditions* for this maximizing *NLPP* are :

$$\frac{\partial f(\mathbf{x})}{\partial x_j} = \lambda_1 \frac{\partial g_1(\mathbf{x})}{\partial x_j} + \lambda_2 \frac{\partial g_2(\mathbf{x})}{\partial x_j}, \text{ for } j = 1, 2$$

$$\lambda_i\, g_i(\mathbf{x}) = 0, \quad g_i(\mathbf{x}) \leq 0, \quad \lambda_i \geq 0 \text{ for } i = 1, 2;$$

where λ_1, λ_2 are *Lagrangian* multipliers.

These conditions are thus written as :

$$(a) \begin{cases} 2 = \lambda_1 x_2 + 2\lambda_2 x_1 \\ 3 = \lambda_1 x_1 + 2\lambda_2 x_2 \end{cases} \quad (b) \begin{cases} \lambda_1 [x_1x_2 - 8] = 0 \\ \lambda_2 [x_1^2 + x_2^2 - 20] = 0 \end{cases} \quad (c) \begin{cases} x_1x_2 - 8 \leq 0 \\ x_1^2 + x_2^2 - 20 \leq 0 \end{cases} \quad (d)\ \lambda_1 \geq 0, \lambda_2 \geq 0.$$

If the point (2, 4) satisfies these conditions, then we must have from *(a)* $\lambda_1 = 1/6$ and $\lambda_2 = 1/3$. From $(x_1, x_2) = (2, 4)$ and $(\lambda_1, \lambda_2) = (1/6, 1/3)$, it is clear that the conditions *(b), (c)* and *(d)* are satisfied. Hence the optimum solution obtained by graphical method satisfies the *Kuhn-Tucker* conditions for a maxima.

Example 3. (Non-linear objective function and linear constraints).

Minimize $z = x_1^2 + x_2^2$, *subject to the constraints* :

$$x_1 + x_2 \geq 4, \quad 2x_1 + x_2 \geq 5, \text{ and } x_1, x_2 \geq 0.$$

Solution. Let Ox_1 and Ox_2 be the set of rectangular cartesian coordinate axes in the plane of the paper. Because of the non-negativity restrictions $x_1 \geq 0, x_2 \geq 0$, the feasible region will lie in the first quadrant only.

We now plot the lines $x_1 + x_2 = 4$ and $2x_1 + x_2 = 5$. The constraint $x_1 + x_2 \geq 4$ is satisfied by all the points lying in the region shaded by vertical lines, while the constraint $2x_1 + x_2 \geq 5$ is satisfied by all the points lying in the region shaded by horizontal lines only. As shown in the figure below, the region shaded by both the vertical and horizontal lines is unbounded convex feasible region X_2ABCX_1. But, our object is to search for a point (x_1, x_2) which gives a minimum value of $x_1^2 + x_2^2$ and lies in the convex region. The desired point will be a point of the region at which a side of the convex region is tangent to the circle. Now we can proceed as follows :

The gradient of the tangent to the circle $x_1^2 + x_2^2 = k$ (where $z = k$, say) is obtained by differentiating the equation of this circle. That is,

$$2x_1 + 2x_2 \frac{dx_2}{dx_1} = 0 \quad \text{or} \quad \frac{dx_2}{dx_1} = -\frac{x_1}{x_2} \qquad \text{...(i)}$$

Gradient of the line $x_1 + x_2 = 4$ is -1 and the gradient of the line $2x_1 + x_2 = 5$ is -2.

If the line $x_1 + x_2 = 4$ is the tangent to the circle $x_1^2 + x_2^2 = k$, then

$$\frac{dx_2}{dx_1} = -\frac{x_1}{x_2} = -1 \quad \text{or} \quad x_1 = x_2. \qquad \text{...(ii)}$$

If the line $2x_1 + x_2 = 5$ is the tangent to the circle $x_1^2 + x_2^2 = k$, then

$$\frac{dx_2}{dx_1} = -\frac{x_1}{x_2} = -2 \quad \text{or} \quad x_1 = 2x_2. \qquad \text{...(iii)}$$

Therefore, the point at which the line $x_1 + x_2 = 4$ is tangent to the cricle is obtained by solving the equations $x_1 + x_2 = 4$ and $x_1 = x_2$ to give us $x_1 = 2, x_2 = 2$.

Similarly, the point at which the line $2x_1 + x_2 = 5$ touches the circle is obtained by solving the equations $2x_1 + x_2 = 5$ and $x_1 = 2x_2$ to give us $x_1 = 2, x_2 = 1$. This indicates that :

(i) the line $x_1 + x_2 = 4$ touches the cirlce $x_1^2 + x_2^2 = k$ at the point (2, 2); and

(ii) the line $2x_1 + x_2 = 5$ touches the circle $x_1^2 + x_2^2 = k$ at the point (2, 1).

But, the point (2, 1) lies outside the convex region and hence it will not give us the desired solution. Thus, obviously, the desired solution is given by other point (2, 2) which gives us the min. $\mathbf{z} = 2^2 + 2^2 = 8$.

Ans. $x_1 = 2, x_2 = 2$, min. $\mathbf{z} = 8$.

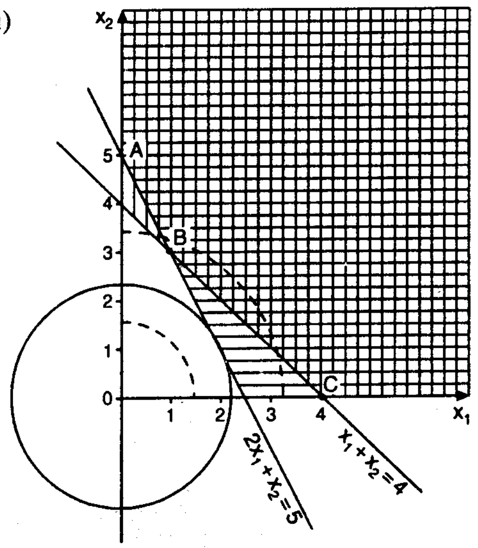

Fig. 28.2.

EXAMINATION PROBLEMS

Solve the following non-linear programming problems graphically :

1. Maximize $\mathbf{z}_1 = 8x_1^2 + 2x_2^2$ and Minimize $\mathbf{z}_2 = x_1 + x_2$, subject to the constraints :
$x_1 x_2 \geq 8,\ x_1^2 + x_2^2 \leq 9,\ x_1 \leq 2$, and $x_1, x_2 \geq 0$.

2. Min. $\mathbf{z} = (x_1 - 1)^2 + (x_2 - 2)^2$ subject to the constraints :
$0 \leq x_1 \leq 2,\ 0 \leq x_1 \leq 1$
[Ans. $x_1 = 0,\ x_2 = 1$, min. $\mathbf{z} = 2$]

3. Min. $\mathbf{z} = (x_1 - 4)^2 + (x_2 - 4)^2$ subject to the constraints :
$x_1 + x_2 \leq 6,\ x_1 - x_2 \leq 1$
$2x_1 + x_2 \geq 6,\ \tfrac{1}{2} x_1 - x_2 \geq -4$
$x_1, x_2 \geq 0$.

4. Max. $\mathbf{z} = 2x_1 - x_1^2 + x_2$ subject to the constraints :
$2x_1 + 3x_2 \leq 6$
$2x_1 + x_2 \leq 4$
$x_1, x_2 \geq 0$.

5. (i) Maximize $\mathbf{z} = x_1$, subject to the constraints :
$(1 - x_1)^3 - x_2 \geq 0$
$x_1, x_2 \geq 0$.

(ii) Maximize $\mathbf{z} = x_1$, subject to the constraints :
$(3 - x_1)^3 - (x_2 - 2) \geq 0$
$(3 - x_1)^3 + (x_2 - 2) \geq 0$
$x_1, x_2 \geq 0$.

Also show that the *Kuhn-Tucker* necessary conditions for a maxima do not hold. What do you conclude ?
[Ans. (i) $x_1 = 1,\ x_2 = 0$, max. $\mathbf{z} = 1$. Constraint qualification is not satisfied]
(ii) $x_1 = 3,\ x_2 = 2$, max. $\mathbf{z} = 3$. Constraint qualification is not satisfied]

6. Min. $\mathbf{z} = (x_1 - 2)^2 + (x_2 - 1)^2$, subject to the constraints :
$-x_1^2 + x_2 \geq 0$
$-x_1 - x_2 + 2 \geq 0$
$x_1, x_2 \geq 0$.
[Ans. $x_1 = 1,\ x_2 = 1$, min. $\mathbf{z} = 1$]

7. Min. $\mathbf{z} = 4(x_1 - 6)^2 + 6(x_2 - 2)^2$ subject to the constraints :
$0.5x_1 + x_2 \leq 4$
$3x_1 + x_2 \leq 15$
$x_1 + x_2 \geq 1$
[Ans. $x_1 = 129/29,\ x_2 = 48/29$, min. $\mathbf{z} = 7800/841$]

8. Maximize $\mathbf{z} = x_1 + 2x_2$, subject to the constraints :
$$x_1^2 + x_2^2 \leq 1,\ 2x_1 + x_2 \leq 2, \text{ and } x_1, x_2 \geq 0.$$

9. Solve graphically the following non-linear programming problem (NLPP).
Maximize $Z = 8x_1 - x_1^2 + 8x_2 - x_2^2$
subject to the constraints : $x_1 + x_2 \leq 12,\ x_1 - x_2 \leq 4$ and $x_1, x_2 \geq 0$. **[Agra (M.Sc. Math.) 99]**

✱✱✱

29

QUADRATIC PROGRAMMING
(Wolfe's and Beale's Method)

29.1. INTRODUCTION

In ***Unit* 2,** we considered optimization techniques for linear programming problems only. Because of linearity, we were able to develop a very efficient algorithm (called the *simplex method*) for handling such problems. Unlike the linear programming case, no such general algorithms exist for solving all non-linear programming problems. However, for problems with certain suitable structures, efficient algorithms have been developed. Also, it is often possible to convert the given non-linear problem into one in which these structures become visible.

The general mathematical programming problem (GMPP) can be defined as the problem of determining $\mathbf{x} \in R^n$ so as to optimize (maximize or minimize) the objective function

(*a*) $\mathbf{z} = f(\mathbf{x})$

subject to the constraints :

(*b*) $g_i(\mathbf{x})$ ($\leq$, = or $\geq$) b_i , $i = 1, 2, ..., m$

and (*c*) $\mathbf{x} \geq 0$,

where $f(\mathbf{x})$ and $g_i(\mathbf{x})$ are the real valued functions of $\mathbf{x}$ for $i = 1, 2, \ldots , m$ and b_i'*s* are real constants.

If may be observed that the above GMPP reduces to the *general linear programming problem if*

(*a*) $f(\mathbf{x})$ and (*b*) $g_i(\mathbf{x})$ for $i = 1, 2, \ldots , m$ are all linear in **x.**

In such cases, the problem can be solved by *Simplex Method* or it's modifications as discussed so far (in ***Unit* 2**).

As defined in the preceding chapter, the GMPP reduces to general non-linear programming problem (GNLPP) if

either $f(\mathbf{x})$ and $g_i(\mathbf{x})$ for some or all $i = 1, 2, \ldots , m$

or $f(\mathbf{x})$ only

or $g_i(\mathbf{x})$ only for some or all $i = 1, 2, \ldots , m$

are non-linear in **x.** Further, these functions are assumed to be continuously differentiable.

Unlike linear programming, the optimal solution to a NLPP can be found anywhere on the boundary of the feasible region and even at some interior point of it. In recent years, several methods of NLPP have been developed. But, an efficient simplex like technique for a GNLPP is still required to be developed. A few available techniques for some particular cases of GNLPP shall be discussed in this book.

A well known quadratic programming model, dealing with the problem of selecting an investment portfolio that will yield a given expected total return with a minimum variance was developed by **Markowitz.** The problem often referred to as the portfolio selection model, assumes that the investor wishes to maximize his anticipated returns while he considers variance of return as undersirable.

Supppose the total fund available to an investor is B. There are n channels of investment. The expected return of the ith source is m_i, the variance of the return of the ith type of investment is σ_i^2, the covariance between the return of ith and jth investment is σ_{ij}.

Hence, if an amount x_i ($i = 1, 2, \ldots, m$) is invested on the ith type of investment then the expected retrun is

$$\sum_{i=1}^{n} m_i x_i$$

The variance of the overall return of the investment is

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j$$

where $\sigma_{ij} = \sigma_i^2$. As higher return and lower variance are desirable quantities from the point of view of an investor, the objective function can be taken as

$$f = \sum_{i=1}^{n} m_i x_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j$$

The total amount the investor can spend is B. Hence

$$\sum_{i=1}^{n} x_i = B.$$

The investor wants f to be maximized subject to the above linear constraint. The function f can be shown to be a ***concave function*** of the variables. The solution of the problem, thus, can be achieved using quadrantic programming methods. Apart from the constraint given above, the problem can accomodate other linear constraints involving the decision variables.

Q. Enumerate the investment portfolio selection problem as a quadratic programming problem. **[IGNOU 97 (Dec.)]**

29.2. KUHN-TUCKER CONDITIONS : NON-NEGATIVE CONSTRAINTS

So far we have obtained the necessary conditions for a point $\mathbf{x}^* \in R^n$ to be a relative maximum of $f(\mathbf{x})$ subject to the constraints $g_i(\mathbf{x}) \le 0$, $i = 1, 2, \ldots, m$, $\mathbf{x} \ge 0$. These conditions (called the *Kuhn-Tucker Conditions*) were obtained by changing each inequality constraint to an equation by adding a ***squared slack variable*** s_i^2, imposing the first-order conditions (for maxima) on the first-partial derivative of the *Lagrangian* function, and then simplifying the result. The following conditions are obtained :

$$(a)\ \frac{\partial f(\mathbf{x})}{\partial x_j} = \sum_{i=1}^{m} \frac{\lambda_i \partial g_i(\mathbf{x})}{\partial x_j},\quad (b) - \lambda_i g_i(\mathbf{x}) = 0,\quad (c)\ g_i(\mathbf{x}) \le 0,\quad (d)\ \lambda_i \ge 0\quad (j = 1, 2, \ldots, n;\ i = 1, 2, \ldots, m)$$

It has been noticed that the non-negativity constraints ($\mathbf{x} \ge 0$) were completely ignored while obtaining these conditions. However, we always kept in mind to discard all such solutions not satisfying the condition $\mathbf{x} \ge 0$.

Now at this stage, we can consider the non-negativity constraints as one of the constraints and then derive the *Kuhn-Tucker conditions* for the resulting problem.

The problem may be restated as follows :

$$\textit{Max. } \mathbf{z} = f(\mathbf{x}),\ \mathbf{x} \in R^n,\ \textit{subject to the constraints}:$$
$$g_i(\mathbf{x}) \le 0 \textit{ and } -\mathbf{x} \le 0\quad (\textit{for } i = 1, \ldots, m).$$

Obviously, there are $m + n$ inequality constraints and so we introduce $(m + n)$ squared slack variables $s_1^2, s_2^2, \ldots, s_m^2, s_{m+1}^2, \ldots, s_{m+n}^2$ in the respective inequalities in order to convert them to the following equations :

$$g_i(\mathbf{x}) + s_i^2 = 0 \qquad \text{for} \qquad i = 1, 2, \ldots, m$$
$$-x_j + s_{m+j}^2 = 0 \qquad \text{for} \qquad j = 1, 2, \ldots, n.$$

To obtain the necessary conditions for maximum of $f(\mathbf{x})$, the associated *Lagrangian* function becomes :

$$L(\mathbf{x}, \lambda, \mathbf{s}) = f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i [g_i(\mathbf{x}) + s_i^2] - \sum_{j=1}^{n} \lambda_{m+j} [-x_j + s_{m+j}^2]$$

where $\mathbf{s} = (s_1, s_2, \ldots, s_{m+n})$, and $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{m+n})$ are the *Lagrangian* multipliers. The *Kuhn-Tucker 7conditions* are then given by

$$\frac{\partial L}{\partial x_j} = \frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i(\mathbf{x})}{\partial x_j} + \lambda_{m+j} = 0, \qquad \text{for } j = 1, \dots, n$$

$$\frac{\partial L}{\partial \lambda_i} = -(g_i + s_i^2) = 0, \qquad \text{for } i = 1, 2, \dots, m$$

$$\frac{\partial L}{\partial \lambda_{m+j}} = -(-x_j + s_{m+j}^2) = 0, \qquad \text{for } j = 1, 2, \dots, n$$

$$\frac{\partial L}{\partial s_i} = -2\lambda_i s_i = 0, \qquad \text{for } i = 1, 2, \dots, m.$$

$$\frac{\partial L}{\partial s_{m+j}} = 2\lambda_{m+j}\, s_{m+j} = 0, \qquad \text{for } j = 1, 2, \dots, n.$$

The simplified form of these *Kuhn-Tucker conditions* for the problems : max $\mathbf{z} = f(\mathbf{x})$, $g_i(x) \le 0$, $\mathbf{x} \ge 0$, may be presented in the following form :

(a) $\frac{\partial f}{\partial x_j} = \sum_{i=1}^{m} \lambda_i \frac{\partial g_i(\mathbf{x})}{\partial x_j} - \lambda_{m+j}$ $(j = 1, 2, \dots, n)$; *(b)* $\lambda_i\, [g_i(\mathbf{x})] = 0$ $(i = 1, \dots, m)$;

(c) $-\lambda_{m+j} x_j = 0$ $(j = 1, 2, \dots, n)$, *(d)* $g_i(\mathbf{x}) \le 0$ $(i = 1, \dots, m)$;

(e) $\lambda_j, \lambda_{m+j}, x_j \ge 0$ $(i = 1, 2, .., m ; j = 1, 2, \dots, n)$.

Also, note that these conditions are sufficient if $f(\mathbf{x})$ is concave and all $g_i(\mathbf{x})$ are convex in **x**. Likewise, the *Kuhn-Tucker conditions* for GNLPP (min. case) are sufficient also if $f(\mathbf{x})$ is convex and all $g_i(\mathbf{x})$ are concave in **x**.

29.3. GENERAL QUADRATIC PROGRAMMING PROBLEM [I.A.S. (Main) 88, 86]

Quadratic programming deals with the non-linear programming problem of maximizing (or minimizing) the quadratic objective function subject to a set of linear inequality constraints. The *general quadratic programming* problem can be defined as follows :

Definition. Let $\mathbf{x}^T$ and $\mathbf{c} \in R^n$, and **Q** be a symmetric $n \times n$ real matrix. Then, the problem of maximizing (*i.e.*, determining **x**) so as to maximize

$$f(\mathbf{x}) = \mathbf{cx} + \tfrac{1}{2}\,\mathbf{x}^T \mathbf{Q}\,\mathbf{x}, \quad \text{subject to the constraints :} \qquad ...(29.1)$$

$$A\mathbf{x} \le \mathbf{b} \text{ and } \mathbf{x} \ge 0, \qquad ...(29.2)$$

where $\mathbf{b}^T \in R^m$ and A be $m \times n$ real matrix, is called a *General Quadratic Programming Problem* (GQPP).

The function $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ defines a quadratic form (see Ch. 2 in ***Unit* 1**) with **Q** being a symmetric matrix. The quadratic form $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ is said to be *positive-definite* if $\mathbf{x}^T\mathbf{Q}\mathbf{x} > 0$ for $\mathbf{x} \ne \mathbf{0}$ and *positive-semi-definite* if $\mathbf{x}^T\mathbf{Q}\mathbf{x} \ge 0$ for all **x** such that there is one $\mathbf{x} \ne 0$ satisfying $\mathbf{x}^T\mathbf{Q}\mathbf{x} = 0$. Similarly, $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ is said to be negative-definite and *negative-semi-definite* if $-\mathbf{x}^T\mathbf{Q}\mathbf{x}$ is positive-definite and positive-semi-definite respectively. The fucntion $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ is assumed to be negative-definite in the maximization case, and positive definite in the minimization case. The constraints are assumed to be linear which ensures a convex solution space.

It may be easily verified that :

(i) if $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ is *positive-semi-definite* (or *negative-semi-definite*), then it is *convex* (or *concave*) in **x** over all of R^n, and

(ii) if $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ is *positive-definite* (or *negative-definite*), then it is *strictly convex* (or *strictly concave*) in **x** over all of R^n.

These results help us to decide whether the quadratic objective function $f(\mathbf{x})$ is concave (convex).

Note. For easiness, we may write :

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T, \; \mathbf{c} = (c_1, c_2, \dots, c_n), \; \mathbf{b} = (b_1, b_2, \dots, b_n)^T,$$

$$A = \begin{bmatrix} a_{11} \dots\dots\dots a_{1n} \\ \vdots \qquad\qquad \vdots \\ a_{m1} \dots\dots\dots a_{mn} \end{bmatrix}_{m \times n} \qquad Q = \begin{bmatrix} c_{11} \dots\dots\dots c_{1n} \\ \vdots \qquad\qquad \vdots \\ c_{n1} \dots\dots\dots c_{nn} \end{bmatrix}_{n \times n}$$

We now turn our attention to the problems which are slightly more general than linear programming problems. In such type of problems, we optimize a quadratic function subject to linear constraints. The most well-behaved non-linear algorithm is called quadratic programming. In this algorithm, the objective function is convex (minimization) or concave (maximization) and all the constraints are linear.

The solution to this problem is secured by the direct application of the *Kuhn-Tucker* necessary conditions (see *Theorem 29.4*). Since **z** is strictly convex (concave) and the solution space is a convex set, these necessary conditions (as proved in ***Section* 29.5**) are also sufficient for a global (absolute) optima.

We shall now treat the quadratic programming problem for the maximization case. It is easy to change the formulation to the minimization

Note. (i) If Q is null in (29.1.), we have the standard linear programming problem.

(ii) The prime (′) can also be used for the transpose of matrix, instead of superscript (^{T}).

29.4. TO CONSTRUCT KUHN-TUCKER CONDITIONS FOR QUADRATIC PROGRAMMING PROBLEM

We now construct the *Kuhn-Tucker conditions* for maximization problems as formulated in the above section.

Let us consider a *Quadratic Programming Problem* in the form :

$$\textit{Maximize } \mathbf{z} = f(\mathbf{x}) = \sum_{j=1}^{n} c_j x_j + \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} c_{jk} x_j x_k$$

subject to the constraints :

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \text{ and } x_j \ge 0, \qquad (i = 1, 2, \dots, m, \; j = 1, 2, \dots, n)$$

where $c_{jk} = c_{kj}$ for all j and k (for Q is symmetric); and where $b_i \ge 0$ for $i = 1, \dots, m$.

Introducing slack variables q_i^2 and r_j^2, the problem becomes :

$$\textit{Max. } \mathbf{z} = f(\mathbf{x}) = \sum_{j=1}^{n} c_j x_j + \tfrac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} c_{jk} x_j x_k$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j + q_i^2 = b_i, \text{ for } i = 1, 2, \dots, m$$

$$-x_j + r_j^2 = 0, j = 1, 2, \dots, n. \qquad \text{...(29.3)}$$

We shall now proceed to construct the *Lagrangian function*

$$L(\mathbf{x}, \mathbf{q}, \mu, \lambda, \mathbf{r}) = f(\mathbf{x}) - \left[\sum_{i=1}^{m} \lambda_i \sum_{j=1}^{n} (a_{ij} x_j + q_i^2 - b_i) \right] - \sum_{j=1}^{n} \mu_j (-x_j + r_j^2) \qquad \text{...(29.4)}$$

Forming the necessary conditions, we obtain

$$\frac{\partial L(\cdot)}{\partial x_j} = \frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_i a_{ij} + \mu_j = 0, j = 1, \dots, n \qquad \text{...(29.5)}$$

$$\sum_{j=1}^{n} a_{ij} x_j + q_i^2 - b_i = 0 \qquad \text{...(29.6)}$$

$$\mu_j x_j = 0 \qquad \text{...(29.7)}$$

$$\mathbf{Ax} \le \mathbf{b} \qquad \text{...(29.8)}$$

and finally **x** , λ and μ must all be non-negative.

Rewriting the equation (29.5) we get

$$\frac{\partial L}{\partial x_j} = \left[c_j + \frac{1}{2} \left(2 \sum_{k=1}^{n} c_{jk} x_k \right) \right] - \sum_{i=1}^{m} \lambda_i a_{ij} + \mu_j = 0, j = 1, 2, \dots, n.$$

Letting $q_i^2 = s_i \ge 0$, above equation becomes

$$\mu_j + c_j + \sum_{k=1}^{n} c_{jk} x_k - \sum_{i=1}^{m} \lambda_i a_{ij} = 0, j = 1, 2, \ldots, n \quad \ldots(29.9)$$

$$\mathbf{Ax + Is = b, x \geq 0, s \geq 0, \lambda \geq 0, \mu \geq 0,}$$

and finally $\lambda_i s_i = 0$, $i = 1, \ldots, m$; $\mu_j x_j = 0, j = 1, \ldots, n$.

One important thing to be noted here is that except for the final conditions $\lambda_i s_i = 0 = \mu_j x_j$., the remaining equations are linear functions in $\mathbf{x}$, λ, μ and $\mathbf{s}$. The problem thus becomes equivalent to finding the solution to a set of linear equations which also satisfies the additional conditions $\lambda_i s_i = 0 = \mu_j x_j$. Because $f(\mathbf{x})$ is strictly concave and the solution space is convex, the feasible solution satisfying all these conditions must give the optimum solution directly.

Wolfe suggested a solution procedure for this problem using the ordinary simplex method with slight modification as given in the following section.

Q. 1. Derive Kuhn-Tucker necessary conditions for an optimal solution to a quadratic programming problem.

2. Obtain the Kuhn-Tucker conditions for a solution of the problem : Max $f(\mathbf{x}) = P^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T C \mathbf{x}$, subject to the constraints : $\mathbf{Ax = b}$ and $\mathbf{x} \geq 0$

Wolfe's Method

29.5. WOLFE'S MODIFIED SIMPLEX METHOD

Let the quadratic programming problem be :

$$\textit{Maximize } \mathbf{z} = f(\mathbf{x}) = \sum_{j=1}^{n} c_j x_j + \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} c_{jk} x_j x_k$$

subject to the constraints : $\sum_{j=1}^{n} a_{ij} x_j \leq b_i$, $x_j \geq 0$ $(i = 1, \ldots, m, j = 1, \ldots, n)$

where $c_{jk} = c_{kj}$ for all j and k, $b_i \geq 0$ for all $i = 1, 2, \ldots, m$.

Also, assume that the quadratic form $\sum_{j=1}^{n} \sum_{k=1}^{n} c_{jk} x_j x_k$ be *negative semi-definite.*

Then, the *Wolfe's* iterative procedure may be outlined in the following steps :

Step 1. First, convert the inequality constraints into equations by introducing slack-variables q_i^2 in the ith constraint $(i = 1, \ldots, m)$ and the slack variables r_j^2 in the jth non-negativity constraint $(j = 1, 2, \ldots, n)$.

Step 2. Then, construct the Lagrangian function

$$L(\mathbf{x}, \mathbf{q}, \mathbf{r}, \lambda, \mu) = \mathbf{f}(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i \left[\sum_{j=1}^{n} a_{ij} x_j - b_i + q_i^2 \right] - \sum_{j=1}^{n} \mu_j [-x_j + r_j^2]$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, $\mathbf{q} = (q_1^2, \ldots, q_m^2)$, $\mathbf{r} = (r_1^2, r_2^2, \ldots, r_n^2)$, and

$$\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m), \mu = (\mu_1, \mu_2, \ldots, \mu_n).$$

Differentiating the above function 'L' partially with respect to the components of $\mathbf{x}$, $\mathbf{q}$, $\mathbf{r}$, λ, μ, and equating the first order partial derivatives to zero, we derive *Kuhn-Tucker conditions* from the resulting equations.

Step 3. *Wolfe* (1959) suggested to introduce the non-negative artificial variable v_j, $j = 1, 2, \ldots, n$ in the *Kuhn-Tucker conditions*

$$c_j + \sum_{k=1}^{n} c_{jk} x_k - \sum_{i=1}^{m} \lambda_i a_{ij} + \mu_j = 0$$

for $j = 1, 2, \ldots, n$ and to construct an objective function

$$\mathbf{z_v} = v_1 + v_2 + \ldots + v_n.$$

Step 4. In this step, obtain the initial basic feasible solution to the following linear programming problem :

$$\text{Min. } z_v = v_1 + v_2 + \ldots + v_n ,$$

subject to the constraints :

$$\sum_{k=1}^{n} c_{jk} x_k - \sum_{i=1}^{m} \lambda_i a_{ij} + \mu_j + v_j = -c_j \quad (j = 1, 2, \ldots, n)$$

$$\sum_{j=1}^{n} a_{ij} x_j + q_i^2 = b_i \quad (i = 1, 2, \ldots, m),$$

$$v_j, \lambda_i, \mu_j, x_j \geq 0 \quad (i = 1, \ldots, m ; j = 1, \ldots, n)$$

and satisfying the complementary slackness condition :

$$\sum_{j=1}^{n} \mu_j x_j + \sum_{i=1}^{m} \lambda_i s_i = 0, \text{ (where } s_i = q_i^2)$$

or

$$\lambda_i s_i = 0 \text{ and } \mu_j x_j = 0 \quad (\text{for } i = 1, \ldots, m ; j = 1, \ldots, n).$$

Step 5. Now, apply two-phase simplex method in the usual manner to find an optimum solution to the LP problem constructed in *Step 4*. The solution must satisfy the above complementary slackness condition.

Step 6. The optimum solution thus obtained in *Step* 5 gives the optimum solution of given QPP also.

Important remarks on *Wolfe's* method :

1. If the quadratic programming problem is given in the minimization form, then convert it into maximization one by suitable modifications in $f(\mathbf{x})$ and the '$\geq$' constraints.
2. With the exceptional condition of complementary slackness, the problem constructed in *Step 4* is exactly the linear programming problem. So we only need to modify the *simplex algorithm* to include the complementary slackness conditions. Thus, while deciding to introduce s_i $(= q_i^2)$ we must first ensure that : *(i)* either λ_i is not in the solution or *(ii)* λ_i will be removed when s_i enters. This additional check is not difficult to perform within the simplex routine and can be successfully performed.
3. The solution of the above system is obtained by using *Phase I* of simplex method. Since our aim (of course) is to obtain a feasible solution, the solution does not require the consideration of *Phase II*. The only necessary thing is to maintain the condition $\lambda_i s_i = 0 = \mu_j x_j$ all the time. This implies that if λ_i is in the basic solution with positive value, then s_i cannot be basic with positive value. Similarly, μ_j and x_j cannot be positive simultaneously.
4. It should be observed that *Phase I* will end in the usual manner with the sum of all artificial variables *equal to zero* only if the feasible solution to the problem exists.

Q. 1. What is Quadratic programming ? Explain Wolfe's method of solving it.

2. Mention briefly the Wolfe's algorithm for solving a quadratic programming problem given in the usual notations :
Max. $z = f(\mathbf{x}) + \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x}$, *s.t.* $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$.

3. Discuss Wolfe's method for solving a quadratic programming problem. **[Delhi (OR). 90]**

4. Describe a quadratic programming problem and outline a method of solving it. **[IAS (Main) 97]**

29.5–1. Illustrative Examples on Wolfe's Method

Example 1. *Apply Wolfe's method for solving the quadratic programming problem :*

$$\textit{Max. } z_x = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2, \textit{ subject to}$$

$$x_1 + 2x_2 \leq 2, \textit{ and } x_1, x_2 \geq 0.$$

Solution.

Step 1. First, we write all the constraint inequalities with '$\leq$' sign as follows :

$$x_1 + 2x_2 \leq 2, \ -x_1 \leq 0, -x_2 \leq 0.$$

$$\begin{aligned} x_1 + 4x_2 + q_1^2 &= 4 \\ x_1 + x_2 \qquad + q_2^2 &= 2 \\ -x_1 \qquad\qquad + r_1^2 &= 0 \\ -x_2 \qquad\qquad + r_2^2 &= 0. \end{aligned}$$

Step 2. **To construct the *Lagrangian* function.** The *Lagrangian* function now becomes :

$L(x_1, x_2, q_1, q_2, r_1, r_2, \lambda_1, \lambda_2, \mu_1, \mu_2)$

$= (2x_1 + 3x_2 - 2x_1^2) - \lambda_1 (x_1 + 4x_2 + q_1^2 - 4) - \lambda_2 (x_1 + x_2 + q_2^2 - 2) - \mu_1 (-x_1 + r_1^2) - \mu_2 (-x_2 + r_2^2)$

The objective function $\mathbf{z} = 2x_1 + 3x_2 - 2x_1^2$ is concave in x_1, x_2 because the term $-2x_1^2$ represents a *negative semi-definite* quadratic form. Consequently, the maxima of L (.) will be the maxima of z.

Step 3. Here, we get the *Kuhn-Tucker conditions* as follows :

$$\frac{\partial L}{\partial x_1} = 2 - 4x_1 - \lambda_1 - \lambda_2 + \mu_1 = 0, \quad \frac{\partial L}{\partial x_2} = 3 - 4\lambda_1 - \lambda_2 + \mu_2 = 0$$

Defining, $q_1^2 = s_1$, $q_2^2 = s_2$, we have

$$\lambda_1 s_1 = \lambda_2 s_2 = 0, \quad \mu_1 x_1 = \mu_2 x_2 = 0, \quad x_1 + 4x_2 + s_1 = 4, \quad x_1 + x_2 + s_2 = 2,$$

and finally $x_1, x_2, s_1, s_2, \lambda_1, \lambda_2, \mu_1, \mu_2 \geq 0$.

Step 4. **To construct the modified L.P. problem.**

Now, introducing the artificial variables v_1 and v_2, the modified L.P. problem becomes :

$$\text{Max. } \mathbf{z}_v = -v_1 - v_2, \text{ subject to },$$

$$\begin{aligned} 4x_1 + 0x_2 + \lambda_1 + \lambda_2 - \mu_1 \qquad + v_1 \qquad &= 2 \\ 0x_1 + 0x_2 + 4\lambda_1 + \lambda_2 \qquad - \mu_2 \qquad + v_2 &= 3 \\ x_1 + 4x_2 \qquad\qquad\qquad\qquad + s_1 &= 4 \\ x_1 + 2x_2 \qquad\qquad\qquad\qquad + s_2 &= 2 \end{aligned}$$

$$x_1, x_2, \lambda_1, \lambda_2, v_1, v_2, \mu_1, \mu_2, s_1, s_2 \geq 0,$$

satisfying the complementary slackness condition

$$\mu_1 x_1 + \mu_2 x_2 + \lambda_1 s_1 + \lambda_2 s_2 = 0$$

where we have replaced q_1^2 by s_1 and q_2^2 by s_2.

The optimum solution to the above L.P. problem can be obtained by *two-phase simplex method* as below :

Writing the above set of equations in matrix form, we get

$$\begin{bmatrix} 4 & 0 & 1 & 1 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 4 & 1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \lambda_1 \\ \lambda_2 \\ \mu_1 \\ \mu_2 \\ v_1 \\ v_2 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 2 \end{bmatrix}$$

Step 5. **To obtain the initial table for *Phase I*.**

		$c_j \rightarrow$	0	0	0	0	0	0	−1	−1	0	0	
B	$\mathbf{c_B}$	$\mathbf{x_B}$	$\mathbf{x_1}$	$\mathbf{x_2}$	λ_1	λ_2	μ_1	μ_2	$\mathbf{v_1}$	$\mathbf{v_2}$	$\mathbf{s_1}$	$\mathbf{s_2}$	
v_1	−1	2	[4]	0	1	1	−1	0	1	0	0	0	
v_2	−1	3	0	0	4	1	0	−1	0	1	0	0	
s_1	0	4	1	4	0.	0	0	0	0	0	1	0	
s_2	0	2	1	2	0	0	0	0	0	0	0	1	
	$\mathbf{z}_v = -5$		−4	0	−5	−2	1	1	0	0	0	0	$\leftarrow \Delta_j$
			↑						↓				

This table indicates that *any one* of x_1 , λ_1 and λ_2 may enter the basis. But since s_1 , s_2 are in the basis, λ_1 , λ_2 cannot enter the basis ($\because \lambda_1 s_1 = 0$ and $\lambda_2 s_2 = 0$). Hence x_1 enters the basis.

First Iteration. Introducing x_1 into the basis and droping v_1 from it, we get

		$c_j \to$	0	0	0	0	0	−1	−1	0	0
B	c_B	x_B	x_1	x_2	λ_1	λ_2	μ_1	μ_2	v_2	s_1	s_2
x_1	0	½	1	0	¼	¼	−¼	0	0	0	0
v_2	−1	3	0	0	4	1	0	−1	1	0	0
s_1	0	7⁄2	0	4	−¼	−¼	¼	0	0	1	0
s_2	0	3⁄2	0	[2]	−¼	−¼	¼	0	0	0	1
	$z_v = -3$		0	0	−4	−1	0	−1	0	0	0
				↑							↓

This table now indicates that either λ_1 or λ_2 enters the basis, but these cannot enter the basis because s_1 , s_2 are in the basis. Now since μ_2 is not in the basis so that $s_2 = 0$ and therefore, we can enter x_2 into the basis ($\because \mu_2 x_2 = 0$).

Second Iteration. We introduce x_2 into the basis and drop s_2.

		$c_j \to$	0	0	0	0	0	0	−1	0	0
B	c_B	x_B	x_1	x_2	λ_1	λ_2	μ_1	μ_2	v_2	s_1	s_2
x_1	0	½	1	0	¼	¼	−¼	0	0	0	0
v_2	−1	3	0	0	4	1	0	−1	1	0	0
s_1	0	½	0	0	¼	[¼]	−¼	0	0	1	−2
x_2	0	¾	0	1	−⅛	−⅛	⅛	0	0	0	½
	$z_v = -3$		0	0	−4	−1	0	1	0	0	0
						↑				↓	

Again λ_1 cannot enter the basis since s_1 is in the basis. Also, since s_2 is not in the basis, λ_2 enters the basis.

Third Iteration. We introduce λ_2 into the basis, and drop s_1 from it.

		$c_j \to$	0	0	0	0	0	0	−1	0	0	
B	c_B	x_B	x_1	x_2	λ_1	λ_2	μ_1	μ_2	v_2	s_1	s_2	
x_1	0	0	1	0	0	0	0	0	0	−1	2	
v_2	−1	1	0	0	3	0	1	−1	1	−4	8	
λ_2	0	2	0	0	1	1	−1	0	0	4	−8	
x_2	0	1	0	1	0	0	0	0	0	½	−½	
	$z_v = -1$		0	0	−3	0	−1	1	0	4	−8	$\leftarrow \Delta_j$
			↓								↑	

Fourth Iteration. We introduce s_2 into the basis and drop x_1.

		$c_j \to$	0	0	0	0	0	0	−1	0	0	
B	c_B	x_B	x_1	x_2	λ_1	λ_2	μ_1	μ_2	v_2	s_1	s_2	
s_2	0	0	½	0	0	0	0	0	0	−½	1	
v_1	−1	1	−4	0	[3]	0	1	−1	1	0	0	
λ_2	0	2	4	0	1	1	−1	0	0	0	0	
x_2	0	1	¼	1	0	0	0	0	0	1/2	0	
	$z_v = -1$		4	0	−3	0	−1	1	0	0	0	$\leftarrow \Delta_j$
					↑				↓			

We compute, $\Delta_1 = (0, -1, 0, 0)\ (1, 0, 0, 0) - 0 = 0, \Delta_2 = 0, \Delta_3 = -3, \Delta_4 = -1$

$$\Delta_5 = 0, \Delta_6 = 1, \Delta_7 = 1, \Delta_8 = \Delta_9 = \Delta_{10} = 0.$$

We now enter second iteration.

Step 7. Second Iteration. Since $\mu_2 = 0$, x_2 can be introduced with s_1 as the leaving variable. Thus, we get Table 29.7.

Second Iteration Table 29.7.

		$c_j \rightarrow$	0	0	0	0	0	0	−1	−1	0	0	
B	c_B	x_B	x_1	x_2	λ_1	λ_2	μ_1	μ_2	v_1	v_2	s_1	s_2	
x_1	0	1	[1]	0	1	1	−1/2	0	1/2	0	0	0	
v_2	−1	1	0	0	3	1	0	−1	0	[1]	0	0	
x_2	0	4/3	0	[1]	−2/3	−2/3	1/3	0	−1/3	0	1/3	0	
s_2	0	2/3	0	0	−4/3	−4/3	2/3	0	−2/3	0	−1/3	[1]	
	z = −1		0	0	−3	−1	0	1	1	0	0	0	$\leftarrow \Delta_j$
					↑					↓			

We compute Δ_j's as before.

Step 8. Third Iteration.

Since $s_1 = 0$, λ_1 can be introduced with v_2 as the leaving variable. Thus, we shall get Table 29.8.

Third Iteration Table 29.8.

		$c_j \rightarrow$	0	0	0	0	0	0	−1	−1	0	0	
B	c_B	x_B	x_1	x_2	λ_1	λ_2	μ_1	μ_2	v_1	v_2	s_1	s_2	
x_1	0	2/3	1	0	0	2/3	−1/2	1/3	1/2	−1/3	0	0	
λ_1	0	1/3	0	0	1	1/3	0	−1/3	0	1/3	0	0	
x_2	0	14/9	0	1	0	−4/9	1/3	−2/9	−1/3	2/9	1/3	0	
s_2	0	10/9	0	0	0	−8/9	2/3	−4/9	−2/3	4/9	−1/3	1	
	z = 0		0	0	0	0	0	0	1	1	0	0	$\leftarrow \Delta_j$

Since both v_1 and v_2 are out of the basic solution, the computation is now complete. The optimal solution is :

$$x_1^* = 2/3,\ x_2^* = 14/9,\ \lambda_1 = 1/3,\ \lambda_2 = 0,\ \mu_1 = \mu_2 = 0,\ v_1 = v_2 = 0,\ s_1 = 0, \text{ and } s_2 = 10/9,$$

which satisfies the complementary slackness conditions ($\mu_1 x_1 = 0$, $\mu_2 x_2 = 0$, $\lambda_1 s_1 = 0$, $\lambda_2 s_2 = 0$) and the restriction on the signs of the *Lagrange multipliers* ($\lambda_1, \lambda_2, \mu_1, \mu_2$).

The max. value of the objective function can be computed from the original objective function (not from modified). Thus,

$$z_x^* = 2\ (2/3) + 14/9 - 4/3 = 14/9 \cdot$$

Example 3. *Use Wolfe's method to solve the quadratic programming problem :*

$$Max.\ z = 2x_1 + 3x_2 - 2x_1^2, \textit{ subject to the constraints :}$$

$$x_1 + 4x_2 \le 4,\ x_1 + x_2 \le 2, \textit{ and } x_1, x_2 \ge 0.$$

Solution :

Step 1. First we convert the inequality constraints into equations by using slack variables q_1^2 and q_2^2, respectively. Also, treating the non-negativity conditions : $x_1 \ge 0, x_2 \ge 0$ as inequality constraints we convert them into equations by using the slack variables r_1^2 and r_2^2, respectively. Thus the given problem can be converted in the following form :

$$\text{Max. } z = 2x_1 + 3x_2 - 2x_1^2, \text{ subject to the constraints :}$$

$$\begin{aligned} &\text{Max. } z_x = 2x_1 + x_2 - x_1^2, \text{ subject to} \\ &2x_1 + 3x_2 + q_1^2 = 6 \\ &2x_1 + x_2 + q_2^2 = 4 \\ &-x_1 + r_1^2 = 0 \\ &-x_2 + r_2^2 = 0. \end{aligned}$$

Step 3. To obtain the *Kuhn-Tucker* conditions, we construct the *Lagrange* function
$L(x_1, x_2, q_1, q_2, r_1, r_2, \lambda_1, \lambda_2, \mu_1, \mu_2)$,
$= (2x_1 + x_2 - x_1^2) - \lambda_1 (2x_1 + 3x_2 + q_1^2 - 6) - \lambda_2 (2x_1 + x_2 + q_2^2 - 4) - \mu_1 (-x_1 + r_1^2) - \mu_2 (-x_2 + r_2^2)$.
The necessary and sufficient conditions are

$$\frac{\partial L}{\partial x_1} = 2 - 2x_1 - 2\lambda_1 - 2\lambda_2 + \mu_1 = 0, \qquad \frac{\partial L}{\partial x_2} = 1 - 3\lambda_1 - \lambda_2 + \mu_2 = 0.$$

Now defining : $s_1 = q_1^2$, $s_2 = q_2^2$, we have $\lambda_1 s_1 = 0$, $\lambda_2 s_2 = 0$, $\mu_1 x_1 = 0$, $\mu_2 x_2 = 0$.

Also, $\quad 2x_1 + 3x_2 + s_1 = 6$
$\quad 2x_1 + x_2 + s_2 = 4,$

and finally $\quad x_1, x_2, s_1, s_2, \lambda_1, \lambda_2, \mu_1, \mu_2 \geq 0.$

Step 4. To construct the modified linear programming problem.
Now introducing the artificial variable v_1 and v_2, the modified linear programming problem becomes :

Max. $z_v = -v_1 - v_2$, *subject to*

$$\begin{aligned} &2x_1 + 0x_2 + 2\lambda_1 + 2\lambda_2 - \mu_1 + v_1 = 2 \\ &0x_1 + 0x_2 + 3\lambda_1 + 1\lambda_2 - \mu_2 + v_2 = 1 \\ &2x_1 + 3x_2 + s_1 = 6 \\ &2x_1 + x_2 + s_2 = 4. \end{aligned}$$

with all variables non-negative and $\mu_1 x_1 = 0$, $\mu_2 x_2 = 0$, $\lambda_1 s_1 = 0$, $\lambda_2 s_2 = 0$.

Step 5. To construct initial table for *Phase* I.

Starting Table 29.5.

		$c_j \rightarrow$	0	0	0	0	0	0	−1	−1	0	0	
B. Var.	c_B	x_B	x_1	x_2	λ_1	λ_2	μ_1	μ_2	v_1	v_2	s_1	s_2	
v_1	−1	2	2	0	2	2	−1	0	1	0	0	0	
v_2	−1	1	0	0	3	1	0	−1	0	1	0	0	
s_1	0	6	2	3	0	0	0	0	0	0	1	0	
s_2	0	4	2	1	0	0	0	0	0	0	0	1	
	$z = -3$		−2	0	−5	−3	1	1	0	0	0	0	$\leftarrow \Delta_j$
			↑						↓				

We compute $\quad \Delta_1 = \mathbf{c_B}\, \mathbf{x_1} - c_1 = (-1, -1, 0, 0)\,(2, 0, 2, 2) - 0 = -2$, etc.

We now start first iteration in the next step.

Step 6. First iteration. Since $\mu_1 = 0$, x_1 is introduced into the basic solution with v_1 as the leaving variable λ_1 and λ_2 cannot be introduced because s_1 and s_2 are basic variables. (Since $\mu_2 = 0$, x_2 can also be introduced with s_1 as the leaving variable). This gives the transformed Table 29.6.

First Iteration Table 29.6.

		$c_j \rightarrow$	0	0	0	0	0	0	−1	−1	0	0	
B. var.	c_B	x_B	x_1	x_2	λ_1	λ_2	μ_1	μ_2	v_1	v_2	s_1	s_2	
x_1	0	1	[1]	0	1	1	−½	0	1/2	0	0	0	
v_2	−1	1	0	0	3	1	0	−1	0	[1]	0	0	
s_1	0	4	0	3	−2	−2	1	0	−1	0	[1]	0	
s_2	0	2	0	1	−2	−2	1	0	−1	0	0	[1]	
	$z_v = -1$		0	0	−3	−1	0	1	1	0	0	0	$\leftarrow \Delta_j$
				↑							↓		

Step 2. Now, introducing the slack variables q_1^2, r_1^2, r_2^2, our problem becomes of the form :

$$\text{Max. } z_x = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

subject to

$$x_1 + 2x_2 + q_1^2 = 2$$
$$-x_1 + r_1^2 = 0$$
$$-x_2 + r_2^2 = 0.$$

Step 3. Here to obtain the *Kuhn-Tucker* conditions, we construct the *Lagrange* function

$$L(x_1, x_2, q_1, r_1, r_2, \lambda_1, \mu_1, \mu_2)$$
$$= (4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2) - \lambda_1 (x_1 + 2x_2 + q_1^2 - 2) - \mu_1 (-x_1 + r_1^2) - \mu_2 (-x_2 + r_2^2).$$

The necessary and sufficient conditions are :

$$\frac{\partial L}{\partial x_1} = 4 - 4x_1 - 2x_2 - \lambda_1 + \mu_1 = 0, \frac{\partial L}{\partial x_2} = 6 - 2x_1 - 4x_2 - 2\lambda_1 + \mu_2 = 0.$$

Defining $s_1 = q_1^2$, we have $\lambda_1 s_1 = 0$, $\mu_1 x_1 = 0$, $\mu_2 x_2 = 0$.

Also $x_1 + 2x_2 + s_1 = 2$, and finally, $x_1, x_2, s_1, \lambda_1, \mu_1, \mu_2 \geq 0$.

Step 4. ***To construct the modified linear programming problem.***

Now, introducing the artificial variables v_1 and v_2, the modified linear programming problem becomes :

Max. $z_v = -v_1 - v_2$ *subject to*

$$4x_1 + 2x_2 + \lambda_1 - \mu_1 + v_1 = 4$$
$$2x_1 + 4x_2 + 2\lambda_1 - \mu_2 + v_2 = 6$$
$$x_1 + 2x_2 + s_1 = 2$$

where all variables are non-negative and $\mu_1 x_1 = 0$, $\mu_2 x_2 = 0$, $\lambda_1 s_1 = 0$.

Now, all these constraint-equations can be written in matrix form as follows :

$$\begin{bmatrix} 4 & 2 & 1 & -1 & 0 & 1 & 0 & 0 \\ 2 & 4 & 2 & 0 & -1 & 0 & 1 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \lambda_1 \\ \mu_1 \\ \mu_2 \\ v_1 \\ v_2 \\ s_1 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 2 \end{bmatrix}.$$

Step 5. ***To construct initial table of Phase I.***

The initial *Table* 29.1 for *Phase I* is obtained by introducing the artificial variables v_1 and v_2 as above.

Starting Table 29.1*

		$c_j \rightarrow$	0	0	0	0	0	−1	−1	0	
Basic Var.	c_B	x_B	x_1	x_2	λ_1	μ_1	μ_2	v_1	v_2	s_1	
v_1	−1	4	4	2	1	−1	0	[1]	0	0	
v_2	−1	6	2	4	2	0	−1	0	1	0	
s_1	0	2	1	2	0	0	0	0	0	1	
	$z_v = -10$		−6	−6	−3	1	1	0	0	0	$\leftarrow \Delta_j$
			↑					↓			

Here, we compute

$$\Delta_1 = (-1, -1, 0)(4, 2, 1) - 0 = -6, \Delta_2 = (-1, -1, 0)(2, 4, 2) - 0 = -6, \text{ etc.}$$

We now enter first iteration in the next step.

Note. The notations B, c_B, x_B, c_j, Δ_j are used for modified problem obtained in step 4 (not for original problem).

Step 6. First Iteration. Since $\mu_1 = 0$, x_1 is introduced into the basic solution with v_1 as the leaving variable.

*The column headings x_1, x_2, ... in Table 29.1 represent the vectors associated with the variables.

Notice that λ_1 cannot enter because s_1 is the basic variable. This gives the following transformed table by our usual rules of transformation.

First Iteration Table 29.2

	$c_j \rightarrow$		0	0	0	0	0	−1	−1	0	
Basic var.	c_B	x_B	x_1	x_2	λ_1	μ_1	μ_2	v_1	v_2	s_1	
x_1	0	1	[1]	1/2	1/4	−1/4	0	1/4	0	0	
v_2	−1	4	0	3	3/2	1/2	−1	−1/2	1	0	
s_1	0	1	0	3/2	−1/4	1/4	0	−1/4	0	[1]	
	$z_v = -4$		0	−3	−3/2	−1/2	1	3/2	0	0	$\leftarrow \Delta_j$
				↑						↓	

We compute $\Delta_2 = (0, -1, 0)(1/2, 3, 3/2) - 0 = -3$, $\Delta_3 = (0, -1, 0)(1/4, 3/2, -1/4) - 0 = -3/2$, etc.

Step 7. Second Iteration. Since $\mu_2 = 0$, x_2 is introduced into the basic solution with s_1 as the leaving vector. By usual rules of matrix transformation, we get next improved table.

Second Iteration Table 29.3

	$c_j \rightarrow$		0	0	0	0	0	−1	−1	0
Basic Var.	c_B	x_B	x_1	x_2	λ_1	μ_1	μ_2	v_1	v_2	s_1
x_1	0	2/3	[1]	0	1/3	−1/3	0	1/3	0	−1/3
v_2	−1	2	0	0	2	0	−1	0	[1]	−2
x_2	0	2/3	0	[1]	−1/6	1/6	0	−1/6	0	2/3
	$z = -2$		0	0	−2↑	0	1	1	0↓	2←

Step 8. Third Iteration Since $s_1 = 0$, hence λ_1 can be introduced into the basic solution.

Third Iteration Table 29.4.

	$c_j \rightarrow$		0	0	0	0	0	−1	−1	0	
B	c_B	x_B	x_1	x_2	λ_1	μ_1	μ_2	v_1	v_2	s_1	
x_1	0	1/3	[1]	0	0	−1/3	1/6	1/3	−1/6	0	
λ_1	0	1	0	0	[1]	0	−1/2	0	1/2	−1	
x_2	0	5/6	0	[1]	0	1/6	−1/12	−1/6	1/12	1/2	
	$z_v = 0$		0	0	0	0	0	1	1	0	$\leftarrow \Delta_j$

Here all the Δ_j are ≥ 0. Hence this last table gives us the optimal solution for Phase I. Since $z_v = 0$, the given solution is feasible also.

Thus the required optimal solution is given by $x_1^* = \frac{1}{3}$, $x_2^* = \frac{5}{6}$.

The optimal value z_x^* can be computed from the original objective function as follows :

$$z_x^* = 4\left(\frac{1}{3}\right) + 6\left(\frac{5}{6}\right) - 2\left(\frac{1}{3}\right)^2 - 2\left(\frac{1}{3}\right)\left(\frac{5}{6}\right) - 2\left(\frac{5}{6}\right)^2 = \frac{25}{6}.$$

Example 2. *Apply Wolfe's method to solve the quadratic programming problem :*

$$\text{Max. } z_x = 2x_1 + x_2 - x_1^2, \text{ subject to}$$

$$2x_1 + 3x_2 \leq 6,\ 2x_1 + x_2 \leq 4, \text{ and } x_1, x_2 \geq 0.$$

Also, solve this problem by Beale's method (see sec. ***29.6****) and verify your answer.* **[Delhi (OR). 90]**

Solution :

Step 1. Writing all the constraint inequalities with '≤' sign we obtain

$$2x_1 + 3x_2 \leq 6,\ 2x_1 + x_2 \leq 4,\ -x_1 \leq 0,\ -x_2 \leq 0.$$

Step 2. Now, introducing the slack variables q_1^2, q_2^2, r_1^2, r_2^2, our problem becomes of the form :

Fifth Iteration. We introduce λ_1 into the basis and drop v_2 from it.

		$c_j \rightarrow$	0	0	0	0	0	0	0	0	
B	$\mathbf{c_B}$	$\mathbf{x_B}$	$\mathbf{x_1}$	$\mathbf{x_2}$	λ_1	λ_2	μ_1	μ_2	$\mathbf{s_1}$	$\mathbf{s_2}$	
s_2	0	0	½	0	0	0	0	0	– ½	1	
λ_1	0	⅓	– 4⁄3	0	1	0	⅓	– ⅓	0	0	
λ_2	0	5⁄3	16⁄3	0	0	1	– 4⁄3	1/3	0	0	
x_2	0	1	¼	1	0	0	0	0	1/2	0	
	$\mathbf{z}_v = 0$		0	0	0	0	0	0	0	0	$\leftarrow \Delta_j$

Since all $\Delta_j = 0$, an optimum solution has been reached for *Phase-1* of the *modified* L.P. problem. The optimum solution is : $x_1 = 0, x_2 = 1, \lambda_1 = 1/3, \lambda_2 = 5/3, \mu_1 = \mu_2 = 0, s_1 = s_2 = 0$.

These values satisfy the complementary slackness conditions : $\mu_1 x_1 = 0, \mu_2 x_2 = 0, \lambda_1 s_1 = 0, \lambda_2 s_2 = 0$, and also satisfy the restrictions on the signs of the *Lagrange multipliers*.

The maximum value of $\mathbf{z} = 2x_1 + 3x_2 - 2x_1^2$ is 3.

Hence the required optimal solution is $x_1 = 0, x_2 = 1$, max. $\mathbf{z} = 3$.

EXAMINATION PROBLEMS

Use Wolfe's method to solve the following problems :

1. Min. $\mathbf{z} = x_1^2 + x_2^2 + x_3^2$, subject to
   $x_1 + x_2 + 3x_3 = 2$
   $5x_1 + 2x_2 + x_3 = 5$
   $x_1, x_2, x_3 \geq 0$.
   [**Ans.** $x_1 = 0.81, x_2 = 0.35, x_3 = 0.35$, min $\mathbf{z} = 0.857$]

2. Min. $\mathbf{z} = -x_1 - x_2 - x_3 + ½ (x_1^2 + x_2^2 + x_3^2)$, subject to
   $x_1 + x_2 + x_3 - 1 \leq 0$
   $4x_1 + 2x_2 - 7/2 \leq 0$
   $x_1, x_2, x_3 \geq 0$.
   [**Ans.** $x_1 = x_2 = x_3 = 1/3$, min $\mathbf{z} = -15/18$]

3. Max. $f(x_1, x_2) = 1.8x_1 + 3x_2 - 0.001x_1^2 - 0{\cdot}005x_2^2 - 100$,
   subject to the constraints :
   $2x_1 + 3x_2 \leq 2500$,
   $x_1 + 2x_2 \leq 1500$,
   $x_1, x_2 \geq 0$.
   [**Ans.** $x_1 = x_2 = 500$, max. $\mathbf{z} = 800$]

4. Write the *Kuhn-Tucker conditions* for the following problem :
   Min $f(x) = x_1^2 + x_2^2 + x_3^2$, subject to
   $2x_1 + x_2 - x_3 \leq 0$
   $1 - x_1 \leq 0$
   $-x_3 \leq 0$.
   Also solve this problem. **[I.A.S. 82]**

5. Max. $\mathbf{z} = 8x_1 + 10x_2 - x_1^2 - x_1^2 - x_2^2$
   subject to the constraints :
   $3x_1 + 2x_2 \leq 6$ and $x_1, x_2 \geq 0$.
   [**Ans.** $x_1 = 4/13$, $x_2 = 33/13$, max. $\mathbf{z} = 267/13$]

6. Max. $\mathbf{z} = 2x_1 + x_2 - x_1^2$ subject to
   $2x_1 + 3x_2 \leq 6$
   $2x_1 + x_2 \leq 4$, and $x_1, x_2 \geq 0$
   [Dibrugarh (Stat.) 94]
   [**Ans.** $x_1 = 2/3$, $x_2 = 14/9$, max. $\mathbf{z} = 22/9$]

7. Write the Kuhn-Tucker conditions for the following problem. Hence solve it by Wolfe's method.
   Maximize $f(x) = 2x_1 + 5x_2 + x_1x_2 - x_1^2 - x_2^2$
   subject to $3x_1 - x_2 \leq 10$, $x_1, x_2 \geq 0$. **[Virbhadra 2000]**

Beale's Method

29.6. BEALE'S METHOD

Another approach to solve a quadratic programming problem has been suggested by *Beale* (1959)*. Unlike *Wolfe's* method, this approach does not use the *Kuhn-Tucker conditions*. Instead, this method involves the *partitioning* of variables into *basic* and *non-basic* variables and the results of classical calculus are used. At each iteration, the objective function is expressed in terms of non-basic variables only.

Let the QPP be given in the form : Max. $f(\mathbf{x}) = \mathbf{cx} + ½\, \mathbf{x}^T Q\, \mathbf{x}$, subject to the constraints : $A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0$,

* Beale, E.M.L. (1959) : On quadratic programming, Naval Research Logistics Quarterly, Vol. 5, pp. 227-243.

where $\mathbf{x} = (x_1, \ldots, x_{n+m})^T$, $\mathbf{c}$ is $1 \times n$, A is $m \times (n+m)$, and Q is symmetric. Without any loss of generality, every QPP with linear constraints can be written in this form.

The *Beale's* iterative procedure for solving such type of QPP problems can be outlined in the following *steps* :

Step 1. First express the given QPP with linear constraints in the above form by introducing *slack* and/or *surplus* variables, etc.

Step 2. Now select arbitrarily m variables as basic and the remaining n variables as non-basic. With this partitioning, the constraint equation $A\mathbf{x} = \mathbf{b}$ can be written as

$$(B, R)\begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_{NB} \end{bmatrix} = \mathbf{b} \quad \text{or} \quad B\mathbf{x}_B + R\mathbf{x}_{NB} = \mathbf{b} \qquad \text{...(i)}$$

where $\mathbf{x}_B$ and $\mathbf{x}_{NB}$ denote the *basic* and *non-basic* vectors, respectively. Also, the matrix A is partitioned to submatrices B and R corresponding to $\mathbf{x}_B$ and $\mathbf{x}_{NB}$, respectively.

According to this partitioning, above equation (i) can be written as

$$\mathbf{x}_B = B^{-1}\mathbf{b} - B^{-1}R\mathbf{x}_{NB} \qquad \text{...(ii)}$$

Step 3. Express the *basic* $\mathbf{x}_B$ in terms of *non-basic* $\mathbf{x}_{NB}$ only, using the given & additional constraint equations, if any.

Step 4. Express the objective function $f(\mathbf{x})$ also in terms of $\mathbf{x}_{NB}$ only using the given and additional constraints, if any.

Thus, we observe that by increasing the value of any of the non-basic variables ($\mathbf{x}_{NB}$), the value of the objective function can be improved.

It is also important to note here that the constraints on the new problem become

$$B^{-1}R\,\mathbf{x}_{NB} \le B^{-1}\mathbf{b} \quad (\text{since } \mathbf{x}_B \ge 0)$$

Thus, any component of $\mathbf{x}_{NB}$ can increase only until $\partial f/\partial x_{NB}$ becomes zero or one or more components of $\mathbf{x}_B$ are reduced to zero.

Also, note that we face the possibility of having more than m non-zero variables at any step of iterations. This stage comes when the new point generated at some step occurs where $\partial f/\partial x_{NB}$ becomes zero. Geometrically, this means that we are no longer at an extreme point of the convex set formed by the constraints, and thus no longer have a basic solution with respect to the original constraint set. When this happens, we simply define a new variable s_i, where

$$s_i = df/\partial x_{NB}\,, \text{ and a new constraint } s_i = 0.$$

Step 5. At this stage, we now have $m + 1$ non-zero variables and $m + 1$ constraints, which is a basic solution to the extended set of constraints.

Step 6. We go on repeating the above outlined procedure until no further improvement in the objective function may be obtained by increasing one of the non-basic variables.

This technique will give us the optimal solution in a finite number of steps. For the proof of convergence refer *Beale's* (1959) research paper. Now, we shall illustrate this technique in detail by solving a number of examples.

Q. Describe briefly the *Beale's* method for solving Quadratic programming problem.

29.6-1. Illustrative Examples on Beale's Method

Example 4. *Use Beale's method for solving the quadratic programming problem (of Example 1) :*

$$\text{Max. } \mathbf{z}_x = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2\,, \textit{ subject to}$$

$$x_1 + 2x_2 \le 2, \textit{ and } x_1, x_2 \ge 0.$$

Solution. ***First Iteration.***

Step 1. Introducing slack variable x_3, the given problem becomes

$$\text{Max. } \mathbf{z}_x = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2, \quad \text{subject to}$$

Note. It can be only one variable in the basis because there is only one constraint.

$$x_1 + 2x_2 + x_3 = 2, \quad \text{and} \quad x_1, x_2, x_3 \geq 0.$$

Selecting x_1 arbitrarily to be the basic variable, we get

$$x_1 = 2 - 2x_2 - x_3 \quad \text{where } \mathbf{x}_B = (x_1), \mathbf{x}_{NB} = \begin{pmatrix} x_2 \\ x_3 \end{pmatrix}.$$

Step 2. Expressing $\mathbf{z}_x$ in terms of $\mathbf{x}_{NB}$, we find

$$f(x_2, x_3) = 4(2 - 2x_2 - x_3) + 6x_2 - 2(2 - 2x_2 - x_3)^2 - 2(2 - 2x_2 - x_3)x_2 - 2x_2^2$$

$$\frac{\partial f(\mathbf{x}_{NB})}{\partial x_2} = -8 + 6 - 4(2 - 2x_2 - x_3)(-2) - 2(2 - 4x_2 - x_3) - 4x_2$$

Now evaluating this partial derivative at $\mathbf{x}_{NB} = \mathbf{0}$, *i.e.* $x_2 = 0, x_3 = 0$, we get

$$\frac{\partial f(\mathbf{x}_{NB})}{\partial x_2} = -8 + 6 + 16 - 4 = 10$$

This indicates that the objective function will increase if x_2 is increased. Now, we should observe whether the partial derivative with respect to x_3 gives a more promising alternative.

$$\frac{\partial f(\mathbf{x}_{NB})}{\partial x_3} = -4 + 4(2 - 2x_2 - x_3) + 2x_2.$$

At the point $\mathbf{x}_{NB} = \mathbf{0}$, *i.e.* $x_2 = x_3 = 0$, we find

$$\frac{\partial f(\mathbf{x}_{NB})}{\partial x_3} = 4.$$

Thus increase in x_2 will give better improvement in the objective function.

Step 3. **How much x_2 should or may increase.**

We must now determine how much x_2 should or may increase. The maximum value of x_2 allowed to attain is determined by checking two quantities. They are *(i)* the value of x_2 for which $\partial f(\mathbf{x}_{NB})/\partial x_2$ vanishes, and *(ii)* the largest value of x_2 can attain without deriving the basic variables negative. Then x_2 will be the minimum of these two.

Since $x_1 = 2 - 2x_2 - x_3$ and $x_3 = 0$, x_1 will become negative if x_2 is increased to a value greater than 1.

The partial derivative $\partial f(\mathbf{x}_{NB})/\partial x_2$ vanishes at $\mathbf{x}_2 = 5/6$ (for $x_3 = 0$).

Now, taking minimum of (1, 5/6), we find $x_2 = 5/6$, and the new basic variable is x_2. We now initiate a new iteration by solving for x_2 in terms of x_1 and x_3.

Second Iteration

Step 4. Thus, $x_2 = 1 - \frac{1}{2}(x_1 + x_3)$

In this case,
$$\mathbf{x}_B = (x_2), \mathbf{x}_{NB} = \begin{pmatrix} x_1 \\ x_3 \end{pmatrix}.$$

Expressing $\mathbf{z}_x$ in terms of (x_1, x_3) gives,

$$f(x_1, x_3) = 4x_1 + 6(1 - \tfrac{1}{2}x_1 - \tfrac{1}{2}x_3) - 2x_1^2 - 2x_1(1 - \tfrac{1}{2}x_1 - \tfrac{1}{2}x_3) - 2[1 - \tfrac{1}{2}x_1 - \tfrac{1}{2}x_3]^2$$

$$\frac{\partial f}{\partial x_1} = 4 + 6(-\tfrac{1}{2}) - 4x_1 - 2x_1(-\tfrac{1}{2}) - 2(1 - \tfrac{1}{2}x_1 - \tfrac{1}{2}x_3) - 4(1 - \tfrac{1}{2}x_1 - \tfrac{1}{2}x_3)(-\tfrac{1}{2}) = 1 - 3x_1.$$

$$\frac{\partial f}{\partial x_3} = 0 + 6(-\tfrac{1}{2}) - 0 - 2x_1(-\tfrac{1}{2}) - 4(1 - \tfrac{1}{2}x_1 - \tfrac{1}{2}x_3)(-\tfrac{1}{2}). = -1 - x_3$$

$$\left(\frac{\partial f}{\partial x_1}\right)_{x_1 = 0, x_3 = 0} = 1, \quad \left(\frac{\partial f}{\partial x_3}\right)_{x_1 = 0, x_3 = 0} = -1.$$

This indicates that x_1 can be introduced to increase $\mathbf{z}_x$.

Step 5. If x_1 is increased to a value greater than 2, x_2 will become negative, since $x_2 = 1 - \frac{1}{2}(x_1 + x_3)$ and $x_3 = 0$. The partial derivative becomes zero at $x_1 = \frac{1}{3}$. Thus, taking minimum of $(2, \frac{1}{3})$ we find $x_1 = \frac{1}{3}$, and the new basic variable is x_1.

Since $\left(\frac{\partial f}{\partial x_3}\right)_{x_1=0,\, x_3=0} = -1$ (which is negative), x_3 cannot become basic and thus the optimal solution has been attained. Hence the optimal solution is : $x_1 = \frac{1}{3}$, $x_2 = \frac{5}{6}$, $x_3 = 0$, max $\mathbf{z} = \frac{25}{6}$.

Example 5. *Solve the following quadratic programming problem by Beale's method.*

$$Max.\ \mathbf{z}_x = 10x_1 + 25x_2 - 10x_1^2 - x_2^2 - 4x_1x_2, \text{ subject to}$$

$$x_1 + 2x_2 + x_3 = 10,\ x_1 + x_2 + x_4 = 9,\ \text{and } x_1, x_2, x_3, x_4 \geq 0.$$

Also, solve this problem by Wolfe's method and compare the efficiency of both the methods, with respect to easiness.

Solution. First Iteration.

Step 1. Selecting x_1 and x_2 arbitrarily to be the basic variables, we obtain $x_1 = 8 + x_3 - 2x_4$, $x_2 = 1 - x_3 + x_4$, where $\mathbf{x}_B = (x_1, x_2)$, $\mathbf{x}_{NB} = (x_3, x_4)$.

Step 2. Now, expressing $\mathbf{z}_x$ in terms of (x_3, x_4) gives

$$f(x_3, x_4) = 10(8 + x_3 - 2x_4) + 25(1 - x_3 + x_4) - 10(8 + x_3 - 2x_4)^2 - (1 - x_3 + x_4)^2 - 4(8 + x_3 - 2x_4)(1 - x_3 + x_4)$$

$$\frac{\partial f(\mathbf{x}_{NB})}{\partial x_3} = 10 - 25 - 20(8 + x_3 - 2x_4) + 2(1 - x_3 + x_4) - 4(1 - x_3 + x_4) + 4(1 + x_3 - 2x_4)$$

$$\therefore \quad \left(\frac{\partial f}{\partial x_3}\right)_{x_3=0,\, x_4=0} = -145.$$

This indicates that the objective function will decrease if x_3 is increased. This happens contrary to our desire to increase the objective function. The partial derivative with respect to x_4 will give us a more suitable alternative :

$$\frac{\partial f(\mathbf{x}_{NB})}{\partial x_4} = -20 + 25 - 20(-2)(8 + x_3 - 2x_4) - 2(1 - x_3 + x_4) + 8(1 - x_3 + x_4) - 4(8 + x_3 - 2x_4).$$

At the point $x_3 = 0$, $x_4 = 0$, we obtain $\partial f(\mathbf{x}_{NB})/\partial x_4 = 299$.

This indicates that increase in x_4 will certainly improve the objective function. So, we now proceed to decide how much x_4 should or may increase.

Step 3. If x_4 is increased to a value greater than 4, x_1 will become negative, since $x_1 = 8 + x_3 - 2x_4$ and $x_3 = 0$. The partial derivative becomes zero at $x_4 = \frac{299}{66}$.

Taking minimum of $(4, \frac{299}{66})$, we find $x_4 = 4$, and the new basic variables are x_4 and x_2. We now start with new iteration.

Second Iteration

Step 4. We start with solving for x_2 and x_4 in terms of x_1 and x_3. Thus

$$x_2 = 5 - \frac{1}{2}(x_1 + x_3),\ x_4 = 4 + \frac{1}{2}(x_3 - x_1).$$

In this case, $\mathbf{x}_B = (x_2, x_4)$, $\mathbf{x}_{NB} = (x_1, x_3)$.

Step 5. Expressing $\mathbf{z}_x$ in terms of (x_1, x_3) gives

$$f(x_1, x_3) = 10x_1 + 25[5 - \tfrac{1}{2}(x_1 + x_3)] - 10x_1^2 - [5 - \tfrac{1}{2}(x_1 + x_3)]^2 - 4x_1[5 - \tfrac{1}{2}(x_1 + x_3)]$$

$$\left(\frac{\partial f}{\partial x_1}\right)_{x_1=0,\, x_2=0} = -\frac{35}{2},\ \left(\frac{\partial f}{\partial x_3}\right)_{x_1=0,\, x_3=0} = -\frac{15}{2}.$$

Since both the partial derivatives are negative, hence neither x_1 nor x_3 non-basic variable can be introduced to increase $\mathbf{z}_x$ and thus the optimal solution has been obtained. The optimal solution is given by $x_1 = x_3 = 0$, and $x_2 = 5$, $x_4 = 4$.

EXAMINATION PROBLEMS

Solve the following problems by *Beale's* method :

1. Max. $z = 2x_1 + 3x_2 - x_1^2$, subject to $x_1 + 2x_2 \le 4,\ x_1, x_2 \ge 0$ [**Ans.** $x_1 = 1/4,\ x_2 = 15/8$, max. $z = 97/16$]
2. Max. $z = 2x_1 + 2x_2 - 2x_2^2$, subject to $x_1 + 4x_2 \le 4,\ x_1 + x_2 \le 2$ [**Ans.** $x_1 = 0,\ x_2 = 1$, max $z = 3$]
3. Max. $z = 6x_1 + 3x_2 - x_1^2 + 4x_1x_2 - 4x_2^2$, subject to the constraints : $x_1 + x_2 \le 3,\ 4x_1 + x_2 \le 9,\ x_1\ x_2 \ge 0$. [**Ans.** $x_1 = 2,\ x_2 = 1$, max. $z = 15$]
4. Min. $z = 183 - 44x_1 - 42x_2 + 8x_1^2 - 12x_1x_2 + 17x_2^2$ subject to the constraints : $2x_1 + x_2 \le 10,\ x_1, x_2 \ge 0$. [**Ans.** $x_1 = 3.8,\ x_2 = 2.4$, min $z = 19$]
5. Max. $z = 1/4\,(2x_3 - x_1) - 1/2\,(x_1^2 + x_2^2 + x_3^2)$, subject to the constraints : $x_1 - x_2 + x_3 = 1$, and $x_1, x_2, x_3 \ge 0$. [**Ans.** $x_1 = 1/8,\ x_2 = 0,\ x_3 = 7/8$, max. $z = 1/64$]
6. Max. $z = -4x_1^2 - 3x_2^2$, subject to : $x_1 + 3x_2 \ge 5,\ x_1 = 4x_2 \ge 4;\ x_1, x_2 \ge 0$. **[Delhi (Stat.) 95]**

29.7. SIMPLEX METHOD FOR QUADRATIC PROGRAMMING

This section deals with the solution of quadratic programming problem by the method exactly similar to *Simplex Technique* in linear programming. This method can be successfully adopted to high speed computations. We can apply this method if the constraints of the problem are linear and the quadratic objective function can be written as the product of two linear functions, *i.e.* our problem is of the from :

$$\text{Max. } \mathbf{z} = (\mathbf{c'x} + \alpha)\,(\mathbf{C'x} + \beta)$$

subject to $A\mathbf{x} = \mathbf{b},\ \mathbf{x} \ge 0$, where

(i) A is $m \times n$ matrix, *(ii)* $\mathbf{x}, \mathbf{c}, \mathbf{C}$ are $n \times 1$ column vectors, *(iii)* *b is* $m \times 1$ *column vector,*
(iv) α, β are scalars and the prime { ′ } denotes the *transpose* of a vector.

Here it is assumed that $(\mathbf{c'x} + \alpha)$, $(\mathbf{C'x} + \beta)$ are positive for all feasible solutions and the set 'S' of feasible solutions is bounded closed convex polyhedron. Also, at least two distinct feasible solutions exist. Since the proof of the algorithm is beyond the scope of this book, we shall demonstrate the procedure by a simple numerical example.

29.7-1. Demonstration By Example

To illustrate the procedure we consider the following example.

Example 6. *Maximize* $\mathbf{z} = (2x_1 + 3x_2 + 2)(x_2 - 5)$, *subject to the constraints* :

$$x_1 + x_2 \le 1,\ 4x_1 + x_2 \ge 2,\ \textit{and}\ x_1, x_2 \ge 0.$$

Solution.

Step 1. By introducing slack and surplus variables, we convert the inequalities to equations, as follows :

$$x_1 + x_2 + x_3 \qquad = 1 \qquad ...(i)$$
$$4x_1 + x_2 \qquad - x_4 = 2 \qquad ...(ii)$$

Substituting the value of x_1 from *(i)* in *(ii)*, we get

$$4(1 - x_2 - x_3) + x_2 - x_4 = 2, \quad \text{or} \quad 3x_2 + 4x_3 + x_4 = 2.$$

Thus, our problem becomes : Max. $\mathbf{z} = (2x_1 + 3x_2 + 2)(x_2 - 5)$ subject to the constraints

$$x_1 + x_2 + x_3 = 1,\quad 3x_2 + 4x_3 + x_4 = 2,\quad \text{and}\quad x_1, x_2, x_3, x_4 \ge 0.$$

Step 2. **To find the initial basic feasible solution.**

Writing the constraint equations in matrix form, we get

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 3 & 4 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Thus, the initial basic feasible solution becomes $x_1 = 1,\ x_4 = 2,\ x_2 = x_3 = 0$, and the value of the objective function is

$$z = (2 \times 1 + 3 \times 0 + 2)\,(0 - 5) = -\,20.$$

Step 3. **Initial Iteration.** We construct the following table using the symbols of simplex routine.

B	c_B	C_B	x_B	$x_1\ (\beta_1)$	x_2	x_3	$x_4\ (\beta_2)$	Minimum Ratio $\left(\dfrac{x_B}{x_3}\right)$
			$c_j \rightarrow$	2	3	0	0	
			$C_j \rightarrow$	0	1	0	0	
x_1	2	0	1	1	1	1	0	1/1
x_4	0	0	2	0	3	[4]	1	2/4
	$z^{(1)} = c_B x_B + \alpha$			0	-1	2	0	$\Delta_j^{(1)}$
	$= 2 + 2 = 4$			0	-1	0	0	$\Delta_j^{(2)}$
	$z^{(2)} = C_B x_B + \beta$			—	2/3	1/2	—	η_j
	$= 0 - 5 = -5$			—	1/3	-10	—	Δ_j
	$z = z^{(1)} . z^{(2)} = -20$					↑	↓	

We compute $\quad \Delta_j^{(1)} = c_B x_j - c_j$ and $\Delta_j^{(2)} = C_B x_j - c_j$:

$$\Delta_2^{(1)} = c_B x_2 - c_2 = (2, 0)\,(1, 4) - 3 = -1 \quad \Delta_3^{(1)} = c_B x_3 - c_3 = (2, 0)\,(1, 4) - 0 = -2,$$

$$\Delta_2^{(2)} = C_B x_2 - c_2 = (0, 0)\,(1, 3) - 1 = -1 \quad \Delta_3^{(2)} = C_B x_3 - C_3 = (0, 0)\,(1, 4) - 0 = 0.$$

We compute $\eta_j = \min\left[\dfrac{x_B}{x_j}\right]$ for non-basic vectors, *i.e.* for $j = 2, 3$. Thus we get

$$\eta_2 = \min\left[\frac{1}{1}, \frac{2}{3}\right] = \frac{2}{3},\ \eta_3 = \min\left[\frac{1}{1}, \frac{2}{4}\right] = \frac{1}{2}.$$

We now compute the net-evaluation Δ_j to test the optimality, Δ_j is computed in this case by the formula :

$$\Delta_j = z^{(1)}\,\Delta_j^{(2)} + z^{(2)}\,\Delta_j^{(1)} - \eta_j\,\Delta_j^{(1)}\,\Delta_j^{(2)}$$

Thus, we get

$$\Delta_2 = z^{(1)}\,\Delta_2^{(2)} + z^{(2)}\,\Delta_2^{(1)} - \eta_2\,\Delta_2^{(1)}\,\Delta_2^{(2)} = 4 \times (-1) + (-5) \times (-1) + \tfrac{2}{3} \times (-1) \times (-1) = -4 + 5 - \tfrac{2}{3} = \tfrac{1}{3}$$

$$\Delta_3 = z^{(1)}\,\Delta_3^{(2)} + z^{(2)}\,\Delta_3^{(1)} - \eta_3\,\Delta_3^{(1)}\,\Delta_3^{(2)} = 4 \times 0 + (-5)\,(2) - \tfrac{1}{2} \times 2 \times 0 = -10$$

The solution under test will be optimal only when all $\Delta_j \geq 0$. So, at this stage the solution is not optimal.

We proceed to improve the initial solution in the next step.

Step 4. In the above initial table, min. $\Delta_j = -10$. Thus, **z** can be increased by taking x_3 into the basis. The method of determining the departing variable and also the new values of x_{ij}, x_B, $\Delta_j^{(1)}$, $\Delta_j^{(2)}$, corresponding to new basic feasible solution, will be the same as for linear programming problem. Minimum ratio rule indicates that β_2 (*i.e.* x_4) will be leaving vector. Hence key-element is 4. We construct the improved table as below.

B	c_B	C_B	x_B	x_1	x_2	x_3	x_4	
			c_j	2	3	0	0	
			C_j	0	1	0	0	
x_1	2	0	½	1	¼	0	$-$ ¼	
x_3	0	0	½	0	¾	1	¼	
	$z^{(1)} = c_B x_B + \alpha$			0	$-$ 5/2	0	$-$ ½	$\leftarrow \Delta_j^{(1)}$
	$= (2, 0)\,(½, ½) + 2 = 3$			0	-1	0	0	$\leftarrow \Delta_j^{(2)}$
	$z^{(2)} = C_B x_B + \beta$			—	2/3	—	2	$\leftarrow \eta_j$
	$= (0, 0)\,(½, ½) - 5 = -5$							
	$\therefore z = z^{(1)} z^{(2)} = 3\,(-5) = -15$			0	47/6	0	5/2	$\leftarrow \Delta_j$

In above table, we compute

$\Delta_2^{(1)} = \mathbf{c}_B \mathbf{x}_2 - c_2 = (2, 0)\,(\,1/4\,, 3/4\,) - 3 = -5/2,\ \Delta_4^{(1)} = \mathbf{c}_B \mathbf{x}_4 - c_4 = (2, 0)\,(\,-1/4\,, 1/4\,) - 0 = -1/2,$

$\Delta_2^{(2)} = \mathbf{C}_B \mathbf{x}_2 - C_2 = (0, 0)\,(\,1/4\,, 3/4\,) - 1 = -1,\ \ \Delta_4^{(2)} = \mathbf{C}_B \mathbf{x}_4 - C_4 = (0, 0) - 0 = 0.$

$$\eta_2 = \min\left[\frac{1/2}{1/4}, \frac{1/2}{3/4}\right] = 2/3,\ \eta_4 = \min\left[-, \frac{1/2}{1/4}\right] = 2$$

$$\Delta_2 = \mathbf{z}^{(1)}\,\Delta_2^{(2)} + \mathbf{z}^{(2)}\,\Delta_2^{(1)} - \eta_2\Delta_2^{(1)}\,\Delta_2^{(2)} = 3 \times (1) + (-5) \times (-5/2) - 2/3 \times (5/2) \times (1) = 47/6\,.$$

$$\Delta_4 = \mathbf{z}^{(1)}\,\Delta_4^{(2)} + \mathbf{z}^{(2)}\,\Delta_4^{(1)} - \eta_4\,\Delta_4^{(1)}\,\Delta_4^{(2)} = 3 \times 0 + (-5) \times (-1/2) - 2 \times (-1/2) \times 0 = 5/2.$$

Since all $\Delta_j \geq 0$, we have reached the local maximum $\mathbf{z} = -15$, and locally optimum basic feasible solution is $x_1 = 1/2$, $x_2 = 0$, $x_3 = 1/2$, $x_4 = 0$.

However, in this illustrative example, the local maximum thus obtained is also a global maximum but it may not be true in general. Hence, we can use the technique of cutting plane method to obtain the global maximum. The solution of minimization problems can also be obtained analogously.

EXAMIANTION PROBLEMS

1. What is meant by quadratic programming ? How does quadratic programming problem differ from the linear programming problem ?
2. Is it correct to say that in the quadratic programming, the objective equation and then constraints both should be quadratic ? If not, give your own comments.
3. Discuss Beale'e method for solving a quadratic programming problem. Hence or otherwise solve : Min. $\mathbf{z} = x_1^2 + 3x_2^2$, subject to the constraints :

   $x_1 + 3x_2 \geq 5,\ 0.5x_1 + 2x_2 \geq 2$, and $x_1, x_2 \geq 0$.

   [Ans. $x_1 = 5/7$, $x_2 = 10/7$, *Min.* $\mathbf{z} = 400/49$]
4. Discuss any one method for solving a quadratic programming problem and solve :

   Min. $x_1^2 + x_2^2 - 4x_1 - 2x_2 + 5$, subject to

   $x_1 + x_2 \leq 4$, and $x_1, x_2 \geq 0$.
5. Discuss Wolfe's method for solving a quadratic programming problem. Hence or othewise solve :

   Min. $\mathbf{z} = 6 - 6x_1 + 2x_1^2 - 2x_1x_2 + 2x_2^2$, subject to the constraints :

   $x_1 + x_2 \leq 2$, and $x_1, x_2 \geq 0$ **[Delhi (OR). 93, (Maths.) 70]**

   [Ans. $x_1 = 3/2$, $x_2 = 1/2$, min. $\mathbf{z} = 1/2$]
6. Consider the problem :

   Min. $\mathbf{z} = -4x_1 + x_1^2 - 2x_1x_2 + 2x_2^2$, subject to the constraints

   $2x_1 + x_2 \leq 6,\ x_1 - 4x_2 \leq 0$, and $x_1, x_2 \geq 0$.

   Show that $\mathbf{z}^*$ is strictly convex and then solve the problem by any of the quadratic programming techniques.

   [Ans. $x_1 = 38/13$, $x_2 = 2/13$, min. $\mathbf{z} = 116/13$]
7. The manufacturing and raw material costs for making each of two products *A, B* is proportional to the squares of the quantity made. The products are made from a limited supply of a particular raw material and are both processed on the same machine.

   It takes 30 minutes to process one unit of product *A* and 20 minutes of each unit of *B* and the machine operates for a maximum of 40 hours a week. Product *A* needs 1 kg. and product *B* needs 3 kg. of the raw material per unit which is limited in supply of 180 kg. per week.

   If the net income from the products are Rs. 160 and Rs. 600 per unit and manufacturing costs are $2x_1^2$ and $3x_2^2$ respectively, find how much of each product should be produced.
8. A factory is faced with a decision regarding the number of units of a product it should produce during months of January and February respectively. At the end of January sufficient units must be on hand so as to supply regular customers with a total of at least 100 units. Furthermore, at the end of February, the required quantity will be 200 units. Assume that factory ceases production at the end of February. The production cost *C* is a simple function of output *X* and is given by $C = 2X^2$. In addition to production cost, units produced in January which are not sold until February incur an inventory cost of Rs. 8 per unit. Assume the initial inventory to be zero. Formulate the problem as a *quadratic programming problem* and show that the minimum cost solution is to produce 149 units in January and 151 units in February. The number of units produced must be equal to the number demanded and distributed.
9. Write short notes on :

   (a) Quadratic Programming

   (b) Application of non-linear programming problem.

***

30

SEPARABLE PROGRAMMING

30.1. INTRODUCTION

Separable programming deals with such non-linear programming problems in which the objective function as well as all the constraints are separable. In many decision making situations, the non-linear profit or cost functions are related by relatively smooth curves, rather than sharp curves. Thus, breaking points are obtained on such curves. In such cases, the non-linear objective function, with a smooth curve for it, may be approximated by a series of *piecewise linear segments.* This approximation will introduce some error which may be negligibly small in many cases. Such error can be reduced much by increasing the number of linear segments. But on the other hand, excessive increase of linear segments will enlarge the size of the problem thus consuming more computational time to obtain the optimal solution.

Piecewise linear approximation can be done for *convex* as well as *concave* functions. It has been observed earlier that if all the linear segments of the objective function are *concave (convex)*, then a valid optimal solution needs the objective function to be maximized (minimized). Similarly, non-linear constraint functions can also be approximated by linear segments.

Thus a NLPP can be reduced (approximately) to a LPP and the usual simplex method can be used to get an optimal solution. First we shall discuss about the separable functions.

30.2. SEPARABLE FUNCTIONS

Definition. *A function $f(x_1, x_2, \ldots, x_n)$ is said to be separable if it can be expressed as the sum of n single valued functions $f_1(x_1), f_2(x_2), \ldots, f_n(x_n)$, i.e. $f(x_1, x_2, \ldots, x_n) = f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n)$.*

For example, the linear function given by

$$g(x_1, x_2, \ldots, x_n) = c_1 x_1 + c_1 x_2 + \ldots + c_n x_n$$

(where c's are constants) is a separable function. On the other hand, the function defined by

$$g(x_1, x_2, x_3) = x_1^3 + x_2^2 \cos(x_1 + x_3) + x_3 \,.\, 3^{x_2} + \log(x_1 + x_2)$$

is not a separable function.

Reducible to separable forms. Sometimes the functions are not directly separable but can be made separable by simple substitutions.

For example, in the case of maximizing $\mathbf{z} = x_1 x_2$, we let $y = x_1 x_2$. Then, $\log y = \log x_1 + \log x_2$. Hence the problem becomes :

$$Max.\ \mathbf{z} = y, \ \ subject\ to\ \ \log y = \log x_1 + \log x_2$$

which is clearly separable. In above substitution, it is assumed that x_1 and x_2 are both positive variables because the logarithmic function is undefined for non-positive values.

If x_1 and x_2 assume zero values (*i.e.* $x_1, x_2 \geq 0$), we can handle the situation as follows. We define the new variables u_1 and u_2 by the equations : $u_1 = x_1 + v_1$ and $u_2 = x_2 + v_2$ where v_1 and v_2 are positive constants. This indicates that the new variables u_1 and u_2 are strictly positive. Now, we can make the substitution : $x_1 = u_1 - v_1$, $x_2 = u_2 - v_2$, and $x_1 x_2 = (u_1 - v_1)(u_2 - v_2) = u_1 u_2 - v_2 u_1 - v_1 u_2 + v_1 v_2$.

If we let $y = u_1 u_2$, then the problem becomes :

$$Max.\ \mathbf{z} = y - v_2 u_1 - v_1 u_2 + v_1 v_2\,,\ subject\ to\ the\ condition$$

$$\log y = \log u_1 + \log u_2$$

which is properly separable.

Other examples of the functions that can be made readily separable are $e^{x_1 + x_2}$ and $x_1^{x_2}$.

30.3. DEFINITIONS

Separable Programming Problem. *A NLPP in which the objective function can be expressed as a linear combination of several different single variable functions, of which some or all are non-linear, is called a* ***separable programming problem.***

Convex Programming. *Non-linear programming which has the problem of minimizing a convex objective function (or maximizing a concave objective fucntion) in the convex set of points is called* ***convex programming.***

In above definitions, nothing has been explained about the constraints of the problem. In general, we can take the constraints to be non-linear.

Separable Convex Programming Problem. *A separable programming problem in which the separate functions are all convex can be defined as a* ***separable convex programming*** *problem with separable objective function.*

Thus, if $f(\mathbf{x})$ be the objective function, then for separable convex programming, $f(\mathbf{x})$ must be separable as

$$f(\mathbf{x}) = f_1(x_1) + f_2(x_2) + \dots + f_n(x_n)$$

where $f_1(x_1), f_2(x_2), \dots, f_n(x_n)$ are all convex.

For example, if $f(\mathbf{x}) = 7x_1^2 + 2x_2^2 - 5x_1 + 3x_2$, then by letting

$$f_1(x_1) = 7x_1^2 - 5x_1 \quad \text{and} \quad f_2(x_2) = 2x_2^2 + 3x_2$$

we may write $f(\mathbf{x}) = f_1(x_1) + f_2(x_2)$, where $f_1(x_1)$ and $f_2(x_2)$ are both convex functions.

We now proceed to discuss how piecewise linear approximations can reduce a given separable convex (or concave) nonlinear programming problem to a linear programming problem so that it can be easily solved by using the simplex method.

30.4. PIECE-WISE LINEAR APPROXIMATION OF NONLINEAR FUNCTION

Consider the nonlinear objective function : *Maximize* $\mathbf{z} = \sum_{j=1}^{n} f_j(x_j)$, *subject to the constraints* :

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \; i = 1, 2, \dots, m \quad \textit{and} \quad x_j \geq 0, \textit{for all } j$$

where $f_j(x_j)$ is a nonlinear function in x_j. Now our aim is to reduce the nonlinear objective function into a linear form by approximating each $f_j(x_j)$ over its prescribed domain. A linear approximation for each $f(x)$ is shown in *Fig.* 30.1 below.

The points (a_k, b_k), $k = 1, 2, \dots, K$ are called the breaking points joining the linear segments which approximate the funciton $f(x)$. Let w_k denote a non-negative weight associated with the kth breaking point such that $\sum_{k=1}^{K} w_k = 1$.

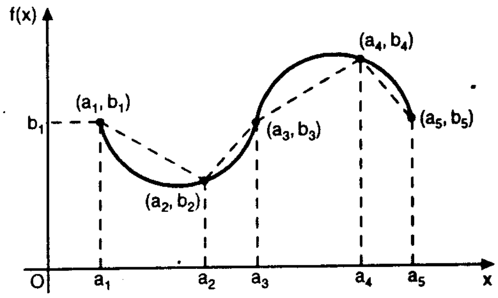

Fig. 30.1

Assume that additional constraints are imposed (if necessary) so that all w_k and $w_{k'}$ but $w_{k'+1}$ are equated to zero. Then any point on the line joining the breaking points $(a_{k'}, b_{k'})$ and $(a_{k'+1}, b_{k'+1})$ can be defined by properly specifying $w_{k'}$ and $w_{k'+1}$. This means that such a point will be the weighted average of $(a_{k'}, b_{k'})$ and $(a_{k'+1}, b_{k'+1})$. Keeping this point in mind, it follows that $f(x)$ and x can be approximated by

$$f(x) = \sum_{k=1}^{K} b_k w_k, \text{ where } x = \sum_{k=1}^{K} a_k w_k,$$

The necessary additional constraints for the validity of the approximation are :

$$0 \le w_1 \le y_1$$
$$0 \le w_2 \le y_1 + y_2,$$
$$0 \le w_3 \le y_2 + y_3,$$
$$\vdots \qquad \vdots$$
$$0 \le w_{k-1} \le y_{k-2} + y_{k-1},$$
$$0 \le w_k \le y_{k-1},$$
$$\sum_{k=1}^{K} w_k = 1, \quad \sum_{k=1}^{K-1} y_k = 1, \; y_k = 0 \text{ or } 1 \text{ for all } k.$$

Now suppose that $y_{k'} = 1$, then from the last constraint given above all other $y_k = 0$. The immediately preceding constraints will then ensure that $0 \le w_{k'} \le y_{k'} = 1$ and $0 \le w_{k'+1} \le y_{k'} = 1$. Thus the remaining constraints should give $w_k \le 0$ and therefore all other $w_k = 0$ as desired.

Using above approximation, we can substitute the corresponding values of x and $f(x)$ in the original problem. In order to ensure the validity of approximation the additional constraints should also be added. Obviously, the size of the problem may increase, thus increasing the computational time for an optimal soltution.

We now consider the separable problem in the following section.

30.5. REDUCTION OF SEPARABLE PROGRAMMING PROBLEM TO L.P.P.

Let us now consider the separable programming problem :

$$Max. \text{ (or } Min.) \ \mathbf{z} = \sum_{j=1}^{n} f_j(x_j),$$

subject to the constraints :

$$\sum_{j=1}^{n} g_{ij}(x_j) \le b_i, \; x_j \ge 0 \quad (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$$

where some or all $g_{ij}(x_j)$, $f_j(x_j)$ *are non-linear.*

This problem can be approximated as a mixed integer programming problem as follows :

Let the number of breaking points for jth variable 'x_j' be equal to K_j and a_{jk} be its kth breaking value.

Let w_{jk} be the weight associated with the kth breaking point of jth variable. Then the equivalent mixed problem is :

$$Max. \text{ (or } Min.) \ \mathbf{z} = \sum_{j=1}^{n} \sum_{k=1}^{K_j} f_j(a_{jk}) \, w_{jk},$$

subject to the constraints :

$$\sum_{j=1}^{n} \sum_{k=1}^{K_j} g_{ij}(a_{jk}) \, w_{jk} \le b_i, \; i = 1, 2, \ldots, m$$
$$0 \le w_{j1} \le y_{j1}$$
$$0 \le w_{jk} \le y_{j.k-1} + y_{jk}, \; k = 2, 3, \ldots, K_{j-1}$$
$$0 \le w_{jK_j} \le y_{j.k_{j-1}}.$$

$$\sum_{k=1}^{K_j} w_{jk} = 1, \quad \sum_{k=1}^{K_j - 1} y_{jk} = 1$$

$$y_{jk} = 0 \text{ or } 1, k = 1, 2, \ldots, K_j, j = 1, 2, \ldots, n.$$

The variables for the approximating problem are given by w_{jk} and y_{jk}.

We can use the regular simplex method for solving the approximate problem under the additional constraints involving y_{jk}. The restricted basis condition indicates that no more than two w_{jk} can appear in the basis. Also, two w_{jk} can be positive only if they are adjacent. Thus, the strict optimality condition of the simplex method is used to select the entering variable w_{jk} only if it satisfies the above restrictions. Otherwise, the variable w_{jk} having the next best optimality indicator $(z_{jk} - c_{jk})$ is considered for entering the soltuion. The process is repeated until the optimality condition is satisfied or until it is impossible to introduce new w_{jk} without violating the restricted basis condition, whichever occurs first. At this stage, the final table yields the approximate optimum solution to the problem.

Remark. *The restricted basis method* can only guarantee a local optimum to the approximate problem while the mixed intger-programming method provides the global optimum. Also, in these two methods, the approximate solution may not be a feasible solution to the original problem. In fact, the approximating problem may give rise to additional extreme points which do not exist in the original problem. This depends mainly on the degree of accuracy of the linear approximation used.

30.6. SEPARABLE PROGRAMMING ALGORITHM

The *iterative* procedure for the separable programming problem (as defined in **Sec. 30.3**) can be outlined in the following algorithm.

Step 1. If the objective function is of minimization form, convert it into maximization.

Step 2. Test whether the functions $f_j(x_j)$ and $g_{ij}(x_j)$ satisfy the concavity (convexity) conditions required for the maximization (minimization) of non-linear programming problem. If the conditions are not satisfied, the method is not applicable, otherwise go to next step.

Step 3. Divide the interval $0 \le x_j \le t_j$ $(j = 1, 2, \ldots, n)$ into a number of mesh points a_{jk} $(k = 1, 2, \ldots, K_j)$ such that $a_{j1} = 0, a_{j1} < a_{j2} < \ldots < a_{jK_j} = t_j$.

Step 4. For each point a_{jk}, compute piecewise linear approximation for each $f_j(x_j)$ and $g_{ij}(x_j)$, $j = 1, 2, \ldots, n$; $i = 1, 2, \ldots, m$.

Step 5. Using the computations of *step 4*, write-down the piecewise linear approximation of the given non-linear programming problem.

Step 6. Now solve the resulting linear programming problem by *two phase simplex method.* For this method consider w_{i1} $(i = 1, 2, \ldots, m)$ as artificial variables. Since the costs associated with them are not given, we assume them to be zero. Then *Phase I* of this method is automatically complete. Therfore, the initial simplex table of *Phase I* is optimum and hence will be the starting simplex table for *Phase II.*

Step 7. Finally, we obtain the optimum solution x_j^* of the original problem by using the relations :

$$x_j^* = \sum_{k=1}^{K_j} a_{jk} w_{jk} \quad (j = 1, 2, \ldots, n).$$

Q. 1. What do you mean by separable and/or non-linear convex programming ? How will you solve the separable non-linear programming problem :

$$\text{Min. } z = \sum_{j=1}^{n} f_{oj}(x_j), \text{ subject to the constraints : } \sum_{j=1}^{n} f_{oj}(x_j) \ge b_i \; (i = 1, 2, \ldots, m),$$

2. Show that if $f_{oj}(x_j)$ is strictly convex and $f_{ij}(x_j)$ is concave for $i = 1, \ldots, m$, then we can discard the additional restriction in the approximated separable non-linear programming problem (SLPP) of above question and solve the resulting LPP to find an approximate optimal solution to SNLPP.

The following numerical examples will make the above steps clear.

30.6–1. Illustrative Examples

Example 1. *Use separable programming algorithm to the non-linear programming problem :*

$$\text{Max. } \mathbf{z} = x_1 + x_2^4 \text{, subject to the constraints :}$$

$$3x_1 + 2x_2^2 \le 9, x_1 \ge 0, x_2 \ge 0.$$

Solution. Although, the exact optimum solution to this problem can be obtained by inspection as $x_1^* = 0, x_2^* = \sqrt{9/2} = 2.13$, and max. $\mathbf{z} = 20.25$. But, we demonstrate here how the approximation can be used.

Step 1. The objective function is already present in the maximization form. So we proceed to next step.

Step 2. Separable functions are : $f_1(x_1) = x_1$, $f_2(x_2) = x_2^4$ and $g_{11}(x_1) = 3x_1$, $g_{12}(x_2) = 2x_2^2$.
Since $f_1(x_1)$ and $g_{11}(x_1)$ are already in the linear form, we left them in their present form. Moreover, we observe that the above separable functions satisfy the concavity-convexity conditions for the maximization problem.

Step 3. The constraints of the problem suggests $x_1 \le 3$ and $x_2 \le \sqrt{9/2} = 2.13$.
So we can take $t_1 = 3$ and $t_2 = 3$ as the upper limits for the variables x_1 and x_2 respectively. Therefore, we divide the closed interval [0, 3] into four equal parts, *i.e.*

$$a_{j1} = 0, a_{j1} < a_{j2} < a_{j3} < a_{j4} = 3 \ (j = 1, 2)$$

Step 4. Now consider $f_2(x_2) = x_2^4$ and $g_{12}(x_2) = 2x_2^2$, it is assumed that there are four breaking points ($K_2 = 4$). Since the value of $x_2 \le 3$, then

k	a_{2k}	$f_{2(a_{2k})}$	$g_{12(a_{2k})}$
1	0	0	0
2	1	1	2
3	2	16	8
4	3	81	18

This gives us

$$f_2(x_2) \cong w_{21} f_2(a_{21}) + w_{22} f_2(a_{22}) + w_{23} f_2(a_{23}) + w_{24} f_2(a_{24})$$
$$\cong w_{21} \,.\, 0 + w_{22} \,.\, 1 + w_{23} \,.\, 16 + w_{24} \,.\, 81 \qquad \cong w_{22} + 16 w_{23} + 81 w_{24}$$

Similarly, for the function $g_{12}(x_2)$, we have

$$g_{12}(x_2) \cong w_{21}\, g_{12}(a_{21}) + w_{22}\, g_{12}(a_{22}) + w_{23}\, g_{12}(a_{23}) + w_{24}\, g_{12}(a_{24})$$
$$\cong w_{21} \,.\, 0 + w_{22} \,.\, 2 + w_{23} \,.\, 8 + w_{24} \,.\, 18 \cong 2w_{22} + 8w_{23} + 18w_{24}$$

Thus, the reduced L.P.P. now becomes :

$$\text{Max. } \mathbf{z} = x_1 + w_{22} + 16 w_{23} + 81 w_{24}, \text{ subject to the constraints :}$$

$$3x_1 + 2w_{22} + 8w_{23} + 18w_{24} \le 9, \; w_{21} + w_{22} + w_{23} + w_{24} = 1 \text{ and } w_{21}, w_{22}, w_{23}, w_{24} \ge 0,$$

with the additional restriction that :

(i) for each $j = 1, 2$, more than two w_{jk} are positive, and

(ii) if two w_{jk} are positive, they must correspond to adjacent points.

To solve the approximate problem by simplex method :

Introducing the slack variable $s_1 \ge 0$, the inequality constraint is converted into an equation. Thus, the reduced L.P.P. now becomes :

$$\text{Max. } \mathbf{z} = x_1 + w_{22} + 16 w_{23} + 81 w_{24} + 0 s_1, \text{ subject to the constraints :}$$

$$3x_1 + 2w_{22} + 8w_{23} + 18w_{24} + s_1 = 9, \; w_{21} + w_{22} + w_{23} + w_{24} = 1 \text{ and } w_{2j} \ge 0, j = 1, 2, 3, 4.$$

This reduced L.P.P. can be solved by *Phase-II* of *two-phase* simplex method by treating w_{21} as the artificial variable whose cost in the objective function is taken zero.

Thus the intial simplex table for *Phase-II* is given as follows :

Starting Table

		$c_j \rightarrow$	1	1	16	81	0	0	
Basic Var.	c_B	x_B	x_1	w_{22}	w_{23}	w_{24}	s_1	w_{21}	Min. Ratio
s_1	0	9	3	2	8 ↑	18	1	0	9/8
w_{21}	0	1	0	1	← [1]	1	0	1	7/1
	$z = 0$		−1	−1	−16 ↑	−81	0	0 ↓	$\leftarrow \Delta_j$

In this table, the optimality indicator Δ_j shows that w_{24} is the entering variable. Since w_{21} is artificial basic, it must be dropped before w_{24} enters the solution (restricted basis condition). From the feasibility condition (minimum ratio rule), it is observed that s_1 must be the leaving variable. This means that w_{24} cannot enter the solution. So we consider the next best entering variable w_{23}. Again w_{21} must be dropped first. From the feasibility condition, it follows that w_{21} is the leaving variable as desired. The new table thus becomes as below :

First Iteration Table

			1	1	16	81	0	0	
Basic Var.	c_B	x_B	x_1	w_{22}	w_{23}	w_{24}	s_1	w_{21}	Min. (x_B/w_{24})
s_1	0	1	3	−6	0	← [10] ↑	1	−8	1/10 ←
w_{23}	16	1	0	1	1	1	0	1	1/1
	$z = 16$		−1	15	0	−65 ↑	0 ↓	16	$\leftarrow \Delta_j$

Obviously, w_{24} is the entering variable. Since w_{23} is in the basis, w_{24} is an admissible entering variable. The minimum ratio rule indicates that s_1 will leave the solution. Thus we get the following table.

Second Iteration Table

		$c_j \rightarrow$	1	1	16	81	0	0
Basic Var.	c_B	x_B	x_1	w_{22}	w_{23}	w_{24}	s_1	w_{21}
w_{24}	81	1/10	3/10	−6/10	0	1	1/10	−8/10
w_{23}	16	9/10	−3/10	16/10	1	0	−1/10	18/10
	$z = c_B x_B = 45/2 = 22.5$		37/2	−24	0	0	−13/2	−36

This table shows that w_{21} and w_{22} are the candidates for the entering variable. Since w_{21} is not an adjacent point to the basic variables w_{23} and w_{24}, it cannot be admitted. Again, w_{22} also cannot be admitted since w_{24} cannot be dropped. Consequently, the process ends at this point and the given solution is the best feasible solution for the reduced L.P. problem.

***Step* 5.** Now, to obtain the solution in terms of original variables x_1 and x_2, we consider $w_{23} = 9/10$, $w_{24} = 1/10$.

Therefore, $x_2 = 2w_{23} + 3w_{23} = 2 \times (9/10) + 3 \times (1/10) = 2.1$, $x_1 = 0$ and $z = 22.5$.

Note. It is important to note here that the approximate optimum value of x_2 (= 2.1) is very near to the actual optimum value (= 2.13). However, the value of the objective function z differs by about 10% error. This approximation may be further improved by increasing the number of breaking points.

Example 2. *Use the separable programming algorithm to solve the non-linear programming problem :*

Max. $z = 3x_1 + 2x_2$, *subject to the constraints :*

$$4x_1^2 + x_2^2 \leq 16, \text{ and } x_1, x_2 \geq 0.$$

Solution :

***Step* 1.** The objective function is already given in maximization form. So, we proceed to next step.

***Step* 2.** Let us suppose $f_1(x_1) = 3x_1$, $f_2(x_2) = 2x_2$, $g_{11}(x_1) = 4x_1^2$, $g_{12}(x_2) = x_2^2$.

We observe that these functions satisfy the concavity (convexity) conditions. Because $f_1(x_1)$ and $f_2(x_2)$ are already linear, they are left in their present form.

***Step* 3.** The constraints of the problem suggest, $x_1 \leq 2$ and $x_2 \leq 4$.

We suppose that $t_1 = 4$ and $t_2 = 4$ are the upper limits for the variables x_1 and x_2 respectively. So, we divide the closed interval [0, 4] into four sub-intervals of equal size. The number of sub-intervals for x_1 and x_2 should be necessarily the same. But, it is not necessary that any one of them be of equal size.

To obtain the approximating linear programming for the given non-linear problem, we divide the interval $0 \leq x_j \leq 4$ into five mesh points a_{jk} ($j = 1, 2$; $k = 1, 2, 3, 4, 5$) such that

$$a_{j1} = 0,\ a_{j1} < a_{j2} < a_{j3} < a_{j4} < a_{j5} = 4.$$

***Step* 4.** For each point a_{jk}, we compute the piecewise linear approximation for each of $f_j(x_j)$ and $g_{1j}(x_j)$. For $j = 1, 2$ and $k = 1, 2, 3, 4, 5$, we have

k	a_{jk}	$f_1(x_1)$	$f_2(x_2)$	$g_{11}(x_1)$	$g_{12}(x_2)$
1	0	0	0	0	0
2	1	3	2	4	1
3	2	6	4	16	4
4	3	9	6	36	9
5	4	12	8	64	16

This gives the piecewise linear approximations :

$$f_1(x_1) \cong 0w_{11} + 3w_{12} + 6w_{13} + 9w_{14} + 12w_{15}, \quad f_2(x_2) \cong 0w_{21} + 2w_{22} + 4w_{23} + 6w_{24} + 8w_{25}$$

$$g_{11}(x_1) \cong 0w_{11} + 4w_{12} + 16w_{13} + 36w_{14} + 64w_{15}\ g_{12}(x_2) \cong 0w_{21} + 1w_{22} + 4w_{23} + 9w_{24} + 16w_{25}$$

***Step* 5.** Using the piecewise linear approximations obtaind in *Step* 4, we get the approximating LPP of the given problem as :

$$\text{Max. } \mathbf{z} = (0w_{11} + 3w_{12} + 6w_{13} + 9w_{14} + 12w_{15}) + (0w_{21} + 2w_{22} + 4w_{23} + 6w_{24} + 8w_{25})$$

subject to the constraints :

$$(0w_{11} + 4w_{12} + 16w_{13} + 36w_{14} + 64w_{15}) + (0w_{21} + 1w_{22} + 4w_{23} + 9w_{24} + 16w_{25}) \leq 16$$

$w_{11} + w_{12} + w_{13} + w_{14} + w_{15} = 1,\ w_{21} + w_{22} + w_{23} + w_{24} + w_{25} = 1$, and $w_{jk} \geq 0$ ($j = 1, 2$; $k = 1, 2, 3, 4, 5$), with the additional restriction that :

(*i*) for each j, more than two w_{jk} are positive, and

(*ii*) if two w_{jk} are positive, they must correspond to adjacent points.

Step* 6. To solve the above LPP by *Simplex Method.

We introduce the slack variable s_1 for converting the inequality constraint into an equation. Now, we are able to solve the reduced LPP by *Phase-II* of the *two-phase simplex method*, treating w_{11} and w_{21} as the artificial variables whose costs in the objective function are taken zero.

Thus we get the initial simplex table as follows :

Starting Table of Simplex Method

		$c_j \rightarrow$	3	6	9	12	2	4	6	8	0	0	0
B	c_B	x_B	w_{12}	w_{13}	w_{14}	w_{15}	w_{22}	w_{23}	w_{24}	w_{25}	s_1	w_{11}	w_{21}
s_1	0	16	4	16	36	64	1	4	9	16	1	0	0
w_{11}	0	1	1	1	1	1	0	0	0	0	0	1	0
w_{21}	0	1	0	0	0	0	1	1	1	[1]	0	0	1
	$\mathbf{z} = 0$		−3	−6	−9	−12	−2	−4	−6	−8 ↑	0	0	0 ↓

In this table, min. $\Delta_j = -12$ indicates that we must enter w_{15} and drop s_1. But, this violates the additional restriction. So we search for the next best vector to enter the basis. Above table indicates that either s_1 or w_{21} can be the departing variable satisfying the additional restrictions. Here we select w_{21} as the leaving variable.

Introducing w_{25} and dropping w_{21}, we get the first interation table.

We consider the problem :

$$Min. f(\mathbf{x}) = \sum_{j=1}^{n} P_j(\mathbf{x}) \qquad ...(31.11)$$

where $P_j(\mathbf{x})$ has the form

$$P_j(\mathbf{x}) = c_j \prod_{i=1}^{k} (x_i)^{a_{ij}}, \ j = 1, 2, ..., n.$$

It is assumed that all $c_j > 0$ and that n is finite. The exponents a_{ij} are real but unrestricted in sign. The function $f(\mathbf{x})$ takes the form of a polynomial except that the exponents a_{ij} may be negative. For the reason that all $c_j > 0$ and being closely related to polynomials, *Duffin* and *Zener* have given $f(\mathbf{x})$ the name *posynomial.*

This problem will be called as the *primal* problem. The variables x_i are assumed to be strictly positive so that the region $x_i \leq 0$ represents the infeasible space.

31.5. TO DERIVE NECESSARY CONDITIONS FOR OPTIMALITY

The problem (31.11) can be approached by taking the partial derivatives with respect to each x_r and requiring the result equal to zero. Thus

$$\frac{\partial f(\mathbf{x})}{\partial x_r} = \sum_{j=1}^{n} \frac{\partial P_j(\mathbf{x})}{\partial x_r} = 0, \ r = 1, ..., k. \qquad ...(31.12)$$

But, $$\frac{\partial P_j(\mathbf{x})}{\partial x_r} = \sum_{j=1}^{n} c_j a_{rj} (x_r)^{a_{ij}-1} \prod_{i \neq r} (x_i)^{a_{ij}} = \frac{a_{rj}}{x_r} P_j(\mathbf{x}), \ r = 1, ..., k.$$

Putting this result into previous equation (31.12) gives

$$\frac{1}{x_r} \sum_{j=1}^{n} a_{rj} P_j(\mathbf{x}) = 0.$$

Since each x_r is strictly positive (> 0) and each $c_j > 0, f(\mathbf{x}^o)$ will be positive. Thus, we may divide $\partial f(\mathbf{x})/\partial x_r$ by $f(\mathbf{x}^0)$, to get

$$\sum_{J=1}^{n} \frac{a_{rj} P_j}{f(\mathbf{x}^o)} = 0, \ r = 1, ..., k.$$

Let us now make a simple transformation of variables. We define

$$y_j = \frac{P_j}{f(\mathbf{x}^0)}, j = 1, 2, ... n. \qquad ...(31.13)$$

Using this definition and the necessary condition, we find that

$$\sum_{j=1}^{n} a_{rj} y_j = 0, \ r = 1, ..., k \qquad ...(31.14)$$

By virtue of the definition of y_j, we obtain $$\sum_{j=1}^{n} y_j = \frac{1}{f(\mathbf{x}^0)} \sum_{j=1}^{n} P_j \qquad ...(31.15)$$

which must be equal to 1 at the optimal solution.

Thus, summarizing the results, we have

$$\sum_{j=1}^{n} y_j = 1 \text{ (normality)}, \quad \sum_{j=1}^{n} a_{rj} y_j = 0, \ r = 1, ..., k \text{ (orthogonality)}$$

These necessary conditions are known as the *orthogonality* and *normality* conditions.

It will be more convenient to work in matrix notation. We define

$$A = \begin{bmatrix} 1 & \cdots & 1 \\ a_{11} & & a_{1n} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kn} \end{bmatrix}, \ y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \ b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Thus, we require from normality and orthogonality conditions

$$Ay = b \qquad ...(31.16)$$

We have now reduced the original, non-linear problem to one of finding the correct set of **y** which solves these linear non-homogeneous equations.

(*i*) There will be no solution if *Rank* $(A, \mathbf{b}) >$ *Rank* (A), where $(A, \mathbf{b})$ denotes the marix A augumented by **b**.

$$(A, \mathbf{b}) = \begin{bmatrix} 1 & \dots & 1 & b_0 \\ a_{11} & \dots & a_{1n} & b_1 \\ \vdots & & \vdots & \vdots \\ a_{k1} & \dots & a_{kn} & b_k \end{bmatrix}$$

(*ii*) There will be unique solution if A is square matrix and

$$Rank\ (A, \mathbf{b}) = Rank\ (A)$$

(*iii*) There will be an infinite number of solutions if

$$n > k + 1 \text{ or } Rank\ (A) < n.$$

When conditions *(i)* exists there is no vector $\mathbf{x} > 0$ for which $f(\mathbf{x})$ achieves a minimum. There is a unique minimum when condition *(ii)* is satisfied. When condition *(iii)* is the result, additional work must be done to find the global minimum.

It is interesting to note that when condition *(ii)* is satisfied, we simply solve for **y** by

$$\mathbf{y} = A^{-1}\mathbf{b}. \qquad ...(31.17)$$

Thus the optimal solution is obtained (in terms of **y**) by carrying out simple algebraic manipulations.

We now proceed to simplify the expression for optimum value of the objective function, *i.e.*, min. $f(\mathbf{x})$.

31.6. TO FIND EXPRESSION FOR MINIMUM $F(\mathbf{x})$

We know that at the optimal solution

$$f(\mathbf{x}^o) = P_j/y_j = c_j \prod_{i=1}^{k} (x_i)^{a_{ij}} / y_j \text{ [from (31.13)]}$$

Raising both sides to the power, 'y_j', and taking the product, we find

$$\prod_{j=1}^{n} f(\mathbf{x}^o)^{y_j} = \prod_{j=1}^{n} \left[\left(\frac{c_j}{y_j} \right) \prod_{i=1}^{k} (x_i)^{a_{ij}} \right]^{y_j} \qquad ...(31.18)$$

The left-hand side reduces to $f(\mathbf{x}^o)$ because $\sum_1^n y_j = 1$,

$$\prod_{j=1}^{n} f(\mathbf{x}^o)^{y_j} = [f(\mathbf{x}^o)]^{y_1 + y_2 + \dots + y_n} = f(\mathbf{x}^o)$$

Since all products are finite, the orders of multiplication may be reversed on the right-hand side of (31.18) to give us

$$\prod_{j=1}^{n} \left[\left(\frac{c_j}{y_j} \right) \prod_{i=1}^{k} (x_i)^{a_{ij}} \right]^{y_j} = \prod_{j=1}^{n} \left(\frac{c_j}{y_j} \right)^{y_j} \prod_{j=1}^{n} \left(\prod_{i=1}^{k} (x_i)^{a_{ij}} \right)^{y_j}$$

$$= \prod_{j=1}^{n} \left(\frac{c_j}{y_j} \right)^{y_j} \prod_{i=1}^{k} (x_i) \sum_j a_{ij} y_j$$

$$= \prod_{j=1}^{n} \left(\frac{c_j}{y_j} \right)^{y_j} \prod_{i=1}^{k} x_i^0 \quad \text{[by virtue of (31.14)]}$$

$$= \prod_{j=1}^{n} \left(\frac{c_j}{y_j} \right)^{y_j} \qquad ...(31.19)$$

The $\min f(\mathbf{x}) = f(\mathbf{x}^o) = \prod_{j=1}^{n} \left(\frac{c_j}{y_j} \right)^{y_j}$, and therefore

$$f(\mathbf{x}) \geq \prod_{j=1}^{n} \left(\frac{c_j}{y_j} \right)^{y_j}, \qquad \text{...(31.20)}$$

where y_j must satisfy the orthogonality and normality conditions derived earlier.

When there is a unique solution for **y** (condition *(ii)* is satisfied), the problem is solved except for calculating the values of the x_i from

$$c_j \prod_{i=1}^{k} (x_i)^{a_{ij}} = y_j f(\mathbf{x}^0). \qquad \text{...(31.21)}$$

When condition *(iii)* is satisfied, we must have

$$\max \prod_{j=1}^{n} (c_j/y_j)^{y_j}, \text{ subject to } A\mathbf{y} = \mathbf{b},$$

since $\min f(\mathbf{x}) = \min \prod_{j=1}^{n} (c_j/y_j)^{y_j}$.

The above procedure shows that the solutions to the original polynomial $f(\mathbf{x})$ can be transformed into the solution of a set of linear equations in y_j. We observe that y_j's are determined from the necessary conditions for a minimum. It can be shown, however, that these conditions are also sufficient. The proof may be seen in *Wilde* and *Beightler* [*, p. 5.66] and hence it is not reproduced here.

The y_j -variables actually define the dual variables associated with the above $f(x)$ -primal. In order to show this relationship, consider the primal problem in the form

$$f(\mathbf{x}) = \sum_{j=1}^{n} y_j \, (P_j/y_j).$$

Now define the function

$$f(\mathbf{y}) = \prod_{j=1}^{n} \left(P_j/y_j \right)^{y_j}$$

Since $\sum_{j=1}^{n} y_j = 1$ and $y_j > 0$, then by ***Cauchy's inequality*** ******, we have $f(\mathbf{y}) \leq f(\mathbf{x})$.

The function $f(\mathbf{y})$ with its variables $y_1, y_2, ..., y_n$ defines the dual problem to the above primal. Also, by *duality theorem we have*

$$\max_{y_j} f(\mathbf{y}) = \min_{x_i} f(\mathbf{x}).$$

31.7. ILLUSTRATIVE EXAMPLES

Example 1. *When $n = k + 1$, solve the problem :*

Minimize $z_x = 7x_1 x_2^{-1} + 3x_2 x_3^{-2} + 5x_1^{-3} x_2 x_3 + x_1 x_2 x_3$ *and* $x_1, x_2, x_3 \geq 0$ *by geometric programming method.*

Solution.

***Step* 1.** The function z_x may be written as

$$z_x = 7x_1 x_2^{-1} x_3^0 + 3x_1^0 x_2^1 x_3^{-2} + 5x_1^{-3} x_2^1 x_3^1 + x_1^1 x_2^1 x_3^1.$$

so that

---

* Wilde, D., and C. Beightler, *Foundations of optimization,* Engliewood Cliffs, N.J Prentice-Hall, 1967.

** The Cauchy's inequality states that for $z_j > 0$,

$$\sum_{j=1}^{n} w_j z_j \geq \prod_{j=1}^{n} (z_j)^{w_j} \quad \text{where } w_j > 0 \text{ and } \sum_{j=1}^{n} w_j = 1.$$

This is called arithmetic-geometric mean inequality.

$$(c_1, c_2, c_3, c_4) = (7, 3,5, 1) \text{ and} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -3 & 1 \\ -1 & 1 & 1 & 1 \\ 0 & -2 & 1 & 1 \end{bmatrix}.$$

***Step* 2.** The orthogonality and normality conditions are thus given by

$$\begin{bmatrix} 1 & 0 & -3 & 1 \\ -1 & 1 & 1 & 1 \\ 0 & -2 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

or equivalently,

$$\begin{aligned} y_1 - 0y_2 - 3y_3 + y_4 &= 0 \\ -y_1 + y_2 + y_3 + y_4 &= 0 \\ 0y_1 - 2y_2 + y_3 + y_4 &= 0 \\ y_1 + y_2 + y_3 + y_4 &= 1 \end{aligned}$$

This gives the unique solution, $y_1^* = 1/2, y_2^* = 1/6, y_3^* = 5/24, y_4^* = 3/24$.

***Step* 4.** Thus,

$$z_x^* = \left(\frac{7}{12/24}\right)^{12/24} \left(\frac{3}{4/24}\right)^{4/24} \left(\frac{5}{5/24}\right)^{5/24} \left(\frac{1}{3/24}\right)^{3/24} = 761/50.$$

***Step* 5.** Since $z_x^* = \min. f(\mathbf{x}) = P_j/y_j^*$ or $P_j = y_j^* z_x^*$, then

$$7x_1x_2^{-1} = P_1 = \frac{1}{2}\left(\frac{761}{50}\right) = \frac{761}{100}, \quad 3x_2x_3^{-2} = P_2 = \frac{1}{6}\left(\frac{761}{50}\right) = \frac{127}{50}.$$

$$5x_1^{-3}x_2x_3 = P_3 = \frac{5}{24}\left(\frac{761}{50}\right) = \frac{317}{100}, \quad x_1x_2x_3 = P_4 = \frac{1}{8}\left(\frac{761}{50}\right) = \frac{19}{10}.$$

The solution of these equations is given by $x_1^* = 1315$, $x_2^* = 1.21$, $x_3^* = 1.2$,
Which gives the optimal solution to the primal problem.

Example 2. *When $n > k + 1$, solve the problem :*

$$Min.\ z_x = 5x_1x_2^{-1} + 2x_1^{-1}x_2 + 5x_1 + x_2^{-1}$$

by geometric programming.

Solution.

***Step* 1.** The orthogonality and normality conditions are given by

$$\begin{bmatrix} 1 & -1 & 1 & 0 \\ -1 & 1 & 0 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

***Step* 2.** Since $n > k+1$, these equations do not give the required y_j directly. Thus solving for y_1, y_2 and y_3 in terms of y_4, we obtain

$$\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 \\ y_4 \\ 1 - y_4 \end{bmatrix},$$

or equivalently, $y_1 = 1/2\,(1 - 3y_4)$, $y_2 = 1/2\,(1 - y_4)$, $y_3 = y_4$.

***Step* 3.** Now, the corresponding dual problem may be written as

$$\text{Max. } z_y = \left(\frac{5}{1/2\,(1 - 3y_4)}\right)^{(1-3y_4)/2} \left(\frac{2}{1/2(1 - y_4)}\right)^{(1-y_4)/2} \times \left(\frac{5}{y_4}\right)^{y_4} \left(\frac{1}{y_4}\right)^{y_4}.$$

This becomes a problem of maxima of one variable only. So our forward technique of differential calculus may be easily applied. Taking logarithm on both sides, we get $F(y_4)$ equal to

First Iteration Table

		$c_j \rightarrow$	3	6	9	12	2	4	6	8	0	0
B	c_B	x_B	w_{12}	w_{13}	w_{14}	w_{15}	w_{22}	w_{23}	w_{24}	w_{25}	s_1	w_{11}
s_1	0	0	[4]	16	36	64	−15	−12	−7	0	1	0
w_{11}	0	1	1	1	1	1	0	0	0	0	0	1
w_{25}	8	1	0	0	0	0	1	1	1	1	0	0
	z = 8		−3 ↑	−6	−9 ×	−12 ×	6	4	2	0	0 ↓	0

Form this table, we observe that any of the variables w_{12}, w_{13}, w_{14} and w_{15} corresponding to Δ_1, Δ_2, Δ_3 and Δ_4 respectively can enter the basis. But, in order to satisfy the additional restriction, we decide that w_{12} enters the basis and s_1 leaves it. Thus, we get the following second iteration table.

Second Iteration Table

		$c_j \rightarrow$	3	6	9	12	2	4	6	8	0	0
B	c_B	x_B	w_{12}	w_{13}	w_{14}	w_{15}	w_{22}	w_{23}	w_{24}	w_{25}	s_1	w_{11}
w_{12}	3	0	1	4	9	16	−15/4	−3	−7/4	0	1/4	0
w_{11}	0	1	0	−3	−8	−15	15/4	3	7/4	0	−1/4	1
w_{25}	8	1	0	0	0	0	1	1	1	1	0	0
	z = 8		0	6	18	36	−21/4 ↑	−5	−13/4	0	3/4	0 ↓

Introducing w_{24} and droping w_{11}, we get the following third iteration table.

Third Iteration Table

		$c_j \rightarrow$	3	6	9	12	2	4	6	8	0
B	c_B	x_B	w_{12}	w_{13}	w_{14}	w_{15}	w_{22}	w_{23}	w_{24}	w_{25}	s_1
w_{12}	3	1	1	1	1	1	0	0	0	0	0
w_{24}	6	4/7	0	12/7	32/7	60/7	15/7	12/7	1	0	−1/7
w_{25}	8	3/7	0	12/7	32/7	60/7	−8/7	−8/7	0	1	1/7
	z = 69/7		0	3/7	22/7	67/7	12/7	4/7	0	0	9/14

Since all $\Delta_j \geq 0$, the optimal solution of the reduced LPP is :

$$w_{12} = 1,\ w_{24} = 4/7,\ \text{and } w_{25} = 3/7, \text{ remaining } w\text{'s are zero.}$$

***Step* 7. To calculate solution of the original problem.**

The optimal solution to the given non-linear programming problem can be obtained by the formula :

$$x_j^* = \sum_{k=1}^{5} a_{jk}\, w_{jk},\ j = 1, 2.$$

Thus, we get

$$x_1^* = a_{11}w_{11} + a_{12}w_{12} + a_{13}w_{13} + a_{14}w_{14} + a_{15}w_{15} = (0)(0) + (1)(1) + (2)(0) + (3)(0) + 4(0) = 1$$

$$x_2^* = a_{21}w_{21} + a_{22}w_{22} + a_{23}w_{23} + a_{24}w_{24} + a_{25}w_{25} = (0)(0) + (1)(0) + (2)(0) + 3(4/1) + 4(3/7) = 24/7$$

Hence the optimal solution to the given problem is finally obtained as :

$$x_1^* = 1,\ x_2^* = 24/7,\ \text{max. } z = 69/7.$$

EXAMINATON PROBLEMS

Solve the following problems by separable programming algorithm :

1. Max. $z = (x_1 - 2)^2 + (x_2 - 2)^2 2.$
   subject to the constraints :
   $x_1 + 2x_2 \leq 4$
   $x_1, x_2 \geq 0$

2. Max. $z = 16 - 2(x_1 - 3)^2 - (x_2 - 7)^2$
   subject to the constraints :
   $x_1^2 + x_2 \leq 16$
   $x_1, x_2 \geq 0$

[**Ans.** $x_1 = 1.6$, $x_2 = 1.2$, Max. $\mathbf{z} = 0.8$]

[**Hint.** Let $f_1(x_1) = 8 - 2(x_1 - 3)^2$ and $f_2(x_2) = 8 - (x_2 - 7)^2$ and proceed in the usual manner.]
[**Ans.** $x_1 = 3$, $x_2 = 7$, max. $\mathbf{z} = 16$]

3. Min. $\mathbf{z} = (x_1 - 2)^2 + (x_2 - 1)^2$
subject to the constraints :

(i) $-x_1^2 + x_2 \geq 0$
$-x_1 - x_2 + 2 \geq 0$
$x_1, x_2 \geq 0.$

[**Ans.** $x_1 = 1$, $x_2 = 1$, min. $\mathbf{z} = 1$]

(ii) $x_1 - 2x_2 + 1 = 0.$
$-\frac{1}{4} x_1^2 - x_2^2 + 1 \geq 0$
$x_1, x_2 \geq 0.$

[**Ans.** $x_1 = 0.82$, $x_2 = 0.94$, min. $\mathbf{z} = 1.4$]

4. Show how the following problem can be made separable :
Max. $\mathbf{z} = x_1x_2 + x_3 + x_1x_3$
subject to
$x_1x_2 + x_2 + x_1x_3 \leq 10,$
$x_1, x_2, x_3 \geq 0.$

5. Consider the problem :
Max. $\mathbf{z} = x_1x_2x_3$, subject to
$x_1^2 + x_2 + x_3 \leq 4$, and $x_1, x_2, x_3 \geq 0$
Approximate the problem as a linear programming model for use with the restricted basis method.

6. Find the minimum of $f(\mathbf{x}) = (x_1 + 1)^2 + (x_2 - 2)^2$, such that
$x_1 - 2 \leq 0$, $x_2 - 1 \leq 0$, and $x_1, x_2 \geq 0$.

7. Maximize $\mathbf{z} = 3x_1^2 + 2x_2^2$, such that $x_1^2 + x_2^2 \leq 25$, $9x_1 - x_2^2 \leq 27$, and $x_1, x_2 \geq 0$
Solve the above problem for x_1 and x_2 and find the optimum value of the objective function.

8. Consider the NLPP : Min. $\mathbf{z} = x_1^2 + 2x_2^2 - 2x_1$, subject to the constraints : $x_1^2 + x_2^2 \leq 4$, and $x_1, x_2 \geq 0$.
Is this problem a convex programming problem ? If not, indicate how will you proceed to solve this problem.

9. Show that the non-linear non-convex programming problem of minimizing

$$f(\mathbf{x}) = a_0 + b_{01} x_1 + \left(\sum_{j=2}^{5} b_{0j} x_j \right) x_1 \text{, subject to the cosntraints :}$$

$$0 \leq a_{i1}x_1 + \left(\sum_{j=1}^{5} a_{ij}x_j \right) x_1 \leq b_i \, (i = 1, 2, 3,), \; l_i \leq x_i \leq u_i, \, j = 1, 2, 3, 4, 5$$

can be transformed into (a convex) LPP by setting
$y_j = x_j x_1$ $(j = 1, \dots, 5)$ and $y_1 = x_1$, where
$a_0, b_{oj}, a_{ij}, b_i, l_i$, and u_i are real constants.

***

31.8. FORMULATION OF GEOMETRIC PROGRAMMING PROBLEM : WITH EQUALITY CONSTRAINTS

We shall now discuss the case when we wish to minimize an objective function which is a sum of polynomials subject to equality constraints of the same form, *i.e.*,

Minimize $z_x = f(\mathbf{x})$, subject to $g_i(\mathbf{x}) = \sum_{r=1}^{P_i} C_{ir} P_{ir}(\mathbf{x}) = 1,\ i = 1,\ldots,m$

where P_i, denotes the number of terms in the ith constraint and

$$P_{ir}(\mathbf{x}) = \prod_{j=1}^{k} (x_j)^{a_{irj}}.$$

31.9. TO OBTAIN 'NORMALITY' AND 'ORTHOGONALITY' CONDITIONS

Although the notations are somewhat awkward, the concept is the same as we have discussed previously. We first form the *Lagrange* function,

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i [g_i(\mathbf{x}) - 1]$$

and require

$$(i)\ \frac{\partial L}{\partial x_l} = 0 = \frac{\partial f(\mathbf{x})}{\partial x_l} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i(\mathbf{x})}{\partial x_l},\ l = 1,\ldots,k \qquad (ii)\ \frac{\partial L}{\partial \lambda_i} = 0 = g_i(\mathbf{x}) - 1,\ i = 1,\ldots,m.$$

We note here that our constraints are of the form $g_i(\mathbf{x}) = 1$. Thus, infact , so long as the right-hand side is positive we may obtain this form by a simple linear transformation. The case when $g_i(\mathbf{x}) = 0$ is not permissible, because our solution space requires $\mathbf{x} > 0$. When the right-hand side is negative, solution procedures have been obtained, However, the arguments are beyond the scope of this presentation. The interested students may consult the ***Selected References.***

Let us investigate the condition *(i)* in more detail.

$$\frac{\partial L}{\partial x_l} = 0 = \sum_{j=1}^{n} \frac{a_{lj} P_j(\mathbf{x})}{x_l} + \sum_{i=1}^{m} \lambda_i \left[\sum_{r=1}^{P_i} \frac{a_{irl} P_{ir}(\mathbf{x})}{x_l} \right].$$

We may again introduce variables y_j and y_{ir} as follows :

We may define,
$$y_j = \frac{P_j}{f(\mathbf{x}^0)},\ y_{ir} = \frac{\lambda_i P_{ir}}{f(\mathbf{x}^0)} \qquad \ldots(31.22)$$

Again, remember that
$$\sum_{j=1}^{n} y_j = 1. \qquad \ldots(31.23)$$

Furthermore, we have orthogonality conditions

$$\sum_{j=1}^{n} a_{lj} y_j + \sum_{i=1}^{m} \sum_{r=1}^{P_i} a_{irl} y_{ir} = 0,\ l = 1,\ldots,k \qquad \ldots(31.24)$$

This condition comes from substituting the definitions of y_j and y_{ir} into $\partial L/\partial x_l = 0$.

In the unconstrained case, the y_j were all positive, since

$$y_j = P_j/f(\mathbf{x}^0) > 0. \qquad \ldots(31.25)$$

In the equality-constrained case, the y_j are again positive. However, the y_{ir} may be negative, because we do not require λ_i to be non-negative. It is desirable to have all $y_{ir} > 0$ to construct a dual function. Also, we note here that if we reverse the order to construct the *Lagrange* function, the sign of the *Lagrange multipliers* will change. Hence, if we face one problem where one of the λ_{ir} is negative, we can reverse its sign simply by writing that term in the *Lagrange* function as $\lambda_q [1 - g_q(\mathbf{x})]$.

$$\log z_y = \frac{1-3y_4}{2}[\log 10 - \log(1-3y_4)] + \frac{1-y_4}{2}[\log 4 - \log(1-y_4)] + y_4[\log 5 - \log y_4 + (\log 1 - \log y_4)]$$

The value of y_4 maximizing $\log z_y$ must be unique (because the primal problem has a unique minimum). Hence, differentiating with repect to y_4 we get.

$$\frac{\partial F}{\partial y_4} = -\frac{3}{2}\log 10 - \left[\left(-\frac{3}{2}\right) + \left(-\frac{3}{2}\right)\log(1-3y_4)\right] + (-½)\log 4 - [(-½) + (-½)\log(1-y_4)]$$
$$+ \log 5 - [(1+\log y_4) + \log 1 - [1 + \log y_4].$$

But, by necessary condition of maxima and minima, we must have

$$\partial F/dy_4 = 0.$$

Thus, after simplification, we get

$$-\log\left[\frac{2\times 10^{3/2}}{5}\right] + \log\frac{(1-3y_4)^{3/2}(1-y_4)^{1/2}}{y_4^2} = 0, \text{ or } \frac{\sqrt{[(1-3y_4)^3(1-y_4)]}}{y_4^2} = 12.6,$$

which gives $y_4^* \cong$ 0.16. Hence $y_3^* = 0.16$, $y_2^* = 0.42$ and $y_1^* = 0.26$

The value of $\quad z_x^* = z_y^* = \left(\frac{5}{.26}\right)^{.26}\left(\frac{2}{.42}\right)^{.42}\left(\frac{5}{.16}\right)^{.16}\left(\frac{1}{.16}\right)^{16} \cong 9.506$.

Hence $\quad P_3 = .16 \times 9.506 = 1.52 = 5x_1$, $P_4 = .16 \times 9.506 = 1.52 = x_2^{-1}$.

The solution here gives $\quad x_1^* = .304$ and $x_2^* = .66$.

Example 3. (Inventory problem). *In the economic lot-size problem :*

$$\textit{Minimize } f(q) = ½ C_1 q + C_3 R/q,$$

to find the optimum iventory level.

Solution. In this example, $P_1 = q^1, a_{11} = 1, P_2 = q^{-1}, a_{12} = -1$

we have two y's one for each term. Forming the normality condition, we obtain $y_1 + y_2 = 1$.

The orthogonality condition $\sum_{j=1}^{2} a_{1j} = 0$ gives $1.y_1 + (-1)y_2 = 0$ or $y_1 - y_2 = 0$.

From this, we deduce that $y_1 = y_2 = ½$

$$f(q^*) = \left(\frac{C_1/2}{y_1}\right)^{y_1}\left(\frac{RC_3}{y_2}\right)^{y_2} = \left(\frac{C_1/2}{½}\right)^{1/2}\left(\frac{RC_3}{½}\right)^{1/2} = \sqrt{(2C_1C_3R)}$$

and from the defining equation for y_1, we have

$$y_1 f(q^*) = \frac{1}{2}C_1P_1 = \frac{1}{2}C_1 q$$

Thus $½\sqrt{(2C_1C_3R)} = ½ C_1 q$, or $q^* = \sqrt{(2C_3R/C_1)}$

Alternatively this formula has been proved in '*Inventory Management*'.

EXAMINATION PROBLEMS

Using geometric programming, solve the following problems :

1. Minimize $f(x) = c_1x_1^{-1}x_2^{-1}x_3^{-1} + c_2x_2x_3 + c_3x_1x_3 + c_1x_1x_2$ where $c_i > 0$, $x_j > 0$, $i = 1, 2, 3, 4$ and $j = 1, 2, 3$.
   [**Ans.** Min. $f(\mathbf{x}) = (⅝c_1)^{2/5}(5c^2)^{1/5}(5c_4)^{1/5}$]
2. Minimize $f(\mathbf{x}) = 5x_1x_2^{-1}x_3^2 + x_1^{-2}x_2^{-1} + 10x_2^2 + 2x_1^{-1}x_2x_3^{-2}$, and $x_1, x_2, x_3 \geq 0$
   [**Ans.** Min. $f(\mathbf{x}) = 10.28$, $x_1 = 1.26$, $x_2 = 0.41$, $x_3 = 0.59$]
3. Minimize $f(\mathbf{x}) = 2x_1 + 4x_2 + \frac{10}{x_1x_2}$ subject to $x_1, x_2 \geq 0$.
   [**Ans.** Min $f(\mathbf{x}) = 112.9$; $x_1 = 14.1$ and x_2 ='23]
4. Min. $z = 4x_1 + x_1x_2^{-1} + 4x_1^{-1}x_2$ subject to $x_1, x_2 \geq 0$.
5. Min. $z = 40x_1^{-1}x_2^{-1}x_3^{-1} + 40x_2x_3 + 20x_1x_2 + 10x_1x_3$; $x_1, x_2, x_3 \geq 0$.
   [**Ans.** $x_1 = 2$, $x_2 = 1$, $x_3 = ½$, min z = 100]
6. Max. $z = 2x_j^{-1}x_2^2 + x_1^4x_2^{-2} + 4x_1^2$ subject to $x_1, x_2 \geq 0$.
   [**Ans.** The necessary conditions are not satisfied for $x_1, x_2 \geq 0$. The problem has an infimum at $x_j = 0$, *i.e.* $z \to 0$ as $x_j \to 0$ for all j.

31

GEOMETRIC PROGRAMMING

31.1. INTRODUCTION

In this chapter, we shall focus our attention on a rather interesting technique called *'geometric programming'* for solving a special type of non-linear problems. This technique is initially derived from inequalities rather than the calculus and its extensions. This technique was given the name *'geometric programming'* because the geometric-arithmetic mean inequality was the basis of its development. Geometric programming, developed by *R. Duffin* and *C. Zener* (1964), finds the solution to the problem by considering an associated dual problem (to be defined later). The advantage here is that it is usually much simpler to work with the dual problem than with the primal.

This chapter will present the unconstrained case of geometric programming, and to do this, we shall derive the inequality using the classical optimization theorem developed in chapter 27 of this unit. Then using the inequality, we shall indicate how these relationships may be used to obtain optimal solutions to non-linear problems. It will be observed that when the problem has a special structure, the solution may be obtained simply by solving a set of linear equations.

The objective here is only to familiarize the readers with this type of analysis. Those interested in more details may refer to the excellent book by *Wilde* and *Beightler* (see the references) for a more detailed treatment of the subject.

31.2. FORMULATION OF GEOMETRIC PROGRAMMING PROBLEM (UNCONSTRAINED TYPE)

The objective and constraint functions in the problem that geometric programming deals with are of the following type.

We wish to maximize
$$z = f(\mathbf{x}) = \prod_{j=1}^{n} x_j \quad \text{...(31.1)}$$

subject to
$$\sum_{j=1}^{n} x_j = c < \infty, \quad x_j \geq 0,\ j = 1, ..., n.$$

The maximum will obviously not occur where any of the $x_j = 0$ since $f(x)$ is also zero at this point. For the moment we shall ignore these inequalities and solve the simpler problem :

$$Max.\, f(x) = \prod_{j=1}^{n} x_j \quad \text{...(31.2)}$$

$$\text{subject to} \quad \sum_{j=1}^{n} x_j = c < \infty$$

31.3. TO FIND GEOMETRIC-ARITHMETIC MEAN INEQUALITY

The given problem is : Max. $f(\mathbf{x}) = \prod_{j=1}^{n} x_j$, subject to $\sum_{j=1}^{n} x_j = c < \infty$.

Forming the *Lagrange* function, we obtain

$$L(x, \lambda) = f(x) + \lambda \left(\sum_{j=1}^{n} x_j - c \right) \qquad ...(31.3)$$

The necessary conditions are

$$\frac{\partial L}{\partial x_j} = \prod_{j=1, j \neq i}^{n} x_j + \lambda = 0, \; i = 1, 2, ..., n. \qquad ...(31.4)$$

Solving for λ, we find
$$\lambda = -\prod_{j=1, j \neq i}^{n} x_j \qquad ...(31.5)$$

But, i can be any integer from 1 to n and so we can write

$$\lambda = -\prod_{j=1, j \neq k}^{n} x_j \qquad ...(31.6)$$

Now, equating the two results (31.5) and (31.6), we obtain

$$x_1 x_2 \ldots x_{i-1} \, x_{i+1} \ldots x_n = x_1 x_2 \ldots x_{k+1} \ldots x_n.$$

Since we have assumed $x_i \neq 0$, we obtain $x_k^o = x_i^o = a$ for all i, $k = 1, 2, ...n$ where a is some constant.

But,
$$\sum_{j=1}^{n} x_j = c = \sum_{j=1}^{n} a = na.$$

Thus,
$$x_i^o = a = c/n, \; i = 1, ..., n \text{ and } f(\mathbf{x}^0) = \prod_{j=1}^{n} \left(\frac{c}{n}\right) = \left(\frac{c}{n}\right)^n.$$

We have thus proved that
$$\text{Max.} f(\mathbf{x}) = (c/n)^n.$$

Then it follows that
$$f(\mathbf{x}) \leq (c/n)^n, \qquad ...(31.7)$$

where $c = \sum_{j=1}^{n} x_j$.

Therefore,
$$f(\mathbf{x}) = \prod_{j=1}^{n} x_j \leq \left(\sum_{j=1}^{n} x_j / n \right)^n \qquad ...(31.8)$$

Now, taking nth root of each side gives

$$\left(\prod_{j=1}^{n} x_j \right)^{1/n} \leq \frac{1}{n} \sum_{j=1}^{n} x_j \qquad ...(31.9)$$

with equality only when $x_j = c/n$. This is the geometric-arithmetic mean inequality.

We have obtained an upper bound on $f(\mathbf{x})$ or looking at the problem differently, a lower bound on $\left(\frac{1}{n}\right) \sum_{j=1}^{n} x_j$, where

$$\frac{1}{n} \sum_{j=1}^{n} x_j \geq \left(\prod_{j=1}^{n} x_j \right)^{1/n} \qquad ...(31.10)$$

Optimization problems may be approached from either viewpoint. We again encounter the dual relationship discussed in **chapter 7** of author's book ***'Linear Programming and The Theory of Games'***. The dual problem will, in many cases, be easier to solve.

31.4. MORE GENERAL FORMULATION OF GEOMETRIC PROGRAMMING PROBLEM (UNCONSTRAINED TYPE)

A more general form of the inequality (31.10) is

$$\sum_{j=1}^{n} y_j x_j \geq \prod_{j=1}^{n} (x_j)^{y_j},$$

where y_j's are non-negative weights whose sum is unity. *Zener, Duffin,* and *Peterson* used this result to derive the geometric programming relationships. However, we shall pursue an argument more closely related to classical optimization theory. This approach is suggested by *Wilde* and *Beighter* (1967).

We are again, in certain cases, able to construct a highly non-linear problem as one of solving a system of linear equations,

$$\sum_{j=1}^{n} y_j = 1 \text{ (normality)}, \qquad \sum_{j=1}^{n} a_{lj}\, y_j + \sum_{i=1}^{m} \sum_{r=1}^{P_i} a_{irl}\, y_{ir} = 0;\ l = 1,\ldots,k \text{ (orthogonality)}$$

When these equations have a unique solution, the optimal solution of the original problem has been obtained. All that is required is that $f(\mathbf{x}^o)$ and $\mathbf{x}$ be calculated from the definition of y_j and y_{ir}. In the case when there are infinite number of solutions, we must again resort to maximizing the dual function given by

$$\mathbf{z}_y = f(\mathbf{y}) = \prod_{j=1}^{n}\left(\frac{c_j}{y_j}\right)^{y_j} \prod_{i=1}^{m}\left[\sum_{r=1}^{P_i}\left(\frac{C_{rj}}{y_{rj}}\right)^{y_{rj}}\right] \prod_{i=1}^{m} (V_i)^{v_i} \qquad \ldots(31.26)$$

where $V_i = \sum_{r=1}^{P_i} y_{ir}$ subject to *orthogonality* and *normality* conditions.

Although the function in (31.26) may seem to be very complicated to work with, it will appear to be much easier to handle than the original problem. The reason is that the constraints are now linear. In addition, we may have a choice to work with the algorithm of the dual function which is linear in the variable $\delta_j = \log y_j$ and $\delta_{ir} = \log y_{ir}$. The following illustrative example will make all these concepts clear.

31.10. ILLUSTRATIVE EXAMPLE

Example 4. *Solve the geometric programming problem*

$$\textit{Min } \mathbf{z}_x = 2x_1x_2^{-3} + 4x_1^{-1}x_2^{-2} + 32/3\, x_1x_2, \textit{ subject to } 10x_1^{-1}x_2^2 = 1.$$

Solution.

***Step* 1.** The corresponding dual function is given by

$$\mathbf{z}_y = \left(\frac{2}{y_1}\right)^{y_1}\left(\frac{4}{y_2}\right)^{y_2}\left(\frac{32}{3y_3}\right)^{y_3}\left(\frac{0.1}{y_4}\right)^{y_4}(y_4)^{y_4}$$

and the constraints are :

$$y_1 + y_2 + y_3 = 1,\quad y_1 - y_2 + y_3 - y_4 = 0,\quad -3y_1 - 2y_2 + y_3 + 2y_4 = 0.$$

***Step* 2.** Expressing each y_j in terms of y_1, we obtain $y_2 = 1 - (4/3)\, y_1$, $y_3 = 1/3\, y_1$, $y_4 = (8/3)\, y_1 - 1$.

$$\text{Thus, } \mathbf{z}_y = f(y_1) = \left(\frac{2}{y_1}\right)^{y_1}\left(\frac{4}{(1-(4/3)\,y_1)}\right)^{1-(4/3)\,y_1}\left(\frac{32}{y_1}\right)^{1/3\,y_1}(0.1)^{(8/3)\,y_1 - 1}$$

***Step* 3.** Now working with this maxima-minima problem of single variable, we take logarithm of both sides of

$$F(y_1) = \log\,[f(y_1)] = y_1 \log\frac{2}{y_1} + (1 - 4/3\, y_1)\log\frac{4}{1 - 4/3\, y_1} + \frac{y_1}{3}\log\frac{32}{y_1} + \left(\frac{8}{3}y_1 - 1\right)\log 0.1$$

Differentiating w.r.t. y_1, we obtain

$$\frac{dF}{dy_1} = \log\frac{2}{y_1} + 2 - \frac{46y_{12}}{3} + \log\frac{32}{y_1} + \frac{8}{3}\log 0.1$$

which becomes zero at $y_1 = 0.662$. Thus, we get $y_2 = 0.217$, $y_3 = 0.221$, $y_4 = 0.766$.

***Step* 4.** Now, with the help of these variables, we can compute x_1, x_2 and $f(\mathbf{x})$ from the definition of y_j given by eqn. (31.22) :

$$y_1 = \frac{P_1}{f(\mathbf{x}^o)} = \frac{2x_1x_2^{-2}}{f(\mathbf{x}^o)},\quad y_2 = \frac{P_2}{f(\mathbf{x}^o)} = \frac{4x_1^{-1}x_2^{-2}}{f(\mathbf{x}^o)},\quad y_3 = \frac{P_3}{f(\mathbf{x}^o)} = \frac{32x_1x_2}{3f(\mathbf{x}^o)},\quad y_4 = \frac{P_4}{f(\mathbf{x}^o)} = \frac{0.1\,\lambda_a x_a^{-1}x_2^{-2}}{f(\mathbf{x}^o)}$$

Dividing y_1 by y_3, we get $y_1/y_3 = 3 = (3/16)\, x_2^{-4}$ or $x_2^4 = 1/16$ or $x_2 = 1/2$.

From the costraint, we know that $0.1\, x_1\, (1/2)^{-2} = 1$ or $x_2 = 2.5$.

The values are consistent with the values obtained by using the definition of y_2 and y_4.

31.10. PROBLEM WITH INEQUALITY CONSTRAINT

The similar conditions and dual function can be obtained when there are inequality constraints of the from $g_i(\mathbf{x}) \leq 1$. The proof of the orthogonality conditions is left as an exercise for the students. The dual function is again obtained through the use of inequalities. The interested ones should consult the ***Selected References*** to see further as to how one arrives at the dual function (31.26).

We can use the dual problem [(31.26), (31.27)] to obtain bounds on the optimal solution. Since

$$f(\hat{\mathbf{x}}) \geq f(\mathbf{x}^o) \geq f(\hat{\mathbf{y}}) \qquad ...(31.28)$$

We can select any feasible **x**, say $\hat{\mathbf{x}}$, and any feasible **y**, say $\hat{\mathbf{y}}$, and now the optimal solution is bounded by

$$f(\hat{\mathbf{x}}) \geq f(\mathbf{x}^o) \geq f(\hat{\mathbf{y}}). \qquad ...(31.29)$$

This gives convenient stopping rule in the case when there is no unique solution to the orthogonality and normality conditions, and we approach the problem using some direct methods to maximize the dual.

EXAMINATION PROBLEMS

Solve the following problems by geometric programming :

1. Min. $\mathbf{z}_x = 2x_1^{-1}x_2^2 + x_1^4x_2^{-2} + 4x_1^2$, and $x_1, x_2 \geq 0$.
2. Min. $\mathbf{z}_x = 5x_1x_2^{-1}x_3^2 + x_1^{-2}x_3^{-1} + 10x_2^3 + 2x_1^{-1}x_{2_3}^{-2}$, and $x_1, x_2, x_3 > 0$
3. Min. $\mathbf{z}_x = 2x_1^2x_2^{-3} + 8x_1^{-3}x_2 + 3x_1x_2$, and $x_1, x_2 > 0$.
4. Min. $\mathbf{z}_x = 2x_1^3x_2^{-3} + 4x_1^{-2}x_2 + x_1x_2 + 8x_1x_2^{-1}$, and $x_1, x_2 \geq 0$.
5. Min. $\mathbf{z}_x = 5.x_1/x_2 + 10x_1^2x_2 + 3/x_1$, and $x_1, x_2 \geq 0$.
6. Min. $\mathbf{z}_x = x_1x_2/x_3^2 + 2\, x_3/x_1x_2 + 5x_3$.
7. Set-up the necessary conditions to solve the following problem by geometric programming :

   Min. $\mathbf{z}_x = 3x_1/x_2 + x_2^2/x_1 + x_1^2x_2$, subject to $\frac{1}{4}.\, x_1^2/x_2 + \frac{1}{9}x_2x_1 = 1$, $2(1/x_1^2) + 4(x_2/x_1^3) = 2$.

✱✱✱

32

Fractional Programming

32.1. INTRODUCTION

In this chapter, we shall discuss a newly developed important technique of mathematical programming which is named as *Linear Fractional Programming.* This technique is used to solve the problem of maximizing the fraction of two linear functions subject to a set of linear equalities and the non-negativity constraints. This problem can be directly solved by starting with a basic feasible solution and showing the conditions for improving the current basic feasible solution. In order to test optimality of the solution we shall establish the optimality criterion. Ultimately, the problem can be easily solved by the method which is similar to *'Simplex Method'* of linear programming.

32.2. IMPORTANCE OF FRACTIONAL PROGRAMMING IN PRACTICAL SITUATIONS

The linear fractional programming problems have recently been a topic of great importance in non-linear programming. *J.R. Isbel* and *W.H. Marlow* (1956) discussed an example of fractional programming. In *Military,* programming games have this form when troops are in the field and the decision to be taken is how to distribute the fire among several possible types of targets. The fractional programming method is useful in solving the problem in *Economics* whenever the different economic activities utilize the fixed resources in proportion to the level of their values. In financial analysis of a firm , the purpose of optimization is to find the optimum of a specific index number, usually the most favourable ratio of revenues and allocation. Therefore, such type of problems play an important role in *'finance'*

32.3. MATHEMATICAL FORMULATION OF LINEAR FRACTIONAL PROGRAMMING PROBLEM

Mathematically, the *linear fractional programming problem* can be formulated as follows :

$$Max.\ \mathbf{z} = (\mathbf{c}'\,\mathbf{x} + \alpha)/(\mathbf{C}'\,\mathbf{x} + \beta) \qquad ...(32.1)$$

subject to the constraints :

$$A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0, \qquad ...(32.2)$$

where *(i)* **x, c,** and **C** are $n \times 1$ column vectors , *(ii)* A is an $m \times n$ matrix, *(iii)* **B** is an $m \times 1$ column vector

(iv) The primes {$'$ } over the vectors **c** and **C** denote the transpose of vectors, and

(v) α, β are some scalars.

Further, it is assumed that the constraints set $S = \{\mathbf{x} \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$ is non-empty and bounded.

Charnes and **Cooper** (1962) solved the above problem by resolving it into two ordinary linear programming problems (under transformation $\mathbf{y} = t\mathbf{x}$) :

Prob. 1. Max. $\mathbf{c}'\mathbf{y} + \alpha t$, subject to the constraints :

$$A\mathbf{y} - \mathbf{b}t = 0, \mathbf{d}'\mathbf{y} + \beta t = 1, \mathbf{y} \geq 0, t \geq 0.$$

Prob. 2. Max. $- \mathbf{c}'\mathbf{y} - \alpha t$, subject to the constraints :

$$A\mathbf{y} - \mathbf{b}t = 0, \ -\mathbf{d}'\mathbf{y} - \beta t = 1, \mathbf{y} \geq 0, t \geq 0.$$

But in the following section, we develop an algorithm for the solution of programming problems with linear fractional functions without converting it into linear programming problems.

32.4. LINEAR FRACTIONAL PROGRAMMING ALGORITHM

Let us consider the linear fractional programming problem :

$$Max.\ \mathbf{z} = (\mathbf{c}'\mathbf{x} + \alpha)/(C'\mathbf{x} + \beta)$$

subject to the constraints :

$$A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq 0,$$

with the additional assumption that the denominator is positive for all feasible solutions.

(a) Notations used. Let $\mathbf{x}_B$ denote the starting basic feasible solution such that

$$B\mathbf{x}_B = \mathbf{b} \text{ or } \mathbf{x}_B = B^{-1}\mathbf{b}, \mathbf{x}_B \geq 0,$$

where $B = (\beta_1, \beta_2, ..., \beta_m)$.

Further, we let $\mathbf{z}^{(1)} = \mathbf{c}'_B\mathbf{x}_B + \alpha$ and $\mathbf{z}^{(2)} = C'_B\mathbf{x}_B + \beta$, where $\mathbf{c}'_B$ and $\mathbf{C}'_B$ are the vectors having their components as the coefficients associated with the basic variables in the numerator and the denominator of the objective function, respectively. Also for basic feasible solution, we assume that

$$\mathbf{x}_j = B^{-1}\mathbf{a}_j,\ \mathbf{z}_j^{(1)} = \mathbf{c}'_B\mathbf{x}_j,\ \mathbf{z}_j^{(2)} = \mathbf{C}'_B\mathbf{x}_j$$

are obtainable for every column $\mathbf{a}_j$ belonging to A but not to B.

(b) How to improve the initial basic feasible solution. In order to examine the possibility of determining another basic feasible solution with the improved value of the objective function $\mathbf{z} = \mathbf{z}^{(1)}/\mathbf{z}^{(2)}$, we are concerned only to those basic feasible solutions in which only one column of basis matrix B is changed.

Let the new basic feasible solution be denoted by $\hat{\mathbf{x}}_B$. Then

$$\hat{\mathbf{x}}_B = \hat{B}^{-1}\mathbf{b},\text{ where } \hat{B} = (\hat{\beta}_1, \hat{\beta}_2, ..., \hat{\beta}_m).$$

That is, a new non-singular matrix is obtained from B by replacing β_r by $\mathbf{a}_j$. Thus, the columns of the new matrix $\hat{\mathbf{B}}$ are given by

$$\hat{\beta}_i = \beta_i\ (i \neq r),\ \hat{\beta}_r = a_j.$$

Now, we find the value of the new basic variables in terms of original ones and the x_{ij}, *i.e.*

$$\hat{x}_{Bi} = x_{Bi} - \frac{x_{Bi}}{x_{rj}} x_{Br}\ (i \neq r),\quad \hat{x}_{Br} = \frac{x_{Br}}{x_{rj}} = \theta \text{ (say)}$$

where $\mathbf{a}_j = \sum_{i=1}^{m} x_{ij}\,\beta_i$.

After determining the new basic feasible solution it remains to justify whether '$\mathbf{z}$' is improved. For this, value of the objective function for the original basic feasible solution is $\mathbf{z} = \mathbf{z}^{(1)}/\mathbf{z}^{(2)}$.

Let the new value of the objective function be $\hat{\mathbf{z}} = \hat{\mathbf{z}}^{(1)}/\hat{\mathbf{z}}^{(2)}$.

Therefore, we have

$$\hat{\mathbf{z}}^{(1)} = \mathbf{z}^{(1)} - \theta(\mathbf{z}_j^{(1)} - c_j) \quad\text{and}\quad \hat{\mathbf{z}}^{(2)} = \mathbf{z}^{(2)} - \theta(\mathbf{z}_j^{(2)} - C_j)$$

where $\mathbf{z}_j^{(1)}$ and $\mathbf{z}_j^{(2)}$ are associated with the original basic feasible solutions.

Now the value of the new objective function will improve if

$$\frac{\mathbf{z}^{(1)} - \theta(\mathbf{z}_j^{(1)} - c_j)}{\mathbf{z}^{(2)} - \theta(\mathbf{z}_j^{(2)} - C_j)} > \frac{\hat{\mathbf{z}}^{(1)}}{\hat{\mathbf{z}}^{(2)}} \quad\text{or}\quad \frac{\mathbf{z}^{(1)} - \theta(\mathbf{z}_j^{(1)} - c_j)}{\mathbf{z}^{(2)} - \theta(\mathbf{z}_j^{(2)} - C_j)} - \frac{\hat{\mathbf{z}}^{(1)}}{\hat{\mathbf{z}}^{(2)}} > 0,$$

$$\hat{\mathbf{z}}^{(2)}[z^{(1)} - \theta[\mathbf{z}_j^{(1)} - c_j)] - \hat{\mathbf{z}}^{(1)}[\mathbf{z}^{(2)} - \theta(\mathbf{z}_j^{(2)} - C_j)] > 0$$

(since denominator of the objective function is positive for all feasible solutions, *i.e.* $\hat{\mathbf{z}}^{(2)}$ and $\mathbf{z}^{(2)}$ are positive)

or $$\hat{\mathbf{z}}^{(1)}[\mathbf{z}^{(2)} - \theta(\mathbf{z}_j^{(2)} - C_j)] - \hat{\mathbf{z}}^{(2)}[\mathbf{z}^{(1)} - \theta(\mathbf{z}_j^{(2)} - c_j)] < 0$$

($\theta = x_{Br}/x_{rj}$ is positive in the non-degenerate case, and if $\theta = 0$, then $\hat{\mathbf{z}} = \mathbf{z}$)

Let us denote $\quad \Delta_j = \mathbf{z}^{(1)}[\mathbf{z}^{(2)} - \theta(\mathbf{z}_j^{(2)} - C_j)] - \mathbf{z}^{(2)}[\mathbf{z}^{(1)} - \theta(\mathbf{z}_j^{(1)} - c_j)]$ (say)

Now, $\Delta_j > 0$ under the following three cases.

Case 1. If $\mathbf{z}_j^{(2)} - C_j > 0$, then $$\frac{(\mathbf{z}_j^{(1)} - c_j)}{(\mathbf{z}_j^{(2)} - C_j)} < \frac{\mathbf{z}^{(1)}}{\mathbf{z}^{(2)}}$$

Case 2. If $z_j^{(2)} - C_j < 0$, then $\dfrac{(z_j^{(1)} - c_j)}{(z_j^{(2)} - C_j)} > \dfrac{z^{(1)}}{z^{(2)}}$

Case 3. If $z_j^{(2)} - C_j = 0$, then $z_j^{(1)} - c_j > 0$.

We now prove the following *theorem :*

Theorem. *Given a basic feasible solution* $\mathbf{x}_B = B^{-1}\mathbf{b}$. *If for any column vector* $\mathbf{a}_j$ *in A but not in B,* $\Delta_j > 0$ *holds, and if at least one* $x_{ij} > 0$ $(i = 1, 2, \ldots m)$, *then it is possible to obtain a new basic feasible solution by replacing one of the columns in B by* $\mathbf{a}_j$ *and the new value of the objective function satisfies* $\hat{\mathbf{z}} \geq \mathbf{z}$.

Proof. We now wish to show that for any $\mathbf{a}_j$ in A but not in B at least one $x_{ij} > 0$.

If possible, let us suppose that all $x_{ij} \leq 0$ $(i = 1, 2, \ldots, m)$. The basic feasible solution is given by

$$\sum_{i=1}^{m} x_{Bi}\, \beta_i = \mathbf{b} \qquad \text{...(i)}$$

Now suppose that we add and subtract $\theta \mathbf{a}_j$ (θ is any scalar) to (i,) then we have

$$\sum_{i=1}^{m} x_{Bi}\, \beta_i + \theta\, \mathbf{a}_j - \theta\, \mathbf{a}_j = \mathbf{b} \qquad \text{...(ii)}$$

Since

$$-\theta\, \mathbf{a}_j = -\theta \sum_{i=1}^{m} x_{ij}\, \beta_i, \qquad \text{...(iii)}$$

using (iii) in (ii), we have

$$\sum_{i=1}^{m} (x_{Bi} - \theta x_{ij})\beta_i + \theta\, \mathbf{a}_j = \mathbf{b}.$$

If $\theta > 0$, then $x_{Bi} - \theta x_{ij} \geq 0$.

Since by assumption $x_{ij} \leq 0$ $(i = 1, 2, \ldots, m)$,

$$x_{B1} - \theta x_{1j},\ x_{B2} - \theta x_{2j}, \ldots, x_{Bm} - \theta x_{mj},$$

and $\theta = x_{Br}/x_{rj}$ is a feasible solution for all $\theta > 0$. Thus, the set S of feasible solutions is unbounded contrary to our hypothesis.

Thus, we have proved in the algorithm that if we begin by basic feasible solution and if the vector $\mathbf{a}_j$ is in A but not in basis having

$$\Delta_j < 0, \qquad \text{...(iv)}$$

then we can get another basic feasible solution such that $\hat{\mathbf{z}} \geq \mathbf{z}$.

In the absence of degeneracy $\hat{\mathbf{z}}$ is strictly greater than $\mathbf{z}$, *i.e.* $\hat{\mathbf{z}} > \mathbf{z}$. This means that we can move from one basic feasible solution to another, changing one vector at a time so long as there exist some $\mathbf{a}_j$ in A but not in B under the condition $\Delta_j < 0$, and at each iteration $\mathbf{z}$ is improved (*i.e.* increased in the case of maximization).

(c) Convergence of algorithm. The algorithm cannot continue indefinitely. The reason is that there exists only a finite number of basic feasible solutions and in the absence of degeneracy no basis can ever be repeated, because $\mathbf{z}$ is improved at every step and the same solution cannot yield two distinct values of $\mathbf{z}$, while at the same time the optimum has to occur at one of the basic feasible solutions. So the process will terminate only when all $\Delta_j \geq 0$ for the columns of A but not in B. But, for the columns of A belonging to B, we have

$$\mathbf{z}_j^{(1)} = \mathbf{c}'_B \mathbf{X}_j = \mathbf{c}'_B B^{-1} \mathbf{a}_j = \mathbf{c}'_B B^{-1}\beta_j = c_j,$$

and

$$\mathbf{z}_j^{(2)} = \mathbf{C}'_B \mathbf{X}_j = \mathbf{C}'_B B^{-1} \mathbf{a}_j = \mathbf{C}'_B B^{-1}\beta_j = C_j.$$

Hence $\Delta_j = \mathbf{z}^{(1)} (\mathbf{z}_j^{(2)} - c_j) - \mathbf{z}^{(2)} (\mathbf{z}_j^{(1)} - c_j)$

(d) Summary of above discussion. Now the results obtained from the above discussion can be summarized as follows.

If the problem : Max. $\mathbf{z} = (\mathbf{c}'\, \mathbf{x} + \alpha)/(\mathbf{C}'\, \mathbf{x} + \beta)$ subject to $A\mathbf{x} = \mathbf{b}$. $\mathbf{x} \geq 0$, has a basic feasible solution $\mathbf{x}_B = B^{-1}\mathbf{b}$ with

$$\mathbf{z}^* = (\mathbf{c}'_B\, \mathbf{x}_B + \alpha)/(\mathbf{C}'_B\, \mathbf{x}_B + \beta)$$

such that $\Delta_j \geq 0$ for every column $\mathbf{a}_j$ in A, then $\mathbf{z}^*$ will be the maximum value of $\mathbf{z}$ and the basic feasible solution will be an optimum solution.

32.5. COMPUTATIONAL PROCEDURE OF FRACTIONAL ALGORITHM

The following numerical example can better explain the computational procedure of linear fractional programming algorithm.

Example. *Consider the fractional programming problem*

$$Max.\ \mathbf{z} = \frac{5x_1 + 3x_2}{5x_1 + 2x_2 + 1},\ subject\ to:$$

$$3x_1 + 5x_2 \leq 15\ ,\ \ 5x_1 + 2x_2 \leq 10,\ \ and\ \ x_1, x_2 \geq 0.$$

Solution. Introducing the slack variables $x_3 \geq 0$ and $x_4 \geq 0$, the problem in the standard form becomes :

$$\text{Max. } \mathbf{z} = \frac{5x_1 + 3x_2}{5x_1 + 2x_2 + 1} = \frac{\mathbf{z}^{(1)}}{\mathbf{z}^{(2)}} \text{ (say) , subject to :}$$

$$3x_1 + 5x_2 + x_3 = 15,\ \ 5x_1 + 2x_2 + x_4 = 10,\ \ \text{and}\ \ x_1, x_2, x_3, x_4 \geq 0.$$

For the starting table, we find $\Delta_1 = -5, \Delta_2 = -3, x_1 = x_2 = 0$. We choose min. Δ_j (Δ_1 in this case). Thus, $\mathbf{z}$ can be increased by taking $\mathbf{x}_1$ in the basis. The method to determine the leaving variable and also the new values of x_{ij}, $\mathbf{x}_B$, $\Delta_j^{(1)}$, $\Delta_j^{(2)}$ corresponding to improved solution will be the same as for ordinary simplex method. Thus, x_4 will be the departing variable. Here $\alpha = 0$ and $\beta = 1$.

Starting Table

Basic Var.	C_B	c_B	$\mathbf{x}_B$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3(\beta_1)$	$\mathbf{x}_4(\beta_2)$	Min. ($\mathbf{x}_B/\mathbf{x}_1$)
			$c_j \rightarrow$	5	3	0	0	
			$C_j \rightarrow$	5	2	0	0	
x_3	0	0	15	3	5	1	0	15/3
$\leftarrow x_4$	0	0	10	[5]	2	0	1	10/5 $\leftarrow$
	$\mathbf{z}^{(1)} = \mathbf{c}_B\mathbf{x}_B + \alpha = 0$			-5	-3	0	0	$\leftarrow \Delta_j^{(1)}$
	$\mathbf{z}^{(2)} = \mathbf{C}_B\mathbf{x}_B + \beta = 1$			-5	-2	0	0	$\leftarrow \Delta_j^{(2)}$
	$\therefore\ \mathbf{z} = \mathbf{z}^{(1)}/\mathbf{z}^{(2)} = 0$			-5	-3	–	–	$\leftarrow \Delta_j$
				$\uparrow$			$\downarrow$	

First Iteration Table

Introducing $\mathbf{x}_1$ and dropping $\mathbf{x}_4(\beta_2)$, we get the following table :

Basic Var.	C_B	c_B	$\mathbf{x}_B$	$\mathbf{x}_1(\beta_2)$	$\mathbf{x}_2$	$\mathbf{x}_3(\beta_1)$	$\mathbf{x}_4$	Min. ($\mathbf{x}_B/\mathbf{x}_2$)
			$c_j \rightarrow$	5	3	0	0	
			$C_j \rightarrow$	5	2	0	0	
x_3	0	0	9	0	[19/5]	1	$-3/5$	9/19/5 = 45/19 $\leftarrow$
x_1	5	5	2	1	2/5	0	1/5	2/2/5 = 5
	$\mathbf{z}^{(1)} = \mathbf{c}_B\mathbf{x}_B = 10$			0	-1	0	1	$\leftarrow \Delta_j^{(1)}$
	$\mathbf{z}^{(2)} = \mathbf{C}_B\mathbf{x}_B + \beta = 11$			0	0	0	1	$\leftarrow \Delta_j^{(2)}$
	$\mathbf{z} = \mathbf{z}^{(1)}/\mathbf{z}^{(2)} = 10/11$			–	-11	–	1	$\leftarrow \Delta_j$
					$\uparrow$	$\downarrow$		

Second Iteration Table

Introducing $\mathbf{x}_2$ and dropping $\mathbf{x}_3$ (β_1), we get the following table :

			$c_j \rightarrow$	5	3	0	0	
			$C_j \rightarrow$	5	2	0	0	
Basic Var.	$\mathbf{C}_B$	$\mathbf{c}_B$	$\mathbf{x}_B$	$\mathbf{x}_1(\beta_2)$	$\mathbf{x}_2(\beta_1)$	$\mathbf{x}_3$	$\mathbf{x}_4$	Min. $(\mathbf{x}_B/\mathbf{x}_4)$
x_2	2	3	45/19	0	1	5/19	−3/19	–
$\leftarrow x_1$	5	5	20/19	1	0	−2/19	5/19	$\frac{29}{19}/\frac{5}{19} \leftarrow$
	$\mathbf{z}^{(1)} = \mathbf{c}_B\mathbf{x}_B + \alpha = 235/19$			0	0	5/19	16/19	$\leftarrow \Delta_j^{(1)}$
	$\mathbf{z}^{(2)} = \mathbf{C}_B\mathbf{x}_B + \beta = 209/19$			0	0	0	1	$\leftarrow \Delta_j^{(2)}$
	$\therefore\ \mathbf{z} = \mathbf{z}^{(1)}/\mathbf{z}^{(2)} = 235/209$			–	–	1045/361	−1121/361	$\leftarrow \Delta_j$
				↓			↑	

Final Table

Introducing $\mathbf{x}_4$ and removing $\mathbf{x}_1$ (β_2) we get the following table.

			$c_j \rightarrow$	5	3	0	0	
			$C_j \rightarrow$	5	2	0	0	
Basic Var.	C_B	c_B	$\mathbf{x}_B$	$\mathbf{x}_1$	$\mathbf{x}_2(\beta_1)$	$\mathbf{x}_3$	$\mathbf{x}_4(\beta_2)$	
x_2	2	3	3	3/5	1	1/5	0	
x_4	0	0	4	19/5	0	−2/5	1	
	$\mathbf{z}^{(1)} = \mathbf{c}_B\mathbf{x}_B + \alpha = 9$			−16/5	0	3/5	0	$\leftarrow \Delta_j^{(1)}$
	$\mathbf{z}^{(2)} = \mathbf{C}_B\mathbf{x}_B + \beta = 7$			−19/5	0	2/5	0	$\leftarrow \Delta_j^{(2)}$
	$\mathbf{z} = \mathbf{z}^{(1)}/\mathbf{z}^{(2)} = 9/7$			59/5	–	3/5	–	$\leftarrow \Delta_j$

Since all $\Delta_j \geq 0$, we have reached the optimum solution : $x_1 = 0, x_2 = 3,\ x_4 = 4$, Max. $\mathbf{z} = 9/7$.

EXAMINATION PROBLEMS

Solve the following linear fractional programming problems :

1. Max. $\mathbf{z} = \dfrac{2x_1 + 3x_2}{x_1 + x_2 + 7}$, subject to the constraints :
$3x_1 + 5x_2 \leq 15,\ 4x_1 + 3x_2 \leq 12$ and $x_1, x_2 \geq 0$.

2. Max. $\mathbf{z} = \dfrac{-3x_1 - x_2}{x_1 + 2x_2 + 5}$, subject to the constraints :
$x_1 + x_2 \geq 1,\ 2x_1 + 3x_2 \geq 2$, and $x_1, x_2 \geq 0$.

3. Max. $\mathbf{z} = \dfrac{2x_1 + 3x_2}{5x_1 + 7x_2 + 4}$, subject to the constraints :
$3x_1 + x_2 \leq 4,\ x_1 + x_2 \leq 1$. and $x_1, x_2 \geq 0$

4. Min. $\mathbf{z} = \dfrac{-x_1 + 2x_2}{5x_1 + 3x_2 + 2}$, subject to the constraints :
$3x_1 + 6x_2 \leq 8,\ 5x_1 + 2x_2 \leq 10$ and $x_1, x_2 \geq 0$.

5. Write short note on linear fractional programming.

✱✱✱

33

DYNAMIC PROGRAMMING

33.1. INTRODUCTION

Dynamic programming is a mathematical technique dealing with the optimization of multistage decision process. The word *'programming'* has been used in the mathematical sense of selecting an optimum allocation of resources, and it is *'dynamic'* as it is particularly useful for problems where decisions are taken at several distinct stages, such as everyday or every week. ***Richard Bellman*** developed this technique in early 1950 and invented its name. Dynamic programming can be given a more significant name as ***recursive optimization.*** In dynamic programming, a large problem is splitted into small sub-problems each of them involving only a few variables. This technique converts one problem of n variables into n sub-problems (stages), each in one variable. The optimum solution is obtained in an orderly manner starting from one stage to the next, and is completed till the final stage is reached.

To convert a verbal problem into a multistage structure is not always simple, and sometimes it becomes very difficult and even looks easy to apply. Recursion equations are of standard nature and its computer program runs in a standard routine.

An important point is that—the problem of successive stages be treated separately even though by the very nature of the problem these stages are dependent ? The answer of this question is based on ***'Bellman's Principle of Optimality'*** which is stated in the following section.

Discrete and continuous, deterministic as well as probabilistic models can be solved by this method. Thus dynamic programming method is very useful for solving various problems, such as inventory, replacement, allocation, linear programming, etc. A single constraint problem is relatively simple, but in the problem of more than two constraints more complexities appear.

(i) While solving the problem we use the concepts of stage and state. Moreover, the problem is solved stage by stage and to ensure that suboptimal solution does not result, we **cummulated** the objective function value in a particular way. Working backwards, for every stage, we found the decisions in that stage that will allow us to reach the final destination optimally, starting from each of the states of the stage. These decisions could be taken optimally, without the knowledge of how we actually reach the different states. This has been stated as the *''principle of optimality in dynamic programming literature.''*

(ii) State : The variable that links up two stages is called a state variable. At any stage, the status of the problem can be descibed by the values the state variable can take. These values are referred to as states.

(iii) Stage : The points at which decisions called for are referred to as stages. Each stage can be thought of having a beginning and an end. The different stages come in a sequence, with the ending of a stage marking the beginning of the immediately succeeding stage.

Q. Explain the concepts (not exeeding three sentenses for each) (a) Principle of optimality (b) state (c) stage.

33.2. DECISION TREE AND BELLMAN'S PRINCIPLE OF OPTIMALITY

Decision Tree. *A multistage decision system, in which each decision and state variable can take only finite number of values, can be represented graphically by a 'decision tree'.*

In *Fig.* 33.1, circles representing nodes correspond to *stages* and line between circles denoting arcs correspond to the *decisions.* The node at the top of the tree is the starting node, and there are three possible decisions that can be made.

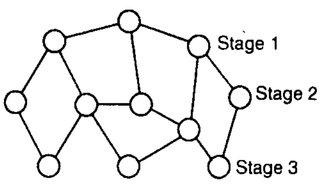

Fig. 33.1

This is represented by three arcs emanating from the node. Associated with each arc is a ***return*** and an ***output***, *i.e.*, a resulting stage. These three nodes, in turn, represent input to next stage 2. Now, there are three 'starting' nodes and the same process is repeated. This process will continue until all stages are converted. A set of arcs, which starts from node '*start*' and end in the last stage, is a *feasible path*. The return from this path is the sum of returns (or *product of returns*) from the arcs in the path. The objective is to find the path which yields maximum return.

To find an optimal path, start with four input nodes at stage 3, and find an arc from each of them which maximize the return. Take $f_3(x)$ as the return from this stage and $D_3(x)$ as the decision when someone is in a stage x, *i.e.*, node x.

Now, consider a two-stage system consisting of stage 3 and 2 having three input nodes. Find optimal paths and returns from each of these nodes to the end of stage 3. For example, consider a node at stage 2. There are two arcs emerging from it and the out-put node is input for stage 3. From earlier calculations, optimal paths are known from stage 3 to the top of 3. Thus, to find the optimal value for a given node, find an arc which maximizes returns from the arc combined with the optimal return from the output node. Once, these are calculated, the same concept could be extended to a three-stage system to determine optimal path from '*start*' to the '*end*' of decision tree. The fundamental concept is only to consider the optimal return from output nodes, instead of considering returns that are not optimal, with respect to out-put nodes. *Ideally, if an optimal solution is obtained for a system, any portion of it must be optimal. This is called the '**Bellman's Principle of Optimality**' on which the concept of dynamic programming is based.*

Bellman's Principle of Optimality : **[Meerut (OR) 2003, 02; Kanpur 96; Raj. (M. Phil) 92, 91]**

"*An optimal policy (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*".

Mathematically, this can be written as

$$f_N(x) = \max_{d_n \in \{x\}} \left[r(d_n) + f_{N-1}\{T(x, d_n)\} \right]$$

where $f_N(x)$ = the optimal return from an N-stage process when initial state is x

$r(d_n)$ = immediate return due to decision d_n

$T(x, d_n)$ = the transfer function which gives the resulting state

$\{x\}$ = set of admissible decisions.

Consider the implication of this principle as a multistage decision problem. *The problem which does not satisfy the principle of optimality cannot be solved by the dynamic programming method.*

Q. 1. State the *'Principle of Optimality'* in dynamic programming and give a mathematical formulation of a dynamic programming problem. **[Meerut (Maths) 98, 91]**

2. State and explain *Bellman's* principle of optimality in dynamic programming.
[JNTU (B. Tech.) 2002; Meerut (Maths.) 99; Tamilnadue B.E. (Resource Man.) 97]

3. Explain *Bellman's* principle of optimality and give classical formulation and the dynamic programming formulation of any problem. **[Rajaesthan (M. Phil) 93]**

33.3. SOLUTION OF PROBLEM WITH FINITE NUMBER OF STAGES

The solution of problems by dynamic programming is usually done in two stages :

(*i*) The development of functional equations for the problem.

(*ii*) To solve functional equations for determining the optimal policy.

Unlike linear programming, there does not exist a standard mathematical formulation of the dynamic programming problem. This is a general type of approach to problem solving, and functional equations used must be developed to fit the individual situation. Dynamic programming theory, however, develops the so called '*functional equation approach*' which offers a unifying, but not fixed, format of expressing the decision problem mathematically. This ability of deriving functional equations can probably be developed by an exposure to a wide variety of dynamic programming applications and a study of characteristics which are common to all of these situations. To understand, a large number of examples are presented in this chapter.

33.4. CONCEPT OF DYNAMIC PROGRAMMING

*Consider an optimal sub-division problem where a positive quantity **b** is to be divided into **n** parts. The object is to determine the optimum sub-division of **b** in order to maximize the product of **n** parts.*

This problem can be solved by using the method of *Lagrangian* multipliers, but in more complicated example, simultaneous equations resulting from the classical calculus aproach may be extremely difficult to solve. Also the calculus approach cannot be applicable if non-differentiable functions are involved. If it is possible to reformulate the n-variable problem as a series of n problems each in one variable, then computational procedure is expected to be reduced to some extent.

The dynamic programming approach removes these difficulties by first breaking the problem into smaller sub-problems, and each sub-problem is referred to as a *stage.* A stage signifies a portion of the decision problem for which a '*separate*' decision can be taken. The resulting decision will also be meaningful if it is optimal for the stage it represents and can be used directly as a part of the optimal solution to the problem. In general, number of stages in a problem may be finite or infinite.

The computational efficiency of dynamic programming stems from the fact that the optimum solution can be obtained by converting the problem into stages and then considering one stage at a time. For example, in an inventory problem, there are some situations where a policy of producing each month to minimize the inventory cost for the month immediately affected will minimize the inventory cost for the whole year.

In order to understand the step-by-step (*iterative*) procedure in dynamic programming, a few *Dynamic Programming Models* are discussed in a systematic manner.

Q. 1. Explain a dynamic programming problem. **[Meerut 2002]**

2. What is dynamic programming and what sort of problems can be solved by it ? State and establish Bellman's Principle of Optimality.

3. State the principle of optimality in dynamic programming. Describe the basic features which characterize a dynamic programming problem.

33.5. MODEL I : MINIMUM PATH PROBLEM

Example 1. *Once upon a time there lived Mr. Banerjee in Bombay who decided to travel from Bombay to Calcutta. In those days, stage-coach was the only means of public transportation from Bombay to Calcutta. His travel agent showed him various stage coach routes at that time available. Each block on the map presents a stage.*

Since the travelling through hostile state presented serious hazards, to life, Mr. Banerjee decided to purchase an insurance policy. The cost of policy depended upon the route he selected : greater the danger, higher the cost. Mr. Banerjee is a thrifty man and would like to spend the minimum amount of money in his trip on insurance. Finding the minimum cost policy is a very difficult problem for Mr. Banerjee, so he decided to call on his friend Mr. Bellman to see if he could help him. After spending few days on this issue, Bellman came with the following solution procedure (Mr. Banerjee insisted that he would like to see the calculations in detail).

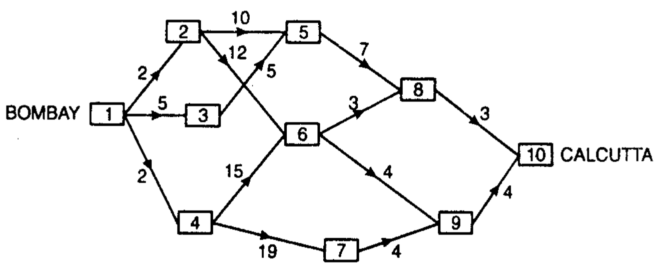

Fig. 33.2

Solution. Suppose,

$f_n(s)$ = minimum *policy cost* when he is in state s with n more stages to reach his final destination

$$= \min_{s} [r(d_n) + f_{n-1}\{T(s, d_n)\}]$$

$$f_0(s) = 0.$$

Consider any stage, say n, where he has to make a decision. Now we can use backward recursive approach.

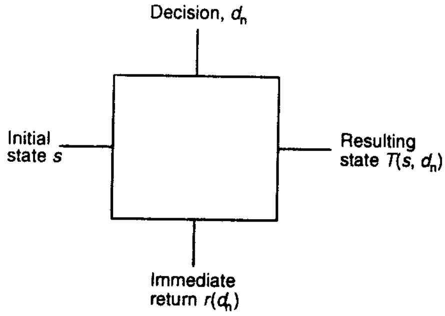

Fig. 33.3

In this example, $f_0(10) = 0$ initially (for $n = 0$).

For $n = 1, f_1(s) = \min_{d_1} [r(d_1) + 0]$, where $s = (8, 9)$.

Since, $f_1(8) = 3$ (route 8 – 10) and $f_1(9) =$ 4 (route 9 – 10), therefore for $n = 2, f_2(s) = \min_{\{d\}} [r(d_2) +$ f_1 (resulting state)] where $s = 5, 6, 7$; and

$d = \boxed{5} \rightarrow (5-8)$; $6 <^{(6-8)}_{(6-9)}$; $\boxed{7} \rightarrow (7-9)$.

The data in a tabular form can be written as follows :

Table 33.1. (For $n = 2$)

Initial state s	Decision d_n	Immediate cost $r(d_n)$	Resulting state $T(s, d_n)$	Optimal return from resulting state $f_{n-1}\{T(s, d_n)\}$	$f_n(s)$	Optimal cost policy
5	5—8	7	8	3	10*	5—8
6	6—8	3	8	3	6*	6—8
	6—9	4	9	4	8	
7	7—9	4	9	4	8*	7—9

Similarly, for $n = 3$, following table is obtained :

Table 33.2. (For $n = 3$)

s	d_n	$r(d_n)$	$T(s, d_n)$	$f_{n-1}\{T(s, d_n)\}$	$f_n(s)$	Optimal policy
2	2—5	10	5	10	20	
	2—6	12	6	6	12*	2—6
3	3—5	5	5	10	15	
	3—6	10	6	6	12*	3—6
	3—7	7	7	8	15	
4	4—6	15	6	6	12*	4—6
	4—7	13	7	8	15	

For $n = 4, f_4(s = 1) = \min_{\{d\}} [r(d_4) + f_3 \text{ (resulting state)}]$

$$= \min \begin{cases} 1\text{—}2 = 2 + 12 = 14* \\ 1\text{—}3 = 5 + 12 = 17 \\ 1\text{—}4 = 2 + 12 = 14* \end{cases}$$

Therefore, the minimum cost policies are 1—2—6—8—10 and 1—4—6—8—10. The cost of each policy is 14 units.

Ques. *Banerjee has a friend R. Chawla. who lives in State* 3. *If Banerjee wants to visit him, how much more would it cost to buy the insurance ?*

To find out the new cost, one need not go through the whole calculations again. This information is contained in the Tables. According to *Bellman's principle*, the optimal cost from State 3 to Calcutta is 12 units, and the minimum cost of going to State 3 is 5 units. Therefore, it will cost 17 units. If *Banerjee* values that his visit with *Chawla* is worth 2 units, should he visit him ?

In this *Model*, Mr. Bellman started his calculations from destination. Such a formulation is called the *backward formulation*. In this particular instance, he could have started calculations from Bombay, *i.e.* from the starting point. Then such a formulation is called the *forward formulation*. Depending upon the situation, the formulation may be *backward* or *forward*. In many cases, the backward or forward formulation is predetermined by the problem itself. *The main advantage of the forward formulation is that, if few more stages are added in future, all previous calculations could be used.*

Example 2. *Find the shortest path from vertex A to vertex B along arcs joining various vertices lying between A and B (Fig. 33.4). Length of each path is given.*

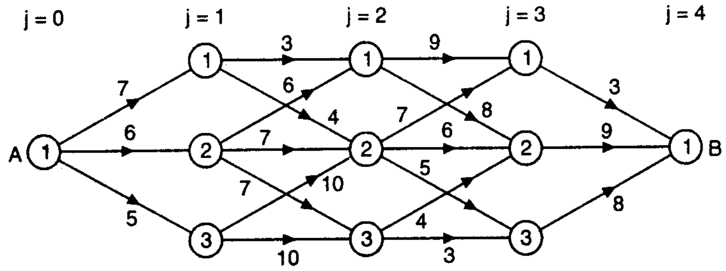

Fig. 33.4

Solution. Step 1. ***Formulation.*** Divide vertices into five stages 0, 1, 2, 3 and 4 denoted by subscript j. For $j = 0$ and $j = 4$, there is only single vertex A and B, respectively. But, for $j = 1, 2, 3$ there are three vertices in each stage. Each time one moves from stage j to stage $j + 1$, *i.e.* from any one vertex in stage j to any other vertex in stage $j + 1$.

Each move will change the *state* of the system denoted by s_j. Thus, s_0 is the state in which the node A lies. Also, s_0 has only state value, say $s_0 = 1$. State s_2 has only three possible values; say 1, 2, 3 corresponding to three vertices in stage 2, and so on. Possible alternative paths from one stage to the other will be called *decision* variables denoted by d_j (the decision which takes from state s_{j-1} to state s_j). The return or the gain which obviously being the function of decision will be denoted by $f_j(d_j)$. Here d_j can be identified with the length of the corresponding arc, and thus simplify matters by considering $f_j(d_j) = d_j$.

The minimum path from state s_0 to any vertex in state s_j will be denoted by $F_j(s_j)$. For example, $F_2(1)$ will denote the minimum path from vertex A to vertex 1 in stage 2.

Now, the problem is to find the minimum path $F_4(s_4)$, and the values of decision variables d_1, d_2, d_3 and d_4.

Step 2. ***To obtain functional equations.*** Start from vertex B *backwards*. Obviously, d_4 can either be 3 or 9 or 8. If $d_4 = 3$, then $s_3 = 1$. Similarly, $d_4 = 9 \Rightarrow s_3 = 2$; $d_4 = 8 \Rightarrow s_3 = 3$. Hence the minimum path from A to B is either through $s_3 = 1$ or 2 or 3 according as d_4 is 3, 9 or 8.

Thus, $$F_4(s_4) = \min\,[3 + F_3(1)\,,\, 9 + F_3(2)\,,\, 8 + F_3(3)] = \min_{d_4}\,[d_4 + F_3(s_3)]$$

In a similar way,

$$F_3(1) = [9 + F_2(1),\, 7 + F_2(2)],\;\; F_3(2) = [8 + F_2(1)\,,\, 6 + F_2(2),\, 4 + F_2(3)],\;\; F_3(3) = [5 + F_2(2),\, 3 + F_2(3)]$$

In general, $$F_3(s_3) = \min_{d_3}\,[d_3 + F_2(s_2)],\; s_3 = 1, 2, 3.$$

Similarly, $$F_2(s_2) = \min_{d_2}\,[d_2 + F_1(s_1)].$$

Finally, $F_1(s_1) = d_1$. The general recursion formula thus becomes

$$F_j(s_j) = \min_{d_j}\,[d_j + F_{j-1}\,(s_{j-1})],\, j = 4, 3, 2, \text{ with } F_1(s_1) = d_1.$$

Step 3. ***Determination of the minimum path.*** Now, it is possible to determine $F_4(s_4)$ recursively with the help of the recursion formula by tabulating the information given in the problem as follows :

State s_0

d_1	7	6	5
s_1	1	2	3

State s_1

d_2 \ s_2	3	4	6	7	10
1	1	—	2	—	—
2	—	1	—	2	3
3	—	—	—	2	3

State s_2

s_3 \ d_3	3	4	5	6	7	8	9
1	—	—	—	—	2	—	1
2	—	3	—	2	—	1	—
3	3	—	2	—	—	—	—

State s_3

s_4 \ d_4	3	9	8
1	1	2	3

From these tables, it is concluded that a function of the form $s_{j-1} = \psi_j(s_j, d_j)$ exists which is called the *stage transformationfunction*. It is possible that s_{j-1} may not be defined for all combinations of s_j and d_j. Such possibilities, where the transformation is not feasible, are indicated by a dash in above tables.

Now, recursive operations can be made by using recursive formulae as indicated in the following tables.

j	s_1	d_1	$F_1(s_1)$
1	1	7	7
	2	6	6
	3	5	5

		$F_1(s_1)$					$d_2 + F_1(s_1)$					$F_2(s_2)$
j	s_2 \ d_2	3	4	6	7	10	3	4	6	7	10	*Min.*
2	1	7	—	6	—	—	10	—	12	—	—	10
	2	—	7	—	6	5	—	**11**	—	13	15	11
	3	—	—	—	6	5	—	—	—	**13**	15	13

		$F_2(s_2)$							$d_3 + F_2(s_2)$							$F_3(s_3)$
j	s_3 \ d_3	3	4	5	6	7	8	9	3	4	5	6	7	8	9	*Min.*
3	1	—	—	—	—	11	—	10	—	—	—	—	**18**	—	19	18
	2	—	13	—	11	—	10	—	—	**17**	—	**17**	—	18	—	17
	3	13	—	11	—	—	—	—	**16**	—	16	—	—	—	—	16

		$F_3(s_3)$			$d_4 + F_3(s_3)$			$F_4(s_4)$
j	s_4 \ d_4	3	9	8	3	9	8	*Min.*
4	1	**18**	17	16	**21**	26	24	21

Thus, minimum path from A to B is obtained, *i.e.* $f_4(s_4) = 21$. By tracing the minimum path and decision backwards (as indicated by numbers in bold type), successive distances are 7, 4, 7, 3 through the States $s_0 = 1$, $s_1 = 1$, $s_2 = 2$, $s_3 = 1$ and $s_4 = 1$.

Q. 1. Show how the functional equation technique of dynamic programming can be used to determine the shortest route when it is constrained to pass through a set of specified nodes which is definite subset of the set of nodes of a given network.

2. State Bellman's principle of optimality and explain by an illustrative example how it can be used to solve multistage decision problems. **[Raj. Univ. (M. Phil) 90]**

33.6. MODEL II : SINGLE ADDITIVE CONSTRAINT, MULTIPLICATIVELY SEPARABLE RETURN

Consider the problem : To maximize $z = \prod_{j=1}^{n} f_j(y_j)$, subject to $\sum_{j=1}^{n} a_j y_j = b$, $y_j \geq 0$, $a_j \geq 0$.

First, introduce state variables, *i.e.* $s_j = \Sigma a_j y_j = b,\ s_{j-1} = s_j - a_j y_j,\ j = 2, 3, \ldots, n.$

Let $F_j(s_j) = \max_{y_1, y_2, \ldots, y_j} \prod_1^n f_j(y_j)$, then the general recursion formula becomes

$$F_j(s_j) = \max_{y_j} [f_j(y_j) F_{j-1}(s_{j-1})], j = n, n-1, \ldots, 2$$

or

$$F_1(s_1) = f_1(y_1).$$

Example 3. ***(Continuous Variables).*** *Find the value of max* $(y_1 y_2 y_3)$ *subject to*

$y_1 + y_2 + y_3 = 5$; $y_1, y_2, y_3 \geq 0.$ **[JNTU (Mech. & Prod.) 2004; Kanpur 2000; IAS (Maths.) 98]**

Solution. Here the state variables are

$$s_3 = y_1 + y_2 + y_3,\ s_2 = s_3 - y_3 = y_1 + y_2,\ s_1 = s_2 - y_2 = y_1$$

Also, $F_3(s_3) = \max_{y_3} [y_3 F_2(s_2)]$, $F_2(s_2) = \max_{y_2} [y_2 F_1(s_1)]$, $F_1(s_1) = y_1 = s_2 - y_2$

Hence, $F_2(s_2) = \max_{y_2} [y_2 (s_2 - y_2)]$.

Using differential calculus to maximize $y_2 (s_2 - y_2)$, we get $y_2 = s_2/2$.

Therefore, using the *Bellman's principle* of optimality

$$F_3(s_3) = \max_{y_3} [y_3 s_2^2/4] = \max_{y_3} [y_3 (s_3 - y_3)^2/4]$$

Again, using calculus, we get $y_3 = s_3/3 = 5/3$.

Also, $y_2 = 5/3$, $y_1 = 5/3$ and hence max $y_1 y_2 y_3 = 125/27$.

Example 4. (Optimal Sub-division Problem). *Divide a given quantity b into n parts so as to maximize their product. Let* $f_n(b)$ *denote the value. Show that*

$$f_1(b) = b, \text{ and } f_n(b) = \max_{0 \leq z \leq b} \{z f_{n-1}(b - z)\}$$

Hence find $f_n(b)$ *and the division that maximized it.*

[Meerut (OR) 2003, 02; Delhi (Stat.) 95; I.A.S. (Math.) 94; Meerut 93]

Solution. In dynamic programming approach, there is sequential procedure to find the optimal policy considering the last decision first and proceeding backward to the decision.

Step 1. ***To develop functional equations.*** Let y_i be the ith part of b ($i = 1, 2, \ldots, n$), and each i may be regarded as a stage. Alternatives at each stage are infinite in this case, since y_i may assume any non-negative value which satisfies $y_1 + y_2 + y_3 + \ldots + y_n = b$. This means y_i is continuous.

Also, let $f_n(b)$ denote the maximum attainable prouduct which depends on n (the number of parts into which the quantity b is to be divided) because the quantity b is fixed. Thus $f_n(b)$ becomes a function of the discrete variable n ($n = 1, 2, 3, \ldots$).

For $n = 1$, the result $f_1(b) = b$ is trivially true.

Now, consider the case for $n = 2$ in which the quatity b is divided into two parts, say $y_1 = z$ and $y_2 = b - z$,

$$\therefore\ f_2(b) = \max y_1 y_2 = \max_{0 \leq z \leq b} \{z(b - z)\} \qquad \ldots(33.4a)$$

Since $b - z = f_1(b - z)$ by the definition of f_1. Therefore,

$$f_2(b) = \max_{0 \leq z \leq b} \{z f_1(b - z)\} \qquad \ldots(33.4b)$$

Similarly, consider the case for $n = 3$. Take one of three parts as z leaving an amount $(b - z)$ for further division into two parts. By the definition of f_2 [from the equation (33.4a)], the maximum attainable product after dividing $(b - z)$ into two parts is $f_2(b - z)$. So the conditional maximum product for b divided into three parts (given the initial choice of z) is given by $z f_2(b - z)$.

Then by the principle of optimality

$$f_3(b) = \max_{0 \leq z \leq b} \{z f_2(b - z)\} \qquad \ldots(33.5)$$

Likewise, the functional equation for $n = m$ is given by

$$f_m(b) = \max_{0 \le z \le b} \{z f_{m-1}(b-z)\} \quad ...(33.6)$$

Step 2. ***To solve functional equations for determining the optimal policy.***

From the equation (33.4*a*), $f_2(b) = \max_{0 \le z \le b} [z(b-z)]$

The function $z(b-z)$ attains its maximum value for $z = \frac{1}{2} b$ satisfying the restriction $0 \le z \le b$ (using differential calculus).

Now, for $n = 2$, ***Optimal policy :*** $(\frac{1}{2} b, \frac{1}{2} b)$ and $f_2(b) = \frac{1}{4} b^2 = (\frac{1}{2} b)^2$ ...(33.7)

Since $f_2(b) = b^2/4$, $f_2(b-z) = (b-z)^2/4$. Hence, the equation (33.5) becomes

$$f_3(b) = \max_{0 \le z \le b} \left\{ z \frac{(b-z)^2}{4} \right\}$$

in which $\phi(z) = [z(b-z)^2/4]$ is known function of single variable z. The maximum value of $[z(b-z)^2/4]$ is attained for $z = b/3$.

Since $f_2(b-z) = f_2(b - \frac{1}{3} b) = f_2(\frac{2}{3} b) = [(\frac{2}{3} b)^2]/4 = (\frac{1}{3} b)^2$, then for $n = 3$,

Optimal policy : $(\frac{1}{3} b, \frac{1}{3} b, \frac{1}{3} b)$ and $f_3(b) = (\frac{1}{3} b)^3$ ...(33.8)

Further, suppose the optimal policy :

$$(b/n, b/n, b/n, \ldots, b/n)\;;\; f_n(b) = (b/n)^n \quad ...(33.9)$$

holds for $n = 2, 3, 4, \ldots, m$, then it only remains to show that this result will also hold for $n = m + 1$, thereby establishing the result by induction for general n.

Now, for $n = m + 1$, the functional equation becomes

$$f_{m+1}(b) = \max_{0 \le z \le b} \{z f_m(b-z)\} \quad ...(33.10a)$$

but $f_m(b-z) = [(b-z)/m]^m$ [from equation (33.9)], so

$$f_{m+1}(b) = \max_{0 \le z \le b} \left\{ z \left(\frac{b-z}{m} \right)^m \right\} \quad ...(33.10b)$$

in which $F(z) = z\left(\frac{b-z}{m}\right)^m$ is again a known function of single variable z. The maximum value $z[(b-z)/m]^m$ is attained for $z = b/(m+1)$*, (*see foot note*).

Therefore, $$f_m(b-z) = \left(\frac{b-z}{m}\right)^m = \left(\frac{b - \frac{b}{m+1}}{m}\right)^m = \left(\frac{b}{m+1}\right)^m$$

$$f_{m+1}(b) = \left(\frac{b}{m+1}\right)\left(\frac{b}{m+1}\right)^m = \left(\frac{b}{m+1}\right)^{m+1}$$

Thus the optimal policy will be $\left(\frac{b}{m+1}, \frac{b}{m+1}, \ldots, \frac{b}{m+1}\right)$

Hence the result (7.9) is true for general n.

---

* Since $F(z) = z\left(\frac{b-z}{m}\right)^m$, $\frac{dF}{dz} = mz\left(\frac{b-z}{m}\right)^{m-1}\left(-\frac{1}{m}\right) + \left(\frac{b-z}{m}\right)^m$

But $dF/dz = 0$ for maximum or minimum. Therefore,

$$\left(\frac{b-z}{m}\right)^{m-1}\left(-z + \frac{b-z}{m}\right) = 0$$

which gives either $z = b$ or $z = b/(m+1)$.

Further, it can be shown that $\frac{d^2F}{dz^2}$ is negative for $z = \frac{b}{m+1}$.

Therefore, maximum value of $F(z)$ is attained for $z = b/(m+1)$.

Q. 1. State *Bellman's* principle of optimality and use it to solve the problem : Max $x_1 x_2 x_3 \ldots x_n$, subject to $x_1 + x_2 + x_3 + \ldots + x_n = c$, and $x_1, x_2, x_3, \ldots, x_n \geq 0$.

2. What is *Bellman's* principle of optimality ? Apply this principle to divide a given quantity c into n parts so as to maximize their product. **[Raj. Univ. (M. Phil) 91]**

3. Use dynamic programming technique to solve the following problem :
Maximize $z = x_1 x_2 x_3 x_4$, subject to the constraints :
$x_1 + x_2 + x_3 + x_4 = 12$, $x_1, x_2, x_3, x_4 \geq 0$. **[Tamilnadue B.E. (Resource Mangt.) 97]**

4. Determine the maximum value of $z = p_1 \, p_2 \ldots p_n$, subject to the constraints : $\sum_{i=1}^{n} c_i p_i \leq x$, $0 \leq p_i \leq 1$ $(i = 1, 2, \ldots n)$

(assume that $c_i > x$ for all i). **[IAS (Maths.) 96]**

Example 5 (Discrete Variables). *A Government space project is conducting research on a certain engineering problem that must be solved before man can fly to moon safely.*

These 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 A, B and C—will not succeed are 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. *Since the objective is to minimize this probability, the decision has been made to assign two or more top scientists among the three teams in order to lower it as much as possible.*

The following table gives the estimated probability that the respective teams will fail when 0, 1 or 2 additional scientists are added to that team :

		Team		
		A	*B*	*C*
Number of New	0	0.40	0.60	0.80
Scientists	1	0.20	0.40	0.50
	2	0.15	0.20	0.30

How should the additional scientists be allocated to the team ? **[Delhi (OR) 93]**

Solution. In this problem, the research teams are corresponding to the *stages* in the dynamic programming formulation.

Step 1. ***Formulation of the Problem.*** Let

$s \rightarrow$ denote the number of new scientists still available for assignment at that stage.

$x_j \rightarrow$ the number of additional scientists allocated to team (stage j).

$p_j(x_j) \rightarrow$ denote the probability of failure for team j if it is assigned x_j additional scientists as prescribed in the table.

Then the formulation of the programming problem becomes :

$$\textit{Min } \mathbf{z} = p_1(x_1)\, p_2(x_2)\, p_3(x_3) \text{, subject to the constraints}$$

$$x_1 + x_2 + x_3 = 2, \text{ and } x_1, x_2, x_3 \geq 0,$$

where x_1, x_2, x_3 *are integers.*

Step 2. ***To obtain the recursive equations :***

Let $f_j(x_j)$ be the value of the optimal allocation for teams 1 through j both inclusive.

Thus, for $j = 1$, $\quad f_1(x_1) = \{p_1(x_1)\}$

If $f_j(s, x_j)$ be the probability associated with the optimum solution $f_j^*(s)$, $j = 1, 2, \ldots, n$, then

$$f_j(s, x_j) = p_j(x_j) \times \min\,[p_{j+1}(x_{j+1})\, p_{j+2}(x_{j+2})\, p_{j+3}(x_{j+3})]$$

such that $\sum_{i=j}^{3} x_i = s$, and x_i are non-negative integers, $j = 1, 2, 3$.

The recursive equations thus obtained are :

$$f_j^*(s) = \min_{x_j \leq s} f_j(s, x_j) \quad \text{and} \quad f_j(s, x_j) = p_j(x_j) \cdot f_{j+1}^*(s - x_j)$$

where $$f_j^*(s) = \min_{x_j \le s} [p_j(x_j) f_{j+1}(s - x_j)], j = 1, 2, 3.$$

So, when $j = 3$, $$f_3^*(s) = \min_{x_3 \le s} p_3(x_3).$$

Step 3. ***Solution of recursive equations*** :

The solution begins with $f_3^*(s)$ and completes when $f_1^*(s)$ is obtained.

Since all the quantities in the recurrence equation are discrete, the differential calculus method *cannot* be used. Let the optimal policy be denoted by x_j^* , $j = 1, 2, 3$. Now proceed backward from $j = 3$ for each stage one by one.

Computations for One-Stage Problem ($j = 3$)

s	$f_3^*(s)$	x_3^*
0	0.80	0
1	0.50	1
2	0.30	2

Computations for Two-Stage Problem ($j = 2$)

	$f_2(s, x_2) = p_2(x_2) f_3^*(s - x_2)$			Optimum Sol.	
s \ x_2	0	1	2	$f_2^*(s)$	x_2^*
0	(0.60) (0.80) = **0.48**			0.48	0
1	(0.60) (0.50) = **0.30**	(0.40) (0.80) = 0.32		0.30	0
2	(0.60) (0.30) = **0.18**	(0.40) (0.50) = 0.20	(0.20) (0.80) = **0.16**	0.16	2

Computations for Three-Stage Problem ($j = 1$)

	$f_1(s, x_1) = p_1(x_1) f_2^*(s - x_1)$			Optimum Sol.	
s \ x_1	0	1	2	$f_1^*(s)$	x_1^*
0	(0.40) (0.48) = **0.48**			0.192	0
1	(0.40) (0.30) = 0.120	(0.20) (0.48) = **0.096**		0.096	1
2	(0.40) (0.16) = 0.064	(0.20) (0.30) = **0.060**	(0.15) (0.48) = 0.072	0.060	1

Therefore, optimum solution will have $x_1^* = 1$ making $s = 1$ at the second stage, so that $x_2^* = 0$ making $s = 1$ at the third stage, so that $x_3^* = 1$.

Hence, first and third terms should each receive one additional scientist. The new probability that all the three teams will fail would then become 0.060.

Example 6 (Maximization Problem). *A truck can carry a total of 10 tons of product. Three types of product are available for shipment. Their weights and values are tabulated. Assuming that at least one of each type must be shipped determine the loading which will maximize the total value.*

Type	*Value (Rs.)*	*Weight (tons)*
A	20	1
B	50	2
C	60	2

Solution. Since there are three types of units *A*, *B* and *C* to be loaded, it is a three stage problem. Let x_j ($j = 1, 2, 3$) be the decision variable. Also, let $f_j(x_j)$ be the amount of the optimal allocation for the three products.

If $f_j(s, x_j)$ be the quantity associated with the optimum solution $f_j^*(s)$, ($j = 1, 2, \ldots, n$), then the recursive equations are

$$f_j^*(s) = \max_{0 \le x_j \le s} f_j(s, x_j) \quad \text{and} \quad f_j^*(s, x_j) = \max_{0 \le x_j \le s} [P_j(x_j) f_{j+1}^*(s - x_j)], j = 1, 2, 3,$$

where $P_j(x_j)$ denotes the expected value obtained from allocation of x_j tons of weight to the j-type product.

Now perform the following tabular computations :

First Stage (J = 3)

s_1 \ x_1	1	2	3	$f_1^*(s)$	x_1^*
2	$1 \times 60 =$ **60**	—	—	60	1
3	$1 \times 60 =$ **60**	—	—	60	1
4	60	$2 \times 60 =$ **120**	—	120	2
5	60	$2 \times 60 =$ **120**	—	120	2
6	60	120	$3 \times 60 =$ **180**	180	3
7	60	120	$3 \times 60 =$ **180**	180	3

Second Stage (J = 2)

s_2 \ x_2	1	2	3	$f_2^*(s)$	x_2^*
4	1 (50) + 60 = **110**	—	—	110	1
5	**110**	—	—	110	1
6	50 + 120 = **170**	2 (50) + 60 = 160	—	170	1
7	**170**	160	—	170	1
8	50 + 180 = **230**	100 + 120 = 220	3 (50) + 60 = 210	230	1
9	**230**	100 + 120 = 220	150 + 60 = 210	230	1

Third Stage (J = 1)

s_3 \ x_3	1	2	3	4	5	6	$f_3^*(s)$	x_3^*
100	1 (20) + 230 = 250	2 (20) + 230 = **270**	3 (20) + 170 = 230	4 (20) + 170 = 250	5 (20) + 110 = 210	6 (20) + 110 = 200	270	2

Thus the optimal solution is given by $x_1^* = 3, x_2^* = 1$ and $x_3^* = 2$ with $f_3^*(s) = 270$.

This answer interprets : product 3 tons of type *A*, one ton of type *B* and 2 tons of type *C*, *must be shipped to give the maximum value of Rs. 270.*

Example 7. *(i) A ship is to be loaded with certain items. Each unit of item i has a weight w_i and a value v_i (i = 1, 2, 3,). The maximum cargo weight permitted is W. Using the following table, determine the most valuable cargo load which will not exceed the maximum permissible weight, and W =* 10. **[Delhi (OR) 93]**

(ii) *Solve the above problem with the data and W =* 1000.

i	w_i	v_i
1	5	4
2	8	10
3	3	6

i	w_i	v_i
1	495	220
2	500	750
3	510	1012

Solution. Proceed as *Example 6.*

33.7. MODEL III : SINGLE ADDITIVE CONSTRAINT, ADDITIVELY SEPARABLE RETURN

Consider the problem in which the objective or return function z is an additively separable function of n variables y_j and $f_j(y_j)$ is a function of y_j. *Find y_j, $1 \le j \le n$, which minimize* $z = \sum_{j=1}^{n} f_j(y_j)$ subject to the constraints :

$$\sum_{j=1}^{n} a_j y_j \ge b, \; a_j \text{ and } b \text{ are real numbers, where } a_j \ge 0, y_j \ge 0, b > 0.$$

This is an n-stage problem where the suffix j indicates the *stage*. Since values of y_j are to be decided, y_j is called *decision variable.* The *return* at the jth stage is the function $f_j(y_j)$. Thus, each decision y_j is associated with a return function $f_j(y_j)$.

Now introduce *state variables* $s_0, s_1, s_2, \ldots, s_n$.

$$s_n = a_1y_1 + a_2y_2 + \ldots + a_ny_n \geq b$$
$$s_{n-1} = a_1y_1 + a_2y_2 + \ldots + a_{n-1}y_{n-1} = s_n - a_ny_n,$$
$$s_{n-2} = a_1y_1 + a_2y_2 + \ldots + a_{n-2}y_{n-2} = s_{n-1} - a_{n-1}y_{n-1}$$
$$\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots$$
$$s_1 = a_1y_1 = s_2 - a_2y_2.$$

Also, $$s_{j-1} = T_j(s_j, y_j), \; 1 \leq j \leq n$$

is the *stage transformation function* and indicates that each stage variable is a function of next state and decision variables.

$F_n(s_n)$ denotes the minimum value of z for any feasible value of s_n, where s_n being the function of all decision variables. Thus

$$F_n(s_n) = \min_{y_1, y_2, \ldots, y_n} [f_1(y_1) + f_2(y_2) + \ldots + f_n(y_n)], \; s_n \geq b.$$

First, choose a particular value of y_n and minimize z over the remaining $n-1$ variables. Hence

$$F_n(s_n) = \min_{y_1, y_2, \ldots, y_{n-1}} \left[\sum_{j=1}^{n-1} f_j(y_j) \right] = f_n(y_n) + F_{n-1}(s_{n-1})$$

Values of $y_1, y_2, \ldots, y_{n-1}$ for which $\sum_{j=1}^{n-1} f_j(y_j)$ is minimum keeping y_n fixed thus depend upon s_{n-1} which in turn is a function of s_n and y_n. Therefore, the minimum over all y_n for any feasible s_n would now become

$$F_n(s_n) = \min_{y_n} [f_n(y_n) + F_{n-1}(s_{n-1})].$$

If the value of $F_{n-1}(s_{n-1})$ is known for all y_n, the function to be minimized would involve only a single variable y_n. This minimization now becomes easy and can be done by simple methods. Similarly, the recursion formula is

$$F_j(s_j) = \min_{y_j} [f_j(y_j) + F_{j-1}(s_{j-1})], \; 1 \leq j \leq n \quad \text{and} \quad F_1(s_1) = f_1(y_1).$$

Now starting with $F_1(s_1)$ and recursively optimizing to obtain $F_2(s_2), F_3(s_3), \ldots$, we obtain $F_n(s_n)$ for each feasible s_n. Each time optimization occurs over a single variable.

Example 8. *Minimize* $z = y_1^2 + y_2^2 + y_3^2$ *subject to* $y_1 + y_2 + y_3 \geq 15$, *and* $y_1, y_2, y_3 \geq 0$.

[Meerut 2005; JNTU (Mech. & Prod.) 2004; Agra 94]

Solution. Decision variables y_1, y_2, y_3 and stage variables s_1, s_2, s_3 are defined as

$$\left\{\begin{array}{ll} s_3 = y_1 + y_2 + y_3 \geq 15 & \\ s_2 = y_1 + y_2 & = s_3 - y_3 \\ s_1 = y_1 & = s_2 - y_2 \end{array}\right\} \quad \text{and} \quad \left\{\begin{array}{l} F_3(s_3) = \min_{y_3} [y_3^2 + F_2(s_2)] \\ F_2(s_2) = \min_{y_2} [y_2^2 + F_1(s_1)] \\ F_1(s_1) = y_1^2 = (s_2 - y_2)^2 \end{array}\right\}$$

Thus $$F_2(s_2) = \min_{y_2} [y_2^2 + (s_2 - y_2)^2]$$

By calculus, $y_2^2 + (s_2 - y_2)^2$ is minimum if its derivative with respect to y_2 is zero, *i.e.* $2y_2 - 2(s_2 - y_2) = 0$ which gives $y_2 = s_2/2$. Hence $F_2(s_2) = s_2^2/2$.

Now, $$F_3(s_3) = \min_{y_3} [y_3^2 + F_2(s_2)] = \min_{y_3} [y_3^2 + (s_3 - y_3)^2/2] \quad \text{(using *Bellman's principle*)}$$

Again, using calculus, for minimum of the function of single variable y_3, $2y_3 - (s_3 - y_3) = 0$, or $y_3 = s_3/3$.

Hence, $F_3(s_3) = s_3^2/3$, $s_3 \geq 15$.

Since $F_3(s_3)$ is minimum for $s_3 = 15$, the minimum value of $y_1^2 + y_2^2 + y_3^2$ becomes 75, where $y_1 = y_2 = y_3 = 5$.

Q. Obtain the functional equations of dynamic programming for solving the problem :
Minimize $d_1^2 + d_2^2 + d_3^2$, subject to $d_1 + d_2 + d_3 = K$, $K > 0$ and $d_1, d_2, d_3 \geq 0$.

Example 9. *Use dynamic programming to show that* $-\sum_{i=1}^{n} p_i \log p_i$, *subject to* $\sum_{i=1}^{n} p_i = 1$, *is maximum, when* $p_1 = p_2 = p_3 = \ldots = p_n = 1/n$. **[Kanpur 2000; Agra 98; 97, 96; Rohilkhand 94, 93; Delhi (OR) 93; Raj. Univ. (M. Phil.) 92; I.A.S. (Main) 83]**

Solution. This problem can be considered as to divide unity into n parts, $p_1, p_2, p_3, \ldots, p_n$, such that the quantity $p_1 \log p_1 + p_2 \log p_2 + p_3 \log p_3 + \ldots + p_n \log p_n$ is minimum.

Let $f_n(1)$ denote the minimum attainable sum regarded as a function of discrete variable n (number of parts into which the unity is to be divided).

For $n = 1$, $\quad f_1(1) = p_1 \log p_1 = 1 \log 1$ (because $p_1 = 1$ only) ...(33.11)

Now consider the case for $n = 2$ in which the unity is to be divided into two parts, say $p_1 = z$ and $p_2 = 1 - z$, then

$$f_2(1) = \min_{0 \leq z \leq 1} [p_1 \log p_1 + p_2 \log p_2] \text{ or } f_2(1) = \min_{0 \leq z \leq 1} [z \log z + (1-z) \log (1-z)] \quad ...(33.12a)$$

Since, $f_1(1-z) = (1-z) \log (1-z)$ from the equation (33.11),

$$f_2(1) = \min_{0 \leq z \leq 1} [z \log z + f_1(1-z)] \quad ...(33.12b)$$

But, by simple calculus, it can be easily verified that the minimum value of the function

$$F(z) = z \log z + (1-z) \log (1-z) \text{ [from equation (33.12a)]}$$

is attained for $z = 1/2$. Thus for $n = 2$, optimal policy is given by

$$p_1 = p_2 = 1/2 \text{ and } f_2(1) = 2 (1/2 \log 1/2) \quad ...(33.13)$$

Similarly for $n = 3$, take one of the three parts as z leaving an amount $(1-z)$ for further division into two parts. Using *Bellman's principle* of optimality

$$f_3(1) = \min_{0 \leq z \leq 1} [z \log z + f_2(1-z)] \quad ...(33.14a)$$

Since $f_2(1) = 2 (1/2 \log 1/2)$ from equation (33.13), $f_2(1-z) = 2\left(\frac{1-z}{2}\right)\log \frac{1-z}{2}$

Hence the equation (33.14a) becomes

$$f_3(1) = \min_{0 \leq z \leq 1} \left[z \log z + 2\left(\frac{1-z}{2}\right)\log \frac{1-z}{2} \right] \quad ...(33.14b)$$

in which $z \log z + 2\left(\frac{1-z}{2}\right)\log \frac{1-z}{2} = F(z)$, say, is a known function of a single variable z. Again, it can be easily observed by differential calculus that minimum value of this function $F(z)$ is attained for $z = 1/3$ satisfying the restriction $0 \leq z \leq 1$.

Since, $\quad f_2(1-z) = f_2(1 - 1/3) = f_2(2/3) = 2 (2/6 \log 2/6)$ [from the equation (33.13)]

$$= 1/3 \log 1/3 + 1/3 \log 1/3.$$

Thus for $n = 3$, optimal policy is given by

$$p_1 = p_2 = p_3 = 1/3 \text{ and } f_3(1) = 3 (1/3) \log 1/3 \quad ...(33.15)$$

Further, suppose that the optimal policy $p_1 = p_2 = p_3 = \ldots = p_n = 1/n$ for which $f_n(1) = n [(1/n) \log (1/n)]$ holds for $n = 2, 3, 4, \ldots, m$. Then, it only remains to show that this result will also hold for $n = m + 1$, thus establishing the result by induction for general value of n.

By the principle of optimality,

$$f_{m+1}(1) = \min_{0 \le z \le 1} [z \log z + f_m(1-z)]$$

$$= \min_{0 \le z \le 1} \left[z \log z + m\left(\frac{1-z}{m} \log \frac{1-z}{m}\right)\right]$$

in which the function $F(z) = z \log z + (1-z) \log [(1-z)/m]$ is again a function of a single variable z. The minimum value of this function is attained for $[z = 1/(m+1)]$*.

Since, $$f_m(1-z) = f_m\left(\frac{m}{m+1}\right) = m\left(\frac{m/(m+1)}{m}\right)\log\left(\frac{m/(m+1)}{m}\right)$$

$$= \frac{m}{m+1} \log \frac{1}{m+1} = \frac{1}{m+1} \log \frac{1}{m+1} + \frac{1}{m+1} \log \frac{1}{m+1} + \ldots + \ldots\ m \text{ times},$$

so the optimal policy for $n = m+1$ will be $p_1 = p_2 = p_3 = \ldots = p_{m+1} = 1/(m+1)$ for which

$$f_{m+1}(1) = (m+1)\,[\,1/(m+1) \log 1/(m+1)\,]$$

Hence the result is true for $n = m+1$ also.

Thus, the optimal policy $p_1 = p_2 = p_3 = \ldots = p_n = 1/n$ will be true for general n.

EXAMINATION PROBLEM

1. Use dynamic programming to show that : $x_1 \log x_1 + x_2 \log x_2 + \ldots + x_n \log x_n$ subject to the constraints $x_1 + x_2 + \ldots + x_n = k$ and $x_i \ge 0, i = 1, 2, \ldots, n$ is minimum when $x_1 = x_2 = \ldots = x_n = k/n$, where $k > 0$ is constant. **[Delhi (MA/M.Sc. II Maths.) 96]**
2. Use dynamic programming to show that : $z = p_1 \log p_1 + p_2 \log p_2 + \ldots + p_n \log p_n$, subject to the constraints : $p_1 + p_2 + \ldots + p_n = 1$; and $p_y \ge o\ (y = 1, 2, \ldots, n)$ is minimum when $p_1 = p_2 = \ldots = p_n = -1/n$. **[JNTU (MCA III) 2004]**

Example 10. *Use the principle of optimality to find the maximum value of*

$$z = b_1x_1 + b_2x_2 + b_3x_3 + \ldots + b_nx_n$$

when $x_1 + x_2 + x_3 + \ldots + x_n = c$, *and* $x_1, x_2, x_3, \ldots, x_n \ge 0, b_1 > 0, b_2 > 0, \ldots, b_n > 0$. **[Meerut (Maths) 97P, 90]**

Solution. The problem can be considered as to divide the positive quantity c into n parts $x_1, x_2, \ldots, x_n$ so that the expression $b_1x_1 + b_2x_2 + \ldots + b_nx_n$ is maximum. We assume that $b_1 < b_2 < b_3 < \ldots, < b_n$.

Let $f_n(c)$ denote the maximum attainable sum of $b_1x_1 + b_2x_2 + \ldots + b_nx_n$

Recursive Equations. If z_i be the ith part $(i = 1, 2, 3, \ldots, n)$ of the quantity, then the recursive equations of the problem are

$$f_1(x_1) = \max_{z_1 = x_1} \{b_1 z_1\} = b_1x_1 \text{ and } f_i(x_i) = \max_{0 \le z_i \le x_i} \{b_iz_i + f_{i-1}(x_i - z_i)\},\ i = 1, 2, \ldots, n.$$

Solution of recursive equations. For one stage problem $(i = 1)$, $f_1(x_1) = b_1x_1$.

This gives $f_1(c) = b_1c$ (which is trivially true).

For two stage problem $(i = 1, 2)$

$$f_2(x_2) = \max_{0 \le z_2 \le x_2} \{b_2z_2 + f_1(x_2 - z_2)\}$$

or $$f_2(c) = \max_{0 \le z \le c} \{b_2z + f_1(c-z)\} = \max_{0 \le z \le c} \{(b_2 - b_1)z + b_1c\} \text{ for } x_2 = c \text{ and } z_2 = z$$

If $b_2 - b_1$ is positive, then this is maximum for $z = c$, otherwise it will be minimum.

---

* $F(z) = z \log z + (1-z) \log \dfrac{1-z}{m}$

$$\frac{dF}{dz} = z\frac{1}{z} + \log z + (1-z)\frac{m}{1-z}\left(-\frac{1}{m}\right) + \log\frac{1-z}{m}(-1)$$

$= 0$ for maximum or minimum

which gives, $z = \dfrac{1}{m+1}$ for which $\dfrac{d^2F}{dz^2}$ is negative.

Thus, $f_2(c) = b_2 c.$

Similarly, for three stage problem ($i = 1, 2, 3$)

$$f_3(x_3) = \max_{0 \le z_3 \le x_3} \{b_3 z_3 + f_2(x_3 - z_3)\}$$

or

$$f_3(c) = \max_{0 \le z \le c} \{b_3 z + f_2(c - z)\} = \max_{0 \le z \le c} \{b_3 z + b_2(c - z)\} = \max_{0 \le z \le c} \{(b_3 - b_2)\, z + b_2 c\}$$

Again, if $b_3 - b_2$ is positive, then it gives maximum value for $z = c$, otherwise gives the minimum value.

Thus, $f_3(c) = b_3 c.$

From the results of three stages 1, 2, 3 it can be easily shown by induction method that $f_n(c) = b_n c$.

Hence the optimal policy will be $(0, 0, 0 \ldots, x_n = c)$ with $f_n(c) = b_n c$.

Now we shall give such example in which only integral values of decision variables are considered.

Example 11. *A student has to take examination in three courses X, Y, and Z. He has three days available for study. He feels it would be best to devote a whole day to the study of the same course, so that he may study a course for one day, two days or three days or not at all. His estimates of grades he may get by study are as follows.*

Study days \ Course	X	Y	Z
0	1	2	1
1	2	2	2
2	2	4	4
3	4	5	4

How should he plan to study so that he maximizes the sum of his grades.

Solution. Let n_1, n_2 and n_3 be the number of days he should study the courses X, Y and Z, respectively. If $f_1(n_1)$, $f_2(n_2)$, $f_3(n_3)$ be the grades earned by such a study, then the problem becomes :

Maximize $z = f_1(n_1) + f_2(n_2) + f_3(n_3)$ subject to $n_1 + n_2 + n_3 \le 3$ and *integers.*

Here, n_j are the decision variables and $f_j(n_j)$ are the corresponding return functions for $j = 1, 2, 3$.

Now, introducing state variables s_j as follows :

$$s_3 = n_1 + n_2 + n_3 \le 3$$
$$s_2 = n_1 + n_2 \quad = s_3 - n_3$$
$$s_1 = n_1 \quad = s_2 - n_2$$

Thus, state transformation functions are defined as

$$s_{j-1} = T_j(s_j, n_j), \; j = 2, 3.$$

Recursive equations applicable here are :

$$F_j(s_j) = \max_{n_j} [f_j(n_j) + F_{j-1}(s_{j-1})], \text{ and } F_1(s_1) = f_1(n_1), j = 2, 3$$

where $F_3(s_3) = \max_{n_1, n_2, n_3} [f_1(n_1) + f_2(n_2) + f_3(n_3)]$ for any feasible value of s_3. Then the required solution would become $\max_{s_3} F_3(s_3)$.

Recursive operations leading to the answer are tabulated as follows :

Stage returns $f_j(n_j)$

j \ n_j	0	1*	2	3
1	1	[2]	2	4
2	2	2	4	5
3	1	2	4	4

Stage transformation s_{j-1}, $j = 2, 3$

s_j \ n_j	0	1	2	3
0	0	—	—	—
1	1	0	—	—
2	2	1	0	—
3	3	2	1	0

Recursive Operations

	$f_n(n_2)$				$F_1(s_1) = f_1(n_1)$				$f_2(n_2) + F_1(s_1)$				$F_2(s_2)$
s_2 \ n_2	0	1	2	3	0	1	2	3	0*	1	2	3	
0	[2]	—	—	—	[1]	—	—	—	[3]	—	—	—	[3]
1	2	2	—	—	2	1	—	—	4	3	—	—	4
2	2	2	4	—	2	2	1	—	4	4	5	—	5
3	2	2	4	5	4	2	2	1	6	4	6	6	6

n_3 \ s_3	$f_3(n_3)$				$F_2(s_2) = f_2(n_2)$				$f_3(n_3) + F_2(s_2)$				$F_3(s_3)$
	0	1	2	3	0	1	2	3	0	1	2*	3	
0	1	—	—	—	3	—	—	—	4	—	—	—	4
1	1	2	—	—	4	3	—	—	5	5	—	—	5
2	1	2	4	—	5	4	3	—	6	6	7	—	7
3	1	2	[4]	4	6	5	[4]	3	7	7	[8]	7	[8]

Proceeding backwards through enclosed type numbers, the optimal policy is obtained as $n_3 = 2, n_2 = 0, n_1 = 1$, keeping in view $n_1 + n_2 + n_3 \leq 3$. The required maximum return is 8.

Example 12. *State the principle of optimality and apply it to solve the following problems ;*

(a) *A member of a certain political party is making plans for his election to the parliament. He has received the service of six volunteer workers and wishes to assign them to three districts in such a way as to maximize their effectiveness. He feels that it would be inefficient to assign a worker to more than one district but he is willing to assign no worker to any one of the district if they can accomplish in other districts.*

The following table gives the estimated increase in the number of votes in his favour in each district if it were allocated various number of workers :

Number of workers	*Districts*		
	1	2	3
0	0	0	0
1	25	20	33
2	42	38	43
3	55	54	47
4	63	65	50
5	69	73	52
6	74	80	53

How many of the six workers should be assigned to each of the three districts in order to maximize total estimated increase in the number of votes in his favour. **[I.I.I.E. (Grad.) 91]**

(b) *Solve the above problem by adding one more column for 4th district as :*
0 13 24 32 39 45 50.

Solution. Let the *three* districts be taken as three stages in a dynamic programming formulation.

Step 1. ***Formulation of the problem* :**

Let $x_j \rightarrow$ number of workers at the jth stage from the previous one, where $j = 1, 2, 3$.

$V_j(x_j) \rightarrow$ expected number of votes when x_j workers are assigned to jth district.

Then the problem can be formulated as a linear programming problem :

$$\text{Maximize } z = V_1(x_1) + V_2(x_2) + V_3(x_3)\text{, subject to the constraints}$$

$$x_1 + x_2 + x_3 = 6 \text{ and } x_1, x_2, x_3 \geq 0.$$

Step 2. ***To obtain the recurrence relation* :**

Let there be s workers available for remaining j districts and x_j be the initial assignment. Define $f_j(x_j)$ as the value of the optimal assignment for district 1 through 3 both inclusive. Thus for stage $j = 1$,

$$f_1(s, x_1) = \{V_1(x_1)\}.$$

If $f_j(s, x_j)$ be the profit associated with the optimum solution $f_j^*(s)$, $j = 1, 2, 3$, then

$$f_1^*(s) = \max_{0 \leq x_1 \leq s} [V_1(x_1)]$$

The recurrence relation thus obtained is

$$f_j(s, x_j) = V_j(x_j) + f_{j+1}(s - x_j), \text{ for } j = 1, 2, 3$$

and

$$f_j^*(s) = \max_{0 \leq x_j \leq s} [V_j(x_j) + f_{j+1}^*(s - x_j)]$$

Step 3. *Solution of the problem* :

Now the solution to this problem starts with $f_3^*(s)$ and is completed when $f_1^*(s)$ is obtained. Since all the values in the recurrence relation are discrete, tabular method will be used. The optimal policy is denoted by x_j^*, $j = 1, 2, 3$.

From this computation table it is evident that the maximum increase in the number of votes is 129. The optimum solution is $x_1^* = 2$ which makes $s = 6 - 2 = 4$ for two stage problem. Hence $x_2^* = 3$ which makes $s = 4 - 3 = 1$. It gives $x_3^* = 1$.

Computations for one-stage problem

s	$f_3^*(s)$	x_3^*
0	0	0
1	33	1
2	43	2
3	47	3
4	50	4
5	52	5
6	53	6

Computations for Two-Stage Problem

s \ x_2	$f_2(s, x_2) = V_2(x_2) + f_3(s - x_2)$: 0	1	2	3	4	5	6	Optimum Sol. $f_2^*(s)$	x_2^*
0	0 + 0 = **0**							0	0
1	0 + 33 = 33	20 + 0 = 20						33	0
2	0 + 43 = 43	20 + 33 = **53**	38 + 0 = 38					53	1
3	0 + 47 = 47	20 + 43 = 63	38 + 33 = **71**	54 + 0 = 54				71	2
4	0 + 50 = 50	20 + 47 = 67	38 + 43 = 81	54 + 33 = **87**	65 + 0 = 65			87	3
5	0 + 52 = 52	20 + 50 = 70	38 + 47 = 85	54 + 43 = 97	65 + 33 = **98**	73 + 0 = 73		98	4
6	0 + 53 = 53	20 + 52 = 72	38 + 50 = 88	54 + 47 = 101	65 + 43 = **108**	73 + 33 = 106	80 + 0 = 80	108	4

Computations for Three-Stage Problem

s \ x_1	$f_1(s, x_1) = V_1(x_1) + f_2(s - x_1)$: 0	1	2	3	4	5	6	Optimum Sol. $f_1^*(s)$	x_1^*
6	0 + 108 = 108	25 + 98 = 123	42 + 87 = 129	55 + 71 = 126	63 + 53 = 116	69 + 33 = 102	74 + 0 = 74	129	2

Finally, the optimum solution is obtained and the maximum increase in the number of votes = 129.

Optimum Distribution of 6 Workers to 3 Districts

x_1^*	x_2^*	x_3^*
2	3	1

(b) Repeat above procedure with 4th district.

Example 13. *Seven units of capital can be invested in four activities with the return from each activity given in the accompanying table. Find the allocation of capital to each activity that will maximize the total return.*

Q	$g^1(Q)$	$g^2(Q)$	$g^3(Q)$	$g^4(Q)$
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	8	8

Solution. ***Step 1. Formulation of the problem* :**

Let us consider four activities as four stages. The decision variable x_j ($j = 1, 2, 3, 4$) denotes the number of units which can be invested at the jth stage.

Now let $R_j(x_j)$ be the expected return from the allocation of x_j units to activity j. Then, the problem can be formulated as a linear programming problem :

$$Max\ z = R_1(x_1) + R_2(x_2) + R_3(x_3) + R_4(x_4),$$ subject to the constraints :

$$x_1 + x_2 + x_3 + x_4 = 7, \text{ and } x_j \geq 0, j = 1, 2, 3, 4.$$

***Step 2. To obtain the recurrence relationship* :**

Let there be s units available for remaining j activities and x_j be the initial allocation. If $f_j(x_j)$ defines the value of the optimum allocation for four activities, then the recurrence relation becomes

$$f_1(s, x_1) = \{R_1(x_1)\}, \text{ which implies } f_1^*(s) = \max_{0 \le x \le s} \{R_1(x_1)\}$$

and

$$f_j^*(s) = \max_{0 \le x_j \le s} \{R_j(x_j) + f_{j+1}^*(s - x_j)\}, j = 1, 2, 3, 4.$$

***Step* 3. *Solution of the problem* :**

The solution to this problem starts with $f_4^*(s)$ and is completed when $f_1^*(s)$ is obtained.

Computations for one-stage problem

s	$f_4^*(s)$	x_4^*
0	0	0
1	1	1
2	3	2
3	5	3
4	6	4
5	7	5
6	8	6
7	8	6 or 7

Computations for two-stage problem

s \ x_3	0	1	2	3	4	5	6	7	$f_3^*(s)$	x_3^*
0	0+0								0	0
1	0+1	2+0							2	1
2	0+3	2+1	3+0						3	0, 1, 2
3	0+5	2+3	3+1	4+0					5	0, 1
4	0+6	2+5	3+3	4+1	5+0				7	1
5	0+7	2+6	3+5	4+3	5+1	5+0			8	1, 2
6	0+8	2+7	3+6	4+5	5+3	5+1	5+0		9	1, 2, 3
7	0+8	2+8	3+7	4+6	5+5	5+3	5+1	5+0	10	1, 2, 3, 4

Computations for three-stage problem

s \ x_2	0	1	2	3	4	5	6	7	$f_2^*(s)$	x_2^*
0	**0+0**								0	0
1	0+2	**3+0**							3	1
2	0+3	**3+2**	5+0						5	1, 2
3	0+5	3+3	5+2	**7+0**					7	2, 3
4	0+7	3+5	5+3	**7+2**	**9+0**				9	3, 4
5	0+8	3+7	5+5	7+3	9+2	10+0			11	4
6	0+9	3+8	5+7	7+5	9+3	**10+2**	11+0		12	2, 3, 4, 5
7	0+10	3+9	5+8	7+7	9+5	10+3	11+2	12+0	14	3, 4

Computations for four-stage problem

s \ x_1	0	1	2	3	4	5	6	7	$f_1^*(s)$	x_1^*
7	0+14=14	2+12=14	4+11=15	6+9=15	7+7=14	8+5=13	9+3=12	9+0=9	**15**	2, 3

From this table maximum profit is obtained as 15. The optimum solution is $x_1^* = 2$ or 3, which gives $s = 7 - 2 = 5$ or $s = 7 - 3 = 4$ for three stage problem. Hence $x_2^* = 4$ when $x_1^* = 2$, and $x_2^* = 3$ or 4 when $x_1^* = 3$; $x_2^* = 4$ gives $s = 5 - 4 = 1$, which gives $x_3^* = 1$; $x_2^* = 3$ gives $x_3^* = 0$. Further, $x_3^* = 1$ makes $s = 1 - 1 = 0$ which gives $x_4^* = 0$, and $x_3^* = 0$ makes $s = 0 - 0$ which gives $x_4^* = 0$.

Finally, following *three* alternative optimum solutions are obtained such that sum of each row must be 7 :

$g^1(Q)$	$g^2(Q)$	$g^3(Q)$	$g^4(Q)$
2	4	1	0
3	3	1	0
3	4	0	0

and the maximum return is 15.

Example 14. (Production Allocation Problem). *The owner of a chain of four grocery stores has purchased six crates of fresh strawberries. The estimated probability distribution of potential sales of the strawberries before spoilage differ among the four stores. The following table gives the estimated total expected profit at each store, when it is allocated various number of crates.*

For administrative reasons, the owner does not wish to split crates between stores. However, he is willing to distribute zero crates to any of his stores.

Find the allocation of six crates to four stores as to maximize the expected profit.

	Store 1	2	3	4
Number of Crates 0	0	0	0	0
1	4	2	6	2
2	6	4	8	3
3	7	6	8	4
4	7	8	8	4
5	7	9	8	4
6	7	10	8	4

Solution. Let the *four stores* be considered as *four stages.*

Step 1. ***Formulation of the problem*** **:**

Let $x_j \rightarrow$ number of crates allocated at the jth stage, $j = 1, 2, 3, 4$.

$P_j(x_j) \rightarrow$ expected profit from allocation of x_j crates to store j.

Now the problem can be formulated as a linear programming problem as follows :

$$Max\ z = P_1(x_1) + P_2(x_2) + P_3(x_3) + P_4(x_4) \text{ subject to the constraints :}$$

$$x_1 + x_2 + x_3 + x_4 = 6 \text{ and } x_1, x_2, x_3, x_4 \geq 0.$$

Step 2. ***To obtain the recurrence relations :***

Let there be s crates available for remaining j stores and x_j be the initial allocation. Define $f_j(x_j)$ as the value of the optimal allocation for stores 1 through 4 both inclusive. Therefore, for stage $j = 1$,

$$f_1(s, x_1) = \{P_1(x_1)\}.$$

If $f_j(s, x_j)$ denotes the profit associated with the optimum solution $f_j^*(s)$ $(j = 1, 2, 3, 4)$, then

$$f_1^*(s) = \max_{0 \leq x_1 \leq s} \{P_1(x_1)\}$$

Therefore, the recurrence relation is obtained as

$$f_j(s, x_j) = P_j(x_j) + f_{j+1}^*(s - x_j), j = 1, 2, 3, 4$$

and

$$f_j^*(s) = \max_{0 \leq x_j \leq s} \{P_j(x_j) + f_{j+1}^*(s - x_j)\}.$$

Step 3. ***Solution of the problem :***

The solution to this problem can be started with $f_4^*(s)$ and is completed when $f_1^*(s)$ is determined.

Computations for first stage problem

s	$f_4^*(s)$	x_4^*
0	0	0
1	2	1
2	3	2
3	4	3
4	4	3, 4
5	4	3, 4, 5
6	4	3, 4, 5, 6

Computations for second stage problem

	$f_3(s, x_3) = P_3(x_3) + f_4^*(s - x_3)$							Optimum Sol.	
s \ x_3	0	1	2	3	4	5	6	$f_3^*(s)$	x_3^*
0	0 + 0							0	0
1	0 + 2	6 + 0						6	1
2	0 + 3	6 + 2	8 + 0					8	1, 2
3	0 + 4	6 + 3	8 + 2	8 + 0				10	2
4	0 + 4	6 + 4	8 + 3	8 + 2	8 + 0			11	2
5	0 + 4	6 + 4	8 + 4	8 + 3	8 + 2	8 + 0		12	2
6	0 + 4	6 + 4	8 + 4	8 + 4	8 + 3	8 + 2	8 + 0	12	2, 3

Computations for third-stage problem

	$f_2(s, x_2) = P_2(x_2) + f_3^*(s - x_2)$							Optimum Sol.	
s \ x_2	0	1	2	3	4	5	6	$f_2^*(s)$	x_2^*
0	0 + 0							0	0
1	**0 + 6**	2 + 0						6	1
2	**0 + 8**	**2 + 6**	4 + 0					8	0, 1
3	**0 + 10**	**2 + 8**	**4 + 6**	6 + 0				10	0, 1, 2
4	0 + 11	**2 + 10**	**4 + 8**	**6 + 6**	8 + 0			12	1, 2, 3
5	0 + 12	2 + 11	**4 + 10**	**6 + 8**	**8 + 6**	9 + 0		14	2, 3, 4
6	0 + 12	2 + 12	4 + 11	**6 + 10**	**8 + 8**	9 + 6	10 + 0	16	3, 4

Computations for fourth-stage problem

	$f_1(s, x_1) = P_1(x_1) + f_2^*(s - x_1)$							Optimum Sol.	
s \ x_1	0	1	2	3	4	5	6	$f_1^*(s)$	x_1^*
6	0 + 16 = 16	4 + 14 = **18**	6 + 12 = **18**	7 + 10 = 17	7 + 18 = 15	7 + 6 = 13	7 + 0 = 7	18	1, 2

From above computations, it is observed that the maximum profit of Rs. 18 can be obtained by choosing the following *eight alternative solutions* such that sum of each row must be 6 :

Distribution on 6 Crates to 4 Stores

Store 1 x_1^*	Store 2 x_2^*	Store 3 x_3^*	Store 4 x_4^*
1	2	2	1
1	3	1	1
1	3	2	0
1	4	1	0
2	1	2	1
2	2	1	1
2	2	2	0
2	3	1	0

This solution may also be obtained by careful inspection of the given data but, in general, it is not so obvious.

Example 15. *The profit associated with each of the four activities as a function of the man-hours allocated to each activity is given in the following table. If man-hours are available each day, how should allocation of time be made so that the profit per day is maximized ?*

H :	0	1	2	3	4	5	6	7	8
$g^1(H)$:	0	1	3	6	9	12	14	15	16
$g^2(H)$:	0	2	5	8	11	13	15	16	17
$g^3(H)$:	0	3	7	10	12	13	13	13	13
$g^4(H)$:	0	5	5	8	10	10	12	13	14

Use dynamic programming technique to solve the above problem ?

Solution. Let the four activities be considered as *stages* of dynamic programming problem.

Step 1. Formulation of the problem. The decision variable x_j ($j = 1, 2, 3, 4$) will denote the number of man-hours available at the jth stage. If $P_j(x_j)$ denotes the profit from the allocation of x_j hours to jth activity, then the problem becomes : *Max* $z = P_1(x_1) + P_2(x_2) + P_3(x_3) + P_4(x_4)$, subject to the constraints

$$x_1 + x_2 + x_3 + x_4 = 8 \text{ and } x_j \geq 0; j = 1, 2, 3, 4.$$

This problem is similar to previous one. So proceeding in the same way, it can be verified that maximum profit is 23 which can be achieved by choosing any of the following alternative optimum solutions :

$g^1(H)$:	0	0	0	0	0	0	0
$g^2(H)$:	2	3	3	4	4	4	5
$g^3(H)$:	3	2	3	3	2	4	3
$g^4(H)$:	3	3	2	1	2	0	0

EXAMINATION PROBLEMS

1. Obtain the functional equation for maximizing $z = g_1(x_1) + g_2(x_2) + \dots + g_n(x_n)$. subject to $x_1 + x_2 + \dots + x_n = c$ and $x_j \ge 0, j = 1, 2, 3, \dots, n$.

   [**Ans.** $f_1(c) = \max_{0 \le z \le c} \{g_1(z)\} = g_1(c)$, $f_n(c) = \max_{0 \le z \le c} \{g_1(z) + f_{n-1}(c - z)\}$.]

2. (a) Obtain the functional equations of dynamic programming for solving the problem :

$$\min \sum_{j=1}^{n} r_j^{\alpha}, \ \alpha > 0, \text{ subject to } \sum_{j=1}^{n} r_j^{\alpha} \ge a, a \ge 0 ; r_j \ge 0 ; (j = 1, 2, \dots, n)$$

   [**Hint.** Divide a into n parts $r_1, r_2, \dots, r_n$ so that $r_1^{\alpha} + r_2^{\alpha} + r_3^{\alpha} + \dots + r_n^{\alpha}$ is minimum.]

   (b) If $\alpha = 2$ and $n = 3$, what will be the functional equation ?

3. Let us define the function $f_N(a) = \min_{R} \sum_{i=1}^{N} x_i^P$, $p > 0$, where R is defined by

$$\text{(i) } \sum_{i=1}^{N} x_i \le a, \ a > 0, \quad \text{(ii) } x_i \ge 0 \text{ for all } i.$$

   (a) Show that $f_N(a)$ satisfies the recursive relation

$$f_N(a) = \min_{0 \le x \le a} [x^P + f_{N-1}(a - x)], \ N \ge 2, \text{ and } f_1(a) = a^P,$$

   (b) Prove that if $0 < p < 1$; $f_N(a) = a^P$,

   (c) Prove that if $p > 1$; $f_N(a) = N(a/N)^{P-1}$. **[Meerut (Math.) Jan. 98 BP]**

33.8. MODEL IV : SINGLE MULTIPLICATIVE CONSTRAINT, ADDITIVELY SEPARABLE RETURN

Consider the problem : Minimize $z = f_1(y_1) + f_2(y_2) + \dots + f_n(y_n)$ subject to the constraints

$$y_1 y_2 \dots y_n \ge p, \ p \ge 0, \ y_j \ge 0 \text{ for all } j.$$

State variables are defined as

$$s_n = y_n y_{n-1} \dots y_2 y_1 \ge p$$
$$s_{n-1} = s_n / y_n = y_{n-1} \, y_{n-2} \dots y_2 y_1$$
$$\dots \quad \dots \quad \dots \quad \dots \quad \dots$$
$$s_2 = s_3 / y_3 = y_2 y_1$$
$$s_1 = s_2 / y_2 = y_1$$

These state variables are stage transformations of the type $s_{j-1} = T_j(s_j, y_j)$.

Let $F(s_n)$ be the minimum value of the objective function for any feasible s_n. Thus, proceeding as earlier obtain the recursion formula

$$f_j(s_j) = \min_{y_j} [f_j(y_j) + F_{j-1}(s_{j-1})], \ 2 \le j \le n$$

which will lead to the required situation.

Example 16. *Use Bellman's principle of optimality to minimize* $z = y_1 + y_2 + \dots + y_n$, *subject to the constraints :*

$$y_1 y_2 \dots y_n = d, \ y_j \ge 0 \text{ for } j = 1, 2, \dots, n.$$

[Agra 96; Delhi (OR) 93; Meerut (Maths.) 93P, I.A.S. (Maths.) 92; Raj. Univ. (M. Phil) 90]

Solution. Let $f_n(d)$ denote the minimum attainable sum $y_1 + y_2 + y_3 + \dots + y_n$ when the quantity d is factorized into n factors.

For $n = 1$, d is factorized into one factor only, so $f_1(d) = \min_{y_1 = d} \{y_1\} = d$.

For $n = 2$, d is factorized into two factors y_1, y_2.

If $y_1 = z$ and $y_2 = d/z$, then

$$f_2(d) = \min \{y_1 + y_2\} = \min_{0 \le z \le d} \{ z + d/z \} = \min_{0 \le z \le d} \{ z + f_1(d/z) \} \qquad [\text{since } f_1(d/z) = d/z]$$