

**OPTIMIZATION ALGORITHM FOR NONLINEAR  
SEPARABLE AND RATIONAL OBJECTIVES**

**BY**

**HARUNA MOSHOOD**

**M.TECH/SSSE/99/2000/455**

**A PROJECT SUBMITTED TO THE  
MATHEMATICS AND COMPUTER SCIENCE  
DEPARTMENT, FEDERAL UNIVERSITY OF  
TECHNOLOGY, MINNA, NIGERIA**

**IN PARTIAL FULFILMENT OF THE REQUIREMENT  
FOR THE AWARD OF MASTER OF TECHNOLOGY  
(M.TECH.) DEGREE IN APPLIED MATHEMATICS.**

**JULY; 2003**

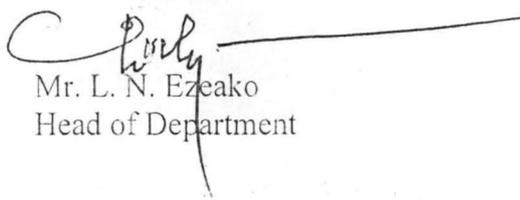
## CERTIFICATION

This thesis titled "OPTIMIZATION ALGORITHM FOR NON-LINEAR SEPARABLE AND RATIONAL OBJECTIVES" by Haruna Moshood, meets the regulations governing the award of the degree of Masters of Technology in Mathematics. Federal University of Minna and is approved for its contribution to knowledge and literary presentation.



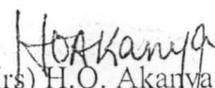
Dr. S. A. Reju  
Supervisor

Date: 16-12-2003  
31-8-03



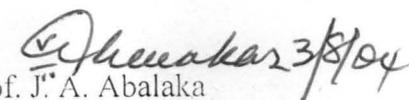
Mr. L. N. Ezeako  
Head of Department

Date: 26-1-2004



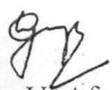
Prof. (Mrs) H.O. Akanya  
Dean, School of Science  
and Science Education

Date: 4th -02-2004



Prof. J. A. Abalaka  
Dean, Postgraduate School

Date: .....



Prof. A. U. Afuwape  
External Examiner

Date: 31/7/2003

## **ACKNOWLEDGEMENTS**

First of all, I give thanks to the Almighty God for given me the strength, idea and courage to pick up this project topic, and for me to reach this stage in my life.

I will like to acknowledge and thank my entire Family, most especially my beloved late Father and Mum for their support in everything; I will not also forget the support giving to me by my wife Engr, (Mrs.) Amamat O. Haruna and my little kid Master AbdulNafi'u Y. Haruna.

My sincere gratitude to Dr. S. Reju who is my Project Supervisor, for his guidance, constructive criticisms, going through the manuscript, making necessary corrections and offering useful suggestions throughout the research period.

To my remaining lecturers, Prof. Adegboye and Dr. Yomi Ayesinmi, I say a very **BIG THANK YOU.**

I also appreciate the encouragement giving to me by my H.O.D Mallam Moshood Badaru, Mallam AbdulSalami, Shehu Bolaji and Gideon T. Okedayo.

I will like to acknowledge the advice, assistance and supports I always received from my two special friends Mr. Tajudeen Yahaya and Mr. Dauda O. Issa.

I will no forget the kindness of the following people, Mallam Isiaka Akanbi, Mallam Sule Ahmed and Mr. Ladi Awodi.

Let me also acknowledge the Unquantifiable assistance rendered to me by Ahmad Sadiq and Hassan Akande.

I will also like to acknowledge the effort of Ismail Akande for typing my manuscript.

**HARUNA MOSHOOD**

JUNE, 2002

## TABLE OF CONTENT

TITLE PAGE .....	I
CERTIFICATION .....	II
DEDICATION .....	III
ACKNOWLEDGEMENT .....	IV
TABLE OF CONTENT .....	V-VIII
ABSTRACT .....	IX

### CHAPTER I

1.0 INTRODUCTION TO OPTIMIZATION THEORY .....	1
1.1 Mathematical Model .....	2
1.1.1 Characteristics and Types of Models .....	2
1.1.2 Model Classification .....	3
1.1.3 Specific Models of Interest .....	4
1.2 Standard Formulation .....	5
1.2.1 Formulating an Optimization Problem .....	6
1.3 Nonlinear Programming .....	8
1.3.1 Types of Nonlinear Objective Functions .....	9
1.3.1.(A) A nonlinear function in one variable .....	9
1.3.1 (B) A nonlinear function of several unconstrained .....	11
1.3.1 (C) A nonlinear function of several constrained variables.....	14
Definition 1.3.1 Feasible Region .....	14
Definition 1.3.2 Objective Function.....	15

1.4 Local and Global Optima .....	15
Definition 1.4.1 Global Maximum (Unconstrained Problem) .....	16
Definition 1.4.2 Local maximum (Unconstrained problem).....	16
Theorem 1.4.1 .....	16-17
Definition 1.4.3 Gradient Vector .....	17-18
Definition 1.4.4 Hessian matrix .....	18-19
Theorem 1.4.2 .....	19-20
Theorem 1.4.3 .....	20-21
Corollary 1.4.4 .....	21
Theorem 1.4.5 .....	21-22
Corollary 1.4.6 .....	22
Corollary 1.4.7 .....	22
Example 1.3.....	22-23
Definition 1.4.5 Global maximum (constrained problem).....	23-24
Definition 1.4.6 Local maximum (constrained problem) .....	24
1.5 Langragian Multiplier's And Equality Constrained Problem .....	24-27
1.6 Kuhn Tucker Conditions .....	27-29
1.7 The Langragian Solution Method .....	29-30
Example 1.4 & 1.5 .....	30-35
Summary .....	35
CHAPTER II	
2.0 OPTIMIZATION PROBLEM WITH SEPARABLE OBJECTIVE.....	36
2.1 Introduction .....	36

2.1.1 Definition .....	36-37
2.2 Transformation of Nonlinear Programming Problem (NLPP) To Separable Programming (SPP) .....	37-38
Illustration 2.1 .....	38
2.2.1 General Revision Linear Approximation Technique for Univariate Function	38-39
2.2.2 The General Piecewise Linear Approximation Technique for N-Variable Functions .....	39-42
2.3 Optimization Algorithm For Separable Objective .....	42
Example 2.2 .....	42-48
Example 2.3 .....	48-53
2.4 Langragian Multiplier Method (Exact Solution) .....	53-54
More Examples .....	54-75
2.5 Conversion of Non-Separability Programming Problem (NSPP) To Separable Programming Problems .....	75
2.5.1 Non-Separability Due To Cross Product Term .....	75-76
Examples .....	76-78
2.5.2 Nonseparability Due To Exponential Function.....	78-79
Examples .....	79-80
2.6 Computation of Nonlinear Programming Model .....	80-81
Remarks .....	81
Summary .....	81
CHAPTER III	
3.0. OPTIMIZATION MODEL WITH RATIONAL OBJECTIVE.....	82

3.1 Introduction .....	82
3.1.1 Definition (Fractional Programming) .....	82
3.2 Conversion of Fraction Programming Problem Into a Linear Model.....	82
3.2.1 Constrained/Unconstrained Problem .....	82
Example 3.1 .....	83-84
3.3 Optimization Algorithm For Rational Objective .....	84-85
3.4 Alternative Optimization Algorithm For Rational Objective .....	85-86
Examples 3.2&3.3.....	86-94
3.5 Langragian Method (Exact Optimal Value) .....	94-95
Examples.....	95-111
CHAPTER IV	
4.0 COMPUTATIONAL ALGORITHM FOR SEPARABLE OBJECTIVE .....	112
4.1 Pseudocode .....	112
4.2 Flowchart.....	113-114
4.3 Output of the Separable Objective Code.....	115-123
4.4 Graphical Solutions.....	124-127
CHAPTER V	
5.0 COMPUTATIONAL ALGORITHM FOR RATIONAL OBJECTIVE AND CONCLUSION .....	128
5.1 Output of the Rational Objective Using Mathcad.....	128-129
5.2 Graphical Solutions.....	130
5.3 Analysis of Results, Recommendation and Conclusion.....	131-132
Appendix (Separable Code).....	131-146
References .....	147

## ABSTRACT

In calculus we learned how to obtain the minimum and the maximum of a function by setting its derivative equal to zero. Unfortunately, minimization or maximization (Optimization) problems encountered in industry are not that simple. Usually, optimization should take place while satisfying a number of constraints imposed on the system. In cases where the constraints and the function to be optimized are expressed analytically, the Lagrangian method of undetermined multipliers can be used to obtain the optimum solution.

Linear programming methods developed in the 1950's can be very effective in cases where both the constraints and the function to be optimized are linear. Variations of linear programming methods are also available for cases where the function to be optimized is quadratic (quadratic programming) and for cases where the non-linear constraints or objective or both can be expressed as piecewise linear functions (Separable programming) and also for the cases where the non-linear objective is ratio of two linear functions can be expressed as a linear model (fractional programming).

# CHAPTER ONE

## 1.0 INTRODUCTION TO OPTIMIZATION THEORY

Optimization is concerned with achieving the best outcome of a given operation while satisfying certain restrictions. Human beings, guided and influenced by their natural surroundings, almost instinctively perform all functions in a manner that economizes energy or minimize discomfort and pain. The motivation is to exploit the available limited resources in a manner that maximizes output or profit. The early inventions of the lever or pulley mechanisms are clear manifestation of man's desire to maximize mechanical efficiency.

Physicist, Chemists, Mathematicians, Engineers, Economists, Operations researchers, Managers and practicing computer Scientists are often interested in achieving optimal solutions to their problems. These problems may be to determine designs, programs, trajectories, allocation of resources, approximations of functions. Frequently, different designs or programs, all satisfying the conditions arising from the actual situation are compared, and once is chosen that also as the best in terms of an optimality criterion. Optimization techniques, if properly applied, will automatically examine different designs or plans and select an optimum.

### Example 1.1

Consider the following optimization problem (model).

A chemical company must send  $1000\text{m}^3$  of chlorine gas to its research laboratory in another state because the gas is extremely dangerous, a special hermetically sealed rectangular railroad car must be built for transporting it. The material from which the top and the bottom must be constructed costs \$2000 per square meter, while the siding costs half as much, however, only  $50\text{m}^2$  of siding can be obtained. Moreover, the maximum height of the car permitted by tunnels and other overhead clearances is 3m. Regardless of the car's dimensions each round

trip to the laboratory will cost \$8000. Assuming no time limit on the overall procedure, what dimensions minimize the total cost of constructing the car and delivering the gas.

Let  $d$ ,  $w$  and  $h$  be the car's length, width and height.

The objective is to minimize overall cost;

i.e. minimize  $8000(1000/dwh) + 2dw(2000) + 2dh + 2wh(1000)$ .

Where the three terms are contributed by transportation cost, top and bottom material, and siding, respectively. The constraints mentioned in the problem are

$$2dh + 2wh \leq 50$$

$$\text{and } h \leq 3$$

Also non-negativity restriction

$$D, h, w \geq 0$$

How can such problems be solved in general?

## 1.1 Mathematical Models

### 1.1.1 Characteristics and types of Models

#### Mathematical programming problem

The problem of optimizing a numerical function of one or more variables when they are constrained in some manner is called a mathematical programming problem. Specifically, the purpose of such a problem is to determine the value of  $n$  variables  $x_1, x_2, \dots, x_n$  that optimize the function.

$$Z = f(x_1, x_2, \dots, x_n) \dots\dots\dots 1.1.1$$

Subject to the constraints

$$g(x_1, x_2, \dots, x_n) \{ \leq = \geq \} b_i, \quad i = 1, 2, \dots, m \dots\dots\dots 1.1.2$$

It is usually assumed that the value of the  $n$  variables cannot be negative numerically. The non-negativity restriction on the variables may be stated as

$$x_j \geq 0, \quad j = 1, 2, \dots, n \dots\dots\dots 1.1.3$$

Also, it is usually desired to determine the optimal value (minimum or maximum) of the function  $z$  in 1.1.1, which is called the objective function.

The formulation of business and economic questions as mathematical programming problems has resulted in the successful resolution of many complex real-world optimization situations. Most of the applications of mathematical programming to business and economics involve the maximization of revenues or profits and the minimization of costs.

### 1.1.2 Model Classification

A real-world optimization problem may be classified in five ways.

- (1) The functional relationship in the problem may be known (deterministic) or uncertain (probabilistic).
- (2) the function  $f(x_1, x_2, \dots, x_n)$  and  $g_i(x_1, x_2, \dots, x_n)$ ,  $i = 1, 2, \dots, m$  in 1.1.1 and 1.1.2 may be linear in  $x_1, x_2, \dots, x_n$ ; at least one function in the set may be non-linear.
- (3) The functions may be continuously differentiable (Smooth) or non-differentiable (non-smooth).
- (4) The variable  $x_1, x_2, \dots, x_n$  in the mathematical programming problem may be continuous or may be restricted to integer values.
- (5) The optimization may take place at a fixed point in time (static) or during an interval of time (dynamic).

Most mathematical programming models are deterministic; given  $x_1, x_2, \dots, x_n$ , the values of  $f_1, g_2, g_2, \dots, g_m$  are uniquely determined, most of the current application of mathematical programming to business and economic problems assume that all model functions are linear. There is a very simple reason for this. The simple method, derived by Dantzig in 1947 (1963), is extremely efficient procedure for solving linear programming problems. When this method is programmed on a computer it is possible to solve linear problems involving hundreds of variables and thousands of constants. If one or more of the functions is non-linear, the problem is always more difficult to solve than linear ones. Thus, even though the real-

world problem may be complex and inherently highly non-linear, successful modeling of it may be possible by using many variables and constraints in a linear formulation.

Most algorithms devised to solve mathematical programming problems require that the functions in the model be continuously, differentiable; thus all functions typically must be smooth.

### 1.1:3 Specific models of interest.

The best known mathematical programming model is linear programming model. All functions in 1.1 and 1.2 are linear in the variables  $x_1, x_2, \dots, x_n$ , the model may be written as

$$Z = f(x_1, x_2, \dots, x_n) = \sum_{j=1}^n f_j(x_j) \quad \dots\dots\dots 1.1.4$$

subject to

$$g_i(x_1, x_2, \dots, x_n) = \sum_{j=1}^n g_{ij}x_j \quad \{\leq = \geq\} b_i \quad i = 1, 2, \dots, m$$

$$x_j \geq 0, j = 1, 2, \dots, n \quad \dots\dots\dots 1.1.6$$

Where the  $f$ 's,  $b_i$ 's and  $a_{ij}$ 's are known constants.

The linear programming model has been successfully used to solve a variety of business, economic and scientific problems.

If in the model,

$$\text{Optimize } Z = f(x_1, x_2, \dots, x_n)$$

$$\text{Subject to } g_i(x_1, x_2, \dots, x_n) \{\leq = \geq\} b_i \quad i = 1, 2, \dots, m$$

$$x_j \geq 0, j = 1, 2, \dots, n$$

At least one function in the set  $f, g_1, g_2, \dots, g_m$  is non-linear, it is called a non-linear programming model. We observed that a non-linear problem is generally much more difficult to solve than a linear one. Many algorithms have been developed to alleviate this, among

which are the followings, separable, programming, fractional programming algorithms, Langragian multiplier's technique e.t.c.

A special case of the general nonlinear programming model which has received a great deal of attention, is the quadratic programming model in both chapters 2 and 3. In this model, the objective function is quadratic in  $x_1, x_2, \dots, x_n$  and the constraints are linear. Specifically the model is

$$\text{Optimize } Z = \sum_{j=1}^n f_j x_j + \sum_{j=1}^n g_{ij} x_i x_j \dots\dots\dots 1.1.7$$

$$\text{Subject to } \sum_{j=1}^n h_{ij} x_j \leq b_i \quad i = 1, 2, \dots, m \dots\dots\dots 1.1.8$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n \dots\dots\dots 1.1.9$$

Where the  $f_j$ 's,  $g_{ij}$ 's and  $h_{ij}$ 's are known constraints.

### 1.3 Standard Formulation.

The notation adopted in the work for variables, objective function and constraints is summarized in the following formulation of the optimization problem.

$$\text{Optimize } f(x_1, x_2, \dots, x_n) \dots\dots\dots 1.2.1$$

$$\text{Subject to } g_i(x_1, x_2, \dots, x_n) \leq b_i \quad i = 1, 2, \dots, m \dots\dots\dots 1.2.2$$

$$g_i(x_1, x_2, \dots, x_n) = b_k \quad k = 1, 2, \dots, n \dots\dots\dots 1.2.3.$$

$$x_j \geq 0 \quad j = 1, 2, \dots, n \dots\dots\dots 1.2.4$$

where  $x$  denotes a vector of variable in the components  $x_i, i = 1, 2, \dots, n$ . an optimization problem is said to be linear when both the objective function and the constraints are linear functions of the variables  $x_i$  i.e. they can be expressed in the form

$$f(x) = f_1 x_1 + f_2 x_2 + \dots + f_n x_n = f^T x$$

Linear optimization problems are solved by a branch of mathematical programming called linear programming. The optimization problem is said to be non-linear if either the objective function or the constraints are non-linear fraction of the variables.

### 1.2.1 Formulating an optimization problem

An optimization problem is an exercise in mathematical modeling that require great care in setting up the model. Four steps are involved:

- (1) Decide the exact objective to be optimized. Many different objectives are possible.
- (2) Set up the objective function using as many variables as are required. Try for accuracy rather than compactness. Make sure that all the terms have the same dimensional units.
- (3) Set up all the restraints and relationships between the variables.
- (4) If possible, reduce the objective function in step (2) to independent variables. Step (2) may contain both dependent and independent variables.
- (5) The objective function is now ready for solution. If it contains independent variables only, the differential can be set equal to zero to optimize the expression, or alternatively, tabulation can be made. If the objective function contains dependent variables in addition to independent variables, a Lagrange expression can be tried. If this fails, the objective function and its constraints (restraints) must be optimized, using the skill and ingenuity of the separable, fractional Quadratic programmers e.t.c.

As an illustration of the above, consider the Example 1.2

#### Example 1.2

A cheese shop has 20Kg of a seasonal fruit mix and 60Kg of an expensive cheese with which it will make two cheese spreads, deluxe and rectangular, that are popular during Christmas week. Each pound of the deluxe spread consists of 0.2Kg of the fruit mix and 0.8Kg of the expensive cheese, while each pound of the regular spread consists of 0.2Kg of the fruit mix, 0.3Kg of the expensive cheese, and 0.5Kg of a filler cheese which is cheap and in plentiful

supply. From past pricing policies, the shop has found that the demand for each spread depends on its price as follows:

$$D_1 = 190 - 25P_1 \quad \text{and} \quad D_2 = 250 - 50P_2$$

Where  $D$  denotes demand (in Kilograms),  $P$  denotes price (in dollars per Kg), and the Subscripts 1 and 2 refer to the deluxe respectively. How many Kgs of prices should the cheese shop prepare, and what prices should it establish, if it wishes to maximize income and be left with no inventory of either spread at the end of Christmas week?

Solution

Mathematical equivalent of the example.

Let  $x_1$  kgs of deluxe spread and  $x_2$  kgs of regular spread be made. If all the products can be sold, the objective is to

$$\text{Maximize } Z = P_1x_1 + P_2x_2 \quad \dots\dots\dots 1.2.5$$

Now, all product will indeed be sold (and none will be left over in inventory) if production does not exceed demand, i.e., if  $x_1 \leq D_1$  and  $x_2 \leq D_2$ . This gives the constraints.

$$x_1 + 25P_1 \leq 190 \quad \text{and} \quad x_2 + 50P_2 \leq 250 \quad \dots\dots\dots 1.2.6$$

From the availability of fruit mix,

$$0.2x_1 + 0.2x_2 \leq 20 \quad \dots\dots\dots 1.2.7$$

and from the availability of expensive cheese,

$$0.8x_1 + 0.3x_2 \leq 60$$

There is no constraint on the filler cheese, since the shop has as much as it needs. Finally, neither production nor price can be negative; so four hidden constraints are  $x_1 \geq 0$ ,  $x_2 \geq 0$ ,  $P_1 \geq 0$  and  $P_2 \geq 0$ . Combining these conditions with 1.2.5 through 1.2.8 we obtain the mathematical programming problem as follows:

$$\text{Maximize } Z = P_1x_1 + P_2x_2$$

$$\text{Subject to: } 0.2x_1 + 0.2x_2 \leq 20$$

$$0.8x_1 + 0.3x_2 \leq 60$$

$$x_1 + 25P_1 \leq 190$$

$$x_2 + 50P_2 \leq 250 \quad \dots\dots\dots 1.2.9$$

With all variables non-negative.

System (1.2.9) is a quadratic programming problem in the variables  $x_1$ ,  $x_2$ ,  $P_1$  and  $P_2$ . It can be simplified if we note that for any fixed positive  $x_1$  and  $x_2$  the objective function increases as either  $P_1$  or  $P_2$  must be such that the constraints 1.2.6 become equations, where  $P_1$  and  $P_2$  may be eliminated from the objective function. We then have a quadratic function in  $x_1$  and  $x_2$ .

$$\text{Maximize } Z = (7.6 - 0.04x_1)x_1 + (5 - 0.02x_2)x_2$$

$$\text{Subject to : } 0.2x_1 + 0.2x_2 \leq 20$$

$$0.8x_1 + 0.3x_2 \leq 60$$

with  $x_1$  and  $x_2$  non-negative.

\dots\dots\dots 1.2.10

### 1.3 Non-linear Programming.

In this work, emphasis is placed on nonlinear programming than linear programming due to the fact that the work centered on Non-linear separable and fractional programming on chapters two and three respectively. Although the simplex method was later utilized in finding solutions to both the piecewise linear approximation model in chapter two and the equivalent linear model of the fractional programming in chapter three.

The introduction of nonlinear functions in the mathematical programming problem usually insures more difficulty in solving the problem than if all functions are linear. The primary difficulty introduced by the nonlinear functions in the potential existence of relative or local minimal or maximal of the objective function.

The existence of local optima arises due to the nonlinearity of the objective function  $f(x)$ , the nonlinearity of one or more constraint functions  $g_i(x)$ , or a combination effect of the nonlinearity in  $f(x)$  and in one or more of the constraints functions.

### 1.3.1 Types Of Nonlinear Objective Functions.

#### 1.3.2A A Nonlinear Function In One Variable.

That is, optimizing a nonlinear objective function of a single variable. Note that many of the techniques for solving several-variable nonlinear optimization problems actually employ single-variable optimization in one of the steps. To begin, it is convenient to postulate "maximization" as the sense of optimization throughout the following discussion. [ If the real problem is to minimize an objective function  $f(x)$ , then can reformulate the method so as to maximize  $-f(x)$ .]

It is assumed that the functions considered possessed continuous first and second derivatives and partial derivative everywhere.

Consider a function of a single variable, such as that shown figure 1.3.1. A necessary condition for a particular solution,  $x = x^*$  to be either a minimum or maximum is that

$$df(x)/dx = 0 \text{ at } x = x^* \quad \dots\dots\dots 1.3.1$$

Thus in figure 1.3.1, there are five solutions satisfying these conditions. To obtain more information about these five so called critical points, it is necessary to examine the second derivative. Thus, if

$$d^2f(x)/dx^2 > 0 \text{ at } x = x^* \quad \dots\dots\dots 1.3.2$$

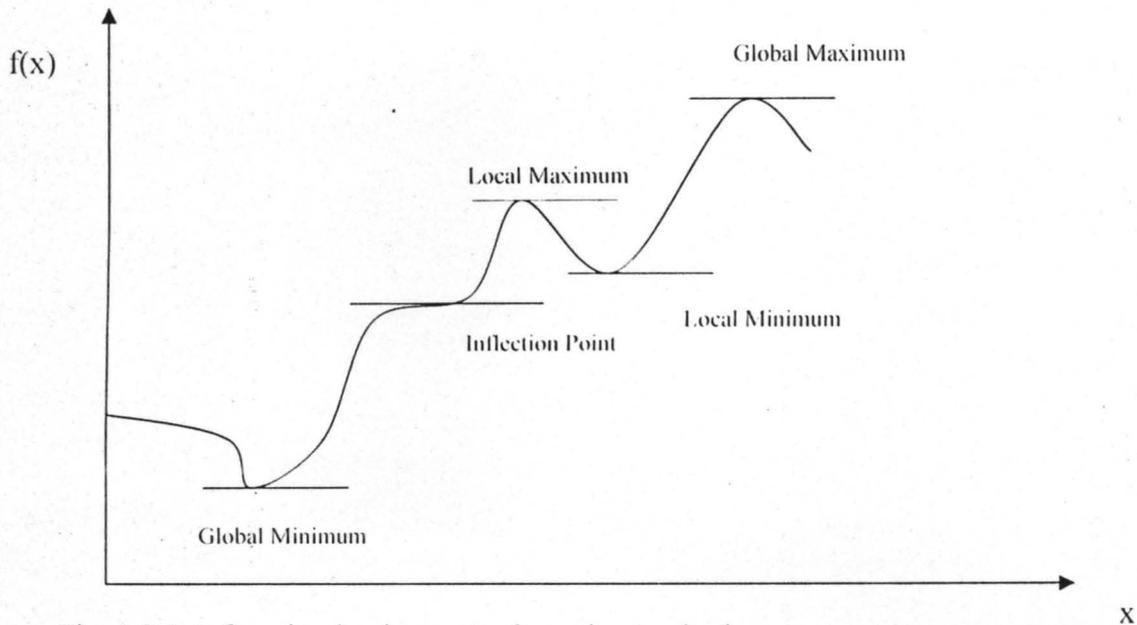


Fig. 1.3.1 A function having several maxima and minma.

Then  $x^*$  must be at least a local minimum (i.e.  $f(x^*) \leq f(x)$  for all  $x$  sufficiently close to  $x^*$ ). So  $x^*$  must be is a local minimum if  $f(x)$  is strictly convex with neighbourhood of  $x^*$ . Similarly, a sufficient condition for  $x^*$  to be a local maximum (given that it satisfies the necessary condition) is that  $f(x)$  is strictly concave with a neighbourhood of  $x^*$  (that is, the second derivative is negative at  $x$ ). If the second derivative is zero, the point may not even be an inflection point and it is necessary to examine higher derivatives.

To find a global minimum (i.e. a solution  $x^*$  such that  $f(x^*) \leq f(x)$  for all  $x$ ) it is necessary to compare the local minima and identify the one that yields the smallest value of  $f(x)$ . If this value is less than that  $f(x)$  as  $x \rightarrow -\infty$  and as  $x \rightarrow +\infty$  (or at the endpoints of the function, if it is only defined over a finite interval) then this point is a global minimum.

However, if  $f(x)$  is known to be either a convex or concave function, in particular, if  $f(x)$  is a convex function, then any solution  $x^*$ , such that

$$df(x)/dx = 0 \text{ at } x = x^*$$

is known automatically to be a global minimum. In other words this condition is not only a necessary but a sufficient condition for a global minimum of a convex function. If this

function is strictly convex, then this solution must be the only global minimum. Similarly, if  $f(x)$  is a concave function, then having

$$df(x)/dx = 0 \text{ at } x = x^*$$

becomes both necessary and sufficient condition for  $x^*$  to be a global maximum.

If for any  $x_1$  and  $x_2$  in  $I = [-\infty, \infty]$ , where  $x_1 < x_2$ , and for all  $p$ ,  $0 \leq p \leq 1$ ,  $f(x)$  satisfies

$$pf(x_1) + (1-p)f(x_2) \geq f(px_1 + (1-p)x_2) \quad \dots\dots\dots 1.3.3 \text{ (Convex function)}$$

A function is unimodal whenever it is concave, that is, if for any  $x_1$  and  $x_2$  in  $I$ , where  $x_1 < x_2$ , and for all  $p$ ,  $0 \leq p \leq 1$ ,  $f(x)$  satisfies

$$pf(x_1) + (1-p)f(x_2) \leq f(px_1 + (1-p)x_2) \quad \dots\dots\dots 1.3.4 \text{ (Concave function.)}$$

### 1.5.1B A Nonlinear Function Of Several Unconstrained Variables.

That is, maximizing a nonlinear function of several unconstrained variables. There are two motivating reasons for studying this problem. Firstly, an analysis of the multidimensional, unconstrained, nonlinear maximization problem sets the stage for the analyses of constrained models. The algorithmic difficulties to be overcome here are also present in the constrained cases. Secondly, a constrained problem can often be solved by first converting it to an unconstrained problem. We postulate that  $f(x_1, x_2, \dots, x_n)$  is smooth and possesses a finite maximum value, occurring at the finite values  $(x_1^*, x_2^*, \dots, x_n^*)$ . Abbreviating a set of value for  $(x_1, x_2, \dots, x_n)$ , by the symbol  $x$ , and the expression  $f(x_1, x_2, \dots, x_n)$  by the symbol  $f(x)$ , these assumptions can be stated more precisely as :

- (i) For all values of  $x$ ,  $f(x)$  is uniquely defined and finite.
- (ii) For all values of  $x$ , every partial derivative  $\partial f / \partial x_j$  is uniquely defined, finite and continuous, and hence  $f(x)$  is continuous
- (iii)  $f(x)$  possesses a finite maximum  $f^*$
- (iv) For any possible value of  $f(x)$ , say  $f$ , there exists an associated finite number  $M_f$  such that every  $|x_j| \leq M_f$  if  $f(x) \geq f$ .

Applying differential calculus, we can state the following.

Necessary condition for maximum. Given assumptions (i) through (iii), the function  $f(x)$  has a maximum at  $x^*$  only if  $\partial f(x) / \partial x_j = 0$  for  $j = 1, 2, \dots, n$ .

The validity of the result is easy to see. Suppose there is a variable  $x_j$  such that  $\partial f(x^*) / \partial x_j > 0$ . Then  $f(x)$  can be increased by increasing  $x_j^*$  by a small amount. Analogously, if  $\partial f(x^*) / \partial x_j < 0$ , then  $f(x)$  can be increased by decreasing  $x_j^*$  by a small amount. But unfortunately, without imposing further restrictions on the shape of  $f(x)$ , the necessary condition is not sufficient for a maximum:  $x^*$  may not maximize  $f(x)$  when all  $\partial f(x^*) / \partial x_j = 0$ . The illustration in figure 1.3.2 shows why. The derivative  $\partial f / \partial x_j = 0$  at points a, b, c, d, e as well as at g which gives the only global maximum.

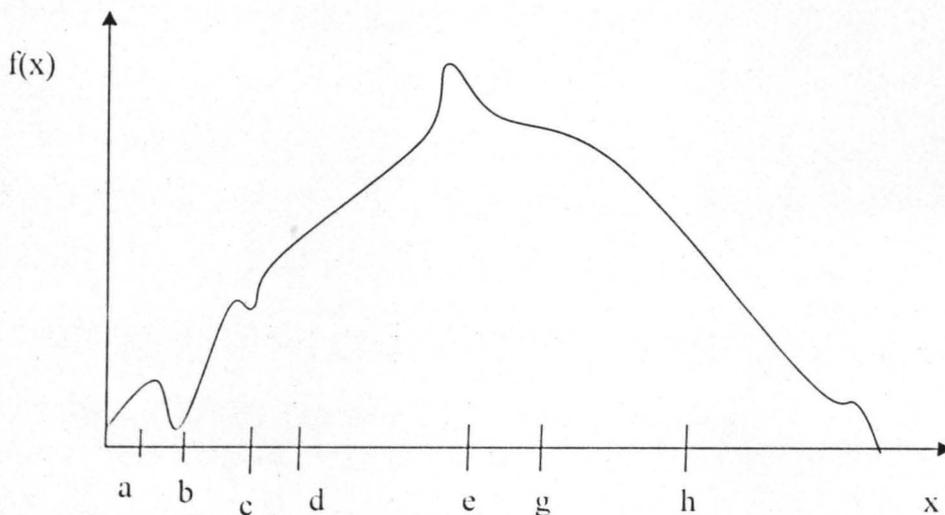


Figure 1.3.2 Example with Multiple Local Optima.

After identifying the critical points that satisfy the condition  $\partial f(x_1, x_2, \dots, x_n) / \partial x_j = 0$  at  $(x_1, x_2, \dots, x_n) = (x_1^*, x_2^*, \dots, x_n^*)$  for  $j = 1, 2, \dots, n$ , each such point would then be classified as a local minimum or maximum if the function is strictly convex or strictly concave respectively, within a neighbourhood of the point. The global minimum and maximum would be found by comparing the relative minima and maxima and then checking the value of the function as some of the variables approach  $-\infty$  or  $+\infty$ .

However, if the function is known to be convex or concave, then a critical point must be a global minimum or a global maximum respectively.

Algorithmic Description.

Many of the computational technique for maximizing  $f(x)$  can be expressed in a standardized format.

Step 1. Select an arbitrary initial trial point  $x^0$ .

Step 2. Terminate the iterations if  $\partial f / \partial x_j = 0$ , for  $j = 1, 2, \dots, n$  at the initial point  $x^k$ .

Otherwise, determine values  $y_j^k$ , for  $j = 1, 2, \dots, n$ , and continue on step 3.

Steps 3 calculate a new trial point.

$$x_j^{k+1} = x_j^k + y_j^k \text{ for } j = 1, 2, 3, \dots, n \quad \dots\dots\dots 1.3.5$$

Return to step 2, where  $x^{k+1}$  replace  $x^k$ .

For most nonlinear objective functions, the iterative process never obtains an  $x^k$  such that all  $\partial f / \partial x_j = 0$ .

Suppose you want to maximize  $f(x_1, x_2) = -(x_1 - 3)^2 - 4(x_2 - 2)^2$ .

You can tell by inspecting  $f(x)$  that the unique optima solution is  $x_1^* = 3, x_2^* = 2$ , yielding  $f(x^*) = 0$ . [since  $f(x)$  is a quadratic function, the optimal  $x$  can be calculated directly from the necessary conditions  $\partial f / \partial x_j = 0$ , for  $j = 1, 2, \dots, n$ .

Nevertheless, the quadratic example serves to exhibit the main ideas of the algorithm.] The contour levels of  $f(x_1, x_2)$  are ellipses starting with

Step 1, let  $x_1^0 = x_2^0 = 0$ , and hence  $f(x^0) = -25$ . since

$$\partial f / \partial x_1 = -2(x_1 - 3) \text{ and } \partial f / \partial x_2 = -8(x_2 - 2) \quad \dots\dots\dots 1.3.6$$

Step 2 Indicates further improvement is possible.

In preparation for choosing the dimensions  $d_1$  and  $d_2$ , draw the tangent line to the contour  $f(x)$  at  $x^0$ , given by the equation

$$\partial f (x_1 - x_1^0) / \partial x_1 + \partial f (x_2 - x_2^0) / \partial x_2 = 0. \quad \dots\dots\dots 1.3.7$$

For  $x_1^0 = x_2^0 = 0$ , (3) yields

$$6x_1 + 16x_2 = 0 \text{ or } x_2 = -3x_1/8$$

the value of  $f(x) = -25$

for  $d_1^0 = 1$  and  $d_2^0 = 2/3$

The algorithmic will continue to improve the values of  $x_1$  and  $x_2$ , also that of  $d_1$  and  $d_2$ .

### 1.3.1C A Nonlinear Function Of Several Constrained Variables

That is, optimizing a nonlinear function with nonlinear constraints. The aim here is to solve optimization problems containing nonlinear constraints. For the sake of definiteness, suppose the model is stated as

$$\text{Maximize } f(x_1, x_2, \dots, x_n) \cdot \dots \dots \dots 1.3.8$$

$$\text{Subject to } g_i(x_1, x_2, \dots, x_n) \leq 0, \quad i=1, 2, \dots, m \quad \dots \dots \dots 1.3.9$$

$$x_j \geq 0 \quad j = 1, 2, \dots, n \quad \dots \dots \dots 1.3.10$$

(1.3.8) and (1.3.9) above can be viewed as a canonical statement of a nonlinear programming problem (NPP). Here, the constraints function  $g_i(x)$  and objective function  $f(x)$  are to be postulated upon as follows:

#### Definition 1.3.1

Feasible Region.

The assumptions on each nonlinear function  $g_i(x)$  are given in terms of its shape and smoothness characteristics. To set the stage, a real value functions  $g(x)$  is defined to be convex if, for any two points  $x \neq y$ , and for all  $p$ ,  $0 \leq p \leq 1$ ,

$$pg(x_1, x_2, \dots, x_n) + (1-p)g(y_1, y_2, \dots, y_n) \geq g(px_1 + (1-p)y_1, \dots, px_n + (1-p)y_n) \text{ Convex} \\ \dots \dots \dots 1.3.11$$

and strictly convex if there is a strictly inequality ( $>$ ) for  $0 < p < 1$  (Note that if  $-g(x)$  is concave, then  $g(x)$  is given).

A related characteristics of a convex function is that for any two points  $x$  and  $y$ .

$$g(y) \geq g(x) + \sum_{j=1}^n \frac{\partial g(x)}{\partial x_j} (y_j - x_j) \quad \text{Convex} \dots\dots\dots 1.3.12$$

$g_i(x)$  in 1.3.9 satisfy the following shape and smoothness assumptions.

- (1) Each  $g_i(x)$  is uniquely defined, finite and convex for all values of  $(x_1, x_2, \dots, x_n)$
- (2) Each  $\partial g_i(x) / \partial x_j$  is continuous for all  $x$  satisfying the constraints in 1.3.9.

Definition 1.3.2

Objective Function: the function  $f(x)$  is also hypothesized to satisfy certain shape and smoothness assumptions (i) through (iv)

- (i)  $f(x)$  is single-valued and finite for each  $x$  satisfying the constraints 1.3.9
- (ii) Every partial derivative  $\partial f(x) / \partial x_j$  is a single-valued finite and continuous at each  $x$  satisfying the constraint 1.3.9
- (iii)  $f(x)$  possesses a finite maximum  $f^*$  over all values of  $x$  satisfying the constraints 1.3.9
- (iv)  $f(x)$  is concave over all values of  $x$  satisfying the constraints 1.3.9

It is the purpose of this chapter to develop the basic theory upon which methods devised to solve the nonlinear programming problem are typically based. Among the topics considered are the definitions of local and global optima, the necessary and sufficient conditions for identifying an optimizing point, and the mathematical constraints introduced into this identification process by nonlinearity.

The final section contains some applications of this material to nonlinear programming example problems.

1.4 Local and Global Optima

The concept of local and global optima plays an extremely important role in nonlinear programming.

Definition 1.4.1 Global Maximum (Unconstrained Problem): The unconstrained function  $f(x)$  is said to take on its global maximum at the point  $x^*$  if  $f(x) \leq f(x^*)$  for all  $x$  over which the function  $f(x)$  is defined.

Definition 1.4.2 Local Maximum (Unconstrained Problem). The unconstrained function  $f(x)$  is said to take on a local maximum at the point  $x^0$  if constants  $\epsilon$  and  $\delta$ ,  $0 < \epsilon < \delta$ , exist such that for all  $x$  satisfying  $0 < |x - x^0| < \delta$ ,  $f(x) \leq f(x^0)$ , where  $f(x)$  is defined for all points in some  $\delta$ -neighbourhood of  $x^0$ .

Figure 1.4.1 illustrates a local and global maximum for a univariate function. Notice from definition 1.4.1 and 1.4.2 that a global maximum is also a local maximum. A familiar theorem from differential calculus is now introduced, which states the necessary conditions for a point  $x^0$  to be a local (or global) maximum.

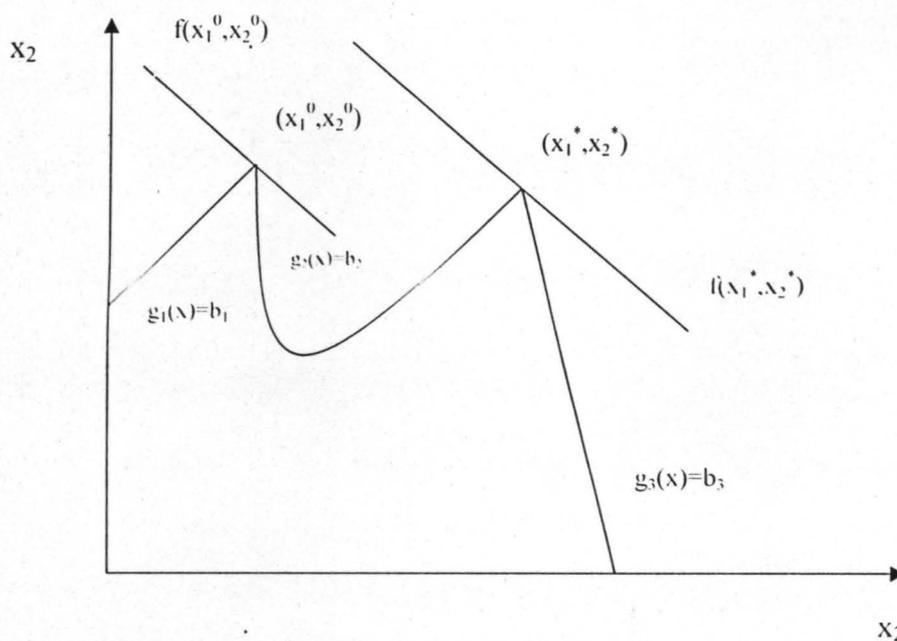


Figure 1.4.1 Illustration of local optima.

Theorem 1.4.1 If  $f(x)$  assumes a relative (local) maximum at  $x^0$ , then  $x^0$  must be a solution to the set of  $n$  equations

$$\frac{\partial f(x)}{\partial x_j} = 0, \quad j = 1, 2, \dots, n$$

Proof:

Suppose that  $f(x)$  assumes a local maximum at  $x^0$ , then from the definition of a local maximum, an  $\epsilon > 0$  must exist such that for all points  $x$  in a  $\delta$ -neighbourhood of  $x^0$ ,  $f(x) \leq f(x^0)$ . In particular, consider a point in the  $\delta$ -neighbourhood of  $x^0$  of the form  $x = x^0 + he_j$  where  $e_j = [0, 0, \dots, 0, 1, 0, \dots, 0]$  with the 1 placed in the  $j$ th position of  $e_j$  and  $0 < |h| < \epsilon$ . Then

$$f(x^0 + he_j) \leq f(x^0) \quad j = 1, 2, \dots, n \quad \dots\dots\dots 1.4.1$$

For all  $h, 0 < |h| < \epsilon$ . Dividing (1.4.1) by  $h$  results in the expressions

$$[f(x^0 + he_j) - f(x^0)] / h \leq 0 \text{ if } h > 0 \quad j = 1, 2, \dots, n \quad \dots\dots\dots 1.4.2$$

$$[f(x^0 + he_j) - f(x^0)] / h \geq 0 \text{ if } h < 0 \quad j = 1, 2, \dots, n \quad \dots\dots\dots 1.4.3$$

on taking the limit of (1.4.2) and (1.4.3) as  $h \rightarrow 0$ , it follows from the definition of a partial derivative that

$$\partial f(x^0) / \partial x_j \leq 0 \quad \text{for } h \rightarrow 0, \quad h > 0$$

$$\partial f(x^0) / \partial x_j \geq 0 \quad \text{for } h \rightarrow 0, \quad h < 0$$

Thus

$$\partial f(x^0) / \partial x_j = 0, \quad j = 1, 2, \dots, n \quad \dots\dots\dots 1.4.4$$

The condition in (1.4.4) can be conveniently displayed in vector notation in terms of the gradient vector of  $f(x)$ .

Definition 1.4.3. The Gradient Vector

The gradient vector of  $f(x) = f(x_1, x_2, \dots, x_n)$ , denoted by  $\nabla f(x)$ , is the  $n \times 1$  column vector whose components are in the first-order partial derivatives of  $f(x)$ .

$$\nabla f(x) = \left( \frac{\partial f(x)}{\partial x_1} \dots \frac{\partial f(x)}{\partial x_n} \right)^T \quad \dots\dots\dots 1.4.5$$

The condition in 1.4.4 stated in vector form is  $\nabla f(x^0) = 0$

If a point  $x^0$  satisfies 1.4.4, it might not be a maximum point. Theorem 1.4.1 provides only the necessary condition for  $x^0$  to be a maximizing point. In the univariate case 1.4.4 may be satisfied at a minimizing point, a maximizing point, or a point of inflection as illustrated in figure 1.4.2. in the n-multidimensional case where as illustrated in figure 1.4.2. in the n-multidimensional case where  $x^1 = [x_1, x_2, \dots, x_n]$ , the analogy to the univariate case is a maximizing point, minimizing point or saddle point. A saddle point is the multidimensional analogy to the inflection point in the univariate case. A saddle point for the bivariate case  $[x^1 = (x_1, x_2)]$  is illustrated in figure 1.4.3. The sufficient condition for  $x^0$  to be a maximizing point can be expressed as a property of the Hessian matrix of  $f(x)$ .

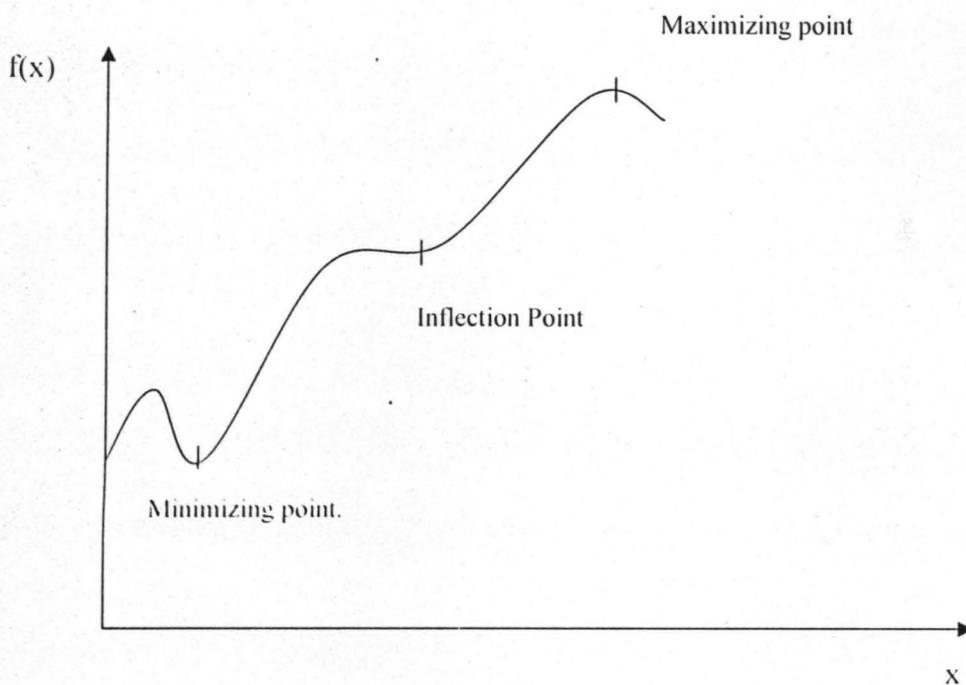


Figure 1.4.2 Possible Solution points to  $df(x)/dx_j = 0$

Definition 1.4.4. The Hessian Matrix.

The Hessian matrix of  $f(x) = f(x_1, x_2, \dots, x_n)$ , denoted by  $H(x)$ , is the  $n \times n$  matrix whose elements are second order partial derivatives of  $f(x)$ :

$$H(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(x)}{\partial x_2^2} & \dots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix} \dots\dots\dots 1.4.6$$

Theorem 1.4.2. A sufficient condition for  $f(x) = f(x_1, x_2, \dots, x_n)$  to have a local maximum at the point  $x^0$  where  $\nabla f(x^0) = 0$  is that the Hessian matrix  $H(x)$  be negative definite i.e, for any  $y^1 = (y_1, y_2, \dots, y_n)$ , except  $y = 0$ ,  $y^1 H(x)y < 0$ .

Proof:

This theorem can be proved by applying Taylor's theorem to the function  $f(x)$ . Taylor's theorem states that for any two points  $x_1$  and  $x_2 = x_1 + h$ , there exists a scalar  $\theta$ ,  $0 \leq \theta \leq 1$ , such that

$$f(x_2) = f(x_1) + \nabla f^1(x_1)h + 0.5h^1 H[\theta x_1 + (1-\theta)x_2]h \dots\dots\dots 1.4.7$$

Applying (1.4.7) to  $f(x)$ , where  $x_1 = x^0$  and  $x_2 = x^0 + h$ , produces the expression

$$f(x^0 + h) = f(x^0) + \nabla f^1(x^0)h + 0.5h^1 H[\theta x^0 + (1 - \theta)(x^0 + h)]h$$

Since  $\nabla f(x^0) = 0$ .

$$f(x^0 + h) = f(x^0) + 0.5h^1 H[\theta x^0 + (1 - \theta)(x^0 + h)]H$$

or

$$f(x^0 + h) - f(x^0) = 0.5h^1 H[\theta x^0 + (1 - \theta)(x^0 + h)]H \dots\dots\dots 1.4.8$$

If the right-hand side of 1.4.8 is negative for all  $h$  in a  $\delta$ -neighbourhood of  $x^0$ , by definition 1.4.2,  $x^0$  must be a local maximum, since  $f(x^0 + h) - f(x^0) \leq 0$  if this is the case.

The second partial derivatives  $\partial^2 f(x^0) / \partial x_i \partial x_j$  will have the same sign as

$\partial^2 f[\theta x^0 + (1 - \theta)(x^0 + h)] / \partial x_i \partial x_j$  provided that the point  $\theta x^0 + (1 - \theta)(x^0 + h)$  is in a suitable  $\delta$ -neighbourhood of  $x^0$ . Thus the right hand side of 1.4.8 is negative only if  $\lambda^1 H(x^0) < 0$ ; i.e. the Hessian matrix evaluated at  $x^0$ ,  $H(x^0)$ , must be negative definite to insure that  $x^0$  is a maximizing point.

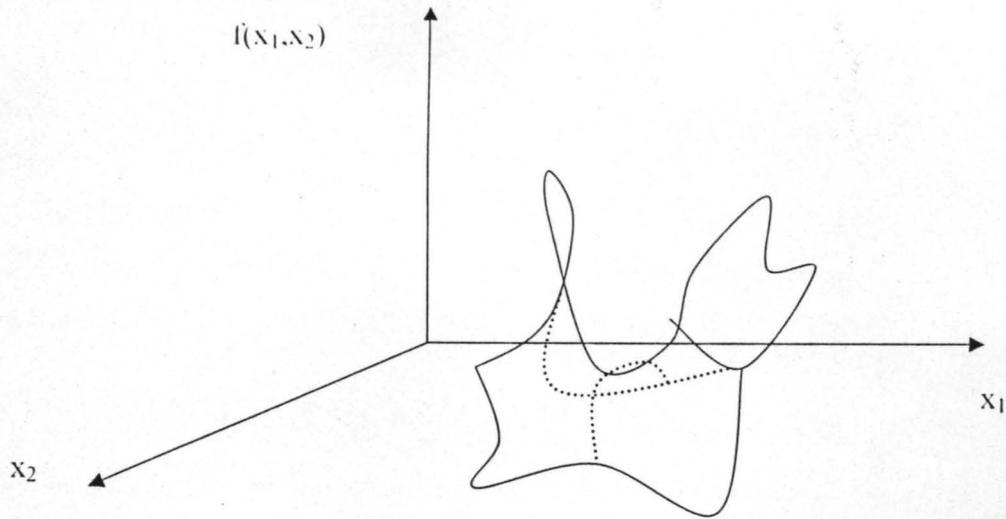


Figure 1.4.3. Two Dimensional Saddle Points.

We can now state and prove necessary and sufficient conditions, for every local minimum of a function to be global minimum we start with the following sufficient conditions.

Theorem 1.4.3

Let  $f$  be a real function on  $C \subset \mathbb{R}^n$  and let  $\bar{\alpha} = f(\bar{x})$ ,  $\bar{x} \in C$ . Suppose that  $\delta(f, \alpha)$  is lower semi-continuous (lsc) at  $\bar{\alpha}$ . If  $\bar{x}$  is a local minimum of  $f$ , then it is also a global minimum of  $f$  on  $C$ .

Proof:

Suppose that the hypothesis hold and  $\bar{x}$  is not a global minimum of  $f$  on  $C$ . Hence there exist a point  $\tilde{x} \in C$  such that

$$f(\tilde{x}) < f(\bar{x}) \quad \dots\dots\dots 1.4.9$$

Define the sequence  $\{\alpha^i\}$  by

$$\alpha^i = [1/i f(\tilde{x}) + (1 - 1/i) f(\bar{x})], \quad i = 1, 2, \dots\dots \quad \dots\dots\dots 1.4.10$$

Clearly,  $\lim_{i \rightarrow \infty} (\alpha^i) = f(\bar{x}) = \bar{\alpha}$  ..... 1.4.11

and  $\bar{x} \in S(f, \bar{\alpha})$ . From 1.4.9, 1.4.10 and 1.4.11, it follows that

$f(\tilde{x}^i) \leq \alpha^i \leq f(\bar{x}), i = 1, 2, \dots$  ..... 1.4.12

and  $\tilde{x}^i \in S(f, \alpha^i), i = 1, 2, \dots$  Hence  $\{\alpha^i\} \subset G_f$ .

Since  $S(f, \alpha)$  is assumed to be lsc at  $\bar{\alpha}$ , there exists a natural number  $K$  and a sequence  $\{x^i\}$ , converging to  $\bar{x}$ , such that  $x^i \in S(f, \alpha^i)$  for  $i = K, K+1, \dots$ . Hence

$f(x^i) \leq \alpha^i, i = K, K+1, \dots$  ..... 1.4.13

and by 1.4.9

$f(x^i) < f(\bar{x}), i = K, K+1, \dots$  ..... 1.4.14

since  $\{x^i\} \rightarrow \bar{x}$ , for a sufficient small  $\delta > 0$  there exists a natural number  $K_\delta$  such that

$x^i \in C \cap N_\delta(\bar{x}), i = K_\delta, K_\delta + 1, \dots$  which, by the hypothesis, will also satisfy

$f(x^i) \geq f(\bar{x}), i = K_\delta, K_\delta + 1, \dots$  ... 1.4.15, contradicting 1.4.14

Corollary 1.4.4

Let  $f$  be a real function on  $C \subset R^n$ . If  $S(f, \alpha)$  is lsc on  $G_f$  then every local minimum of  $f$  is also a global minimum of  $f$  on  $C$ .

We now prove a converse result of theorem 1.4.3

Theorem 1.4.5

Let  $f$  be a real function on  $C \subset R^n$  and let  $\bar{\alpha} \in G_f$ . Then  $S(f, \alpha)$  is lsc at  $\bar{\alpha}$  if any of the following assumptions hold,

- (i) Every  $x \in C$  satisfying  $f(x) = \bar{\alpha}$  is a global minimum of  $f$  on  $C$ .
- (ii) None of the points  $x \in C$  satisfying  $f(x) = \bar{\alpha}$  is a local minimum of  $f$ .

Proof:

Suppose that the hypothesis hold and  $S(f, \alpha)$  is not lsc at  $\bar{\alpha}$ . Then there exists an open set

$A \subset \mathbb{R}^n$  such that

$$A \cap S(f, \bar{\alpha}) \neq \emptyset \quad \dots\dots\dots 1.4.16$$

and for every  $\delta > 0$  there exists an  $\alpha(\delta) \in N_\delta(\bar{\alpha}) \cap G_f$  such that

$$A \cap S(f, \alpha(\delta)) = \emptyset \quad \dots\dots\dots 1.4.17$$

Therefore we can find a sequence  $\{\alpha^i\} \subset G_f$  converging to  $\bar{\alpha}$  such that

$$A \cap S(f, \alpha^i) = \emptyset, \quad i = 1, 2, \dots\dots \quad \dots\dots\dots 1.4.8$$

It follows that  $\alpha^i < \bar{\alpha}$  for all  $i$ , otherwise if  $\alpha^k \geq \bar{\alpha}$  for some  $k$ , then  $S(f, \bar{\alpha}) \subset S(f, \alpha^k)$ , and, by

1.4.16; we get a contradiction to 1.4.18: for every  $\bar{x} \in A \cap S(f, \bar{\alpha})$  it follows that  $\bar{x} \notin S(f, \alpha^i)$

for very  $i$ ; and since  $\{\alpha^i\} \rightarrow \bar{\alpha}$ , we have  $f(\bar{x}) = \bar{\alpha}$ . Moreover, for every  $x \in A$ ,

$$f(x) \geq \bar{\alpha} = f(\bar{x}).$$

Since  $A$  is an open set, it follows that  $\bar{x}$  must be a local minimum of  $f$ , and clearly,  $\bar{x}$  is not a global minimum of  $f$  of  $C$ . This result contradicts the hypotheses.

Corollary 1.4.6

Let  $f$  be a real function on  $C \subset \mathbb{R}^n$ . If every local minimum of  $f$  is a global minimum of  $f$  on  $C$ , then  $S(f, \alpha)$  is lsc on  $G_f$ .

Finally, as an immediate result of corollaries 1.4.4 and 1.4.6 we obtain:

Corollary 1.4.7

Let  $f$  be a real function on  $C \subset \mathbb{R}^n$ . Every local minimum of  $f$  is a global minimum of  $f$  on  $C$  if and only if  $S(f, \alpha)$  is lsc on  $G_f$ .

As an illustration consider example 1.3

Example 1.3

$$f(x) = f(x_1, x_2, x_3) = 16x_1 + 24x_2 - 4x_1^2 - 3x_2^2 - x_3^2$$

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \frac{\partial f(x)}{\partial x_3} \end{pmatrix} = \begin{pmatrix} -8x_1 + 16 \\ -6x_2 + 24 \\ -2x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

The condition  $\nabla f(x) = 0$  generates a system of three linear equations in three unknowns. The solution to this system is  $x^1 = [x_1, x_2, x_3] = [2, 4, 0]$ .

The Hessian matrix  $H(x)$  is now determined.

$$\frac{\partial^2 f(x)}{\partial x_1^2} = -8$$

$$\frac{\partial^2 f(x)}{\partial x_1 \partial x_2} = 0$$

$$\frac{\partial^2 f(x)}{\partial x_1 \partial x_3} = 0$$

$$\frac{\partial^2 f(x)}{\partial x_2^2} = -6$$

$$\frac{\partial^2 f(x)}{\partial x_2 \partial x_3} = 0$$

$$\frac{\partial^2 f(x)}{\partial x_3^2} = -2$$

Thus the Hessian matrix evaluated at  $x^0 = [2, 4, 0]$  is

$$H(x^0) = \begin{pmatrix} -8 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & -2 \end{pmatrix}$$

The scalar quantity  $y' H(x^0) y$  is

$$(y_1, y_2, y_3) \begin{pmatrix} -8 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = -8(y_1)^2 - 6(y_2)^2 - 2(y_3)^2$$

Which is clearly less than zero for any  $y' = [y_1, y_2, y_3]$ ;  $y \neq 0$ .

Thus  $x^0 = [2, 4, 0]$  is a maximizing point.

Definition 1.4.5 Global Maximum (Constrained Problem).

The function  $f(x)$  is said to take on its global maximum at the point  $x^*$  if  $f(x) \leq f(x^*)$  for all  $x$  (including  $x^*$ ) that belong to the feasible set of points  $x$ , where the set  $X$  represents the constraint region.

In the equality-constrained problem, for example,  $x$  belongs to  $X$  if  $x$  satisfies  $g_i(x) = b_i$ ,

$i = 1, 2, \dots, m$ .

Definition 1.4.6 Local Maximum (Constrained Problems).

The function  $f(x)$  is said to take on a local maximum at  $x^0$  if  $x^0$  belongs to  $X$  and there exists an  $\epsilon > 0$  such that for every  $x \neq x^0$  that belongs to  $X$  and is in an  $\epsilon$  - neighbourhood of  $x^0$ ,  $f(x) \leq f(x^0)$ .

1.5 Lagrangian Multipliers and Equality-Constrained Problems.

Before investigating the general non-linear programming given 1.3.8 – 1.3.10, it is necessary to first introduce the method of Lagrangian multipliers for solving the equality-constrained mathematical programming problem. The problem is specified as

$$\text{Maximize } z = f(x) = f(x_1, x_2, \dots, x_n) \quad \dots\dots\dots 1.5.1$$

$$\text{Subject to } g_i(x) = g_i(x_1, x_2, \dots, x_n) = b_i \quad \dots\dots\dots 1.5.2$$

$$i = 1, 2, 3, \dots, m.$$

The Lagrangian function corresponding to 1.5.1 and 1.5.2 is

$$L(x, \lambda) = f(x) + \sum_{j=1}^n \lambda_j [ b_j - g_j(x) ] \quad \dots\dots\dots 1.5.3$$

The necessary condition for a point  $[x^*, \lambda^*] = [x_1^*, x_2^*, \dots, x_n^*, \lambda_1^*, \lambda_2^*, \dots, \lambda_m^*]$  to maximize  $L(x, \lambda)$  are,

from Theorem 1.4.1,

$$\frac{\partial L(x, \lambda)}{\partial x_j} = \frac{\partial f(x)}{\partial x_j} - \sum_{j=1}^m \lambda_j \cdot \frac{\partial g_j(x)}{\partial x_j} = 0 \quad \dots\dots\dots 1.5.4$$

$$\frac{\partial L(x, \lambda)}{\partial x_j} = b_i - g_i(x) = 0 \quad i = 1, 2, \dots, m \quad \dots\dots\dots 1.5.5$$

For the general nonlinear programming problem, it will now be shown why a solution  $x^*$  to (1.5.4) and (1.5.5), which is a local maximum of  $L(x, \lambda)$  in (1.5.3), is also a local maximum for 1.5.1 and 1.5.2.

To determine this result, first assume that  $n = 2$  and  $m = 1$  so that the (1.5.1) and (1.5.2) problem is

$$\text{Maximize } z = f(x_1, x_2) \quad \dots\dots\dots 1.5.6$$

$$\text{Subject to } g_1(x_1, x_2) = b_1 \quad \dots\dots\dots 1.5.7$$

If the conditions of the implicit function theorem are satisfied, it must be possible to write  $x_2$  in terms of  $x_1$  so that  $x_2 = \phi_1(x_1)$ . The theorem then guarantees that  $\phi_1(x_1)$  is differentiable. The objective function can be written using  $\phi_1(x)$  as a univariate function in  $x_1$  and the (1.5.6) and (1.5.7) problem is equivalent to the unconstrained problem.

$$\text{Maximize } z = f[x_1, \phi_1(x_1)]$$

The necessary condition for  $x_1^0$  to be a local optimum of  $f(x_1, \phi_1(x_1))$  is

$$\frac{\partial f[x_1, \phi_1(x_1)]}{\partial x_1} = 0$$

But recall from differential calculus that the total derivative  $d/dx_1$  of  $f(x_1, x_2)$  can be written as

$$\frac{df(x_1, x_2)}{dx_1} = \frac{\partial f(x_1, x_2)}{\partial x_1} + \frac{\partial f(x_1, x_2)}{\partial x_2} \frac{dx_2}{dx_1} \quad \dots\dots\dots 1.5.8$$

But  $x_2 = \phi_1(x_1)$ . If  $\phi_1(x_1)$  is substituted for  $x_2$  in (1.5.8) and the total derivative  $df/dx_1$  is evaluated at  $(x_1^0, x_2^0)$ , +

$$\frac{df(x_1^0, x_2^0)}{dx_1} = \frac{\partial f(x_1^0, x_2^0)}{\partial x_1} + \frac{\partial f(x_1^0, x_2^0)}{\partial x_2} \cdot \frac{d\phi_1(x_1^0)}{dx_1} = 0 \quad \dots\dots\dots 1.5.9$$

since  $g_1(x_1, x_2) = b_1$

$$\frac{dg_1(x_1, x_2)}{dx_1} = \frac{\partial g_1(x_1, x_2)}{\partial x_1} + \frac{\partial g_1(x_1, x_2)}{\partial x_2} \frac{d\phi_1(x_1)}{dx_1} = 0 \quad \dots\dots\dots 1.5.10$$

where  $\phi_1(x_1)$  has been substituted for  $x_2$  in the last term. From (1.5.10)

$$\frac{d\phi_1(x_1)}{dx_1} = \frac{-\frac{\partial g_1(x_1, x_2)}{\partial x_1}}{\left[ \frac{\partial g_1(x_1, x_2)}{\partial x_2} \right]} \dots\dots\dots 1.5.11$$

Now substitute the right hand side of (1.5.11) for  $d\phi_1(x_1)/dx_1$  in (1.5.9) where  $d\phi_1(x_1)/dx_1$  is evaluated at  $(x_1^0, x_2^0)$ . Then

$$\frac{\partial f(x_1^0, x_2^0)}{\partial x_1} - \frac{\partial f(x_1^0, x_2^0)}{\partial x_2} \left[ \frac{\frac{\partial g_1(x_1^0, x_2^0)}{\partial x_1}}{\frac{\partial g_1(x_1^0, x_2^0)}{\partial x_2}} \right] = 0 \dots\dots 1.5.12$$

and define  $\lambda_1$  as

$$\lambda_1 = \left[ \frac{\frac{\partial f(x_1^0, x_2^0)}{\partial x_2}}{\frac{\partial g_1(x_1^0, x_2^0)}{\partial x_2}} \right] = 0$$

Then 1.5.12 can be written as

$$[\partial f(x_1^0, x_2^0) / \partial x_1] - \lambda_1 [\partial g_1(x_1^0, x_2^0) / \partial x_1] = 0 \dots\dots\dots 1.5.13$$

Directly from the definition  $\lambda_1$  it follows that

$$[\partial f(x_1^0, x_2^0) / \partial x_2] - \lambda_1 [\partial g_1(x_1^0, x_2^0) / \partial x_2] = 0 \dots\dots\dots 1.5.14$$

Additionally,  $(x_1^0, x_2^0)$  must satisfy

$$g_1(x_1, x_2) = b_1$$

Therefore, by using the implicit function theorem, it is possible to write the necessary conditions for determining a local maximum to (1.5.6) and (1.5.7) in the form (1.5.13) – (1.5.14)

Now consider the Lagrangian function corresponding to (1.5.6) and (1.5.7).

$$L(x, \lambda) = L(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda_1 [b_1 - g_1(x_1, x_2)]$$

The necessary conditions for the maximizing  $L(x, \lambda)$  are, from Theorem 1.4.1,

$$\frac{\partial L(x, \lambda)}{\partial x_1} = \frac{\partial f(x_1, x_2)}{\partial x_1} - \lambda_1 \frac{\partial g_1(x_1, x_2)}{\partial x_1} = 0 \quad \dots\dots\dots 1.5.16$$

$$\frac{\partial L(x, \lambda)}{\partial x_2} = \frac{\partial f(x_1, x_2)}{\partial x_2} - \lambda_1 \frac{\partial g_1(x_1, x_2)}{\partial x_2} = 0 \quad \dots\dots\dots 1.5.17$$

$$\frac{\partial L(x, \lambda)}{\partial \lambda_1} = b_1 - g_1(x_1, x_2) = 0 \quad \dots\dots\dots 1.5.18$$

The necessary conditions for a point  $x^0$  to maximize  $L(x, \lambda)$ , given by (1.5.16) – (1.5.18), are necessary conditions for the equality-constrained problem in (1.5.6) and (1.5.7).

It is possible to extend the above argument from  $m = 2$  case to the general  $n$  – variate case to show that the necessary conditions to maximize the Lagrangian function  $L(x, \lambda)$  in (1.5.2) are equivalent to the necessary conditions to maximize  $f(x)$  in the equality constrained problem (1.5.1) and (1.5.2).

### 1.6 Kuhn Tucker Conditions.

In 1951 Kuhn and Tucker [1951] developed necessary and sufficient solutions for optimal solutions to the general mathematical programming problem. The Kuhn Tucker conditions provided a frame work from which numerous computational method have been developed for solving certain types of the nonlinear programming problem. A thorough treatment of the conditions is therefore essential before chapters two and three. Furthermore, the Kuhn-Tucker conditions provide considerable height into the nature of the nonlinear problem.

In developing the necessary conditions, the nonlinear programming problem in (1.3.8) and (1.3.9) is recast in the form

$$\begin{aligned}
&\text{Maximize } z = f(x) && \dots\dots\dots 1.6.1 \\
&\text{Subject to } g_i(x) \leq b_i && i = 1, 2, \dots, r \\
&g_i(x) \geq b_i && i = r + 1, \dots, s \\
&g_i(x) = b_i && i = s + 1, \dots, m \\
&x \geq 0 && \dots\dots\dots 1.6.2
\end{aligned}$$

Normally, equality constraints will not often occur in mathematical programming problems. The inclusion of equality constraints does perhaps surprisingly introduce further difficulties in the solving of this problem, adding  $r$  slack variables  $x_{ti}$ ,  $i = 1, 2, \dots, r$

And  $s$  surplus variables  $x_{ti}$ ,  $i = r + 1, \dots, s$  the above problem may be rewritten as

$$\begin{aligned}
&\text{Maximize } Z = f(x) \\
&\text{Subject to } g_i(x) + x_{ti} = b_i && i = 1, 2, \dots, r \\
&g_i(x) - x_{ti} = b_i && i = r + 1, \dots, s \\
&g_i(x) = b_i && i = s + 1, \dots, m \\
&&& \dots\dots\dots 1.6.3
\end{aligned}$$

where  $[x_1, x_2]^T = [x_1, x_2, \dots, x_n; x_{t1}, x_{t2}, \dots, x_{ts}]$

By applying Lagrangian function technique to this problem. We have for this problem the Lagrangian function as:

$$L(x, \lambda) = f(x) + \sum_{i=1}^r [\lambda_i \cdot [b_i - g_i(x) - x_{ti}]] + \sum_{i=r+1}^s \lambda_i [b_i - g_i(x) - x_{ti}] + \sum_{i=s+1}^m \lambda_i [b_i - g_i(x)]$$

..... 1.6.4

Where  $\lambda_i$ ,  $i = 1, 2, \dots, m$ , are the Lagrangian multipliers associated with the constraints in 1.6.3.

By differentiating  $L(x, \lambda)$  with respect to  $\lambda$  and  $x$  (including  $x_{ti}$ ) and setting these differentials equal to zero, a set of necessary conditions for optimizing the unconstrained function  $L(x, \lambda)$  results; they are

$$\frac{\partial L(x, \lambda)}{\partial x_j} = \frac{\partial f(x)}{\partial x_j} - \sum_{i=1}^m \lambda_i \cdot \frac{\partial g_i(x)}{\partial x_j} = 0 \quad j = 1, 2, \dots, n \quad \dots\dots\dots 1.6.5$$

$$\frac{\partial L(x, \lambda)}{\partial \lambda_i} = b_i - g_i(x) - x_{ii} = 0 \quad i = 1, 2, \dots, r \quad \dots\dots\dots 1.6.6$$

$$\frac{\partial L(x, \lambda)}{\partial \lambda_i} = b_i - g_i(x) + x_{ii} = 0 \quad i = r+1, \dots, S \quad \dots\dots\dots 1.6.7$$

$$\frac{\partial L(x, \lambda)}{\partial \lambda_i} = b_i - g_i(x) = 0 \quad i = s+1, \dots, m \quad \dots\dots\dots 1.6.8$$

$$\frac{\partial L(x, \lambda)}{\partial x_{ii}} = -\lambda_i = 0 \quad i = 1, 2, \dots, r \quad \dots\dots\dots 1.6.9$$

$$\frac{\partial L(x, \lambda)}{\partial x_{ii}} = +\lambda_i = 0 \quad i = 1, 2, \dots, s \quad \dots\dots\dots 1.6.10$$

Thus if  $x_{ii} > 0$  for  $i = 1, 2, \dots, s$ , then  $\lambda_i = 0, i = 1, 2, \dots, s$ . By 1.6.9 and 1.6.10 it also follows that at the optimizing point  $x_{ii}^*, \lambda_i^* = 0$ ,

Since if  $x_{ii} > 0$ , then  $\lambda_i^* = 0$ .

1.6.1 The Lagrangian Solution method involves the following steps:

(1) Find the constrained maximum of  $f(x)$ . Frequently, by inspecting the function, it is apparent that the Unconstrained maximum will not be feasible, so that this step may be deleted. If this solution is feasible, it will be the global maximum, and there is no need to proceed to the next step.

(2) Solve the Lagrangian function based only on the  $m-s$  equality constraints  $g_i(x) = b_i$ ,

$i = s + 1, \dots, m$ . If this solution satisfies the remaining constraints, it will be global maximum of  $f(x)$  and the process may be stopped.

(3) Add one of the inequality constraints to the Lagrangian function in (2) treating it as if it were active. Solve this Lagrangian system. If the solution satisfies the remaining  $s-1$  constraints, stop. Otherwise drop the current inequality constraint, add another, and repeat the process. If all  $s$  inequality constraints fail to yield a feasible solution when treated individually as equality constraints proceed to (4).

(4) Repeat the process by now adjoining pairs of inequality constraints to the Lagrangian function in (2), treating them as active constraints. Continue until a feasible solution to the  $s-2$  remaining constraints is encountered or all  $C_2^s = S!/2!(S-2)!$  pairs have been exhausted. If the latter occurs proceed to (5).

(5) Continue the process taking all  $C_a^s$  combinations for  $a = 3, 4, \dots, 5$  until a feasible solution is encountered.

Two example problems presented illustrate the Lagrangian algorithm and the application of the Kuhn-Tucker conditions

#### Example 1.4

The problem is

$$\text{Maximize } z = f(x_1, x_2) = -(x_1 - 11)^2 - 4(x_2 - 6)^2$$

$$\text{Subject to } 2x_1 + x_2 \leq 18$$

$$x_1 + 2x_2 \leq 16$$

$$x_1, x_2 \geq 0$$

From the graphic representation of the problem given in figure 1.6.1, it is apparent that the solution occurs at intersection of the two lines  $2x_1 + x_2 = 18$  and  $x_1 + 2x_2 = 16$ . The Lagrangian algorithm will now be applied to verify this conjecture and to illustrate the technique.

Step 1:

The unconstrained global maximum from inspection if  $f(x_1, x_2)$  occurs at (11,6), which clearly is not feasible.

Step 2:

From the Lagrangian function using the constraint  $x_1 + 2x_2 = 16$ .

$$L_1(x, \lambda) = -(x_1 - 11)^2 - 4(x_2 - 6)^2 + \lambda_1(16 - x_1 - 2x_2)$$

$$\frac{\partial L_1(x, \lambda)}{\partial x_1} = -2(x_1 - 11) - \lambda_1 = 0$$

$$\frac{\partial L_1(x, \lambda)}{\partial x_2} = -8(x_2 - 6) - 2\lambda_1 = 0$$

$$\frac{\partial L_1(x, \lambda)}{\partial \lambda_1} = 16 - x_1 - 2x_2 = 0$$

..... 1.6.11

The solution to 1.6.11 is  $x_1 = 7.5$ ,  $x_2 = 4.25$ ,  $\lambda_1 = 7$ , and  $\lambda_2 = 0$ . since the point (7.5, 4.25) is not feasible, the process continues.

Step 3:

Form the lagrangian function using the constraints  $2x_1 + x_2 = 18$

$$L_2(x, \lambda) = -(x_1 - 11)^2 - 4(x_2 - 6)^2 + \lambda_2(18 - 2x_1 - x_2)$$

$$\frac{\partial L_2(x, \lambda)}{\partial x_1} = -2(x_1 - 11) - 2\lambda_2 = 0$$

$$\frac{\partial L_2(x, \lambda)}{\partial x_2} = -8(x_2 - 6) - \lambda_2 = 0$$

$$\frac{\partial L_2(x, \lambda)}{\partial \lambda_2} = 18 - 2x_1 - x_2 = 0$$

..... 1.6.12

The solution to (1.6.12) is  $x_1 = 6.3$ ,  $x_2 = 5.4$ ,  $\lambda_1 = 0$  and  $\lambda_2 = 4.8$ .

Since (6.3, 5.4) also is not feasible, the process continues.

Step 4:

Form the Lagrangian function using both constraints  $x_1 + 2x_2 = 16$  and  $2x_1 + x_2 = 18$ .

$$L_3(x, \lambda) = -(x_1 - 11)^2 - 4(x_2 - 6)^2 + \lambda_1(16 - x_1 - 2x_2) + \lambda_2(18 - 2x_1 - x_2)$$

$$\left. \begin{aligned} \frac{\partial L_3(x, \lambda)}{\partial x_1} &= -2(x_1 - 11) - \lambda_1 - 2\lambda_2 = 0 \\ \frac{\partial L_3(x, \lambda)}{\partial x_2} &= -8(x_2 - 6) - 2\lambda_1 - \lambda_2 = 0 \\ \frac{\partial L_3(x, \lambda)}{\partial \lambda_1} &= 16 - x_1 - 2x_2 = 0 \\ \frac{\partial L_3(x, \lambda)}{\partial \lambda_2} &= -18 - 2x_1 - x_2 = 0 \end{aligned} \right\} \dots\dots\dots 1.6.13$$

The solution to 1.6.13 is  $x_1 = 6.67$ ,  $x_2 = 4.67$ ,  $\lambda_1 = 4.2$  and  $\lambda_2 = 2.33$ . The point  $(6.67, 4.67)$  is the intersection of the two lines and hence is feasible. Therefore, the global maximum occurs at the point  $x^* = (6.67, 4.67)$  and  $f(x^*)$  is  $-25.82$ .

Since the constraint space is convex and  $f(x_1, x_2)$  is a concave function, the Kuhn-Tucker conditions are also sufficient so that  $(6.67, 4.67)$  is the global maximizing point of  $f(x_1, x_2)$ .

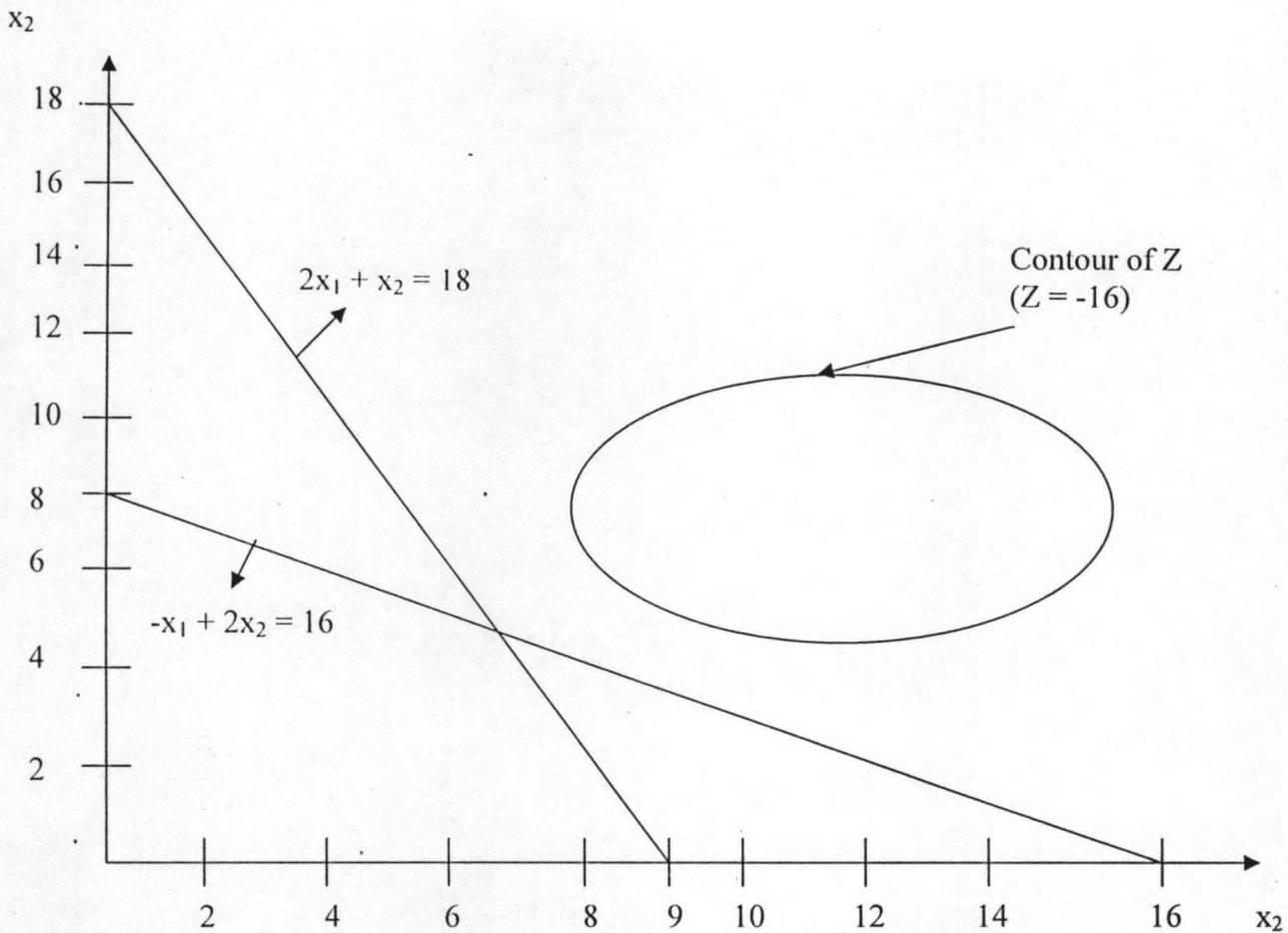


Figure 1.6.1. Graphic solution to example 1.2.

### Example 1.5

The problem is

$$\text{Maximize } Z = f(x_1, x_2) = -(x_1 - 4)^2 - (x_2 - 4)^2$$

$$\text{Subject to } x_1 + x_2 \leq 4$$

$$x_1^2 + x_2^2 = 4$$

$$x_1, x_2 \geq 0$$

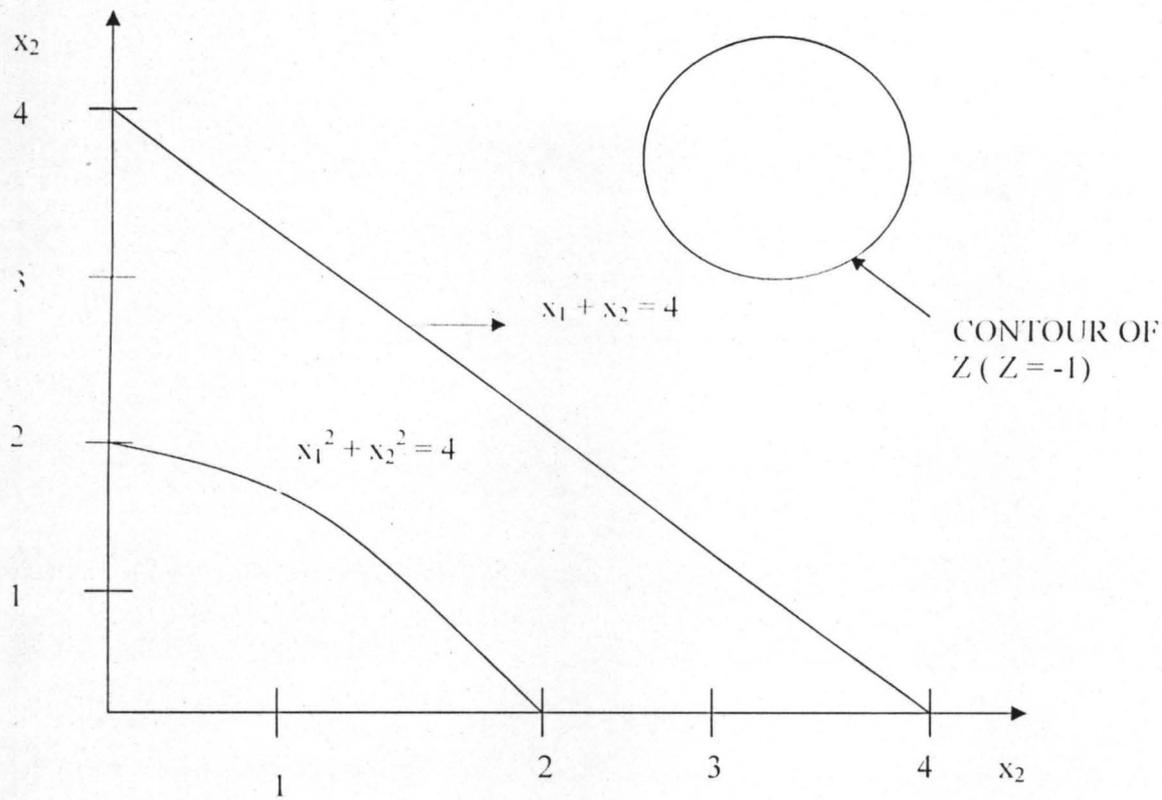


Figure 1.6.2. Graphic solution to example 1.5.

The Lagrangian method will be used to solve this problem also;

Step 1:

The unconstrained maximum from inspection  $f(x_1, x_2)$  occurs at (4, 4), which is not feasible.

Step 2:

Form the Lagrangian function using the constraint  $x_1 + x_2 = 4$

$$L_1(x, \lambda) = -(x_1 - 4)^2 - (x_2 - 4)^2 + \lambda_1(4 - x_1 - x_2)$$

$$\frac{\partial L_1(x, \lambda)}{\partial x_1} = -2(x_1 - 4) - \lambda_1 = 0$$

$$\frac{\partial L_1(x, \lambda)}{\partial x_2} = -2(x_2 - 4) - \lambda_1 = 0$$

$$\frac{\partial L_1(x, \lambda)}{\partial \lambda_1} = 4 - x_1 - x_2 = 0$$

..... 1.6.14

The solution to 1.6.14 is  $x_1 = x_2 = 2$ ,  $\lambda_1 = 4$ , and  $\lambda_2 = 0$ . Since  $(2, 2)$  is not feasible, the process continues.

Step3

Form the Lagrangian function using the constraint  $x_1^2 + x_2^2 = 4$ .

$$L_2(x, \lambda) = -(x_1 - 4)^2 - (x_2 - 4)^2 + \lambda_2(4 - x_1^2 - x_2^2)$$

$$\frac{\partial L_2(x, \lambda)}{\partial x_1} = -2(x_1 - 4) - 2x_1\lambda_2 = 0$$

$$\frac{\partial L_2(x, \lambda)}{\partial x_2} = -2(x_2 - 4) - 2x_2\lambda_2 = 0$$

$$\frac{\partial L_2(x, \lambda)}{\partial \lambda_2} = 4 - x_1^2 - x_2^2 = 0$$

..... 1.6.15

The solution to 1.6.15 is  $x_1 = x_2 = (2)^{1/2}$ ,  $\lambda_1 = 0$ , and  $\lambda_2 = 1.83$ , since  $((2)^{1/2}, (2)^{1/2})$  is feasible, the maximum is  $f((2)^{1/2}, (2)^{1/2})$ .

Summary

This chapter discussed generally the nonlinear programming (function) for both single variable and the several variables. Also, the chapter was able to treat some conditions that give the local or global optimum. Some of the theorems that deal with the local and global optima conditions were stated in the chapter.

## 2.0 OPTIMIZATION PROBLEMS WITH SEPARABLE OBJECTIVE.

### 2.1 INTRODUCTION

An approximate solution method known as separable programming is presented for the problem.

Maximize

$$Z = f(x) = \sum_{j=1}^n f_j(x_j) \quad \dots\dots\dots 2.1.1$$

Subject to

$$g_i = \sum_{j=1}^n g_{ij}(x_j) \quad \{\leq = \geq\} b_i$$

$$i = 1, 2, 3, \dots, m \quad \dots\dots\dots 2.1.2$$

$$j = 1, 2, 3, \dots, n \quad \dots\dots\dots 2.1.3$$

This model required that  $m + 1$  functions  $Z, g_1, g_2, \dots, g_m$  are separable in the  $n$  variables  $x_1, x_2, \dots, x_n$ , if the problem does not satisfy the separability requirement. It may be possible, through the transformation of variable, to recast the problem so that it does.

To illustrate the basic mechanism upon which the approximation procedure is based, consider for example the univariate function  $f(x) = 8 + 4x - x^2$ , which is plotted in figure 2.2.1. By connecting lines at points on  $f(x)$  at equal intervals of one unit of length on the  $x$ -axis beginning at origin. It is possible to approximate  $f(x)$  reasonably well in a piecewise linear fashion. Denote the interval remarking points on the  $x$ -axis by  $x_k, k = 0, 1, 2, \dots, 5$  and  $f(x_k)$  by  $f_k$ . It is now possible to write down the approximating piecewise linear function, denoted by  $\Gamma(x')$ , for each of the five straight lines over the range  $0 \leq x \leq 5$  in figure 2.1

#### 2.1.1 Definition Of Separable Programming Problem

When the nonlinear objective or constraint(s) or both can be expressed as piecewise linear functions it is then said to be a separable programming problem

## 2.2 Transformation of nonlinear programming Problem (NLPP) to separable programming problem (SPP).

Recalling that the equation for a straight line can be expressed as  $f(x) = y = mx + c$

$$= f_k + ((f_{k+1} - f_k)(x - x_k))/(x_{k+1} - x_k) \quad \dots\dots\dots 2.2.1$$

$$k = 0, 1, 2, 3, 4$$

For example, the line segment connecting the point (0,8) with (1,11) is given by

$$f(x) = 8 + [(11 - 8)/(1 - 0)](x - 0) = 3x + 8$$

Also, the line segment connecting the point (1,11) with (2,12) is given by

$$f(x) = 11 + [(12 - 11)/(2 - 1)](x - 1) = x + 10 \text{ e.t.c.}$$

It is possible to use  $f(x)$  to approximate approximate  $f(x)$  at some point  $x$ , say between the  $k$ th and  $k+1$  points on the  $x$ -axis. By defining a constant  $\lambda$  that must satisfy  $0 \leq \lambda \leq 1$ ,

$$x = \lambda x_{k+1} + (1 - \lambda)x_k \quad \dots\dots\dots 2.2.2$$

Solving 2.2.2 for  $x - x_k$  gives  $x - x_k = \lambda(x_{k+1} - x_k)$ ; and upon substituting this into 2.2.1,

$$\begin{aligned} f(x) &= f_k + ((f_{k+1} - f_k)\lambda(x_{k+1} - x_k))/(x_{k+1} - x_k) \\ &= f_k + \lambda(f_{k+1} - f_k) \quad \dots\dots\dots 2.2.3 \end{aligned}$$

or

$$= \lambda f_{k+1} + (1 - \lambda)f_k$$

In 2.2.2 let  $\lambda = \lambda_{k+1}$  and  $(1 - \lambda) = \lambda_k$ , thus

$$x = \lambda_k x_k + \lambda_{k+1} x_{k+1}$$

where  $x$  is between  $x_k$  and  $x_{k+1}$ , it is possible to determine unique values for  $\lambda_k$  and  $\lambda_{k+1}$  such that

$$x = \lambda_k x_k + \lambda_{k+1} x_{k+1} \quad \dots\dots\dots 2.2.4$$

$$f(x) = \lambda_k f_k + \lambda_{k+1} f_{k+1} \quad \dots\dots\dots 2.2.5$$

where

$$+ \lambda_{k+1} = 1 \quad \dots\dots\dots 2.2.6$$

$$, \lambda_{k+1} \geq 0 \quad \dots\dots\dots 2.27$$

ustration 2.1

s an illustration of using the above formula, suppose that it is desired in the example to provide  
 t approximate value of  $f(x)$  when  $x = 5/4$ . From 2.2.4 and 2.2.6 it follows that  $\lambda_1 = 3/4$  and  $\lambda_2 = 1/4$   
 nce  $5/4$  lies between  $x_1 = 1$  and  $x_2 = 2$ . Since  $f_1 = 11$  and  $f_2 = 12$ , it follows from 2.2.5 that  $f(5/4)$   
 $(3/4)11 + (1/4)12 = 11.25$  where  $f(5/4) = 11.44$ .

2.1 General Piecewise Linear Approximation Technique For Univariate Function

he above formulae can be generated by allowing  $r$  segments (not necessarily of equal length)  
 ver the range of  $x$ ; Specified by  $0 \leq x \leq b$  where  $b$  is an upper bound on  $x$ ; for any  $x, 0 \leq x \leq b$ .

$$x = \sum_{k=0}^r \lambda_k x_k \quad \dots\dots\dots 2.2.8$$

$$f(x) = \sum_{k=0}^r \lambda_k f_k \quad \dots\dots\dots 2.2.9$$

$$\sum_{k=0}^r \lambda_k = 1 \quad \dots\dots\dots 2.2.10$$

$$\lambda_k \geq 0, \quad k = 0, 1, 2, \dots, r \quad \dots\dots\dots 2.2.11$$

with the additional condition that no more than two of  $\lambda_k$  shall be positive, and if two are positive,  
 they must be adjacent.

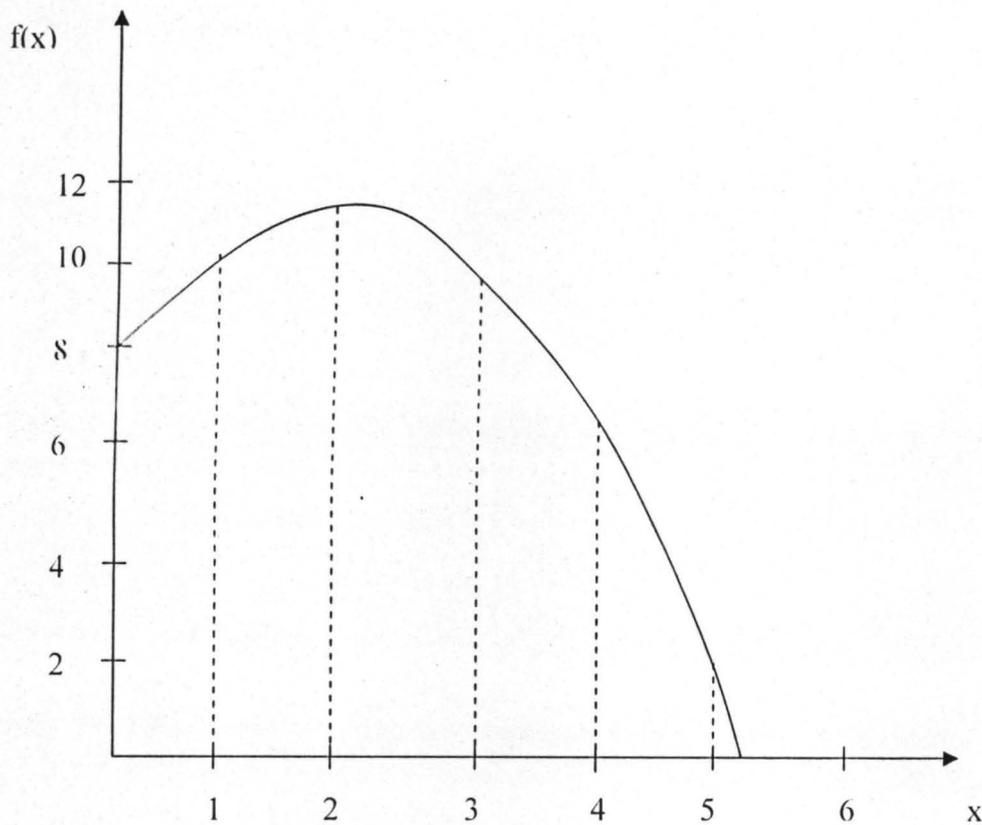


Figure 2.2.1 Illustration of a piecewise linear approximation of a Curvilinear.

### 2.2.2 The General Piecewise Linear Approximation Technique for n Variables Functions.

The general piecewise linear approximation technique described above for a univariate function may be extended in a straightforward manner to functions of n variables if the functions are separable. For example, suppose

$$f(x_1, x_2) = (x_1 - 2)^2 + (x_2 - 4)^2$$

Since  $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$  where

$$f_1(x_1) = (x_1 - 2)^2 \text{ and } f_2(x_2) = (x_2 - 4)^2,$$

The results given in 2.4.1 – 2.4.4 may be applied to each of the functions  $f_1(x)$  and  $f_2(x_2)$ , the

$$f_1(x_1) = \hat{f}_1(x_1) = \sum_{k=0}^{r_1} \lambda_{k1} f_{k1}$$

$$f_2(x_2) = f_2(x_2) = \sum_{k=0}^{r_2} \lambda_{k2} f_{k2}$$

$$\text{where } x_1 = \sum_{k=0}^{r_1} \lambda_{k1} x_{k1} \quad \text{and} \quad x_2 = \sum_{k=0}^{r_2} \lambda_{k2} x_{k2}$$

The function  $f(x_1, x_2)$  above may now be approximated by  $f_1(x_1) + f_2(x_2)$ .

Notice that it is possible to use a different number of intervals on  $x_1$  and  $x_2$  axes;  $r_1$  might not equal  $r_2$ .

The general piecewise linear approximation rules may now be stated for a separable n-variate function  $f(x_1, x_2, \dots, x_n)$ ;

$$x_j = \sum_{k=0}^{r_j} \lambda_{kj} x_{kj} \quad j = 1, 2, \dots, n \quad \dots\dots\dots 2.2.1$$

$$f_j(x_j) = \sum_{k=0}^{r_j} \lambda_{kj} f_{kj} \quad j = 1, 2, \dots, n \quad \dots\dots\dots 2.2.2$$

$$\sum_{k=0}^{r_j} \lambda_{kj} = 1 \quad j = 1, 2, \dots, n \quad \dots\dots\dots 2.2.3$$

$$\lambda_{kj} \geq 0, \quad j = 1, 2, \dots, n$$

$$k = 0, 1, 2, \dots, r_j$$

\dots\dots\dots 2.2.4

And for any given  $j$ , no more than two  $\lambda_{kj}$  may be positive, and if exactly two are positive, they must be adjacent. In 2.2.2  $f_{kj} = f_j(x_k)$ .

These results may be applied directly to the functions  $f_j$  and  $g_{ij}$  in 2.1.1 and 2.1.2, generating the following approximating functions;

$$f_j(x_j) = \sum_{k=0}^{r_j} \lambda_{kj} f_{kj} \quad j = 1, 2, \dots, n$$

$$g_{ij}(x_j) = \sum_{k=0}^{r_j} \lambda_{kj} g_{kij} \quad \begin{matrix} i = 1, 2, \dots, m \\ j = 1, 2, \dots, n \end{matrix}$$

Where 2.2.3, 2.2.4 and the adjacency condition are satisfied. The original problem in 2.1.1 to 2.1.3 may now be replaced with the following approximating problem:

$$\text{Maximize } z = f(x) = \sum_{j=1}^n \sum_{k=0}^{r_j} \lambda_{kj} \cdot f_{kj} \quad \dots\dots\dots 2.2.5$$

$$\text{Subject to } \sum_{j=1}^n \sum_{k=0}^{r_j} \lambda_{kj} g_{kij} \{ \leq \geq \} b_i \quad i = 1, 2, \dots, m \quad \dots\dots\dots 2.2.6$$

$$\sum_{k=0}^{r_j} \lambda_{kj} = 1 \quad j = 1, 2, \dots, n \quad \dots\dots\dots 2.2.7$$

$$\lambda_{kj} \geq 0, \quad j = 1, 2, \dots, n; \quad k = 0, 1, 2, \dots, r_j \quad \dots\dots\dots 2.2.8$$

And the condition that for any given  $j$ , no more than two  $\lambda_{kj}$  may be positive; if exactly two are positive, they must be adjacent.

If it were not for this condition, 2.2.5 – 2.2.8 would delineate a linear programming problem in the  $\lambda_{kj}$ . However, the simplex algorithm may be applied if the basis entry is restricted so that not more than two positive  $\lambda_{kj}$  are allowed for any  $j$ ; if exactly two  $\lambda_{kj}$

occur for some  $j$ , they must be adjacent. Once a solution in the  $\lambda_{kj}$  has been determined in the fashion, 2.2.1 may be used to convert them to  $x_j$  values.

### 2.3 Optimization Algorithm for Separable Objective.

Here, separable programming is utilized to solve optimization problem. A continuous nonlinear separable function is considered, approximated by a piecewise linear function, and then solve the final problem by the simplex algorithm. This is essentially what the following algorithm does.

Step I: Express the functions involved in the problem as separate functions.

Step II: Partition the domain of each variable into a number of segments.

Step III: Evaluate the separate functions at the end point of segment.

Step IV: Generate a linear programming problem in the  $\lambda_{jk}$  with suitable constraints on the  $\lambda_{jk}$ .

Step V: Solve for the  $\lambda_{jk}$

Step VI: Solve for the original variables.

Example 2.2 For illustration purpose consider the problem

$$\text{Maximize } Z = f(x_1, x_2) = 3x_1 + 2x_2$$

$$\text{Subject to } g(x_1, x_2) = 4x_1^2 + x_2^2 \leq 16$$

$$x_1, x_2 \geq 0$$

The problem is illustrated in figure in figure 2.3.1. From this figure it is apparent that by selecting  $0 \leq x_1 \leq 2$  and  $0 \leq x_2 \leq 4$ , the feasible region will be considered. Suppose now that  $r_1 = 2$  and  $r_2 = 4$  are selected.

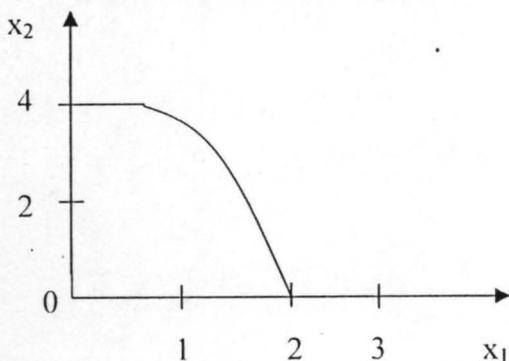


Figure 2.3.1 Illustration of a nonlinear function.

Step I: Express the functions as separable functions

This can be done very easily by utilizing 2.2.9. Since it can be easily determined that the functions are indeed separable in our illustrative example this is the case, and so

$$f(x_1, x_2) = f_1(x_1) + f_2(x_2) \text{ where}$$

$$f_1(x_1) = 3x_1, \quad f_2(x_2) = 2x_2$$

Note, however, that the first constraint is also a separable function and we must also operate on it.

Since equation 2.2.2 is general we can say, for all constraints, that

$$g_i = g_{i1}(x_1) + g_{i2}(x_2) + \dots + g_{in}(x_n) \quad i = 1, 2, \dots, m$$

In this case we have only the one constraint to concern ourselves with and we have

$$g_1(x_1, x_2) = g_{11}(x_1) + g_{12}(x_2)$$

where  $g_{11}(x_1) = 4x_1^2$ ,  $g_{12}(x_2) = x_2^2$

$x_{1k}$	$x_{2k}$	$\dots$	$x_{nk}$	$f_1(x_{1k})$	$f_2(x_{2k})$	$\dots$	$f_n(x_{nk})$	$g_{11}(x_{1k})$	$g_{12}(x_{2k})$	$\dots$	$g_{1n}(x_{nk})$	$\dots$	$g_{m1}(x_{1k})$	$g_{m2}(x_{2k})$	$\dots$	$g_{mn}(x_{nk})$
----------	----------	---------	----------	---------------	---------------	---------	---------------	------------------	------------------	---------	------------------	---------	------------------	------------------	---------	------------------

Figure 2.3.2 Tabular Format for Evaluating Functions.

Using the above format, a table outline for illustrative problem is to be construct below. And the table is to be filled. At  $k = 0$ ,  $x_{1k} = x_{1,0}$  and, this brings the lower range of our domain of interests, obviously  $x_{10} = 0$ . Furthermore, at  $k = 1$ ,  $x_{1,1} = 1$  (from step 2 above) and  $x_{1,2} = 2$ . The column for  $x_{1k}$  and  $x_{2k}$  should be completed before starting with that of column for  $f_1(x_{1k})$ .

From step 1 above

$$f_1(x_{1k}) = 3x_{1k}$$

And at  $k = 0$                        $x_{1,1} = 0$

Thus  $f_1(x_{1k}) = 0$                        $k = 0$

Similarly  $f_1(x_{1k}) = 3$                        $k = 1$

The evaluation of the separate functions as required by this step of the algorithm should be completed before the next step.

k	$x_{1k}$	$x_{2k}$	$g_{11}(x_{1k})$	$g_{12}(x_{2k})$	$f_1(x_{1k})$	$f_2(x_{2k})$
0						
1						
2						
3						
4						

Figure 2.3.3 Table Outline for illustrative problem

0	0	0	0	0	0	0
1	1	1	4	1	3	2
2	2	2	16	4	6	4
3		3		9		6
4		4		16		8

Step IV generate LPP in the  $\lambda_{jk}$  with suitable constraints on the  $\lambda_{jk}$

Maximize

$$f = 0\lambda_{10} + 3\lambda_{11} + 6\lambda_{12} + 0\lambda_{20} + 2\lambda_{21} + 4\lambda_{22} + 6\lambda_{23} + 8\lambda_{24}$$

$$\text{subject to } g_1 = 0\lambda_{10} + 4\lambda_{11} + 16\lambda_{12} + 0\lambda_{20} + \lambda_{21} + 4\lambda_{22} + 9\lambda_{23} + 16\lambda_{24} \leq 16$$

$$\lambda_{10} + \lambda_{11} + \lambda_{12} = 1$$

$$\lambda_{20} + \lambda_{21} + \lambda_{22} + \lambda_{23} + \lambda_{24} = 1$$

$$\lambda_{jk} \geq 0 \quad j = 1, 4; k = 0, 1, 2, 3, 4$$

This can be written as follows

$$Z - f = 0\lambda_{10} + 3\lambda_{11} + 6\lambda_{12} + 0\lambda_{20} + 2\lambda_{21} + 4\lambda_{22} + 6\lambda_{23} + 8\lambda_{24}$$

$$g_1 + \lambda = 0\lambda_{10} + 4\lambda_{11} + 16\lambda_{12} + 0\lambda_{20} + \lambda_{21} + 4\lambda_{22} + 9\lambda_{23} + 16\lambda_{24} + \lambda = 16$$

$$\lambda_{10} + \lambda_{11} + \lambda_{12} = 1$$

$$\lambda_{20} + \lambda_{21} + \lambda_{22} + \lambda_{23} + \lambda_{24} = 1$$

$$\lambda_{jk} \geq 0 \quad j = 1, 2; k = 1, 2, 3, 4.$$

This can be put in the Tabular form as seen below

Initial Tableau (Fig. 2.3.5)

Cost coeff:	0	3	6	0	2	4	6	8	0	Basic	Index
Variable:	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	$\lambda$	Variable	b
	0	4	16	0	1	4	9	16	1	$\lambda$	16
	1	1	1	0	0	0	0	0	0	$\lambda_{10}$	1
	0	0	0	1	1	1	1	1	0	$\lambda_{20}$	1
$c_j - z_j$ :	0	3	6	0	2	4	6	8	0		0

From the simplex criterion for basis entry variable  $\lambda_{24}$  is selected to enter. However,  $\lambda_{20}$  is already in the basis so that  $\lambda_{24}$  cannot enter (two  $\lambda$ s in the set can be in a basis only if they are adjacent) unless  $\lambda_{20}$  departs. The simplex procedure for basis departure (16/16, 1/0, 1/1) indicates that either  $\lambda$  or  $\lambda_{20}$  can be eliminated. Therefore,  $\lambda_{24}$  is allowed to enter the basis and  $\lambda_{20}$  departs.

The solution to the Initial Tableau is  $\lambda = 16$ ,  $\lambda_{10} = 1$  and  $\lambda_{20} = 1$  with  $\lambda_{24}$  entering the new basis and  $\lambda_{20}$  exiting the first tableau is given in figure 2.3.6

Figure 2.3.6

Cost coeff:	0	3	6	0	2	4	6	8	0	Basic	Index
Variable:	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	$\lambda$	Variable	b
	0	4	16	-16	-15	-12	-7	0	1	$\lambda$	0
	1	1	1	0	0	0	0	0	0	$\lambda_{10}$	1
	0	0	0	1	1	1	1	1	0	$\lambda_{24}$	1
$c_j - z_j$ :	0	3	6	-8	-6	-4	-2	0	0		-8

The solution to the first tableau is  $\lambda = 0$ ,  $\lambda_{10} = 1$  and  $\lambda_{24} = 1$ .  $\lambda_{12}$  should enter next, but this is impossible for if it enters,  $\lambda$  would depart, leaving  $\lambda_{10}$  in the basis not adjacent to  $\lambda_{12}$ . The next best contender is  $\lambda_{11}$ . If  $\lambda_{11}$  enters  $\lambda$  departs leaving two adjacent  $\lambda$ s in the first set in

the basis, namely  $\lambda_{10}$  and  $\lambda_{11}$ . Thus  $\lambda_{11}$  enters the basis, and the slack variable  $\lambda$  departs.

The second tableau is given in figure 2.3.7

Figure 2.3.7

Cost coeff:	0	3	6	0	2	4	6	8	0	Basic	Index
Variable:	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	$\lambda$	Variable	b
	0	1	4	-4	-15/4	-3	-7/4	0	1/4	$\lambda_{11}$	0
	1	0	-3	4	15/4	3	7/4	0	-1/4	$\lambda_{10}$	1
	0	0	0	1	1	1	1	1	0	$\lambda_{24}$	1
$c_j - z_j$ :	0	0	-6	4	21/4	5	13/4	0	-3/4		-8

The solution to the second Tableau is  $\lambda_{11} = 0$ ,  $\lambda_{10} = 1$  and  $\lambda_{24} = 1$ . According to the simplex criterion  $\lambda_{21}$  should enter the basis next, but this is not possible since  $\lambda_{10}$  would have to depart, leaving two non adjacent  $\lambda$  s i.e.  $\lambda_{21}$  and  $\lambda_{24}$  in the basis. Therefore the only  $\lambda$  that can enter is  $\lambda_{23}$  with the  $\lambda_{10}$  departing. The third Tableau is given in figure 2.3.8

Figure 2.3.8 (Third Tableau)

Cost coeff:	0	3	6	0	2	4	6	8	0	Basic	Index
Variable:	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	$\lambda$	Variable	b
	0	1	1	0	0	0	0	0	0	$\lambda_{11}$	1
	4/7	0	-12/7	16/7	15/7	12/7	1	0	-1/7	$\lambda_{23}$	4/7
	-4/7	0	12/7	-9/7	-8/7	-5/7	0	1	1/7	$\lambda_{24}$	3/7
$c_j - z_j$ :	-13/7	0	-24/7	-24/7	-12/7	-4/7	0	0	-2/7		-69/7

The solution to the third Tableau is  $\lambda_{11} = 1$ ,  $\lambda_{23} = 4/7$  and  $\lambda_{24} = 3/7$ . Upon checking the simplex criterion, it is found that this solution is optimal. From (2.4.1) it follows that

$$x_1^* = (0)\lambda_{10} + (1)\lambda_{11} + (2)\lambda_{12} = 1$$

$$x_2^* = (0)\lambda_{20} + (1)\lambda_{21} + (2)\lambda_{22} + (3)\lambda_{23} + (4)\lambda_{24} = 24/7 \text{ or } 3.43$$

and from step IV above  $f^* = 3(1) + 6(4/7) + 8(3/7) = 9.86$

And the original problem yields  $3(1) + 2(3.43) = 9.86$

Example 2.2. This example is an improved fashion of the same example considered above. Using just 5 segments for  $x_1$  variable as against 3 segments used for the Example 2.1.

k	$x_{1k}$	$x_{2k}$	$g_{11}(x_{1k})$	$g_{12}(x_{2k})$	$f_1(x_{1k})$	$f_2(x_{2k})$
0	0	0	0	0	0	0
1	.5	1	1	1	1.5	2
2	1	2	4	4	3	4
3	1.5	3	9	9	4.5	6
4	2	4	16	16	6	8

Figure 2.3.9 The completed Evaluation Table.

Step IV Generate a linear programming problem in the  $\lambda_{jk}$ . The 2.2.2 is utilized here and this is to be combined with the evaluation in figure 2.3.4, the illustrative problem now becomes

$$\text{Maximize } f(x_1, x_2) = \sum_{k=0}^4 \lambda_{1k} f_{1k} + \sum_{k=0}^4 \lambda_{2k} f_{2k} \quad \dots\dots\dots 2.3.1$$

$$= 0\lambda_{10} + 1.5\lambda_{11} + 3\lambda_{12} + 4.5\lambda_{13} + 6\lambda_{14} + 0\lambda_{20} + 2\lambda_{21} + 4\lambda_{22} + 6\lambda_{23} + 8\lambda_{24}$$

$$\text{Subject to } g_1(x_1, x_2) = \sum_{k=0}^4 \lambda_{1k} g_{1k} + \sum_{k=0}^4 \lambda_{2k} g_{2k} \quad \dots\dots\dots 2.3.2$$

$$= 0\lambda_{10} + \lambda_{11} + 4\lambda_{12} + 9\lambda_{13} + 16\lambda_{14} + 0\lambda_{20} + \lambda_{21} + \lambda_{22} \cdot 4 + 9\lambda_{23} + 16\lambda_{24}$$

The approximating linear programming problem (LPP) becomes

Maximize  $z = f^* =$

$$0\lambda_{10} + \lambda_{11} \cdot 1.5 + 3\lambda_{12} + 4.5\lambda_{13} + 6\lambda_{14} + 0\lambda_{20} + \lambda_{21} \cdot 2 + \lambda_{22} \cdot 4 + 6\lambda_{23} + 8\lambda_{24} \quad \dots\dots\dots 2.3.3$$

$$\text{Subject to } g' = 0\lambda_{10} + \lambda_{11} + 4\lambda_{12} + 9\lambda_{13} + 16\lambda_{14} + 0\lambda_{20} + \lambda_{21} + \lambda_{22} + 9\lambda_{23} + 16\lambda_{24} \leq 16$$

..... 2.3.4

$$\sum_{k=0}^4 \lambda_{1,k} = \lambda_{10} + \lambda_{11} + \lambda_{12} + \lambda_{13} + \lambda_{14} = 1 \quad \text{..... 2.3.5}$$

$$\sum_{k=0}^4 \lambda_{2,k} = \lambda_{20} + \lambda_{21} + \lambda_{22} + \lambda_{23} + \lambda_{24} = 1 \quad \text{..... 2.3.6}$$

Definition 2.3.1 Property Of Adjacent weight ( $\lambda$ 's)

Given that all the constraints are linear, if  $f(x)$  in 2.1.1 is to be maximized, and  $f_j(x_j)$  is a concave function, then every optimal solution to the approximate model based on 2.4.1 – 2.4.4 contains only one  $\lambda_k$ , or at most, two adjacent weight  $\lambda_k$  and  $\lambda_{k+1}$  at positive levels. The same solution property holds if  $f(x)$  is to be minimized and  $f_j(x_j)$  is convex.

Now, for ease in utilizing the optimization algorithm in subsequent steps the above adjacency property must be observed, that is no more other than two of the  $\lambda$ 's associated with  $x_1$  and  $x_2$  are greater than zero, and if two  $\lambda$ 's associated with a variable are greater than zero, then the fact that those two  $\lambda$ 's must be adjacent points has not been stated mathematically as a constraints. However, it must be remembered always that we have such a constraint as we move from one table to another. It should be noted that the Linear Programming problem (LPP) depends on three constraints and ten variables.

Step V Solve for the  $\lambda_{jk}$

This is a familiar ground and by introducing a slack variable  $\lambda$ , the first constraint can be written as equality. Artificial variable may also be introduced to the second and third constraints and proceed to use the simplex algorithm to solve the piecewise linear approximation if the basis entry restriction is satisfied.

The approximating LPP can now be rewritten as follows

$$g_1 + \lambda = 0\lambda_{10} + \lambda_{11} + 4\lambda_{12} + 9\lambda_{13} + 16\lambda_{14} + 0\lambda_{20} + \lambda_{21} + \lambda_{22} + 9\lambda_{23} + 16\lambda_{24} + \lambda = 16$$

$$\lambda_{10} + \lambda_{11} + \lambda_{12} + \lambda_{13} + \lambda_{14} = 1$$

$$\lambda_{20} + \lambda_{21} + \lambda_{22} + \lambda_{23} + \lambda_{24} = 1$$

and  $z - f =$

$$0\lambda_{10} + \lambda_{11} \cdot 1.5 + 3\lambda_{12} + 4.5\lambda_{13} + 6\lambda_{14} + 0\lambda_{20} + \lambda_{21} \cdot 2 + \lambda_{22} \cdot 4 + 6\lambda_{23} + 8\lambda_{24} - z = 0$$

This can be put in the tabular form as shown below:

Figure 2.3.5 (Initial Tableau)

Cost coeff:	0	1.5	3	4.5	6	0	2	4	6	8	0	Basic	Index
Variable:	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	$\lambda$	Variable	b
	0	1	4	9	16	0	1	4	9	16	1	$\lambda$	16
	1	1	1	1	1	0	0	0	0	0	0	$\lambda_{10}$	1
	0	0	0	0	0	1	1	1	1	1	0	$\lambda_{20}$	1
$c_j - z_j$ :	0	1.5	3	4.5	6	0	2	4	6	8	0		0

From the simplex criterion for basis entry variable  $\lambda_{24}$  is selected to enter. However,  $\lambda_{20}$  is already in the basis so that  $\lambda_{24}$  cannot enter (two  $\lambda$ 's in the set can be in a basis only if they are adjacent) unless  $\lambda_{20}$  departs. The simplex procedure for basis departure (16/16, 1/0, 1/1) indicates that either  $\lambda$  or  $\lambda_{20}$  can be eliminated. Therefore,  $\lambda_{24}$  is allowed to enter the basis and  $\lambda_{20}$  departs.

The solution to the initials tableau is  $\lambda = 16$ ,  $\lambda_{10} = \lambda_{20} = 1$  with  $\lambda_{24}$  entering the new basis and  $\lambda_{20}$  exiting. The 1<sup>st</sup> Tableau is given in figure 2.3.11

Figure 2.3.11 (1<sup>st</sup> tableau)

Cost coeff:	0	1.5	3	4.5	6	0	2	4	6	8	0	Basic	Index
Variable:	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	$\lambda$	Variable	b
	0	1	4	9	16	-16	-15	-12	-7	0	1	$\lambda$	0
	1	1	1	1	1	0	0	0	0	0	0	$\lambda_{10}$	1
	0	0	0	0	0	1	1	1	1	1	0	$\lambda_{24}$	1
$C_j - Z_j$ :	0	1.5	3	4.5	6	-8	-6	-4	-2	0	0		-8

The solution to the first Tableau is  $\lambda = 0, \lambda_{10} = \lambda_{24} = 1$ .  $\lambda_{14}$  should enter next, but this is impossible for if it enters,  $\lambda$  would depart, leaving  $\lambda_{10}$  in the basis not adjacent to  $\lambda_{24}$ . Also  $\lambda_{13}, \lambda_{12}$  would leave two non-adjacent  $\lambda$ s in the basis. The next best contender is  $\lambda_{11}$ . If  $\lambda_{11}$  enters  $\lambda$  departs leaving two adjacent  $\lambda$ s in the first set in the basis, namely  $\lambda_{10}$  and  $\lambda_{11}$ . Thus  $\lambda_{11}$  enter the basis, and the slack variable  $\lambda$  departs. The second Tableau is given in figure 2.3.12

Figure 2.3.12 (Second tableau)

Cost coeff:	0	1.5	3	4.5	6	0	2	4	6	8	0	Basic	Index
Variable:	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	$\lambda$	Variable	b
	0	1	4	9	16	-16	-15	-12	-7	0	1	$\lambda_{11}$	0
	1	0	-3	-8	-15	16	15	12	7	0	-1	$\lambda_{10}$	1
	0	0	0	0	0	1	1	1	1	1	0	$\lambda_{24}$	1
$C_j - Z_j$ :	0	0	-3	-9	-18	16	16.5	14	8.5	0	-1.5		-8

The solution to the second Tableau is  $\lambda_{11} = 0, \lambda_{10} = 0, \lambda_{24} = 1$ .

$\lambda_{21}$  or  $\lambda_{22}$  should not enter the basis, they are not adjacent to  $\lambda_{24}$ . The best contender is  $\lambda_{23}$ . that is

$\lambda_{20}$  would leave the basis and  $\lambda_{23}$  will enter the basis in the figure 2.3.13

Figure 2.3.13 (Third Tableau)

Cost coeff:	0	1.5	3	4.5	6	0	2	4	6	8	0	Basic	Index
Variable:	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	$\lambda$	Variable	b
	1	1	1	1	1	0	0	0	0	0	0	$\lambda_{11}$	0
	1/7	0	-3/7	-8/7	-15/7	16/7	15/7	12/7	1	0	-1/7	$\lambda_{23}$	1/7
	-1/7	0	3/7	8/7	15/7	-9/7	-8/7	-5/7	0	1	1/7	$\lambda_{24}$	6/7
$c_j - z_j$ :	-17/14	0	-9/14	5/7	3/14	-24/7	-12/7	-4/7	0	0	-2/7		-129/14

The solution to the third Tableau is  $\lambda_{11}=1$ ,  $\lambda_{23} = 1/7$  and  $\lambda_{24} = 6/7$ .

$\lambda_{13}$  should enter the basis, then  $\lambda_{24}$  would leave the basis leaving two non adjacent  $\lambda$ s in the basis i.e.  $\lambda_{11}$  and  $\lambda_{13}$ . the next best contender is  $\lambda_{12}$ .  $\lambda_{11}$  would leave the basis and  $\lambda_{12}$  would enter in the figure 2.3.14

Figure 2.3.14 (Fourth Tableau)

Cost coeff:	0	1.5	3	4.5	6	0	2	4	6	8	0	Basic	Index
Variable:	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	$\lambda$	Variable	b
	1	1	1	1	1	0	0	0	0	0	0	$\lambda_{12}$	1
	4/7	3/7	0	-5/7	-12/7	16/7	15/7	12/7	1	0	-1/7	$\lambda_{23}$	4/7
	-4/7	-3/7	0	5/7	12/7	-9/7	-8/7	-5/7	0	1	1/7	$\lambda_{24}$	3/7
$c_j - z_j$ :	-13/14	-9/14	0	1/14	-3/14	-24/7	-12/7	-4/7	0	0	-2/7		-138/14

The solution to the fourth Tableau is  $\lambda_{12} = 1$ ,  $\lambda_{23} = 4/7$  and  $\lambda_{24} = 3/7$ .  $\lambda_{13}$  should enter next,  $\lambda_{24}$  would leave the basis for  $\lambda_{13}$  to enter the basis in the figure 2.3.15 (Fifth Tableau)

Figure 2.3.15 (Fifth Tableau)

Cost coeff:	0	1.5	3	4.5	6	0	2	4	6	8	0	Basic	Index
Variable:	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	$\lambda$	Variable	B
	9/5	8/5	1	0	-7/5	9/5	8/5	1	0	-7/5	-1/5	$\lambda_{12}$	2/5
	0	0	0	0	0	1	1	1	1	1	0	$\lambda_{23}$	1
	-4/5	-3/5	0	1	12/5	-9/5	-8/5	-1	0	7/5	1/5	$\lambda_{24}$	3/5
$C_j - Z_j$ :	-9/5	-3/5	0	0	-3/5	-33/10	-8/5	-1/2	0	-1/10	-3/10		-693/70

The solution to the fifth tableau is  $\lambda_{12} = 2/5$ ,  $\lambda_{13} = 3/5$  and  $\lambda_{23} = 1$ . Upon checking the simplex criterion, it is found that this solution is optimal. From 2.2.1 it follows that

$$x_1^* = (0)\lambda_{10} + .5\lambda_{11} + \lambda_{12} + 1.5\lambda_{13} + 2\lambda_{14} = \left(\frac{2}{5}\right)1 + 1.5\left(\frac{3}{5}\right) = 1.3$$

$$x_2^* = (0)\lambda_{20} + \lambda_{21} + \lambda_{22} \cdot 2 + \lambda_{23} \cdot 3 + 4\lambda_{24} = 3(1) = 3$$

And from Step IV above  $f^* = 3(2/5) + 4.5(3/5) + 6(1) = 9.9$

The original problem yields  $3x_1 + 2x_2 = 9.9$

Comparing this optimal value with the other one before it, we are able to see that the more the segments considered, the better the piecewise linear approximation procedure (or value).

The exact optimal solution to the above problem can be found using the Lagrangian multipliers method.

#### 2.4 By Lagrangian method

(1) Form the lagrangian function

$$L(x, \lambda) = 3x_1 + 2x_2 + \lambda(16 - 4x_1^2 - x_2^2) \quad \dots\dots\dots 2.4.1$$

$$\frac{\partial L(x, \lambda)}{\partial x_1} = 3 - 8\lambda x_1 = 0$$

$$x = 3/8\lambda \quad \dots\dots\dots 2.4.2$$

$$\frac{\partial L(x, \lambda)}{\partial x_2} = 2 - 2\lambda x_2 = 0$$

$$\rightarrow x_2 = 1/\lambda \quad \dots\dots\dots 2.4.3$$

$$\text{also, } \frac{\partial L(x, \lambda)}{\partial \lambda} = 16 - 4x_1^2 - x_2 = 0$$

$$\rightarrow 4x_1^2 + x_2^2 = 16 \quad \dots\dots\dots 2.4.4$$

Substitute for 2.4.2 and 2.4.3 in 2.4.4;

$$36/64\lambda^2 + 1/\lambda^2 = 16 \quad \dots\dots\dots 2.4.5$$

$$(36 + 64)/64\lambda^2 = 16$$

$$\therefore \lambda = \pm 5/16 \quad \dots\dots\dots 2.4.6$$

$$\text{so, } x_1^* = 3/8(5/16) = 1.2 \quad \dots\dots\dots 2.4.7$$

$$\text{Also, } x_2^* = 1/(5/16) = 3.2 \quad \dots\dots\dots 2.4.8$$

The function yields  $3x_1 + 2x_2 = 10$ .

This value is the exact value of the problem, which is also the optimal value. This shows that the first solution to the approximating problem is fairly good in this case.

The following should be noted when dealing with the approximate model i.e if every  $f_j(x_j)$  is concave, and if every  $g_{ij}(x_j)$  is convex, then a solution obtained by the simplex method will be a global optimal solution for the model . But if these conditions are not satisfied, then by modifying the simplex method, we can obtain at least a local optimum for the approximate model.

Nonlinear Objective with Nonlinear Constraints

More examples considered as follows

Example 2.4

Solve the problem.

$$\text{Maximize } x_1^2 - x_1 + x_2$$

$$\text{Subject to } x_1 + x_2^2 \leq 4, \quad x_1, x_2 \geq 0$$

Solution:

### Step I

$$f(x_1, x_2) = f_1(x_1) + f_2(x_2) \text{ where}$$

$$\text{where } f_1(x_1) = x_1^2 - x_1, \quad f_2(x_2) = x_2$$

$$g_i = g_{i1}(x_1) + g_{i2}(x_2) + \dots + g_{in}(x_n) \quad i = 1, 2, \dots, m$$

$$g_1(x_1, x_2) = g_{11}(x_1) + g_{12}(x_2)$$

$$\text{where } g_{11}(x_1) = x_1, \quad g_{12}(x_2) = x_2^2$$

### Step II

From the original problem we see that both  $x_1$  and  $x_2$  must be greater than zero.

The first constraint indicates that  $x_1 \leq 4$  and  $x_2 \leq 2$ . (The variable does not necessarily need to have the same domain). In our own case, let us partition the domain of each variable into four segments, thus we will have five grid points.

### Step III

$$\text{At } k = 0, \quad x_{1k} = x_{10} = 0$$

$$\text{At } k = 1, \quad x_{1k} = x_{11} = 1 \text{ e.t.c.}$$

$$\text{Also } f_1(x_{1k}) = x_{1k}^2 - x_{1k}$$

$$k = 0 \quad x_{10} = 0$$

$$\text{Thus } f_1(x_{1k}) = 0 \quad k = 0$$

$$\text{Similarly, } f_1(x_{1k}) = 0 \quad k = 1$$

Use the table below to evaluate the separate functions.

Figure 2.3.16

k	$x_{1k}$	$x_{2k}$	$g_{11}(x_{1k})$	$g_{12}(x_{2k})$	$f_1(x_{2k})$	$f_2(x_{2k})$
0	0	0	0	0	0	0
1	1	.5	1	.25	0	.5
2	2	1	2	1	2	1
3	3	1.5	3	2.25	6	1.5
4	4	2	4	4	12	2

The completed Evaluation Table.

Step IV

The original problem can now be written as follows

$$\text{Maximize } f(x_1, x_2) \approx \sum_{k=0}^4 \lambda_{1k} f_{1k} + \sum_{k=0}^4 \lambda_{2k} f_{2k}$$

$$= 0\lambda_{10} + \lambda_{11}(0) + 2\lambda_{12} + 6\lambda_{13} + 12\lambda_{14} + 0\lambda_{20} + \lambda_{21}(0.5) + \lambda_{22} + 1.5\lambda_{23} + 2\lambda_{24}$$

$$\text{Subject to } g_1(x_1, x_2) \approx \sum_{k=0}^4 \lambda_{1k} g_{1k} + \sum_{k=0}^4 \lambda_{2k} g_{2k}$$

$$0\lambda_{10} + \lambda_{11} + 2\lambda_{12} + 3\lambda_{13} + 4\lambda_{14} + 0\lambda_{20} + \lambda_{21}(.25) + \lambda_{22} + 2.25\lambda_{23} + 4\lambda_{24} \leq 4$$

$$\sum_{k=0}^4 \lambda_{1k} = \lambda_{10} + \lambda_{11} + \lambda_{12} + \lambda_{13} + \lambda_{14} = 1$$

$$\sum_{k=0}^4 \lambda_{2k} = \lambda_{20} + \lambda_{21} + \lambda_{22} + \lambda_{23} + \lambda_{24} = 1$$

$$\lambda_{jk} \quad j = 1, 2; \quad k = 0, 1, 2, 3, 4$$

Step V

By introducing a slack variable  $s$ , we can write the first constraint as an equality. Proceed by applying Simplex method to solve the piecewise linear functions as follows;

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	S	b
S	0	1	2	3	.4	0	.25	1	2.25	4	1	4
$\lambda_{10}$	1	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	0	1	1	1	1	1	0	1
	0	0	2	6	12	0	.5	1	1.5	2	0	0

Initial tableau (figure 2.3.17)

By letting  $\lambda_{14}$  enter the basis  $\lambda_{14}$  will replace S while  $\lambda_{10}$  and  $\lambda_{20}$  remain the basis. Clearly  $\lambda_{10}$  and  $\lambda_{14}$  are not adjacent points and this is a situation we can not tolerate. Alternatively, we allow  $\lambda_{14}$  to replace  $\lambda_{10}$ , in that case we would have  $s$ ,  $\lambda_{14}$  and  $\lambda_{20}$  in our basis, a perfect condition as shown in the next table.

First tableau (Figure 2.3.18)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	S	b
S	-4	-3	-2	-1	0	0	.25	1	2.25	4	1	0
$\lambda_{14}$	1	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	0	1	1	1	1	1	0	1
	-12	-12	-10	-6	.0	0	.5	1	1.5	2	0	-12

Improved solution.

The first tableau yields the solution  $\lambda_{14} = 1$ ,  $\lambda_{20} = 1$ , all other variables = 0. Either  $\lambda_{24}$ ,  $\lambda_{23}$ ,  $\lambda_{22}$  or  $\lambda_{21}$  would make  $b_i$  negative or the adjacency condition would not be met, if any other variable

enters. We have therefore found the optimal solution to our linear piecewise approximation. It now only remains to translate our solution into terms of the original variables  $x_1$  and  $x_2$ .

Step VI

$$x_1^* = \sum_{k=0}^4 \lambda_{1k} x_{1k} = (0)(0) + (0)(1) + (0)(2) + (0)(3) + (1)(4) = 4$$

and

$$x_2^* = \sum_{k=0}^4 \lambda_{2k} x_{2k} = (1)(0) + (0)(.5) + (0)(1) + (0)(1.5) + (0)(2) = 0$$

And the evaluation of the objective function yields

$$x_1^2 - x_1 + x_2 = 12$$

Another case of nonlinear separable objective is treated as follows.

#### Example 2.6 Nonlinear Objective with Linear Constraints

Solve the following problem by separable programming

$$\text{Maximize } x_1 - x_2 - x_1^2$$

$$\text{Subject to } x_1 + x_2 \geq 4$$

$$x_1, x_2 \geq 0$$

If our answer is  $x_1 = 1$ ,  $x_2 = 3$ ,  $f(x_1, x_2) = -3$ , how do we account for the fact that we obtained an exact answer by an approximation method?

Solution.

Step I

$$f(x_1, x_2) = f_1(x_1) + f_2(x_2) \text{ where}$$

$$\text{where } f_1(x_1) = x_1 - x_1^2 \quad ; \quad f_2(x_2) = -x_2$$

$$g_1 = g_{11}(x_1) + g_{12}(x_2)$$

where  $g_{11}(x_1) = x_1$ ,  $g_{12}(x_2) = x_2^2$

Step II

Determination of domains of the variables

$$x_1 - x_1^2 = 0 \quad \rightarrow \quad x_1(1 - x_1) = 0$$

$\rightarrow x_1 = 0$  or  $1$

if  $x_1 = 1$ , then  $x_2 = 3$  by the constraint.

Step III

k	$x_{1k}$	$x_{2k}$	$g_{11}(x_{1k})$	$g_{12}(x_{2k})$	$f_1(x_{2k})$	$f_2(x_{2k})$
0	0	0	0	0	0	0
1	.25	1	.25	1	3/16	-1
2	.5	2	.5	2	1/4	-2
3	1	3	1	3	0	-3

Complete evaluation table (Figure 2.3.19)

Step IV

Generation of LPP for the piecewise approximation

Maximize  $f$  = Minimize  $-f$  =

$$0\lambda_{10} - 3/16\lambda_{11} - 1/4\lambda_{12} + (0)\lambda_{13} + 0\lambda_{20} + \lambda_{21} + 2\lambda_{22} + 3\lambda_{23}$$

Subject to

$$g_1 = 0\lambda_{10} + 1/4\lambda_{11} + 1/2\lambda_{12} + \lambda_{13} + 0\lambda_{20} + \lambda_{21} + 2\lambda_{22} + 3\lambda_{23} \geq 4$$

$$\lambda_{10} + \lambda_{11} + \lambda_{12} + \lambda_{13} = 1$$

$$\lambda_{20} + \lambda_{21} + \lambda_{22} + \lambda_{23} = 1$$

This can be rewritten as follows

$$-f = 0\lambda_{10} - 3/16\lambda_{11} - 1/4\lambda_{12} + (0)\lambda_{13} + 0\lambda_{20} + \lambda_{21} + 2\lambda_{22} + 3\lambda_{23}$$

$$g_1 - s_1 + s_2 = 0\lambda_{10} + 1/4\lambda_{11} + 1/2\lambda_{12} + \lambda_{13} + 0\lambda_{20} + \lambda_{21} + 2\lambda_{22} + 3\lambda_{23} - s_1 + s_2 = 4$$

$$\lambda_{10} + \lambda_{11} + \lambda_{12} + \lambda_{13} = 1$$

$$\lambda_{20} + \lambda_{21} + \lambda_{22} + \lambda_{23} = 1$$

where  $s_1$  and  $s_2$  are surplus and artificial variables respectively

Step V

To solve for  $\lambda_{jk}$ , we use the Simplex method and adjacency conditions are satisfied.

Figure 2.3.20

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$S_1$	$S_2$	b
$S_2$	0	.25	.5	1	0	1	2	3	-1	1	4
$\lambda_{10}$	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	1	1	1	1	0	0	1
	0	-3/16	-1/4	0	0	1	2	3	0	M	0

$\lambda_{12}$  would enter the basis and  $\lambda_{10}$  would leave, -1/4 is the most negative and this guess the pivot because we are minimizing.

Figure 2.3.21

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$S_1$	$S_2$	B
$S_2$	-0.5	-.25	0	1/2	0	1	2	3	-1	1	7/2
$\lambda_{12}$	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	1	1	1	1	0	0	1
	1/4	1/16	0	1/4	0	1	2	3	0	M	1/4

The solution is  $\lambda_{12} = 1$ ,  $\lambda_{20} = 1$  &  $S_2 = 7/2$

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$S_1$	$S_2$	b
$S_2$	-0.5	-.25	0	1/2	-3	-2	-1	0	-1	1	1/2
$\lambda_{12}$	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{23}$	0	0	0	0	1	1	1	1	0	0	1
	1/4	1/16	0	1/4	-3	-2	-1	0	0	M	-11/4

( Figure 2.3.22)

$S_2 = 1/2$ ,  $\lambda_{12} = 1/4 = 1$ ,  $\lambda_{23} = 1$

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$S_1$	$S_2$	b
$\lambda_{13}$	-1	-0.5	0	1	-6	-4	-2	0	-2	2	1
$\lambda_{12}$	2	3/2	1	0	6	4	2	0	2	-2	0
$\lambda_{23}$	0	0	0	0	1	1	1	1	0	0	1
	1/8	-1/16	0	0	-3/2	-1	-1/2	0	1/2	-1/2	-3

Figure 2.3.23

The solution is  $\lambda_{13} = 1$ ,  $\lambda_{12} = 0$ ,  $\lambda_{23} = 1$

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$S_1$	$S_2$	b
$\lambda_{13}$	1	1	1	1	0	0	0	0	0	0	1
$S_1$	1	3/4	1/2	0	3	2	1	0	1	-1	0
$\lambda_{23}$	0	0	0	0	1	1	1	1	0	0	1
	-3/8	-7/16	-1/4	0	-3	-2	-1/4	0	0	0	-3

Figure 2.3.24

The solution has reached its optimal value. The solution to the table as  $\lambda_{13} = 1$ ,  $\lambda_{23} = 1$  and slack variable,  $S_1 = 0$ .

Step VI

Solve for the original problem

$$x_1^* = 0 + 0 + 0 + (0)(1) = 1$$

$$x_2^* = 0 + 0 + 0 - (3)(1) = -3, \text{ so the original objective yields } -3.$$

As a further illustration, consider example 2.7

Example 2.7

$$\text{Maximize } [(6x_1 - 3x_1^2) + (4x_2 - 2x_2^2) + (2x_3 - 1/3(x_3^2))] \dots\dots\dots (1)$$

$$\text{Subject to } x_1 + 2x_2 + x_3 \leq 4 \text{ all } x_i \geq 0. \dots\dots\dots (2)$$

The terms in  $f(x)$  have been grouped in (1) to display the function's separable character, each component function in parenthesis is concave. Let the grids for  $x_1$  and  $x_2$  be (0,.4,.7,1) and for  $x_3$  be (0,1,1.5,2,3); let

To get the domain of  $x_1, x_2$  and  $x_3$ , we find the 1<sup>st</sup> derivatives of them.

Figure 2.3.25

k	x <sub>1k</sub>	x <sub>2k</sub>	x <sub>3k</sub>	g <sub>11</sub>	g <sub>12</sub>	g <sub>13</sub>	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>
0	0	0	0	0	0	0	0	0	0
1	.4	.4	1	.4	.8	1	1.92	1.28	1.667
2	.7	.7	1.5	.7	1.4	1.5	2.73	1.82	2.25
3	1	1	2	1	2	2	3	2	2.667
4	0	0	3	0	0	3	0	0	3

$$x_1 = (0)\lambda_{10} + .4\lambda_{11} + .7\lambda_{12} + \lambda_{13} + 0\lambda_{14}$$

$$x_2 = (0)\lambda_{20} + .4\lambda_{21} + .7\lambda_{22} + \lambda_{23} + 0\lambda_{24}$$

$$x_3 = 0\lambda_{30} + \lambda_{31} + 1.5\lambda_{32} + 2\lambda_{33} + 3\lambda_{34}$$

Now the problem can be written as follows

$$\text{Maximize } f = (0)\lambda_{10} + 1.92\lambda_{11} + 2.73\lambda_{12} + 3\lambda_{13} + (0)\lambda_{20} + 1.28\lambda_{21} + 1.82\lambda_{22} + 2\lambda_{23} + 0\lambda_{30} + 1.667\lambda_{31} + 2.25\lambda_{32} + 2.667\lambda_{33} + 3\lambda_{34}$$

Subject to

$$g = (0)\lambda_{10} + .4\lambda_{11} + .7\lambda_{12} + \lambda_{13} + (0)\lambda_{20} + .8\lambda_{21} + 1.4\lambda_{22} + 2\lambda_{23} + 0\lambda_{30} + \lambda_{31} + 1.5\lambda_{32} + 2\lambda_{33} \leq 4$$

By simplex criterion, we add slack variable  $\lambda$  to the  $g$  i.e  $g + \lambda = 16$

$$\text{Also } \sum_{k=0}^3 \lambda_{1k} = 1, \quad \sum_{k=0}^3 \lambda_{2k} = 1, \quad \sum_{k=0}^4 \lambda_{3k} = 1$$

Figure 2.3.26

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{33}$	$\lambda_{34}$	$\lambda$	b
$\lambda$	0	-4	-7	1	0	-8	1.4	2	0	1	1.5	2	3	1	4
$\lambda_{10}$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{30}$	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1
	0	1.92	2.73	3	0	1.28	1.82	2	0	1.6667	2.25	2.6667	3	0	0

By simplex criterion, either  $\lambda_{12}$  or  $\lambda_{34}$ , so that either  $\lambda_{10}$  or  $\lambda_{30}$  will depart but let  $\lambda_{13}$  enters  $\lambda_{10}$  departs as shown below in Initial Tableau.

First Tableau (Fig.2.3.27)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{33}$	$\lambda_{34}$	$\lambda$	b
$\lambda$	-1	-6	-3	0	0	.8	1.4	2	0	1	1.5	2	3	1	3
$\lambda_{13}$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{30}$	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1
	-3	-1.08	-2.7	0	0	1.28	1.82	2	0	1.6667	2.25	2.6667	3	0	-3

The solution to the first Tableau is  $\lambda = 3$ ,  $\lambda_{13} = 1$ ,  $\lambda_{20} = 1$ ,  $\lambda_{30} = 1$  and  $\lambda_{34}$  enters the basis, so that the  $\lambda_{30}$  departs the basis as shown in second Tableau.

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{33}$	$\lambda_{34}$	$\lambda$	b
$\lambda$	-1	-6	-3	0	0	.8	1.4	2	-3	-2	-1.5	-1	0	1	0
$\lambda_{13}$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{34}$	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1
	-3	-1.08	-2.7	0	0	1.28	1.82	2	-3	1.3333	-7.5	-3.3333	0	0	-6

Second Tableau ( 2.3.28)

The solution to the basis nor the  $\lambda = 0$ ,  $\lambda_{13} = 1$ ,  $\lambda_{20} = 1$ ; and  $\lambda_{23}$  cannot enter the basis nor the  $\lambda_{22}$  because both are not adjacent to  $\lambda_{20}$ . Therefore  $\lambda_{21}$  enters the basis, so that the  $\lambda$  will leave the basis as shown in the Third Tableau.

Third Tableau (2.3.29)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{33}$	$\lambda_{34}$	$\lambda$	b
$\lambda_{21}$	10/8	-3/4	-3/8	0	0	1	7/4	5/2	-15/4	-5/2	-15/8	-10/8	0	10/8	0
$\lambda_{13}$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{20}$	5/4	3/4	3/8	0	1	0	-3/4	-3/2	15/4	5/2	15/8	5/4	0	-5/4	1
$\lambda_{34}$	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1
	-1.4	-.12	.21	0	0	0	-0.42	-1.2	1.8	1.8667	1.65	1.2667	0	-1.6	-6

The solution to the Third Tableau is  $\lambda_{13} = 1$ ,  $\lambda_{20} = 1$ ,  $\lambda_{21} = 0$ ,  $\lambda_{34} = 1$ .  $\lambda_{31}$  or  $\lambda_{30}$  or  $\lambda_{32}$  would not enter the basis because adjacent condition did not hold. But  $\lambda_{33}$  will enter the basis and  $\lambda_{20}$  leaves the basis as shown in the Fourth Tableau as shown.

Fourth Tableau(2.3.30)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{33}$	$\lambda_{34}$	$\lambda$	b
$\lambda_{21}$	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{13}$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{23}$	1	3/5	3/10	0	4/5	0	-3/5	-6/5	3	2	3/2	1	0	-1	4/5
$\lambda_{34}$	-1	-3/5	-3/10	0	-4/5	0	3/5	6/5	-2	-1	-1/2	0	1	1	1/5
	-2.67	-.88	-.17	0	-1.07	0	0.34	0.32	-2.0	-.67	-.25	0	0	-.33	-7.01336

The solution to the Fourth Table is  $\lambda_{13} = 1$ ,  $\lambda_{21} = 1$ ,  $\lambda_{33} = 4/5$  and  $\lambda_{34} = 1/5$ .  $\lambda_{34}$  will depart the basis and  $\lambda_{22}$  enters the basis as shown in the fifth Tableau.

Fifth Tableau(2.3.31)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{33}$	$\lambda_{34}$	$\lambda$	b
$\lambda_{21}$	5/3	1	1/2	0	7/3	1	0	-1	10/3	5/3	5/6	0	-5/3	-5/3	2/3
$\lambda_{13}$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{23}$	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1
$\lambda_{22}$	-5/3	-1	-1/2	0	-4/3	0	1	2	-10/3	-5/3	-5/6	0	5/3	5/3	1/3
	-2.1	-.54	0	0	-1.44672	0	0	-0.36	-.8667	-.1	0.333	0	-.5667	-.9	-7.1267

The solution to the fifth Tableau is  $\lambda_{21} = 2/3$ ,  $\lambda_{22} = 1/3$ ,  $\lambda_{13} = 1$  and  $\lambda_{33} = 1$ .  $\lambda_{21}$  departs the basis and  $\lambda_{32}$  enters the basis as shown in the sixth Tableua

Sixth Tableau(2.3.32)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{33}$	$\lambda_{34}$	$\lambda$	b
$\lambda_{32}$	2	6/5	3/5	0	14/5	6/5	0	-6/5	4	2	1	0	-2	-2	4/5
$\lambda_{13}$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{33}$	-2	-6/5	-3/5	0	-14/5	-6/5	0	6/5	-3	-1	0	1	3	2	1/5
$\lambda_{22}$	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1
	-2.2	-5.8	-0.2	0	-1.56	-0.4	0	-0.32	-9.9	-1.66	0	0	-5.0	-8.334	-7.15334

The optimal solution is  $\lambda_{13} = 1$ ,  $\lambda_{22} = 1$ ,  $\lambda_{32} = 4/5 = 0.8$  and  $\lambda_{33} = 1/5 = 0.2$  with objective function value 7.15334. This corresponds to  $x_1^* = 1$ ,  $x_2^* = .7$  and  $x_3^* = 1.6$  with a value 7.1667.

The exact Solution.

$$\text{Maximize } [f_1(x_1) + f_2(x_2) + f_3(x_3) = (6x_1 - 3x_1^2) + (4x_2 - 2x_3^2) + (2x_3 - (1/3)x_3^2)]$$

Subject to  $x_1 + 2x_2 + x_3 \leq 4$  & all  $x_j \geq 0$

Lagrange's function

$$L(x_1, x_2, x_3, \lambda) = 6x_1 - 3x_1^2 + 4x_2 - 2x_2^2 + 2x_3 - (1/3)x_3^2 + \lambda(4 - x_1 - 2x_2 - x_3) \dots\dots\dots (1)$$

$$\frac{\partial L}{\partial x_1} = 6 - 6x_1 - \lambda = 0 \dots\dots\dots (2)$$

$$\frac{\partial L}{\partial x_2} = 4 - 4x_2 - 2\lambda = 0 \dots\dots\dots (3)$$

$$\frac{\partial L}{\partial x_3} = 2 - \frac{2}{3}x_3 - \lambda = 0 \dots\dots\dots (4)$$

$$\frac{\partial L}{\partial \lambda} = 4 - x_1 - 2x_2 - x_3 = 0 \dots\dots\dots (5)$$

from :  $\lambda = 6 - 6x_1 \dots\dots\dots (6)$

Substitute 6 into 3 & 4 :  $12x_1 + 4x_2 = 8 \dots\dots\dots (7)$

$$18x_1 - 2x_3 = 12 \quad \dots\dots\dots (8)$$

from 5  $x_1 + 2x_2 + x_3 = 4$

$$12x_1 - 4x_2 = 8$$

$$18x_1 - 2x_3 = 12$$

Using the Gauss-Jordan Elimination method:

$$\begin{pmatrix} 1 & 2 & 1 & 4 \\ 3 & -1 & 0 & 2 \\ 9 & 0 & -1 & 6 \end{pmatrix} \implies \begin{pmatrix} 1 & 2 & 1 & 4 \\ 3 & -1 & 0 & 2 \\ 10 & 2 & 0 & 10 \end{pmatrix} \implies \begin{pmatrix} 1 & 2 & 1 & 4 \\ 3 & -1 & 0 & 2 \\ 16 & 0 & 0 & 14 \end{pmatrix}$$

$$\implies \begin{pmatrix} 16 & 0 & 0 & 14 \\ 3 & -1 & 0 & 2 \\ 1 & 2 & 1 & 4 \end{pmatrix} \implies \begin{pmatrix} 1 & 0 & 0 & \frac{7}{8} \\ 0 & 1 & 0 & \frac{5}{8} \\ 1 & -2 & -1 & \frac{-25}{8} \end{pmatrix} \implies \begin{pmatrix} 1 & 0 & 0 & \frac{7}{8} \\ 0 & 1 & 0 & \frac{5}{8} \\ 0 & 0 & 1 & \frac{15}{8} \end{pmatrix}$$

$\therefore x_1 = 7/8, x_2 = 5/8$  and  $x_3 = 15/8$

Also  $\lambda = 6 - 6(7/8) = 3/4$

$\therefore$  Exact optimal solution is 7.25.

Further Example On Non-Separable Problem.(Objective function with cross product term)

Maximize  $4x_1^2 + 4x_1x_2 + x_2^2$

Subject to  $3x_1 + x_2 \leq 4$

$$4x_1 + 3x_2 \leq 7$$

$$x_1x_2 \geq 0$$

Step I

Convert the objective function into separable functions as follows.

$4x_1^2 + 4x_1x_2 + x_2^2$  becomes

$$4x_1^2 + 4(x_3^2 - x_4^2) + x_2^2 \text{ where}$$

$$\text{where } x_1 = 1/2(x_3 + x_4)$$

$$x_2 = 1/2(x_3 - x_4)$$

$$\therefore 4x_1x_2 = 4(x_3^2 - x_4^2)$$

would be added as new constants.

The original problem can be written as follows

$$\text{Maximize } 4x_1^2 + 4(x_3^2 - x_4^2) + x_2^2$$

$$\text{Subject to } 3x_1 + x_2 \leq 4$$

$$4x_1 + 3x_2 \leq 7$$

$$2x_1 = x_3 + x_4$$

$$2x_2 = x_3 - x_4$$

$$x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \quad x_4 \geq 0$$

Solve for  $x_j$  using the separable programming technique

Construct the evaluation table for the  $\lambda_{jk}$

k	$x_{1k}$	$x_{2k}$	$x_{3k}$	$x_{4k}$	$g_{11}$	$g_{12}$	$g_{21}$	$g_{22}$	$g_{31}$	$g_{32}$	$g_{41}$	$g_{42}$	$f_1$	$f_2$	$f_3$	$f_4$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	.5	.5	1	0	1.5	.5	2	1.5	1	0	1	0	1	.25	1	0
2	1	1	2	0	3	1	4	3	2	0	2	0	4	1	4	0

This can be written in the form below:

$$\text{Maximize } z = (0)\lambda_{10} + \lambda_{11} + 4\lambda_{12} + .25\lambda_{21} + \lambda_{22} + 0\lambda_{30} + \lambda_{31} + \lambda_{32} + \lambda_{40} + \lambda_{42}$$

$$g_1 = 0\lambda_{10} + 1.5\lambda_{11} + 3\lambda_{12} + 0\lambda_{20} + 0.5\lambda_{21} + \lambda_{22} \leq 4$$

$$g_2 = 0\lambda_{10} + 2\lambda_{11} + 4\lambda_{12} + 0\lambda_{20} + 1.5\lambda_{21} + 3\lambda_{22} \leq 7$$

$$g_3 = 0\lambda_{30} + \lambda_{31} + 2\lambda_{32} + 0\lambda_{40} + 0\lambda_{41} + 0\lambda_{42} = 2$$

$$g_4 = 0\lambda_{30} + \lambda_{31} + 2\lambda_{32} + 0\lambda_{40} + 0\lambda_{41} + 0\lambda_{42} = 2$$

$$\therefore \sum_{k=0}^2 \lambda_{1k} = 1 \quad , \quad \sum_{k=0}^2 \lambda_{2k} = 1 \quad , \quad \sum_{k=0}^2 \lambda_{3k} = 1 \quad , \quad \sum_{k=0}^2 \lambda_{4k} = 1$$

To solve for the  $\lambda_{jk}$ , use the ordinary Simplex method so as to solve for  $x_1, x_2, x_3$  and  $x_4$  in the original problem.

(Initial Tableau)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{40}$	$\lambda_{41}$	$\lambda_{42}$	$S_1$	$S_2$	$S_3$	$S_4$	b
$S_1$	0	1.5	3	0	.5	1	0	0	0	0	0	0	1	0	0	0	4
$S_2$	0	2	4	0	1.5	3	0	0	0	0	0	0	0	1	0	0	7
$S_3$	0	0	0	0	0	0	0	1	2	0	0	0	0	0	1	0	2
$S_4$	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	1	2
$\lambda_{10}$	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{30}$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1
$\lambda_{40}$	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1
	0	1	4	0	0.25	1	0	1	4	0	0	0	0	0	M	M	0

Where  $S_1$  and  $S_2$  represent slack values,  $S_3$  and  $S_4$  represent artificial variables. Either the  $\lambda_{32}$  or

$\lambda_{12}$  would enter the basis. Let  $\lambda_{32}$  enters the basis then  $\lambda_{30}$  would leave the basis.

basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{40}$	$\lambda_{41}$	$\lambda_{42}$	$S_1$	$S_2$	$S_3$	$S_4$	b
$S_1$	0	1.5	3	0	.5	1	0	0	0	0	0	0	1	0	0	0	4
$S_2$	0	2	4	0	1.5	3	0	0	0	0	0	0	0	1	0	0	7
$S_3$	0	0	0	0	0	0	-4	-3	0	0	0	0	0	0	1	0	0
$S_4$	0	0	0	0	0	0	-4	-3	0	0	0	0	0	0	0	1	0
$\lambda_{10}$	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{32}$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1
$\lambda_{40}$	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1
	0	1	4	0	0.25	1	-4	-3	0	0	0	0	0	0	MM		-4

The solution to the tableau is  $\lambda_{10} = \lambda_{20} = \lambda_{32} = \lambda_{40} = 1$ ,  $S_1 = 4$ ,  $S_2 = 7$ ,  $S_3 = S_4 = 0$ .

Here  $\lambda$  using simplex Criterion,  $\lambda_{12}$  would enter the basis,  $\lambda_{10}$  would leave the basis as shown below.

Second Tableau

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{40}$	$\lambda_{41}$	$\lambda_{42}$	$S_1$	$S_2$	$S_3$	$S_4$	b
$S_1$	-3	1.5	0	0	.5	1	0	0	0	0	0	0	1	0	0	0	1
$S_2$	-4	-2	0	0	1.5	3	0	0	0	0	0	0	0	1	0	0	3
$S_3$	0	0	0	0	0	0	-4	-3	0	0	0	0	0	0	1	0	0
$S_4$	0	0	0	0	0	0	-4	-3	0	0	0	0	0	0	0	0	0
$\lambda_{12}$	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{32}$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1
$\lambda_{40}$	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1
	-4	-3	0	0	0.25	1	-4	-3	0	0	0	0	0	0	M	M	-8

The solution to the Tableau is  $\lambda_{12} = \lambda_{20} = \lambda_{32} = \lambda_{40} = 1$ .  $S_1 = 1$ ,  $S_2 = 3$ ,  $S_3 = S_4 = 0$ .

Here by using Simplex method  $\lambda_{22}$  would have entered while  $\lambda_{20}$  depart the basis in the next tableau as follows.

(Third Tableau)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{40}$	$\lambda_{41}$	$\lambda_{42}$	$S_1$	$S_2$	$S_3$	$S_4$	b	
$S_1$	-3	-1.5	0	-1	-0.5	0	0	0	0	0	0	0	1	0	0	0	0	
$S_2$	-4	-2	0	-3	-1.5	0	0	0	0	0	0	0	0	1	0	0	0	
$S_3$	0	0	0	0	0	0	-4	-3	0	0	0	0	0	0	1	0	0	
$S_4$	0	0	0	0	0	0	-4	-3	0	0	0	0	0	0	0	1	0	
$\lambda_{12}$	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{22}$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1
$\lambda_{40}$	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1
	-4	-3	0	-1	-0.75	0	-4	-3	0	0	0	0	0	0	M	M		-9

The solution to the Tableau is  $\lambda_{20} = \lambda_{23} = \lambda_{32} = \lambda_{40} = S_1 = S_2 = S_3 = S_4 = 0$ . Also the optimal value for the objective function is 9.  $x_1, x_2, x_3$  and  $x_4$  can be found using  $\lambda$ s.

$$x_1^* = \sum_{k=0}^2 \lambda_{k1} x_{k1} = 0\lambda_{10} + .5\lambda_{11} + \lambda_{12} = (0)(0) + (.5)(0) + (1)(1) = 1,$$

$$x_2^* = \sum_{k=0}^2 \lambda_{2k} x_{2k} = 0\lambda_{20} + .5\lambda_{21} + \lambda_{22} = (0)(0) + (.5)(0) + (1)(1) = 1,$$

$$x_3^* = \sum_{k=0}^2 \lambda_{3k} x_{3k} = 0\lambda_{30} + \lambda_{31} + 2\lambda_{32} = (0)(0) + (1)(0) + 2(0) = 0,$$

$$\text{and } x_4^* = \sum_{k=0}^2 \lambda_{4k} x_{4k} = 0\lambda_{40} + 0\lambda_{41} + 0\lambda_{42} = (0)(1) + (0)(0) + (0)(0) = 0.$$

$$x_1^* = 1, x_2^* = 1, x_3^* = 2 \text{ and } x_4^* = 0$$

To solve for the original problem let look for  $x_1, x_2, x_3$  and  $x_4$  and substitute for these in the original Problem.

So the original problem  $4x_1^2 + 4x_1x_2 + x_2^2$  yields 9.

## 2.5 Conversion of Non-Separability Programming Problems To Separable Programming Problems.

A function would not be separable, if it involves one of the following:

- (1) Cross product terms
- (2) Exponential function

### 2.5.1 Non – separability due to Cross Product

Suprisingly, many nonlinear expression can be put into separable form by introducing additional variables and constraints.

Suppose the model contains in the objective function or constraints (or both) the product of two expressions:

$$f(x_1, x_2, \dots, x_n) \cdot g(x_1, x_2, \dots, x_n) \quad \dots\dots\dots 2.5.1$$

A simple example is  $x_1 \cdot x_2$ . As the first step in obtaining separability, substitute the single variable  $w$  whenever 2.5.1 appears.

Secondly, introduce two more variables  $y$  and  $z$ , and relate them by adding the pair of new constraints.

$$f(x_1, x_2, \dots, x_n) - (y + z) = 0 \text{ and } g(x_1, x_2, \dots, x_n) - (y - z) = 0 \quad \dots\dots\dots 2.5.2$$

Thirdly, to complete the procedure, impose the separable type constraint relating  $w$  to  $y$  and  $z$ :

$$w - (y^2 - z^2) = 0 \quad \dots\dots\dots 2.5.3$$

This ensures that  $w = f(x_1, x_2, \dots, x_n) \cdot g(x_1, x_2, \dots, x_n)$

The constraints 2.5.2 and 2.5.3 are added to the model if 2.5.1 is simply  $x_1 x_2$  the transformation requires three additional variables and  $x_1 x_2$  would be to take its log, forming

$$\text{Log } w = \log x_1 + \log x_2 \quad \dots\dots\dots 2.5.4$$

where  $w = x_1 x_2$ . Then  $x_1 x_2$  would be replaced by  $w$  and 2.5.4 would be added to the constraint set

This method adds only one variable.

To illustrate the above, convert each problem below into a separable form.

Example 2.3

(1) Maximize  $4x_1 + 2x_2 - x_1 x_2$

Subject to  $x_1 + x_2^2 \geq 5$

$$x_1 x_2 \leq 4$$

$$x_1, x_2 \geq 0$$

(2) maximize  $x_1 x_2 + x_2 x_3 + x_1 x_2 x_3$

Subject to  $x_1^2 + x_1 - 2x_1 x_2 x_3 \leq 0$

$$2x_2^2 + 3x_2 + 6x_2 x_3 \leq 12$$

$$x_3^2 + 4x_1 x_3 \leq 5$$

every  $x_j \geq 0$

Solution

Separable form

$$(1) \text{ Let } x_3 = 1/3 (x_1 + x_2) \text{ and } x_4 = 1/2(x_1 - x_2)$$

$$\text{so } x_1x_2 = x_3^2 - x_4^2$$

So the original problem can now be written as

$$\text{Maximize } 4x_1 + 2x_2 - x_3^2 - x_4^2$$

$$\text{Subject to } -x_1 - x_2^2 \leq -5$$

$$x_3^2 - x_4^2 \leq 4, \quad x_3 = 1/2 (x_1 + x_2) ; \quad x_4 = 1/2(x_1 - x_2)$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 = 1/2 (x_1 + x_2) \geq 0 \text{ and } ; \quad x_4 = 1/2(x_1 - x_2) \geq 0$$

Here, we added two variables  $x_3$  and  $x_4$  and equal number of constraints to the model.

$$(2) \text{ Let } w = x_1x_2, \quad h = x_2x_3 \text{ and } k = x_1x_3 \quad \& \quad y = x_1x_2x_3$$

The original problem becomes:

$$\text{Maximize } w + h + y$$

$$\text{Subject to } x_1^2 + x_1 - 2y \leq 0$$

$$2x_2^2 + 3x_2 + 6h \leq 12$$

$$x_3^2 + 4k \leq 5$$

Take log of  $w = x_1x_2$ ,  $h = x_2x_3$  and  $k = x_1x_3$  &  $y = x_1x_2x_3$  and add them to the constraints set in the above:

$$\text{Log } w = \log x_1 + \log x_2$$

$$\text{Log } h = \log x_2 + \log x_3$$

$$\log k = \log x_1 + \log x_2$$

$$\log y = \log x_1 + \log x_2 + \log x_3$$

$$\text{also } x_1, x_2, x_3 \geq 0$$

$$w, h, k, y > 0$$

The above optimization problem can be stated better as follows:

$$\text{Maximize } w + h + y$$

$$\text{Subject to } x_1^2 + x_1 - 2y \leq 0$$

$$2x_2^2 + 3x_2 + 6h \leq 12$$

$$x_3^2 + 4k \leq 5$$

$$\text{Log } w = \log x_1 + \log x_2$$

$$\text{Log } h = \log x_2 + \log x_3$$

$$\log k = \log x_1 + \log x_2$$

$$\log y = \log x_1 + \log x_2 + \log x_3$$

$$\text{every } x_j \geq 0 ; h > 0, k > 0, w > 0 \text{ and } y > 0$$

Note that we added four new variables and equal number of constraints to the model.

## 2.5.2 Exponential Function.

A second illustration, suppose the model contains the expression of the form:

$$\left( h(x_1, x_2, \dots, x_n) \right)^{f(x_1, x_2, \dots, x_n)}$$

where  $h(x_1, x_2, \dots, x_n) > 0$  for all possible  $x_j$ . Here the function  $e^w$  is substituted for 2.5.6 and the new constraint.

$$w - f(x_1, x_2, \dots, x_n) \cdot \ln h(x_1, x_2, \dots, x_n) = 0 \quad \dots\dots\dots 2.5.6$$

is added to the model. Although 2.5.6 is itself not separable, it can be treated by the approach used to separate the expression in 2.5.5 (where  $g = \ln h$ )

To illustrate the above, convert the maximization problem into a separable form.

Example 2.4

Maximize  $5(2 + x_1x_2 + x_2)(x_3/x_4)^{x_1 - x_2}$

Subject to  $x_1x_2x_3x_4 \leq 16$

$x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 2 \quad x_4 \geq 2.$

Solution.

Separable form of the above is as follows:

Let

$w = x_1x_2, h = x_1 - x_2, k = x_1x_2x_3x_4 \quad \& \quad Z = x_3/x_4$

Also let  $e^p = [Z(x)]^h$

$P - h \ln Z = 0$  where  $q = \ln Z$

$\therefore p - hq = 0 \rightarrow P = hq$

$\log p = \log h + \log q$

Also  $r = ye^p, y = 5(2 + x_1x_2 + x_2)$

Take log of the following and add them to the set of constraints i.e

$w = x_1x_2, k = x_1x_2x_3x_4, Z = x_3/x_4, P = hq, r = ye^p$

$\log w = \log x_1 + \log x_2$

$\log k = \log x_1 + \log x_2 + \log x_3 + \log x_4$

$$\log Z = \log x_3 - \log x_4$$

$$\log P = \log h + \log q$$

$$\log r = p \ln e + \log y$$

$$\text{and also } \log y = \log 5 + \log(2 + w + x_2)$$

The original problem can be rewritten in separable form as follows:

Maximize  $r$

Subject to  $k \leq 16$

$$\log w = \log x_1 + \log x_2$$

$$\log k = \log x_1 + \log x_2 + \log x_3 + \log x_4$$

$$\log Z = \log x_3 - \log x_4$$

$$\log P = \log h + \log q$$

$$\log r = p \ln e + \log y$$

$$\log y = \log 5 + \log(2 + w + x_2)$$

$$x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 2 \quad x_4 \geq 2$$

Also  $h > 0, k > 0, w > 0, q > 0, p > 0, r > 0, y > 0, z > 0, k > 0$

We added eight new variables and 6 new constraints to the model.

## 2.6 Computation of Nonlinear Programming Model.

As we have seen, converting a nonlinear programming model into an approximate with separable functions increases the size of the model in two ways:

Firstly, the separability transformation introduces new constraints and variables.

Secondly the subsequent linearization expands the number of constraints and variables even further.

If the original problem contains only a few nonlinearities, this approach is quite workable; otherwise the algorithm may not work.

If the objective and constraint functions utilize the property of adjacent weights,  $\lambda_s$ , then we can employ the ordinary simplex algorithm, which is an available tool. The modified method that forces this property is not widely available.

## 2.7 Remarks

In applying the method to an actual problem, we must of course, consider the selection of the grid for each approximation. Broadly stated, the goal is to select the grids/segments that yield a nearly optimal solution for the original problem.

## Summary

This chapter treated the following with examples, transformation of nonlinear programming problem to separate programming problem, optimization Algorithm for separable objective and conversion of Nonseparability programming problem to separable programming problems.

### 3.0 OPTIMIZATION MODELS WITH RATIONAL OBJECTIVE

#### 3.1 INTRODUCTION

Ratio of Linear Forms. Suppose the objective function to be maximized is

$$C(x) = \frac{c_0 + \sum_{j=1}^n c_j x_j}{f_0 + \sum_{j=1}^n f_j x_j} \dots\dots\dots 3.1.1$$

Sometimes the model is referred to as fractional programming and occasionally as hyperbolic programming. To avoid having to consider a host of possibilities in the exposition assume that the  $x_j$  are so constrained that the denominator in 3.1.1 is strictly positive for all feasible values of  $x_j$ , and that the maximum of  $C(x)$  is finite.

##### 3.1.1 Definition. Fractional Programming Problem

When the nonlinear (fractional) objective can be expressed as a linear function, it is then said to be a Rational Programming Problem.

#### 3.2 CONVERSION OF FRACTIONAL PROGRAMMING INTO A LINEAR MODEL.

##### 3.2.1 CONSTRAINED / UNCONSTRAINED PROBLEM

Suppose the objective function to be maximized is 3.1.1 above.

To convert the problem to a linear model, define the variable  $r$  as

$$r = \left( f_0 + \sum_{j=1}^n f_j x_j \right)^{-1} \dots\dots\dots 3.1.2$$

And hence 3.1.1 can be written as

$$C_0 r + \sum_{j=1}^n C_j x_j r \dots\dots\dots 3.1.3$$

By assumption,  $r > 0$  for all feasible values of  $x_j$ . Next make a change of variables.

$$y_j = rx_j \quad \dots\dots\dots 3.1.4$$

Then the transformed model becomes

$$\text{Maximize } c_0 r + \sum_{j=1}^n c_j y_j \quad \dots\dots\dots 3.1.5$$

where, according to 3.1.2 the  $r$  and  $y_j$  must satisfy the linear constraint

$$rf_0 + \sum_{j=1}^n f_j y_j = 1 \quad \dots\dots\dots 3.1.6$$

and  $r > 0$ . Note that the change of variables 3.1.4 must be performed in any other constraints on  $x_j$

. For example, if any additional conditions are linear

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad \text{for } i = 1, 2, 3, \dots, m \quad \dots\dots\dots 3.1.7$$

The transformed constraints are found by multiplying each relation in 3.1.7 by  $r$ , yielding

$$\sum_{j=1}^n (a_{ij} y_j) - b_i r = 0 \quad \text{for } i = 1, 2, 3, \dots, m \quad \dots\dots\dots 3.1.8$$

To illustrate the above, convert the problem below into an equivalent linear model.

Example 3.1

$$\text{Maximize } \frac{-3 + 2x_1 + 4x_2 - 5x_3}{6 + 3x_1 - x_2}$$

Subject to :

$$x_1 - x_2 \geq 0$$

$$7x_1 + 9x_2 + 10x_3 \leq 30$$

$$x_1 \geq 0, \quad x_2 \geq 1 \quad x_3 \geq 0$$

Solution

The equivalent linear model is as follows:

$$\text{Let } r = 1/(6 + 3x_1 - x_2)$$

Then the objective becomes

$$\text{Maximize } -3r + 2rx_1 + 4rx_2 - 5rx_3$$

$$\text{Subject to } rx_1 - rx_2 \geq 0$$

$$7rx_1 + 9rx_2 + 10rx_3 \leq 30r$$

$$rx_1 \geq 0, rx_2 \geq 0, rx_3 \geq 0$$

By assumption  $r \geq 0$

Also we change  $x_j$  to  $y_j$  as follows i.e.

$$y_j = rx_j \quad j = 1, 2, 3$$

So, the above optimization problem can be written in linear form as follows:

$$\text{Maximize } -3r + 2y_1 + 4y_2 - 5y_3$$

$$\text{Subject to } y_1 - y_2 \geq 0$$

$$7y_1 + 9y_2 + 10y_3 \leq 30r$$

$$y_1 \geq 0, y_2 \geq r, y_3 \geq 0$$

### 3.3 OPTIMIZATION ALGORITHM FOR RATIONAL OBJECTIVE

Here, fractional programming is utilized to solve an optimization problem. A continuous nonlinear fractional function is considered, approximated by the Simplex algorithm. That is essentially what the following algorithm does.

Step I .

Express the functions ( i.e numerator / denominator and constraints) involved in the problem as separate functions.

Step II

Define  $r$  as the reciprocal of the denominator of the objective such that  $r > 0$ .

### Step III

Multiply the function in the numerator of the objective and the constraint(s) by  $r$  and change  $rx_j$  to  $y_j$ .

### Step IV

Generate an equivalent linear or nonlinear programming problem in the  $y_j$ ;  $j = 1, 2, \dots, m$  with suitable constraints on  $r$  and  $y_j$ .

### Step V

Generate a linear programming problem in the  $\lambda_{jk}$  with suitable constraints on the  $\lambda_{jk}$  (if step IV involves the nonlinear function(s))

### Step VI

Solve for the  $\lambda_{jk}$

### Step VII

Solve for the  $y_j$

### Step VIII

Solve for the original variables

## 3.4 ALTERNATIVE OPTIMIZATION ALGORITHM FOR RATIONAL OBJECTIVE

Here, the fractional model is split into the function in the numerator and the function in the denominator combines with constraints.

### Step I

Express the functions involved in the problem as separate functions.

### Step II

Solve for the function in the numerator / constraints using Simplex method.

### Step III

Solve for the function in the denominator / constraints using Simplex method.

#### Step IV

Apply separable programming algorithm to either II or III (or both) when it consists of nonlinear functions.

#### Step V

Divide the result of step II by that of the Step III.

#### Step VI

Solve for the variables

Note that the values of the variables are suppose to be the same for both the Step II and III.

#### Example 3.2

For illustrative purposes let us solve the problem

$$\text{Maximize } \frac{x_1^2 - x_1 + x_2}{x_1 + 2}$$

Subject to

$$x_1 + x_2^2 \leq 4$$

$$x_1, x_2 \geq 0$$

#### Step I

Express the function as separate functions.

This can easily be done as follows:

$$f_1 = x_1^2 - x_1, f_2 = x_2 \quad \text{Numerator}$$

$$h_1(x_1) = x_1 + 2 \quad \text{Denominator}$$

For constraints

$$g_{11}(x_1) = x_1 \text{ \& } g_{12}(x_2) = x_2^2.$$

Step II Define  $r$  as the reciprocal of the objective function.

From the objective function, let the reciprocal of given denominator of the fractional model be  $r$ .

$$\text{i.e } r = 1/(x_1 + 2)$$

### Step III

Multiply the function in the numerator of the objective and the constraint(s) by  $r$  and change  $rx_j$  to  $y_j$ .

Here the numerator is multiply by  $r^2$  and also the constraints

$$r(x) = r^2 x_1^2 - r^2 x_1 + r^2 x_2$$

$$\text{Also } r^2 x_1 + r^2 x_2^2 \leq 4r^2$$

$$r x_1, r x_2 \geq 0$$

This can be written as

$$\text{Maximize } rc(y) = y_1^2 - ry_1 + ry_2$$

$$\text{Subject to } ry_1 + y_2^2 \leq 4r^2$$

$$y_1, y_2 \geq 0$$

### Step IV

Generate a linear programming problem in the  $\lambda_{jk}$  with suitable constraints on the  $\lambda_{jk}$ .

Since the objective and the constraints involves nonlinear then the tabular format would be utilized as follows;

k	$y_{1k}$	$y_{2k}$	$g_{11}(y_{1k})$	$g_{12}(y_{2k})$	$f_1(y_{1k})$	$f_2(y_{2k})$
0	0	0	0	0	0	0
1	r	.5r	$r^2$	$.25 r^2$	0	$.5 r^2$
2	2r	r	$2 r^2$	$r^2$	$2 r^2$	$r^2$
3	3r	1.5r	$3 r^2$	$2.25 r^2$	$6 r^2$	$1.5 r^2$
4	4r	2r	$4 r^2$	$4 r^2$	$12 r^2$	$2 r^2$

The equivalent linear model is :

$$g_1 = 0\lambda_{10} + r^2\lambda_{11} + 2r^2\lambda_{12} + 3r^2\lambda_{13} + 4r^2\lambda_{14} + 0\lambda_{20} + .25r^2\lambda_{21} + r^2\lambda_{22} + 4r^2\lambda_{24} + S = 4$$

$$f^* = 0\lambda_{10} + 0\lambda_{11} + 2r^2\lambda_{12} + 6r^2\lambda_{13} + 12r^2\lambda_{14} + 0\lambda_{20} + .5r^2\lambda_{21} + r^2\lambda_{22} + 1.5r^2\lambda_{23} + 2r^2\lambda_{24}$$

$$\sum_{k=0}^4 \lambda_{1-k} = \lambda_{10} + \lambda_{11} + \lambda_{12} + \lambda_{13} + \lambda_{14} = 1$$

$$\sum_{k=0}^4 \lambda_{2k} = \lambda_{20} + \lambda_{21} + \lambda_{22} + \lambda_{23} + \lambda_{24} = 1$$

also,  $k=0$

Step VI Solve for the  $\lambda_{jk}$

We utilized the simplex algorithm here to solve for the  $\lambda$ s just like we've done with the example in the chapter contains separable model.

Fig 3.3.1 (Initial tableau)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	$S_1$	b
S	0	$r^2$	$2r^2$	$3r^2$	$4r^2$	0	$.25r^2$	$r^2$	$2.25r^2$	$4r^2$	0	4
$\lambda_{10}$	1	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	0	1	1	1	1	1	0	1
	0	0	$2r^2$	$6r^2$	$12r^2$	0	$.5r^2$	$r^2$	$1.5r^2$	$2r^2$	0	0

Since  $r > 0$ ,  $12r^2$  is the most positive entry in the last row and should therefore lead us to the pivot, so  $\lambda_{14}$  enters and  $\lambda_{10}$  departs [ since  $4/4r^2$  is greater than the ratio  $1/1$  in the second row]

Fig 3.3.2 (First Tableau)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	$S_1$	b
S	$-4r^2$	$-3r^2$	$-2r^2$	$-r^2$	0	0	$.25r^2$	$r^2$	$2.25r^2$	$4r^2$	1	$4 - 4r^2$
$\lambda_{14}$	1	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	0	1	1	1	1	1	0	1
	$-12r^2$	$-12r^2$	$-10r^2$	$-6r^2$	0	0	$.5r^2$	$r^2$	$1.5r^2$	$2r^2$	0	$-12r^2$

Since the value of  $r$  has been determined using Lagrangian method to be  $1/6$ , the objective function is  $12 r^2/r$  i.e.  $12r$ .

The value of the objective function is therefore  $12(1/6) = 2$ .

From the last tableau above  $S = 4 - 4r^2$ ,  $\lambda_{14} = 1$  and  $\lambda_{20} = 1$ .

Step V and VI:

$$\therefore y_1 = 0\lambda_{10} + r\lambda_{11} + 2r\lambda_{12} + 3r\lambda_{13} + 4r\lambda_{14}$$

$$= 0 + 0 + 0 + 0 + 4r(1) = 4r$$

$$y_2 = 0\lambda_{20} + .5r\lambda_{21} + r\lambda_{22} + 1.5r\lambda_{23} + 2r\lambda_{24}$$

$$= 0(1) + 0 + 0 + 0 + 0 = 0$$

By Step III,  $y_1 = rx_1$  then  $4r = rx_1$  then  $x_1 = 4$

$$y_2 = rx_2 \quad \text{then} \quad 0 = rx_2 \quad \text{then} \quad x_2 = 0$$

$\therefore$  the original problem becomes  $(4^2 - 4 + 0)/(4 + 2) = 12/6 = 2$

$$\text{Also } r = 1/(x_1 + 2) = 1/(4 + 2) = 1/6$$

### Example 3.3

As an illustration, consider last example for Alternative Algorithm.

Step I

$$\text{Maximize } x_1^2 - x_1 + x_2$$

Numerator/Constraints

$$\text{Subject to } x_1 + x_2^2 \leq 4$$

$$x_1, x_2 \geq 0$$

$$f(x_1, x_2) = f_1(x_1) + f_2(x_2)$$

$$= x_1^2 - x_1 + x_2$$

$$g_1(x_1, x_2) = g_{11}(x_1) + g_{12}(x_2)$$

$$= x_1 + x_2^2 \leq 4$$

Step II

k	$x_{1k}$	$x_{2k}$	$g_{11}(x_{1k})$	$g_{12}(x_{2k})$	$f_1(x_{2k})$	$f_2(x_{2k})$
0	0	0	0	0	0	0
1	1	.5	1	.25	0	.5
2	2	1	2	1	2	1
3	3	1.5	3	2.25	6	1.5
4	4	2	4	4	12	2

Figure 3.4.1

$$\therefore \Gamma = 0\lambda_{10} + \lambda_{11}(0) + 2\lambda_{12} + 6\lambda_{13} + 12\lambda_{14} + 0\lambda_{20} + \lambda_{21}(0.5) + \lambda_{22} + 1.5\lambda_{23} + 2\lambda_{24}$$

$$g^* = 0\lambda_{10} + \lambda_{11} + 4\lambda_{12} + 3\lambda_{13} + 4\lambda_{14} + 0\lambda_{20} + \lambda_{21}(.25) + \lambda_{22} + 2.25\lambda_{23} + 4\lambda_{24} \leq 4$$

$$\sum_{k=0}^4 \lambda_{1k} = \lambda_{10} + \lambda_{11} + \lambda_{12} + \lambda_{13} + \lambda_{14} = 1$$

$$\sum_{k=0}^4 \lambda_{2k} = \lambda_{20} + \lambda_{21} + \lambda_{22} + \lambda_{23} + \lambda_{24} = 1$$

$$\lambda_{jk} \geq 0, \quad j = 1, 2; \quad k = 0, 1, 2, 3, 4$$

Fig 3.4.2 (first tableau)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	S	b
S	0	1	2	3	4	0	.25	1	2.25	4	1	4
$\lambda_{10}$	1	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	0	1	1	1	1	1	0	1
	0	0	2	6	12	0	.5	1	1.5	2	0	0

$\lambda_{14}$  will enter and  $\lambda_{10}$  will depart, the slack variable will not depart since the  $\lambda_{14}$  and  $\lambda_{10}$  are not two adjacent  $\lambda$ s.

$$\lambda_{10} = 1, \lambda_{20} = 1 \text{ \& } S = 4.$$

Fig 3.4.3 (last tableau)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	S	b
S	-4	-3	-2	-1	0	0	.25	1	2.25	4	1	0
$\lambda_{14}$	1	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	0	1	1	1	1	1	0	1
	-2	-12	-10	-6	0	0	.5	1	1.5	2	0	-12

If  $\lambda_{24}$  replaces  $\lambda_{20}$ ; it will definitely make the  $S = b_1 = -4$  which violates the constraints since no  $b_i$  may ever be negative. Also  $\lambda_{21}, \lambda_{22}, \lambda_{23}$  will also makes the  $b_i$ .

$$\therefore S = 0, \lambda_{14} = 1, \lambda_{20} = 1, f^* = 12$$

Step V

$$x_1^* = 0 + 0 + 0 + 0 + 4(1) = 4$$

$$x_2^* = 0 + 0 + 0 + 0 + 0 = 0$$

Evaluation of the objective function yields

$$x_1^2 - x_2 + x_2 = 12$$

Repeating Step I for the denominator/Constraint

Maximize  $x_1 + 2$

Subject to  $x_1 + x_2^2 \leq 4$

$$x_1, x_2 \geq 0$$

$$h(x_1, x_2) = h_1(x_1) + h_2(x_2)$$

$$= x_1 + 2 + 0$$

$$g_1(x_1, x_2) = g_{11}(x_1) + g_{12}(x_2)$$

$$= x_1 + x_2^2 \leq 4$$

Step III

k	$x_{1k}$	$x_{2k}$	$g_{11}(x_{1k})$	$g_{12}(x_{2k})$	$f_1(x_{2k})$
0	0	0	0	0	2
1	1	.5	1	.25	3
2	2	1	2	1	4
3	3	1.5	3	2.25	5
4	4	2	4	4	6

Figure 3.4.4

Approximate functions.

$$f^* = 2\lambda_{10} + 3\lambda_{11} + 4\lambda_{12} + 5\lambda_{13} + 6\lambda_{14}$$

$$g_1 = 0\lambda_{10} + \lambda_{11} + 2\lambda_{12} + 3\lambda_{13} + 4\lambda_{14} + 0\lambda_{20} + .25\lambda_{21} + \lambda_{22} + 2.25\lambda_{23} + 4\lambda_{24} + S = 4$$

$$\sum_{k=0}^4 \lambda_{j,k} = 1 \quad j = 1, 2$$

$$\lambda_{jk} \geq 0, \quad j = 1, 2; \quad k = 0, 1, 2, 3, 4$$

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	S	b
S	0	1	2	3	4	0	.25	1	2.25	4	1	4
$\lambda_{10}$	1	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	0	1	1	1	1	1	0	1
	2	3	4	5	6	0	0	0	0	0	0	0

Figure 3.4.5

$\lambda_{14}$  enters while the  $\lambda_{10}$  departs, since  $\lambda_{10}$  &  $\lambda_{14}$  are not two adjacent  $\lambda$ s.

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{13}$	$\lambda_{14}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{23}$	$\lambda_{24}$	S	b
S	-4	-3	-2	-1	0	0	.25	1	2.25	4	1	0
$\lambda_{14}$	1	1	1	1	1	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	0	0	1	1	1	1	1	0	1
	-4	-3	-2	-1	0	0	0	0	0	0	0	-6

Figure 3.4.6

The procedure as reached its terminal stage. no more positive entries in the last row.

$\therefore \lambda_{14} = 1, \lambda_{20} = 1$  &  $S = 0$  just like that of the numerator (or formal operation)

Step V

$$x_1^* = 4(1) = 4$$

$$x_2^* = 0(1) = 0$$

Objective function yields  $x_1 + 2 = 4 + 2 = 6$

Now evaluation of the original problem yields

$$(x_1^2 - x_1 + x_2)/(x_1 + 2) = 12/6 = 2$$

Note that this method yields the exact value of the problem.

### 3.5 DETERMINATION OF THE EXACT OPTIMAL VALUES

Application of Lagrangian multipliers to the fractional programming problem yields the following

As an illustration, consider example 3.4

Example 3.4

$$L = y_1^2 - ry_1 + ry_2 + \lambda(4r^2 - ry_1 - y_2^2)$$

$$\frac{\partial L}{\partial y_1} = 2y_1 - r - r\lambda = 0 \quad \text{i.e. } y_1 = (r + r\lambda)/2 \quad \dots\dots\dots 3.5.1$$

$$\frac{\partial L}{\partial y_2} = r - (2y_2)\lambda = 0 \quad \text{i.e. } y_2 = r/2\lambda \quad \dots\dots\dots 3.5.2$$

$$\frac{\partial L}{\partial \lambda} = 4r^2 - ry_1 - y_2^2 = 0 \quad \text{i.e. } ry_1 + y_2^2 = 4r^2 \quad \dots\dots\dots 3.5.3$$

Substitute (3.5.1) & (3.5.2) in (3.5.3):

$$r(r + r\lambda)/2 + (r/2\lambda)^2 = 4r^2$$

$$2\lambda^3 - 14\lambda^2 + 1 = 0 \quad \dots\dots\dots 3.5.4$$

By Newton – Raphson method

$$y_1 = 7.9898r/2 = 3.9949r$$

$$y_2 = r/13.9796$$

Now, we determines  $y_1$ ,  $y_2$  &  $r$

$$\text{Since } y_1 = rx_1 \text{ i.e. } 3.9949r = rx_1$$

$$\therefore x_1 = 3.9949$$

$$\text{Also, } y_2 = rx_2 \text{ i.e. } 0/0715r = rx_2$$

$$x_2 = .0715$$

The objective function yields  $(3.9949^2 - 3.9949 + 0.0715)/(3.9949 + 2) = 2.007678$

Error = Exact value – approximate value

$$= 2.0000 - 2.007678 = -0.007678$$

$$x_1 \text{ (Error)} = 4 - 3.9949 = .0051$$

$$x_2 \text{ (Error)} = 0 - 0.0715 = - 0.00715$$

To determine the exact optimal values using Lagrangian method. The numerator / constraint(s) and denominator / constraint(s).

Consider example 3.5 for illustration.

Example 3.5

$$\text{Maximize } x_1^2 - x_1 + x_2$$

$$\text{Subject to } x_1 + x_2^2 \leq 4$$

$$x_1, x_2 \geq 0$$

$$L = x_1^2 - x_1 + x_2 + \lambda(4 - x_1 - x_2^2)$$

$$\frac{\partial L}{\partial x_1} = 2x_1 - x_1 - \lambda = 0 \quad \dots\dots\dots 3.5.5$$

$$\frac{\partial L}{\partial x_2} = 1 - (2x_2)\lambda = 0 \quad \dots\dots\dots 3.5.6$$

$$\frac{\partial L}{\partial \lambda} = x_1 + x_2^2 = 4 \quad \dots\dots\dots 3.5.7$$

$$\text{From (3.5.5); } x_1 = (\lambda + 1)/2 \quad \dots\dots\dots 3.5.8$$

$$\text{From (3.5.6) } x_2 = 1/2\lambda \quad \dots\dots\dots 3.5.9$$

Substitute (3.5.8) & (3.5.9) in (3.5.7);

$$(\lambda + 1)/2 + 1/4\lambda^2 = 4 \text{ i.e}$$

$$2\lambda^3 - 14\lambda^2 + 1 = 0$$

By Newton – Raphson method

$$f(\lambda) = 2\lambda^3 - 14\lambda^2 + 1 = 0$$

$$df(\lambda)/d\lambda = 6\lambda^2 - 28\lambda$$

$$\therefore \lambda_n = \lambda_{n-1} - f(\lambda_{n-1})/f'(\lambda_{n-1}) \quad \text{where } \lambda_0 = 7$$

$$\lambda_1 = 7 - [2(7)^3 - 14(7)^2 + 1] / [6(7)^2 - 28(7)]$$

$$= 6.989795918$$

$$x_1 = [6.989795918 + 1] / 2$$

$$= 3.9949$$

$$x_2 = 1 / [2(6.989795918)]$$

$$= 0.071532846 = 0.0715$$

: Objective function yields  $x_1^2 - x_1 + x_2 = 12.03584459$  with error of  $-0.03584459$ .

Also for the denominator it goes like this

Maximize  $x_1 + x_2$

Subject to  $x_1 + x_2^2 \leq 4$

$$x_1, x_2 \geq 0$$

$$L = x_1 + 2 + \lambda(4 - x_1 - x_2^2) \quad \dots\dots\dots 3.5.10$$

$$\frac{\partial L}{\partial x_1} = 1 - \lambda = 0 \rightarrow \lambda = 1 \quad \dots\dots\dots 3.5.11$$

$$\frac{\partial L}{\partial x_2} = 0 \quad \dots\dots\dots 3.5.12$$

$$\frac{\partial L}{\partial \lambda} = x_1 + x_2^2 = 4 \quad \dots\dots\dots 3.5.13$$

:. Substitute for (3.5.11) in (3.5.10):

$$L = x_1 + 2 + 1(4 - x_1 - x_2^2)$$

Differentiate L now w.r.t.  $x_2$ , we have:

$$\frac{\partial L}{\partial x_2} = -2x_2 = 0 \quad \text{i.e. } x_2 = 0$$

Substitute for  $x_2 = 0$  in (3.5.13), we have :

$$x_1 + 0^2 = 4$$

$$\text{i.e. } x_1 = 4$$

So, objective function yields  $x_1 + 2 = 6$

$\therefore$  The original problem yields

$$\begin{aligned} (x_1^2 - x_1 + 2) / (x_1 + 2) &= 12.03584459 / 6 \\ &= 2.005974098 \end{aligned}$$

Just the error of 0.005974098, which is minimal in nature.

Note that for both numerator (objective function) and denominator (objective function) the values of  $x_1$  and  $x_2$  are constants. But the values of the  $\lambda$  differs.

### 3.6 Nonlinear Objective (Ratio of two linear functions) with linear constraints.

Consider the following illustrative example.

#### Example 3.6

$$\text{Maximize } \frac{3x_1 + x_2 + 2}{x_1 + x_2 + 1}$$

$$\text{Subject to } x_1 + 2x_2 \leq 3$$

$$3x_1 + x_2 \leq 4$$

$$x_1 \geq 1, \quad x_2 \geq 0$$

**Solution.** It is obvious that all the functions involved are separable, step I holds.

#### Step II

$$\text{Let } r = 1/(x_1 + x_2 + 1)$$

where  $r > 0$

Step III

$$\text{Maximize } C(x) = 3r x_1 + r x_2 + 2r$$

$$\text{Subject to } rx_1 + 2r x_2 \leq 3r$$

$$3rx_1 + rx_2 \leq 4r$$

$$r x_1 \geq r, rx_2 \geq 0$$

Step III

This can be rewritten as :

$$\text{Maximize } C(y) = 3y_1 + y_2 + 2r$$

$$\text{Subject to } y_1 + 2y_2 \leq 3r$$

$$3y_1 + y_2 \leq 4r$$

$$y_1 \geq r, y_2 \geq 0$$

where  $y_1 = rx_1$  and  $y_2 = rx_2$

Step VII solve for  $y_j$  using ordinary Simplex method

Fig 3.6.1

Basis	$y_1$	$y_2$	$S_1$	$S_2$	$S_3$	$S_4$	b
$S_1$	1	2	1	0	0	0	$3r$
$S_2$	3	1	0	1	0	0	$4r$
$S_4$	1	0	0	0	-1	1	$r$
	-3	-1	0	0	0	M	$2r$

Let  $S_1$  and  $S_2$  be slack variables

$S_3$  and  $S_4$  are Artificial variables

Figure 3.6.2

Basis	$y_1$	$y_2$	$S_1$	$S_2$	$S_3$	$S_4$	b
$S_1$	0	2	1	0	1	-1	$2r$
$S_2$	0	1	0	1	3	-3	$r$
$y_1$	1	0	0	0	-1	1	$r$
	0	-1	0	0	-3	$M+3$	$5r$

The solution  $y_1 = r, y_2 = 0 = S_3 = S_4, S_1 = 2r, S_2 = r$

This can be improved further by considering most negative entry in the last row:

Figure 3.6.3

Basis	$y_1$	$y_2$	$S_1$	$S_2$	$S_3$	$S_4$	b
$S_1$	0	0	1	-2	-5	5	0
$y_2$	0	1	0	1	3	-3	$r$
$y_1$	1	0	0	0	-1	1	$r$
	0	0	0	1	0	$M$	$6r$

The solution to the table is  $y_1 = r, y_2 = r; S_1 = S_2 = S_3 = S_4 = 0$  while  $C(y) = 6r$

If  $y_1 = rx_1 = r$       i.e  $x_1 = 1$

$y_2 = rx_2 = r$       i.e  $x_2 = 1$

But from the step II

$y_1 + y_2 + r = 1$

$\therefore r + r + r = 1$       i.e  $3r = 1 \implies r = 1/3$

or  $r = 1/(x_1 + x_2 + 1)$  yields  $1/(1 + 1 + 1) = 1/3$

So  $C(y) = 6r = 6(1/3) = 2$

Also, the original problem yields  $(3x_1 + x_2 + 2)/(x_1 + x_2 + 1)$

$= (3 + 1 + 2)/(1 + 1 + 1) = 6/3 = 2$

Now, using the Alternative Algorithm, we have the following:

Example 3.7

Numerator / Constraints

Maximize  $3x_1 + x_2 + 2$

Subject to  $x_1 + 2x_2 \leq 3$

$3x_1 + x_2 \leq 4$

$x_1 \geq 1, \quad x_2 \geq 0$

This can be solved using ordinary simplex method as follows:

Figure 3.6.4

Basis	$x_1$	$x_2$	$S_1$	$S_2$	$S_3$	$S_4$	b
$S_1$	1	2	1	0	0	0	3
$S_2$	3	1	0	1	0	0	4
$S_4$	1	0	0	0	-1	1	1
	-3	-1	0	0	0	M	2

Figure 3.6.5

Basis	$x_1$	$x_2$	$S_1$	$S_2$	$S_3$	$S_4$	b
$S_1$	0	2	1	0	1	-1	2
$S_2$	0	1	0	1	3	-3	1
$S_4$	1	0	0	0	-1	1	1
	0	-1	0	0	-3	M + 3	5

The solution is  $C(x) = 5, x_1 = 1, x_2 = 0, S_1 = 2, S_2 = S_3 = 0$

Basis	$x_1$	$x_2$	$S_1$	$S_2$	$S_3$	$S_4$	b
$S_1$	0	0	1	-2	-5	5	0
$S_2$	0	1	0	1	3	-3	1
$S_4$	1	0	0	0	-1	1	1
	0	0	0	1	0	M	6

Figure 3.6.6

The solution is  $z_1(x) = 6$ ,  $x_1 = 1$ ,  $x_2 = 1$ ,  $S_1 = 0 = S_1 = S_3 = S_4 = 0$ .

So the original problem yields  $3x_1 + x_2 + 2 = 3 + 1 + 2 = 6$

Denominator /Constraints

Maximize  $x_1 + x_2 + 1$

Subject to  $x_1 + 2x_2 \leq 3$

$$3x_1 + x_2 \leq 4$$

$$x_1 \geq 1, x_2 \geq 0$$

This can be solved using ordinary Simplex method just like we did above for the numerator / constraints problem. Now, we adopt the simplex criterion as follows:

Basis	$x_1$	$x_2$	$S_1$	$S_2$	$S_3$	$S_4$	b
$S_1$	1	2	1	0	0	0	3
$S_2$	3	1	0	1	0	0	4
$S_4$	1	0	0	0	-1	1	1
	-1	-1	0	0	0	M	1

Figure 3.6.7

Basis	$x_1$	$x_2$	$S_1$	$S_2$	$S_3$	$S_4$	b
$S_1$	0	2	1	0	1	-1	2
$S_2$	0	1	0	1	3	-3	1
$x_1$	1	0	0	0	-1	1	1
	0	-1	0	0	-1	$M + 1$	2

Figure 3..6.8

Basis	$x_1$	$x_2$	$S_1$	$S_2$	$S_3$	$S_4$	b
$S_1$	0	0	1	-2	-5	5	0
$x_2$	0	1	0	1	3	-3	1
$x_1$	1	0	0	0	-1	1	1
	0	0	0	1	2	$M - 2$	3

Figure 3.6.9

The solution to the last tableau is  $z_2(x) = 3$ ,  $x_1 = 1$ ,  $x_2 = 1$   $S_1 = S_2 = S_3 = S_4 = 0$ .

The original objective yields  $x_1 + x_2 + 1 = 1 + 1 + 1 = 3$

Therefore the original problem  $C(x) = Z_1(x)/Z_2(x)$

$$= (3x_1 + x_2 + 1) / (x_1 + x_2 + 1) = 6/3 = 2.$$

One more example considered. Here the objective function comprises of the quadratic function in the numerator while the denominator of the fractional (objective) function is linear in nature.

### Example 3.8

Maximize  $(4x_1^2 + 4x_1x_2 + x_2^2)/(2x_1 + x_2)$

Subject to  $3x_1 + x_2 \leq 4$

$$4x_1 + 3x_2 \leq 7$$

$$x_1, x_2 \geq 0$$

Step I

Convert the objective (Numerator) function to separable function as follows:

$4x_1^2 + 4x_1x_2 + x_2^2$  becomes

$$4x_1^2 + 4(x_3^2 - x_4^2) + x_2^2$$

where  $x_1 = \frac{1}{2}(x_3 + x_4)$

$$x_2 = \frac{1}{2}(x_3 - x_4)$$

$$\therefore x_1x_2 = (x_3^2 - x_4^2)$$

would be added as new constraints.

The original problem can be written as follows:

$$\text{Maximize } [4x_1^2 + 4(x_3^2 - x_4^2) + x_2^2] / (2x_1 - x_2)$$

$$\text{Subject to } 3x_1 + x_2 \leq 4$$

$$4x_1 + 3x_2 \leq 7$$

$$2x_1 = x_3 + x_4$$

$$2x_2 = x_3 - x_4$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

Step II

$$\text{Let } r = 1/(2x_1 + x_2)$$

where  $r > 0$

Step III

Multiply the numerator and the constraints by  $r$ , as follows

$$rz(x) = 4r^2x_1^2 + 4r^2x_1x_2 + r^2x_2^2 \text{ i.e. } 4r^2x_1^2 + 4r^2(x_3^2 - x_4^2) + r^2x_2^2$$

$$3r x_1 + r x_2 \leq 4r$$

$$4r x_1 + r x_2 \leq 7r$$

$$r x_3 + r x_4 = 2rx_1$$

$$rx_3 - rx_4 = 2rx_2$$

$$r x_1, r x_2, r x_3, r x_4 \geq 0$$

$$\text{where } y_1 = r x_1, y_2 = r x_2, y_3 = r x_3, y_4 = r x_4$$

Then the problem can now be rewritten as follows:

$$4y_1^2 + 4y_3 - 4y_4^2 + y_2^2$$

$$3y_1 + y_2 \leq 4r$$

$$4y_1 + 3y_2 \leq 7r$$

$$y_3 + y_4 = 2y_1$$

$$y_3 - y_4 = 2y_2$$

$$y_1 \geq 0, y_2 \geq 0, y_3 \geq 0, y_4 \geq 0$$

Since  $y_1 \leq r, y_2 \leq r$  solving simultaneously constraints (1) and (2) we have

$$2y_1 = 2r \text{ for } b_3$$

$$2y_2 = 2r \text{ for } b_4$$

Step IV

Solve for  $y_j$  using the separable programming technique we construct the evaluation table for  $\lambda_{jk}$

k	$y_{1k}$	$y_{2k}$	$y_{3k}$	$y_{4k}$	$g_{11}$	$g_{12}$	$g_{21}$	$g_{22}$	$g_{31}$	$g_{32}$	$g_{41}$	$g_{42}$	$f_1$	$f_2$	$f_3$	$f_4$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.5	.5	.5	1	0	1.5	.5	2	1.5	1	0	1	0	1	.25	1	0
1	1	1	2	0	3	1	4	3	2	0	2	0	4	1	4	0

Figure 3.6.9

This can be written in the form below:

$$\text{Maximize } rz = (0)\lambda_{10} + \lambda_{11} + 4\lambda_{12} + .25\lambda_{21} + \lambda_{22} + 0\lambda_{30} + \lambda_{31} + 4\lambda_{32} + 0\lambda_{40} + 0\lambda_{41} + 0\lambda_{42}$$

subject to

$$g_1 = (0)\lambda_{10} + 1.5\lambda_{11} + 3\lambda_{12} + 0\lambda_{20} + 0.5\lambda_{21} + \lambda_{22} \leq 4r$$

$$g_2 = 0\lambda_{10} + 2\lambda_{11} + 4\lambda_{12} + 0\lambda_{20} + 1.5\lambda_{21} + 3\lambda_{22} \leq 7r$$

$$g_3 = 0\lambda_{30} + \lambda_{31} + 2\lambda_{32} + 0\lambda_{40} + 0\lambda_{41} + 0\lambda_{42} = 2r$$

$$g_4 = 0\lambda_{30} + \lambda_{31} + 2\lambda_{32} + 0\lambda_{40} + 0\lambda_{41} + 0\lambda_{42} = 2r$$

$$\sum_{k=0}^2 \lambda_{1k} = 1, \quad \sum_{k=0}^2 \lambda_{2k} = 1, \quad \sum_{k=0}^2 \lambda_{3k} = 1, \quad \sum_{k=0}^2 \lambda_{4k} = 1$$

To solve for the  $\lambda_{jk}$ , use the ordinary Simplex method so as to solve for  $x_1, x_2, x_3$  and  $x_4$  in the original problem.

Fig. 3.6.10 (Initial Tableau)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{40}$	$\lambda_{41}$	$\lambda_{42}$	$S_1$	$S_2$	$S_3$	$S_4$	b(r)
$S_1$	0	1.5	3	0	.5	1	0	0	0	0	0	0	1	0	0	0	4
$S_2$	0	2	4	0	1.5	3	0	0	0	0	0	0	0	1	0	0	7
$S_3$	0	0	0	0	0	0	0	1	2	0	0	0	0	0	1	0	2
$S_4$	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	1	2
$\lambda_{10}$	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{30}$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1
$\lambda_{40}$	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1
	0	1	4	0	0.25	1	0	1	4	0	0	0	0	0	M	M	0

Where  $S_1$  and  $S_2$  represent slack variables,  $S_3$  and  $S_4$  represents artificial variables. Either the  $\lambda_{32}$  or  $\lambda_{12}$  would enter the basis, let  $\lambda_{32}$  enters the basis. Then the  $\lambda_{30}$  would leave the basis.

Figure 3.6.11 (First tableau)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{40}$	$\lambda_{41}$	$\lambda_{42}$	$S_1$	$S_2$	$S_3$	$S_4$	b(r)
$S_1$	0	1.5	3	0	.5	1	0	0	0	0	0	0	1	0	0	0	4
$S_2$	0	2	4	0	1.5	3	0	0	0	0	0	0	0	1	0	0	7
$S_3$	0	0	0	0	0	0	-4	-3	0	0	0	0	0	0	1	0	0
$S_4$	0	0	0	0	0	0	-4	-3	0	0	0	0	0	0	0	1	0
$\lambda_{10}$	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{32}$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1
$\lambda_{40}$	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1
	0	1	4	0	0.25	1	-4	-3	0	0	0	0	0	0	MM		-4r

The solution to the tableau is  $\lambda_{10} = \lambda_{20} = \lambda_{32} = \lambda_{40} = 1, S_1 = 4, S_2 = 7, S_3 = S_4 = 0$ .

Here by using simplex Criterion,  $\lambda_{12}$  would enter the basis,  $\lambda_{10}$  would leave the basis as shown in the next table (figure 3.21)

Second Tableau (3.6.12)

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{40}$	$\lambda_{41}$	$\lambda_{42}$	$S_1$	$S_2$	$S_3$	$S_4$	b(r)
$S_1$	-3	-1.5	0	0	.5	1	0	0	0	0	0	0	1	0	0	0	1
$S_2$	-4	-2	0	0	1.5	3	0	0	0	0	0	0	0	1	0	0	3
$S_3$	0	0	0	0	0	0	-4	-3	0	0	0	0	0	0	1	0	0
$S_4$	0	0	0	0	0	0	-4	-3	0	0	0	0	0	0	0	0	0
$\lambda_{12}$	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{20}$	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{32}$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1
$\lambda_{40}$	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1
	-4	-3	0	0	0.25	1	-4	-3	0	0	0	0	0	0	M	M	-8r

The solution to the Tableau is  $\lambda_{12} = \lambda_{20} = \lambda_{32} = \lambda_{40} = 1$ .  $S_1 = 1$ ,  $S_2 = 3$ ,  $S_3 = S_4 = 0$ .

Here by using Simplex method  $\lambda_{22}$  would have enter while  $\lambda_{20}$  departs the basis in the next tableau as follows.

Figure 3.6.13 (third tableau) .

Basis	$\lambda_{10}$	$\lambda_{11}$	$\lambda_{12}$	$\lambda_{20}$	$\lambda_{21}$	$\lambda_{22}$	$\lambda_{30}$	$\lambda_{31}$	$\lambda_{32}$	$\lambda_{40}$	$\lambda_{41}$	$\lambda_{42}$	$S_1$	$S_2$	$S_3$	$S_4$	$b(r)$	
$S_1$	-3	-1.5	0	-1	-.5	0	0	0	0	0	0	0	1	0	0	0	0	
$S_2$	-4	-2	0	-3	-1.5	0	0	0	0	0	0	0	0	1	0	0	0	
$S_3$	0	0	0	0	0	0	-4	-3	0	0	0	0	0	0	1	0	0	
$S_4$	0	0	0	0	0	0	-4	-3	0	0	0	0	0	0	0	1	0	
$\lambda_{12}$	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{22}$	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{32}$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1
$\lambda_{40}$	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1
	-4	-3	0	-1	-.75	0	-4	-3	0	0	0	0	0	0	M	M		-9r

The solution to the Tableau is  $\lambda_{12} = \lambda_{22} = \lambda_{32} = \lambda_{40} = r$

$S_1 = S_2 = S_3 = S_4 = 0$ . Also the optimal value for the objective function is  $9r^2$ . let find  $y_1, y_2, y_3$  and  $y_4$  using  $\lambda$ s.

Step VII

$$y_1^* = \sum_{k=0}^2 \lambda_{k1} y_{k1} r = 0\lambda_{10} + .5\lambda_{11} + \lambda_{12} = (0)(0) + (.5)(0) + (1)(1)r = r,$$

$$y_2^* = \sum_{k=0}^2 \lambda_{2k} y_{2k} r = 0\lambda_{20} + .5\lambda_{21} + \lambda_{22} = (0)(0) + (.5)(0) + (1)(1)r = r,$$

$$y_3^* = \sum_{k=0}^2 \lambda_{3k} y_{3k} r = 0\lambda_{30} + \lambda_{31} + 2\lambda_{32} = (0)(0) + (1)(0) + 2(1)r = 2r,$$

$$\text{and } y_4^* = \sum_{k=0}^2 \lambda_{4k} y_{4k} r = 0\lambda_{40} + 0\lambda_{41} + 0\lambda_{42} = (0)(1) + (0)(0) + 0(0) = 0$$

$$\therefore y_1^* = r, y_2^* = r, y_3^* = 2r \text{ and } y_4^* = 0$$

Step VIII

To solve for the original problem, let look for  $x_1, x_2, x_3$  and  $x_4$  and substitute for these in the original problem

$$\text{Since } y_j = rx_j \quad j = 1, 2, 3, 4,$$

$$\text{then } y_1 = rx_1 = r \quad \text{i.e. } x_1^* = 1,$$

$$y_2 = rx_2 = r \quad \text{i.e. } x_2^* = 1,$$

$$y_3 = rx_3 = 2r \quad \text{i.e. } x_3^* = 2,$$

$$y_4 = rx_4 = 0 \quad \text{i.e. } x_4^* = 0$$

So the original problem yields

$$(4x_1^2 + 4x_1x_2 + x_2^2) / (2x_1 + x_2) =$$

$$[4(1) + 4(1)(1) + 1^2] / [2(1) + 1] = 9/3 = 3$$

But the optimal solution from the tableau is  $rz = 9r^2$  where  $r = 1/(2x_1 + x_2) = 1/(2(1) + 1) = 1/3$

So the optimal value of the objective function is  $9(1/3) = 3$

Note that the objective function  $z$  was multiplied by  $r$ , so that the fraction at the right hand side also becomes

$$C_0 r^2 + \sum_{j=1}^2 C_j x_j r^2$$

where  $C_0 = 0$ . Therefore we have a situation where

$$rz = \sum_{j=1}^2 r^2 C_j x_j$$

. In the above example  $rz = 9r^2$ , so that  $z = 9r$  where  $r = 1/3$ , then

$$z^* = 9(1/3) = 3.$$

It was observed that the objective function could be factorized to  $2x_1 + x_2$  which can be combined with the given constraints and solved ordinarily by Simplex method to get the same answer as that of the above. Consider the solution to the above problem using ordinary Simplex method;

Example 3.9

Maximize  $z = 2x_1 + x_2$

Subject to  $3x_1 + x_2 \leq 4$

$$4x_1 + 3x_2 \leq 7$$

$$x_1, x_2 \geq 0$$

This can be transformed as follows, using Simplex criterion.

$$3x_1 + x_2 + S_1 = 4$$

$$4x_1 + 3x_2 + S_2 = 7 \text{ where } S_1 \text{ \& } S_2 \text{ are slack variables.}$$

$$z - 2x_1 - x_2 + 0S_1 + 0S_2 = 0$$

Now,

Initial tableau

Basis	$x_1$	$x_2$	$S_1$	$S_2$	b
$S_1$	3	1	1	0	4
$S_2$	4	3	0	1	7
	-2	-1	0	0	0

Figure 3.23

First Tableau

Basis	$x_1$	$x_2$	$S_1$	$S_2$	b
$x_1$	1	$1/3$	$1/3$	0	$4/3$
$S_2$	0	$5/3$	$-4/3$	1	$5/3$
	0	$-1/3$	$2/3$	0	$8/3$

Figure 3.24

The solution to the first tableau is  $x_1 = 4/3$ ,  $S_2 = 5/3$ ,  $S_1 = x_2 = 0$  and  $Z = 8/3$ . Then the  $x_2$  will enter the basis, while the  $S_2$  departs the basis as follows:

Basis	$x_1$	$x_2$	$S_1$	$S_2$	b
$x_1$	1	0	$3/5$	$-1/5$	1
$x_2$	0	1	$-4/5$	$3/5$	1
	0	0	$2/5$	$1/5$	$9/3$

Second tableau.

So, the solution to the last tableau is  $x_1 = x_2 = 1$ ,  $S_1 = S_2 = 0$  and  $Z = 9/3 = 3$  which is the optimal value of the objective function  $2x_1 + x_2$  where  $x_1 = 1$ ;  $x_2 = 1$  are optimal value for variables  $x_1$  and  $x_2$ .

By comparing the example 3.8 and 3.9, we observed that the objective function in each case is  $9/3 = 3$  while  $x_1 = 1$  and  $x_2 = 1$  in the two examples.

Remarks.

We observed that the idea of separable programming works for the rational objective and the idea of combining numerator and denominator with the constraints, so that the results of the latter divide by the results of the formal also works.

## 4.0 COMPUTATIONAL ALGORITHM FOR SEPARABLE OBJECTIVE

### 4.1 PSEUDOCODE

Pseudocode for Separable Objective

Step I : INPUT OBJECTIVE FUNCTION  $f(x)$  AND CONSTRAINTS  $g_i$ ,  $i = 1, 2, \dots, m$ .

Step II: DETERMINE THE DOMAINS OF THE VARIABLES,  $x_i$ ,  $i = 1, 2, \dots, n$

Step III: IF FUNCTIONS ARE SEPARABLE, THEN PARTITION THE DOMAIN OF THE VARIABLES INTO NUMBERS OF SEGMENTS, OTHERWISE EXPRESS THE FUNCTIONS AS SEPARATE FUNCTIONS.

Step IV: EVALUATE SEPARATE FUNCTIONS AT ENDPPOINTS OF EACH SEGMENT.

Step V: GENERATE LINEAR PROGRAMMING PROBLEM IN  $\lambda_{jk}$ ,  $\lambda_{jk} \geq 0$

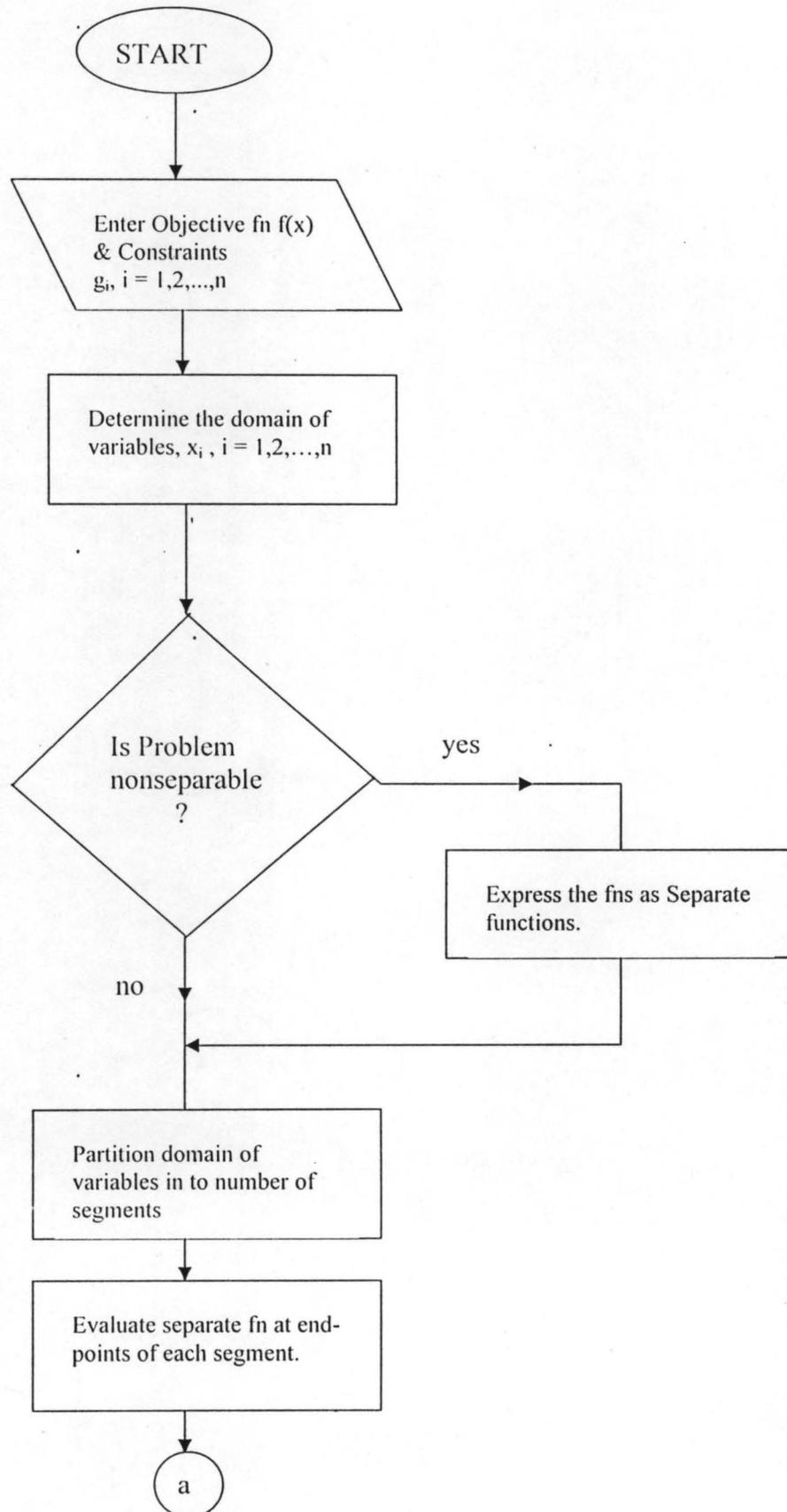
Step VI: SOLVE FOR  $\lambda_{jk}$  USING MODIFIED SIMPLEX METHOD.

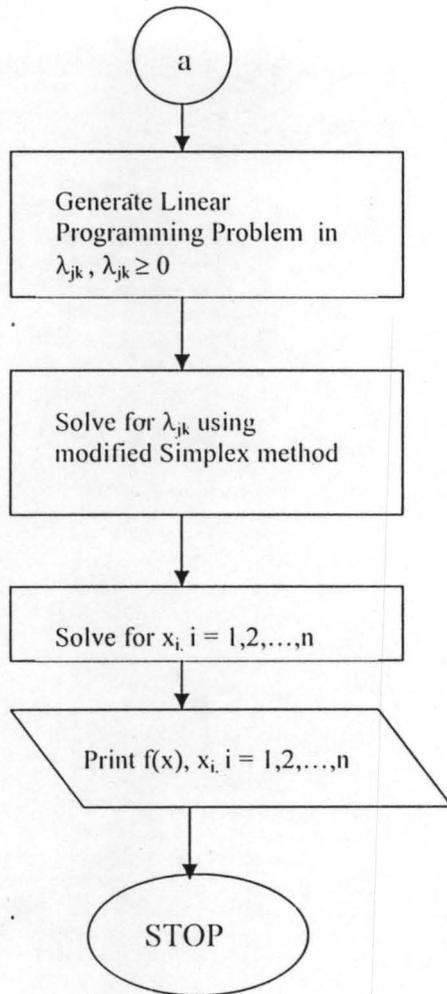
Step VII: SOLVE FOR  $x_i$ ,  $i = 1, 2, \dots, n$ .

Step VIII: PRINT  $f(x)$ ,  $x_i$ ,  $i = 1, 2, \dots, n$ .

## 4.2 FLOWCHART

COMPUTATIONAL ANALYSIS  
Application of Separable Algorithm  
Flow Chart for Separable Objective





### 4.3 OUTPUT OF THE SEPARABLE OBJECTIVE CODE

**Separable Objective - [Step I]**

Objective Function

Constraint

Add  
Clear  
GO

**Separable Objective - [Step II]**

Number of Segments   Range  Values

Enter the domain for each variable

x1	0	3
x2	0	4

Previous  Next

**Separable Objective - [Step III]**

k	x 1k	x 2k	f1(x1k)	f2(x2k)	g11	g12	g21	g22
0	0	0	0	0	0	0	0	0
1	1.5	2	7.59375	2	1.5	2E-08	1.5E-08	2
2	3	4	243	4	3	4E-08	3E-08	4

Previous  Next

Separable Objective - [Form4]

	L10	L11	L12	L20	L21	L22	S1	S2	Right Side
S1	0	1.5	3	0	0	0	1	0	3
S2	0	0	0	0	2	4	0	1	4
L10	1	1	1	0	0	0	0	0	1
L20	0	0	0	1	1	1	0	0	1
	0	7.59375	243	0	2	4	0	0	0

Previous | Iterate

Separable Objective - [Form4]

	L10	L11	L12	L20	L21	L22	S1	S2	Right Side
S1	-3	-1.5	0	0	0	0	1	0	0
S2	0	0	0	-4	-2	0	0	1	0
L12	1	1	1	0	0	0	0	0	1
L22	0	0	0	1	1	1	0	0	1
	-243	-235.4063	0	-4	-2	0	0	0	-247

Previous | Iterate

Separable Objective - [Form5]

Variable	Value
x1	3
x2	4
f	-247

Separable Objective - [Step 1]

Objective Function

Constraint

Add  
Clear  
GO

### Separable Objective - [Step II]

Number of Segments

Range

Enter the domain for each variable

Values

x1	0	2
x2	0	4

Previous

Next

### Separable Objective - [Step III]

k	x 1k	x 2k	f1(x1k)	f2(x2k)	g11	g12
0	0	0	0	0	0	0
1	0.5	1	1.5	2	1	1
2	1	2	3	4	4	4
3	1.5	3	4.5	6	9	9
4	2	4	6	8	16	16

Previous

Next

### Objective - [Form4]

L11	L12	L13	L14	L20	L21	L22	L23	L24	S1	Right Side
1	4	9	16	0	1	4	9	16	1	16
1	1	1	1	0	0	0	0	0	0	1
0	0	0	0	1	1	1	1	1	0	1
1.5	3	4.5	6	0	2	4	6	8	0	0

ous

Iterate

Results

Objective - [Form4]

.11	L12	L13	L14	L20	L21	L22	L23	L24	S1	Right Side
1	4	9	16	-16	-15	-12	-7	0	1	0
1	1	1	1	0	0	0	0	0	0	1
0	0	0	0	1	1	1	1	1	1	0
1.5	3	4.5	6	-8	-6	-4	-2	0	0	-8

ous

Iterate

Results

Objective - [Form4]

.11	L12	L13	L14	L20	L21	L22	L23	L24	S1	Right Side
1	4	9	16	-16	-15	-12	-7	0	1	0
0	-3	-8	-15	16	15	12	7	0	-1	1
0	0	0	0	1	1	1	1	1	0	1
0	-3	-9	-18	16	16.5	14	8.5	0	-1.5	-8

ous

Iterate

Results

Objective - [Form4]

.11	L12	L13	L14	L20	L21	L22	L23	L24	S1	Right Side
1	1	0.9999996	1	152557E-07	768372E-07	192093E-07	0	0	470348E-08	1
0	-0.4285714	-1.142857	-2.142857	2.285714	2.142857	1.714286	1	0	-0.1428571	0.1428571
0	0.4285714	1.142857	2.142857	-1.285714	-1.142857	-0.7142857	0	1	0.1428571	0.8571429
0	0.6428572	0.7142861	0.2142851	-3.428572	-1.714285	-0.5714287	0	0	-0.2857142	-9.214286

ous

Iterate

Results

**Objective - [Form4]**

L11	L12	L13	L14	L20	L21	L22	L23	L24	S1	Right Side
1	1	0.9999996	1	152557E-07	768372E-07	192093E-07	0	0	470348E-08	1
0.4285714	0	-0.7142859	-1.714285	2.285715	2.142857	1.714286	1	0	-0.1428572	0.5714286
-0.4285714	0	0.7142859	1.714285	-1.285715	-1.142857	-0.7142858	0	1	0.1428572	0.4285714
-0.6428572	0	142919E-02	-0.4285724	-3.428573	-1.714285	-0.5714288	0	0	-0.2857142	-9.857143

**Objective - [Form4]**

L11	L12	L13	L14	L20	L21	L22	L23	L24	S1	Right Side
1.6	1	0	-1.399998	1.8	1.599998	0.9999996	0	-1.399999	-0.1999999	0.4000004
544454E-09	0	0	833737E-08	0.9999999	0.9999999	0.9999999	1	1	277248E-09	1
-0.5999998	0	1	2.399999	-1.8	-1.599999	-0.9999998	0	1.4	0.2	0.5999998
-0.5999997	0	0	-0.6000023	-3.3	-1.599998	-0.4999996	0	-0.1000008	-0.3	-9.900001

**Separable Objective - [Form5]**

Variable	Value
x1	1.3
x2	3
f	-9.900001

### Separable Objective - [Step I]

Objective Function

$$6x_1 - 3x_1^2 + 4x_2 - 2x_2^2 + 2x_3 - 0.33333333x_3^2$$

Constraint

$$x_1 + 2x_2 + x_3 - 4$$

Add

$$x_1 + 2x_2 + x_3 - 4$$

Clear

GO

### Separable Objective - [Step II]

Number of Segments

4

Range

Enter the domain for each variable

Values

x1	0	0.4	0.7	1	0
x2	0	0.4	0.7	1	0
x3	0	1	1.5	2	3

Previous

Next

### Separable Objective - [Step III]

k	x1k	x2k	x3k	f1(x1k)	f2(x2k)	f3(x3k)	g11	g12	g13
0	0	0	0	0	0	0	0	0	0
1	0.4	0.4	1	1.92	1.28	1.666667	0.4	0.8	1
2	0.7	0.7	1.5	2.73	1.82	2.25	0.7	1.4	1.5
3	1	1	2	3	2	2.666667	1	2	2
4	0	0	3	0	0	3	0	0	3

Previous

Next

Objective - [Form4] \_ | [ ] | X

.21	L22	L23	L24	L30	L31	L32	L33	L34	S1	Right Side
0.8	1.4	2	0	0	1	1.5	2	3	1	4
0	0	0	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	0	0	0	1
0	0	0	0	1	1	1	1	1	1	1
1.28	1.82	2	0	0	1.666667	2.25	2.666667	3	0	0

Objective - [Form4] \_ | [ ] | X

L21	L22	L23	L24	L30	L31	L32	L33	L34	S1	Right Side
0.8	1.4	2	0	-3	-2	-1.5	-1	0	1	1
0	0	0	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	0	0	0	1
0	0	0	0	1	1	1	1	1	1	1
1.28	1.82	2	0	-3	-1.333333	-0.7500002	-0.3333335	0	0	-3

Objective - [Form4] \_ | [ ] | X

.21	L22	L23	L24	L30	L31	L32	L33	L34	S1	Right Side
0.8	1.4	2	0	-3	-2	-1.5	-1	0	1	0
0	0	0	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	0	0	0	1
0	0	0	0	1	1	1	1	1	1	1
1.28	1.82	2	0	-3	-1.333333	-0.7500002	-0.3333335	0	0	-6

## Objective - [Form4]

.21	L22	L23	L24	L30	L31	L32	L33	L34	S1	Right Side
1	1.75	2.5	0	-3.75	-2.5	-1.875	-1.25	0	1.25	0
0	0	0	0	0	0	0	0	0	0	1
0	-0.75	-1.5	1	3.75	2.5	1.875	1.25	0	-1.25	1
0	0	0	0	1	1	1	1	1	0	1
0	-0.42	-1.2	0	1.8	1.866666	1.65	1.266666	0	-1.6	-6

jus

Iterate

Results

## Objective - [Form4]

.21	L22	L23	L24	L30	L31	L32	L33	L34	S1	Right Side
1	1	0.9999999	1	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	1
0	-0.6	-1.2	0.8	3	2	1.5	1	0	-1	0.8
0	0.6	1.2	-0.8	-2	-1	-0.5	0	1	1	0.2
0	0.3399999	0.3199998	-1.013333	-2	-0.6666664	-0.2499999	0	0	-0.3333335	-7.013333

jus

Iterate

Results

## Objective - [Form4]

.21	L22	L23	L24	L30	L31	L32	L33	L34	S1	Right Side
1	0	-1	2.333333	3.333333	1.666667	0.8333333	0	-1.666667	-1.666667	0.6666667
0	0	0	0	0	0	0	0	0	0	1
0	0	0	2.78155E-08	0.9999999	1	1	1	1	589457E-08	1
0	1	2	-1.333333	-3.333333	-1.666667	-0.8333333	0	1.666667	1.666667	0.3333333
0	0	-0.3599999	-0.5600001	-0.8666667	999998E-02	333332E-02	0	-0.5666664	-0.8999999	-7.126667

jus

Iterate

Results

Objective - [Form4]



.21	L22	L23	L24	L30	L31	L32	L33	L34	S1	Right Side
1.2	0	-1.2	2.8	4	2	1	0	-2	-2	0.8000001
0	0	0	0	0	0	0	0	0	0	1
-1.2	0	1.2	-2.8	-3	-1	0	1	3	2	0.1999999
1	1	1	1	0	0	0	0	0	0	1
3999999E-02	0	-0.3199999	-0.6533334	-1	-0.1666666	0	0	-0.4999998	-0.8333333	-7.153333



ms

Iterate

Results

Separable Objective - [Form5]

Variable	Value
x1	1
x2	0.7
x3	1.6
f	-7.153333

## 4.4 GRAPHICAL SOLUTIONS

In one of the examples above we considered the function

$$f(x_1, x_2) := x_1^5 + x_2$$

over the region

$$x_1 \leq 3 \quad x_2 \leq 4 \quad x_1 \geq 0 \quad x_2 \geq 0$$

We can draw the points under consideration using a parametric surface plot. First define a set of

$x_1$  and  $x_2$  values which range over the critical region

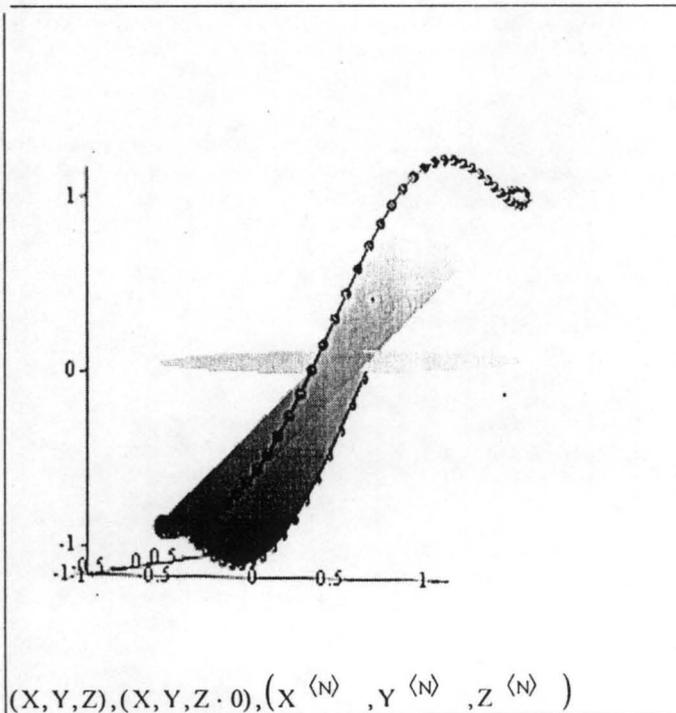
$$N := 100 \quad i := 0..N \quad j := 0..N$$

$$X_{i,j} := \frac{j}{N} \cdot \cos\left(i \cdot \frac{2\pi}{N}\right) \quad Y_{i,j} := \frac{j}{N} \cdot \sin\left(i \cdot \frac{2\pi}{N}\right)$$

and then compute the corresponding  $z$  values

$$Z_{i,j} := f(X_{i,j}, Y_{i,j})$$

The corresponding plot looks like



In one of the examples above we considered the function

$$f(x_1, x_2) := 3 \cdot x_1 + 2 \cdot x_2$$

over the region

$$4 \cdot x_1^2 + x_2^2 \leq 16 \quad x_1 \geq 0 \quad x_2 \geq 0$$

We can draw the points under consideration using a parametric surface plot. First define a set of  $x_1$

and  $x_2$  values which range over the critical region

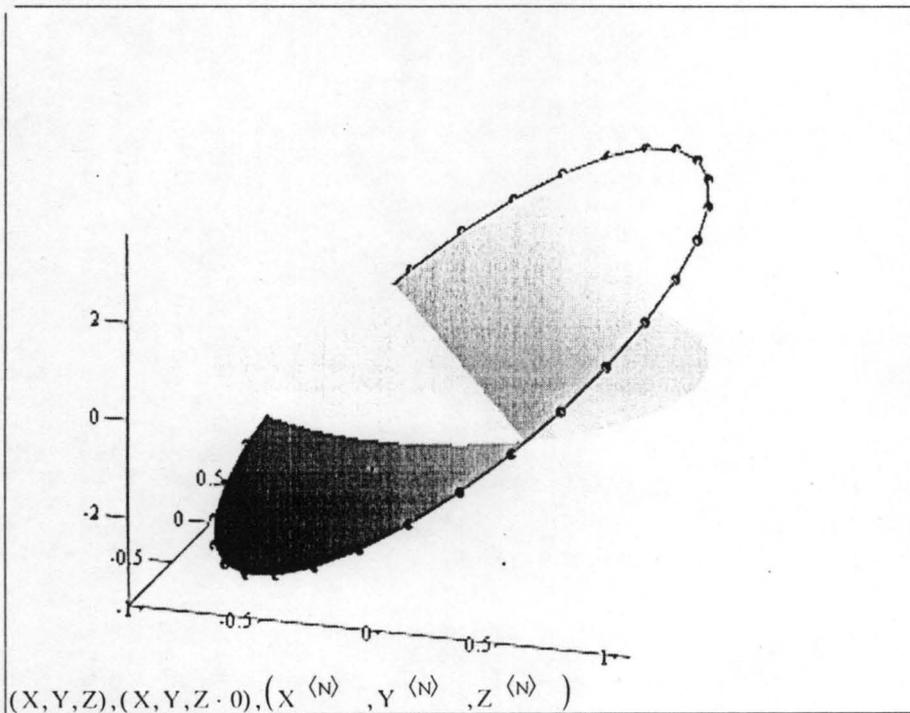
$$N := 30 \quad i := 0..N \quad j := 0..N$$

$$X_{i,j} := \frac{j}{N} \cdot \cos\left(i \cdot \frac{2\pi}{N}\right) \quad Y_{i,j} := \frac{j}{N} \cdot \sin\left(i \cdot \frac{2\pi}{N}\right)$$

and then compute the corresponding  $z$  values

$$Z_{i,j} := f(X_{i,j}, Y_{i,j})$$

The corresponding plot looks like



In one of the examples above we considered the function

$$f(x_1, x_2) := x_1^2 - x_1 + x_2$$

over the region

$$x_1 + x_2^2 \leq 4 \quad x_1 \geq 0 \quad x_2 \geq 0$$

We can draw the points under consideration using a parametric surface plot. First define a set of  $x_1$

and  $x_2$  values which range over the critical region

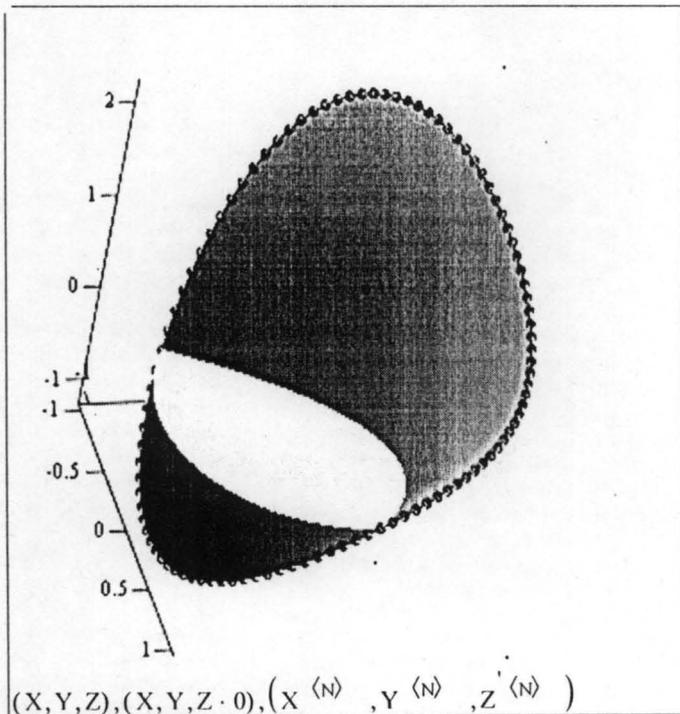
$$N := 100 \quad i := 0..N \quad j := 0..N$$

$$X_{i,j} := \frac{j}{N} \cdot \cos\left(i \cdot \frac{2\pi}{N}\right) \quad Y_{i,j} := \frac{j}{N} \cdot \sin\left(i \cdot \frac{2\pi}{N}\right)$$

and then compute the corresponding  $z$  values

$$Z_{i,j} := f(X_{i,j}, Y_{i,j})$$

The corresponding plot looks like



In one of the examples above we considered the function  $x_3 := 1.6$

$$f(x_1, x_2) := (6 \cdot x_1 - 3 \cdot x_2^2) + (4 \cdot x_2 - 2 \cdot x_2^2) + (2 \cdot x_3 - 0.33333333 \cdot x_3^2)$$

over the region

$$x_1 + 2 \cdot x_2 + x_3 \leq 4 \quad x_3 \geq 0 \quad x_1 \geq 0 \quad x_2 \geq 0$$

We can draw the points under consideration using a parametric surface plot. First define a set of

$x_1$ ,  $x_2$  and  $x_3$  values which range over the critical region

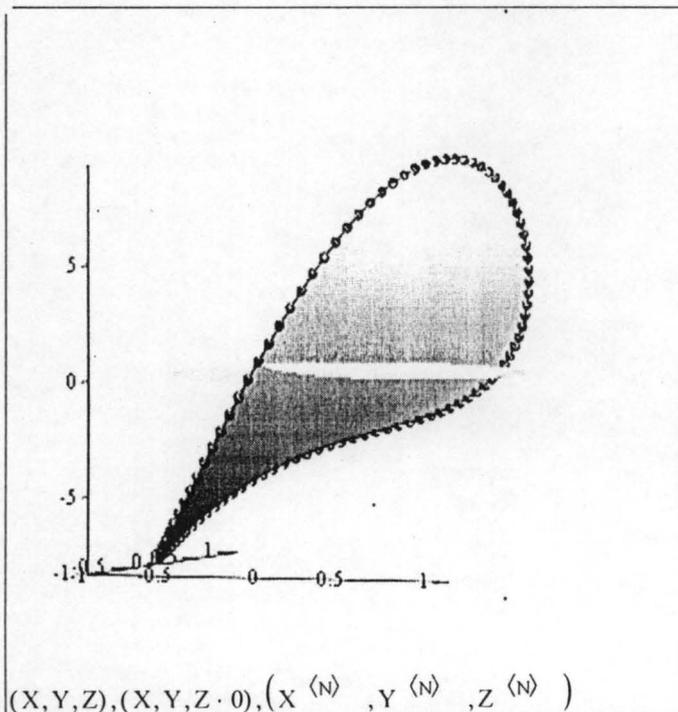
$$N := 100 \quad i := 0..N \quad j := 0..N$$

$$X_{i,j} := \frac{j}{N} \cdot \cos\left(i \cdot \frac{2\pi}{N}\right) \quad Y_{i,j} := \frac{j}{N} \cdot \sin\left(i \cdot \frac{2\pi}{N}\right)$$

and then compute the corresponding z values

$$Z_{i,j} := f(X_{i,j}, Y_{i,j})$$

The corresponding plot looks like



## 5.0 COMPUTATIONAL ALGORITHM FOR RATIONAL OBJECTIVE

### 5.1 ALGORITHM/OUTPUT OF RATIONAL OBJECTIVE USING MATHCAD

#### 5.2.1 Example 5.1

$$f(x_1, x_2) := \frac{x_1^2 - x_1 + x_2}{x_1 + 2}$$

$$x_1 := 4 \quad x_2 := 0$$

Given

$$x_1 + x_2^2 \leq 4$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$\text{Maximiz}(f, x_1, x_2) = \begin{pmatrix} 3.99 \\ 0.101 \end{pmatrix}$$

$$f(x_1, x_2) := \frac{x_1^2 - x_1 + x_2}{x_1 + 2} \quad f(x_1, x_2) = 2$$

We can draw the points under consideration using a parametric surface plot. First define a set of  $x_1$  and  $x_2$  values which range over the critical region

$$N := 100 \quad i := 0..N \quad j := 0..N$$

$$X_{i,j} := \frac{j}{N} \cdot \cos\left(i \cdot \frac{2\pi}{N}\right) \quad Y_{i,j} := \frac{j}{N} \cdot \sin\left(i \cdot \frac{2\pi}{N}\right)$$

and then compute the corresponding  $z$  values

$$Z_{i,j} := f(X_{i,j}, Y_{i,j})$$

### 5.2.2 Example 5.2

$$f(x_1, x_2) := \frac{3x_1 + x_2 + 2}{x_1 + x_2 + 1}$$

$$x_1 := 1 \quad x_2 := 1$$

Given

$$x_1 + 2 \cdot x_2 \leq 3$$

$$3 \cdot x_1 + x_2 \leq 4$$

$$x_1 \geq 1$$

$$x_2 \geq 0$$

$$\text{Maximize}(f, x_1, x_2) = \begin{pmatrix} 1.333 \\ 0 \end{pmatrix} \quad f(x_1, x_2) = 2$$

$$f(x_1, x_2) := \frac{3x_1 + x_2 + 2}{x_1 + x_2 + 1}$$

We can draw the points under consideration using a parametric surface plot. First define a set of  $x_1$  and  $x_2$  values which range over the critical region

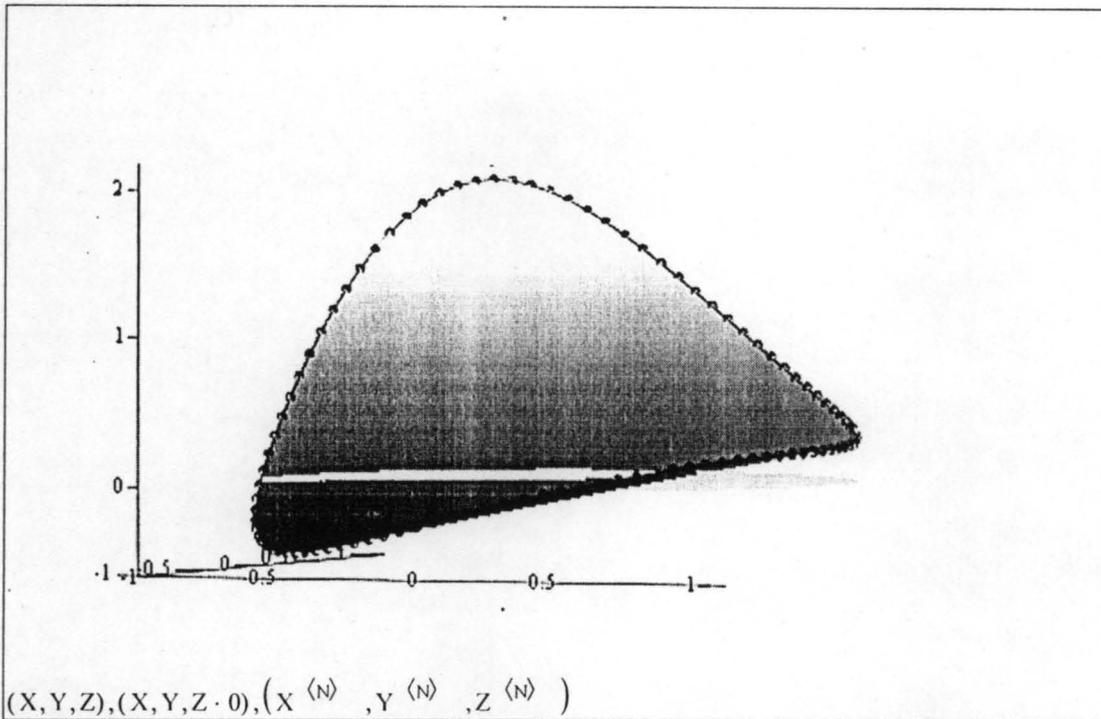
$$N := 100 \quad i := 0..N \quad j := 0..N$$

$$X_{i,j} := \frac{j}{N} \cdot \cos\left(i \cdot \frac{2\pi}{N}\right) \quad Y_{i,j} := \frac{j}{N} \cdot \sin\left(i \cdot \frac{2\pi}{N}\right)$$

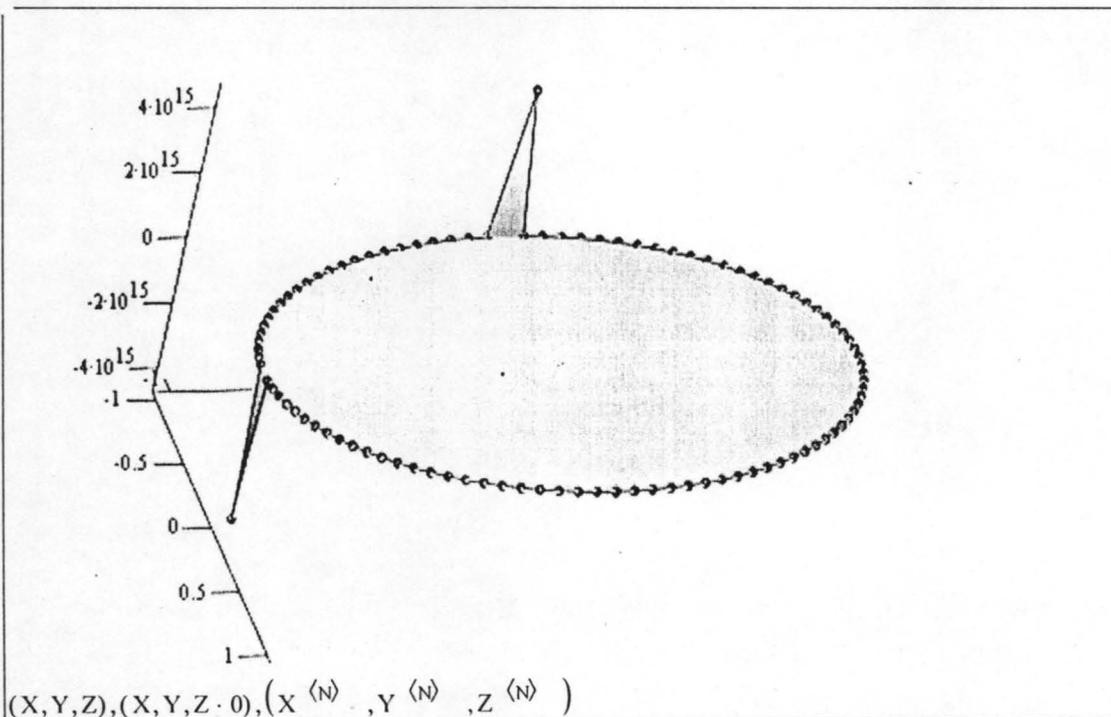
and then compute the corresponding z values

$$Z_{i,j} := f(X_{i,j}, Y_{i,j})$$

## 5.2 GRAPHICAL SOLUTIONS



Output of Example 5.1



Output of Example 5.2

## REMARK (PART OF CH.V)

Two different methods are considered in this case, the first one is the direct utilization of the rational programming to solve problems in chapter V and the method worked perfectly for them. While another method considered for the case of splitting fractional objective function into numerator function combined with constraints and the denominator function combined with constraints and the latter divide the formal in order to get the value of rational function.

## 5.3 ANALYSIS OF RESULT, RECOMMENDATION AND CONCLUSION

### 5.3.1 ANALYSIS

The first, second and fourth chapters discussed and practicalised idea of separable Algorithm, and also introduced theory on separable objective. While the two remaining chapters, that is chapter three and five discussed and practicalised idea of Rational Algorithm and they also introduced theory on rational objective. We are able to solve the rational objective using powerful mathematical software package, Mathcad. It solves the problem with ease and can be viewed from any region.

### 5.3.2 RECOMMENDATION

The code in chapter V can successively solve the Nonlinear (Separable) programming problem most especially problems with power of 2 (i.e Quadratic in nature) while that of power greater than 2 are not exhaustive in practice or considered. This work can be considered for day-to-day activities in industries, home management, and factory production e.t.c.

### 5.3.3 CONCLUSION.

From the foregoing analysis, it is cleared that the more the numbers of variables, constraints and cross product terms, the more the computation time, utilization of computer space and the iterations. The goal of adopting the separable method is to select the grids that yield a nearly optimal solution for the original problem. The more the grids (or segments) considered in the cases of both the separable and the rational objectives, the more we tend to close towards exact values of the objective functions. That is to say that the method gives fairly good approximation. A linear function is both concave and convex, therefore a basic solution that is locally optimal must be globally optimal.

One drawback of approximating function by piecewise linear function as described in chapter two that achieving a close approximation requires a large number line segments (variables), whereas such a fine grid for gridpoints (breakpoints) is needed only in the immediate neighbourhood of an optimal solution. Therefore, more sophisticated approaches that use a succession of two-segment piecewise linear functions have been developed in chapter IV to obtain successively closer approximations within this immediate neighbourhood. This kind of approach tends to be both faster and more accurate in closely approximating an optimal solution.

## APPENDIX (SEPARABLE CODE)

### 'Module 1

Public Lambda() As Single '2D array rows=n, cols=S, h=m+n+1 ==>(m=no of constraints, 1 for f, n for no vars) \_

r,c,h. h=0 f; h=1 to m constraints; h=m to m+n sum of lamda for each of n vars

Public BaseVariables() As Variable 'vector size=m+n

Public Constant() As Single 'vector size=m+n+1

Public x() As Single 'matrix nxS

Public g() As Single 'matrix mxnxS

Public f() As Single 'matrix nXS

Public S As Integer 'no of segments, k=0 to s

Public n As Integer 'no of variables

Public m As Integer 'no of constraints'

Public k As Integer 'current segment no

Dim theRow(3) As Single

Dim Entering As Variable

Dim Leaving As Variable

Dim pr As Integer, pc As Integer

Public Function PivotColumn(Order As Integer) As Integer

Dim Temp() As Integer

ReDim Temp(n \* (S + 1))

Dim Store As Integer

For i = 1 To n \* (S + 1)

Temp(i) = i

Next i

For i = 1 To n \* (S + 1)

For j = 1 To n \* (S + 1)

If Lambda(Convert(Temp(i), S + 1).Row, Convert(Temp(i), S + 1).Col, 4) >

Lambda(Convert(Temp(j), S + 1).Row, Convert(Temp(j), S + 1).Col, 4) Then

Store = Temp(j)

Temp(j) = Temp(i)

Temp(i) = Store

End If

Next j

Next i

PivotColumn = Temp(Order)

End Function

Public Function PivotRow(PivotCol As Integer) As Integer

Dim rows(3) As Integer, saved As Integer

If Lambda(Convert(PivotCol, S + 1).Row, Convert(PivotCol, S + 1).Col, 1) <> 0 Then theRow(1)

= Constant(1) / Lambda(Convert(PivotCol, S + 1).Row, Convert(PivotCol, S + 1).Col, 1) Else

theRow(1) = 99999999999999#

If Lambda(Convert(PivotCol, S + 1).Row, Convert(PivotCol, S + 1).Col, 2) <> 0 Then theRow(2)

= Constant(2) / Lambda(Convert(PivotCol, S + 1).Row, Convert(PivotCol, S + 1).Col, 2) Else

theRow(2) = 99999999999999#

```

If Lambda(Convert(PivotCol, S + 1).Row, Convert(PivotCol, S + 1).Col, 3) <> 0 Then theRow(3)
= Constant(3) / Lambda(Convert(PivotCol, S + 1).Row, Convert(PivotCol, S + 1).Col, 3) Else
theRow(3) = 999999999999999#
rows(1) = 1: rows(2) = 2: rows(3) = 3
For i = 1 To 3
  For j = 1 To 3
    If theRow(rows(i)) < theRow(rows(j)) Then
      saved = rows(i)
      rows(i) = rows(j)
      rows(j) = saved
    End If
  Next j
Next i
If theRow(rows(1)) = theRow(rows(2)) Then
  If BaseVariables(rows(1)).Row = Convert(PivotCol, S + 1).Row Then
    PivotRow = rows(1)
  Else
    PivotRow = rows(2)
  End If
Else
  PivotRow = rows(1)
End If
End Function

```

```

Sub Init()
Dim i As Integer, j As Integer, k As Integer

```

```

ReDim Lambda(n, S, 4)

```

```

For i = 1 To n
  For k = 0 To S
    Lambda(i, k, 1) = g(1, i, k)
  Next k
Next i

```

```

For j = 1 To n
  For k = 0 To S
    Lambda(j, k, 4) = f(j, k)
  Next k
Next j

```

```

For j = 1 To n
  For k = 0 To S
    If j = 1 Then Lambda(j, k, 2) = 1 Else Lambda(j, k, 2) = 0
  Next k
Next j

```

```

For j = 1 To n
  For k = 0 To S
    If j = 2 Then Lambda(j, k, 3) = 1 Else Lambda(j, k, 3) = 0
  Next k

```

Next j

Lambda(0, 0, 1) = 1

Lambda(0, 0, 2) = 0

Lambda(0, 0, 3) = 0

Lambda(0, 0, 4) = 0

ReDim BaseVariables(m + n)

BaseVariables(1).Col = 0

BaseVariables(1).Row = 0

BaseVariables(2).Row = 1

BaseVariables(2).Col = 0

BaseVariables(3).Row = 2

BaseVariables(3).Col = 0

ReDim Constant(m + n + 1)

Constant(1) = Form1.Constraints(1).RightSide

Constant(2) = 1

Constant(3) = 1

Constant(4) = 0

End Sub

Public Function Convert(Number As Integer, Base As Integer) As Variable

Convert.Col = Number Mod Base - 1

Convert.Row = (Number \ Base) Mod Base + 1

If Convert.Col < 0 Then

Convert.Col = Convert.Col + Base

Convert.Row = Convert.Row - 1

End If

End Function

Public Sub Iterate()

Dim i As Integer

Dim enteringRow As Integer

i = 1

Do

pc = PivotColumn(i)

pr = PivotRow(pc)

enteringRow = Convert(pc, S + 1).Row

If BaseVariables(1).Row = enteringRow And pr <> 1 Then

problemvar = 1

ElseIf BaseVariables(2).Row = enteringRow And pr <> 2 Then

problemvar = 2

ElseIf BaseVariables(3).Row = enteringRow And pr <> 3 Then

problemvar = 3

Else

problemvar = 0

End If

If problemvar <> 0 And Abs(BaseVariables(problemvar).Col - Convert(pc, S + 1).Col) <> 1

Then

```

    i = i + 1
Else
    Leaving.Row = BaseVariables(pr).Row
    Leaving.Col = BaseVariables(pr).Col

    BaseVariables(pr).Col = Convert(pc, S + 1).Col
    BaseVariables(pr).Row = Convert(pc, S + 1).Row

    Entering.Row = BaseVariables(pr).Row
    Entering.Col = BaseVariables(pr).Col
    ElementaryRowOps
    Exit Do
End If
Loop
End Sub
Sub ElementaryRowOps()
Dim i As Integer, Row As Integer
'PivotRow/2
For i = 0 To n * (S + 1)
    Lambda(Convert(i, S + 1).Row, Convert(i, S + 1).Col, pr) = _
    Lambda(Convert(i, S + 1).Row, Convert(i, S + 1).Col, pr) / PivotNumber
Next i
Constant(pr) = Constant(pr) / PivotNumber

'row=row-row(pr)*pivotrow
For Row = 1 To 4
    If Row <> pr Then
        Constant(Row) = _
        -Lambda(Convert(pc, S + 1).Row, Convert(pc, S + 1).Col, Row) * _
        Constant(pr) + _
        Constant(Row)
        For i = 0 To n * (S + 1)
            Lambda(Convert(i, S + 1).Row, Convert(i, S + 1).Col, Row) = _
            -Lambda(Convert(pc, S + 1).Row, Convert(pc, S + 1).Col, Row) * _
            Lambda(Convert(i, S + 1).Row, Convert(i, S + 1).Col, pr) + _
            Lambda(Convert(i, S + 1).Row, Convert(i, S + 1).Col, Row)
        Next i
    End If
Next Row
End Sub
Function PivotNumber() As Single
PivotNumber = Lambda(Convert(pc, S + 1).Row, Convert(pc, S + 1).Col, pr)
End Function

```

**'Class Module SubExp**

```
Public Value As String
Public Function Var() As String
If InStr(Value, "x") = 0 Then
    Var = ""
Else
    Var = Mid(Value, InStr(Value, "x"), InStr(Value + "^", "^") - InStr(Value, "x"))
End If
End Function
Public Function Exponent() As Single
Dim i As Integer
i = InStr(Value, "^")
If i = 0 Then
    Exponent = 1
Else
    Exponent = Right(Value, Len(Value) - i)
End If
End Function
Public Function Multiplier() As Single
If Var <> Empty Then
    Multiplier = Val(Left(Value, InStr(Value, "x") - 1))
    If Not IsNumeric(Left(Value, InStr(Value, "x") - 1)) Then
        If Left(Value, InStr(Value, "x") - 1) = "-" Then
            Multiplier = -1
        Else
            Multiplier = 1
        End If
    End If
ElseIf InStr(Value, "^") <> 0 Then
    Multiplier = Left(Value, InStr(Value, "^"))
Else
    Multiplier = Value
End If
End Function
```

### 'Class Module Expression

```
Public Value As String
Public Function Evaluate(Bindings As Collection) As Single
Dim subs As Collection
Dim E As SubExp, B As Binding
Set subs = Split("+")
For Each E In subs
    For Each B In Bindings
        If InStr(E.Value, B.Var) > 0 Then

            Exit For
        End If
    Next B
    Evaluate = Evaluate + Eval(E, B.Value)
Next E
End Function

Public Function Split(Tokens As String) As Collection
Dim noToks As Integer, i As Integer, Pos As Integer
Dim SubExp1 As SubExp
Dim Exp As String
Exp = Value
noToks = Len(Tokens)
Dim Toks() As String
ReDim Toks(noToks)
For i = 1 To noToks
    Toks(i) = Mid(Tokens, i, 1)
Next i
Set Split = New Collection
Pos = FindToken(Toks, noToks, Exp)
While Pos <> 0
    Set SubExp1 = New SubExp
    SubExp1.Value = Left(Exp, Pos - 1)
    Exp = Right(Exp, Len(Exp) - Pos + 1)
    Split.Add SubExp1
    Pos = FindToken(Toks, noToks, Exp)
Wend
Set SubExp1 = New SubExp
SubExp1.Value = Exp
Split.Add SubExp1
End Function

Private Function FindToken(T() As String, no As Integer, Exp As String)
Dim f As Integer
FindToken = InStr(2, Exp, T(1))
For i = 1 To no
    f = InStr(2, Exp, T(i))
    If FindToken = 0 Then FindToken = f
    If f < FindToken And f <> 0 Then FindToken = f
Next i
End Function

Private Function Eval(E As SubExp, V As Single) As Single
```

```

Eval = E.Multiplier * V ^ E.Exponent
End Function
Public Function RightSide() As Single
Dim Temp As Collection, SubE As SubExp
Set Temp = Split("+ -")
For Each SubE In Temp
    If SubE.Var = Empty Then
        RightSide = -SubE.Multiplier
        Exit For
    End If
Next SubE
End Function

```

### **'Class Module Binding**

```

Public Var As String
Public Value As Single

```

```

Attribute VB_Name = "SeparateFunctions"

```

```

'Step I

```

```

Option Explicit

```

```

Public f() As Expression ' n

```

```

Public g() As Expression 'mxn

```

```

Public Sub Start()

```

```

Dim Temp As Collection

```

```

Dim Temp2 As SubExp

```

```

Dim newExp As Expression

```

```

Dim j As Integer, i As Integer

```

```

ReDim f(1 To 10)

```

```

With Form1

```

```

    Set Temp = .Objective.Split("+ -")

```

```

    Do

```

```

        j = j + 1

```

```

        Set Temp2 = Temp.Item(1)

```

```

        Dim se As SubExp

```

```

        Set se = New SubExp

```

```

        Set newExp = New Expression

```

```

        For Each se In Temp

```

```

            If se.Var = Temp2.Var Then newExp.Value = newExp.Value + se.Value

```

```

        Next se

```

```

        i = 1

```

```

        Do While i <= Temp.Count

```

```

            If Temp.Item(i).Var = Temp2.Var Then

```

```

                Temp.Remove i

```

```

                i = 0

```

```

            End If

```

```

            i = i + 1

```

```

        Loop

```

```

        If j / 10# = Int(j / 10) Then ReDim Preserve f(1 To j + 10)

```

```

        Set f(j) = newExp

```

```

    Loop Until Temp.Count = 0

```

```

End With

```

```

n = j 'no of variables
End Sub
Public Sub Dofor(gm As Integer, Exp As Expression)
Dim Temp As Collection
Dim Temp2 As SubExp
Dim newExp As Expression
Dim j As Integer, i As Integer
With Form1
    Set Temp = Exp.Split("+ -")
    Do
        j = j + 1
        Set Temp2 = Temp.Item(1)
        Dim se As SubExp
        Set se = New SubExp
        Set newExp = New Expression
        For Each se In Temp
            If se.Var = Temp2.Var Then newExp.Value = newExp.Value + se.Value
        Next se
        i = 1
        Do While i <= Temp.Count
            If Temp.Item(i).Var = Temp2.Var Then
                Temp.Remove i
                i = 0
            End If
            i = i + 1
        Loop
        Set g(gm, j) = newExp
    Loop Until Temp.Count = 0
End With
End Sub

```

#### **'Form Module Step II**

```

Private Sub Command1_Click()
Form1.Show
Form1.SetFocus
Unload Me
End Sub

```

```

Private Sub Command2_Click()
linearization.S = Val(Text1.Text)
Form3.Show
End Sub

```

```

Private Sub Form_Load()
Dim Temp As Collection, i As Integer
Dim temp2 As Collection
Set temp2 = New Collection
For i = 1 To n
    Set Temp = SeparateFunctions.f(i).Split("+ -")
    temp2.Add Temp.Item(1)
Next i

```

```

Grid.rows = n + 1
Grid.Col = 0
For i = 1 To n
    Grid.Row = i
    Grid.Text = temp2(i).Var
Next i
End Sub

```

```

Private Sub grid_KeyDown(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
Case 46: Grid.Text = Empty
Case 8: Grid.Text = Empty
'Case Else: MsgBox Str(KeyCode)
End Select
End Sub

```

```

Private Sub Grid_KeyPress(KeyAscii As Integer)
If Grid.Row <> 0 And Grid.Col <> 0 Then
    Grid.Text = Grid.Text + Chr(KeyAscii)
End If
End Sub

```

### **'Form Module Step III**

```

Private Sub Form_Load()
Dim i As Integer, j As Integer, StartVal As Single, Interval As Single
Dim k As Integer
ReDim linearization.x(n, S)
ReDim linearization.f(n, S)
ReDim linearization.g(m, n, S)
'x
For i = 1 To n
    Form2.Grid.Row = i
    Form2.Grid.Col = 1
    StartVal = Val(Form2.Grid.Text)
    Form2.Grid.Col = 2
    Interval = (Val(Form2.Grid.Text) - StartVal) / S
    For j = 0 To S
        linearization.x(i, j) = StartVal
        StartVal = StartVal + Interval
    Next j
Next i
'f
Dim Binds As Collection, Bind As Binding
Set Bind = New Binding
Set Binds = New Collection
Binds.Add Bind
For i = 1 To n
    Form2.Grid.Row = i
    Form2.Grid.Col = 0

```

```

Binds.Item(1).Var = Form2.Grid.Text
For j = 0 To S
    Binds.Item(1).Value = linearization.x(i, j)
    linearization.f(i, j) = SeparateFunctions.f(i).Evaluate(Binds)
Next j
Next i
'g
For i = 1 To m
    For j = 1 To n
        Form2.Grid.Row = j
        Form2.Grid.Col = 0
        Binds.Item(1).Var = Form2.Grid.Text
        For k = 0 To S
            Binds.Item(1).Value = linearization.x(j, k)
            linearization.g(i, j, k) = SeparateFunctions.g(i, j).Evaluate(Binds)
        Next k
    Next j
Next i

Grid.Cols = 2 * n + m * n + 1
Grid.rows = S + 2
'show table
Grid.Row = 0: Grid.Col = 0: Grid.Text = "k"
For i = 0 To S
    Grid.Row = i + 1
    Grid.Text = Str(i)
Next i

For i = 1 To n
    Grid.Col = i: Grid.Row = 0: Grid.Text = "x" + Str(i) + "k"
    For j = 0 To S
        Grid.Row = j + 1
        Grid.Text = linearization.x(i, j)
    Next j
Next i

For i = 1 To n
    Grid.Col = n + i: Grid.Row = 0: Grid.Text = "f" + LTrim(Str(i)) + "(x" + LTrim(Str(i)) + "k)"
    For j = 0 To S
        Grid.Row = j + 1
        Grid.Text = linearization.f(i, j)
    Next j
Next i

For i = 1 To m
    For j = 1 To n
        Grid.Col = 2 * n + (i - 1) * n + j
        Grid.Row = 0: Grid.Text = "g" + LTrim(Str(i)) + LTrim(Str(j))
        For k = 0 To S
            Grid.Row = k + 1
            Grid.Text = linearization.g(i, j, k)
        Next k
    Next j
Next i

```

```
Next k  
Next j  
Next i
```

```
End Sub
```

```
Private Sub NextCommand_Click()  
Form4.Show  
Form4.SetFocus  
End Sub
```

#### **'Form Module IV**

```
Private Sub Command1_Click()  
Form3.Show  
Form3.SetFocus  
Unload Me  
End Sub
```

```
Private Sub Command2_Click()  
Iterate  
Form_Load  
End Sub
```

```
Private Sub Form_Initialize()  
Init  
End Sub
```

```
Private Sub Form_Load()  
Dim i As Integer, j As Integer, k As Integer  
Grid.rows = 5  
Grid.Cols = n * (S + 1) + 3  
Grid.Row = 0  
Grid.Col = 0  
For i = 1 To n  
    For j = 0 To S  
        Grid.Col = Grid.Col + 1  
        Grid.Text = "L" & LTrim(Str(i)) & LTrim(Str(j))  
    Next j  
Next i  
For i = 1 To n * (S + 1)  
    For j = 1 To 4  
        Grid.Row = j  
        Grid.Col = i  
        Grid.Text = Lambda(Convert(i, S + 1).Row, Convert(i, S + 1).Col, j)  
    Next j  
Next i  
Grid.Col = n * (S + 1) + 1
```

```
'slack var  
Grid.Row = 0  
Grid.Text = "L00"  
For i = 1 To 4  
    Grid.Row = i  
    Grid.Text = Lambda(0, 0, i)  
Next i
```

```
'objective  
For i = 1 To n * (S + 1)  
    Grid.Row = 4  
  
    Grid.Col = i  
    Grid.Text = Lambda(Convert(i, S + 1).Row, Convert(i, S + 1).Col, 4)
```

```
Next i
'base vars
Grid.Col = 0
Grid.Row = 1: Grid.Text = "L" & LTrim(Str(linearization.BaseVariables(1).Row)) &
LTrim(Str(linearization.BaseVariables(1).Col))
Grid.Row = 2: Grid.Text = "L" & LTrim(Str(linearization.BaseVariables(2).Row)) &
LTrim(Str(linearization.BaseVariables(2).Col))
Grid.Row = 3: Grid.Text = "L" & LTrim(Str(linearization.BaseVariables(3).Row)) &
LTrim(Str(linearization.BaseVariables(3).Col))
```

```
'constants
Grid.Col = n * (S + 1) + 2
Grid.Row = 0: Grid.Text = "Right Side"
Grid.Row = 1: Grid.Text = Constant(1)
Grid.Row = 2: Grid.Text = Constant(2)
Grid.Row = 3: Grid.Text = Constant(3)
Grid.Row = 4: Grid.Text = Constant(4)
End Sub
```

```
Public Objective As Expression
Public Constraints As Collection
```

```
Private Sub AddCommand_Click()
Dim E As Expression
Set E = New Expression
E.Value = Text2.Text
Constraints.Add E
List1.AddItem Text2.Text
End Sub
```

```
Private Sub Command1_Click()
Dim i As Integer, TempExp.As Expression
Objective.Value = Text1.Text
Start
m = List1.ListCount
Set TempExp = New Expression
ReDim SeparateFunctions.g(1 To m + 1, 1 To n + 1)
For i = 1 To m
    TempExp.Value = List1.List(i - 1)
    Dofor i, TempExp
Next i
Form2.Show
End Sub
```

```
Private Sub Form_Load()
Set Objective = New Expression
Set Constraints = New Collection
End Sub
```

## REFERENCES

Fredrick S. Hiller, Gerald J. Lieberman, 1995, Introduction to Operations Research, sixth edition. McGraw-Hill, Singapore.

HARVEY M. WAGNER (1989) Principles of Operations Research with application to managerial decisions, second edition. Price Hill, New Delhi.

Gary E. Whitehouse, Ben L. Wechsler (1976), Applied Operations Research, International edition, John Wiley and sons.

Frederic C. Jelen, James H. Black (1983), Cost and Optimization Engineering, second edition. McGraw-Hill Coy, USA..

S.A. Hovanessian (1976), Computational Mathematics in Engineering, first edition, D.C Health and Coy, Los Angeles, California.

Roger C. Pfaffenberger, David A. Walker (1976) Mathematical Programming for Economics and Business, First edition, IOWA State University Press, IOWA STATE.

RICHARD BRONSON (1983), Schaum's Outline series Theory & Problems of Operations Research, McGraw-Hill Coy, Singapore.

MORDEXAI AVRIEL (1976) Nonlinear Programming Analysis and Methods, Prentice Hall Inc. New Jersey.

Raphael T. Haftka and Zafer Gurdal (1992), Elements of Structural Optimization, the third revised, Dordecht, Netherlands.-