COMPARATIVE STUDY OF DIFFERENT METHODS FOR SOLVING LINEAR FRACTIONAL PROGRAMMING PROBLEM

by

MOHAMMED FORHAD UDDIN
Roll No. 100109009P, Registration No. 0110649,
Session: October 2001

MASTER OF PHILOSOPHY
in Mathematics

Department of Mathematics,
Bangladesh University of Engineering and Technology, Dhaka-1000
December 2004

The Thesis Entitled

COMPARATIVE STUDY OF DIFFERENT METHODS FOR SOLVING LINEAR FRACTIONAL PROGRAMMING PROBLEM

Submitted By

MOHAMMED FORHAD UDDIN

Registration No. 0110649, Session: October, 2001, a part-time M. Phil. student in Mathematics has been accepted as satisfactory in partial fulfillment for the

Degree of

MASTER OF PHILOSOPHY in Mathematics

on December 28, 2004

Board of Examiners

1. **Mr. Md. Obayedullah** Assistant Professor Department of Mathematics, BUET	28/12/04	Supervisor (Chairman)
2. **Dr. Md. Ainul Islam** Professor Department of Mathematics, Dhaka University		Co-Supervisor
3. **Dr. Nilufar Farhat Hossain** Professor and Head Department of Mathematics, BUET	28/12/04	Member (Ex-Officio)
4. **Dr. Md. Mustafa Kamal Chowdhury** Professor Department of Mathematics, BUET	28/12/04	Member
5. **Dr. Ganesh Chandra Roy** Associate professor Department of Mathematics, Chittagong University	28/12/04	Member (External)

ABSTRACT

In this thesis, we study the established methods of Charnes & Cooper, Bitran & Novaes and Swarup's primal & dual simplex type for solving linear fractional programming problem, suggest a modification for Swarup's simplex type method and compare the methods among themselves. To apply these methods on large-scale linear fractional programming problem, we need computer-oriented program of these methods. To fulfill this purpose, we develop computer program (FORTRAN) of these methods and apply on a sizable large-scale linear fractional programming problem of an agricultural farm. Finally, conclusion is drawn in favour of our modified approach of Swarup's primal simplex method.

CANDIDATE'S DECLARATION

I hereby declare that the work which is being presented in the thesis entitled "Comparative study of different methods for solving linear fractional programming problem." submitted in partial fulfillment of the requirement for the award of the degree of Master of Philosophy in Mathematics, in the Department of Mathematics, Bangladesh University of Engineering and Technology, Dhaka is an authentic record of my own work. I have not submitted the matter presented in this thesis for the award of any other degree in this or any other university.

Date: December 28, 2004

(MOHAMMED FORHAD UDDIN)

ACKNOLEDGEMENT

All praises are due to Almighty for enabling me to complete this dissertation successfully. This is a great pleasure to express my wholehearted gratitude and indebtedness to my reverend supervisor Md. Obayedullah, Department of Mathematics, Bangladesh University of Engineering and Technology and co-supervisor Professor Md. Ainul Islam, Department of Mathematics, University of Dhaka, who have inspired me to initiate this work. They have extended their full cooperation by offering constant advice, encouragement and proper guidance throughout the entire period of work and preparation of this thesis. Undoubtedly, it would be impossible to complete this work without their invaluable supervision.

I express my deep regards to my honorable teacher, Dr. Nilufar Farhat Hossain, Professor and Head, Department of Mathematics, Bangladesh University of Engineering and Technology for providing me help, advice and necessary research facilities.

I also express my gratitude to my honorable teachers Dr. Md. Zakerullah, Professor Dr. Md. Mustafa Kamal Chowdhury, Md. A. K. Hazra, Mohammad Isa, Md. Abdul Quddus Mean, Dr. Md. Elias, Dr. Md. Abdul Maleque, and Dr. Md. Abdul Hakim Khan of the Department of Mathematics, Bangladesh University of Engineering and Technology for their invaluable guidance and helpful suggestions to complete this research work successfully.

I also offer my best regards to all respected teachers of this Department for their affections, encouragement and kind cooperation. Convey my thanks and best wishes to my friend Md. Moshiour Rahman of the Department of Mathematics, BRAC University, for his better suggestions and cooperation.

I acknowledge the help, co-operation of all office staff of this Department.

Finally, I convey my regards to my beloved parents, my wife and my relatives who have initiated and encouraged me to accomplish this work properly.

CONTENTS

CHAPTER -1

Introduction:

In this thesis, we study the established methods of Charnes-Cooper's[1962], Bitran-Novaes [1972] and Swarup[1964 & 1965] for solving Linear Fractional Programming (LFP) problem, suggest a modification for Swarup[1964] primal simplex type method and compare the methods among themselves. But a large scale LFP problem, which involves a numerous amount of data, constraints and variables, cannot be handled analytically with pencil and paper. To overcome the complexities of large-scale Linear Programming (LP) problem William et al. [1992] and Gillet [1998] developed computer program (FORTRAN). Here, we also generalize computer program (FORTRAN) William et al. [1992] for the methods of Charnes-Cooper's[1962], Bitran-Novaes [1972] and Swarup[1964 & 1965] to solve all types of LFP problems, whatever the size of data involve in it. To illustrate the purpose, we solve a sizable large-scale LFP on return to investment problem of an agricultural farm, which is formulated in section 1.5. To present our study, we require the following prerequisites:

1.1 Mathematical Programming problem or Mathematical Program (MP) deals with the optimization (maximization or minimization) of a function of several variables subject to a set of constraints (inequalities or equalities) imposed on the values of variables.

The general MP in n-dimensional Eucliadean space R^n can be stated as follows:

(MP) Maximize $f(x)$

Subject to

$$g_i(x) \le 0 \quad , i = 1,2,\ldots\ldots,m \tag{1.1}$$

$$h_j(x) = 0 \quad , j = 1,2,\ldots\ldots,p \tag{1.2}$$

$$x \in S \tag{1.3}$$

Where $x = (x_1, x_2, \ldots, x_n)^T$ is the vector of unknown decision variables and $f(x)$, $g_i(x)$, $(i = 1, 2, \ldots, m)$, $h_j(x)$, $(j = 1, 2, \ldots, p)$ are the real valued functions

The function $f(x)$ is known as objective function, and inequalities (1.1), equation (1.2) and the restriction (1.3) are referred to as the constraints. We have started the MP as maximization one. This has been done without any loss of generality, since a minimization problem can always be converted into a maximization problem using the identity

$$\min f(x) = \max(-f(x)) \tag{1.4}$$

i.e., the minimization of $f(x)$ is equivalent to the maximization of $(-f(x))$.

The set S is normally taken as a connected subset of R^n. Here the set S is taken as the entire space R^n. The set $X = \{x \in S, g_i(x) \le 0 ; h_j(x) = 0, i = 1, 2, \ldots, m, j = 1, 2, \ldots, p\}$ is known as the feasible region, feasible set or constraint set of the program MP and any point $x \in X$ is a feasible solution or feasible point of the program MP which satisfies all the constraints of MP. If the constraint set X is empty (i.e. $X = \phi$), then there is no feasible solution, in this case the program MP is inconsistent.

A feasible point $x^0 \in X$ is known as a global optimal solution to the program MP if

$$f(x) \le f(x^0), x \in X \tag{1.5}$$

A **global optimal solution** x^0 of MP program is indeed a **global maximum** point of the program MP. A point x^0 is said to be a strict global maximum point of $f(x)$ over X if the strict inequality (<) in (1.5) holds for all $x \in X$ and $x \neq x^0$

A point $x^* \in X$ is a local or **relative maximum** point of $f(x)$ over X if there exists some $\varepsilon > 0$ such that

$$f(x) \le f(x^*), \forall x \in X \cap N_\varepsilon(x^*).$$

Where $N_\varepsilon(x^*)$ is the neighborhood of x^* having radius ε. Similarly, global minimum and local minimum can be defined by changing the sense of inequality.

The MP can be broadly classified into two categories, **unconstrained optimization** problem and **constrained optimization problem.** If the constraint set X is the whole space R^n, program MP is then known as an unconstrained optimization problem, in this case, we are interested in finding a point of R^n at which the objective function has an optimum value . On the contrary, if **X** is a proper subset of R^n

If both the objective function and the constraint set are linear, then MP is called **a linear programming problem (LPP)** or **a linear program (LP)**

On the other hand, non-linearity of the objective function or constraints gives rise to **non-linear programming problem or a non-linear program (NLP)** Several algorithms have been developed to solve certain NLP

1. 2 General Linear Program (GLP)

The GLP is to optimize a linear function subject to linear equality and inequality constraints. In other words, we need to determine the value of $x_1, x_2, \ldots, x_n$ that solve the program

$$(GLP) \quad \text{Maximize (or Minimize)} \quad Z = \sum_{j=1}^{n} c_j \tag{1.6}$$

Subject to

$$\sum_{j=1}^{n} a_{ij} x_j (\leq = \geq) b_i \quad , i = 1, 2, \ldots, m \tag{1.7}$$

$$x_j \geq 0 \tag{1.8}$$

in which c_j, (j=1,2,....,n) be the profit (or cost) coefficient , a_{ij} (i =1,2, ...,m, j=1,2,.....,n) be the coefficients matrix $A=(a_{ij})$ and x_j be the decision variables .

The linear function (1.6) which is to be optimized (maximized or minimized) is known as the objective function of the GLP The inequation (1.7) are constraints of the GLP An n-tuple $(x_1, x_2, \ldots, x_n)^t \in R^n$ which satisfies the constraints of the is known as a solution to the GLP.

Feasible solution: Any solution x_j, $(j = 1,2 \ldots n)$ to the GLP is called a feasible solution if it satisfies equations (1.7) and the non-negative restrictions (1.8).

Optimal solution: A feasible solution x_j, $(j = 1,2 \ldots n)$ is said to be an optimal solution to the GLP if it gives the maximum (or minimum) value of the objective function (1.6)

Constraint set : The set of feasible solution to the GLP is called a constraint set if
$X=\{(x_1,x_2,\ldots,x_n)^T ; (x_1,x_2,\ldots,x_n)^T \in R^n$ and (1.7) holds at $(x_1,x_2,\ldots,x_n)^T\}$.

Standard Linear Program (LPI): Every GLP can be reduced to an equivalent LPI as explained below:

(i) Conversion of right hand side constraint to non-negative : If a right hand side constant of a constraint is negative , it can be made non-negative by multiplying both sides of the constraints by -1(if necessary).

(ii) Conversion of inequality constraint to equality :

(a) Slack Variable: For an inequality constraint of the form

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

adding a non-negative variable x_{n+1} can be made equation

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

and the non-negative variable x_{n+1} is called the slack variable

(b) Surplus Variable : For an inequality constraint of the form

$$\sum_{j=1}^{n} a_{ij} x_j \ge b_j \quad (i = 1,2,\ldots,m : b_i \ge 0),$$

subtracting a non-negative variables x_{n+1} can be made equation

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

and the non-negative variable x_{n+1} is called the surplus variable .

So, without any loss of generality a **standard linear program** can be written as follows

$$(LPI) \qquad Maximize \quad Z = c^T x \tag{1.9}$$

$$Subject\ to \quad Ax = b \tag{1.10}$$

$$x \geq 0 \tag{1.11}$$

Where c, $x \in R^n$, A is an $m \times n$ matrix. $b \geq 0$ & $b \in R^m$.

In LPI, the $m \times n$ matrix $A = (a_{ij})$ $(i=1,2, \ldots,m , j=1,2, \ldots,n)$ is the coefficient matrix of the equality constraints ,$b = (b_1,b_2, \ldots,b_m)^t$ is the vector of right hand side constraints, the component of c are the profit factors , $x = (x_1,x_2,\ldots,x_n)^t \in R^n$ is the vector of variables called the decision variables and constraint (1.11) are known as non-negative constraints . The column vector of the matrix A referred to as activity vectors

Now we present the following definitions for standard linear program LPI :

Feasible solution: A vector $x = (x_1,x_2,\ldots, x_n)^T$ is a feasible solution of the LPI if it satisfies condition (1.10) and (1.11) .

Basic solution: A basic solution $x = (x_1,x_2,\ldots,x_n)^T$ to a LPI is a solution obtained by setting (n-m) variables equal to zero and solving the remaining m variables , provided that the determinant of the coefficients of these m variables are non-zero . The m variables are called basic variables.

Basic feasible solution: A basic feasible solution $x = (x_1,x_2, \ldots,x_n)^t$ to the LPI is a basic solution which also satisfies (1.11) , that is ,all basic variables are non-negative .

Non-degenerate basic feasible solution: A basic feasible solution $x = (x_1, x_2, \ldots, x_n)^T$ is said to be non-degenerate if it has exactly m positive (non zero) variables x_j, ($j = 1,2,\ldots,n$), that is, all basic variables are positive On the other hand , the solution is degenerate if one or more of the m basic variables are zero

Optimal solution: A basic feasible solution $x = (x_1, x_2, \ldots, x_n)^T$ is said to be an optimal solution or optimum if it maximize the objective function while satisfying the condition (1.10) and (1.11) , that is , $f(x^0) \geq f(x), \forall\, x \in X$

Basic solution and some Notations

Basic solution: Consider the constraints (1 10) i.e., Ax =b , are constraints and rank (A)=m (≤n). Let B be any non-singular m×m sub matrix made up of the columns of A and R be the remaining portion of matrix A Further, suppose that x_B is the vector of variables associated with the columns of B Then (1 10) can be written as-

$$[B, R]\begin{bmatrix} x_B \\ x_{NB} \end{bmatrix} = b$$

$$or, Bx_B + Rx_{NB} = b$$

That is , the general solution of (1 10) is given by

$$x_B = B^{-1}b - B^{-1}Rx_{NB}$$

$$or, x_B + B^{-1}Rx_{NB} = B^{-1}b \qquad (1.12)$$

Where the (n-m) variables x_{NB} can be assigned arbitrary values .The particular solution of (1.10) is given by

$$x_B = B^{-1}b \,, x_{NB} = 0 \qquad (1.13)$$

is called the basic solution to the system Ax =b with respect to the basic matrix B. The variables x_{NB} are known as the vector of non-basic variables and the variables x_B are said to be the vector of basic variables.

It should be noted that the column of A associated with the basic matrix B are linearly independent and all non-basic variables of x_{NB} are zero in a basic solution The equation (1.13) is known as feasible canonical form, if the basic solution given by it is feasible, that is, $x_B \geq 0$. Suppose there exists a basic feasible solution to the constraints (1.10) and (1.11).The coefficient of the variables in the objective function Z , after the basic variables from it have been eliminated , are called **relative profit factors** [In a minimization problem, we call cost factors in place of relative profit factors].

In order to find the relative profit factors corresponding to the basic matrix B, we partition the profit vector c as

$$c^T = \left[c_B^T, c_{NB}^T\right],$$

Where c_B and c_{NB} are the profit vectors corresponding to the variables x_B and x_{NB} respectively. The objective function then is

$$Z = c^T x = c_B^T x_B + c_{NB}^T x_{NB} \tag{1.14}$$

Subtracting in this equation (1.14) the values of x_B from (1.12), we get ,

$$\begin{aligned} Z &= c_B^T B^{-1} b - c_B^T B^{-1} R x_{NB} + c_{NB}^T x_{NB} \\ &= \overline{Z} + \left[c_{NB}^T - c_B^T B^{-1} R\right] x_{NB} \\ &= \overline{Z} + \overline{c}_B^T x_B + \overline{c}_{NB}^T x_{NB} \\ &= \overline{Z} + \overline{c}^T \mathbf{x}. \end{aligned}$$

Where

$$\overline{c} = \begin{bmatrix} c_B \\ c_{NB} \end{bmatrix}$$

$$\overline{c}_B = 0$$

$$\overline{c}_{NB}^T = c_{NB}^T - c_B^T B^{-1} R$$

$$\overline{Z} = c_B^T B^{-1} b$$

Here $\overline{c}$ is the vector of relative profit factors corresponding to the basic matrix B and $\overline{Z}$ is the value of the objective function at the basic solution is given by (1.13) Observe that the components of $\overline{c}$ corresponding to the basic variables are zero, which ought to be as is evident from the definition of $\overline{c}$

1.3 Simplex method

The simplex method is an iterative procedure for solving a linear program in a finite number of steps and provides all the information about the program. Also it indicates whether or not the program is feasible If the program is feasible, it either finds an optimal solution or indicates that an unbounded solution exists At first G B. Dantzig developed this method in 1950. Following Dantzig [1963], Gillet [1988] described the simplex method as below:

Basically the simplex method is an iterative procedure that can be used to solve any linear programming model if the needed computer time and storage are available It is assumed that the original linear programming model

$$\textit{Maximize} \quad Z = \sum_{j=1}^{p} c_j x_j \tag{1.15}$$

Subject to

$$\sum_{j=1}^{p} a_{ij} x_j (\le = \ge) b_i \quad , i = 1,2,\ldots,m \tag{1.16}$$

$$x_j , b_i \ge 0 \tag{1.17}$$

has been converted to the equivalent standard LP model.

$$\textit{Maximize} \quad Z = \sum_{j=1}^{p} c_j x_j \tag{1.18}$$

Subject to

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad , i = 1,2,\ldots,m \tag{1.19}$$

$$x_j \ge 0 \tag{1.20}$$

Which includes slack variables that have been added to the left side of each less than or equal to constraint, surplus variables that have been subtracted from the left side of each greater than or equal to constraint, and artificial variables that have been added to the left side of each greater than or equal to constraint and each equality. It is assumed that the profit coefficients for the slack and surplus variables are zero while the coefficients for the artificial variables are arbitrary small negative numbers (algebraically), say $-M$. The equivalent model necessarily assures us that each equation contains a variable with a coefficient of 1 in that equation and a coefficient zero in each of the other equations. If the original constraint was less than or equal to constraint, the slack variable in the corresponding equation will satisfy the condition just stated. Likewise, the artificial variables that have been added to the greater than or equal to constraint and each equality satisfy the condition for each of the remaining equations in the equivalent model. These slack and artificial variables are the basic variables in the initial basic solution of the equivalent problem.

The equivalent model is now rewritten as

Maximize Z (1.21)

Subject to

$$Z - \sum_{j=1}^{n} c_j x_j = 0 \tag{1.22}$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad , i = 1, 2, \ldots, m \tag{1.23}$$

$$x_j \geq 0 \tag{1.24}$$

Since $c_j = -M$ for each artificial variable, we must multiply by $-M$ each equation represented by (1. 23) that contains an artificial variable and add the resulting equations to equation (1.23) to give

Maximize Z (1.25)

Subject to

$$Z - \sum_{j=1}^{n} c_j x_j = b_0 \tag{1.26}$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad , i = 1, 2, \ldots, m \tag{1.27}$$

$$x_j \geq 0 \qquad (1.28)$$

Where $b_0 = -M\sum_{*} b_i$ and * represent the equations containing artificial variables. This assures us that each equation in (1.27) contains a slack or artificial variable that has a coefficient of 1 in that equation and a coefficient of zero in each of the other equations in (1 27) as well as in equation (1 26) Equation (1.26) will be referred to as the objective function equation.

We will now present the general simplex method. A computer –oriented algorithm will be followed to carry out this method

Step1: Obtain an initial basic feasible solution of the equivalent model; that is, let x_{Bi}, i=1,2, ……. … . ,m

Be the initial basic feasible solution where x_{Bi} denotes the i^{th} basic variable and corresponds to the slack or artificial variable in the i^{th} equation.

Step2: Select the non-basic variable with the most negative coefficient in the equation (1.26) as the variable to enter as a basic variable in the new basic feasible solution. If all coefficients in equation (1.26) are non-negative, an optimal solution of the original model only if the basic variables are void of any artificial variables with a positive value. That is to say, if at least one basic variable is an artificial variable with a positive value in the optimal solution of equivalent model, then there are no feasible solutions of the original model.

Step3: Select a basic variable to leave the set of variables that are present in the current basic feasible solution. The basic variable in the equation corresponding to the minimum ratios of the b_i's to the corresponding positive coefficient of entering variable in each equation represented by (1.27) will leave and not be a part of the next basic feasible solution. Let equation r contains the leaving variable. If there are no non-negative ratios, then the objective function is unbounded above (That is no finite optimal solution exists)

Step 4: Perform elementary transformations on equation (1.26) and (1.27) until the coefficient of the entering variable from step 2 is one in equation r and zero in every other equation including equation (1 26). This can be accomplished by the Gauss-Jordan elimination method for solving a system of linear equations The new basic feasible solution is $x_{Bi} = b_i^*$, $i = 1,2,\ldots\ldots,m$ Where x_{Bi} $(i \neq r)$ corresponds to the same basic variables in the previous basic feasible solution and x_{Br} corresponds to the new basic variable that just entered the basic solution.

Step 5: Let equation (1.26) and (1 27) now represent the transformed system of linear equations from Step 5 Return to step 2.

Properties of the Simplex Method:

The important properties of the simplex method are summarized here for convenient ready reference.

1) The simplex method for maximizing the objective function starts at a basic feasible solution for the equivalent model and moves to an adjacent basic feasible solution that does not decrease the value of the objective function. If such a solution does not exist, an optimal solution for the equivalent model has been reached That is, if all of the coefficients of the non-basic variables in the objective function equation are greater than or equal to zero at some point, then an optimal solution for the equivalent model has been reached.
2) If an artificial variable is in an optimal solution of the equivalent model at a non-zero level, then no feasible solution for the original model exists. On the contrary, if the optimal solution of the equivalent model does not contain an artificial variable at a non-zero level, the solution is also optimal for the original model
3) If all of the slack, surplus, and artificial variables are zero when an optimal solution of the equivalent model is reached, then all of the constraints in the original model are strict "equalities" for the values of the variables that optimize the objective function
4) If a non-basic variable has zero coefficients in the objective function equation when an optimal solution is reached, there are multiple optimal solutions. In fact, there is infinity of optimal solutions The simplex method finds only one optimal solution and stops.
5) Once an artificial variable leaves the set of basic variables (the basic), it will never enter the basis again. So all calculations for that variable can be ignored in future steps.

6) When selecting the variable to leave the current basis

   a) If two or more ratio is smallest, choose one arbitrarily.

   b) If a positive ratio does not exist, the objective function in the original model is not bounded by the constraints. Thus, a finite optimal solution for the original model does not exist

7) If a basis has a variable at the zero level, it is called a degenerate basis.

8) Although cycling is possible, there have never been any practical problems for which the simplex method failed to converge.

Linear Fractional Program (LFP) :

Recently various optimization problems, involving the optimization of the ratio of functions, e. g; time/cost, volume/cost, profit/cost, loss/cost or other quantities measuring the efficiency of the system have been the subject of wide interest in non-linear programming problem. Such problems are known as LFP.

If the objective function of a mathematical programming problem is the ratio of two linear functions and the constraints are linear, it is called a linear fractional programming problem, or LFP. Likewise LP, a standard LFP can be expressed as follows

$$(LFP) \qquad Maximize \quad F(x) = \frac{c^T x + \alpha}{d^T x + \beta} \tag{1.29}$$

Subject to

$$x \in X = \{x \in R^n \,.\, Ax = b,\, x \geq 0\} \tag{1.30}$$

Where x, c, d $\in R^n$; b$\in R^m$; $\alpha, \beta \in R$; A is an $m \times n$ matrix and superscript T denotes transpose

For simplicity of notation, throughout this chapter and hence forth, we can omit the transpose sign T over vectors In an inner product of two vectors, one can assume that the left hand side vectors be a row vector and right side vector be a column vector.

Now a day, linear fraction criteria are frequently encountered in business and economics such as

Corporate Planning

Min [debt –to-equity ratio]

Max [return on investment]

Max [out put per employee]

Min [actual cost –to-standard cost]

Bank Balance Sheet Management

Min [risk assets –to-capital]

Max [actual capital –to-required capital]

Min [foreign loans –to-total loans]

Min [residential mortgages –to-total mortgages]

Linear fractional objective also occur in other areas of science, engineering and social sciences .Now we consider a real life problem

1.5 A Production Problem of a certain agricultural farm:

Suppose a farmer has 1000000/= taka by which he can cultivate maximum 50 hectors of land. The farmer wishes to cultivate different crops (Rice, Wheat, Jute, Potatoes, Pulse, Maize, Mustard seed, Tomatoes, Brinjal , onion, Cauliflower , Cabbages and Beans). He has the following data for per hector

Name of Crops	Cost of seeds	Fertilizer cost	Irrigation cost	Pest Management	Cultivation cost	Labour cost	Return
Rice	375	4260	4500	500	1500	1500	25200
Wheat	1500	4660	1500	400	2000	1500	23000
Jute	210	1580	700	800	2000	1800	14000
Potatoes	22500	6050	1500	600	1500	1200	66000
Pulse	1000	1780	700	800	1200	1500	13800
Maize	400	5960	1500	400	1200	1500	21700
Mustard seed	500	5840	700	400	1200	1500	20100
Tomatoes	500	11870	3000	800	2000	1500	39220
Brimjal	500	6130	3500	1000	1500	1500	28000
Onion	7000	6825	1000	200	1500	1800	36350
Cauliflower	1000	6550	4000	600	1500	2000	31180
Cabbage	1000	7445	3500	500	1500	2000	30000
Beans	200	4025	1000	200	1500	1500	16800

In addition the farmer has the following limitations of expenditures:

Maximum investment for seeds is taka 135000/=

Maximum investment for fertilizer is taka 286000/=

Maximum investment for irrigation is taka 115000/=

Maximum investment for pest management is taka 30000/=

Maximum investment for cultivation is taka 95000/=

Maximum investment for labor is taka 100000/=

And the farmer has a fixed expenditure taka 5000/=.

The objective is to maximize the ratio of **return to investment**. This leads to a LFP.

Formulation:

The three basic steps in constructing a LFP model are as follows:

Step1: Identify the unknown variables to be determined (decision variables) and represent them in terms of algebraic symbols.

Step 2: Identify all the restrictions or constraints in the problem and express them as linear equations or inequalities, which are linear functions of the unknown variables.

Step 3: Identify the objective or criterion and represent it as a ratio of two linear functions of the decision variables, which is to be maximized (or minimized)

Now, we shall formulate above problem as follows:

Step 1: (Identify the Decision variables)

For this problem the unknown variables are the hectors of lands planted for different crops. So,

let x_1 = The hectors of land planted for Rice

x_2 = The hectors of land planted for Wheat

x_3 = The hectors of land planted for Jute

x_4 = The hectors of land planted for Potatoes

x_5 = The hectors of land planted for Pulse

x_6 = The hectors of land planted for Maize

x_7 = The hectors of land planted for Mustard seed

x_8 = The hectors of land planted for Tomatoes

x_9 = The hectors of land planted for Brinjal

x_{10} = The hectors of land planted for Onion

x_{11} = The hectors of land planted for Cauliflower

x_{12} = The hectors of land planted for Cabbage

and x_{13} = The hectors of land planted for Beans

Step 2: (Identify the Constraint)

In this problem constraints are the limited availability of fund for different purposes as follows:.

1. Since the farmer wishes to cultivate maximum 50 hectors of land, so we have

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} \leq 50$$

2 Since the farmer has Maximum investment for seeds is taka 135000/=, so we have

$$375x_1 + 1500x_2 + 210x_3 + 22500x_4 + 1000x_5 + 400x_6 + 500x_7 + 500x_8 + 500x_9 + 7000x_{10}$$
$$+1000x_{11} + 1000x_{12} + 200x_{13} \leq 135000$$

3. Since maximum investment for fertilizer is taka 286000/=, so we have

$$4260x_1 + 4660x_2 + 1580x_3 + 6050x_4 + 1780x_5 + 5960x_6 + 5840x_7 + 11870x_8 + 6130x_9$$
$$+ 6825x_{10} + 6550x_{11} + 7445x_{12} + 4025x_{13} \le 286000$$

4 Since the farmer has Maximum investment for irrigation is taka 115000/=, so we have

$$4500x_1 + 1500x_2 + 700x_3 + 1500x_4 + 700x_5 + 1500x_6 + 700x_7 + 3000x_8 + 3500x_9$$
$$+ 1000x_{10} + 4000x_{11} + 3500x_{12} + 1000x_{13} \le 115000$$

5. Since the farmer has Maximum investment for Pest management is taka 30000/=, so we have

$$500x_1 + 400x_2 + 800x_3 + 600x_4 + 800x_5 + 400x_6 + 400x_7 + 800x_8 + 1000x_9$$
$$+ 200x_{10} + 600x_{11} + 500x_{12} + 200x_{13} \le 30000$$

6 Since the farmer has Maximum investment for Cultivation cost is taka 95000/=, so we have

$$1500x_1 + 2000x_2 + 2000x_3 + 1500x_4 + 1200x_5 + 1200x_6 + 1200x_7 + 2000x_8 + 1500x_9$$
$$+ 1500x_{10} + 1500x_{11} + 1500x_{12} + 1500x_{13} \le 95000$$

7. Since the farmer has Maximum investment for labour is taka 100000/=, so we have

$$1500x_1 + 1500x_2 + 1800x_3 + 1200x_4 + 1500x_5 + 1500x_6 + 1500x_7 + 1500x_8 + 1500x_9$$
$$+ 1800x_{10} + 2000x_{11} + 2000x_{12} + 1500x_{13} \le 100000$$

We must assume that the variables x_i, $i = 1, 2, \ldots, 13$ are not allowed to be negative. That is, we do not make negative quantities of any product.

Step 3: (Identify the objective)

In this case, the objective is to maximize the ratio of total return and investment by different crops. That is,

$$Max\,F(x) = \frac{\begin{array}{c}25200x_1 + 23000x_2 + 14000x_3 + 66000x_4 + 13800x_5 + 21700x_6 + 20100x_7 \\ + 39220x_8 + 28000x_9 + 36350x_{10} + 31180x_{11} + 30000x_{12} + 16800x_{13}\end{array}}{\begin{array}{c}5000 + 12635x_1 + 11560x_2 + 7090x_3 + 33350x_4 + 6980x_5 + 10960x_6 + 10140x_7 \\ + 19670x_8 + 14130x_9 + 18325x_{10} + 15650x_{11} + 15945x_{12} + 8425x_{13}\end{array}}$$

Now, we have expressed our problem as a mathematical model. Since the objective function is the ratio of return to investment and all of the constraints functions are linear , the problem can be modeled as the following LFP model:

$$Max\,F(x)=\frac{\begin{array}{c}25200x_1+23000x_2+14000x_3+66000x_4+13800x_5+21700x_6+20100x_7\\+39220x_8+28000x_9+36350x_{10}+31180x_{11}+30000x_{12}+16800x_{13}\end{array}}{\begin{array}{c}5000+12635x_1+11560x_2+7090x_3+33350x_4+6980x_5+10960x_6+10140x_7\\+19670x_8+14130x_9+18325x_{10}+15650x_{11}+15945x_{12}+8425x_{13}\end{array}}$$

Subject to

$$x_1+x_2+x_3+x_4+x_5+x_6+x_7+x_8+x_9+x_{10}+x_{11}+x_{12}+x_{13}\leq 50$$

$$375x_1+1500x_2+210x_3+22500x_4+1000x_5+400x_6+500x_7+500x_8+500x_9$$
$$+7000x_{10}+1000x_{11}+10x_{12}+200x_{13}\geq 135000$$

$$4260x_1+4660x_2+1580x_3+6050x_4+1780x_5+5960x_6+5840x_7+11870x_8$$
$$+6130x_9+6825x_{10}+6550x_{11}+7445x_{12}+4025x_{13}\leq 286000$$

$$4500x_1+1500x_2+700x_3+1500x_4+700x_5+1500x_6+700x_7+3000x_8$$
$$+3500x_9+1000x_{10}+4000x_{11}+3500x_{12}+1000x_{13}\leq 115000$$

$$500x_1+400x_2+800x_3+600x_4+800x_5+400x_6+400x_7+800x_8$$
$$+1000x_9+200x_{10}+600x_{11}+500x_{12}+200x_{13}\leq 30000$$

$$1500x_1+2000x_2+2000x_3+1500x_4+1200x_5+1200x_6+1200x_7+2000x_8$$
$$+1500x_9+1500x_{10}+1500x_{11}+1500x_{12}+1500x_{13}\leq 95000$$

$$1500x_1+1500x_2+1800x_3+1200x_4+1500x_5+1500x_6+1500x_7+1500x_8$$
$$+1500x_9+1800x_{10}+2000x_{11}+2000x_{12}+1500x_{13}\leq 100000$$
$$x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10},x_{11},x_{12},x_{13}\geq 0$$

Thus the given problem has been formulated as a LFP We will solve this formulated problem by using different methods

1.6 Summary of the thesis

In this thesis, we study the established methods of Charnes-Cooper's [1962], Bitran-Novaes [1972] and Swarup [1964 & 1965] for solving LFP problem , suggest a modification for Swarup [1964] simplex type method and compare the methods among themselves. But to apply these methods on large- scale LFP problem, we need computer- oriented program of these methods. To fulfill this purpose, we develop computer programs (FORTRAN) of these methods and solve a sizable large-scale return to investment problem, which is formulated in section 1.5. The method of Charnes-Cooper's [1962], Bitran-Novaes [1972] and Swarup [1964 & 1965] are briefly presented in chapter- 2, chapter- 3 and chapter- 4 respectively. Further in section 4.3, we suggest a modified approach of Swarup [1964] simplex type method. In chapter-5, a comparative study is made of the above methods on the base of Islam & Nath [1992] investigations. Finally, conclusion is drawn in favour of our modified approach of Swarup's primal simplex type method.

CHAPTER- 2

CHARNES & COOPER METHOD

In this chapter, we shall discuss briefly the method of Charnes & Cooper [1962] for solving LFP problem defined by (1.29) and (1.30) and develop computer program (FORTRAN) for this method.

Considering all possible cases relating to the sign of denominator of objective function of LFP problem, Islam & Nath [1992] obtained some independent results to investigate how Charnes & Cooper [1962] method can be applied for solving LFP problem

The Summary of the Method

Charnes & Cooper [1962]considered the LFP problem defined by (1.29) and (1 30). They also assumed that-

1) The feasible region X is non-empty and bounded,

2) $c\,x+\alpha$ and $d\,x+\beta$ do not vanish simultaneously in X

Introducing the variable transformation $y = t\,x$, Where $t \geq 0$,Charnes & Cooper [1962] proved that LFP problem is reduced to either of the following two Equivalent Linear Programs (ELPs)

EP) *Maximize* $Z_1 = cy + \alpha t$

Subject to

$$Ay - bt = 0$$

$$dy + \beta t = 1$$

$$y, t \geq 0.$$

And

(EN) *Maximize* $Z_2 = -cy - \beta t$

Subject to

$$Ay - bt = 0$$

$$dy + \beta t = -1$$

$$y, t \geq 0.$$

Then they used the well-known Dantzig [1960] simplex method to solve either Equivalent Positive (EP) or Equivalent Negative (EN) problem.

If one of the problems EP and EN has an optimal solution (y^*, t^*) and the other is inconsistent, then the LFP problem also has an optimal solution $x^* = y^* / t^*$ If any one of the two problems is unbounded, then the LFP problem is unbounded. Therefore, if the problem first is found unbounded, one can avoid solving the other

Remark 2.1

It should be observed that the same reduction can be made using the numerator instead of denominator

Since, $\max \dfrac{c^T x + \alpha}{d^T x + \beta} = \max (-1) \dfrac{d^T x + \beta}{c^T x + \alpha}$

Remark 2.2

Thus, if one knows the sign of either the numerator or the denominator of the objective function, one need only solve a single ordinary linear programming problem.

Though Charnes & Cooper [1962]discussed some cases relating to the sign of denominator of objective function , it seems that they did not exhaust all cases . Next Islam & Nath [1992] considered the following six cases covering all possibilities of the sign of denominator $dx+\beta$ of LFP problem over the feasible region X and obtained some independent results to investigate how Charnes & Cooper [1962] method can be applied for solving LFP problem

Then they discussed from **CASE I** to **CASE VI** indetails relating to the sign of the denominator $dx+\beta$ of the objective function LFP problem and obtained the following results .

CASE I: $dx+\beta>0,\ \forall\ x\in X$

Theorem:2.3.1: If $dx+\beta>0$ for all x belongs to X, then

1) EP has an optimal solution (y^*, t^*) *and* EN is inconsistent ;

2) LFP has an optimal solution $x^* = y^*/t^*$.

CASE II: $dx+\beta<0$, for all x belongs to X

Theorem 2.3.2: If $dx+\beta<0$, for all x belongs to X , then

1) EN has an optimal solution (y^*, t^*) and EP is inconsistent ;

2) LFP has an optimal solution $x^* = y^*/t^*$.

CASE III: $dx+\beta=0$, for all x belongs to X.

Theorem 2.3.3: If : $dx + \beta = 0, \forall x \in X$, then

1) Both EP and EN are inconsistent,

2) LFP problem is undefined.

In this case the objective function of LFP problem becomes undefined and thus the question of solving a problem does not arise

CASE IV: $dx+\beta \geq 0$, for all x belongs to X.

Theorem 2.3.4: Let P be a non-empty sub set of X such that $dx + \beta = 0, \forall x \in P$ and $dx + \beta > 0, \forall x \in X - P$ .If

a) $cx + \alpha > 0, \forall x \in P$, then

1) EP is unbounded and EN is inconsistent;

2) LFP is unbounded

b) $cx + \alpha < 0, \forall x \in X$, then

1) EP has a finite optimal solution (y^*, t^*) and EN is inconsistent,

2) LFP problem has a finite optimal solution $x^* = y^*/t^*$

CASEV: $dx+\beta \leq 0$, for all x belongs to X.

Theorem 2.3.5: Let P be a non-empty sub set of X such that $dx + \beta = 0, \forall x \in P$ and $dx + \beta < 0, \forall x \in X - P$. If

a) $cx+\alpha<0, \forall x\in P$, then

1) EN has an optimal solution (y^*, t^*) and EP is inconsistent ,

2) LFP problem has an optimal solution $x^* = y^*/t^*$.

b) $cx+\alpha<0, \forall x\in X$, then

1) EN is unbounded and EP is inconsistent,

2) LFP problem is unbounded

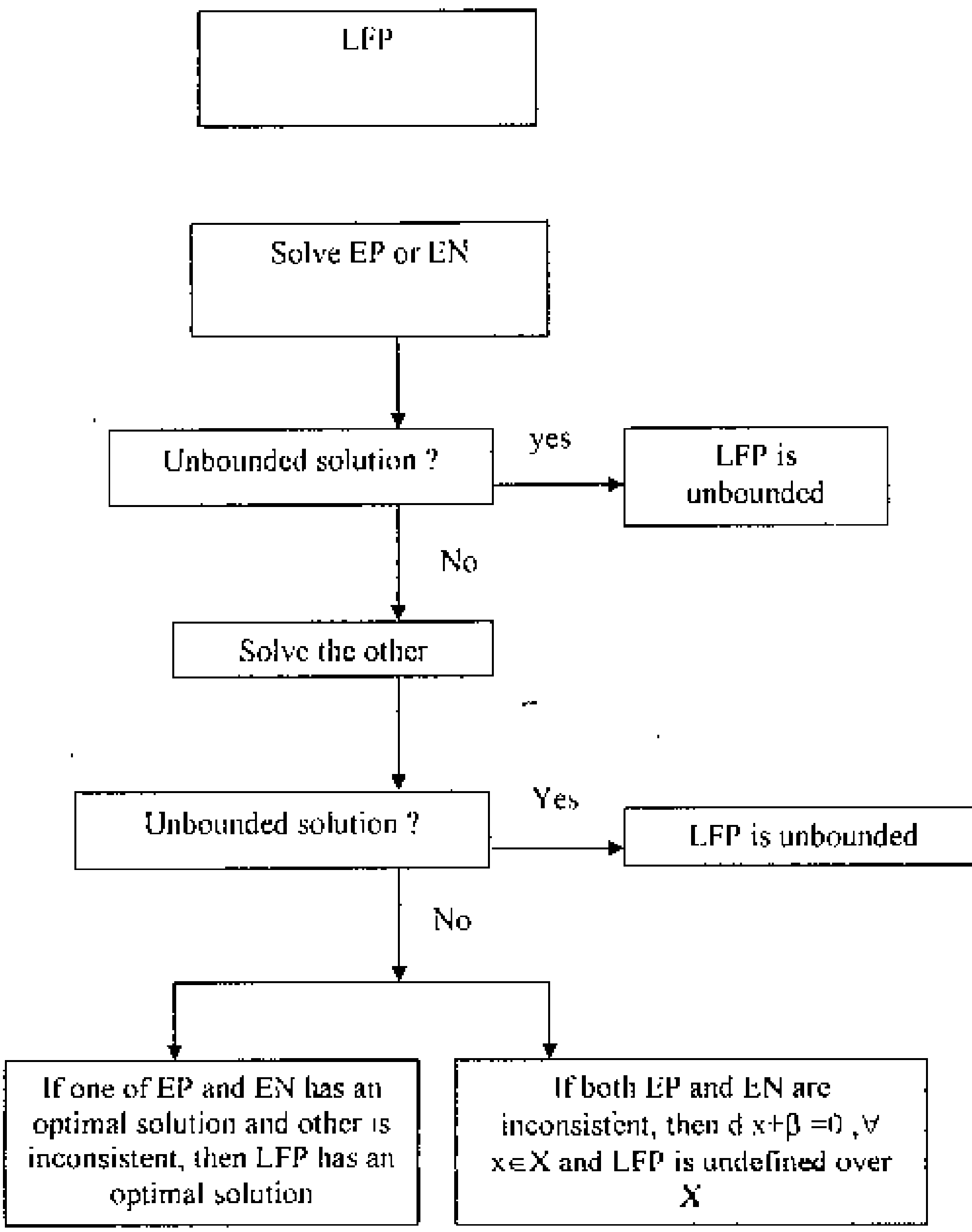
CASEVI: $dx+\beta$ changes sign over X.

Theorem 2.3.5: If $dx+\beta$ changes sign over X, then

1) Either EP or EN is unbounded and other has optimal solution;

2) LFP problem is unbounded.

The solution procedure for LFP problem applying Charnes & Cooper [1962] technique can be summarized in the following diagram



If the sign of the denominator $dx + \beta$ is known over X, the above discussion shows that one can solve LFP problem by solving either EP or EN But in reality, it is rather impossible to know the sign of the denominator $dx+\beta$ over X Since LFP problem can be solved by solving at most two linear programs EP & EN. So for solving LFP problem one must proceed in reverse order. If one of the problems EN & EN has an optimal solution and other is inconsistent, then LFP problem also has an optimal solution. If anyone of the two

Problems EP & EN is unbounded, and then LFP problem is also unbounded. Thus if the problem solved first is unbounded, one need not to solve the other.

We now wish to present Fortran computer program of the method as follows:

Fortran Program for Charnes & Cooper [1962] transformation technique.

```
*****************************************************************************
C    * PROGRAM FOR Charnes & Cooper TECHNIQUE
C    * M   NUMBER OF CONSTRAINTS
C    * N   NUMBER OF VARIABLES
C    * m1  NUMBER OF LESS THAN OR EQUAL TYPE CONSTRAINTS
C    * m2  NUMBER OF GREATER THAN OR EQUAL TYPE CONSTRAINTS
C    * m3  NUMBER OF EQUAL TYPE CONSTRAINTS
C    * ICASE  0 OPTIMAL SOLUTION IS FOUND
C    * ICASE -1 INCONSISTENT SOLUTION IS FOUND
C    * ICASE  1 UNBOUNDED SOLUTION IS FOUND
C    * A(i,j)COEFFICIENT MATRIX OF EP.
C    * B(i,j)COEFFICIENT MATRIX OF NP.
*****************************************************************************
      Parameter(M=3,N=3)
      Real a(M+2,N+1), b(M+2,N+1)
      Integer np,mp,m1,m2,m3,icase,izrov(N),iposv(M),x(N+1)
      Open(1,file='C14.dat')
      Open(2,file='C13.dat')
            Read(1,*) m1,m2,m3
      mp=M+2
      np=N+1

      Read(1,*) ((a(i,j),j=1,N+1),i=1,M+2)
       Read(1,*) ((b(i,j),j=1,N+1),i=1,M+2)


       Call simplx(a,M,N,mp,np,m1,m2,m3,icase,izrov,iposv)

        Write(2,*) " The left hand variables are:"
        Write(2,*) (iposv(j),j=1,M)
         Write(2,*) " The right hand variables are:"
        Write(2,*) (izrov(i),i=1,N)

        Write(2,*) " The value of the icase:"
        write(2,*) icase
       if(icase.eq.1)then
                   GO TO 3
```

```
                 ENDIF
      Write(2,88)((a(i,j),j=1,np),i=1,mp)
88    Format(1x,4(6x,f8.5))


      Call simplx(b,M,N,mp,np,m1,m2,m3,icase,izrov,iposv)
      Write(2,*) " The left hand variables are:"
      Write(2,*) (iposv(j),j=1,M)
       Write(2,*) " The right hand variables are:"
      Write(2,*) (izrov(i),i=1,N)

      Write(2,*) " The value of the icase:"
      write(2,*) icase

      Write(2,78)((b(i,j),j=1,np),i=1,mp)
 78   Format(1x,4(6x,f8.5))
 3    Stop
      End


  Subroutine simplx(a,m,n,mp,np,m1,m2,m3,icase,izrov,iposv)
  Integer icase,m,m1,m2,m3,mp,n,np,izrov(n),
        iposv(m),MMAX,NMAX
      Real a(mp,np), EPS
      Parameter (MMAX=100, NMAX=100, EPS=.0001)
      Integer i, ip,is,k,kh,kp,nL1,L1(NMAX),L3(MMAX)
      Real bmax,q1
 if(m.ne.m1+m2+m3) pause 'bad input constraint counts in
simplex'
      nL1=n
      do 11 k=1,n

      L1(k)=k
      izrov(k)=k
 11     continue
      do 12 i=1,m
if(a(i+1,1).lt.0)   pause 'bad input constraint counts in
simplex'
        iposv(i)=n+i
12    continue

        if(m2+m3.eq.0) go to 30
        do 13  i=1,m2

        L3(i)=1
13    continue
```

```
      do 15 k=1,n+1
      q1=0
      do 14 i=m1+1,m
      q1=q1+a(i+1,k)
14    continue

      a(m+2,k)=-q1
15    continue

10    call simp1(a,mp,np,m+1,L1,nL1,0,kp,bmax)
      if(bmax.le.EPS.and.a(m+2,1).le.- EPS)then

      icase=-1
      return
      else if(bmax.le.EPS.and.a(m+2,1).lt.EPS)then
      do 16 ip=m1+m2+1,m
      if(iposv(ip).eq.ip+n)then
       call simp1(a,mp,np,ip,L1,nL1,1,kp,bmax)
       if(bmax.gt.EPS) go to 1

       endif
16    continue

       do 18 i=m1+1,m1+m2,1
       if(L3(i-m1).eq.1) then

       do 17 k=1,n+1
       a(i+1,k)=-a(i+1,k)
17    continue
        else
       endif
18    continue
       go to 30
       endif

      call simp2(a,m,n,mp,np,ip,kp)
      if (ip.eq.0) then

      icase=-1
      return
      endif

1     call simp3(a,mp,np,m+1,n,ip,kp)
       if(iposv(ip).ge.n+m1+m2+1) then

       do 19 k=1,nL1
       if(L1(k).eq.kp) go to 2
```

```
19      continue

2          nL1=nL1-1
           do 21 is=k,nL1
           L1(is)=L1(is+1)
21      continue

           else
           kh=iposv(ip)-m1-n
           if(kh.ge.1) then
           if(L3(kh).ne.0)then

           L3(kh)=0
           a(m+2,kp+1)=a(m+2,kp+1)+1
            do 22 i=1, m+2
               a(i,kp+1)= -a(i,kp+1)
 22      continue

              endif
              endif
              endif
              is=izrov(kp)
              izrov(kp)=iposv(ip)
              iposv(ip)=is
              go to 10

 30      call simpl(a,mp,np,0,L1,nL1,0,kp,bmax)
             if(bmax.le.EPS) then

               icase=0
               return
               endif

       call simp2(a,m,n,mp,np,ip,kp)
             if(ip.eq.0) then

               icase=1
               return
               endif

      call simp3(a,mp,np,m,n,ip,kp)
                is=izrov(kp)
                izrov(kp)=iposv(ip)
                iposv(ip)=is
                go to 30
                end
```

```
C************************************************************
      Subroutine simp1(a,mp,np,mm,LL,nLL, iabf,kp,bmax)
C     Determines the pivot column
C************************************************************
      Integer iabf,kp,mm,mp,nLL,np,LL(np)
      Real   bmax, a(mp,np)
      Integer k
      Real test
      if(nLL.le.0)then

      bmax=0
      else
      kp=LL(1)
      bmax=a(mm+1,kp+1)
      do 11 k=2,nLL
      if(iabf.eq.0)then

      test=a(mm+1,LL(k)+1)-bmax
      else
      test=abs(a(mm+1,LL(k)+1))-abs(bmax)
      endif
      if(test.gt.0)then

      bmax=a(mm+1,LL(k)+1)
      kp=LL(k)
      endif
 11   continue

      endif
      return
      end


C********************************************************
      Subroutine simp2(a,m,n,mp,np,ip,kp)
C     Determines pivot element
C********************************************************
      Integer  ip,kp,m,mp,n,np
       Real a(mp,np), EPS
       Parameter (EPS=.0001)

       Integer i,k
       Real q,q0,q1,qp
       ip=0
```

```
        do 11 i=1,m
        if(a(i+1,kp+1).lt.-EPS) go to 1
 11   continue

        return
1        q1=-a(i+1,1)/a(i+1,kp+1)
        ip=i
        do 13 i=ip+1,m

        if(a(i+1,kp+1).lt.-EPS)then
            q=-a(i+1,1)/a(i+1,kp+1)

          if(q.lt.q1)then

          ip=i
          q1=q
          elseif(q.eq.q1) then
          do 12 k=1,n
           qp=-a(ip+1,k+1)/a(ip+1,kp+1)
           q0=-a(i+1,k+1)/a(i+1,kp+1)
           if(q0.ne.qp)go to 2

12       continue

2           if(q0.lt.qp)ip=i

           endif
           endif
13     continue
           return
           end

c*****************************************************
        Subroutine simp3(a,mp,np,i1,k1,ip,kp)
C       Matrix operations to exchange a left-hand and
C        right-hand variable
C*****************************************************
       Integer  mp,np,i1,k1,ip,kp
         Real a(mp,np)
          Integer ii,kk
          Real v

        v=1./a(ip+1,kp+1)
        do 12 ii=1, i1+1
        if(ii-1.ne.ip)then
```

```
         a(ii,kp+1)=a(ii,kp+1) *v
         do 11 kk=1,kl+1
         if(kk-1.ne.kp)then

         a(ii,kk)=a(ii,kk)-a(ip+1,kk)*a(ii,kp+1)
         endif
11      continue

         endif
12      continue

         do 13 kk=1,kl+1
         if(kk-1.ne.kp)a(ip+1,kk)=-a(ip+1,kk)*v

13      continue

         a(ip+1,kp+1)=v
         return
         end
```

Now, we solve the following numerical examples of by using the above program

Example 2.1

(LFP) Maximize $Z = \dfrac{-24x_1 - 7}{5x_1 + x_2 + 1}$

Subject to

$$-x_1 + x_2 \le 1$$
$$x_1 - x_2 \le 1$$
$$x_1 + x_2 \le 2$$
$$x_1, \quad x_2 \ge 0$$

The equivalent linear program (ELPs) of the above LFP problem is obtained by setting

$$y_i = tx_i$$

where i=1,2 , $t \ge 0$ as follows.

(EP) Maximize $Z_1 = -24y_1 - 7t$

Subject to

$$\begin{aligned} -y_1 + y_2 - t &\le 0 \\ y_1 - y_2 - t &\le 0 \\ y_1 + y_2 - 2t &\le 0 \\ 5y_1 + y_2 + t &= 1 \\ y_1, y_2, t &\ge 0 \end{aligned}$$

(EN) Maximize $Z_2 = 24y_1 + 7t$

Subject to

$$\begin{aligned} -y_1 + y_2 - t &\le 0 \\ y_1 - y_2 - t &\le 0 \\ y_1 + y_2 - 2t &\le 0 \\ 5y_1 + y_2 + t &= -1 \\ y_1, y_2, t &\ge 0 \end{aligned}$$

Now, applying the above program to solve EP & EN, we have obtained the following data:

For EP:

The left hand variables are (basic variable)
2 5 6 3
The right hand variables are (non-basic variable):
4 1 7
The value of the icase 0 (Optimal solution is found)

-3.50000100	-3.50000100	-9.99999700	-3.50000100
.50000010	-.50000000	-3 00000000	- 50000010
1 00000000	.00000007	-6 00000000	-1 00000000
.50000010	1.50000000	-2.00000000	-.50000010
.50000010	.50000010	-2.00000000	-.50000010
00000000	.00000000	00000000	-1.00000000

For EN:

The left hand variables are

4 5 6 7

The right hand variables are

1 2 3

The value of the icase: -1 (Inconsistent solution is found)

.00000000	24.00000000	00000000	.00000000
.00000000	1.00000000	-1 00000000	1 00000000
.00000000	-1.00000000	1.00000000	.00000000
.00000000	-1.00000000	-1.00000000	.00000000
1 00000000	5.00000000	1 00000000	1.00000000
-1.00000000	-5.00000000	-1 00000000	-1.00000000

The solution of the above **Example 2.1** is:

$y_1 = 0$, $y_2 = 0.5$, $t = 0.5$ and this implies

$x_1 = 0.0$ $x_2 = 1.0$ with Zmax = -3.5

Example 2.2

(LFP) Maximize $Z = \dfrac{5x_1 + 4x_2}{-x_1 - x_2 + 1}$

Subject to

$$-x_1 + 2x_2 \le 1$$
$$x_1 + 3x_2 \le 1$$
$$x_1, \quad x_2 \ge 0$$

Similarly, applying the above program, we have obtained the following data

For EP:

The left hand variables (basic variable) are:

4 2 3

The right hand variables (non-basic variable) are:

1 5 6

The value of the icase 1 (unbounded solution)

For EP.

Since EP is unbounded, we need not to solve EN.

Therefore, the solution of the above **Example 2.2** is unbounded

CHAPTER - 3

BITRAN & NOVAES METHOD

In this chapter, we discuss briefly the method developed by Bitran & Novaes [1972] for solving LFP problem defined by .

$$\text{(LFP)} \qquad \text{Maximize } F(x) = \frac{cx + \alpha}{dx + \beta} \tag{3.1}$$

$$\text{Subject to} \qquad Ax = b \tag{3.2}$$

$$x \geq 0 \tag{3.3}$$

where x , c, d $\in R^n$; $b \in R^m$; α & $\beta \in R$; A is an $m \times n$ matrix , and generalize computer program (FORTRAN) of the method

Assuming the constraint set non –empty and bounded and the denominator of the objective function of LFP problem is positive for all feasible solutions, Bitran & Novaes [1972] method with validity of their results with an illustrative example is presented next section

Bitran & Novaes Method

Bitran & Novaes [1972]considered LFP problem defined by (3 1) to (3.3) assuming the positivity of the denominator of the objective function

They developed the method that can be stated as follows

Step I : First they introduced a new objective function as follows:

$$\textit{Maximize} \quad L = \langle \gamma, x \rangle$$

$$\text{Subject to} \quad Ax = b$$

In which $\gamma = c - [\langle c,d \rangle \ / \ \langle d,d \rangle \ ] d$

And hence they solved the LPP problem applying simplex method which yields a sub optimal solution x^i .

Step II : Again , they introduced another new linear objective function L' as follows:

(LP) Maximize $L' = \langle [c - F(x^i) d \,] \, x \rangle$ (3.4)

Subject to the same set of constraints and hence then solve as before. This leads to another new sub optimal feasible solution x^{i+1}

Step III : Compare x^i with x^{i+1} ; if $x^i = x^{i+1}$, then x^i is the global optimal solution , otherwise go to step II, making $x^i = x^{i+1}$ and repeating the process until the vector x^i remains unchanged.

Next, they discussed they validity of the method as follows:

Validity of the Method

Property 3.1: $(\alpha - F\beta \ ) + \langle (c - Fd \ ), x \rangle = 0$

represents a family of hyperplanes of order n that have a common subset of order n-1.

Property 3.2 : The hyperplane $\alpha + \langle c, x \rangle = 0$ contains S the sub set of order n-1 common to all hyperplaners that satisfy the relation defined by (3.1) to (3.2).

Property 3. 3 : The hyperplane $\beta + \langle d, x \rangle = 0$ contains S the sub set of order n-1 common to all hyperplanes that satisfy the relation defined by (3.1) to (3 2).

Property 3.4 : The hyperplane $\beta + \langle d, x \rangle = 0$ does not intercept the positive orthant.

Property 3.5 : If x_0 is a point that belongs to a particular hyperplane of the family given by (3.1) and such that $\beta + \langle d, x \rangle \neq 0$, then the gradient of $F(x_0)$ is orthogonal to this hyperplane at point x_0.

Property 3.6: Since for all feasible solutions, the relation $x \geq 0$ must always hold ,it has already been shown in property- 4 that $[\beta + \langle d, x \rangle\ ] > 0$ therefore the gradient $(\nabla F)_{x_0}$ has the same sign as vector $[c - F(x_0)]$

Property 3.7: Vector $[c - F(x_0)]$ has the same directions as $(\nabla F)_x$ for any point x such that $F(x) = F(x_0)$.

Property 3.8 : For any $x \geq 0$ and $x_0 \geq 0$ necessary and sufficient condition for having $F(x) > F(x_0)$ is given by $Z > 0$, where

$$Z = [c - F(x_0)\beta] + \langle [c - F(x_0)\, d], x \rangle \qquad (3.5)$$

The solution obtained with the method presented in this section is optimal.
One gets a solution when, in Step III, the simplex leads to a sub optimal point identical to the initial one x'.

The objective function for the simplex in Step II is

$$Max \quad \langle [c - F(x')d], x \rangle \, . \qquad (3.6)$$

with the same set of constraints as in Step I

Suppose now there is a point x^{i+1} such that $F(x^{i+1}) > F(x^i)$. If this is the case, then Property 3.8 yields .

$$Z = \left|\alpha - F(x^i)\beta\right| + \langle \left|c - F(x^i)d\right|, x\rangle \qquad (3\ 7)$$

such that Z >0 .

On the other hand, relations (3.1) to (3.3) yields

$$\left|\alpha - F(x^i)\beta\right| + \langle \left|c - F(x^i)d\right|, x^i \rangle = 0 \qquad (3.8)$$

Subtracting (3.8) from (3 7), one gets,

$$\left\{ \langle \left|c - F(x^i)d\right|, x \rangle \right\} > \left\{ \langle \left|c - F(x^i)d\right|, x^i \rangle \right\} \qquad (3.9)$$

But, if relation (3 9) holds, one can see by looking at the objective function (3.6) that x^{i+1} is a better feasible solution than x^i If this had been happened, one would go back to Step II again until the convergence is attained

Convergence

The simplex used in Step II guarantees that the solution is a vertex of the convex set. Further, Property 3 6 also guarantees that, whenever in Step II the process goes from a vertex x^i to a vertex x^{i+1} , one always has $F(x^{i+1}) > F(x^i)$. This happens because the gradient of $(\nabla F)_x$ has same sign as the objective function given by (3.4). Since we maximize (3 4) , the process moves along in the direction of the gradient . This means that the feasible solutions are always upgraded as long as one applies Step II.

On the other hand, the number of vertexes is finite, which means that one reached the optimal solution (point) within a finite number of steps. Here we develop the computer program (FORTRAN) of Bitran &Novaes [1972] method is as follows.

Fortran Program for Bitran &Novaes [1972] method.

```
************************************************************************

C    * PROGRAM FOR BITRAN & NOVAES METHOD
C    * M    NUMBER OF CONSTRAINTS
C    * N    NUMBER OF VARIABLES
C    * m1 NUMBER OF LESS THAN OR EQUAL TYPE CONSTRAINTS
C    * m2 NUMBER OF GREATER THAN OR EQUAL TYPE CONSTRAINTS
C    * m3 NUMBER OF EQUAL TYPE CONSTRAINTS
C    * ICASE 0 OPTIMAL SOLUTION IS FOUND
C    * ICASE -1 INCONSISTENT SOLUTION IS FOUND
C    * ICASE 1 UNBOUNDED SOLUTION IS FOUND
C    * AA(i,j)COEFFICIENT MATRIX
C    * B(i,j)COEFFICIENT MATRIX AFTER AN ITERATION
C    * C(N+1) NUMERATOR OF THE OBJECTIVE FUNCTION
C    * D(N+1) DENOMINATOR OF THE OBJECTIVE FUNCTION

C*********************************************************************

      Parameter(M=3,N=2,tol=.0001)
      Real aa(M+2,N+1),b(M+2,N+1),c(N+1),d(N+1)
      T,x(N+M+2),z1,z2,sum1,sum2,sum3,sum4,sum33,sum44
      Integer np,mp,m1,m2,m3,icase,izrov(N),iposv(M)
      Open(1,file='b12.dat')
       Open(2,file='bi7.dat')
      Read(1,*) m1,m2,m3
      Read(1,*) a1,a2
      Read(1,*) (c(i),i=2,N+1)
      Read(1,*) (d(j),j=2,N+1)
      sum1=.0
      sum2=.0
       Do 47 i=2,N+1
       sum1=sum1+c(i)*d(i)
 47     sum2=sum2+d(i)**2
        T=sum1/sum2
               aa(1,1)=0.0
             Do 22 i=2,N+1
 22        aa(1,i)=c(i)-T*d(i)

      mp=M+2
      np=N+1
      Read(1,*) ((aa(i,j),j=1,N+1),i=2,M+1)
       Write(2,33) ((aa(i,j),j=1,n+1),i=1,m+1)
 33     Format(1x,14(6x,f8.5))

       Call simplx(aa,M,N,mp,np,m1,m2,m3,icase,izrov,iposv)
```

```
      Write(*,*) " The left hand variables are:"
        Write(*,*) (iposv(j),j=1,M)
        Write(*,*) " The right hand variables are:"
        Write(*,*) (izrov(i),i=1,N)
        Write(*,*) " The value of the icase:"
        write(*,*) icase
        Write(*,78)((aa(i,j),j=i,np),i=1,mp)
 78     Format(1x,3(6x,f8.5))

        Do 31 j=1,M
 31        x(iposv(j))=aa(j+1,1)
         Do 41 k=1,N
41         x(izrov(k))=0.0


       Do 43 i=2,N+1
       sum3=sum3+c(i)*x(i-1)
 43     sum4=sum4+d(i)*x(i-1)
        z1=(sum3+a1)/(sum4+a2)
 21       b(1,1)=0.0
        Do 51 k=N+1,2,-1
 51        b(1,k)=c(k)-(z1*d(k))
            Do 32 i=2,M+1
            b(i,1)=aa(i,1)
 32      Continue
         Do 37 i=2,M+1
         Do 37 j=2,N+1
            b(i,N+3-j)=aa(i,j)
 37      Continue
       Call simplx(b,M,N,mp,np,m1,m2,m3,icase,izrov,iposv)
        Write(*,*) " The left hand variables are:"
        Write(*,*) (iposv(j),j=1,M)
        Write(*,*) " The right hand variables are:"
        Write(*,*) (izrov(i),i=1,N)
        Do 61 i=1,M
 61        x(iposv(i))=b(i+1,1)
         Do 71 j=1,N
 71         x(izrov(j))=0.0

       Do 46 i=2,N+1
       sum33=sum33+c(i)*x(i-1)
 46     sum44=sum44+d(i)*x(i-1)
        z2=(sum33+a1)/(sum44+a2)

         Write(2,*) z2 ,z1
```

```
          z3=abs(z1-z2)
        If(z3.gt.tol)then
        z1=z2
        Do 34 i=2,M+1
        Do 34 j=1,N+1
            aa(i,j)=b(i,j)
 34      Continue
        go to 21
        endif
        Write(2,88)((b(i,j),j=1,np),i=1,mp)
 88     Format(1x,3(6x,f8.5))

      stop
      End

Subroutine simplx(a,m,n,mp,np,m1,m2,m3,icase,izrov,iposv)
Integer icase,m,m1,m2,m3,mp,n,np,izrov(n),iposv(m)
          MMAX,NMAX
        Real a(mp,np), EPS
        Parameter (MMAX=100, NMAX=100, EPS=.0001)
        Integer i,ip,is,k,kh,kp,nl1,L1(NMAX),L3(MMAX)
        Real bmax,q1
        if(m.ne.m1+m2+m3) pause 'bad input constraint
              counts in simplex'
        nl1=n
        do 11 k=1,n

        L1(k)=k
        izrov(k)=k
 11       continue
        do 12 i=1,m
        if(a(i+1,1).lt.0)pause 'bad input constraint
         counts in simplex'
          iposv(i)=n+i
12      continue

          if(m2+m3.eq.0) go to 30
          do 13  i=1,m2

          L3(i)=1
13     continue

          do 15 k=1,n+1
          q1=0
          do 14 i=m1+1,m
          q1=q1+a(i+1,k)
```

```
14      continue
          a(m+2,k)=-q1
 15     continue
 10          call simp1(a,mp,np,m+1,L1,nL1,0,kp,bmax)
          if(bmax.le.EPS.and.a(m+2,1).le.- EPS)then
          icase=-1
          return
          else if(bmax.le.EPS.and.a(m+2,1).lt.EPS)then
          do 16 ip=m1+m2+1,m
          if(iposv(ip).eq.ip+n)then
           call simp1(a,mp,np,ip,L1,nL1,1,kp,bmax)
           if(bmax.gt.EPS) go to 1
           endif
16      continue
           do 18 i=m1+1,m1+m2,1
           if(L3(i-m1).eq.1) then
           do 17 k=1,n+1
           a(i+1,k)=-a(i+1,k)
   17   continue
            else
           endif
   18   continue
           go to 30
           endif
           call simp2(a,m,n,mp,np,ip,kp)
         if (ip.eq.0) then
         icase=-1
         return
         endif
1           call simp3(a,mp,np,m+1,n,ip,kp)
           if(iposv(ip).ge.n+m1+m2+1) then
           do 19 k=1,nL1
           if(L1(k).eq.kp) go to 2
19       continue
2          nL1=nL1-1
           do 21 is=k,nL1
```

```
         L1(is)=L1(is+1)
21      continue

         else
         kh=iposv(ip)-m1-n
         if(kh.ge.1) then
         if(L3(kh).ne.0)then

         L3(kh)=0
         a(m+2,kp+1)=a(m+2,kp+1)+1
          do 22 i=1, m+2
            a(i,kp+1)= -a(i,kp+1)
22      continue

           endif
           endif
           endif
           is=izrov(kp)
           izrov(kp)=iposv(ip)
           iposv(ip)=is
           go to 10

30        call simp1(a,mp,np,0,L1,nL1,0,kp,bmax)
          if(bmax.le.EPS) then

            icase=0
            return
            endif

             call simp2(a,m,n,mp,np,ip,kp)
             if(ip.eq.0) then

             icase=1
             return
             endif

              call simp3(a,mp,np,m,n,ip,kp)
              is=izrov(kp)
              izrov(kp)=iposv(ip)
              iposv(ip)=is
              go to 30
              end
```

```
C*****************************************************************
      Subroutine simp1(a,mp,np,mm,LL,nLL, iabf,kp,bmax)
C     Determines the pivot column
C*****************************************************************

      Integer iabf,kp,mm,mp,nLL,np,LL(np)
      Real   bmax, a(mp,np)
      Integer k
      Real test
      if(nLL.le.0)then

      bmax=0
      else
      kp=LL(1)
      bmax=a(mm+1,kp+1)
      do 11 k=2,nLL
      if(iabf.eq.0)then

      test=a(mm+1,LL(k)+1)-bmax
      else
      test=abs(a(mm+1,LL(k)+1))-abs(bmax)
      endif
      if(test.gt.0)then

      bmax=a(mm+1,LL(k)+1)
      kp=LL(k)
      endif
 11   continue

      endif
      return
      end


C*****************************************************************
      Subroutine simp2(a,m,n,mp,np,ip,kp)
C     Determines pivot element
C*****************************************************************

      Integer  ip,kp,m,mp,n,np
       Real a(mp,np), EPS
       Parameter (EPS=.0001)

       Integer i,k
       Real q,q0,q1,qp
       ip=0
```

```
        do 11 i=1,m
        if(a(i+1,kp+1).lt.-EPS) go to 1
 11     continue

        return

1        q1=-a(i+1,1)/a(i+1,kp+1)
        ip=i
        do 13 i=ip+1,m

        if(a(i+1,kp+1).lt.-EPS)then
            q=-a(i+1,1)/a(i+1,kp+1)

          if(q.lt.q1)then

          ip=i
          q1=q
          elseif(q.eq.q1) then
          do 12 k=1,n
           qp=-a(ip+1,k+1)/a(ip+1,kp+1)
           q0=-a(i+1,k+1)/a(i+1,kp+1)
           if(q0.ne.qp)go to 2

12        continue

2           if(q0.lt.qp)ip=i

           endif
           endif
13      continue

           return
           end

C*****************************************************
        Subroutine simp3(a,mp,np,i1,k1,ip,kp)
C       Matrix operations to exchange a left-hand and
C        right-hand variable
C*****************************************************

       Integer  mp,np,i1,k1,ip,kp
         Real a(mp,np)
          Integer ii,kk
          Real v
```

```
      v=1./a(ip+1,kp+1)
      do 12 ii=1, il+1
      if(ii-1.ne.ip)then

        a(ii,kp+1)=a(ii,kp+1) *v
        do 11 kk=1,kl+1
        if(kk-1.ne.kp)then

        a(ii,kk)=a(ii,kk)-a(ip+1,kk)*a(ii,kp+1)
        endif
11     continue

        endif
12     continue

        do 13 kk=1,kl+1
        if(kk-1.ne.kp)a(ip+1,kk)=-a(ip+1,kk)*v

13    continue

        a(ip+1,kp+1)=v
        return
        end
```

Now, applying the above program to solve the **Production Problem formed in section 1.5 of Chapter-1**, we obtain the following data:

The left hand variables are (basic variables):

2	14	8	1	15	20	18

The right hand variables are (Non-basic variables):

17	4	3	19	5
6	7	11	12	13
16	9	10		

```
The value of the icase:
          0
```

1296630.00000	-1.93697	43305.37000
-1762.65800	-5.74450	2263.61000
894.91930	1066.10300	-1.84668
1283.69000	13192.69000	2719.59700
855.30640	-1256.62300	

656070.90000	-.96538	21957.20000
-851.82430	-2.90308	1175.84200
521.28380	584.46430	-.92399
732.52560	6698.80300	1381.76200
1332.48400	-614.04010	

32.46644	.00020	-.42698
-1.45907	-.00088	-.73864
-.17283	-.34465	.00010
-.01876	-.45188	.12211
.10885	-.72360	

.14448	.00008	-.26217
-.01238	.00048	-.39912
-.40453	-.46478	-.00002
-.10798	-.31357	-.07591
-.12925	-.26829	

7.96701	.00003	-.35953
.42193	.00031	.14129
-.44076	-.44968	-.00015
-.30519	-.48884	-.34983
-.49660	-.08127	

9.42207	-.00031	.04868
.04951	.00008	-.00354
.01811	.25912	.00006
-.56806	.25430	-.69637
-.48299	.07316	

78783.55000	-.20023	-21698.02000
1749.06500	1.12627	38.64083
72.82928	144.65420	-.09704
-106.23910	-6173.11600	-747.10930
-733.84600	898.59700	

25216.72000	.11661	-93.25553
-318.56700	.71907	-598.67360
-606.79270	-697.16960	-.02426
-161.97640	-770.36220	-613.86040
-693.87820	-402.43410	

5928.78100	.04899	-165.92260
-578.67490	.05774	- 615.80900
12.67998	-31.94990	.04529

```
-464.31070          244.67630                     -20.79352
95.23967            117.87700

The maximum value is:

                     1.97636
```

Hence solving the problem, we have obtained the following results.

To obtain maximum return on investment,
the farmer has to plant rice in 9 422207 hectors of land , wheat in 32.46644 hectors of land, tomatoes in 7.96701 hectors of land, and the maximum return on investment is 1.97636.

CHAPTER- 4

SWARUP'S METHODS

In this chapter, we discuss briefly the methods developed by Swarup [1964 & 1965] and we develop computer program (FORTRAN) of these methods. We also suggest a modified approach of Swarup [1964] primal simplex type method for solving LFP problem defined as:

(LFP) $$\text{Maximize } F(x) = \frac{cx+\alpha}{dx+\beta} \tag{4.1}$$

Subject to

$$Ax = b \tag{4.2}$$

$$x \geq 0 \tag{4.3}$$

Where x, c, $d \in R^n$, $b \in R^m$; α & $\beta \in R$; A is an $m \times n$ matrix

We assume the constraint set non-empty and bounded and the denominator of objective function of LFP problem is positive for all feasible solutions.

4.1. Swarup's Primal simplex type method

If the constraint set in the canonical form one can proceed to the initial simplex type table for solving the LFP problem.

Let x_B be the initial basic feasible solution such that

$$Bx_B = b$$

$$\text{or, } x_B = B^{-1} b$$

where $B = (b_1, b_2, \ldots, b_m)$

$$x_B \geq 0$$

Further let $z^1 = c_B x_B + \alpha$

and

$$z^2 = d_B x_B + \beta$$

where c_B and d_B are the vectors having their components as the coefficients associated with the basic variables in the numerator and the denominator of the objective function respectively

Then the value of the objective function for this initial basic feasible solution is $F = \frac{z^1}{z^2}$

In addition, one can assume that for this basic solution

$$\alpha_j = B^{-1} a_j$$

$$z_j^1 = c_B \alpha_j$$
$$z_j^2 = d_B \alpha_j$$

Are known for every column a_j of A not in B.

One can now wish to examine the possibility of finding another basic solution with improved value of $F = \frac{z^1}{z^2}$, he shall confine his attention to those basic feasible solutions in which only one column of B is changed Let x_B^* be the new such basic feasible solution and the new value of the objective function is

$$F^* = \frac{z^{1*}}{z^{2*}}$$

Then

$$x_B^* = B^{-1}b$$

where $\quad B^* = (b_1^*, b_2^*, \ldots, b_m^*)$

i .e . a new non singular matrix B* obtained from B by removing b_r and replacing it by a_j belongs to A not in B The column of the new matrix B* are given by

$$b_i^* = b_i\ , \quad (i \neq \gamma)$$
$$b_\gamma^* = a_j$$

He obtained values of the new basic variables in terms of the original ones and the a_{ij} are

$$x^{*}_{Bi} = x_{Bi} - x_{B\gamma}(a_{ij}/a_{\gamma j}) \quad , i \neq \gamma$$
$$x^{*}_{B\gamma} = (x_{B\gamma}/a_{\gamma j}) = \theta \quad (say)$$

where $\quad a_j = \sum_{i=1}^{m} a_{ij} b_i$

$$z^{1*} = \sum_{i=1}^{m} c_{Bi} x^{*}_{Bi} + \alpha$$
$$= \sum_{i=1}^{m} c_{Bi}(x_{Bi} - x_{B\gamma} a_{ij}/a_{\gamma j}) + c_{B\gamma}(x_{B\gamma}/a_{\gamma j}) + \alpha$$

Again $\quad = \sum_{i=1}^{m} c_{Bi} x_{Bi} - (\sum_{i=1}^{m} c_{Bi} a_{ij}) x_{B\gamma}/a_{\gamma j} + c_j x_{B\gamma}/a_{\gamma j} \quad ,(\because c_{B\gamma} = c_j)$

$$= z^1 + (c_j - z_j^1)\, x_{B\gamma}/a_{\gamma j} = z^1 + \theta\,(c_j - z_j^1)$$

Similarly, $\quad z^{2*} = z^2 + \theta\,(d_j - z_j^2)$

Where z_j^1 & z_j^2 refer to the original basic feasible solution.

The value of the objective function will be improved if

$$F^* > F$$

$$or, \quad [z^1 + \theta\,(c_j - z_j^1)] / [z^2 + \theta\,(d_j - z_j^2)] > z^1/z^2$$

$$or, \quad [z^1 + \theta\,(c_j - z_j^1)] / [z^2 + \theta\,(d_j - z_j^2)] - z^1/z^2 > 0$$

$$or, \quad z^2[z^1 + \theta\,(c_j - z_j^1)] - z^1[z^2 + \theta\,(d_j - z_j^2)] > 0$$

[z^2 & z^{2*} are positive , since the denominator of the objective function is positive for all feasible solution]

$or, \quad z^2(c_j - z_j^1) - z^1(d_j - z_j^2) > 0$

[θ being positive in the non- degenerate case ; if $\theta=0$, $F^*=F$]

Let $\Delta_j = z^2(c_j - z_j^1) - z^1(d_j - z_j^2)$

Now ,

$\Delta_j > 0$ if

Case I : $(d_j - z_j^2) > 0$

$$(c_j - z_j^1)/(d_j - z_j^2) > z^1/z^2$$

Case II : $(d_j - z_j^2) < 0$

$$(c_j - z_j^1)/(d_j - z_j^2) < z^1/z^2$$

Case III: $(d_j - z_j^2) = 0$

$$c_j - z_j^1 > 0$$

Swarup[1964] deduced that given a basic feasible solution $x_B = B^{-1}b$, if for any column a_j in A but not in B, $\Delta_j > 0$ holds and if at least one $a_{ij} > 0$ $(i = 1,2,\ldots,m)$, then it is possible to find a new basic feasible solution by replacing one of the column in B by a_j and new value of the objective function satisfies

$$F^* > F$$

One can show that for any a_j in A not in B at least one $a_{ij} \geq 0$. If possible, let all $a_{ij} \leq 0$ $(i = 1,2,\ldots,m)$

The basic feasible solution is given by

$$\sum_{i=1}^{m} x_{Bi} b_i = b \tag{4.4}$$

Where $\sum_{i=1}^{m} x_{Bi}$ is a component of basic vector. Now adding and subtracting $\theta^* a_j$ (θ^* being any scalar) to (4 3) , one obtains .

$$\sum_{i=1}^{m} x_{Bi} b_i - \theta^* a_j + \theta^* a_j = b \tag{4 5}$$

Since,

$$-\theta^* a_j = -\theta^* \sum_{i=1}^{m} a_{ij} b_i \tag{4.6}$$

Then

$$\sum_{i=1}^{m} (x_{Bi} - \theta^* a_{ij}) b_i + \theta^* a_j = b$$

where $\theta^* > 0$

Therefore $x_{Bi} - \theta^* a_{ij} \geq 0$

Since by assumption , $a_{ij} \leq 0$ $(i = 1, 2, \ldots, m)$

Therefore, $(x_{B1} - \theta^* a_{1j}, \ldots \ldots, x_{Bm} - \theta^* a_{mj})$ and θ^* is a feasible solution for all $\theta^* > 0$.
Thus the feasible set X is unbounded contrary to ones hypothesis of regularity. So for basic feasible solution if there is vector a_j not in basis having

$$\Delta_j > 0 \tag{4.7}$$

Then there exists another basic feasible solution with improved value of the objective function such that

$$F^* \geq F$$

For non-degenerate case

$$F^* > F$$

Thus one can move from one basis to another changing one vector at a time so long as there is some a_j not in basis with condition (4.7) and at each step F is improved.
This process can not continue infinitely, since there is only a finite number of basis and in non-degenerate case, no basis can ever be repeated, since F is increased at every step and the same basis can not yields two different values of F While at the same time the maximum value of the objective function occurs at of the basic feasible solution.

The process will terminate only one-way, that is, when all $\Delta_j \leq 0$ $(j = 1, 2 \ldots n)$ for the column a_j of A not in the basis.

Now for those columns of A which are in the basis

$$z_j^1 = c_B a_j = c_B B^{-1} a_j = c_B B^{-1} b_j = c_j$$

and

$z_j^2 = d_B a_j = d_B B^{-1} a_j = d_B B^{-1} b_j = d_j$

So, $\Delta_j = z^2(c_j - z_j^1) + z^1(d_j - z_j^2) = 0$

i. e., $\Delta_j = 0$.

Hence the summaries of the results are as follows:

Given a basic feasible solution

$$x_B = B^{-1} b$$

with

$$F^0 = \frac{c_B x_B + \alpha}{d_B x_B + \beta}$$

to the problem (4.1)-(4.3) such that all $\Delta_j \leq 0$ for every column a_j in A. Then F^0 is the maximum value of F and the corresponding basic feasible solution is an optimal solution.

Iterative procedure of Swarup's primal simplex type method

For Swarup[1964] primal simplex type algorithm are as follows:

Step I: First one has to convert the LFP problem to its standard form by inserting slack and surplus variables to the constraints. If the constraint set is in a canonical form, go to Step II. If the constraint set is not in a canonical form, go to Step IV.

Step II : Now one has to compute z^1, z^2, relative cost factor $c_j - z_j^1$, relative profit factor $d_j - z_j^2$ and the ratio Δ_j,

where $z^1 = c_B x_B + \alpha$

$z^2 = d_B x_B + \beta$

$z_j^1 = c_B a_j$

$$z_j^2 = d_B a_j$$

and

$$\Delta_j = z^2(c_j - z_j^1) - z^1(d_j - z_j^2)$$

Step III: One has to choose max $\Delta_j > o$ for covering optimality condition and to improve the basic solution. The minimum ratio test is to be applied to determine the new basic variable to enter the basic and the departing variable to leave the basis.

Step IV: If all $\Delta_j \leq 0$ one has reached to the optimal solution, other wise go to previous step.

Step V: If the constraint set is not in a canonical form, introduce artificial variables wherever it required and form an artificial linear objective function. In phase I, solve the problem as a LP. If it is feasible, go to phase II of the LFP problem and solve LFP problem using Step II to Step IV.

Here we develop the computer program (FORTRAN) of the above method is as follows:

Fortran Program for Swarup's primal simplex type method.

```
************************************************************************
C     * PROGRAM FOR SWARUP SIMPLEX TYPE METHOD
C     * M    NUMBER OF CONSTRAINTS
C     * N    NUMBER OF VARIABLES
C     * m1   NUMBER OF LESS THAN OR EQUAL TYPE CONSTRAINTS
C     * m2   NUMBER OF GREATER THAN OR EQUAL TYPE CONSTRAINTS
C     * m3   NUMBER OF EQUAL TYPE CONSTRAINTS
C     * ICASE  0 OPTIMAL SOLUTION IS FOUND
C     * ICASE -1 INCONSISTENT SOLUTION IS FOUND
C     * ICASE  1 UNBOUNDED SOLUTION IS FOUND
C     * A(i,j)COEFFICIENT MATRIX WITH NUMERATOR AND
C       DENOMINATOR OF THE OBJECTIVE FUNCTION
C**************************************************************

      Parameter(M=7,N=13)
      Real a(M+3,N+1)
      Integer np,mp,m1,m2,m3,icase,izrov(N),iposv(M)
       Open(1,tile='s6.dat')
       Open(2,filc='s9.dat')
```

```
      Read(1,*) m1,m2,m3
      mp=M+3
      np=N+1

      Read(1,*) ((a(i,j),j=1,np),i=1,M+2)


       Call simplx(a,M,N,mp,np,m1,m2,m3,icase,izrov,iposv)

        Write(2,*) " The left hand variables are:"
        Write(2,*) (iposv(j),j=1,M)
        Write(2,*) " The right hand variables are:"
        Write(2,*) (izrov(i),i=1,N)
        Write(2,*) " The value of the icase:"
        write(2,*) icase


        Write(2,88)((a(i,j),j=1,N+1),i=1,M+2)
88     Format(3x, 14(5x,f20.10))
       Write(2,*) " The maximum value is:  "
        Write(2,33) a(1,1)/a(2,1)
  33   Format(3x,f20.10)
      Stop
      End

Subroutine simplx(a,m,n,mp,np,m1,m2,m3,icase,izrov,iposv)
Integer  icase,m,m1,m2,m3,mp,n,np,izrov(n),iposv(m)
          MMAX,NMAX
        Real a(mp,np), EPS
        Parameter (MMAX=100, NMAX=100, EPS=.0001)
        Integer i,ip,is,k,kh,kp,nL1,L1(NMAX),L3(MMAX)
        Real bmax,q1
if(m.ne.m1+m2+m3) pause 'bad input constraint counts in
         simplex'

        nL1=n
        do 11 k=1,n

        L1(k)=k
        izrov(k)=k
 11      continue
       do 12 i=1,m
if(a(i+2,1).lt.0)   pause 'bad input constraint counts in
          simplex'
          iposv(i)=n+i
12      continue
```

```
30          call simp1(a,mp,np,0,L1,nL1,0,kp,bmax)
          if(bmax.le.EPS) then

             icase=0
             return
             endif

              call simp2(a,m,n,mp,np,ip,kp)
              if(ip.eq.0) then

              icase=1
              return
              endif

               call simp3(a,mp,np,m,n,ip,kp)
               is=izrov(kp)
               izrov(kp)=iposv(ip)
               iposv(jp)=is
               go to 30
               end

C*************************************************************
        Subroutine simp1(a,mp,np,mm,LL,nLL, iabf,kp,bmax)
C       Determines the pivot row
C*************************************************************

        Integer iabf,kp,mm,mp,nLL,np,LL(np)
        Real    bmax, a(mp,np)
        Integer k
        Real test
        if(nLL.le.0)then
         bmax=0
        else
        kp=LL(1)
        bmax= (a(2,1)*a(mm+1,kp+1))-(a(1,1)*a(mm+2,kp+1))
       do 11 k=2,nLL
        if(iabf.eq.0)then

     test=(a(2,1)*a(mm+1,LL(k)+1)-a(1,1)*a(mm+2,LL(k)+1))
                 -bmax
                else
   test=abs(a(2,1)*a(mm+1,LL(k)+1)-a(1,1)*a(mm+2,LL(k)+1))
               -abs(bmax)


        endif
```

```
      if(test.gt.0)then

      bmax= (a(2,1)*a(mm+1,LL(k)+1)
     -  (i,1)*a(mm+2,LL(k)+1))
      kp=LL(k)
      endif
 11   continue

      endif
      return
      end

C**************************************************
      Subroutine simp2(a,m,n,mp,np,ip,kp)
C      Determines pivot element
C**************************************************
       Integer  ip,kp,m,mp,n,np
       Real a(mp,np), EPS
       Parameter (EPS=.0001)

       Integer i,k
       Real q,q0,q1,qp
       ip=0

       do 11 i=1,m
       if(a(i+2,kp+1).lt.-EPS) go to 1
 11   continue

       return
1       q1=-a(i+2,1)/a(i+2,kp+1)
       ip=i
       do 13 i=ip+1,m

       if(a(i+2,kp+1).lt.-EPS)then
           q=-a(i+2,1)/a(i+2,kp+1)

         if(q.lt.q1)then

         ip=i
         q1=q
         elseif(q.eq.q1) then
         do 12 k=1,n
          qp=-a(ip+2,k+1)/a(ip+2,kp+1)
          q0=-a(i+2,k+1)/a(i+2,kp+1)
          if(q0.ne.qp)go to 2
```

```
12          continue

2              if(q0.lt.qp)ip=i
              endif
              endif
13        continue

              return
              end

c*****************************************************
          Subroutine simp3(a,mp,np,i1,k1,ip,kp)
C         Matrix operations to exchange a left-hand and
C          right-hand variable
C*****************************************************

          Integer  mp,np,i1,k1,ip,kp
           Real a(mp,np)
            Integer ii,kk
            Real v
            v=1./a(ip+2,kp+1)

             do 55 ii=1, i1+2
          if(ii-1.ne.ip+1)then

             a(ii,kp+1)=a(ii,kp+1)*v
               do 11 kk=1,k1+1
            if(kk-1.ne.kp)then

            a(ii,kk)=a(ii,kk)-a(ip+2,kk)*a(ii,kp+1)
             endif
 11       continue
            endif
 55        continue

            do 13 kk=1,k1+1
            if(kk-1.ne.kp)a(ip+2,kk)=-a(ip+2,kk)*v

 13      continue

            a(ip+2,kp+1)=v
            return
            end
```

Now, applying the above program to solve the **Production Problem formed in section 1.5 of Chapter-1**, we obtain the following data:

The left hand variables are (basic variables).

2 14 8 1 18 15 20

The right hand variables are (Non-basic variables):

```
17        4         3         19        5

6         7         16        9         10        11

12        13


The value of the icase: 0


1296630.00000       -1.93697            43305.37000
-1762.65800         -5.74450            2263.61000
894.91930           1066.10300          -1.84668
1283.69000          13192.69000         2719.59700
855.30640           -1256.62300

656070.90000        -.96538             21957.20000
-851.82430          -2.90308            1175.84200
521.28380           584.46430           -.92399
732.52560           6698.80300          1381.76200
1332.48400          -614.04010

32.46644            .00020              -.42698
-1.45907            -.00088             -.73864
-.17283             -.34465             .00010
-.01876             -.45188             .12211
.10885              -.72360

.14448              .00008              -.26217
-.01238             .00048              -.39912
-.40453             -.46478             -.00002
-.10798             -.31357             -.07591
-.12925             -.26829

7.96701             .00003              -.35953
.42193              .00031              .14129
-.44076             -.44968             -.00015
```

-.30519	-.48884	-.34983
-.49660	-.08127	
9.42207	-.00031	.04868
.04951	.00008	-.00354
.01811	.25912	.00006
-.56806	.25430	-.69637
-.48299	.07316	
5928.78100	.04899	-165.92260
-578.67490	.05774	- 615.80900
12.67998	-31.94990	.04529
-464.31070	244.67630	-20.79352
95.23967	117.87700	
78783.55000	-.20023	-21698.02000
1749.06500	1.12627	38.64083
72.82928	144.65420	-.09704
-106.23910	-6173.11600	-747.10930
-733.84600	898.59700	
25216.72000	.11661	-93.25553
-318.56700	.71907	-598.67360
-606.79270	-697.16960	-.02426
-161.97640	-770.36220	-613.86040
-693.87820	-402.43410	

The maximum value is:

1.97636

Hence solving the problem, we have obtained the following results:

To obtain maximum return on investment,

the farmer has to plant rice in 9.422207 hectors of land , wheat in 32.46644 hectors of land, tomatoes in 7.96701 hectors of land, and the maximum return on investment is 1 97636.

4.2. Swarup's dual simplex type method

In this section, we briefly present the dual simplex type method of Swarup[1965] assuming the positivity of the denominator of the objective function of LFP problem defined by (4.1) – (4.3).

In section 4.1 Swarup[1964] showed that any basic feasible solution will be optimal if

$$\Delta_j = z^2(c_j - z_j^1) - z^1(d_j - z_j^2),\ j = 1, 2, \ \dots\dots\ ,n.$$

The above observation presents the following interesting possibility, if one can start with some basic but not feasible solution to a given LFP problem with all $\Delta_j \leq 0$ and remove from this basic solution to another by changing one vector at a time in such a way that he keeps all $\Delta_j \leq 0$ provided no basic is to be repeated, an optimal solution to LFP problem will be obtained in a finite number of iterations That is , the fact that he maintains all $\Delta_j \leq 0$ at each iteration and is not concerned about the feasibility of the basic solution that the dual simplex method should be great help in developing such a method.

Swarup[1965] assumed that the given LFP problem with additional restrictions as follows:

Denominator of the objective function of LFP problem is positive for all basic solutions into the standard form for the application of simplex method.
Now if

$$x_B = A^{-1} b > 0$$

And

$$\Delta_j \leq 0,\ j = 1, 2, \dots\ \dots, n.$$

then he obtained an optimal solution to the LFP problem . He studied the case where one or more $x_{Bi} < 0$ ($i \in I$, I is the set of subscripts for basic variables).

The algorithm for the change of basis in LFP problem is·

Step I: Variable to lease the basis set is obtained as:
$x_{Br} = \min x_{Bi}$, for all $i \in I$ [$x_{Bi} < 0$]

So in LFP , x_{Br} will be driven to zero That is , x_{Br} will leave the basis set

StepII: Variable enter to the basis set is determined from
$\Delta_k / a_{rk} = \min \Delta j / a_{rj}$, for all j , [$a_{rj} < 0$]

Then one obtains $\Delta_k/Z^2[x_{Br}(d_k-Z_k^2)/Z^2+a_{rk}] = \min\limits_j \Delta_j/Z^2[x_{Br}(d_j-Z_j^2)/Z^2+a_{rj}]$

Where the coefficient of Z^2 in the denominator on the right is negative.
Then one assumes,

$$\delta_j = x_{Br}(d_j-Z_j^2)/Z^2+a_{rj}$$

Therefore, the variable x_{Br} to enter the basis set in the LFP problem is determined from

$$\Delta_k/\delta_k = \min_j \Delta_j/\delta_j$$

By adopting this procedure, Swarup[1965]maintain $\Delta_j \le 0$ at each iteration
Moreover, this method for solving LFP problem, one first determines the vector to leave the basis and then the vector to enter the basis This is reverse of what is done in simplex procedure for solving LFP problem

Our computer program (FORTRAN) of the Swarup dual type method is as follow·

Fortran Program for Swarup [1965] dual simplex type method.

```
**************************************************************************
C    * PROGRAM FOR SWARUP DUAL SIMPLEX TYPE METHOD
C    * M    NUMBER OF CONSTRAINTS
C    * N    NUMBER OF VARIABLES
C    * m1   NUMBER OF LESS THAN OR EQUAL TYPE CONSTRAINTS
C    * m2   NUMBER OF GREATER THAN OR EQUAL TYPE CONSTRAINTS
C    * m3   NUMBER OF EQUAL TYPE CONSTRAINTS
C    * ICASE  0 OPTIMAL SOLUTION IS FOUND
C    * ICASE -1 INCONSISTENT SOLUTION IS FOUND
C    * ICASE  1 UNBOUNDED SOLUTION IS FOUND
C    * A(i,j)COEFFICIENT MATRIX WITH NUMERATOR AND
C      DENOMINATOR OF THE OBJECTIVE FUNCTION
C***********************************************************************

      Parameter (M=8,N=13)
      Real a(M+5,N+1)
      Integer np, mp, m1, m2, m3, icase, izrov (N),iposv(M)
       Open(1,file='ss1.dat')
      Open (2,file='ss2.dat')
      Read (1,*) m1,m2,m3
      mp=M+5
      np=N+1

      Read(1,*) ((a(i,j),j=1,np),i=1,M+2)
```

```
      Call simplx(a,M,N,mp,np,m1,m2,m3,icase,izrov,iposv)

       Write(2,*) " The left hand variables are:"
       Write(2,*) (iposv(j),j=1,M)
       Write(2,*) " The right hand variables are:"
       Write(2,*) (izrov(i),i=1,N)
       Write(2,*) " The value of the icase:"
       write(2,*) icase

       Write(2,88)((a(i,j),j=1,N+1),i=1,M+2)
88    Format(1x,7(5x,f20.10))
      Write(2,*) " The maximum value is:  "
       Write(2,33) a(1,1)/a(2,1)
 33   Format(3x,f20.10)
      stop
      end

 Subroutine simplx(a,m,n,mp,np,m1,m2,m3,icase,izrov,iposv)
    Integer  icase,m,m1,m2,m3,mp,n,np,izrov(n),iposv(m)
             MMAX,NMAX
       Real a(mp,np), EPS
       Parameter (MMAX=100, NMAX=100, EPS=.0001)
       Integer i,ip,is,k,kp,nL1,L1(NMAX),L11(MMAX),mL1
       Real bmin,bratio
   if(m.ne.m1+m2+m3) pause 'bad input constraint counts in
           simplex'
       mL1=m
       Do 77 i=1,m
       L11(i)= i
 77    Continue

       nL1=n
       do 11 k=1,n

       L1(k)=k
       izrov(k)=k
 11      continue
       do 12 i=1,m

         iposv(i)=n+i
12     continue


30         call simp1(a,mp,np,0,L11,mL1,0,ip,bmin)
           if(bmin.gt.EPS) then
           write(*,*)"line 62"
```

100808

```
          icase=0
          return
         endif
               write(*,*)"line 66"
          call simp2(a,m,0,L1,nL1,mp,np,0,ip,kp)

          if(kp.eq.0.0) then

          icase=1
          return
          endif

           call simp3(a,mp,np,m,n,ip,kp)

           is=izrov(ip)
           izrov(ip)=iposv(kp)
           iposv(kp)=is
           go to 30
           end


C************************************************************
        Subroutine simp1(a,mp,np,mm,LL1,mLL, iabf,ip,bmin)
C       Determines the pivot row
C************************************************************


        Integer iabf,ip,mm,mp,mLL,np,LL1(mp)
        Real   bmin, a(mp,np)
        Integer k
        Real test
        if(mLL.le.0)then
          bmin=0
        else
       ip=LL1(1)

            bmin= a(ip+2,mm+1)
           do 11 k=2,mLL
        if(iabf.eq.0)then

        test=a(LL1(K)+2,mm+1)-bmin
                else
       test=abs(a(LL1(K)+2,mm+1))-abs(bmin)


       endif
```

```
      if(test.lt.0)then

      bmin= a(LL1(K)+2,mm+1)
      ip=LL1(k)
      endif
 11   continue
       endif

      return
      end

c*******************************************************
      Subroutine simp2(a,m,nn,LL,nLL,mp,iabf1,np,ip,kp)

C     Determines pivot element
C*******************************************************

      Integer  ip,kp,m,mp,np,nn,LL(np),nLL,iabf1,i,k,j
       Real a(mp,np),test
      if(nLL.le.0)then

       else

       do 11 i=1,nLL
       a(m+3,i)=a(2,1)*a(nn+1,i+1)-a(1,1)*a(nn+2,i+1)
       a(m+4,i)=(a(ip+2,1)*a(nn+2,i+1))/a(2,1)
           +a(ip+2,i+1)
 11     continue

       Do 31 ii=1,nLL
       If(a(m+4,ii).lt.0.0)go to 1

 31    continue
       return

 1     q1=a(m+3, ii)/a(m+4,ii)
       kp=ii
       do 100 ii=kp+1, nLL
       If(a(m+4,ii).lt.0.0) then
       q= a(m+3, ii)/a(m+4,ii)
       if(q.eq.q1)then
       kp=ii
       q1=q
       endif
       endif
100    continue
```

```
       return
       endif
       end
C***********************************************************
       Subroutine simp3(a,mp,np,il,kl,ip,kp)
C       Matrix operations to exchange a left-hand and
C        right-hand variable
C***********************************************************
        Integer  mp,np,il,kl,ip,kp
         Real a(mp,np)
          Integer ii,kk
          Real v
          v=1./a(ip+2,kp+1)
                write(*,*) v
           do 55 ii=1, il+2
        if(ii-1.ne.ip+1)then

           a(ii,kp+1)=a(ii,kp+1)*v
             do 11 kk=1,kl+1
          if(kk-1.ne.kp)then

          a(ii,kk)=a(ii,kk)-a(ip+2,kk)*a(ii,kp+1)
           endif
 11      continue
          endif
 55       continue

          do 13 kk-1,kl+1
          if(kk-1.ne.kp)a(ip+2,kk)=-a(ip+2,kk)*v

 13     continue

          a(ip+2,kp+1)=v
          return
          end
```

Now, we consider a numerical example of Swarup dual type method and solve it by the above program.

Example 4.2.1

(LFP) Maximize $Z = \dfrac{-x_1 + x_2 + 2}{x_2 + 2}$

Subject to

$$4x_1 - 3x_2 \geq 2$$
$$x_1 \leq 5$$
$$x_1 \geq 2$$
$$x_1, \quad x_2 \geq 0$$

Now, applying the above program, we have obtained the following data:

The left hand variables are (basic variable):
1 5 2
The right hand variables are (non-basic variable):
4 3
The value of the icase: 0 (Optimal solution is found)

2.0000000000	.3333334000	-.3333333000
4.0000000000	1.3333330000	-.3333333000
2.0000000000	1.3333330000	-.3333333000
3.0000000000	-1.0000000000	.0000000000
2.0000000000	1.0000000000	.0000000000

The maximum value is:
.5000000000

Thus applying Swarup dual type method, one can obtain the following results.

x = (2,2) and Zmax = 0.5

4.3. The modified approach of Swarup's primal simplex type method

In this section, we suggest a modification based on primal simplex type method, which extends the scope of Swarup [1964] method discussed in section 4.1 of this chapter. Assuming the positivity of the denominator of the objective function of LFP problem defined by

(LFP) $\quad Maximize \ F(x) = \dfrac{cx+\alpha}{dx+\beta}$ (4.8)

Subject to

$$Ax = b \tag{4.9}$$

$$x \geq 0 \tag{4.10}$$

Where x, c, d $\in R^n$, b$\in R^m$ α & β $\in$R; A is an m$\times$n matrix.

Swarup[1964]first developed a method for solving LFP problem. However, this method can be applied only when the system Ax = b is in a canonical form, that is, all constraints are less than or equal form ($\leq$). The problem that is not in canonical form, one can solve by using dual simplex type method developed by Swarup[1965]. Likewise, LP problem, dual simplex type method also cannot be applied in the case where the dual feasible basis is not obtained. Let us consider the following numerical example

Example 4.3.1

(LFP) $\quad$ Maximize $\quad Z = \dfrac{x_2 - 5}{-x_1 - x_2 + 9}$

Subject to .

$$2x_1 + 5x_2 \geq 10$$
$$4x_1 + 3x_2 \leq 20$$
$$-x_1 + x_2 \leq 2$$
$$x_1 \quad x_2 \geq 0$$

Now, introducing surplus and slack variables s_1 and s_2 , s_3 to 1st and 2nd & 3rd constraints respectively to make the LFP problem in the standard form as follows:

LFP( ) $\quad$ Maximize $\quad Z = \dfrac{x_2 - 5}{-x_1 - x_2 + 9}$

Subject to

$$2x_1 + 5x_2 - s_1 = 10$$
$$4x_1 + 3x_2 + s_2 = 20$$
$$-x_1 + x_2 + s_3 = 2$$
$$x_1 \quad x_2, s_1, s_2, s_3 \geq 0 \ .$$

Thus the initial basic solution

s_1 = -10, s_2 = 20, s_3 - 2 and x_1= x_2 = 0

Now proceed to construct simplex table as follows:

Initial Table

c_B ↓	d_B ↓	c_j →	0	1	0	0	0
		d_j →	-1	-1	0	0	0
		x_B	x_1	x_2	s_1	s_2	s_3
0	0	s_1= -10	-2	-5	1	0	0
0	0	s_2= 20	4	3	0	1	0
0	0	s_3=2	-1	1	0	0	1
z^1=-5	z^2=9	Z=-9/5					
		c_j-z_j^1	0	1	0	0	0
		d_j-zj^2	-1	-1	0	0	0
		Δ_j →	-5	[4]	0	0	0
		δj	-8/9 ↑	-35/9	1	0	0
		Δ_j/δj	45/8	-36/35	0	-	-

To obtain optimal solution we must maintain all $\Delta_j \le 0$ at each optimization stage. But in the initial table, we observed that $\Delta_2 = 4 > 0$, which indicates failure of Swarup[1965] dual type method

To overcome the above limitation of Swarup[1964 & 1965] method we suggest a modification based on Dantiz [1960] two phase method for solving LP problems.

4.3.1. Solution procedure of modified approach of Swarup [1964]

After introducing slack variables or surplus variables if the constraints set

$$Ax = b \tag{4.11}$$

Is in a canonical form, it indicates that some of the constraints are greater than or equal form and one can not find leading initial basic feasible solution In that case one has to follow the procedure-described bellow

Step I: First, one has to find a initial basic feasible solution of LFP problem. Since, PHASE I of simplex method concerned with finding initial basic feasible solution with respect to artificial objective function and not relevant to original objective function of LFP problem, one can find an initial basic solution as follows

Sub-step 1: We augment the system (4 11) to include a basic set of artificial variables $w_i \geq 0$ (i =1,2,....,m) so that we have augmented the system

$$Ax + Iw = b$$

Where,

$$w = (w_1, w_2, \ldots, w_m)^T.$$

Sub-step 2: Solve the artificial linear program (ALP)

$$\text{(ALP)} \qquad \text{Minimize } w = \sum_{i=1}^{m} w_i \tag{4.12}$$

Subject to

$$Ax + Iw = b \tag{4.13}$$

$$x \geq 0 \tag{4.14}$$

Sub-step 3: Since $w \geq 0$, this problem can not have an unbound solution. Moreover this problem is feasible, since $w^* = b$, $x^* = 0$, is a basic feasible solution in which artificial variables w_i (i =1,2,.. .. ..,m) are basic. Writing the system (4.13) with using the variable coefficient of (4.12), we obtain initial tableau of PHASE I It should be noted that the initial tableau contains rows corresponding to the original objective function F of LFP problem (to be maximized in PHASE II) and PHASE I objective function w. The Simplex method can now be applied to this tableau to minimize w. It would be terminated an optimal basic solution (x^*, w^*) to this PHASE I problem has been found , and in this situation, min $w \geq 0$. This is the end of PHASE I.

Two cases may hold now:

CASE A: min w>0

Here ALP has no solution, since if there is $x \geq 0$ satisfying (4.11), then $(x^*,0)$ is a feasible solution with $w = 0$ to (4.11)-(4 14) and this violets the assumption $w>0$ Thus the simplex method terminates with the conclusion that ALP has no feasible solution.

CASE B: $\min w = 0$

Here all the artificial variables have been zero, i.e, $w_i^* = 0$ Thus ALP has a feasible solution x^*. From the final simplex tableau of PHASE I, one can now delete the objective row corresponding to w, since it has served its purpose with this as the starting basic feasible solution, one proceeds to PHASE II in the next step.

Step II: If PHASE I yields an optimal solution not involving positive artificial variables, one can start PHASE II with original objective function of LFP problem and initial basic feasible solution, which is optimal solution of PHASE I. Then one has to apply the primal simplex type method of Swarup [1964] to maximize F, which terminates as soon as either an optimal solution or an unbounded one. In an unbounded solution, all entries in the pivot column are non-positive corresponding to the greatest opposite relative profit factor.

Step III: Now one has to compute z^1, z^2, relative cost factor $c_j - z_j^1$, relative profit factor $d_j - z_j^2$ and the ratio Δ_j,

where $z^1 = c_B x_B + \alpha$

$$z^2 = d_B x_B + \beta$$

$$z_j^1 = c_B a_j$$

$$z_j^2 = d_B a_j$$

and

$$\Delta_j = z^2(c_j - z_j^1) - z^1(d_j - z_j^2)$$

Step IV: One has to choose $\max \Delta_j > 0$ for covering the optimality condition and to improve the basic solution and the minimum ratio test is to be app[lied to determine the new basic variable to enter the basis and the departing variable to leave the basis.

Step V: If all $\Delta_j \leq 0$ in the previous step, then one has reached to the optimal solution. Otherwise one has to go to Step II

Now, we solve the above Example 4.3.1 applying our modified approach of Swarup [1964] as follows:

Step I: We have to first find a initial basic feasible solution of the given LFP problem. To do this, we consider the following ALP:

(ALP) Minimize $L^* = w$

Subject to

$$2x_1 + 5x_2 - s_1 + w = 10$$
$$4x_1 + 3x_2 + s_2 = 20$$
$$-x_1 + x_2 + s_3 = 2$$
$$x_1 \quad x_2, s_1, s_2, s_3, w \geq 0$$

Now we construct the simplex table for PHASE-I as follows.

PHASE-I

c_B ↓	c_j →	0	0	0	0	0	1	B
	Basis↓	x_1	x_2	s_1	s_2	s_3	w	
1	w	2	5	-1	0	0	1	10
0	s_2	4	3	0	1	0	0	20
0	s_3	-1	1	0	0	1	0	2
$c_j^* = c_j - z_j$		-2	-5 ↑	1	0	0	0	-10
0	x_2	2/5	1	-1/5	0	0	1/5	2
0	s_2	14/5	0	3/5	1	0	-3/5	14
0	s_3	-7/5	0	1/5	0	1	-1/5	0
$C_j^* = c_j - z_j$		0	0	0	0	0	1	0

Since all $c_j^* \geq 0$ and there is no artificial variable in the last table, it yields a primal feasible solution., this table gives another sub optimal point

$s_1 = 0$, $s_2 = 14$, $s_3 = 0$ and $x_1 = 0$, $x_2 = 20$

Step II: Now the initial basic feasible solution is

$s_2 = 14$, $s_3 = 0$ and $x_2 = 20$

with $\Delta_1 = -23/5$, $\Delta_2 = 4/5$ and $\Delta_3 = \Delta_4 = \Delta_5 = 0$

Now, we construct initial table as follows:

INITIAL TABLE

c_B ↓	d_B ↓	c_j →	0	1	0	0	0	
		d_j →	-1	-1	0	0	0	
		x_{Bi} ↓	x_1	x_2	s_1	s_2	s_3	x_{Bi}/a_{ij}
1	-1	$x_2=2$	2/5	1	-1/5	0	0	-10
0	0	$s_2=14$	14/5	0	3/5	1	0	70/3
0	0	$s_3=0$	-7/5	0	1/5	0	1	0
$z^1=-3$	$z^2=7$	$Z=-3/7$	-	-	-	-		-
		$c_j-z_j^1$	-2/5	0	1/5	0	0	
		$d_j-z_j^2$	-3/5	0	-1/5	0	0	
		Δ_j →	-23/5	0	4/5 ↑	0	0	

Step III: Since max $\Delta_j = \Delta_3 = 4/5 > 0$ and min $x_{Bi}/a_{ij} = 0$. Thus s_1 enter to the basis and s_3 leave to the basis.

FIRST ITERATION

c_B ↓	d_B ↓	c_j →	0	1	0	0	0	
		d_j →	-1	-1	0	0	0	
		x_{Bi} ↓	x_1	x_2	s_1	s_2	s_3	x_{Bi}/a_{ij}
1	-1	$x_2=2$	-1	1	0	0	0	-2/1
0	0	$s_2=14$	7	0	0	1	-3	14/7
0	0	$s_1=0$	-7	0	1	0	5	0/-7
$z^1=-3$	$Z^2=7$	$Z=-3/7$	-	-	-	-		-
		$c_j-z_j^1$	1	0	0	0	-1	
		$d_j-z_j^2$	0	0	0	0	1	
		Δ_j →	7 ↑	0	0	0	-4	

Step IV: Since max $\Delta_j = \Delta_1 = 7 > 0$ and min $x_{Bi}/a_{i1} = 2$. Thus x_1 enter to the basis and s_2 leave to the basis.

FIRST ITERATION

c_B ↓	d_B ↓	c_j →	0	1	0	0	0	
		d_j →	-1	-1	0	0	0	
		x_{Bi} ↓	x_1	x_2	s_1	s_2	s_3	x_{Bi}/a_{ij}
1	-1	$x_2 = 4$	0	1	0	1/7	4/7	
0	0	$x_1 = 2$	1	0	0	1/7	-3/7	
0	0	$s_1 = 14$	0	0	1	1	2	
$z^1 = -1$	$z^2 = 3$	$Z = -1/3$	-	-	-	-		-
		$c_j - z_j^1$	0	0	0	-1/7	-4/7	
		$d_j - z_j^2$	0	0	0	2/7	1/7	
		Δ_j →	0	0	0	-1/7	-11/7	

Since all $\Delta j \leq 0$ in the above table, this table yields an optimal solution.

Thus the solution of the Example 4.3.1 is:

$$x_1 = 2 \ , \ x_2 = 4 \quad \text{with } Z_{max} = -1/3$$

Similarly, by using Charned & Cooper method and Bitran- Novaes method one can also obtain the same result

CHAPTER -5

COMPERATIVE STUDY OF THE METHODS

5.1 Introduction

In this chapter, we discuss the comparative analysis on the Method of Bitran & Novaes [1972], Swarup [1964 & 1965] and Charnes & Cooper [1962] transformation technique considering the sign of the numerator & denominator of the objective function of LFP problem Islam &Nath [1992] investigated on Charnes & Cooper [1962] transformation method and they considered the following six cases:

CASE I: $dx + \beta > 0, \forall x \in X$
CASE II: $dx + \beta < 0, \forall x \in X$
CASE III: $dx + \beta = 0, \forall x \in X$
CASE IV: $dx + \beta \geq 0, \forall x \in X$
CASE V: $dx + \beta \leq 0, \forall x \in X$
CASE VI: $dx + \beta$ changes sign over the feasible region X

Bitran & Novaes [1972] and Swarup [1964& 1965] considered only the case where the denominator: $dx + \beta$ of objective function of LFP problem is strictly positive (i.e, CASE I of Charnes & Cooper [1962]).

If we multiply the CASE II and CASE V by -1, they reduced to CASE I and CASE IV respectively Therefore, we may reduce the above six cases investigated by Islam &Nath [1992] into the following four cases:

CASE I : $dx + \beta > 0, \forall x \in X$
CASE II $dx + \beta = 0, \forall x \in X$
CASE III : $dx + \beta \geq 0, \forall x \in X$
CASE IV: $dx + \beta$ changes sign over the feasible region X

It is further noted that CASE III consists of the following two sub cases:

(a) $dx + \beta \geq 0$ & $cx+\alpha > 0, \forall x \in X$
(b) $dx + \beta \geq 0$ & $cx+\alpha < 0, \forall x \in X$

We can also observe the following:

1. If $dx + \beta = 0$, $\forall x \in X$, the objective function of LFP problem is undefined and thus the question of solving the LFP problem is meaning less

2. If $dx + \beta \geq 0$ & $cx + \alpha > 0$, $\forall x \in X$, the objective function $(cx+\alpha)$)/ $(dx + \beta)$ tends to infinite, where $dx + \beta$ tends to zero for some $x \in X$ and consequently the problem has no finite solution.

3. If $dx + \beta$ changes sign over X , the objective function $(cx+\alpha)$)/ $(dx + \beta)$ becomes undefined at which $dx + \beta$ equals to zero and it tends to infinite at which $dx + \beta$ tends to zero for some $x \in X$ and consequently the problem becomes unbounded.

4. The remaining only case where $dx + \beta \geq 0$ & $cx + \alpha < 0$, $\forall x \in X$. Through in this case the objective function of LFP problem tends to infinite at which $dx + \beta$ tends to zero for some $\forall x \in X$, the problem may have finite solution, as it is maximization one. So, finally it is enough to consider the following two cases instead of six cases considered by Islam &Nath [1992].

CASE A : $dx + \beta > 0$, $\forall x \in X$
CASE B,. $dx + \beta \geq 0$ & $cx + \alpha < 0$, $\forall x \in X$

5.2. If the denominator is strictly positive ($dx + \beta > 0$, $\forall x \in X$)

If the denominator the objective function is strictly positive the Method of Bitran & Novaes [1972], Swarup [1964 & 1965], our modified method of Swarup [1964] and Charnes & Cooper [1962] transformation technique solve the LFP problem successfully. We now illustrate this by simple numerical examples.

Example 5.1:

(LFP) Maximize $Z = \dfrac{-x_1 + x_2 + 2}{-x_2 + 2}$

Subject to

$$4x_1 - 3x_2 \geq 2$$
$$x_1 \leq 5$$
$$x_1 \geq 2$$
$$x_1, x_2 \geq 0$$

Now, we solve the above problem by using Bitran- Novaes method as follows:

Here $c = (-1, 1)$, $\alpha = 2$

and

$d = (0, 1)$, $\beta = 2$

$$\therefore \langle c,d\rangle = \sum_{i=1}^{2} c_i d_i = (-1,1)(0,1) = 1$$

$$\langle d,d\rangle = \sum_{i=1}^{2} d_i^{\,2} = (0,1)(0,1) = 1$$

Thus $\dfrac{\langle c,d\rangle}{\langle d,d\rangle} = 1$

Step I : The linear objective function L is given by

$$L = \langle \gamma, x\rangle = \langle \left[\ c - \frac{\langle c,d\rangle}{\langle d,d\rangle} d\ \right], x\rangle$$

$$= \langle \left[\ (-1,1) - 1(0,1)\ \right], x\rangle$$

$$= -x_1$$

Now we maximize L subject to the same constraints as follows:

LP) Maximize $L = -x_1$

Subject to

$$4x_1 - 3x_2 \geq 2$$

$$x_1 \leq 5$$

$$x_1 \geq 2$$

$$x_1 \quad x_2 \geq 0$$

Inserting surplus variables s_1 & s_3 and slack variable s_2 to the 1^{st}, 3^{rd} & 2^{nd} constraints respectively to make the LP problem to its standard form as follows

(LPI) Maximize $L = -x_1$

Subject to

$$4x_1 - 3x_2 - s_1 = 2$$

$$x_1 + s_2 = 5$$

$$x_1 - s_3 = 2$$

$$x_1, \quad x_2, s_1, s_2, s_3 \geq 0$$

Adding artificial variables w_1 , w_2 to the 1^{st} and 3^{rd} constraints and assign profit -1 to each of the artificial variables and profit zero to all other variables in the objective function, then PHASE – I of the LP is:

(ALP) Maximize $Z = -w_1 - w_2$

Subject to

$$4x_1 - 3x_2 - s_1 + w_1 = 2$$
$$x_1 + s_2 = 5$$
$$x_1 - s_3 + w_2 = 2$$
$$x_1,\ x_2, s_1, s_2, s_3, w_1, w_2 \geq 0$$

Now, we construct simplex table of the PHASE-I as follows :

PHASE-I

c_B ↓	c_j →	0	0	0	0	0	-1	-1	B
	Basis↓	x_1	x_2	s_1	s_2	s_3	w_1	w_2	
-1	w_1	4	-3	-1	0	0	1	0	2
0	s_2	1	0	0	1	0	0	0	5
-1	w_2	1	0	0	0	-1	0	1	2
$c_j^* = c_j - z_j$		5↑	-3	-1	0	-1	0	0	4
0	x_1	1	-3/4	-1/4	0	0	1/4	0	1/2
0	s_2	0	3/4	1/4	1	0	-1/4	0	9/2
-1	w_2	0	3/4	1/4	0	-1	-1/4	1	3/2
$c_j^* = c_j - z_j$		0	3/4↑	1/4	0	-1	-1/4	0	3/2
0	x_1	1	0	0	0	4/3	0	-1	2
0	s_2	0	0	0	1	4/3	0	-1	3
0	x_2	0	1	1/3	0	- 4/3	-1/3	4/3	2
$c_j^* = c_j - z_j$		0	0	0	0	0	-1	-1	0

Since all $c_i^* \leq 0$ and there is no artificial variables in the basis this last table gives an optimal solution for PHASE-I and we construct the simplex table for PHASE-II.

PHASE-II

c_B ↓	c_j →	-1	0	0	0	0	B
	Basis↓	x_1	x_2	s_1	s_2	s_3	
-1	x_1	1	0	0	0	4/3	2
0	s_2	0	0	0	1	4/3	5
0	x_2	0	1	1/3	0	- 4/3	2
$c_j^* = c_j - z_j$		0	0	0	0	-4/3	2

Since all $c_j^* \leq 0$, this table gives a sub optimal point $x^1 = (2, 2)$ with $Z(x^1) = -1/2$.

Step II : Again, the new objective function L' is given by

$$\begin{aligned} L' &= \langle [c - Z(x^1)d \ ], x\rangle \\ &= \langle [(-1,1) - 1/2(0,1) \ ], x\rangle \\ &= -x_1 + 1/2\, x_2 \end{aligned}$$

Now, we maximize L' subject to the same set of constraints and hence applying two phase simplex method , PHASE-II of the problem is given by

PHASE-II

c_B ↓	c_j →	-1	1/2	0	0	0	B
	Basis↓	x_1	x_2	s_1	s_2	s_3	
-1	x_1	1	0	0	0	4/3	2
0	s_2	0	0	0	1	4/3	5
1/2	x_2	0	1	1/3	0	- 4/3	2
$c_j^* = c_j - z_j$		0	0	-1/6	0	-2/3	1

Since all $c_j^* \leq 0$, this table gives another sub optimal point

$x^2 = (2,2)$ with $Z(x^2) = 1/2$

Now, since $x^1 = x^2 = (2,2)$, therefore, we have reached to the optimal solution $x = (2,2)$ with $Z_{max} = 1/2$

Now, we solve the above problem by using **our modified approach** as follows:

(ALP) Minimize $L = w_1 + w_2$

Subject to

$$4x_1 - 3x_2 - s_1 + w_1 = 2$$
$$x_1 \quad + s_2 = 5$$
$$x_1 - s_3 \quad + w_2 = 2$$
$$x_1,\ x_2, s_1, s_2, s_3, w_1, w_2 \geq 0$$

Now, we construct simplex table of the PHASE-I as follows:

PHASE-I

c_B ↓	c_j →	0	0	0	0	0	1	1	B
	Basis↓	x_1	x_2	s_1	s_2	s_3	w_1	w_2	
1	w_1	4	-3	-1	0	0	1	0	2
0	s_2	1	0	0	1	0	0	0	5
1	w_2	1	0	0	0	-1	0	1	2
$c_j^* = c_j - z_j$		-5 ↑	3	1	0	1	0	0	-4
0	x_1	1	-3/4	-1/4	0	0	1/4	0	1/2
0	s_2	0	3/4	1/4	1	0	-1/4	0	9/2
1	w_2	0	3/4	1/4	0	-1	-1/4	1	3/2
$c_j^* = c_j - z_j$		0	-¾ ↑	-1/4	0	1	¼	0	-3/2
0	x_1	1	0	0	0	4/3	0	-1	2
0	s_2	0	0	0	1	4/3	0	-1	3
0	x_2	0	1	1/3	0	-4/3	-1/3	4/3	2
$c_j^* = c_j - z_j$		0	0	0	0	0	1	1	0

Since all $c_j^* \leq 0$ and there is no artificial variables in the basis this last table, it yields a basic feasible solution.

$x_1 = x_2 = 2$ and $s_2 = 3$

Thus initial basic solution of LFP problem.

$x_1=x_2=2$ and $s_2=3$

with $\Delta_1=\Delta_2=\Delta_4=0$ and $\Delta_3=-2/3$, $\Delta_5=-4/3$

Initial Table

$c_B \downarrow$	$d_B \downarrow$	$c_j \longrightarrow$	-1	1	0	0	0
		$d_j \longrightarrow$	0	1	0	0	0
		x_{Bi}	x_1	x_2	s_1	s_2	s_3
-1	0	$x_1=2$	1	0	0	0	-1
0	0	$s_2=3$	0	0	0	1	1
1	1	$x_2=2$	0	1	1/3	0	−4/3
$z^1=2$	$z^2=4$	$Z=1/2$					
		$c_j-z_j^1$	0	0	−1/3	0	1/3
		$d_j-z_j^2$	0	0	−1/3	0	4/3
		$\Delta_j \longrightarrow$	0	0	−2/3	0	−4/3

Since all $x_{Bi} > 0$ & all $\Delta_j \le 0$, we have reached to the optimal solution and the optimal solution is : $x_1=2$ and $x_2=2$ with $Z_{max}=1/2$.

Similarly, by using Swarup **dual type method and Charnes & Cooper method**, one can obtain the following results :

$$x_1=x_2=2 \text{ with } Z_{max}=1/2.$$

5.3. If the denominator $dx+\beta \ge 0$ and the numerator $cx+\alpha < 0$, $\forall x \in X$.

If the denominator $dx+\beta \ge 0$ and the numerator $cx+\alpha < 0$, $\forall x \in X$, the method of Bitran-Novaes may fail, whereas our modified approach of Swarup simplex type method, Swarup dual type method & Charnes-Cooper transformation technique will always recognize and stop at an optimal point , if such a point is reached . We now illustrate this difference by following simple example.

Example 5.2:

(LFP) Maximize $Z=\dfrac{-2x_1-3x_2}{2-x_1-x_2}$

Subject to

$$x_1 + x_2 \le 2$$
$$x_1 + x_2 \ge 1$$
$$x_1 \quad x_2 \ge 0$$

Now, introducing slack and surplus variables s_1 and s_2 to 1st and 2nd constraints respectively to make the LFP in the standard form.

Thus the initial basic solution
$s_1 = 2$, $s_2 = -1$ and $x_1 = x_2 = 0$
with $\Delta_1 = -4$, $\Delta_2 = -6$, $\Delta_3 = \Delta_4 = 0$

Now proceed to construct simplex table as follows

Initial Table

$c_B \downarrow$	$d_B \downarrow$	$c_j \longrightarrow$	-2	-3	0	0
		$d_j \longrightarrow$	-1	-1	0	0
		x_B	x_1	x_2	s_1	s_2
0	0	$s_1 = 2$	1	1	1	0
0	0	$s_2 = -1$	-1	-1	0	1
$z^1=0$	$z^2=2$	$Z=0$				
		$c_j - z_j^1$	-2	-3	0	0
		$d_j - z_j^2$	-1	-1	0	0
		$\Delta_j \longrightarrow$	-4	-6	0	0
		δj	-1/2	-1/2	0	1
		$\Delta_j / \delta j$	8 $\uparrow$	12	-	-
0	0	$s_1 = 1$	0	0	1	1
-2	-1	$x_1 = 1$	1	1	0	-1
$Z^1=-2$	$Z^2=1$	$Z=-2$				
		$c_j - z_j^1$	0	-1	0	-2
		$d_j - z_j^2$	0	0	0	-1
		$\Delta_j \longrightarrow$	0	-1	0	-4

Since all $x_{Bi} > 0$ & all $\Delta_j \le 0$, we have reached to the optimal solution and the optimal solution is :
$x_1 = 1$ and $x_2 = 0$ with $Z_{max} = -2$

Now, we solve the above problem by **using our modified approach** as follows:

(ALP) Minimize $L = w$

Subject to

$$x_1 + x_2 + s_1 = 2$$
$$x_1 + x_2 - s_2 + w = 1$$
$$x_1, x_2, s_1, s_2, w \geq 0$$

Now, we construct simplex table of the PHASE-I as follows:

PHASE-I

c_B ↓	c_j ⟶	0	0	0	0	1	B
	Basis↓	x_1	x_2	s_1	s_2	w	
0	s_1	1	1	1	0	0	2
1	w	1	1	0	-1	1	1
$c_j^* = c_j - z_j$		-1↑	-1	0	1	0	-1
0	s_1	0	0	1	1	-1	1
0	x_1	1	1	0	-1	1	1
$c_j^* = c_j - z_j$		0	0	0	0	1	0

Since all $c_j^* \leq 0$ and there is no artificial variables in the basis this last table, it yields a basic feasible solution

$x_1=1$ and $s_1=1$

Thus initial basic solution of LFP problem

$x_1=1$ and $s_1=1$

with $\Delta_1 = \Delta_3 = 0$ and $\Delta_1 = -1$, $\Delta_4 = -4$

Initial Table

$c_B \downarrow$	$d_B \downarrow$	$c_j \rightarrow$ $d_j \rightarrow$	-2 -1	-3 -1	0 0	0 0
		x_{Bi}	x_1	x_2	s_1	s_2
0 -2	0 -1	s_1=1 x_1=1	0 1	0 1	1 0	1 -1
Z^1=-2	Z^2=1	Z=-2				
		$c_j-z_j^1$ $d_j-z_j^2$	0 0	-1 0	0 0	-2 -1
		$\Delta_j \rightarrow$	0	-1	0	-4

Since all x_{Bi} >0 & all $\Delta_j \le 0$, we have reached to the optimal solution and the optimal solution is : x_1= 1 and x_2 = 0 with Z_{max} = -2.

Similarly, by using **Charnes & Cooper transformation technique,** one can obtain the following results:
x_1= 1 and x_2 = 0 with Z_{max} = -2.

On the other hand if we apply **Bitran-Novaes method** to solve example 5.2; we obtain

Here $c=(-2,-3)$, $\alpha=0$

and

$$d=(-1,-1)\ ,\ \beta=2$$

$$\therefore \langle c,d\rangle=\sum_{i=1}^{2} c_i d_i=(-2,-3)(-1,-1)=5$$

$$\langle d,d\rangle=\sum_{i=1}^{2} d_i^{\,2}=(-1,-1)(-1,-1)=2$$

Thus $\dfrac{\langle c,d\rangle}{\langle d,d\rangle}=5/2$

Step I : The linear objective function L is given by

$$L = \langle \gamma, x \rangle = \langle \left[\ c - \frac{\langle c,d \rangle}{\langle d,d \rangle} d\ \right], x \rangle$$
$$= \langle \left[\ (-2,-3) - 5/2(-1,-1)\ \right], x \rangle$$
$$= x_1/2 - x_2/2$$

Now we maximize L subject to the same constraints as follows:

$$x_1 + x_2 \le 2$$
$$x_1 + x_2 \ge 1$$
$$x_1 \quad x_2 \ge 0$$

Therefore, inserting slack, surplus and artificial variables we obtain:

(ALP) Minimize $L^* = w$

Subject to

$$x_1 + x_2 + s_1 = 2$$
$$x_1 + x_2 - s_2 + w = 1$$
$$x_{1,} \quad x_2, s_1, s_2, w \ge 0$$

Now ,we construct simplex table of PHASE-I as follows :

PHASE-I

c_B ↓	c_j →	0	0	0	0	-1	B
	Basis↓	x_1	x_2	s_1	s_2	w	
0	s_1	1	1	1	0	0	2
-1	w	1	1	0	-1	1	1
$c_j^* = c_j - z_j$		-1 ↑	-1	0	1	0	-1
0	s_1	0	0	1	1	-1	1
0	x_1	1	1	0	-1	1	1
$c_j^* = c_j - z_j$		0	0	0	0	1	0

Since all $c_j^* \ge 0$ and the artificial variable w is out of basis, it yields an optimal solution for PHASE-I.

PHASE-II

$c_B \downarrow$	$c_j \rightarrow$	1/2	-1/2	0	0	B
	Basis$\downarrow$	x_1	x_2	s_1	s_2	
0	s_1	0	0	1	1	1
½	x_1	1	1	0	-1	1
$c_j' = c_j - z_j$		0	-1	0	1/2	-1/2
0	s_2	0	0	1	1	1
½	x_1	1	1	1	0	2
$c_j' = c_j - z_j$		0	-1	-1/2	0	-1

Since all $c_j' \geq 0$ and the last table yields a sub-optimal solution $x^1 = (2,0)$.

But $Z(x^1) = -4/0$, that is, the maximum value is undefined.

Conclusion:

If the constraint set or the feasible region X is bounded and the denominator is strictly positive for all $x \in X$, each of the four algorithm can successfully solve the LFP problem (see also Example 5.1 of this chapter). The method of Bitran-Novaes[1972] and Swarup [1964] are algorithmically equivalent in the sense that two algorithms select the same non-basic variables to enter the next trial solution and remove the same basic variable from the current solution . One can also observed that the technique of Bitran-Novaes [1972], which is the solution of a sequence of linear programs, only checks for optimality of the fractional programs at points that are optimal solution of intermediary linear programs . It is also observed that primal Simplex type method of Charnes & Cooper, Swarup or Bitran –Novaes are not applicable if the constrains fail the feasibility; whereas Swarup [1965]'s dual type method is not applicable if the constraints fail the optimality

If the denominator $dx + \beta \geq 0$ and the numerator $cx+\alpha < 0$, $\forall x \in X$, the method of Bitran-Novaes may fail (see also Example 5.2 of this chapter); whereas our modified approach of Swarup simplex type method, Swarup dual type method & Charnes-Cooper transformation technique will always recognize and stop at an optimal point , if such a point is reached (see also Example 5.2 of this chapter).

Finally, we conclude that without considering the restrictions on the sign of the denominator of the objective function of LFP problem Charnes-Cooper [1962] transformation technique is more applicable . If the constraint set or the feasible region X is bounded and the denominator is strictly positive for all $x \in X$ Bitran-Novaes [1972] method which involves a sequence of linear programs , to solve a LFP problem it takes more time and labor ; but in the same case Swarup simplex type method solve only a single LFP problem. So, Swarup[1964] simplex type method with our modified approach is best one.

Since large-scale real life LFP problem cannot be solved by hand calculations, it requires computer- oriented solution. Hence, here we generalize computer program (FORTRAN) of all these methods for solving LFP problem. So we may conclude that linear fractional programming method and required computer program a mighty method for large scale optimization problem, where it can be applied.

REFERENCES

1. Bitran, G.R. and Novacs, A.G.: Linear Programming with a Fractional Objective Function, Operations Research, Vol. 21, pp. 22-29, (1972).

2. Charnes , A., & Cooper, W.W.: Programming with Fractional Functionals, Naval Research Logistics Quarterly 9, pp. 181-186,(1962).

3. Dantzig , G.B.: Linear Programming and extension, Princeton University Press, Princeton, N. J., (1962).

4. Dantzig , G.B.: Inductive proof of the simplex method, IBM Journal of research and development, Vol.14, No. 5 , (1960) .

5. Gillet B.E.: Introduction to Operations Research, McGraw Hill, Inc., New York, (1998).

6. Islam, M.A., & Nath. G.: Investigation on some Algorithm for solving Linear Fractional Programming Problems , Bangladesh Sci. Res., 14(1) , pp. 1-11, (1996).

7. Lipschutz, S. & Poe, A.: Programming with FORTRAN including Structural Fortran, McGraw Hill, Inc., New York, (1998).

8. Mayo, E.W. & Cwiakala ,M.: Programmng with FORTRAN 77, McGraw Hill, Inc., New York, (1995).

9. Press, H.W., Teukolsky, A.S., Vetterling, T.W. & Flannery , P.B.: Numerical Recipes in FORTRAN 77, Cambridge University Press, (1992).

10. Swarup, K.: Linear Fractional Programming, Operations Research, Vol. 13, No. 6, pp. 1029-1036, (1964).

11. Swarup, K.: Some Aspects of Linear Fractional functionals Programming, Australian Journal of Statistics, Vol. 6(3), pp. 90-104, (1965).