

TITLE PAGE

APPLICATION OF LINEAR PROGRAMMING (SIMPLEX ALGORITHM) TO  
PRODUCTION MIX PROBLEM

A CASE STUDY OF NIGERIA CARTONS AND PACKAGING COMPANY

BY

*OLUFEMI PIUS OLUGBENGA*  
*PGD/MCS/088/96*

THE DEPARTMENT OF MATHEMATICS /COMPUTER SCIENCE  
FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGER STATE

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF POST  
GRADUATE DIPLOMA IN COMPUTER SCIENCE (PGDCS)

MARCH, 1998

## TABLE OF CONTENT

Title Page . . . . .	i
Approval Page . . . . .	iv
Dedication . . . . .	v
Acknowledgment . . . . .	vi
Abstract . . . . .	vii
1.0 <b>CHAPTER ONE</b> . . . . .	2
1.1 BRIEF HISTORY OF THE COMPANY . . . . .	2
1.2 DEFINITION . . . . .	2
1.3 JUSTIFICATION FOR THE COMPANY . . . . .	3
1.4 APPROACH TO THE PROBLEM . . . . .	4
1.5 AREAS OF APPLICATION . . . . .	4
25.0 <b>CHAPTER TWO</b> . . . . .	5
2.1 INTRODUCTION OF SIMPLEX ALGORITHM METHOD . . . . .	5
2.2 THE LOGIC OF THE SIMPLEX METHOD . . . . .	6
2.3 GENERAL FORM OF LINEAR PROGRAMMING MODEL . . . . .	8
2.3.1 MODEL . . . . .	8
2.3.2 LINEAR PROGRAMMING MODEL . . . . .	9
2.3.3 THE CANONICAL FORM OF LINEAR PROGRAMMING MODELS . . . . .	9
2.3.4 THE STANDARD FORM . . . . .	11
2.3.5 THE STANDARD LINEAR PROGRAMMING . . . . .	12
2.4 CONCEPT OF SLACK AND SURPLUS VARIABLES . . . . .	12
2.4.1 SLACK VARIABLES . . . . .	12
2.4.2 SURPLUS VARIABLES . . . . .	13
2.5 GRAPHICAL SOLUTION . . . . .	13
3.0 <b>CHAPTER THREE</b> . . . . .	17
3.1 OPTIMIZATION OF SIMPLEX ALGORITHM . . . . .	17
3.2 COMPUTATIONAL DETAILED OF SIMPLEX ALGORITHM . . . . .	18
3.3 DEFINITION OF THE VARIABLES IN THE TABLEAU . . . . .	20
3.4 STEPS INVOLVED IN CONTRIBUTION TABLEAU II . . . . .	21
3.5 COMPLETING A THIRD TABLEAU . . . . .	22
3.6 SENSITIVITY ANALYSIS . . . . .	23
3.6.1 SHADOW PRICES . . . . .	23
3.7 SPECIAL CASES OF SIMPLEX ALGORITHM . . . . .	24
3.7.1 INFEASIBLE SOLUTION . . . . .	24
3.7.2 UNBOUNDED SOLUTION . . . . .	25
3.7.3 DEGENERACY . . . . .	27
3.7.4 ALTERNATIVE OPTIMA . . . . .	30

APPROVAL PAGE

This project has been read and approved by the Undersigned on behalf of the department of Mathematics and Computer Science, School of Science and Science Education, the Federal University of Technology, Minna, Niger State having met the requirement, for the award of **Post Graduate Diploma (PGD)** in **COMPUTER SCIENCE**.

(1) -----  
**MR. ISAH AUDU**  
**PROJECT SUPERVISOR**

(2) -----  
**PROF. ADEBOYE**  
**HEAD OF DEPARTMENT**

(3) -----  
**EXTERNAL EXAMINER.**

## **DEDICATION**

This project write up is specially dedicated to the Almighty God for His guidance, love and protection on me through out the rigorous pursuit of the study.

I also unreservedly dedicate this project to the family of Mr. &Mrs. S.A. Oladipupo whose supports were immeasurable.

**OLUFEMI PIUS OLUGBENGA**  
MARCH, 1998

## ACKNOWLEDGEMENT

How does one begin to express love and appreciation to people who by their sense of vision, and understanding have made this project see the light of day. It is only the researcher that knows the worth, substance, and extent of the input of these people towards the realisation of this work.

In this regard therefore, I must confess that I lack the appropriate expression that will convey exactly my heart desires, gratification and due tribute to those whose support, ideas and assistance have directly or indirectly brought this project work to its success. My first gratitude is specially given to God Almighty, whose inspiration and wisdom divinely led me through.

I wish to express my appreciation to my project Supervisor Mr. Isah Audu whose critical appraisals, debates guidance, kind and patient attention and understanding enhanced my attainment of clarity on a number of issues and ideas, and further steered this project work to its successful completion. I will also like to express my thanks to the Head of Department Professor Adeboye for his fatherly advice and attitude. My thanks also goes to the course co-ordinator Prince Rasheed Badamosi and other lecturers in the department for the cooperation received from them during our stay at the University.

My sincere gratitude also goes to the Family of Mr and Mrs Oladipupo for their primal assistance and sense of direction provided when it was really needed.

My appreciation and due gratitude also goes to Honourable NJC Ambassador Mike E. Baafour who has been very instrumental to my achievements. May God bless him.

I will also like to extend my sincere gratitude to the family of Mr and Mrs Oguunibe for their support in attaining this feat. My heart felt gratitude to my sister Mrs Catherine Bisi Taiwo for her tireless efforts in making this work a success. Many thanks to my Uncle Mr. Benjamin Adebayo who was always attentive during distress calls.

I cannot forget very good friends of mine whose co-existence and principle of esprit-de-corp gave more interest and insight to the course of work during our short stay in the school. They are Mr. Dada Olumuyiwa A. (Pally), Tōtimi Adekanbi, Babatunde David, I.A. Bolarinwa, Tunder Adamolekun, Muyiwa Oguntuga, akinbo R.Y. Elvis and host of others.

I would also put on record my gratitude and appreciation for the support and assistance received from Alhaja Bada of Federal ministry of Internal Affairs Abuja, May God bless her (Amen).

Finally my thanks goes to mr. G.B. Bello of Federal Ministry of Water Resources and Rural Development, Abuja for taking his time off to type the Manuscript.

## ABSTRACT

The concept of this research work is to critically and objectively find solution to production mix problem in production companies, thus the choice of the research topic “APPLICATION OF LINEAR PROGRAMMING (SIMPLEX ALGORITHM) TO PRODUCTION MIX PROBLEM”.

Applying linear programming by use of simplex Algorithm, prefers an efficient solution to the various production problems encountered and or envisaged in such companies. Within the content and context of this write up, satisfactory and comprehensive analysis were provided in relative to the chosen case study, thus presenting a suitable system in solving production mix problems in related production companies.

**1.1 BRIEF HISTORY OF THE COMPANY**

The Nigeria Carton and Packaging Manufacturing Company Limited (NICAPACO) was founded by Mr. Jamieson Shu and established in 1972.

Mr. J. Shu who is the Chairman of the Company was until then the Managing Director of Nigeria Glass Company Limited (NIGLASCO) at No. 1 Sapara Street Ikeja.

NICAPACO started on a small scale at Ikeja until it was moved to No. 23 Industrial Avenue Ilupeju. They started with the production of corrugated cartons which was not easy in the first few months of establishment.

After many struggle, they succeeded in producing a large number of corrugated cartons for their customer daily. In 1980, when they realised that the company has improved and discovered that there was a lot of waste product after producing cartons, they decided on Utilizing the waste product in producing egg trays. They also produced tube-core.

In 1986, the company set up another company known as Multipak Nigeria Limited, operating alongside with NICAPACO at 23 Industrial Estate Illupeju. For now Multipak has moved to their own permanent site.

**DEFINITION**

- 1.2 Linear Programmimg is a mathematical technique which is widely used in many fields or various disciplines of the world. It is a term which is very useful to economist, mathematicians, statisticians, and so on. Before going into analyzing linear programming. It is very important to know what actually the term means.

As it is commonly known, the term linear is used to describe the relationship between two or more variables which are directly and precisely proportional. For example, if we



say  $X = F(Y)$ , where  $F$  is a linear function, then any change in  $X$  results in a constant proportional change in  $Y$ . If this were graphed, the relationship would be expressed by a straight line, hence linear.

The term programming implies finding a combination of different kinds of activities which fully exploit the available resources. It makes use of certain mathematical techniques to arrive at the best solution for utilizing the firm's limited resources.

When both words were brought together to become linear programming, it can then be defined as a mathematical techniques for determine the best allocation of a firms limited resources. Mathematicians may be more technical for defining linear programming by stating that it is a method of solving problems in which an objective function are to be maximized or minimized when considering certain constraints. An economists might define linear programming as a method of allocating limited resources in a manner that satisfies the laws of demand and supply for the firms products.

In general form, a linear programming can be defined as an optimization method applicable for the solution of a problem in which the objective function and the constraints appears as a linear function of the decesion variables.

### 1.3 **JUSTIFICATION FOR THE RESEARCH**

This research is being carried out in order to know the product or goods produced by NICAPACO, the process undergone during the production of these goods and the problems encountered, so that the company could be helped to employ the best method of linear programming in carrying out production.

A workable program will be developed to help NICAPACO determine the most profitable combination of products they manufactured and also the most efficient method of machine loading. Simplex algorithmin guarantees finding a better solution at each step and an optimal solution in a finite number of steps.

#### 1.4 **APPROACH TO THE PROBLEM**

Before any method of linear programming can be applied to any problem, some basic requirements are necessary, which are as follows:-

1. There must be a well defined objective function  
- it must be defined mathematically.
2. There must be an alternative course of action.
3. Linear objective function and linear constraints must be expressed mathematically.
4. The variables must be interrelated.
5. Resources must be limited in supply.

The five basic requirements stated above are the very first steps it takes in applying linear programming to product mix problem in any manufacturing company.

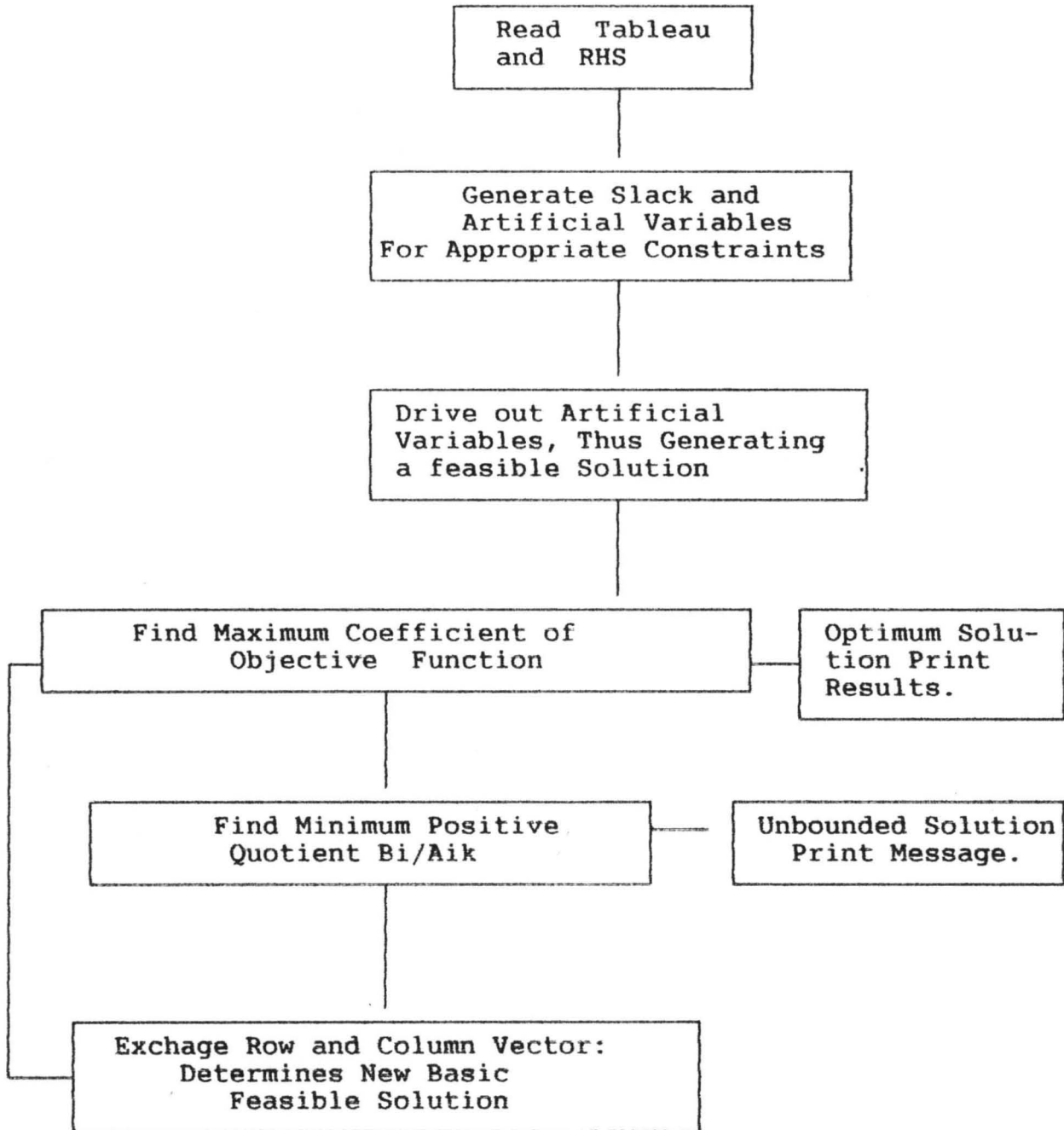
There are various method of linear programming which can be employed to solve problem, they are graphical method, Basic feasible solution, simplex method and so on. Simplex algorithm is being applied in this project and it involves many steps, which will be treated in full detail. It involves finding a better solution at each step and an optimal solution in a finite number of steps.

#### 1.5 **AREAS OF APPLICATION**

Linear programming has been successfully applied in many areas. In physical distribution in determining the most economic and efficient manner of locating manufacturing plants and distribution centers. In inventory scheduling, it allows arrangement of raw materials and semi-finished goods to minimize the firm's capital investment while maximizing efficient production flow. In military it is used in optimum allocation of men and materials. It is also used in oil refinery for optimum product mix.

Linear programming is also useful in food processing industry for optimal shipping plan for the different manufacturing plants to the various warehouse.

**SIMPLEX (SIMPLEX ALGORITHM) LOGIC DIAGRAM.**



In figure 2.2.1, the steps are

- (1) Obtain any basic feasible solution to start with
- (2) Check neighboring solutions to see if they are better
- (3) If there is a better neighboring solution, move to it
- (4) Repeat steps (2) and (3) until improvement is possible.

## 2.3 GENERAL FORM OF LINEAR PROGRAMMING MODEL

### 2.3.1 MODEL

Models are representation of real objects or situations. The mathematical model represent a problem by a system of symbols and mathematical expressions or relationships.

Experimenting with models requires less time and it's less expensive than experimenting with the real objects or situation. Model also reduces the risk associated with experimenting with the real situation.

The accuracy of the conclusions and decisions based on a model are dependent upon how well the model represent the real situation. Under the mathematical models we have the deterministic and stochastic types.

**DETERMINISTIC MODEL:** If all uncontrollable inputs to a model are known and cannot varies, the model is refered to as a deterministic model.

**STOCHASTIC MODEL:** If any of the uncontrollable inputs are uncertain and subject to the variation, the model is refered to as a stochastic model.

### 2.3.2 LINEAR PROGRAMMING MODEL

The general linear programming problem can be described as:-

Given a set of  $M$  linear inequalities or equations with  $N$  variables, we wish to find non-negative values of these variables which will satisfy the constraints and optimize (maximum or minimum) some linear functions of the variables.  $M$  can be greater than, less than or equal to  $N$ .

Mathematically this means:-

$$\text{Optimize } Z_0 = P_1x_1 + P_2x_2 + \dots + P_nx_n$$

$$\text{S.T. } a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n (\leq, =, \geq) b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n (\leq, =, \geq) b_2$$

"

"

"

"

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n (\leq, =, \geq) b_m$$

$$x_1, x_2, \dots, x_n \geq 0$$

Where,

$P_j$  is the worth per unit activity  $j$

$a_{ij}$  is the amount of resource  $i$  that must be allocated to each unit activity of  $x_j$

$b_i$  is the available amount of resources  $i$ ., These are constants determine form from the statement of the problem  $j$  are the decision variables.

### 2.3.3 THE CANONICAL FORM OF LINEAR PROGRAMMING MODELS

$$\text{Maximize } Z_0 = \sum_{j=1}^n P_j X_j$$

$$\text{s.t. } \sum_{j=1}^n a_{ij}x_j \leq b_i, \quad \begin{matrix} i=1, 2, 3, \dots, m \\ j=1, 2, 3, \dots, n \end{matrix}$$

$$x_j \geq 0$$

The characteristics of the canonical form are :-

- (a) All decision variables are either positive or zero
- (b) The constraints are all of less than or equal to type
- (c) The objective function is the maximization type.

To convert any linear programming problem to the canonical form, the following five transformations are used.

- (1) The minimization of a function,  $f(x)$ , is mathematically equivalent to the maximization of the negative expression of this function i.e.  $-f(x)$  that is,

$$\text{Minimize } C = P_1x_1 + p_2x_2 + \dots + p_nx_n$$

is equivalent to

$$\text{maximize } Z_0 = -C = -P_1X_1 - P_2X_2 - \dots - P_nX_n$$

Where  $C = -Z_0$

- (2) To change an inequality in one direction to an inequality in the opposite direction, multiply both side of the inequality by -1.

$$\text{i.e. } a_1x_1 + a_2x_2 (\leq, \geq) b$$

ie equivalent to

$$- a_1x_1 - a_2x_2 (\leq, \geq) -b$$

- (3) Replace an equation by two (weak) inequality in the opposite direction.

$$\text{i.e. } a_1x_1 + a_2x_2 = b$$

is equivalent to

$$a_1x_1 + a_2x_2 \leq b \text{ and } a_1x_1 + a_2x_2 \geq b$$

e.g.  $a_1x_1 + a_2x_2 \leq b$  }

$a_1x_1 + a_2x_2 \geq b$  } Weak Equations

or

$a_1x_1 + a_2x_2 \leq b$  and  $-a_1x_1 - a_2x_2 \leq -b$

equation

- (4) Change an inequality constraints with it's left hand side in the absolute form into 2 regular inequalities

i.e  $|a_1x_1 + a_2x_2| \leq b$  is equivalent to  $a_1x_1 + a_2x_2 \geq -b$  and  $a_1x_1 + a_2x_2 \leq b$

- (5) If a variable is unconstrained (unrestricted) in sign, replace it by 2 non-negative variables i.e if  $x$  is unrestricted in sign, it can be replaced by  $(x^+ - x^-)$  where  $x^+ \geq 0$  and  $x^- \geq 0$ .

### 2.3.4 THE STANDARD FORM

The standard form of linear programming has the following features

- (1) All constraints are equations except non-negativity constraint which remain inequalities.
- (2) The right hand side element of each constraint equation is non-negative.
- (3) All decision variables are non-negative
- (4) The objective function is of the minimization or maximization type.

The standard form problem can be represented as follows:

equation:

$$\text{Max } Z_0 = \sum_{j=1}^n P_j x_j$$

$$\text{s.t. } \sum_{j=1}^n a_{ij} x_j \leq b_i \quad (i=1, 2, 3, \dots, m)$$

$$x_j \geq 0$$

**2.3.5 THE STANDARD LINEAR PROGRAMMING PROBLEM IN MATRIX VECTOR NOTATION**

Optimize  $Z = CX$   
 S.T  $AX = b$   
 $X \geq 0$   
 $b \geq 0$

Where A is an (m x n) matrix, X is an (n x 1) column vector, b is an (m x 1) column vector and C is an (1 x n) row vector.

equation: i.e A (m x n)  $\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ \vdots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix}$   $\begin{matrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{matrix}$

$b(m \times 1)$   $\begin{matrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{matrix}$   $C(1 \times n) = (C_1 \ C_2 \ \dots \ C_n)$

A is referred to as the coefficient matrix, X is the decision vector, b is the requirement vector, C is the profit (cost) vector.

**2.4 CONCEPT OF SLACK AND SURPLUS VARIABLES**

**2.4.1 SLACK VARIABLE**

Any inequality constraint can be changed to equation by augmenting (Adding or Subtracting) the left hand side of such a constraint by a non-negative variable. This new variable is called a slack variable which is added if the inequality is of the less than or equal to type.



i.e  $a_1x_1 + a_2x_2 \leq b$  ( $b \geq 0$ )  
 $a_1x_1 + a_2x_2 + S = b$  where  $S \geq 0$

#### 2.4.2 SURPLUS VARIABLES.

In linear programming terminology, any excess quantity corresponding to a greater than or equal to constraint is referred to as the surplus associated with the constraint.

e.g  $a_1x_1 + a_2x_2 \geq b$   
 $a_1x_1 + a_2x_2 - s = b$  where  $s \geq 0$

#### 2.5 GRAPHICAL SOLUTION

Graphical method is a geometrical representation of linear programming problem. This method can be easily applied in cases involving two constraints and several variables.

However, in real life problems, variables are normally more than two, but for greater insight of what happens in practical life this method is considered.

#### Example:

A company produces two items (I & II) by machining, assembling and painting. The different times require for each product, the processing time available for each stage of the production and the profit on each item are given below.

STAGE	PROCESSING TIME		MAX. TIME AVAILABLE PER WEEK
	(min.)		(min.)
	I		II
Machining	10	5	2500
Assembling	4	10	2000
Painting	1	15	450
Profit	₦50	₦100	

In solving this problem, construction is made of the set of points  $(X_1, X_2)$  which represents a feasible solution of the problem.

This set of points must satisfy all the constraints

$$10x_1 + 5x_2 \leq 2500$$

$$4x_1 + 10x_2 \leq 2000$$

$$x_1 + 1.5x_2 \leq 450$$

Which requires that all points be either on or below the line,

$$10x_1 + 5x_2 = 2500$$

$$4x_1 + 10x_2 = 2000$$

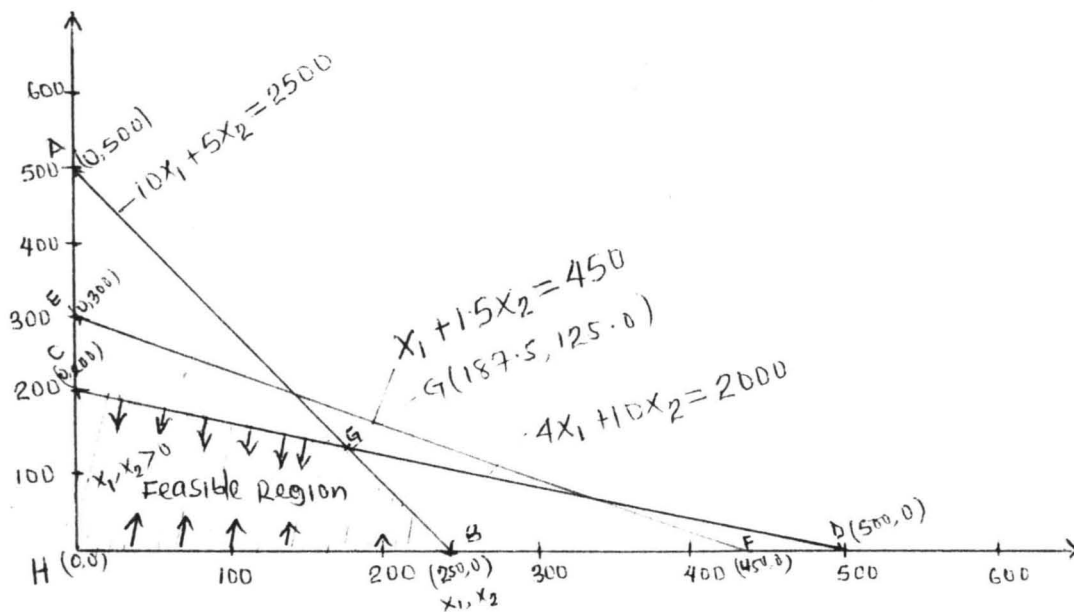
$$x_1 + 1.5x_2 = 450$$

The constraints

$$x_1, x_2 \geq 0.$$

Graph

FIG 2.5.1



From the above figure, the shaded area is called the feasible region, points b,g,c and h are called extreme points of the constraint set and are the feasible to the problem.

To find the optimal value

Vertex of feasible	item I	item II	Objective function
<b>Region</b>			<b><math>50 \times 1 + 100 \times 2</math></b>
C	0	200	20,000
G	187.5	125.0	21,875
B	250	0	12,500
H	0	0	0

After substituting the value of the set of points in the objective function, it is seen that point G with (187.5, 125.0) corresponds to the given optimal feasible solution: that is

The optimal value  $P = 21,875 =$

The geographical interpretation of this is that the company should produce 187.5 or 188 worth of item 1 per week and 125 worth item II per week with a resulting income of  $=21,875 =$  per week.

### 3.2 COMPUTATIONAL DETAILED OF SIMPLEX ALGORITHM

equation

$$\begin{aligned} \text{Max. } P &= 50X_1 + 100X_2 \\ \text{S.T } 10X_1 + 5X_2 &\leq 2500 \text{ (dept 1)} \\ 4X_1 + 5X_2 &\leq 2000 \text{ (dept 2)} \\ X_1 + 1.5X_2 &\leq 450 \text{ (dept 3)} \\ X_1 + X_2 &\geq 0 \end{aligned}$$

In order to use the simplex method, it is necessary first to convert the three inequalities into equations for the departments.

This can be performed by adding slack variable for each department, that is, add to each inequality a variable that will take up the slack or time not used in a department. The following slack variables (in minutes) will be used. S1, S2 and S3 represent unused time in dept. 1,2 and 3 respectively.

It should be noted that the for going slack variables are positive, caused by the department constraint being equal to or less than ( $\leq$ ). It would be negative if the constraint were equal to or greater than ( $\geq$ )

The slack variable S1, is equal to the total amount of time available in department 1 or 2500 minutes, minus any minutes used in processing product I and II. The same type of reasoning is applicable to S2 and S3. The original inequalities for the three departments now can be expressed by writing equation for the slack variables as follows:

$$S1 = 2500 - 10x_1 - 5x_2$$

$$S2 = 2000 - 4x_1 - 10x_2$$

$$S3 = 450x_1 - 1.5x_2$$

However, to state the objective function and these three equations in acceptable form we have:

$$\text{Max. } Z = 50x_1 + 100x_2 + 0S_1 + 0S_2 + 0S_3$$

$$\text{S.T } 2500 \text{ min.} = 10x_1 + 5x_2 + S_1 + 0S_2 + 0S_3$$

$$2000 \text{ min} = 4x_1 + 10x_2 + 0S_1 + S_2 + 0S_3$$

$$450 \text{ min.} = x_1 + 1.5x_2 + 0S_1 + 0S_2 + S_3$$

In addition to converting the inequalities into equations by adding slack variables, the simplex method requires that any unknown which appears in one equation must appear in all equations, the unknown that do not affect an equation are written with zero coefficient.

To simplify the handling of the equations, they can be placed in a tabular form as shown below:

TABLEAU 1

<b>Basis</b>	<b>C<sub>j</sub></b>	<b>50</b>	<b>100</b>	<b>0</b>	<b>0</b>	<b>0</b>	
S1	0	10	5	1	0	0	2500
S2	0	4	10	0	1	0	2000
S3	0	1	1.5	0	0	1	450
	Z <sub>j</sub>	0	0	0	0	0	
	c <sub>j</sub> - z <sub>j</sub>	50	100	0	0	0	

Z<sub>j</sub> is the contribution lost per unit.

c<sub>j</sub> - Z<sub>j</sub> is the net contribution per unit.

### 3.3 DEFINATION OF THE VARIABLES IN THE TABLEAU 1.

Starting with the left- hand column in "tableau 1", the BASIS contains the variables in the solution which are used to determine total contribution. In the initial solution, no product are being made. The values for S1, S2 and S3 must contain all of the unused time in the problem, which is found in the last column of the table. The starting solution will be zero contribution since no units of product I & II are beingproduced i.e ( $0 \times 2500 + 0 \times 2000 + 0 \times 450$ ) in the zj row.

The second column, the "cj" column contains the contribution per unit for slack variables S1, S2 and S3. The zero indicates that the profits are not made on unused time in a department, but on time used.

The body matrix consists of the coefficient for the real product variables. The identity matrix in the first simplex tableau represents the coefficient of the slack variables that have been added to the original inequalities to make the equations. As stated previously any unknown that occurs in one equation must appear in all equations but with zero coefficient so as not to affect the equation.

Referring to the element in the S1 column. First row, the 1 indicates that in order to make 1 minute of S1 available, it would be necessary to give up one of the 2500 minutes in the initial solution, the zero in the S2 column first row indicates that making one minute of S2 available for other purposes has no effect on S1 (slack time of dept 1). The logic used for S2 is also applicable to the S3 column first row. The same type of rationale is applicable to the next two rows S2 and S3.

In the last column, since no units of I & II are being manufactured, the first solution is  $X1 = 0, X2, = 0, S1 = 2500,$   
 $S2 = 2000, S3 = 450.$

The last two rows of the first simplex tableau are used to determine whether or not the solution can be improved upon.

The  $z_j$  row values under the solution indicate an initial solution of zero contribution for the firm, the other five of 0 are the amount by which contribution would be reduced. Another way of defining  $z_j$  row for the five variables is the contribution lost per unit.

For example:

If we desire to make one unit of product I, the coefficient (10, 4, 1 ~) in the body matrix tells us, we must give up 10 minutes of S1 (dept. 1) unused time and one minute of S3 (dept. 3) unused time.

Since slack time is worth 0 per minute, there can be no reduction in contribution, the calculation for how much contribution is lost by adding one unit of product I to production is: No. of minutes of S1 given up = 2, multiplied by contribution per unit S1 =  $2 \times 0 = 0$ . The same is applicable to S2 and S3 and everything summed up.

### 3.4 **STEPS INVOLVED IN CONSTRUCTING TABLEAU II**

There are four steps involved in simplex method.

Step 1- Selecting of the column with the highest positive value.

This is done in the  $c_j - z_j$  row reveals that the largest positive value is 100.

A positive value indicate that a greater contribution can be made by the firm while a negative value indicates the amount by which contribution would decrease if one unit of the variable for that column were brought into the solution. The largest positive amount or number in the last row is selected as the optimal column since we want to maximize total contribution.

When no more positive values remain in the  $c_j - z_j$  row and values are zero or minus in a maximization problem, total contribution is in it's greatest value.

### 3.7 SPECIAL CASES OF SIMPLEX ALGORITHM

There are some special cases of simplex algorithmic method, amongst are:-

- (I) INFEASIBILITY
- (II) UNBOUNDEDNESS
- (III) DEGENERACY
- (IV) ALTERNATIVE OPTIMA.

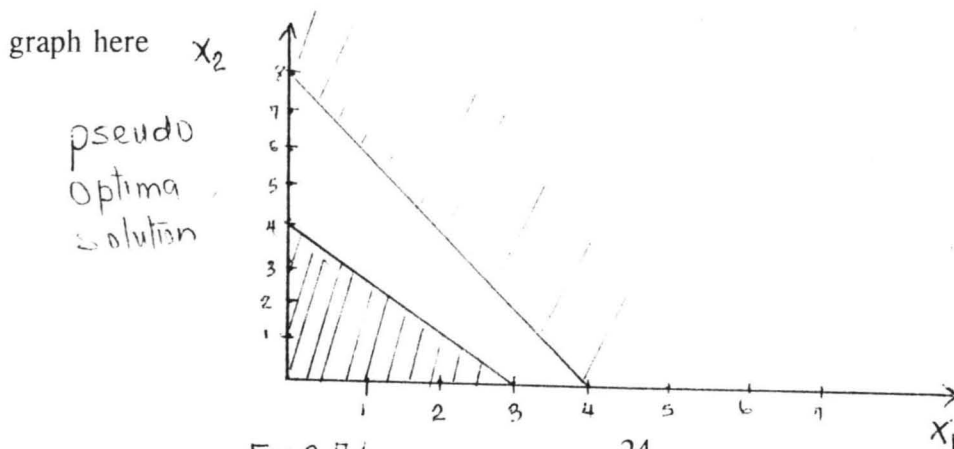
#### 3.7.1 INFEASIBLE SOLUTION

A linear programming problem is infeasible if there is no solution which satisfies all the constraints and non-negativity condition simultaneously.

Infeasibility is recognised when the stopping rule indicates an optimal solution and one or more of the artificial variables remain in the solution at a positive level (value).

The following example illustrates the case of an infeasible solution space. Example:

$$\begin{aligned} \text{MAXIMIZE } Z &= 3x_1 + 2x_2 \\ \text{S.T.} \quad &2x_1 + x_2 \leq 8 \\ &4x_1 + 3x_2 \geq 12 \\ &x_1, x_2 \geq 0 \end{aligned}$$





### 3.7.2 UNBOUNDED SOLUTION

For maximisation problems we say that a linear program is unbounded if the solution may be made infinitely large without violating any constraints.

Unboundedness is recognised when all the  $a_{ij} \leq 0$  in column  $j$ , and the simplex method indicates that variable  $X_j$  is to be introduced into the solution.

Unboundedness in a model can point to only one thing, the model is poorly constructed. The most likely irregularities in such models are:

- (1) One or more non redundant constraints are not accounted for. The parameters (constants) of some constraints are not estimated correctly.

To show unboundedness, we consider the following example.

Example:- (unbounded objective value)

$$\begin{aligned} \text{Maximize } & Z = 2x_1 + x_2 \\ \text{S.T. } & x_1 - x_2 \leq 10 \\ & 2x_1 \leq 40 \\ & x_1, x_2, \geq 0 \end{aligned}$$

#### INITIAL TABLEAU

Basic	X1	X2	V	U	SOLUTION
P2	1	-1	1	0	10
P3	2	0	0	0	40
	-2	-1	0	0	0

We can notice that all the constraints coefficient under  $X_2$  are negative or zero, meaning that  $X_2$  can be increased indefinitely without violating any of the constraints.

Since each unit increase in  $X_2$  will increase  $z$  by 1, an infinite increase in  $X_2$  will also result in an infinite in  $z$ . Thus we conclude without further computation that the problem has no unbounded solution.

GRAPH

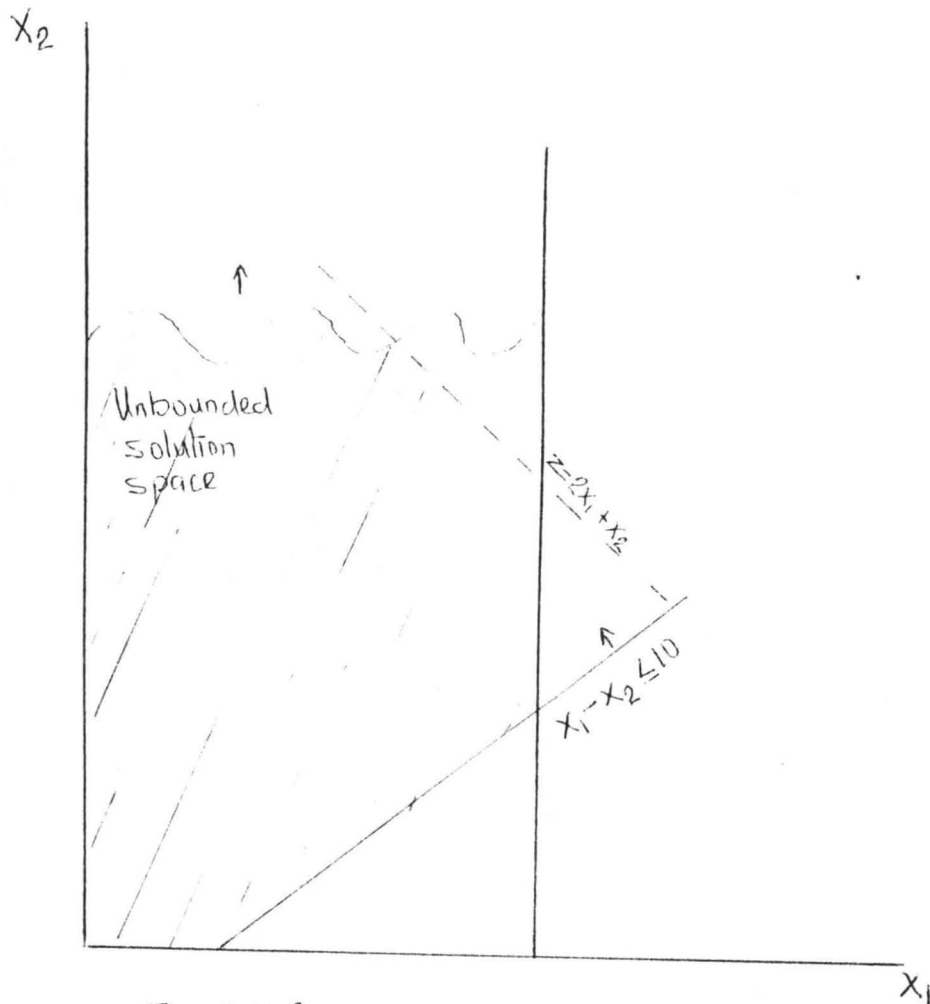


Fig 3.7.2  
8

The general rule for recognizing unboundedness is as follows:

If at any iteration the constraint coefficients of a non-basic variable are non-positive, then the solution space is unbounded in that direction. In addition, if the objective coefficient of that variable is negative in the case of maximization or positive in case of minimization, then the objective value also is unbounded.

### 3.7.3 DEGENERACY

A linear programming problem is said to be degenerated if one or more of the basic variables has a value of 0. The instance of degeneracy is recognised when there is a tie in the minimum ratio column.

In the application of the feasibility condition, tie for the minimum ratio may be broken arbitrarily for the purpose of determining leaving variable. When this happens, however, one or more of the basic variables will necessarily equal to zero in the next iteration. In this case we say that the new solution is degenerate.

From the practical stand point, the condition reveals that the model has at least one redundant constraint. To be able to provide more insight into the practical and theoretical impacts of degeneracy, I consider two numeric examples. The graphical illustration gives a better understanding of this special situation.

Example (Degenerate Optimal Solution)

$$\begin{aligned} \text{Maximize } P &= 3x_1 + 9x_2 \\ \text{S. T.} \quad & x_1 + 4x_2 \leq 8 \\ & x_1 + 2x_2 \leq 4 \\ & x_1, x_2, \geq 0 \end{aligned}$$

Using U and V as slack variables, we list the simplex iteration for the solution. In the

The practical implication of degeneracy is seen in the graphical solution below. Three lines passes through optimum  $(X_1 = 0, X_2 = 2)$ .

Since this is a two dimensional problem, the point is said to be overdetermined, since we only need two line to identify it. For this reason, we conclude that one of the constraints is redundant.

From the theoretical point of view, degeneracy also deals with the phenomenon of cycling.

GRAPH HERE

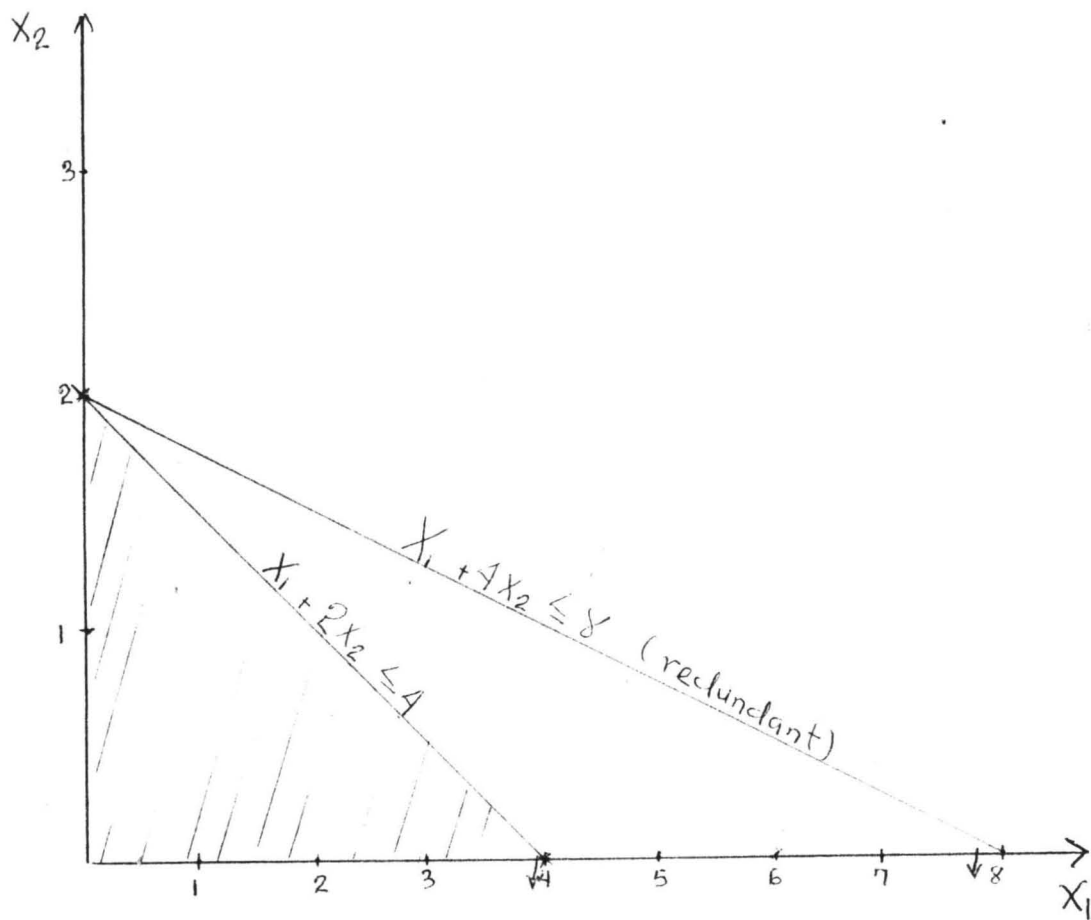


Fig 3.7.3

### 3.7.4 ALTERNATIVE OPTIMA

A linear programming problem with two or more optimal solution is said to have alternative optima.

Alternative optima solution is recognised when in the final simplex tableau  $c_j - z_j = 0$  for one or more variables not in the solution (non-basic-variables).

The next example shows that there is an infinity of such solutions. The example also show the practical significance of encountering alternative optima.

Example

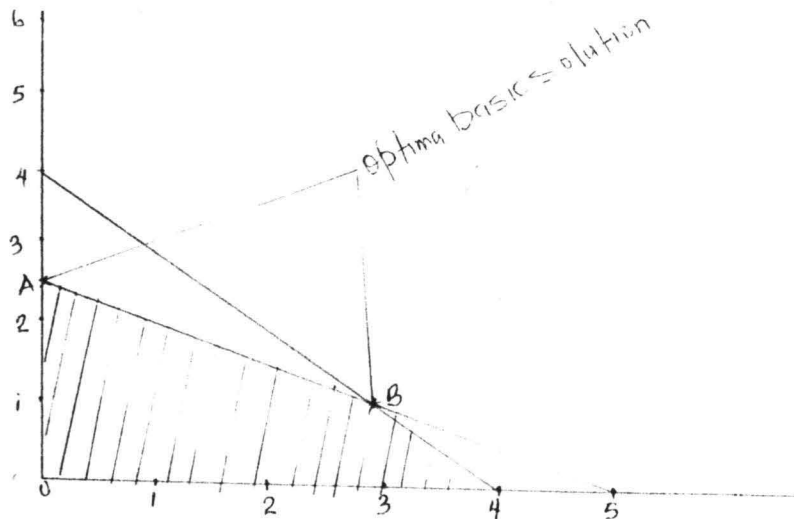
$$\text{Maximize } P = 2 X_1 + 4 X_2$$

$$\text{S.T. } X_1 + 2 X_2 \leq 5$$

$$X_1 + X_2 \leq 4$$

$$X_1, X_2 \geq 0$$

GRAPH



The graph above shows how alternative optima can arise in LP when the objective function is parallel to a binding constraint. Any point on the line segment AB represent an alternative optimum with same objective value  $Z = 10$

Basis	X1	X2	U	V	SOLUTION
P3	1	2	1	0	5
P4	1	1	0	0	4
	-2	-4	0	0	0

**STARTING ITERATIONS**

1	P2	$\frac{1}{2}$	1	$\frac{1}{2}$	0	$\frac{5}{2}$
(OPTIMUM)	P4	$\frac{1}{2}$	0	$-\frac{1}{2}$	1	$\frac{3}{2}$
		0	0	2	0	10

2

(ALTERNATIVE

OPTIMUM)	P1	1	0	-1	2	3
	P2	0	1	1	-1	1
		0	0	2	0	10

We know how alternative options exist from the table because, the coefficient of (non-basic) X1 is zero indicating that X1 can enter the basis solution without changing the value of but causing a change in the value of the variables. Iteration 2 does just that, letting X1 enter the basic solution which will force X4 to leave. This results in the new solution set ( $X1 = 3$ ,  $X2 = 1$  and  $= 10$ ).

In practice, knowledge of alternative optima is useful because it gives management the opportunity to choose the solution that best suits their situation without experiencing any deterioration in the objective value.

If the example represents a product mix solution, it may be advantageous from the standpoint of sales competition to produce two products rather than one. In this case the solution at B would be recommended.

## 4.0 CHAPTER FOUR

### 4.1 ANALYSIS OF COMPUTATIONAL RESULT CASE STUDY ON

#### NICAPACO

Production Mix is simply analysed as the combination and or mixture of various materials in the production of goods, or set of goods in a production company.

Further to, production mix involves an all embracing utilization through accurate combination of required resources, within an efficient frame work towards producing the best product(s).

NICAPACO is a manufacturing company that produces egg trays, cartons and tube cores. They have so many resources which have been grouped into three departments.

There is the slitting machine time in the cutting department, this machine is used for cutting the paper rolls into different sizes, the long machine time in the gluing department, this machine is used for gluing the paper rolls together as thick as the customers want it and other machine time in the finishing department like the stitching, moulding, printing and creasing machine. In this department all the finishing touches to the produce are made.

In NICAPACO, the total labour hours per day scheduled for slitting machine department is 1000 hours. While 1400 hours and 2500 hours respectively were scheduled on long machine dept and other machine in the finishing department.

Each eggs tray produced requires two hours of cutting time, three hours of gluing time and two hours of finishing time. So also each cartons produced requires four hours of cutting time, two hours of gluing time and two hours of finishing time. And out putting each tube core requires two hours of cutting time, two hours of gluing time and one hour of finishing time.

The accounting department analysing the cost and sales figure state that each pack of egg tray produced yeild eighteen naira contribution to profit while each cartons yields twelve naira contribution and each tube core yields fifteen naira contribution to profit.

#### 4.2 DEVELOPING A RELATIONSHIP TO DESCRIBE THIS RESTRICTIONS.

	EGG TRAY	CARTONS	TUBE CORES	MAX. HRS OF WORK PER DAY
Slitting machine	2	4	2	1000
Long machine	3	2	2	1400
Other machine	2	2	1	2500

Let X1 represent egg tray

Let X2 represent Carton

Let X3 represent Tube core

$$\text{Slitting machine} \quad 2X_1 + 2X_2 + 4X_3 \leq 1000$$

$$\text{Long machine} \quad 3X_1 + 2X_2 + 2X_3 \leq 1400$$

$$\text{Other machine} \quad 2X_1 + 2X_2 + X_3 \leq 2500$$

$$X_1, X_2, X_3 \geq 0.$$

#### 4.3 CREATING OBJECTIVE FUNCTION FOR THE PROBLEM

From the result of the cost and sales analysis figure the following function were got.

$$\text{Max. P} \quad = \quad \text{₦}18X_1 + \text{₦}12X_2 + \text{₦}15X_3$$

$$\text{S. T} \quad 2X_1 + 2X_2 + 4X_3 \leq 1000$$

$$3X_1 + 2X_2 + 2X_3 \leq 1400$$

$$2X_1 + 2X_2 + X_3 \leq 2500$$

$$X_1, X_2, X_3 \geq 0.$$



#### 4.4 SOLVING THE PROBLEM USING THE SIMPLEX ALGORITHM METHOD

Before the system can be solved, the inequalities have to be changed to equations by adding slack variables.

Let  $S_1$  represent the unused time in cutting dept.

$S_2$  represent the unused time in gluing dept.

$S_3$  represent the unused time in finishing dept.

$$\begin{aligned} \text{equation} \quad \text{Max P} \quad & \text{N}18X_1 + \text{N}12X_2 + \text{N}15X_3 \\ \text{S.T} \quad & 2X_1 + 4X_2 + 2X_3 + S_1 + 0S_2 + 0S_3 = 1000 \\ & 3X_1 + 2X_2 + 2X_3 + 0S_1 + S_2 + 0S_3 = 14000 \\ & 2X_1 + 2X_2 + X_3 + 0S_1 + 0S_2 + S_3 = 25000 \end{aligned}$$

TABLEAU 1

		X1	X2	X3	S1	S2	S3	Solution
BASIS	CJ	18	12	15	0	0	0	
S1	0	2	4	2	1	0	0	1000
S2	0	3	2	2	0	1	0	1400
S3	0	2	2	1	0	0	1	2500
	ZJ	0	0	0	0	0	0	
CJ-ZJ		18	12	15	0	0	0	

ITERATION 1 (TABLEAU II)

		X1	X2	X3	S1	S2	S3	SOLU TION
BASIS	CJ	18	12	15	0	0	0	
S1	18	1	8/3	2/3	1	-2/3	0	66.6
X1	18	1	2/3	2/3	0	1/3	0	466.7
S3	0	0	2/3	1/3	0	-2/3	1	1566.6
	ZJ	18	12	12	0	6	0	8400.6
CJ-ZJ		0	0	3	0	-6	0	

ITERATION II (TABLEAU III)

		X1	X2	X3	S1	S2	S3	
BASIS	CJ	18	12	15	0	0	0	
X3	15	0	12	1	3/2	-1	0	99.9
X1	18	1	-22/3	0	-1	1	0	400.1
S2	0	0	-10/3	0	-1/2	-1/3	1	1500
	ZJ	18	48	15	4.5	3	0	
CJ-ZJ		0	-36	0	-4.5	-3	0	8700.3

4.5 LIMITATION OF SIMPLEX ALGORITHM METHOD OF LINEAR PROGRAMMING.

There are some cautions or difficulties associated with every mathematical method. Linear programming is no exception. The objective function and constraints can change overnight due to internal or external factors. There is the necessity of keeping data current. The more rigidly the data adheres to the reality of the situations the more reliable the solution will be.

In using this method of linear programming (simplex algorithm) it is very necessary to be sure that there is a practical application for it and even though the problem is correctly stated and formulated mathematically, there may be some limiting factors from practical point of view. For example if not enough time has been allotted to the proper collecting of data, garbage in garbage out will result.

Another example is if the computer program available cannot handle all the constraints. Constraints will have to be dropped which could render the output basically unusable.

Where the objective function and constraints are non-linear, extreme caution must be taken when applying linear programming.

The misapplication of linear programming under non-linear conditions usually results in an incorrect solution.

#### 4.6 GENERAL ANALYSIS OF RESULT

The tableau III above leads us to the subject of shadow prices because of the negative values present in the  $C_j - Z_j$  row.

The subject of shadow prices makes it known to us exactly how much a firm is willing to pay to make additional resources available.

It shows if one more hour of machine time is worth some amount of money, also if it is worth while to pay workers an overtime rate to stay one extra hour each night in order to increase production output. All the above information about the worth of additional resources is valuable management information.

The " $Z_j$ " column contains the contribution per unit for slack variables  $S_1$ ,  $S_2$ , and  $S_3$ . The zero indicates that the contribution per unit is zero. The rationale is that profit are

not made on unused time in a department but on time used. The  $Z_j$  values for the three slack variables are 4.5, 3 and 0 respectively. Thus the shadow price for cutting department is ₦4.5, gluing department ₦3 and finishing department is ₦0. If the management has an opportunity to schedule overtime, it is clear that scheduling that time in the cutting department will make the highest contribution to profit of ₦4.5 per minute.

5.1 **CONCLUSIONS.**

Linear programming (LP) is the most powerful method of constrained optimization available. Its power is staggering when compared with other approaches.

Linear programs entirely deal with problems involving tens of thousands of variables and thousand of constraints. These kinds of problems are virtually intractable by other method of analysis. The traditional methods of constrained optimisation for example, require the solution of systems of partial differential equations notoriously time - consuming and difficult even for small problems.

Linear programming is also particularly attractive for design because it automatically provides extensive information on the sensitivity of the optimal design to different formulations of the problem. This feature is most important, because of our inherent uncertainty about the precise parameters of any situation.

Linear programming is so powerful because it exploits the computers ability to execute simple calculations, such as additions and multiplications, very quickly.

By representing a design problem by a system of linear equations it implies an optimisation procedure that consists of a long series of solutions to these linear equations. This task is both simple - minded and tedious - an ideal combination for digital computer.

As with all mathematical programming techniques, linear programming works because it assumes that certain assumptions can be made about a problem. These assumptions establish a mathematical structure for the problem which can be solved by a particular process. Linear programming assumes that the problem can legitimately be described

or approximated by linear, additive and continuous function. Linear programming as any other mathematical programming technique, is thus limited to specific classes of problems, those that meet its assumptions.

The standard form of linear program consists of two parts, an objective function and constraints. The objective function is an equation that defines the quantity to be optimised. For linear programming this quantity must be a one dimensional scalar quantity.

The variables in the objective function, the  $X_i$  are known as decision variables because we seek to make decisions about them so as to optimise the objective. This name is useful because it focuses attention on the idea that the arguments of the objective functions are the decisions we have. This is most helpful when we try to formulate a real problem.

## 5.2 **RECOMMENDATION:**

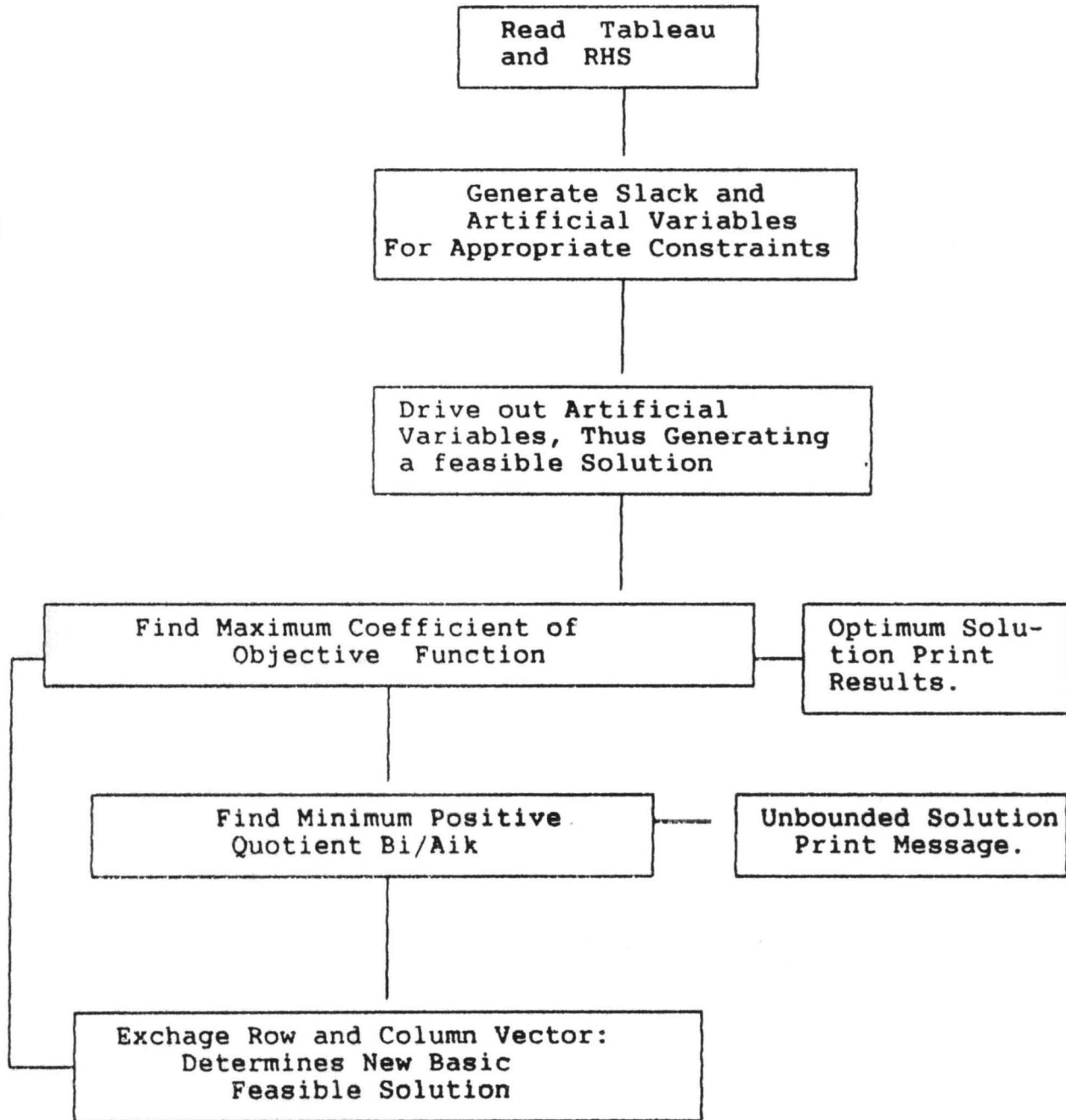
Within the analysis put across in this project work, it could be seen that a lot of factor(s) are inextricably intertwined to give rise to a successful department in the company and the labour force.

It could therefore be observed from the analysis that in the third department which is the finishing department, there were unused time, which means the time allocated to this department is too much.

It will therefore be more of interest and pay off if the management of this company could reduce the time allocated to this department and recruit more workers in order to meet the production level in this department.

However, the optimal value  $P = \text{N}8700$ . This means it is advisable for the company to cut 400 worth of cartons in a day and finish at least 100 cartons in the finishing department with a resulting income of  $\text{N}8700$  per day

**SIMPLEX (SIMPLEX ALGORITHM) LOGIC DIAGRAM.**



## REFERENCES:

1. Bazaraa M.S. and J.J. Jarvis     "Linear Programming and Network flows" (New York John Wiley & Sons Inc. 1974)
2. Edward T. Dowling               "Mathematics for Economists" (Schaum's outline Series in Economics)
3. Kemeny J.G. and Kurtz T.E.     "Basic Programming" Wiley, Third Edition, 1980.
4. Makower M.S. and Williamson E. "Operation Research" (E.U.P. Third edition)
5. Richard de neüfville,           "Applied system Analysis - Engineering Planning & Technology Management"     Massachusetts Institute of Technology (Mc. Graw Hill Pub. Co. Ltd N.Y.)
6. Strum, J.E.                       "Introduction to Linear Programming" (San Francisco; Holden Day, Inc; 1972.
7. Zoints S.                         "Linear and Integer Programming and Network flows" (New York John Wiley & Sons Inc. 1974).



```

1000 CLS : COLOR 2, 0, 3
1010 PRINT : PRINT "LINEAR PROGRAMMING LP (" : PRINT : P
"the linear programming problem looks like this"
1020 PRINT "a11X1 + a12X2 + a13X3 + ..... + a1nXn )= b1"
1030 PRINT "a21X1 + a22X2 + a23X3 + ..... + a2nXn )= b2"
1040 PRINT "a31X1 + a32X2 + a33X3 + ..... + a3nXn = b3"
1050 PRINT "....."
1060 PRINT "....."
1070 PRINT "am1X1 + am2X2 + am3X3 + ..... + amnXn (= bm"
1080 PRINT "to maximize Z = c1X1 + c2X2 + c3X3 +..... + cnXn"
1090 LOCATE 22, 35: COLOR 0, 7: PRINT "press any key to continue....."
1100 RK$ = INPUT$(1): RK$ = "sleep 0"
1101 COLOR 2, 0, 3
1110 CLS : PRINT : PRINT : PRINT : PRINT "the data should be entered in thi
er"
1120 PRINT " a11,a12,a13,.....,a1n"
1130 PRINT " a21,a22,a23,.....,a2n"
1140 PRINT " a31,a32,a33,.....,a3n"
1150 PRINT " ....."
1160 PRINT " am1,am2,am3,.....,amn"
1170 PRINT " c1,c2,c3,.....,cn"
1180 PRINT " b1,b2,b3,.....,bm"
1190 PRINT "*****NOTE THAT \)= ' DATA SHOULD BE GIVEN FIRST FOLLOWED"
1200 PRINT "*****BY \ (= ' DATA*****"
1210 LOCATE 22, 35: COLOR 0, 7: PRINT "press any key to continue....."
1220 RK$ = INPUT$(1): RK$ = "sleep 0"
1230 COLOR 2, 0, 3: CLS
1240 ZZ = 1
1250 REM if zz = +1 intermediate output is printed, but not if zz = -1
1251 INPUT "file name"; frs$
1252 CLS : LOCATE 25, 60: COLOR 0, 7: PRINT "FILE "; frs$
1260 REM read no. of types of constraints and no. of variables
1261 COLOR 2, 0, 3
1270 LOCATE 1, 1: INPUT "name of farmer/farm/bussiness/user"; NF$
1280 INPUT "number of greater than (i.e >=) constraints"; GC
1290 INPUT "number of equal to (i.e =) constraints"; EC
1300 INPUT "number of less than (i.e <=) constraints"; LC
1310 INPUT "number of enterprises to be analysed"; N
1320 CLS
1330 MM = GC + EC: M = MM + LC: MK = GC + LC: N1 = MK + N
1340 K = 0: L = 0: SS = 0
1350 P = N1 + MM: M1 = M + 1: M2 = M + 2: N0 = N1
1360 DIM A(M2, P), B(N0, M1), BS(N0), V(M2), NB(P), SL(P), C(P), CC$(N0), C
, W(N, M1), KX(N, M), RB(N0, 0), SB(M, M)
1370 FOR I = 1 TO N
1380 PRINT "name of enterprise no. "; I; : INPUT " "; CA$(I): CC$(I) = CA$
1390 NEXT I
1400 CLS
1410 FOR I = 1 TO M
1420 PRINT "name of constraint / variable input no. "; I; : INPUT " "; CC$
)
1430 NEXT I
1440 CLS
1450 REM read coeffs. of constraints and objectives
1460 FOR I = 1 TO M1: IF I = M1 THEN RR$ = "c" ELSE RR$ = "a"
1470 FOR J = 1 TO N
1480 IF I = M1 THEN PRINT "value of "; RR$: J; : INPUT " "; A(I, J) ELSE P
value of "; RR$: I; J; : INPUT " "; A(I, J): W(J, I) = A(I, J)
1490 IF I = M1 THEN A(I, J) = -A(I, J)
1500 NEXT J
510 CLS

```

```

11520 NEXT I
11530 REM set up slacks, artificials, label basis & read values into column
11540 IF GC = 0 THEN GOTO 11590 ELSE
11550 FOR I = 1 TO GC
11560 A(I, N + 1) = -1: A(I, N1 + 1) = 1: B(M2, I) = -1
11570 B(I, I) = 1: A(M2, N1 + 1) = 1: BS(I) = N1 + 1
11580 PRINT "value of b"; I; : INPUT " "; A(I, 0): B(I, 0) = A(I, 0): NEXT I
11590 IF EC = 0 THEN GOTO 11640 ELSE
11600 FOR I = GC + 1 TO MM
11610 A(I, N1 + 1) = 1: B(I, I) = 1: A(M2, N1 + 1) = 1
11620 BS(I) = N1 + I: B(M2, I) = -1: PRINT "value of b"; I; : INPUT " "; A(I
11630 B(I, 0) = A(I, 0): NEXT I
11640 IF LC = 0 THEN GOTO 11680 ELSE
11650 FOR I = MM + 1 TO M
11660 A(I, N + I - EC) = 1: B(I, I) = 1: BS(I) = N + 1 - EC: PRINT "value of
I; :
INPUT " "; A(I, 0): B(I, 0) = A(I, 0)
11670 NEXT I
11680 B(M1, M1) = 1: B(M2, M1) = 1
11690 CLS
11691 PRINT "LINEAR PROGRAMMING BY LP96": PRINT
11700 IF MM = 0 THEN PRINT "THERE IS NO PHASE 1": GOTO 11760 ELSE
11710 REM set up artificial function value for phase 1
11720 L = 1: N0 = P: REM n0 is no. of cols of interest
11730 FOR I = 1 TO MM
11740 B(M2, 0) = B(M2, 0) - B(I, 0)
11750 NEXT I
11760 ML = M1 + L: REM m1 = m+2 for phase 1; m+1 for phase 2
11770 PRINT
11780 IF ZZ >= 0 THEN PRINT "INITIAL TABLEAU": GOSUB 13000 ELSE
11790 REM label non-basic variables; nb(j)=0 if j is non-basic
11800 FOR I = 1 TO M: NB(BS(I)) = 1: NEXT I
11810 ZERO = 1E-08: NIL = 9.999999E-22
11820 REM find smallest c(j); coeff. in objective function.
11830 MIN = -ZERO: S = 0: PV = 0: ML = M1 + L
11840 FOR J = 1 TO N0: C(J) = 0
11850 IF NB(J) = 1 THEN GOTO 11910 ELSE
11860 REM calculate the c_j'.
11870 FOR I = 1 TO M: C(J) = C(J) + B(ML, I) * A(I, J): NEXT I
11880 C(J) = C(J) + A(ML, J)
11890 IF C(J) <= MIN THEN GOTO 11910 ELSE
11900 MIN = C(J): S = J
11910 NEXT J
11920 REM if s is still 0 all coeffs. are positive. this is the minimum
11930 IF S = 0 THEN 12290 ELSE
11940 REM find the row of var. to leave basis from minimum of bi/a'(is)
11950 MIN = 1E+20: R = 0
11960 REM calculate the a'(is) and store them in column v.
11970 FOR I = 1 TO M1: V(I) = 0
11980 FOR K = 1 TO M1: V(I) = V(I) + B(I, K) * A(K, S): NEXT K: NEXT I
11990 V(ML) = C(S)
12000 FOR I = 1 TO M
12010 IF V(I) <= ZERO THEN GOTO 12120 ELSE
12020 K = 0
12030 RT = B(I, K) / V(I)
12040 DF = RT - MIN
12050 IF DF >= 0 THEN GOTO 12080 ELSE
12060 R = I: MIN = B(I, 0) / V(I)
12070 GOTO 12120

```

```

12080 IF DF (0) < 0 THEN GOTO 12120 ELSE
12090 K = K + 1
12100 MIN = B(R, K) / V(R)
12110 GOTO 12030
12120 NEXT I
12130 REM if r is still 0 solution is unbounded.
12140 IF R = 0 THEN GOTO 12260 ELSE
12150 IF ZZ >= 0 THEN GOSUB 13150 ELSE
12160 REM update inverse and simplex multipliers.
12170 PV = V(R)
12180 FOR J = 0 TO M1: B(R, J) = B(R, J) / PV: NEXT J
12190 FOR I = 1 TO M1
12200 IF I = R THEN GOTO 12220 ELSE
12210 FOR J = 0 TO M1: B(I, J) = B(I, J) - V(I) * B(R, J): NEXT J
12220 NEXT I
12230 REM reset & relabel basic & non-basic variables
12240 NB(BS(R)) = 0: NB(S) = 1: BS(R) = S: NI = NI + 1
12250 GOTO 11830
12260 PRINT "VARIABLE "; S; " IS UNBOUNDED": PRINT
12270 GOSUB 13150: STOP
12280 GOTO 12760
12290 IF L = 0 THEN GOTO 12390 ELSE
12300 REM if in phase 2 this is min. if in phase 1 GOTO phase 2
12310 REM check that w has been made zero
12320 IF ABS(B(ML, 0)) >= 1E-08 THEN GOTO 12360 ELSE
12330 PRINT "PHASE 1 IS O.K."
12340 L = 0: N0 = N1: REM set l and no. of cols of interest for phase 2
12350 GOTO 11830
12360 PRINT "CONSTRAINTS HAVE NO FEASIBLE SOLUTION": PRINT
12370 PRINT "SUM OF ARTIFICIALS "; B(ML, 0): PRINT
12380 GOSUB 13150: STOP
12390 FOR I = 1 TO M
12400 RB((BS(I)), 0) = B(I, 0)
12410 NEXT I
12420 PRINT "OPTIMUM PRODUCTION PLAN": PRINT
12430 PRINT NF#: " SOLUTION IS OPTIMUM": TAB(50); "RETURN "; B(M1, 0)
INT
12440 PRINT "CON/ENT": TAB(13); "STATUS": TAB(26); "VALUE": TAB(39); "RET/UN
TAB(52); "VAL/UNIT": TAB(65); "NET/RETURN"
12450 PB = 144
12460 FOR I = 1 TO N0
12470 IF RB(I, 0) = 0 THEN 12480 ELSE GOTO 12490
12480 WW# = "NONBASIS": GOTO 12500
12490 WW# = "BASIS"
12500 PA = RB(I, 0)
12510 PA1 = ABS(A(M1, I))
12520 PA2 = PA1 + C(I)
12530 PA3 = PA1 - PA2
12540 PRINT CC#(I);
12550 PRINT TAB(13); WW#: TAB(26); PA; TAB(39); PA1; TAB(52); PA2; TAB(65);
12560 NEXT I
12570 PRINT
12580 PRINT : PRINT NF#: " SOLUTION IS OPTIMUM": TAB(50); "RETURN ";
, 0)
12590 PRINT
12600 PRINT "CON": TAB(13); "STATUS": TAB(26); "DUALvalue": TAB(39); "RHSval
TAB(52); "USAGE": TAB(65); "SLACK"
12610 FOR I = 1 TO M
12620 PRINT CC#(N + I);
12630 IF I (<= GC OR I > MM THEN GOTO 12650 ELSE

```

```

12640 PRINT "EQUATION      NONE": GOTO 12740
12650 IF N0(N + 1) = 1 THEN 12660 ELSE 12700
12660 PRINT TAB(13); "NONBINDING"; TAB(26); C(N + 1);
12670 PA = A(I, 0): PRINT TAB(39); PA;
12680 PA = A(I, 0) - B(I, 0): PRINT TAB(52); PA;
12690 PA = B(I, 0): PRINT TAB(65); PA: GOTO 12740
12700 PRINT TAB(13); "BINDING"; TAB(26); C(N + 1);
12710 PA = A(I, 0): PRINT TAB(39); PA;
12720 PA = A(I, 0) - B(I, 0): PRINT TAB(52); PA;
12730 PA = B(I, 0): PRINT TAB(65); PA
12740 NEXT I
12750 SS = 1: GOSUB 13150
12760 FOR I = 1 TO M
12770 FOR J = 1 TO M
12780 SB(J, I) = B(I, J)
12790 NEXT J
12800 NEXT I
12810 PRINT NF#; "      SOLUTION IS OPTIMUM"; TAB(50); "RETURN      "; B(M1, 0)
INT
12820 PRINT "NONEBASIS * INVERSE COLUMNS": PRINT
12830 PRINT "ENIPR      "; "RETURN      ";
12840 FOR I = 1 TO M
12850 IF BS(I) <= N THEN RR# = "X" ELSE RR# = "S"
12860 IF BS(I) <= N THEN PRINT " "; RR#; BS(I); " "; ELSE PRINT " "; RR#;
) - N; " ";
12870 NEXT I: PRINT
12880 FOR I = 1 TO N
12890 IF RB(I, 0) <> 0 THEN 12960 ELSE
12900 PRINT CA#(I); TAB(10); -C(I); : FOR J = 1 TO M
12910 FOR K = 1 TO M
12920 KX(I, J) = KX(I, J) + W(I, K) * SB(K, J)
12930 NEXT K
12940 PA = KX(I, J): GOSUB 13430
12950 NEXT J: PRINT
12960 NEXT I
12970 END: 'RUN "agriconc.bas"
13000 PRINT "      RHS      ";
13010 FOR J = 1 TO N: PRINT "      X"; J; " "; : NEXT J
13020 PRINT ""
13030 PB = 122
13040 FOR I = 1 TO ML
13050 IF I = M1 THEN PRINT "Z      "; : GOTO 13080 ELSE
13060 IF I = M2 THEN PRINT "W      "; : GOTO 13080 ELSE
13070 PA = A(I, 0): GOSUB 13430
13080 FOR J = 1 TO N
13090 PA = A(I, J): GOSUB 13430
13100 NEXT J
13110 PRINT ""
13120 NEXT I: PRINT ""
13130 ZZ = 1
13140 RETURN
13150 IF L = 1 THEN PRINT "STILL IN PHASE 1"
13160 PB = 122
13170 IF SS = 1 THEN GOTO 13220 ELSE
13180 FOR J = 1 TO N0: PRINT "      C"; J; " "; : NEXT J
13190 PRINT ""
13200 FOR J = 1 TO N0: PA = C(J): GOSUB 13430: NEXT J: PRINT "": PRINT
13210 PRINT "X"; S; "COMES IN, X"; BS(R); " IN CON"; R; "GOES OUT OF BASIS"
13220 PRINT : PRINT NF#; "      SOLUTION IS OPTIMUM"; TAB(50); "RETURN      "
1, 0): PRINT

```

```

230 PRINT TAB(25); "INVERSE OF BASIS": PRINT
240 PRINT "BAS     SLACK ";
250 FOR I = 1 TO M: IF BS(I) (= N THEN RR# = "X" ELSE RR# = "S"
260 IF BS(I) (= N THEN PRINT " "; RR#; BS(I); " "; ELSE PRINT " "; RR#; BS
- N; " ";
270 NEXT I
280 IF SS = 1 THEN PRINT : GOTO 13300 ELSE
290 PRINT " A' IS"
300 FOR I = 1 TO ML
310 IF BS(I) (= N THEN RR# = "X" ELSE RR# = "S"
320 IF I = M1 THEN PRINT "USM"; : GOTO 13350 ELSE
330 IF I = M2 THEN PRINT "ASM"; : GOTO 13350 ELSE
340 IF BS(I) (= N THEN PRINT RR#; BS(I); ELSE PRINT RR#; BS(I) - N;
350 FOR J = 0 TO M
360 PA = B(I, J): GOSUB 13430
370 NEXT J
380 IF SS = 1 THEN GOTO 13400 ELSE
390 PA = V(I): GOSUB 13430
400 PRINT ""
410 NEXT I: PRINT ""
420 RETURN
430 PC = INT(PB / 100)
440 P# = "      "
450 IF PC = 0 THEN PRINT "": GOTO 13470 ELSE
460 PRINT LEFT$(P#, PC);
470 PC = PB - 100 * PC
480 PD = INT(PC / 10): PC = PC - 10 * PD
490 IF PD = 0 THEN PD = 1
500 IF PA < 0 THEN P# = P# + "-"
510 PE = ABS(PA)
520 PE = PE + 5 * 10 ^ (-1 - PC)
530 IF PE >= 10 ^ PD THEN PRINT PA; : RETURN ELSE
540 P# = P# + MID$(STR$(INT(PE)), 2, PD)
550 PRINT RIGHT$(P#, PD + 1);
560 IF PC = 0 THEN RETURN ELSE
570 PRINT ".";
580 PE = INT((PE - INT(PE)) * 10 ^ PC)
590 P# = "0000000000"
600 P# = P# + MID$(STR$(PE), 2, PC)
610 PRINT RIGHT$(P#, PC); : RETURN

```

2500  
 LINEAR PROGRAMMING BY LP96

THERE IS NO PHASE 1

INITIAL TABLEAU

RHS	X 1	X 2	X 3			
1000	2.00	4.00	2.00			
1400	3.00	2.00	2.00			
2500	2.00	2.00	1.00			
Z	-18.00	-12.00	-15.00			
C 1	C 2	C 3	C 4	C 5	C 6	
-18.00	-12.00	-15.00	0.00	0.00	0.00	

X 1 COMES IN, X 5 IN CON 2 GOES OUT OF BASIS

SOLUTION IS OPTIMUM RETURN 0

INVERSE OF BASIS

BAS	SLACK	S 1	S 2	S 3	A' IS
S 1	1000	1.00	0.00	0.00	2.00
S 2	1400	0.00	1.00	0.00	3.00
S 3	2500	0.00	0.00	1.00	2.00
OSM	0.00	0.00	0.00	0.00	-18.00
C 1	C 2	C 3	C 4	C 5	C 6
0.00	0.00	-3.00	0.00	6.00	0.00

X 3 COMES IN, X 4 IN CON 1 GOES OUT OF BASIS

SOLUTION IS OPTIMUM RETURN 8400

INVERSE OF BASIS

BAS	SLACK	S 1	X 1	S 3	A' IS
S 1	66.67	1.00	-0.67	0.00	0.67
X 1	466.6667	0.00	0.33	0.00	0.67
S 3	1566.667	0.00	-0.67	1.00	-0.33
OSM	8400	0.00	6.00	0.00	-3.00

OPTIMUM PRODUCTION PLAN

SOLUTION IS OPTIMUM RETURN 8700

CON/ENT	STATUS	VALUE	RET/UNIT	VAL/UNIT	NET/RETURN
z1	BASIS	400	18	18	0
z2	NONBASIS	0	12	24	-12
z3	BASIS	100	15	15	0
x1	NONBASIS	0	0	4.5	-4.5
x2	NONBASIS	0	0	3	-3
x3	BASIS	1600	0	0	0

SOLUTION IS OPTIMUM RETURN 8700