# SOLVING LINEAR GOAL PROGRAMMING PROBLEM USING LEXICOGRAPHIC AND DUAL SIMPLEX METHODS

**MSc.PROJECT**

**BANCHAMLAK ABEBAW MESFIN**

**AUGUST2019**

**HARAMAYA UNIVERSITY, HARAMAYA**

# Solving Linear Goal Programming Problem Using Lexicographic and Dual Simplex Methods

**A Project submitted to the Department of Mathematics,**
**Postgraduate Program Directorate**
**HARAMAYA UNIVERSITY**

**In Partial Fulfillment of the Requirements for the Degree of**
**MASTER OF SCIENCE IN MATHEMATICS**
**(OPTIMIZATION)**

**Banchamlak Abebaw**

**August 2019**

**Haramaya University, Haramaya**

## HARAMAYA UNIVERSITY

## POSTGRADUATE PROGRAM DIRECTORATE

I hereby certify that I have read and evaluated this Project titled 'Solving Linear Goal Programming Problem: Lexicographic and Dual Simplex Methods' prepared under my guidance by Banchamlak Abebaw. I recommend that it can be submitted as fulfilling the project requirement.

| | | |
|---|---|---|
| Getinet Alemayehu (PhD) | ___________________ | ___________________ |
| Major Advisor | Signature | Date |
| Seleshi Demie (PhD) | ___________________ | ___________________ |
| Co Advisor | Signature | Date |

As member of the board of Examiners of the MSc. Project Open Defense Examination, we certify that we have read and evaluated the Project prepared by Banchamlak Abebaw and examined the candidate. We recommend that the Project be accepted as fulfilling the Project requirement for the degree of Master of Science in Mathematics (Optimization).

| | | |
|---|---|---|
| ___________________ | ___________________ | ___________________ |
| Chairperson | Signature | Date |
| ___________________ | ___________________ | ___________________ |
| Internal Examiner | Signature | Date |
| ___________________ | ___________________ | ___________________ |
| External Examiner | Signature | Date |

# DEDICATION

To all my family members who have been a constant source of motivation, inspiration, and support.

# STATEMENT OF THE AUTHOR

By my signature below, I declare that this Project is my own work. I have followed all ethical and technical principles of scholarship in the preparation, and compilation of this Project. Any scholarly matter that is included in the project has been given recognition through citation.

This Project is submitted in partial fulfillment of the requirements for MSc. degree in Mathematics with specialization in optimization at Haramaya University. The Project is deposited in the Haramaya University Library and is made available to borrowers under the rules of the Library. I solemnly declare that this Project has not been submitted to any other institution anywhere for the award of any academic degree, diploma or certificate.

Brief quotations from this Project may be made without special permission provided that accurate and complete acknowledgement of the source is made. Requests for permission for extended quotations from or reproduction of this Project in whole or in part may be granted by the Head of the Department when in his or her judgment the proper use of the material is in the interest of scholarship. In all other instances, however, permission must be obtained from the author of the Project.

Name: Banchamlak Abebaw Mesfin

Date ________

Department: Mathematics

Signature ________________

# ABBREVIATIONS

| | |
|---|---|
| DM | Decision Making |
| GP | Goal Programming |
| GPP | Goal Programming Problem |
| LGP | Lexicographic goal programming |
| LGP | Linear goal programming |
| LGPP | Linear Goal Programming Problem |
| LP | Linear programming |
| MDD | Multidimensional Dual |
| MOE | Ministry of Education |
| NLPP | Non-Linear Programming Problem |
| WGP | Weighted Goal Programming |

# BIOGRAPHICAL SKETCH

The author was born in 1995 on June 15 in Amhara Regional State, South Gonder Zone, and Ebinat Woreda. She attended her primary education at Bahrsegeda Primary school. Then after, she joined Ebinat Ketema Comprehensive Secondary School attend her secondary education Then she joined Debre Tabor University in 2014 and received Bachelor of Science degree in Mathematics on July4, 2016. She directly joined Postgraduate Program at Haramaya University, College of Natural and Computational Sciences, Department of Mathematics in 2017 to pursue a program of study for MSc. degree in Mathematics with specialization in Optimization.

# ACKNOWLEDGEMENTS

First, I would like to be grateful for the loving and kindness of the Almighty God in best owing me health, strength, patience and protection throughout my life.

I am deeply beholden to my advisors **Dr.Getinet Alemayehu** and **Dr.Seleshi Demie** for their profound commitment and encouragements they have shown throughout my study period. I pay respect and express indebtedness to them because of their inspiring guidance, consistent supervision and especially, their suggestions in selection of tools from the start of the proposal and at every phase of this project work.

I would like to extend my gratitude to Ministry of Education for the provision of the necessary support to complete this postgraduate study.

## TABLE OF CONTENTS

**DEDICATION** **IV**

**STATEMENT OF THE AUTHOR** **V**

**ABBREVIATIONS** **VI**

**BIOGRAPHICAL SKETCH** **VII**

**ACKNOWLEDGEMENTS** **VIII**

**LIST OF TABLES** **X**

**ABSTRACT** **XI**

**1. INTRODUCTION** **1**

1.1. Background of the Study 1

1.2. Statement of the Problem 4

1.3. Objectives 5

**2. LITERATURE REVIEW** **6**

2.1. Goal Programming 6

2.2. Lexicographic Method **Error! Bookmark not defined.**

2.3. Dual Simplex Method **Error! Bookmark not defined.**

**3. MATERIALS AND METHODS** **9**

**4. PRELIMINARIES** **10**

4.1. Basic Concepts and Definitions 10

4.2. Revised Simplex Method 12

**5. SOLVING LINEAR GOAL PROGRAMMING PROBLEM: LEXICOGRAPHIC AND DUA SIMPLEX METHODS** **14**

5.1. Solving Linear Goal Programming Problem by using Dual simplex method 16

5.2. Solving Linear Goal Programming by using Lexicographic Method 45

**6. SUMMARY, CONCLUSION AND RECOMMENDATION** **56**

6.1. Summary 56

6.2. Conclusion 56

6.3. Recommendation 56

**7. REFERENCES** **57**

**8. APPENDIX** **60**

**END** **64**

# LIST OF TABLES

**Table** **Page**


**1:** Table of linear goal programming 19
2:The primal initial table of example 5.1.1 25
3: The primal first iteration of example 5.1.1 26
4: The primal second iteration of example 5.1.1 27
5:The primal $3rd$ iteration of example 5.1.1 27
6: The primal $4th$ iteration of example 5.1.1 28
7:The primal $5th$ iteration of example 5.1.1 29
8 : The primal $6th$ iteration of example 5.1.1 29
9 :The primal 7th iteration of example 5.1.1 30
10: The primal 8th iteration of example 5.1.1 31
11:The dual 1st iteration of example 5.1.1 34
12: The dual 2nd iteration of example 5.1.1 36
13:The dual $3rd$ iteration of example 5.1.1 38
14:The dual 4thiteration of example 5.1.1 39
15:The dual 5th iteration of example 5.1.1 41
16:The primal $1st$ iteration of example 5.2.1 47
17 :The primal $2nd$ iteration of example 5.2.1 48
18 :The primal $3rd$ iteration of example 5.2.1 50
19 : The primal $4th$ iteration of example 5.2.1 51
20:The primal $5th$ iteration of example 5.2.1 53

**Solving Linear Goal Programming Problem Using Lexicographic and Dual Simplex Methods**

# ABSTRACT

*Goal programming is an important class of multi-criteria decision models widely used to solve problems involving conflicting objectives. Primarily to find a compromised solution which will simultaneously satisfy a number of goals. This note proposes a solution algorithm for linear goal programming problems. In solving goal programming problems, the solution methods reduce the multiple goal programming problems into a single objective of minimizing a weighted sum of deviations from goals. In this paper, we propose the goal programming problem as a multi- objective optimization problem of minimizing deviations from individual goals. This procedure eliminates the need of having extra constraints needed with classical formulations and also eliminates the need of any user-defined weight factor for each goal. The main objective of this study was to solve linear goal programming problem by using Lexicographic and Dual Simplex Method*. *An optimal solution is attained when all the goals are reached as close as possible to their aspiration level, while satisfying a set of constraints. Finally, some illustrative examples are presented to show the applicability and effectiveness of the solution method.*

**Key word**: Linear Goal programming, Lexicographic method, Dual simplex method

# 1. INTRODUCTION

## 1.1. Background of the Study

Optimization is the task of finding one or more solutions which correspond to minimizing or maximizing the objective function. It involves finding the minimum or maximum of an objective function $f(x)$ subject to some constraints. If there is no constraint then it is called an unconstrained optimization; otherwise, it is a constrained optimization.

Goal programming is a form of linear programming that considers multiple goals that are often in conflict with each other.

Tamiz et al.(1995) argue that LGP is the most widely used GP variants and the reported applications in the GP literature are related to LGP.goal programming is different from non-linear programming constrained optimization problems where the main idea is to find solutions which optimizes one or more criteria (Deb, 1995).

The objective function of a goal programming model is expressed in terms of the deviations fr om the desired goals and deviations from the desired goals are penalized (Cohon, 2004).

The aim of the goal programming is achieving as much goals as possible by minimizing their deviations from their targets. A detailed discussion about different aspects of goal programming is presented by( Ignizio ,1978).The major strength of Goal Programming is its simplicity and ease of use.

Goal programming problems can be solved by widely available linear programming computer packages as either a single linear programming or in the case of lexicographic variant a series of connected linear programming. Goal programming can therefore handle relatively large number of variables, constraints and objectives.

The differences are an explicit consideration of goals and the various priorities associated with the different goals. In the formulation two types of variable are used. They are decision variables and deviational variables. In goal programming the objective function contains primarily the deviation variables that represent goal the deviational variable represented in two dimensions in the objective function, aposetive and a negative deviation from each goal and constraint.decisition variables are Physical quantities controlled by the decision maker. goal programming is used to manage a set of conflict objectives by minimizing the deviations between the target values and the realized results (Rifai,1994).

If we want to define the basic theoretical framework of goal programming firstly we have to determine goals and especially the difference between fixed and free goals while each one has assigned the target value. Fixed goals represent constraints in the basic model of linear programming and are assigned with some target value. If the model of goal programming contents multiple even in some cases contradictory goals then their importance can be expressed either by using weights or lexicographically. Weights are dimensionless numbers and express the importance of individual goals.

Charnes and Cooper (1977) presented the general goal-programming model, which can be expressed:

$$\min \sum_{i=1}^{p} |f_i(x) - T_i| \tag{1}$$

Subject to $x \in X$

Where $T_i$ denotes the target or goal set by DM for $i^{th}$ objective function $f_i(x)$ and $X$ represents the feasible region from which the choices of vector $x$ must be effected. The criterion then is to minimize the sum of the absolute values of the differences between target values and actual achieved values. If the given objective is multiple then change non linear to linear form.

Note that a more general formulation of the goal-programming objective function is a weighted sum of the $p^{th}$ power of deviation $f_i(x) - T_i$. Such a formulation has been called generalized goal programming. if returning to formulation expressed by equation (1) above the objective function is non-linear and the simplex method with its many inherent advantages cannot be applied directly. However, it is possible to transform equation (1) into linear form, thus reducing goal programming to a special type of linear programming. The transformation defines deviational variables $d_i^+$ and $d_i^-$ such that

$$d_i^+ = \frac{1}{2}\left[\left|\sum_{j=1}^{n} a_{ij}x_j - b_i\right| + \left(\sum_{j=1}^{n} a_{ij}\, x_j - b_i\right)\right] \tag{2}$$

$$d_i^- = \frac{1}{2}\left[\left|\sum_{j=1}^{n} a_{ij}x_j - b_i\right| - \left(\sum_{j=1}^{n} a_{ij}\, x_j - b_i\right)\right] \tag{3}$$

Where $a_{ij}$ is constant coefficient of decision variable

$b_i$ is the right hand side

$x_j$ is decision variables

Examination of equation (2) reveals that $d_i^+$ is the positive deviation from $i^{th}$ target for $i^{th}$ objective (over achievement of a goal). The second slack variable $d_i^-$ is the negative deviation from the $i^{th}$ target for the $i^{th}$ objective (under achievement of a goal). Adding equations (2) and (3) it is seen that; $d_i^+ + d_i^- = |f_i(x) - T_i|$

Thus, the objective function in formulation (1) can be replaced by an equivalent linear relationship, Furthermore, by subtracting (2) from (3), the relationship is as under:

$$d_i^+ - d_i^- = f_i(x) - T_i$$

It is also required that $d_i^+$ and $d_i^-$ be non negative, and since it is not possible to have both under achievement and over achievement of a goal simultaneously, then one or both of the deviational slack variables must have a zero value; that is $d_i^+, d_i^- = 0$ .The constraint is automatically fulfilled by the simplex method. This is because the objective function will drive either (or perhaps both) $d_i^+$ or $d_i^-$ to zero for all $i$.Thus an equivalent linear formulation of equation (1) is

$$\min z = \sum_{i=1}^{m} d_i^+ + d_i^- \quad (4)$$

$$\text{subject to: } \sum_{j=1}^{n} a_{ij}x_j - d_i^+ + d_i^- = b_i, i = 1,2,..m$$

With $d_i^-, d_i^+, x_j \geq 0 \; for \; i = 1,2,..m; j = 1,2,...n$

Both over achievement and underachievement of a goal cannot occur simultaneously. Hence, either one or both of these variables must have a zero value; that is, $d_i^- * d_i^+ = 0$.

The GP model in (4) has an objective function, constraints (called goal constraints) and the same nonnegative restriction on the decision variables as the LP model.

Here instead of maximizing or minimizing the objective function the deviation between the goals is minimized according to the assigned priorities (Slomp and Chowdary, 2002).

## 1.2. Statement of the Problem

Goal programming is an extension of linear programming, which is a mathematical tool to handle multiple, normally conflicting objectives. In such situations, it will be difficult to find a single solution that optimizes the conflicting objectives

The Goal Programming appears to be an appropriate, powerful and flexible technique for decision analysis of the troubled modern decision maker who is burdened with achieving multiple conflicting objectives under complex constraints.

The modeling approach does not attempt to maximize or minimize the objective function directly as in the case of conventional Linear Programming.

As noted by (Romero,2014) GP is a widely used multi-criteria decision-making technique. The GP modeling framework is easy to understand and apply and can be solved using the commercial mathematical programming software. The basic idea is to establish specific goals for each objective, and then to seek a solution that satisfies all the given constraint and then results obtained according to ranked priorities.

Ignizio (1976) gave an algorithm that shows how a Lexicographic method can be solved as a series of linear programming model. Lexicographic method should be used when there is a clear priority ordering amongst the goal to be achieved. Then we were used to Lexicographic and Dual Simplex Method to solve linear goal programming problems.

The project tried to answer the following questions.

- How to find solution of linear goal programming problem by using Lexicographic and Dual simplex Method?
- How to compare the two solution methods?

## 1.3. Objectives

The main objective of this project was to solve linear goal programming problem by using lexicographical and dual simplex methods. The study intended to explore the following specific objectives.

- To use Lexicographic and dual simplex method to find solution of linear goal programming problems.
- To make comparison between the two methods based on spadework.

# 2. LITERATURE REVIEW

## 2.1. Goal Programming

Goal programming is a method for solving multiple objective problems (Sunar and Kahraman, 2001). This method starts by introducing artificial variables to the model and using the two-phase method or big-m method. According to him, this method uses a smaller number of variables in computation and is efficient in solving goal programming problem .but produces entirely different and wrong results if the problem is weighted.

Moreover, some heuristics and metaheuristics have been proposed for solving of multiple objectives problems. According to the research of (Jones, Mirrazavi et al. ,2002) simulated annealing and Tabu search are two of the most popular meta heuristics used in solving multiple objectives.

Calvete and Mateo (1998) presented a lexicographic optimization of multi objective generalized network flow problem based on the underlying ideas of primal dual algorithms for the minimum cost generalized network flow problem. The algorithm is efficient in reaching optimality condition, but tedious in labeling process because of several nodes, arcs, paths which result in multiple solutions.

Baykasogluet.al.(1999) used multiple objective Tabue search algorithms solve linear goal programming models. The danger is that the algorithm may recycle old solutions and become trapped in a loop as indicated by the author which implies that it does not handle all kinds of goal functions and constraints.

Amid et al. (2009) discuss a weighted additive fuzzy multi-objective model for the supplier selection problem that aggregates weighted membership functions of objectives involving minimizing the net cost, net rejected items and net late deliveries, to satisfy capacity and demand constraints. Lee et al. (2009) develop a fuzzy multiple GP model that helps downstream manufacturers to choose thin film transistor liquid crystal display suppliers.

Kasana (2003) developed an alternative algorithm for solving LGPP called grouping algorithm. That considers all goals and real constraints together as one group with the objective function being the sum of all the unwanted deviations, and solves a sequence of LP sub problems each using the optimal solution of the previous sub problems. This algorithm is being dominated by the partitioning method as indicated by the author. He indicated that it is good and performs well only if a large number of goals are satisfied. In other word if an unsatisfied goal is in the final tableau it is inefficient. It utilized sequential method.

Authur and Ravindran (1978) presented an efficient algorithm for solving linear goal programming problems utilizing the hierarchical structure of pre-emptive models using partitioning and elimination procedures that starts by considering only those constraints affecting required structural constraints and the first priority goals.

Goals should multiple optimal solutions exist for this model constraint affecting the next priority are added and the new model solved. But (Schniederjans and Kwak,1982) stated that Arthur and Ravindran computational procedure are limited to problems that have priorities or do not have conflicting goal constraints that lose variables via the variable elimination process. They stated that conflicting goal constraints that are later added to an already optimized tableau as described in their procedure without an iterative adjustment run the risk of violating the original goal programming.

Evans (1984) described GP problem as a technique for finding that solution which minimizes the deviation over all feasible solutions such a solution is called a best compromise solution and that under the assumption that more of each objective is preferred to less a best compromise solution. Min and Storbeck (1991) stated that GP is a technique not designed to find an optimal point but to find an acceptable range and advised that the dispute of Goal Programming. Dominance will continue unless the management scientist can accept goal programming's satisfying principle and not being captivated by the principle of optimality. However in Goal programming there is no method to determine if a solution is better than other.

The computational procedure in goal programming is to select a set of solutions which satisfies the constraints and providing a satisfactory goal, ranked in priority order since GP approach seeks satisfying solutions which come as close to the desired aspiration levels as possible. Antonio et al. (2009) described Pareto dominance relation as the most commonly adopted method in multi objective optimization to compare solutions, which instead of a single optimal solution leads to a set of alternatives with different trade-offs among the objectives. Their solutions are called Pareto optimal solutions or non- dominated solutions.

Charnes and Cooper (1997) pointed out the link between goal programming and multi objective optimization while (Romero, 1985) proved that goal programming is a certain case of the distance function model.

Romero (1991) developed an approach for detection of Pareto inefficiency in a goal programming solution. These methods provide ways of restoring Pareto efficiency by calculating a Pareto-efficient solution that dominates the goal programming solution. These algorithms carry out maximization.

Ignizio (1982) developed another procedure for solving GPP that reduced tableau element with the use of the condensed simplex tableau alongside with the concept of column dropping and reflected p -space to reduce storage. He utilized fully the positive deviational variables in the basis. But( Schniederjans and Kwak,1983) reported that( Ignizio,1982) algorithm requires more computational element manipulation than the Schniederjans and Kwak algorithm, and lso fails to provide useful information that is commonly found in more popular G.P. algorithms such as that of (Lee ,1972), Tamiz et.al.(1995) reviewed current literature on the branch of multi-criteria decision modeling known as GP.

According to Romero and Rehman (2003) both Lexicographic Goal Programming (LGP) and Weighted Goal Programming (WGP) are known and widely used as goal programming methods. Taha (2003) also contributed that the weights and the pre-emptive methods convert the multiple goals into a single objective function stating that these methods do not generally produce the same solution.. Hillier and Lieberman (2001) said that Goal programming problems can be categorized according to the type of mathematical programming model that it fits except for having multiple goals instead of a single objective.

Hillier and Lieberman (2001) concluded that goal programming and its solution procedures provide an effective way of dealing with problems where management wishes to strive towards several goals simultaneously.

According to Ignizio (1978) Goal Programming is a tool that has been proposed as a model and approach for analysis of problems involving multiple conflicting objectives.

He pointed out that actual real-world problems invariably involve non-deterministic system for which a variety of conflicting non-commensurable objectives exist. The concept of satisfaction functions in the goal-programming model was introduced by (Martel and Aouni,1990).

# 3. MATERIALS AND METHODS

Sources in the web and library were used to collect all the pieces of information about linear Goal Programming Problems together with the methods and recorded subsequently.

Specifically,

- Relevant Journals and books were addressed to gather information about Goal Programming problem and the methods to solve the problem.
- The collected information was analyzed and arranged keeping coherence.
- Important definitions, concepts and examples were discussed to make ideas clear.
- Lexicographic and Dual Simplex Methods were used to solve linear goal programming problems.
- To use Dual simplex method after changed primal to Dual we use Revised simplex method to solve linear goal programming problem.
- Lingo software was used to solve linear goal programming problem.
- Generally we use different books, journals, solution methods, software to solve linear goal programming problem.

# 4. PRELIMINARIES

In this chapter we deal with definitions and concepts which are important for the study of solving linear Goal programming problem.

## 4.1. Basic Concepts and Definitions

**Definition4.1.1.**(**Decision Maker)**:The decision maker refer to the person, organization or stakeholder to whom the decision problem under consideration belongs.

**Definition4.1.2.(Achievement Function):**The function that serves to measure the achievement of the minimization of unwanted goal deviation variables in the goal-programming model.

**Definition4.1.3. (Goal Function):**A mathematical function that is to be achieved at a specified level.

**Definition4.1.4. (Goal Program):** A mathematical model consisting of linear or nonlinear functions in which all functions have been transformed into goals.

**Defination4.1.5. (Negative Deviation):** The amount of deviation for a given goal by which it is less than the aspiration levels.

**Definition 4.1.6.(Positive Deviation):**The amount of deviation for a given goal by which it exceeds the aspiration level.

**Definition 4.1.7 (Constraint):** A constraint is a restriction upon the decision variables that must be satisfied in order for the solution to be implementable in practice. This is distinct from the concept of a goal whose non-achievement does not automatically make the solution non-implementable. A constraint is normally a function of several decision variables and can be equality or an inequality.

**Definition 4.1.8.**A linear programming problem can be defined as the problem of maximizing or minimizing a linear function subject to linear constraints. The constraints may be equalities or inequalities.

**Definition 4.1.9.**A function $f(x_1, x_2, \dots, x_n)$ of $x_1, x_2, \dots, x_n$ is a linear function if and only if
For some set of constants,$c_1, c_2, \dots c_n$, $f(x_1, x_2, \dots x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$.
For example, $f(x_1, x_2) = 2x_1 + x_2$ is a linear function of $x_1$and $x_2$.

**Definition4.1.10.**For any linear function $f(x_1, x_2, \dots, x_n)$ and any number $b$,the inequalities $f(x_1, x_2, \dots., x_n) \geq b$ and $f(x_1, x_2, \dots, x_n) \leq b$ are linear inequalities. Thus,$2x_1 + 3x_2 \geq 3$ $and$ $2x_1 + 3x_2 \leq 3$ are linear inequalities.

We have seen that a linear programming can have both equally and in equally constraint it also can have variable that are required to be non-negative as well as those allowed to be unrestricted in sign. Before the simplex algorism can be used to solve linear programming. The LP must be converted in to an equivalent problem in which all constraint are equation and all variable are non-negative. A linear programming in this form is said to be in standard form.

**Defination.4.1.11.(Slack variable).**A linear constraint of the form $\sum a_{ij}x_j \leq b_i$ can be converted into an equality by adding a new non-negative variable to the left-hand side of the inequality. Such a variable is equal to the difference between the right- and left-hand sides of the inequality and is known as slack variable. It represents the waste involved in that phase of the system modeled by the constraint.

**Defination.4.1.12.(Surplus variable):**a linear constraint of the form $\sum \mathrm{a_{ij}x_j} \geq \mathrm{b_i}$can be converted into equality by subtracting a new non-negative variable to the left-hand side of the inequality. Such a variable is equal to the difference between the left and right-hand side of the in equality and is known as surplus variable. It represents excess input into phase of the system modeled by the constraint.

**Defination.4.1.13.(Feasible Region ):**The set of solutions in decision space that satisfy all constraints and sign restrictions in a goal programming form the feasible region. Any solution that falls within the feasible region is deemed to be implementable in practice**.**

## 4.2. Revised Simplex Method

The algorithm for the solution of the LGP model is termed the revised multiphase simplex algorithm .As such it is basically a straight forward modification of revised simplex for LP. Where in the so called two- phase simplex process is utilized (Murtagh,1981).The modification itself permits multiple phases rather than just two as in conventional Lp, Under the assumption that the algorism is to be ultimately implemented.

We begin the algorism by assuming that we have an initial basic feasible solution some representation of $B^{-1}$and the associated program:$b = B^{-1}b$ with the latter designated as the current right-hand side .When employing the LGP model these are trivial requirements because initially ${d_i}^- = b$ and $x, {d_i}^+ = 0$ will always provide a basic feasible solution we assume that all goals are written with non negative right hand side .further the basis associated with $v_B = {d_i}^-$ is the identity matrix and thuse $B^{-1} = I$.

We may then generate the multidimensional shadow price vectors $d_j$ for all non basic variables and determine whether or not the present basic feasible solution is optimal. If so we may stop .otherwise we must proceed to a pivoting operation .pivoting involves the exchange of anon basic variable for a basic variable in a manner such that

(1) the new solution is still a basic feasible solution and

(2) the resultant value of $u^T$ is improved

We now list the steps of the revised simplex algorism for LGP (Ignizio, 1983 ) .

Step1.Initialization.$v_B = {d_i}^-$.Thus $B = I, B^{-1} = I$ and $\beta = b$. Set $k = 1$. Initially all variables are unchecked.

Step2.Develop the pricing vector. Determine:

$$\pi^{(k)^T} = {c_B}^{(k)^T} B^{-1}$$

Step3. Price out all unchecked non-basic columns. Compute:

${d_j}^{(k)} = \pi^{(k)^T} a_j - {c_j}^{(k)}$ For all $j \in N$ where $N$ is the set of non-basic and unchecked variables.

Step 4.Selection of entering non-basic variables. Examine those ${d_j}^{(k)}$as computed in step 3. If none are positive proceed to step 8. Otherwise select the non-basic variable with the most positive ${d_j}^{(k)}$(ties may be broken arbitrarily) as the entering variable. Designate this variable as $v_q$.

Step 5.Update the entering column. Evaluate:

$$a_q = B^{-1}a_q$$

Step 6.Determine the leaving basic variable. We shall designate the leaving variable row as $i = p$. Using the present representation of $\beta$ and the values of $a_q$, as derived in step 5 we determine:

$$\frac{\beta_p}{a_{p,q}} = minimum \left\{\frac{\beta_i}{a_{i,q}}\right\} \text{for } a_{i,q} > 0$$

Again, ties may be broken arbitrarily. The basic variable associated with row $i = p$ is the leaving variable, $v_{B,p}$.

Step7. *Pivot*. We replace $a_p$ in $B$ by $a_q$and compute the *new* basis inverse, $B^{-1}$. Return to step 2.

Step 8.Convergence check. If either one or both of the following conditions holds, stop as we have found the optimal solution(a) if all ${d_j}^{(k)}$as computed in step 3 are negative, or:

(b) If $k = K$ (where $k$ =number of priority levels, or terms in $u^T$). Otherwise check all non-basic variables associated with a negative ${d_j}^{(k)}$ set $k = k + 1$ and return to step 2.

The above eight steps represent the primary elements of the revised multiphase simplex algorithm for LGP.

**Defination4.2.1.(Leaving variable):**The basic variable with the highest negative value is the exiting variable. If there are two candidates for exiting variable, any one is selected. The row of the selected exiting variable is marked as pivotal row.

**Defination4.2.2.(entering variable):**Cost coefficients corresponding to all the negative elements of the pivotal row, are identified. Their ratios are calculated after changing the sign of the elements of pivotal row, the column corresponding to minimum ratio is identified as the pivotal column and associated decision variable is the entering variable. **Defination4.2.3.(Pivotal element):**Pivotal element is exactly same as in the case of simplex method, considering the pivotal element as the element at the intersection of pivotal row and pivotal column.

# 5. Solving Linear Goal Programming Problem Using Lexicographic and Dual Simplex Methods

## 5. 1.Linear Goal Programming

Goal programming is a form of linear programming that considers multiple goals with linear constraint. The goal programming model is one in which all objectives are converted into goals. This conversion is accomplished by an aspiration level to the right-hand side of each objective. For example, an objective such as maximize profit might be restated as the goal: "Obtain x or more units of profit." Further, any profit under the desired x units represents an undesirable or unwanted deviation from the goal (Morris,1964).Linear goal programming or LGP in turn is used to describe the methodology employed to find the program for a model consisting solely of linear goals. Then one seeks the solution that minimizes the distances between that solution and the aspired solution (Romero,1998) .This fundamental idea leads to the following analytical framework. The general structure of the $\mathrm{i}^{\mathrm{th}}$goal can be expressed as:

$\mathrm{g_i o}\ \mathrm{f_i(x)} - \mathrm{d_i}^{+} + \mathrm{d_i}^{-} = \mathrm{T_i}, \mathrm{i} = 1,2, \ldots \mathrm{k},$

Where $k$ is total number of goals.

$f_i(x)$: The mathematical expression for the $\mathrm{i}^{\mathrm{th}}$attribute (a linear function of the $i^{th}$ goal)

$T_i$: The target value for the $i^{th}$goal $g_i$.

$d_i{}^{-}$: The negative deviational variable, quantification of the underachievement of $i^{th}$ goal.

$d_i{}^{+}$: The positive deviational variable, quantification of the overachievement of $i^{th}$ goal.

After the formulation of all small number of goals ,the next aim is to detect the unwanted deviation variables. The variables are unwanted in the sense that these are the variables that DM (decision maker) wants to minimize. To illustrate this procedure the following cases are presented:

Case I. Let us consider $\mathrm{f_i(x)} \geq \mathrm{T_i}$ ,where goal is attached to a maximization type objective. In this case, the DM does not want under achievement with respect to target $T_i$. Consequently, the unwanted deviational variable $d_i{}^{-}$to be minimized.

Case II**.** Let us consider $f_i(x) \leq T_i$ , where goal is attached to a minimization type objective. In this case, the DM does not want over achievement with respect to target $T_i$. Consequently the unwanted deviational variable $d_i{}^{+}$to be minimized.

Case III. Let us consider $f_i(x) = T_i$ ,where goal is to be achieved exactly. In this case the decision maker wants neither over-achievement nor under-achievement with respect to target $T_i$. Hence both the negative deviational variable $d_i{}^-$and $d_i{}^+$positive deviational variables are equally unwanted, making it necessary to minimize $d_i{}^- + d_i{}^+$.

The non-negative variable $d_i{}^-$and $d_i{}^+$are called deviational variables because they represent the deviations below and above the right hand said of the constraint $i$.the deviational variable $d_i{}^+$and $d_i{}^-$are by definition dependent and hence cannot be basic variable simultaneously. This means that in any simplex iteration at most one of the two deviational variables can assume a positive value. If the original $i^{th}$ inequality is of the type $\leq$ and its $d_i{}^- > 0$,then the $i\ ^{th}$ goal is satisfied; otherwise, if $d\quad^+ > 0$, goal $i$ is not satisfied.

Goals are defined as the lifelong aims, which an individual or entity endeavor to achieve something. On the other hand, objectives are the specific milestones which a person plans to achieve in a limited period. These are precise, measurable, time-based, actions that assist in the achievement of goal.

## 5.2. Solving Linear Goal Programming Problem by using Dual simplex method

The dual of the LGP model or the multidimensional dual (Ignizio, 1976) has been efficient computational formulation especially for large-scale goal-programming models. Ignizio (1985) developed a multidimensional dual simplex algorithm for solving GP problems (MDD).

In particular, an attractive feature of the dual simplex-based algorithm for obtaining the solution of the LGP problem it can be implemented easily with any conventional simplex algorisms.

The linear goal programming problem has an associated dual problem called the multidimensional dual problem (Ignizio, 1983). The theory and properties of the dual problem of a linear goal programming problem were explored by (Murkowski and Ignizio, 1983).

The dual which I denoted as the multi dimensional dual was established (Ignizio, 1974).

Ignizio (1983) provide a detailed explanation of the development of the multi-dimensional dual and its transformation from the LGP primal. Here we briefly describe this process.

The LGP primal may be written in general form as follows.

Minimize $z = \left[g_1\left(d_i^{-}, d_i^{+}\right), \ldots, g_k\left(d_i^{-}, d_i^{+}\right)\right]$ (1)

Subject to: $\sum_{i=1}^{n}(c_{ij}\, x_j + d_i^{-} - d_i^{+}) = b_i$ for all $i$ (2)

$x, d_i^{-}, d_i^{+} \geq 0$ (3)

Where

$x_{j=}$The $j^{th}$ decision variable;

$c_{ij=}$ The coefficient of $x_j$ in the $i^{th}$ goal or rigid constraint

$d_i^{-}, d_i^{+} =$ Negative and positive deviations, respectively of goal or rigid constraint;

$b_i =$The right-hand side for rigid constraint $i$ or the aspiration level for goal $i$;

$z =$The LGP Achievement vector

$z_k = g_k\left(d_i^{-}, d_i^{+}\right)$ Where $g_k\left(d_i^{-}, d_i^{+}\right)$ is usually a linear function of the weighted, unwanted deviation variables .

In LGP the specific form of $z_k = g_k\left(d_i^{-}, d_i^{+}\right)$ is typically:

$$z_k = \sum_{i=1}^{m}\left(u_i^{(k)} d_i^{-} + w_i^{(k)} d_i^{+}\right) \quad (4)$$

This permits us to write the LGP primal in matrix notation as shown below:

$$\min z = \left\{ \left(u^{(1)} w^{(1)}\right) \begin{pmatrix} x \\ d_i^{\ -} \\ d_i^{\ +} \end{pmatrix}, \ldots\ldots, \left(u^{(1)} w^{(1)}\right) \begin{pmatrix} x \\ d_i^{\ -} \\ d_i^{\ +} \end{pmatrix} \right\} \qquad (5)$$

$$\text{Subject to} \left[c^{(1,k)} I^{(1,k)} - I^{(1,k)}\right] \begin{pmatrix} x \\ d_i^{\ -} \\ d_i^{\ +} \end{pmatrix} = b \qquad (6)$$

$$x, d_i^{\ -}, d_i^{\ +} \geq 0$$

The theory is similar to that for a conventional linear programming problem and its dual Translated into a preemptive priority framework. The multi-dimensional dual formulation

$$\max z = -b^{(1,K)^T} v + \left\{u^{(1)} b^{(1)}, \ldots, u^{(k)} b^{(k)}\right\} \qquad (7)$$

$$\text{Subject to} \begin{bmatrix} -\left(c^{(1,k)^T}\right) \\ -\left(I^{(1,k)}\right) \\ \left(I^{(1,k)}\right) \end{bmatrix} v \leq \begin{bmatrix} \left(-u^{(1)} c^{(1)}\right)^T \\ (o) \\ \left(u^{(1)} + w^{(1)}\right)^T \end{bmatrix}, \ldots, \begin{bmatrix} \left(-u^{(k)} c^{(k)}\right)^T \\ (o) \\ \left(u^{(k)} + w^{(k)}\right)^T \end{bmatrix} \qquad (8)$$

Where $u =$ the weight assigned to the negative deviation variable $i$ at priority $k$;

$w =$The weight assigned to the positive deviation variable $i$ at priority $k$

$v =$ both unrestricted and multi-dimensional vector.

$I =$Identity matrix

Another expression of Multidimensional dual formulation

The initial linear goal programming primal formulation

$$\min \mathrm{u}^{\mathrm{T}} = \left\{\mathrm{c}^{(1)^{\mathrm{T}}} \mathrm{v}, \mathrm{c}^{(2)^{\mathrm{T}}} \mathrm{v} \ldots \ldots .. \mathrm{c}^{(\mathrm{k})^{\mathrm{T}}} \mathrm{v}\right\} \qquad (1)$$

$$\text{Subject to} \qquad Av = b \qquad (2)$$

$$\mathrm{v} \geq 0 \qquad (3)$$

Further, as $\mathrm{c}^{(\mathrm{k})^{\mathrm{T}}}$ represents the weight given to variable j, at rank k, then all $\mathrm{c}^{(\mathrm{k})^{\mathrm{T}}}$ are, *in* LGP, nonnegative. That is

$$, c^{(k)^T} \geq 0 \qquad (4)$$

We first note that the set of goals is given as:

$Av = b$ (5)

However, the mxn matrix, A, may be partitioned into:

A = (B:N)

where:

B = a mxm nonsingular matrix, designated as the *basis* matrix, and

N = a mx(n m) matrix

Further, the variable set, v, may be similarly partitioned into:

$$v = \begin{pmatrix} v_B \\ \cdots\cdots \\ v_N \end{pmatrix} \quad (6)$$

Where

$v_B$ = the set of basic variables those associated with B and

$v_N$ = the set of non-basic variables those associated with N

Consequently, we may rewrite (5) as

$Bv_B + Nv_N = b$ (7)

And, as $B$ is nonsingular (and thus has an inverse), we may pre-multiply each term in (7) $B^{-}$obtain

$B^{-}Bv_B + B^{-}Nv_N = B^{-}b$

$v_B = -B^{-}Nv_N + B^{-}b$ (8)

Next, examine the LGP achievement function as given in (1). The general, or $k^{th}$ element of u is given as

$$c^{(k)^T}v$$

However, recall that v was partitioned according to (6) and thus the above term may be rewritten as

$$c_B{}^{(k)^T} v_B + c_N{}^{(k)^T} v_N \qquad (9)$$

Where in the subscripts for c reflect those coefficients associated with the set of basic variables or those associated with the non-basic variables "B" or N", respectively.

We may now, using (8), substitute for $v_B$ in (9) to obtain

$$c^{(k)^T} v = c_B{}^{(k)^T} B^{-} b - ( \; c_B{}^{(k)^T} B^{-} N + \; c_N{}^{(k)^T}) v_N \qquad (10)$$

reduced form of linear goal programming primal:

$$\min u^T = \{[c_B{}^{(1)^T}\beta - (\pi^{(1)^T} N - c_N{}^{(1)^T}) v_N], \ldots., [c_B{}^{(k)^T}\beta - (\pi^{(k)^T} N - c_N{}^{(k)^T}) v_N]\} \; \textbf{(*)}$$

Subject to $\quad v_B = \beta - B^{-1} N v_N \qquad$ (**)

$$v = \begin{pmatrix} v_B \\ \cdots. \\ v_N \end{pmatrix} \geq 0 \qquad (***)$$

If we recall that we may write as (**)

$$B v_B + N v_N = b$$

LGP multidimensional Dual is given below

Find $Y$ so as to

$$\max w = b^T Y + \{c_B{}^{(1)^T} B^{-1} b, \ldots, c_B{}^{(k)^T} B^{-1} b\}$$

Subject to $\begin{pmatrix} B^T \\ \ldots. \\ N^T \end{pmatrix} Y \leq \begin{pmatrix} 0 \\ \ldots \ldots \ldots \ldots \ldots \ldots. \\ c_N{}^{(1)} - N^T (B^{-1})^T c_B{}^{(1)} \end{pmatrix}, \ldots \begin{pmatrix} 0 \\ \ldots \ldots \ldots \ldots \ldots \ldots. \\ c_N{}^{(k)} - N^T (B^{-1})^T c_B{}^{(k)} \end{pmatrix}$

$Y$ is unrestricted and multidimensional

An alternative and quite convenient way in which we may summarize(*)—(***) is by means of table.

We may define a basic solution as one in which all non basic variables are set at their bound .for our purpose, this bound shall be zero. thuse

**Table 1:** table of Linear goal programming

| | $v_B$ | $v_N$ | $\beta = RHS$ |
|---|---|---|---|
| $v_B$ | $B^{-}B = I$ | $B^{-}N$ | $B^{-}b$ |

| | | | |
|---|---|---|---|
| $p_1$<br><br>$p_k$ | 0 | ${c_B^{(1)}}^T B^- N - {c_N^{(1)}}^T$<br><br>${c_B^{(k)}}^T B^- N - {c_N^{(k)}}^T$ | ${c_B^{(1)}}^T B^- b$<br><br>${c_B^{(k)}}^T B^- b$ |

If $v_N = 0$ a basic solution result more specifically if $v_N = 0$ then $B^- N v_N = 0$ and thuse $v_{B=B^- b}$

$$v = \begin{bmatrix} v_B \geq 0 \\ - \\ v_N = 0 \end{bmatrix} \geq 0$$

In LGP as in LP the optimal solution may always be found as a basic feasible solution (Charnes and Cooper, 1961).

The three primary conditions associated with the reduced form of the LGP model are feasibility, implimentablity and optimality. Given $v_N = 0$ (non basic variables are zero) this terms are defined as follows.

Feasibility: if $\beta = B^{-1} b \geq 0$ the result solution, or program is denoted as feasible.

Implimentablity: if ${c_B^{(1)}}^T \beta = 0$ then the resultant program is designated as being implementable solution. That is the top ranked set of goal or the set of rigid constraints are satisfied.

Optimality. a basic feasible solution is optimal whenever all shadow price vectors ($d_j$), for non-basic variables, are non-positive.

We simply note that the so-called initial basic feasible solution in LGP always consists of the negative deviation variables. We may note that implimentablity is a condition that is unique to LGP. further unlike linear programming there is no condition of unbounded in LGP. this may be observed by simplex examining the achievement function.

Achievement function could only be unbounded in the case of seeking the lexicographic minimum of achievement function. If one or more element of $u^T$ could decrease to mines infinity this is obviously impossible because $v \geq 0, c^{(k)} \geq 0$ for all $k$.thus the absolute minimum value for any achievement function is zero. Having noted that the multi-dimensional dual is a 'linear programming' problem with multiple and prioritized right-hand sides, the development of a simplex based algorithm for its solution is straightforward. That is, what we shall do is to apply the simplex algorithm of conventional linear programming to a series of related linear programming models. Each model in the series is identical to the linear programming model .with the exceptions that:

(1) The right-hand side changes;

(2) Certain of the constraints will be dropped dependent upon the solution of linear programming.

The different right-hand sides are evident in (1) while the removal of constraints is a result of the observance of complementary slackness. This is clearly just the dual of the 'restricted entry' rule of the multiphase simplex algorithm for the LGP prima. Ignizio (1983) however to clarify we note that if the multi-dimensional dual of (7) and (8) is solved for a single right-hand side in (8), the $k^{th}$ right-hand side, then for the optimal solution to this model we note that:

all non-binding dual constraints in (8) correspond to primal variables that are:

(a) Non-basic in the optimal solution for the $k^{th}$ primal achievement vector.

(b) Associated with negative shadow prices in the optimal tableau.

Thus as well known, any primal variable of the second class (as noted directly above) can never be allowed to become basic in any subsequent tableau and may, in fact, be dropped from the tableau. The drop of such a primal variable corresponds, obviously, to the removal of a specific constraint in the multi-dimensional dual.

These observations were implemented in the multidimensional algorithm (Ignizio,1974)

Step 1: Establish the multi-dimensional dual as given in (7) and (8).

Step 2: Form the linear programming model from (7) and (8) which includes only the $k^{th}$ right-hand side vector of (8). Solve using any conventional simplex algorithm. If $k = K$, go to step 4. Otherwise, go to step 3.

Step 3: For the linear programming model previously solved, remove all non-binding con strains. If the subsequent model has no constraints, go to step 4. Otherwise, set $k = k + 1$ and return to step 2.

Step 4: The present solution is that which is optimal for the multidimensional dual and $k^{th}$ right-hand side. The optimal solution to the LGP primal is given by the shadow prices as associated with the initial set of basic variables for the $k^{th}$primal model.

**Example 5.2.1.(Ignizio,1985)**

We shall assume that we are concerned with the problem of a specific, high tech firm. Although this firm produce numerous items, their particular problem is in regard to the manufacture of just two of these product.

These products, designated for security as "$x_1$" and "$x_2$", are produced in one isolated sector of the plant, via an extremely complex process as carried out .on an exceptionally delicate piece of machinery .

Ones an item is produced, we have just 24 hours, at the maximum, to ship and install the item at a remote governmental installation .

That is unless the finished unit of either product $x_1$ or $x_2$ is installed within 24 hour of its manufacture, the product cannot be enhanced chemically and must be scrapped via an extra ordinarily expensive and time- consuming process; a process that would, in fact drive our company out of business .

The firm has contract with the government to supply up to 30 unit per day of product $x_1$ and up to 15 unit per day of product $x_2$.

However, the government installation, recognizing the delicate nature of the manufacturing process, realizes that receipt of exactly 30 and 15 unit of $x_1$ and $x_2$ ,respectively, is unlikely.

The firm makes an estimated profit per unit of 800 dollar for $x_1$ and 1200 dollar for $x_2$.

They state that they certainly wish to maximize their daily profit.

On the single, specially designed processor, it takes just one minute to produce each unit of $x_1$ and two minutes for each $x_2$.

However, due to the delicate nature of the machine, the firm would like to run it no more than 40 minutes per every 24-hour period.

In the time during which the machine is not running, it may be adjusted and fine turned so as to satisfy the almost critical manufacturing requirements.

Thus, although the machine could conceivably be run for more than 40 minutes per day this would not be highly desirable to the firm.

To model this problem, in baseline form, we shall first define our structural variables:

$x_1$ = number of units of product $x_1$ produced per day;

$x_2$ = number of units of product $x_2$ produced per day.

We next form our objectives and goals, as a function of a structural variable

Our first set of goals will be that of "market demand," the daily (upper) requirements of the government installation. Thus:

$x_1 \leq 30$ (daily demand for $x_1$) (1)

$x_2 \leq 15$ (daily demand for $x_2$) (2

Note carefully that the government, although wanting the upper limits, will accept somewhat fewer units. Further, recall the virtual disaster that would be associated with producing *more* than the daily demands.

The profit objective may be written as follows:

$maximize\ 800x_1 + 1200x_2$ (daily profit)

However, this would be poor modeling practice. That is, in mathematical programming one should always attempt to scale all coefficients so that the difference between the largest and smallest coefficient is minimized. Thus, a more desirable form of the profit objective is

$maximize\ 8x_1 + 12x_2$(daily profit in100 dollar) (3)

Next we note that we would like to limit production time per day to 40 minutes total.

Although the firm does indicate some flexibility about this limit .thus we write this goal as

$x_1 + 2x_2 \leq 40$ (Daily production time) (4)

Finally, although not explicitly mentioned in our problem description, the firm would obviously like to produce as close to 30 unit per day of $x_1$ and 15 unit per day of $x_2$ .in doing so they not only increase their profit but also keep the customer happy .we shall wait for a moment before actually formulating these last goals. however not that they are associated with (1) and (2).

Our next step is to convert any objective into goals .the only objective listed in (1)-(4) is that of maximizing profit .assume that the firm aspired daily profit from these two products is 100,000 dollar we convert

maximize $8x_1 + 12x_2$ in to

$$8x_1 + 12x_2 \geq 1000 \qquad (5)$$

We are now ready to rank order all goals, in conjunction with discussions with the firm's decision makers. For purpose of discussion, we shall assume that the order of presentation coincides with the order of preference. Further, it is obvious that the first two goals (daily requirements) are the only ones that are rigid in this problem. Thus, letting $P_k$ refer to the $k^{th}$ priority or rank:

$P_1$: produce no more items per day of each item than demanded.

$P_2$: achieve a profit of $100,000 per day, or more.

$P_3$: attempt to keep processing time to 40 minutes or less per day.

After including the necessary goal deviation variables and forming the achievement function, we develop the final form of the LGP model as shown below.

Find $x$ so as to

$$\min \mathrm{u}^{\mathrm{T}} = \{(\mathrm{d}_1{}^+ + \mathrm{d}_2{}^+), \mathrm{d}_3{}^-, \mathrm{d}_4{}^+\}$$

Subject to

$$x_1 + 0x_2 + d_1{}^- - d_1{}^+ = 30$$

$$0x_1 + x_2 + d_2{}^- - d_2{}^+ = 15$$

$$8x_1 + 12x_2 + d_3{}^- - d_3{}^+ = 1000$$

$$x_1 + 2x_2 + d_4{}^- - d_4{}^+ = 40$$

$$x, d_i{}^-, d_i{}^+ \geq 0$$

**Solution**: However, the above form of the model is not convenient to work with and thus we replace it by the more general form. We use revised simplex method algorism to change matrix form.

Find $v$ so as to

$$\min \mathrm{u}^{\mathrm{T}} = \{\mathrm{c}^{(1)^{\mathrm{T}}}\mathrm{v}, \mathrm{c}^{(2)^{\mathrm{T}}}\mathrm{v}, \mathrm{c}^{(3)^{\mathrm{T}}}\mathrm{v}\}$$

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

$$\mathrm{v} \geq 0, c^{(k)^T} \geq 0$$

$$\text{Where } v^T = (x_1 x_2 : d_1{}^- d_2{}^- d_3{}^- d_4{}^- : d_1{}^+ d_2{}^+ d_3{}^+ d_4{}^+) \text{ or}$$

$$v^T = (v_1 v_2 : v_3 v_4 v_5 v_6 : v_7 v_8 v_9 v_{10})$$

$$\text{and } c^{(1)^T} = (0\ 0 : 0\ 0\ 0\ 0 : 1\ 1\ 0\ 0)$$

$$c^{(2)^T} = (0\ 0 : 0\ 0\ 1\ 0 : 0\ 0\ 0\ 0)$$

$$c^{(3)^T} = (0\ 0 : 0\ 0\ 0\ 0 : 0\ 0\ 0\ 1)$$

$$\text{Matrix } A = \begin{pmatrix} 1\ 0 : 1\ 0\ 0\ 0 : -1\ 0\ 0\ 0 \\ 0\ 1 : 0\ 1\ 0\ 0 : 0 -1\ 0\ 0 \\ 8\ 12 : 0\ 0\ 1\ 0 : 0\ 0 -1\ 0 \\ 1\ 2 : 0\ 0\ 0\ 1 : 0\ 0\ 0 -1 \end{pmatrix} \quad , \quad b = \begin{pmatrix} 30 \\ 15 \\ 1000 \\ 40 \end{pmatrix} = \text{right hand side}$$

Introduce the slack variables $v_3 v_4 v_5 v_6$

The initial basis is $v_B = d_i^{-}$=Identity matrix

$I = B^{-1=}$The inverse of the present basis;

$\beta$ is the present right-hand side $(v_B = B^{-1}b)$

$u$ is achievement vector

$$\beta = b = \begin{pmatrix} 30 \\ 15 \\ 1000 \\ 40 \end{pmatrix} = b$$

$$\pi^{(k)^T} = c_B{}^{(k)^T} B^{-1}$$

$$u_k = c_B{}^{(k)^T} B^{-1} b = \pi^{(k)^T} \beta$$

Table 2:The primal Initial table of example 5.1.1

| $v_B$ | Basis inverse | rhs |
|---|---|---|
| $v_3$ | 1 0 0 0 | 30 |
| $v_4$ | 0 1 0 0 | 15 |
| $v_5$ | 0 0 1 0 | 1000 |
| $v_6$ | 0 0 0 1 | 40 |
| $\pi^{(1)^T}$ | 0 0 0 0 | 0 |

$$\pi^{(1)^T} = c_B{}^{(1)^T} B^{-1} = (0\ \ 0\ \ 0\ \ 0) B^{-1} = (0\ \ 0\ \ 0\ \ 0)$$

$$u_1 = c_B{}^{(1)^T} B^{-1} b = \pi^{(1)^T} \beta = 0$$

$$d_1{}^{(1)} = \pi^{(1)^T} a_1 - c_1{}^{(1)} = (0\ \ 0\ \ 0\ \ 0) \begin{pmatrix} 1 \\ 0 \\ 8 \\ 1 \end{pmatrix} - 0 = 0$$

$$d_2{}^{(1)} = 0$$

$$d_7^{(1)} = -1$$

$$d_8^{(1)} = -1$$

$$d_9^{(1)} = 0$$

$$d_{10}^{(1)} = 0$$

We note that there are no positive valued $d_j^{(1)}$element for the set of non-basic and unchecked variables. Thus we move to step 8.Checked variables shall never be candidates to enter any subsequent basis then go to for $k = 2$

Table 3: The primal first iteration of example 5.1.1

| $v_B$ | Basis inverse | rhs |
|---|---|---|
| $v_3$ | 1 0 0 0 | 30 |
| $v_4$ | 0 1 0 0 | 15 |
| $v_5$ | 0 0 1 0 | 1000 |
| $v_6$ | 0 0 0 1 | 40 |
| □$^{(2)T}$ | 0 0 1 0 | 1000 |

Moving to step 3 we compute the value of $d_j^{(2)}$ for $v_1, v_2 v_9, v_{10}$

$$d_1^{(2)} = \pi^{(2)T} a_1 - c_1^{(2)} = (0\ \ 0\ \ 1\ \ 0)\begin{pmatrix}1\\0\\8\\1\end{pmatrix} - 0 = 8$$

$$d_2^{(2)} = (0\ \ 0\ \ 1\ \ 0)\begin{pmatrix}0\\1\\12\\2\end{pmatrix} - 0 = 12$$

$$d_9^{(2)} = (0\ \ 0\ \ 1\ \ 0)\begin{pmatrix}0\\0\\-1\\0\end{pmatrix} - 0 = -1$$

$$d_{10}^{(2)} = (0\ \ 0\ \ 1\ \ 0)\begin{pmatrix}0\\0\\0\\-1\end{pmatrix} - 0 = 0$$

Then select $v_2$ entering variable that is $q = 2$

We next update the entering column, $a_2$ in step5

$$a_q = a_2 = B^{-1}\delta_2$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0\,1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 12 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 12 \\ 2 \end{pmatrix} = \begin{pmatrix} a_{1,2} \\ a_{2,2} \\ a_{3,2} \\ a_{4,2} \end{pmatrix}$$

Moving to step 6 the leaving basic variable is determined that is

$$\frac{\beta_p}{a_{p,2}} = \min\left\{\frac{\beta_1}{a_{1,2}}, \frac{\beta_2}{a_{2,2}}, \frac{\beta_3}{a_{3,2}}, \frac{\beta_4}{a_{4,2}},\right\} \quad a_{i,2} > 0$$

Table 4: The primal second iteration of example 5.1.1

| $v_B$ | Basis inverse | rhs | $a_2$ | $\theta$ |
|---|---|---|---|---|
| $v_3$ | 1 0 0 0 | 30 | 0 | ----- |
| $v_4$ | 0 1 0 0 | 15 | 1 | 15 |
| $v_5$ | 0 0 1 0 | 1000 | 12 | $\frac{1000}{12}$ |
| $v_6$ | 0 0 0 1 | 40 | 2 | 40/2 |
| $\pi^{(2)^T}$ | 0 0 1 0 | 1000 | 12 | 1000 /12 |

That is $\theta$ simply the set of ratio and used to determine the leaving basic variable.

$v_2, v_4$ is entering and leaving variables respectively.

Table 5:The primal $3^{rd}$ iteration of example 5.1.1

| $v_B$ | basis inverse | rhs |
|---|---|---|
| $v_3$ | 1 0 0 0 | 30 |
| $v_2$ | 0 1 0 0 | 15 |
| $v_5$ | 0 −12 1 0 | 820 |
| $v_6$ | 0 −2 0 1 | 10 |
| $\square^{(2)^T}$ | 0 −12 1 0 | 820 |

Before proceeding to step 3 notes carefully that $v_2$ has replaced $v_4$ in the position of the second basic variable.

Step3. $$d_1^{(2)} = \pi^{(2)^T} a_1 - c_1^{(2)} = (0\ -12\ \ 1\ \ 0)\begin{pmatrix}1\\0\\8\\1\end{pmatrix} - 0 = 8$$

$$d_4^{(2)} = (0-\ \ 12\ \ 1\ \ 0)\begin{pmatrix}0\\1\\0\\0\end{pmatrix} - 0 = -12$$

$$d_9^{(2)} = (0-12\ \ 1\ \ 0)\begin{pmatrix}0\\0\\-1\\0\end{pmatrix} - 0 = -1$$

$$d_{10}^{(2)} = (0\ -12\ \ 1\ \ 0)\begin{pmatrix}0\\0\\0\\-1\end{pmatrix} - 0 = 0$$

Step 4.$v_1$is entering variable and $q = 1$.We next update the entering column,$a_1$ in step5

$$a_q = B^{-1}a_1$$

$$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & -12 & 1 & 0\\ 0 & -2 & 0 & 1\end{pmatrix}\begin{pmatrix}1\\0\\8\\1\end{pmatrix} = \begin{pmatrix}1\\0\\8\\1\end{pmatrix} = \begin{pmatrix}a_{1,1}\\a_{2,1}\\a_{3,1}\\a_{4,1}\end{pmatrix}$$

$$\frac{\beta_p}{a_{p,1}} = \min\left\{\frac{\beta_1}{a_{1,1}}, \frac{\beta_2}{a_{2,1}}, \frac{\beta_3}{a_{3,1}}, \frac{\beta_4}{a_{4,1}},\right\} \quad a_{i,2} > 0$$

Table 6: The primal $4^{th}$ iteration of example 5.1.1

| $v_B$ | Basis inverse | rhs | $a_2$ | $\theta$ |
|---|---|---|---|---|
| $v_3$ | 1 0 0 0 | 30 | 1 | 30 |
| $v_2$ | 0 1 0 0 | 15 | 0 | $--$ |
| $v_5$ | 0 $-12$ 1 0 | 820 | 8 | $\frac{820}{8}$ |
| $v_6$ | 0 $-2$ 0 1 | 10 | 1 | 10/1 |
| $\pi^{(2)^T}$ | 0 $-12$ 1 0 | 820 | 8 | |

Consequently $v_6$ the smallest $\theta$ ratio is the leaving variable and $v_1$ entering variable

Table 7:The primal $5^{th}$ iteration of example 5.1.1

| $v_B$ | basis inverse | rhs |
|---|---|---|
| $v_3$ | 1 2 0 1 | 20 |
| $v_2$ | 0 1 0 0 | 15 |
| $v_5$ | 0 4 1 − 8 | 740 |
| $v_1$ | 0 2 0 1 | 10 |
| □$^{(2)^T}$ | 0 4 1 − 8 | 740 |

Step3. we determine that

$$d_4^{(2)} = (0\ \ 4\ \ 1\ \ -8)\begin{pmatrix}0\\1\\0\\0\end{pmatrix} - 0 = 4$$

$$d_6^{(2)} = (0\ \ 4\ \ 1-\ \ 8)\begin{pmatrix}0\\0\\0\\1\end{pmatrix} - 0 = -8$$

$$d_9^{(2)} = (0\ \ 4\ \ 1\ \ -8)\begin{pmatrix}0\\0\\-1\\0\end{pmatrix} - 0 = -1$$

$$d_{10}^{(2)} = (0\ \ 4\ \ 1\ \ -8)\begin{pmatrix}0\\0\\0\\-1\end{pmatrix} - 0 = 8$$

Step 4.the entering variable is $v_{10}$ thus $q = 10$

Table 8 : The primal $6^{th}$ iteration of example 5.1.1

| | Basis inverse | rhs | $a_2$ | $\theta$ |
|---|---|---|---|---|
| $v_3$ | 1  2  0  −1 | 20 | 1 | 20 |
| $v_2$ | 0  1  0  0 | 15 | 0 | −− |
| $v_5$ | 0  4  1  −8 | 740 | 8 | $\frac{740}{8}$ |
| $v_1$ | 0  −2  0  1 | 10 | −1 | −− |
| $\pi^{(2)^T}$ | 0  4  1  −8 | 740 | 8 | −− |

Consequently $v_3$ the smallest $\theta$ ratio is the leaving variable an $v_{10}$ entering variable

The pivoting process leads to

**Table 9 :The primal 7th iteration of example 5.1.1**

| $v_B$ | basis inverse | rhs |
|---|---|---|
| $v_{10}$ | 1  2  0  −1 | 20 |
| $v_2$ | 0  1  0  0 | 15 |
| $v_5$ | −8  −12  1  0 | 580 |
| $v_1$ | 1  0  0  0 | 30 |
| $\pi^{(2)^T}$ | 0  −8  −12  −8 | 580 |

Step3. we determine that

$$d_3^{(2)} = -8$$

$$d_4^{(2)} = -12$$

$$d_6^{(2)} = 0$$

$$d_9^{(2)} = -1$$

Step 4.all shadow price vectors are non positive .Neither stopping rule is met so we cheek the variable$v_3, v_4, v_9$.Then set $k = 3$

Table 10: The primal 8th iteration of example 5.1.1

| $v_B$ | basis inverse | rhs |
|---|---|---|
| $v_{10}$ | 1 2 0 − 1 | 20 |
| $v_2$ | 0 1 0 0 | 15 |
| $v_5$ | − 8 − 12 1 0 | 580 |
| $v_1$ | 1 0 0 0 | 30 |
| $\pi^{(3)^T}$ | 1 2 0 − 1 | 20 |

Step 3.We next evaluate $d_j^{(3)}$ for all non basic unchecked variables that is $d_6^{(3)} = -1$.

Step 4 .go to step 8.In step 4 there is no positive shadow price vector then go to step 8.

Further, from the shadow prices for the final LP tableau, we may determine that the optimal primal program is: Optimal solutions are $v_{10}^* = 30, v_2^* = 15, v_5^* = 580, v_1^* = 20$

The achievement vector is determined by

$$u_k = c^{(k)^T} B^{-1} b$$

$u_1^* = (0\ \ 0\ \ 0\ 0)\begin{pmatrix} 1 & 2 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ -8 & -12 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 30 \\ 15 \\ 1000 \\ 40 \end{pmatrix} = 0$ ; Thus all rigid constraints are satisfied

$u_2^* = (0\ \ 0\ \ 1\ \ 0)\begin{pmatrix} 1 & 2 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ -8 & -12 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 30 \\ 15 \\ 1000 \\ 40 \end{pmatrix} = 580$ ; The result is 580 units below the goal of 1000.

$u_3^* = (1\ \ 2\ \ 0\ \ -1)\begin{pmatrix} 1 & 2 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ -8 & -12 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 30 \\ 15 \\ 1000 \\ 40 \end{pmatrix} = 20$ ; The result is 20 units over the aspired goal of 40.

Solution of Dual form:

$$\min z = \{(d_1^+ + d_2^+), d_3^-, d_4^+\}$$

Subject to $x_1 + d_1^- - d_1^+ = 30$

$$x_2 + d_2^{-} - d_2^{+} = 15$$

$$8\mathrm{x}_1 + 12\mathrm{x}_2 + \mathrm{d}_3^{-} - \mathrm{d}_3^{+} = 1000$$

$$\mathrm{x}_1 + 2\mathrm{x}_2 + \mathrm{d}_4^{-} - \mathrm{d}_4^{+} = 40$$

$$\mathrm{x}, \mathrm{d}_\mathrm{i}^{-}, \mathrm{d}_\mathrm{i}^{+} \geq 0 \text{ for } \mathrm{i} = 1, \ldots, 4$$

Initial basis always consists of the negative deviation variables.

$$\mathrm{B} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathrm{N} = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 8 & 12 & 0 & 0 & -1 & 0 \\ 1 & 2 & 0 & 0 & 0 & -1 \end{pmatrix}$$

$$\mathrm{c_B}^{(1)^\mathrm{T}} = (0 \quad 0 \quad 0 \quad 0)$$

$$\mathrm{c_B}^{(2)^\mathrm{T}} = (0 \quad 0 \quad 1 \quad 0)$$

$$\mathrm{c_B}^{(3)^\mathrm{T}} = (0 \quad 0 \quad 0 \quad 0)$$

$$\mathrm{c_N}^{(1)^\mathrm{T}} = (0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0)$$

$$\mathrm{c_N}^{(2)^\mathrm{T}} = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)$$

$$\mathrm{c_N}^{(3)^\mathrm{T}} = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1)$$

Dual

$$\max w = (-30 - 15 - 1000 - 40)Y + \{0,1000,0\}$$

$$\text{Subject to} \begin{bmatrix} -1 & 0 & -8 & -1 \\ 0 & -1 & -12 & -2 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} y \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -8 \\ -12 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

The development of the multi-dimensional dual may be further clarified if one recognizes that, for the primal LGP model given above, the following relationships hold:

$$\mathrm{c}^{(1,\mathrm{k})} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 8 & 12 \\ 1 & 2 \end{bmatrix} \qquad ,$$

$$\mathrm{b}^{(1,\mathrm{k})} = \begin{bmatrix} 30 \\ 15 \\ 1000 \\ 40 \end{bmatrix}$$

Solution for $k = 1$

$$\max w = (-30 - 15 - 1000 - 40)Y + \{0\}$$

$$\text{Subject to} \begin{bmatrix} -1 & 0 & -8 & -1 \\ 0 & -1 & -12 & -2 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} y^{(1)} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$y^{(1)}$ indicate the first right hand side

$$\mathrm{v}^{\mathrm{T}} = \left(\mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3, \mathrm{y}_4, \mathrm{y}_5, \mathrm{y}_6, \mathrm{y}_{7,}\mathrm{y}_8, \mathrm{y}_9, \mathrm{y}_{10}, \mathrm{y}_{11}, \mathrm{y}_{12}, \mathrm{y}_{13}, \mathrm{y}_{14}\right)$$

$$\mathrm{v}^{\mathrm{T}} = \left(\mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3, \mathrm{y}_4, \mathrm{s}_1, \mathrm{s}_2, \mathrm{s}_{3,}\mathrm{s}_4, \mathrm{s}_5, \mathrm{s}_6, \mathrm{s}_7, \mathrm{s}_8, \mathrm{s}_9, \mathrm{s}_{10}\right)$$

$$-\mathrm{y}_1 - 0\mathrm{y}_2 - 8\mathrm{y}_3 - \mathrm{y}_4 + \mathrm{s}_1 = 0$$

$$0\mathrm{y}_1 - \mathrm{y}_2 - 12\mathrm{y}_3 - 2\mathrm{y}_4 + \mathrm{s}_2 = 0$$

$$-\mathrm{y}_1 + \mathrm{s}_3 = 0$$

$$-\mathrm{y}_2 + \mathrm{s}_4 = 0$$

$$-\mathrm{y}_3 + \mathrm{s}_5 = 0$$

$$-\mathrm{y}_4 + \mathrm{s}_6 = 0$$

$$\mathrm{y}_1 + \mathrm{s}_7 = 1$$

$$\mathrm{y}_2 + \mathrm{s}_8 = 1$$

$$sy_3 + s_9 = 0$$

$$y_4 + s_{10} = 0$$

Table 11:The Dual $1^{st}$ iteration of example 5.1.1

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | $s_9$ | $s_{10}$ | rhs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| z | −30 | −15 | −1000 | −40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_1$ | −1 | 0 | −8 | −1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_2$ | 0 | −1 | −12 | −2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_3$ | −1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_4$ | 0 | −1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_5$ | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_6$ | 0 | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $s_7$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| $s_8$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| $s_9$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| $s_{10}$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Step1.Initialization. $v_B = d_i^{-}$.Thus $B = I, B^{-1} = I$ and $\beta = b$.Set $k = 1$.Initially all variables are unchecked.

Step2. Develop the pricing vector. Determine:

$$\pi^{(k)^T} = c_B{}^{(k)^T} B^{-1} \text{ Where } B^{-1} = I$$

$$\pi^{(k)^T} = (0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0)$$

Step3. Price out all unchecked, non-basic columns. Compute:

$d_j{}^{(k)} = \pi^{(k)^T} a_j - c_j{}^{(k)}$ For all $j \in N$ where $N$ is the set of non-basic and unchecked variables.

Where $a_j$ is the $j^{th}$column vector of the constraint and $c_j{}^{(k)}$ is constant coefficient of the basic variable with objective function. Then $d_1{}^{(1)} = 30$

$$d_2{}^{(1)} = 15$$

$$d_3{}^{(1)} = 1000$$

$$d_4^{(1)} = 40$$

Step 4.Selection of entering non-basic variables. Examine those $d_j^{(1)}$as computed in step3. If none are negative, proceed to step 8. Otherwise, select the non-basic variable with the most negative $d_j^{(1)}$(ties may be broken arbitrarily) as the entering variable. Constraint seven and eight are non binding in the solution. In this step there is no negative shadow price vector then go to step 8.

From the revised simplex tableau

$$\begin{bmatrix} v_1^{(1)} \\ v_2^{(1)} \\ v_3^{(1)} \\ v_4^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ Are optimal solutions}$$

$$w^{(1)^*} = z^* = 0$$

For $k = 2$

$$\max\ w = (-30 - 15 - 1000 - 40)Y + \{1000\}$$

$$\text{Subject to} \begin{bmatrix} -1 & 0 & -8 & -1 \\ 0 & -1 & -12 & -2 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} y^{(2)} \leq \begin{bmatrix} -8 \\ -12 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$y^{(2)}$ indicates second right hand side

$$v^T = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}, y_{12}, y_{13}, y_{14})$$

$$v^T = (y_1, y_2, y_3, y_4, a_1, a_2, s_3, s_4, s_5, s_6, s_7, s_8, s_1, s_2)$$

Constraint one and two are multiply by negative one and change the inequality sign .

$$y_1 + 0y_2 + 8y_3 + y_4 - s_1 + a_1 = 8$$

$$0y_1 + y_2 + 12y_3 + 2y_4 - s_2 + a_2 = 12$$

$$-y_1 + s_3 = 0$$

$$-y_2 + s_4 = 0$$

$$-y_3 + s_5 = 0$$

$$-y_4 + s_6 = 0$$

$$y_3 + s_7 = 1$$

$$y_4 + s_8 = 0$$

Table 12: The Dual $2^{nd}$ iteration of example 5.1.1

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $a_1$ | $a_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | $s_1$ | $s_2$ | Rs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| z | −30 | −15 | −1000 | −40 | $-M$ | $-M$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| $a_1$ | 1 | 0 | 8 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 0 | 8 |
| $a_2$ | 0 | 1 | 12 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 12 |
| $s_3$ | −1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_4$ | 0 | −1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_5$ | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_6$ | 0 | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $s_7$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| $s_8$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

Step1.Initialization. $v_B = d_i^-$.Thus,$B = I, B^{-1} = I$ and $\beta = b$.Set $k = 2$.Initially, all variables are unchecked.

Step 2.Develop the pricing vector. Determine:

$\pi^{(k)^T} = c_B{}^{(k)^T} B^{-1}$ Where $B^{-1} = I$ for k=2

$\pi^{(k)^T} = (-M \ \ -M \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 0)$ M is large positive number

Step 3.Price out all unchecked non-basic columns. Compute:

$d_j{}^{(k)} = \pi^{(k)^T} a_j - c_j{}^{(k)}$ For all $j \in N$ where $N$ is the set of non-basic and unchecked variables. Where $a_j$ is the $j^{th}$column vector of the constraint and $c_j{}^{(k)}$ is constant coefficient of the basic variable with objective function. Then

$$d_1{}^{(2)} = (-M \ \ -M \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 0)\begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - (-30) = 30 - \mathrm{M}$$

$$d_2{}^{(2)} = 15 - \mathrm{M}$$

$$d_3{}^{(2)} = 1000 - 20\mathrm{M}$$

$$d_4{}^{(2)} = 40 - 3\mathrm{M}$$

$$d_{13}{}^{(2)} = \mathrm{M}$$

$$d_{14}{}^{(2)} = \mathrm{M}$$

Therefore $k = 3$ and $v_3$enters the basis

Step 4.Selection of entering non-basic variables. Examine those $d_j{}^{(k)}$as computed in step 3. If none are negative proceed to step 8. Otherwise, select the non-basic variable with the most negative $d_j{}^{(2)}$(ties may be broken arbitrarily) as the entering variable. Designate this variable as $v_q$.then $d_3{}^{(2)}$ has highest negative value.

Step 5.Update the entering column. Evaluate:

$$a_q = B^{-1}a_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 12 \\ 0 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 8 \\ 12 \\ 0 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Table 13:The Dual $3^{rd}$ iteration of example 5.1.1

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $a_1$ | $a_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | $s_1$ | $s_2$ | Rhs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| z | $-30$ | $-15$ | $-2000 + 20m$ | $-40$ | $-M$ | $-M$ | 0 | 0 | 0 | 0 | $-1 + 2$ | 0 | 0 | 0 | $-1000 + 20m$ |
| $a_1$ | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | -8 | 0 | $-1$ | 0 | 0 |
| $a_2$ | 0 | 1 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | $-1$ | 0 | 0 | $-1$ | 0 |
| $s_3$ | $-1$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_4$ | 0 | $-1$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_5$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| $s_6$ | 0 | 0 | 0 | $-1$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $v_3$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| $s_8$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

$s_7$Leaving and $v_3$entering variables

$$d_1{}^{(2)} = (-M \ -M \ 0 \ 0 \ 0 \ 0 \ (-1000 + 20m\,) \ 0\,) \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - (-30) = 30 - \mathrm{M}$$

$$d_2{}^{(1)} = 15 - \mathrm{M}$$

$$d_{11}{}^{(1)} = 0$$

$$d_4{}^{(1)} = 40 - 3\mathrm{M}$$

$$d_{13}{}^{(2)} = \mathrm{M}$$

$$d_{14}{}^{(1)} = \mathrm{M}$$

Therefore $k = 4$ and $v_4$enters the basis

Step 4.Selection of entering non-basic variables. Examine those $d_j{}^{(k)}$as computed in step 3. If none are negative proceed to step 8. Otherwise, select the non-basic variable with the most negative $d_j{}^{(2)}$(ties may be broken arbitrarily) as the entering variable. Designate this variable as $v_q$.then $d_4{}^{(2)}$ has highest negative value

Step 5.Update the entering column. Evaluate:

$$a_q = B^{-1}a_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -8 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -12 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 1 \end{bmatrix}$$

Therefore $k = 4$ and $v_4$ enter the basis

Table 14:The Dual 4$^{\text{th}}$iteration of example 5.1.1

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $a_1$ | $a_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | $s_1$ | $s_2$ | Rhs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| z | $-70+3m$ | $-15$ | $-2000+20m$ | $-80+3m$ | $-40+2m$ | $-M$ | 0 | 0 | 0 | 0 | 0 | 0 | $40-3m$ | 0 | $-1000+20m$ |
| $v_4$ | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | -8 | 0 | $-1$ | 0 | 0 |
| $a_2$ | - 2 | 1 | 0 | 0 | -2 | 1 | 0 | 0 | 0 | 0 | 4 | 0 | 2 | $-1$ | 0 |
| $s_3$ | $-1$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_4$ | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_5$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| $s_6$ | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -8 | 0 | -1 | 0 | 0 |
| $v_3$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| $s_8$ | - 1 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 8 | 1 | 1 | 0 | 0 |

$a_1$Leaving and

$v_4$entering variables

$$d_1{}^{(2)} = (-40+2m)\ -M\ \ 0\ \ 0\ \ 0\ \ 0\ (-680-4m\ )\ 0\ ) \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - (-70+3\mathrm{m}) = 30 - \mathrm{M}$$

$$d_2{}^{(1)} = 15 - \mathrm{M}$$

$$d_{11}{}^{(1)} = 0$$

$$d_5{}^{(1)} = 0$$

$$d_{13}{}^{(2)} = \mathrm{M}$$

$$d_{14}{}^{(2)} = \mathrm{M}$$

Therefore $k = 2$ and $v_2$enters the basis

Step 4.Selection of entering non-basic variables. Examine those $d_j{}^{(k)}$as computed in step 3. If none are negative proceed to step 8. Otherwise select the non-basic variable with the most negative $d_j{}^{(k)}$(ties may be broken arbitrarily) as the entering variable. Designate this variable as $v_q$.then $d_2{}^{(2)}$ has highest negative value

Step 5.Update the entering column. Evaluate:

$$a_q = B^{-1}a_q = B^{-1}a_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -8 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & -8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 8 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Table 15:The Dual $5^{\text{th}}$ iteration of example 5.1.1

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $a_1$ | $a_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | $s_1$ | $s_2$ | Rhs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| z | $-40+m$ | -30+m | $-2000+20m$ | $-80+3m$ | -10 | -15 | 0 | 0 | 0 | 0 | 0 | 0 | $10-m$ | $-15+m$ | $-1000+20m$ |
| $v_4$ | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | -8 | 0 | $-1$ | 0 | 0 |
| $v_2$ | -2 | 1 | 0 | 0 | -2 | 1 | 0 | 0 | 0 | 0 | 4 | 0 | 2 | $-1$ | 0 |
| $s_3$ | -1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $s_4$ | -2 | 0 | 0 | 0 | -2 | 1 | 0 | 1 | 0 | 0 | 4 | 0 | 2 | -1 | 0 |
| $s_5$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| $s_6$ | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -8 | 0 | -1 | 0 | 0 |
| $v_3$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| $s_8$ | -1 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 8 | 1 | 1 | 0 | 0 |

The solution obtained at the fifth iteration of the dual simplex method was ${v_2}^{(2)}, {v_3}^{(2)}, {v_4}^{(2)} = 0,1,0$ respectivly.But using the LINGO software

$$\begin{bmatrix} {v_1}^{(2)} \\ {v_2}^{(2)} \\ {v_3}^{(2)} \\ {v_4}^{(2)} \end{bmatrix} = \begin{bmatrix} 8 \\ 12 \\ 0 \\ 0 \end{bmatrix} \text{ Are optimal solutions}$$

$w^{(2)^*} = z^* = -420 + 1000 = 580$ Optimal value

For k=3

$$\begin{bmatrix} {v_1}^{(3)} \\ {v_2}^{(3)} \\ {v_3}^{(3)} \\ {v_4}^{(3)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ Are optimal solutions}$$

$w^{(3)^*} = z^* = 0$ optimal value

**Example 5.1.2 :Ignizio (1985)**

$$min\ z = \{(d_1{}^+ + d_2{}^+), (2d_3{}^- + 3d_4{}^-)\}$$

Subject to $\quad x_1 + x_2 + d_1{}^- - d_1{}^+ = 12$

$$2x_1 + x_2 + d_2{}^- - d_2{}^+ = 20$$

$$16x_1 + 10x_2 + d_3{}^- - d_3{}^+ = 160$$

$$3x_1 + 5x_2 + d_4{}^- - d_4{}^+ = 60$$

$$x, d_i{}^+, d_i{}^- \geq 0\ i = 1,2,3,4$$

Dual

$$max\ z = (-12\ \ -20\ \ -160\ \ -60)v + \{0,500\}$$

Subject to
$$\begin{bmatrix} -1 & -2 & -16 & -3 \\ -1 & -1 & -10 & -5 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} v \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -41 \\ -35 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 3 \end{bmatrix}$$

These approaches were each based on the transformation of all objectives into goals by means of the establishment of an aspiration level or target .First changed to standard form of linear equation by adding logical structural variables.

Solution for k=1

$$max\ z^{(1)} = (-12\ \ -20\ \ -160\ \ -60)v_i{}^{(1)} + \{0\} \text{ for } i = 1,..,4$$

Subject to $\quad -v_1 - 2v_2 - 16v_3 - 3v_4 + s_1 = 0$

$$-v_1 - v_2 - 10v_3 - 5v_4 + s_2 = 0$$

$$-v_1 + s_3 = 0$$

$$-v_2 + s_4 = 0$$

$$-v_3 + s_5 = 0$$

$$-v_4 + s_6 = 0$$

$$v_1 + s_7 = 1$$

$$v_2 + s_8 = 1$$

$$v_3 + s_9 = 0$$

$$v_4 + s_{10} = 0$$

Using the LINGO soft ware

$$\begin{bmatrix} v_1{}^{(1)} \\ v_2{}^{(1)} \\ v_3{}^{(1)} \\ v_4{}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ Are optimal solutions}$$

$$w^{(1)^*} = z^* = 0$$

Solution for k=2

$$max\ z = (-12\ \ -20\ -160\ -60)v_i{}^{(2)} + \{500\}$$

$$\text{Subject to} \begin{bmatrix} -1\ -2\ \ -16\ \ -3 \\ -1\ -1\ -10\ -5 \\ -1\ 0\ 0\ 0 \\ 0\ -1\ 0\ 0 \\ 0\ 0\ -1\ 0 \\ 0\ 0\ 0\ -1 \\ 0\ 0\ 1\ 0 \\ 0\ 0\ 0\ 1 \end{bmatrix} v \leq, \begin{bmatrix} -41 \\ -35 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 3 \end{bmatrix}$$

$$max\ z^{(2)} = (-12\ \ -20\ -160\ -60)v_i{}^{(2)} + \{500\} \text{ for } i = 1, \ldots, 4$$

Subject to

$$v_1 + 2v_2 + 16v_3 + 3v_4 - s_1 + a_1 = 41$$

$$v_1 + v_2 + 10v_3 + 5v_4 - s_2 + a_2 = 35$$

$$-v_1 + s_3 = 0$$

$$-v_2 + s_4 = 0$$

$$-v_3 + s_5 = 0$$

$$-v_4 + s_6 = 0$$

$$v_3 + s_7 = 2$$

$$v_4 + s_8 = 3$$

By using the LINGO software

$$\begin{bmatrix} v_1{}^{(1)} \\ v_2{}^{(1)} \\ v_3{}^{(1)} \\ v_4{}^{(1)} \end{bmatrix} = \begin{bmatrix} 25 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$ Are optimal solutions

$w^{(2)^*} = z^* = -460 + 500 = 40$ optimal value

## 5.2. Solving Linear Goal Programming by using Lexicographic Method

Lexicographic method optimizes the goal ones at a time starting with the highest priority goal and terminating with the lowest.

A lexicographic method is a sequential minimization of each priority whilst maintaining the minimal values reached by all higher priority level minimizations.

This is known as lexicographic or pre-emptive method as introduced and chiefly developed by (Ignizio,1994).The lexicographic structure has been criticized by some authors on the grounds of its incompatibility with utility function theory (Min and Storbeck,1991). However (Romero, 1991) points out its practicality in modeling situations where the decision maker has a pre-defined ordering of the goals in mind and does not wish to make direct trade-off comparisons between goal.

The lexicographic method is based on the minimization of deviations from the most important goal and then continues to the least important goal (Schrage, 1997).

The algorithms in testing for optimality did not consider the fact that there exists a situation where absolute value of coefficient of deviational variable under consideration that is not in the table can re-enter. If both(negative and positive deviational variables) from the same goal constraint are in the achievement function but with different priorities, then the one with higher priority will be in the basis whereas the lesser one will be placed alongside with other variables in the non basis.

Just as in the method of artificial variables, ensure that a variable of higher or equal priority that has been previously satisfied does not re-enter the basis, instead the variable with the next higher coefficient in the row enters. In Lexicographic method, the objectives can be divided into different priority classes. Here it is assumed that no two goals have equal priority. The goals are given ordinal ranking and are called preemptive priority factors. These priority factors have the relationship $p_1 \gg p_2 \gg \cdots p_m$. The coefficients $p_1, p_2, \ldots . p_m$ are not variables or parameters. They do not assume a numerical value; they simply represent levels for priorities. The procedure considers goal constraints as both the objective function and constraints. In the lexicographic method the decision maker must rank the goal of the problem in order of importance .given an n goal of situation the objective of the problem are written as

Minimize $G_1 = \rho_1$ (highest priority)

Minimize $G_n = \rho_n$ (lowest priority)

The variable $\rho_i$ is the component of deviation variable (positive and negative deviation variables) that represent goal $i$.

Consider the Lexicographic method model. The formulation for n variables, m goal constraints, $t$ deviational variables in $z$ and $L$ preemptive priority factors is detailed in (Orumie and Ebong,2014).

$$\text{Minimize } z = \sum_{k=1}^{l} p_k(d_i^-, d_i^+)$$

$$\text{Such that } \sum_{i=1}^{m} a_{ij}x_j + d_i^- - d_i^+ = b_i$$

$$x_{ij}, d_i^-, d_i^+ \geq 0$$

$$\text{For } i = 1, \ldots, .m : j = 1, \ldots, ..n$$

Where $p_k = k^{th}$ priority factor $k = 1,2, \ldots, .L,$

$p_k(d_i^-, d_i^+)$ are set of deviational variables in $z$ with the priorities attached to them. The algorithm is the order of priority, goal programming into a series of single objective of linear programming problem, while on a priority solution as the next priority constraints.

Steps for the Lexicographic method algorithm is

Step 1: set the first goal set as the current goal set

Step2: obtain a Linear Programming solution defining the current goal set as the objective function

Step3: if the current goal set is the final goal set,

**a**. set it equal to the LP objective function value obtained in Step 2, and stop. Otherwise, go to Step 4.

Step4: if the current goal set is achieved or overachieved.

**a**. set it equal to its aspiration level and add the constraint to the constraint set, go to Step 5.

**b**. Otherwise, if the value of the current goal set is underachieved, set the aspiration level of the current goal equal to the LP objective function value obtained in Step 2. Add this equation to the constraint set. Go to Step 5

Step5:set the next goal set of importance as the current goal set. Go to Step 2.

**Example 5.2.1.(Ignizio,1985)**

$$\min \mathrm{u^T} = \{(\mathrm{d_1}^+ + \mathrm{d_2}^+), \mathrm{d_3}^-, \mathrm{d_4}^+\}$$

$$\text{Subject to } x_1 + 0x_2 + d_1^- - d_1^+ = 30$$

$$0x_1 + x_2 + d_2^- - d_2^+ = 15$$

$$8x_1 + 12x_2 + d_3^- - d_3^+ = 1000$$

$$x_1 + 2x_2 + d_4^- - d_4^+ = 40$$

$$x, d_i^-, d_i^+ \geq 0.$$

Solution:

Table16:The primal $1^{st}$ iteration of example 5.2.1

| | $\mathbf{c_j}$ | **0** | **0** | **0** | $\mathbf{-1}$ | **0** | $\mathbf{-1}$ | **0** | **0** | **0** | **0** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{c_B}$ | $y_B$ | $x_1$ | $x_2$ | $d_1^-$ | $d_1^+$ | $d_2^-$ | $d_2^+$ | $d_3^-$ | $d_3^+$ | $d_4^-$ | $d_4^+$ | RHS |
| -1 | $d_1^+$ | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 30 |
| -1 | $d_2^+$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 15 |
| 0 | $d_3^-$ | 8 | 12 | 0 | 0 | 0 | 0 | 1 | $-1$ | 0 | 0 | 1000 |
| 0 | $d_4^+$ | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | $-1$ | 40 |
| | $z_j$ | $-1$ | $-1$ | $-1$ | 0 | -1 | 0 | 0 | 0 | 0 | 0 | |
| | $z_j - c_j$ | $-1$ | $-1$ | $-1$ | 1 | -1 | 1 | 0 | 0 | 0 | 0 | |

In table1 the value of the current goal set is underachieved then set the aspiration level of the current goal equal to the LP objective function value obtained in Step 2. Then add this equation in the objective function to the constraint set. The values of the first goal is underachieved then go to the next goal.

Table 17 : The primal $2^{nd}$ iteration of example 5.2.1

| $c_j$ | | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c_B$ | $y_B$ | $x_1$ | $x_2$ | $d_1^-$ | $d_1^+$ | $d_2^-$ | $d_2^+$ | $d_3^-$ | $d_3^+$ | $d_4^-$ | $d_4^+$ | RHS |
| **0** | $d_1^+$ | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 30 |
| **0** | $d_2^+$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 15 |
| **1** | $d_3^-$ | 8 | 12 | 0 | 0 | 0 | 0 | 1 | −1 | 0 | 0 | 1000 |
| **0** | $d_4^+$ | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | −1 | 40 |
| | $z_j$ | 8 | 12 | 0 | 0 | 0 | 0 | 1 | −1 | 0 | 0 | 1000 |
| | $z_j - c_j$ | 8 | 12 | 0 | 0 | 0 | 0 | 0 | −1 | 0 | 0 | |

The current goal set is achieved set it equal to its aspiration level and add the constraint to the constraint set. Then select $x_2$column because $z_j - c_j$ has highest positive value.

$$\begin{cases} \dfrac{30}{0} = \infty \\ \dfrac{15}{1} = 15 \rightarrow \\ \dfrac{1000}{12} = 83.33 \\ \dfrac{40}{2} = 20 \end{cases}$$

Then$x_2$is entering variable and $d_2^+$ is leaving variable because has smallest ratio.

$$d_3^{-}\text{ New row value}=\begin{cases} 8-\left(\frac{12*0}{1}\right)=8 \\ 12-\left(\frac{12*1}{1}\right)=0 \\ 0-\left(\frac{12*0}{1}\right)=0 \\ 0-\left(\frac{12*0}{1}\right)=0 \\ 0-\left(\frac{1*12}{1}\right)=-12 \\ 0-\left(\frac{12*0}{1}\right)=0 \\ 1-\left(\frac{12*0}{1}\right)=1 \\ -1-\left(\frac{12*0}{1}\right)=-1 \\ 0-\left(\frac{12*0}{1}\right)=0 \\ 0-\left(\frac{12*0}{1}\right)=0 \\ 1000-\left(\frac{12*15}{1}\right)=820 \end{cases}$$

$$d_4^{+}\text{ new row value}=\begin{cases} 1-\left(\frac{2*0}{1}\right)=1 \\ 2-\left(\frac{2*1}{1}\right)=0 \\ 0-\left(\frac{2*0}{1}\right)=0 \\ 0-\left(\frac{2*0}{1}\right)=0 \\ 0-\left(\frac{1*2}{1}\right)=-2 \\ 0-\left(\frac{2*0}{1}\right)=0 \\ 0-\left(\frac{2*0}{1}\right)=0 \\ 0-\left(\frac{2*0}{1}\right)=0 \\ 1-\left(\frac{2*0}{1}\right)=1 \\ -1-\left(\frac{2*0}{1}\right)=-1 \\ 40-\left(\frac{2*15}{1}\right)=10 \end{cases}$$

Solutions in table 17 are $d_1^{+}, x_2, d_3^{-}, d_4^{+} = 30$ ,15,820,10 respectivly.

Table 18 :The primal $3^{rd}$ iteration of example 5.2.1

| | $c_j$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c_B$ | $y_B$ | $x_1$ | $x_2$ | $d_1^-$ | $d_1^+$ | $d_2^-$ | $d_2^+$ | $d_3^-$ | $d_3^+$ | $d_4^-$ | $d_4^+$ | RHS |
| **0** | $d_1^+$ | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 30 |
| **0** | $x_2$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 15 |
| **1** | $d_3^-$ | 8 | 0 | 0 | 0 | −12 | 0 | 1 | −1 | 0 | 0 | 820 |
| **0** | $d_4^+$ | 1 | 0 | 0 | 0 | −2 | 0 | 0 | 0 | 1 | −1 | 10 |
| | $z_j$ | 8 | 0 | 0 | 0 | −12 | 0 | 1 | −1 | 0 | 0 | 820 |
| | $z_j - c_j$ | 8 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 0 | 0 | |

Then select$x_1$column because $z_j - c_j$ has highest value .

$$\begin{cases} \frac{30}{1} = 30 \\ \frac{15}{0} = \infty \\ \frac{820}{8} = 102.5 \\ \frac{10}{1} = 10 \end{cases}$$

Then$x_1$is entering variable and $d_4^+$ is leaving variable because has smallest ratio.

$$d_3^- \text{ new row value} = \begin{cases} 8 - \left(\frac{8*1}{1}\right) = 0 \\ 0 - \left(\frac{8*0}{1}\right) = 0 \\ 0 - \left(\frac{8*0}{1}\right) = 0 \\ 0 - \left(\frac{8*0}{1}\right) = 0 \\ -12 - \left(\frac{8*-2}{1}\right) = 4 \\ 0 - \left(\frac{8*0}{1}\right) = 0 \\ 1 - \left(\frac{8*0}{1}\right) = 1 \\ -1 - \left(\frac{8*0}{1}\right) = -1 \\ 0 - \left(\frac{8*0}{1}\right) = 0 \\ 0 - \left(\frac{8*0}{1}\right) = 0 \\ 820 - \left(\frac{8*10}{1}\right) = 740 \end{cases}$$

$$
d_1^{\;+} \text{ new row value=}
\begin{cases}
1-\left(\frac{1*1}{1}\right)=0\\
0-\left(\frac{1*0}{1}\right)=0\\
1-\left(\frac{1*0}{1}\right)=1\\
0-\left(\frac{1*0}{1}\right)=0\\
0-\left(\frac{1*-2}{1}\right)=2\\
0-\left(\frac{1*0}{1}\right)=0\\
0-\left(\frac{1*0}{1}\right)=0\\
0-\left(\frac{1*0}{1}\right)=0\\
0-\left(\frac{1*1}{1}\right)=-1\\
0-\left(\frac{1*-1}{1}\right)=1\\
30-\left(\frac{1*10}{1}\right)=20
\end{cases}
$$

Solutions in table 18 are $d_1^{\;+}, x_2, d_3^{-}, x_1 = 20, 15{,}740{,}10$ respectivly.

Table 19 : The primal $4^{th}$ iteration of example 5.2.1

| | **$c_j$** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **$c_B$** | $y_B$ | $x_1$ | $x_2$ | $d_1^{-}$ | $d_1^{\;+}$ | $d_2^{-}$ | $d_2^{\;+}$ | $d_3^{\;-}$ | $d_3^{\;+}$ | $d_4^{\;-}$ | $d_4^{\;+}$ | RHS |
| **0** | $d_1^{\;+}$ | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | $-1$ | 1 | 20 |
| **0** | $x_2$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 15 |
| **1** | $d_3^{-}$ | 0 | 0 | 0 | 0 | 4 | 0 | 1 | $-1$ | $-8$ | 8 | 740 |
| **0** | $x_1$ | 1 | 0 | 0 | 0 | $-2$ | 0 | 0 | 0 | 1 | $-1$ | 10 |
| | $z_j$ | 0 | 0 | 0 | 0 | 4 | 0 | 1 | $-1$ | $-8$ | 8 | 1000 |
| | $z_j - c_j$ | 0 | 0 | 0 | 0 | 4 | 0 | 0 | $-1$ | $-8$ | 8 | |

Then select $d_4^{\;+}$ column because $z_j - c_j$ has highest value.

$$\begin{cases} \dfrac{20}{1} = 20 \\ \dfrac{15}{0} = \infty \\ \dfrac{740}{8} = 92.5 \\ \dfrac{10}{-1} = -10 \end{cases}$$

Then $d_4{}^+$ is entering variable and $d_1{}^+$ is leaving variable because has smallest ratio.

$$d_3^- \text{ new row value=} \begin{cases} 0 - \left(\frac{8*0}{1}\right) = 0 \\ 0 - \left(\frac{8*0}{1}\right) = 0 \\ 0 - \left(\frac{8*1}{1}\right) = -8 \\ 0 - \left(\frac{8*0}{1}\right) = 0 \\ 4 - \left(\frac{8*2}{1}\right) = -12 \\ 0 - \left(\frac{8*0}{1}\right) = 0 \\ 1 - \left(\frac{8*0}{1}\right) = 1 \\ -1 - \left(\frac{8*0}{1}\right) = -1 \\ -8 - \left(\frac{8*-1}{1}\right) = 0 \\ 8 - \left(\frac{8*1}{1}\right) = 0 \\ 740 - \left(\frac{8*20}{1}\right) = 580 \end{cases}$$

$$x_1 \quad \text{New row value}=\begin{cases} 1-\left(\frac{-1*0}{1}\right)=1 \\ 0-\left(\frac{-1*0}{1}\right)=0 \\ 0-\left(\frac{-1*1}{1}\right)=1 \\ 0-\left(\frac{-1*0}{1}\right)=0 \\ -2-\left(\frac{-1*2}{1}\right)=0 \\ 0-\left(\frac{-1*0}{1}\right)=0 \\ 0-\left(\frac{-1*0}{1}\right)=0 \\ 0-\left(\frac{-1*0}{1}\right)=0 \\ 1-\left(\frac{-1*-1}{1}\right)=0 \\ -1-\left(\frac{-1*1}{1}\right)=0 \\ 10-\left(\frac{-1*20}{1}\right)=30 \end{cases}$$

Solutions in table 19 are $d_4{}^{+}, x_2, \mathrm{d}_3{}^{-}, x_1 = 20, 15{,}580{,}30$ respectively.

Table 20:The primal $5^{th}$ iteration of example 5.2.1

| **$c_j$** | | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **$c_B$** | $y_B$ | $x_1$ | $x_2$ | $d_1{}^{-}$ | $d_1{}^{+}$ | $d_2{}^{-}$ | $d_2{}^{+}$ | $d_3{}^{-}$ | $d_3{}^{+}$ | $d_4{}^{-}$ | $d_4{}^{+}$ | RHS |
| 0 | $d_4{}^{+}$ | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | $-1$ | 1 | 20 |
| 0 | $x_2$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 15 |
| 1 | $\mathrm{d}_3{}^{-}$ | 0 | 0 | $-8$ | 0 | $-12$ | 0 | 1 | $-1$ | 0 | 0 | 580 |
| 0 | $x_1$ | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 30 |
| | $z_j$ | 0 | 0 | $-9$ | 0 | $-14$ | 0 | 1 | $-1$ | 0 | 0 | 580 |
| | $z_j - c_j$ | 0 | 0 | $-9$ | 0 | $-14$ | 0 | 0 | $-1$ | 0 | 0 | |

Actually the optimization of goal three is not necessary in this problem, because the optimum solution to problem Goal two already yields $d_3{}^{-} = 580$; that is, it is already optimum for goal three. Such computational- saving opportunities should be taken advantage of whenever they arise during the course of implementing the Lexicographic method. All $z_j - c_j \le 0$.

Optimal solutions are $x_1 = 30, x_2 = 15, d_3{}^{-} = 580, d_4{}^{+} = 20$

**Example 5.2.1.(Ignizio,1985)**

$$minz = \{(d_1{}^+ + d_2{}^+), (2d_3{}^- + 3d_4{}^-)\}$$

Subject to $\mathrm{x_1 + x_2 + d_1{}^- - d_1{}^+ = 12}$

$$2x_1 + x_2 + d_2{}^- - d_2{}^+ = 20$$

$$16x_1 + 10x_2 + d_3{}^- - d_3{}^+ = 160$$

$$3x_1 + 5x_2 + d_4{}^- - d_4{}^+ = 60$$

$$x, d_i{}^+, d_i{}^- \geq 0 \; i = 1{,}2{,}3{,}4$$

Solution of goal one (first goal)

$$minz = \{(d_1{}^+ + d_2{}^+)\}$$

Subject to $\mathrm{x_1 + x_2 + d_1{}^- - d_1{}^+ = 12}$

$$2x_1 + x_2 + d_2{}^- - d_2{}^+ = 20$$

$$16x_1 + 10x_2 + d_3{}^- - d_3{}^+ = 160$$

$$3x_1 + 5x_2 + d_4{}^- - d_4{}^+ = 60$$

$$x, d_i{}^+, d_i{}^- \geq 0 \; i = 1{,}2{,}3{,}4$$

from the soft ware

$d_1{}^+, d_2{}^+, x_1, x_2$ all are zero

And $d_1{}^-, d_2{}^-, d_3{}^-, d_4{}^-, = 12{,}20{,}160{,}60$ are optimal solutions respectively.

Optimal values =0

$$minz = \{(2d_3{}^- + 3d_4{}^-)\}$$

Subject to $\mathrm{x_1 + x_2 + d_1{}^- - d_1{}^+ = 12}$

$$2x_1 + x_2 + d_2{}^- - d_2{}^+ = 20$$

$$16x_1 + 10x_2 + d_3{}^- - d_3{}^+ = 160$$

$$3x_1 + 5x_2 + d_4{}^- - d_4{}^+ = 60$$

$$x, d_i{}^+, d_i{}^- \geq 0 \; i = 1{,}2{,}3{,}4$$

Solution of goal two (second goal)

from the soft ware

$d_2{}^-, d_4{}^-, x_1, x_2, = 1.333{,}13.33{,}6.667{,}5.33$ are optimal solutions respectively

Optimal value =40

Both lexicographic and dual simplex methods are solution method of linear goal programming problem. The foremost value of goal programming is in its facility for solving problems by using the given method with hierarchically arranged conflicting goals. One of the major advantages of goal programming is that due to inclusion of the deviational variables there exists a solution to the problem provided that the target values are known, and the goals are in feasible region. To solve linear goal programming problems by using dual simplex method is not only find solutions because dual simplex method use additional simplex algorism .dual simplex method has more complex processes or enervating than lexicographic method .then lexicographic method is the best way of solving linear goal programming than Dual simplex method.

# 6. SUMMARY, CONCLUSION AND RECOMMENDATION

## 6.1. Summary

The two-solution method is suitable for solving linear goal programming problem. In order to solving linear Goal programming problem Minimizes the deviations among the desired levels of goals (targets) and the actual achievements which is accomplished by converting inequalities into equalities by including positive and negative deviation variables that permit under- achievement or overachievement of each goal. The aim of the goal programming is achieving as much goals as possible by minimizing their deviations from their targets. That is the value of the achievement function becomes a single-valued function.

## 6.2. Conclusion

In this project we have discussed lexicographic and dual simplex method to solve linear goal programming problem. The methods are examined on the same examples which were solved by these two methods and we presented the study of lexicographic and dual simplex method for linear goal programming problem. We mean that to compare Dual simplex method and Lexicographic method based on spadework. The illustrative examples also show that it requires more computational time. The solution shows that all the goals have been achieved at the given level. In this study we reformulate the goal programming problem into a multi-objective optimization problem and suggest using the LINGO soft ware to find optimal solutions. But dual simplex method is more complex based on spadework.

## 6.3. Recommendation

Using the Lexicographic and Dual simplex methods we solve linear goal programming problems with the same examples. The following basic recommendations are suggested: LINGO Software is easy, convenient and practical by using of computers.

# 7. REFERENCES

Antonio, L.J., Martınez, S.Z. and Coello ,C.A. 2009. An Introduction to Multi-objective Optimization Techniques. Nova Science Publishers, Inc., Hauppauge,:1-26.

Arthur, J.L. and Ravindran, A. 1978. Note-An Efficient Goal Programming Algorithm Using Constraint Partitioning and Variable Elimination. *Management Science,24(8):867-868.*

Calvete, H.I. and Mateo, P.M. 1998. Lexicographic optimization in generalized network flow problems. *Journal of the Operational Research Society, 49(5):519-529.*

Charnes,A and Cooper, W.W. 1961. *Management Models and Industrial Applications of linear programming*. J. Wiley.

Charnes, A and Cooper ,W.W.1977. Goal programming and multiple objective optimizations. *European Journal of Operational Research 1: 307-322.*

Chowdary, B.V. and Slomp, J.2002. Production planning under dynamic product environment: a multi-objective goal programming approach. University of Groningen.

Cohon, J.L. 2004,Multi objective Programming and Planning, Dover, Mineola, NY.

Evans, G.W.1984. An overview of techniques for solving multi-objective mathematical programs. *Management Science*,30(11):1268-1282.

Hillier, F. and Liebermann,G.2001. Introduction to Operations Research. Seventh Edition; McGraw- Hill, New York, 7:331 – 340.

Ignizio,1985.An algorithm for the linear goal program dual. *Journal of the Operational Research Society 36: 507-515.*

Ignizio, J. P.1978. A Review of Goal Programming: A Tool for Multi-objective Analysis. *Journal of the Operational Research Society. 29(11): 1109-1119.*

Ignizio, J.P. 1976. Goal programming and extensions. Lexington Books.

Ignizio, J.P. 1978. A review of goal programming: A tool for Multi-objective analysis. *Journal of the Operational Research Society, 29(11):1109-1119.*

Ignizio, J.P. 1985. An algorithm for solving the linear goal-programming problem by solving its dual. *Journal of the Operational Research Society, 36(6):507-515.*

Ignizio, J.P.1982. Linear programming in single and multiple-objective systems.

Ignizio,J.P.1974.The development of the multidimensional dual in linear goal programming. Working Paper, The Pennsylvania State University

Ignizio,J.P.1983. Linear Programming in Single and Multiple Objective Systems. Prentice-Hall, Englewood Cliffs, NJ.

Jablonsky, J.2012. Multicriteria approaches for ranking of efficient units in DEA models. *Central European Journal of Operations Research,20(3):435-449.*

Kasana, H.S. and Kumar, K.D.2003. Grouping Algorithm for linear goal programming problems.*Asia-pacific journal of operational research,20(2):191.*

Kwak, N.K. and Schniederjans, M.J.1982. An alternative method for solving goal-programming problems: a reply. *Journal of the Operational Research Society,33(9):859-860*.

Lee, S.M.1972.Goal programming for decision analysis.Philadelphia.Auerbach Publishers:252-260.

Martel, J.M. and Aouni, B. 1990. Incorporating the decision-maker's preferences in the goal-programming model. *Journal of the Operational Research Society,41(12):1121-1132.*

Min, H. and Storbeck, J.1991. On the origin and persistence of misconceptions in goal programming. *Journal of the Operational Research Society*,42(4):301-312.

Morris,W. T. 1964.The Analysis of Management Decisions. Homewood, IL: Richard D. Irwin.

Murtagh, B. A.1981. Advanced Linear Programming. New York: McGraw-Hill.

Orumie, U.C. and Ebong, D.2014. A glorious literature on linear goal programming algorithms. *American Journal of Operations Research,4(02):59.*

Orumie, U.C and Ebong, D.W.2014. An Efficient Method of Solving Lexicographic Linear Goal Programming Problem. *International Journal of Natural Sciences Research , 2014 (34): 2224-3186.*

Rifai, A.K.1994. A Note on the Structure of the Goal-Programming Model: Assessment and Evaluation. *International Journal of Operations and Production Management,* 16: 40-49.

Romero, C. and Rehman, T.2003.Multiple criteria analysis for agricultural decisions .Elsevier.

Romero, C.1985. A survey of generalized goal programming. *European Journal of Operational Research, 25(2):183-191*.

Romero, C.1991. On misconceptions in goal programming. *Journal of the Operational Research Society*, *42*(10):927-928.

Romero, C.2014. Handbook of critical issues in goal programming. Elsevier.

Sen, N. and Nandi, M.2012. Goal programming, its application in management sectors–special attention into plantation management: a review. *International Journal of Scientific and Research Publications,2(9):1-6.*

Sunar, M. and Kahraman, R.2001.A comparative study of Multi-objective optimization methods in structural design.*Turkish Journal of Engineering and Environmental Sciences,25(2):69-78.*

Taha, H.A.2003. Operation Research: An Introduction, seventh edition, prentice-Hill delhi, India. 347 – 360.

Tamiz, M., Jones, D.F. and C. Romero, C.1998.Goal programming for decision making: an overview of the current state-of-the-art, *European Journal of Operational Research, 569−58.*

Tamiz, M., Jones, D.F. and El-Darzi, E.1995.A review of goal programming and its applications. Annals of operations Research,58(1):39-5353.

Zanakis, S. H. and Gupta,S.K.1985. A categorized bibliographic survey of goal programming. Omega 13(3): 211-222.

# 8. APPENDIX

Solving Linear Goal Programming Problem By Using Lingo Soft Ware

```
Min=dplus1+dplus2;
x1+dminus1-dplus1=30;
x2+dminus2-dplus2=15;
8*X1+12*x2+dminus3-dplus3=1000;
X1+2*x2+dminus4-dplus4=40;
end
```

```
Global optimal solution found.
  Objective value:                              0.000000
  Infeasibilities:                              0.000000
  Total solver iterations:                             0
  Elapsed runtime seconds:                          0.06

  Model Class:                                        LP

  Total variables:                     10
  Nonlinear variables:                  0
  Integer variables:                    0

  Total constraints:                    5
  Nonlinear constraints:                0

  Total non-zeros:                     16
  Nonlinear non-zeros:                  0


                                Variable           Value        Reduced Cost
                                  DPLUS1        0.000000            1.000000
                                  DPLUS2        0.000000            1.000000
                                      X1        0.000000            0.000000
                                 DMINUS1        30.00000            0.000000
                                      X2        0.000000            0.000000
                                 DMINUS2        15.00000            0.000000
                                 DMINUS3        1000.000            0.000000
                                  DPLUS3        0.000000            0.000000
                                 DMINUS4        40.00000            0.000000
                                  DPLUS4        0.000000            0.000000

                                     Row    Slack or Surplus      Dual Price
                                       1        0.000000           -1.000000
                                       2        0.000000            0.000000
                                       3        0.000000            0.000000
                                       4        0.000000            0.000000
                                       5        0.000000            0.000000
```

```
Min=dminus3;
x1+dminus1-dplus1=30;
x2+dminus2-dplus2=15;
8*X1+12*x2+dminus3-dplus3=1000;
X1+2*x2+dminus4-dplus4=40;
Dplus1=0;
Dplus2=0;
End
```

```
Global optimal solution found.
  Objective value:                              580.0000
  Infeasibilities:                              0.000000
  Total solver iterations:                             0
  Elapsed runtime seconds:                          0.14

  Model Class:                                        LP

  Total variables:                      8
  Nonlinear variables:                  0
  Integer variables:                    0

  Total constraints:                    5
  Nonlinear constraints:                0

  Total nonzeros:                      13
  Nonlinear nonzeros:                   0


                                Variable           Value        Reduced Cost
                                 DMINUS3        580.0000            0.000000
                                      X1        30.00000            0.000000
                                 DMINUS1        0.000000            8.000000
                                  DPLUS1        0.000000            0.000000
                                      X2        15.00000            0.000000
                                 DMINUS2        0.000000            12.00000
                                  DPLUS2        0.000000            0.000000
                                  DPLUS3        0.000000            1.000000
                                 DMINUS4        0.000000            0.000000
                                  DPLUS4        20.00000            0.000000

                                     Row    Slack or Surplus      Dual Price
                                       1        580.0000           -1.000000
                                       2        0.000000            8.000000
                                       3        0.000000            12.00000
                                       4        0.000000           -1.000000
                                       5        0.000000            0.000000
                                       6        0.000000            8.000000
                                       7        0.000000            12.00000
```

```
Min=dplus4;
x1+dminus1-dplus1=30;
x2+dminus2-dplus2=15;
8*X1+12*x2+dminus3-dplus3=1000;
X1+2*x2+dminus4-dplus4=40;
Dplus1=0;
Dplus2=0;
dminus3=580;
end
```

```
Global optimal solution found.
  Objective value:                              20.00000
  Infeasibilities:                              0.000000
  Total solver iterations:                             0
  Elapsed runtime seconds:                          0.15

  Model Class:                                        LP

  Total variables:                      7
  Nonlinear variables:                  0
  Integer variables:                    0

  Total constraints:                    5
  Nonlinear constraints:                0

  Total nonzeros:                      12
  Nonlinear nonzero:                   0


                       Variable           Value        Reduced Cost
                         DPLUS4        20.00000            0.000000
                             X1        30.00000            0.000000
                        DMINUS1        0.000000           0.3333333
                         DPLUS1        0.000000            0.000000
                             X2        15.00000            0.000000
                        DMINUS2        0.000000            0.000000
                         DPLUS2        0.000000            0.000000
                        DMINUS3        580.0000            0.000000
                         DPLUS3        0.000000           0.1666667
                        DMINUS4        0.000000            1.000000

                            Row    Slack or Surplus      Dual Price
                              1        20.00000           -1.000000
                              2        0.000000           0.3333333
                              3        0.000000            0.000000
                              4        0.000000          -0.1666667
                              5        0.000000            1.000000
                              6        0.000000           0.3333333
                              7        0.000000            0.000000
                              8        0.000000           0.1666667
```

```
max=-30*y1-15*y2-1000*y3-40*y4;

y1+8*y3+y4>=0;
y2+12*y3+2*y4>=0;
y1>=0;
y2>=0;
y3>=0;
y4>=0;
y1<=1;
y2<=1;
y3<=0;
y4<=0;
end
Global optimal solution found.
  Objective value:                              0.000000
  Infeasibilities:                              0.000000
  Total solver iterations:                             0
  Elapsed runtime seconds:                          0.20

  Model Class:                                        LP

  Total variables:                      4
  Nonlinear variables:                  0
  Integer variables:                    0

  Total constraints:                   11
  Nonlinear constraints:                0

  Total non zeros:                     18
  Nonlinear non zeros:                  0


                                Variable           Value        Reduced Cost
                                      Y1        0.000000            30.00000
                                      Y2        0.000000            15.00000
                                      Y3        0.000000            1000.000
                                      Y4        0.000000            40.00000

                                     Row    Slack or Surplus      Dual Price
                                       1        0.000000            1.000000
                                       2        0.000000            0.000000
                                       3        0.000000            0.000000
                                       4        0.000000            0.000000
                                       5        0.000000            0.000000
                                       6        0.000000            0.000000
                                       7        0.000000            0.000000
                                       8        1.000000            0.000000
                                       9        1.000000            0.000000
                                      10        0.000000            0.000000
                                      11        0.000000            0.000000
```

```
MAX = -30 * y1 -15*y2 -10000 * y3 -40*y4+1000;
y1 +8 * y3 +y4 >= 8;
y2 +12 * y3 +2*y4 >= 12;
y1 >= 0;
y2 >= 0;
y3 >= 0;
y4 >= 0;
y3 <= 1;
y4 <= 0;
END
```

```
 Global optimal solution found.
  Objective value:                              580.0000
  Infeasibilities:                              0.000000
  Total solver iterations:                             0
  Elapsed runtime seconds:                          0.04

  Model Class:                                        LP

  Total variables:                      4
  Nonlinear variables:                  0
  Integer variables:                    0

  Total constraints:                    9
  Nonlinear constraints:                0

  Total non zeros:                     16
  Nonlinear nonzeros:                   0
```

| Variable | Value | Reduced Cost |
|---|---|---|
| Y1 | 8.000000 | 0.000000 |
| Y2 | 12.00000 | 0.000000 |
| Y3 | 0.000000 | 9580.000 |
| Y4 | 0.000000 | 0.000000 |

| Row | Slack or Surplus | Dual Price |
|---|---|---|
| 1 | 580.0000 | 1.000000 |
| 2 | 0.000000 | -30.00000 |
| 3 | 0.000000 | -15.00000 |
| 4 | 8.000000 | 0.000000 |
| 5 | 12.00000 | 0.000000 |
| 6 | 0.000000 | 0.000000 |
| 7 | 0.000000 | 0.000000 |
| 8 | 1.000000 | 0.000000 |
| 9 | 0.000000 | 20.00000 |

```
MAX = -30 * y1 -15*y2 -10000 * y3 -40*y4;
y1 +8 * y3 +y4 >= 0;
y2 +12 * y3 +2*y4 >= 0;
y3 >= 0;
y4 >= 0;
y4 <= 0;
END
Global optimal solution found.
  Objective value:                              0.000000
  Infeasibilities:                              0.000000
  Total solver iterations:                             0
  Elapsed runtime seconds:                          0.58

  Model Class:                                        LP

  Total variables:                      4
  Nonlinear variables:                  0
  Integer variables:                    0

  Total constraints:                    6
  Nonlinear constraints:                0

  Total non zeros:                     13
  Nonlinear non zeros:                  0


                                Variable           Value        Reduced Cost
                                      Y1        0.000000            30.00000
                                      Y2        0.000000            15.00000
                                      Y3        0.000000            10000.00
                                      Y4        0.000000            40.00000

                                     Row    Slack or Surplus      Dual Price
                                       1        0.000000            1.000000
                                       2        0.000000            0.000000
                                       3        0.000000            0.000000
                                       4        0.000000            0.000000
                                       5        0.000000            0.000000
                                       6        0.000000            0.000000
```

**Example 2**

```
Min=dplus1+dplus2;
x1+x2+dminus1-dplus1=12;
2*x1+x2+dminus2-dplus2=20;
16*X1+10*x2+dminus3-dplus3=160;
3*X1+5*x2+dminus4-dplus4=60;
end
```

```
Global optimal solution found.
  Objective value:                              0.000000
  Infeasibilities:                              0.000000
  Total solver iterations:                             0
  Elapsed runtime seconds:                          0.56

  Model Class:                                        LP

  Total variables:                     10
  Nonlinear variables:                  0
  Integer variables:                    0

  Total constraints:                    5
  Nonlinear constraints:                0

  Total non zeros:                     18
  Nonlinear non zeros:                  0


                                Variable           Value        Reduced Cost
                                  DPLUS1        0.000000            1.000000
                                  DPLUS2        0.000000            1.000000
                                      X1        0.000000            0.000000
                                      X2        0.000000            0.000000
                                 DMINUS1        12.00000            0.000000
                                 DMINUS2        20.00000            0.000000
                                 DMINUS3        160.0000            0.000000
                                  DPLUS3        0.000000            0.000000
                                 DMINUS4        60.00000            0.000000
                                  DPLUS4        0.000000            0.000000

                                     Row    Slack or Surplus      Dual Price
                                       1        0.000000           -1.000000
                                       2        0.000000            0.000000
                                       3        0.000000            0.000000
                                       4        0.000000            0.000000
                                       5        0.000000            0.000000
```

```
Min=2*dminus3+3*dminus4;
x1+x2+dminus1-dplus1=12;
2*x1+x2+dminus2-dplus2=20;
16*X1+10*x2+dminus3-dplus3=160;
3*X1+5*x2+dminus4-dplus4=60;
Dplus1=0;
Dplus2=0;
end
```

```
 Global optimal solution found.
  Objective value:                              40.00000
  Infeasibilities:                              0.000000
  Total solver iterations:                             5
  Elapsed runtime seconds:                          0.12

  Model Class:                                        LP

  Total variables:                      8
  Nonlinear variables:                  0
  Integer variables:                    0

  Total constraints:                    5
  Nonlinear constraints:                0

  Total non zeros:                     16
  Nonlinear non zeros:                  0
```

| Variable | Value | Reduced Cost |
|---|---|---|
| DMINUS3 | 0.000000 | 1.000000 |
| DMINUS4 | 13.33333 | 0.000000 |
| X1 | 6.666667 | 0.000000 |
| X2 | 5.333333 | 0.000000 |
| DMINUS1 | 0.000000 | 25.00000 |
| DPLUS1 | 0.000000 | 0.000000 |
| DMINUS2 | 1.333333 | 0.000000 |
| DPLUS2 | 0.000000 | 0.000000 |
| DPLUS3 | 0.000000 | 1.000000 |
| DPLUS4 | 0.000000 | 3.000000 |

| Row | Slack or Surplus | Dual Price |
|---|---|---|
| 1 | 40.00000 | -1.000000 |
| 2 | 0.000000 | 25.00000 |
| 3 | 0.000000 | 0.000000 |
| 4 | 0.000000 | -1.000000 |
| 5 | 0.000000 | -3.000000 |
| 6 | 0.000000 | 25.00000 |
| 7 | 0.000000 | 0.000000 |

```
max=-12*y1-20*y2-160*y3-60*y4;
y1+2*y2+16*y3+3*y4>=0;
y1+y2+10*y3+5*y4>=0;
y1>=0;
y2>=0;
y3>=0;
y4>=0;
y1<=1;
y2<=1;
y3<=0;
y4<=0;
end
Global optimal solution found.
  Objective value:                              0.000000
  Infeasibilities:                              0.000000
  Total solver iterations:                             0
  Elapsed runtime seconds:                          0.16

  Model Class:                                        LP

  Total variables:                      4
  Nonlinear variables:                  0
  Integer variables:                    0

  Total constraints:                   11
  Nonlinear constraints:                0

  Total nonzeros:                      20
  Nonlinear nonzeros:                   0


                                Variable           Value        Reduced Cost
                                      Y1        0.000000            12.00000
                                      Y2        0.000000            20.00000
                                      Y3        0.000000            160.0000
                                      Y4        0.000000            60.00000

                                     Row    Slack or Surplus      Dual Price
                                       1        0.000000            1.000000
                                       2        0.000000            0.000000
                                       3        0.000000            0.000000
                                       4        0.000000            0.000000
                                       5        0.000000            0.000000
                                       6        0.000000            0.000000
                                       7        0.000000            0.000000
                                       8        1.000000            0.000000
                                       9        1.000000            0.000000
                                      10        0.000000            0.000000
                                      11        0.000000            0.000000
```

```
max=-12*y1-20*y2-160*y3-60*y4+500;
y1+2*y2+16*y3+3*y4>=41;
y1+y2+10*y3+5*y4>=35;
y1>=0;
y2>=0;
y3>=0;
y4>=0;
y3<=2;
y4<=3;
end
Global optimal solution found.
  Objective value:                              40.00000
  Infeasibilities:                              0.000000
  Total solver iterations:                             3
  Elapsed runtime seconds:                          0.06

  Model Class:                                        LP

  Total variables:                      4
  Nonlinear variables:                  0
  Integer variables:                    0

  Total constraints:                    9
  Nonlinear constraints:                0

  Total nonzeros:                      18
  Nonlinear nonzeros:                   0


                                Variable           Value        Reduced Cost
                                      Y1        25.00000            0.000000
                                      Y2        0.000000            1.333333
                                      Y3        1.000000            0.000000
                                      Y4        0.000000            13.33333

                                     Row    Slack or Surplus      Dual Price
                                       1        40.00000            1.000000
                                       2        0.000000           -6.666667
                                       3        0.000000           -5.333333
                                       4        25.00000            0.000000
                                       5        0.000000            0.000000
                                       6        1.000000            0.000000
                                       7        0.000000            0.000000
                                       8        1.000000            0.000000
                                       9        3.000000            0.000000
```