| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
REPORT NUMBER 88-2155
TITLE LINEAR GOAL PROGRAMMING AS A MILITARY DECISION AID

AUTHOR(S) MAJOR JAMES F. POWELL, USAF

FACULTY ADVISOR MAJOR JACK B. ROBBINS, ACSC/EDJ

SPONSOR COLONEL HALBERT R. SMART
OJCS/J-3
CHIEF, COMBAT OPERATIONS SUPPORT DIVISION

Submitted to the faculty in partial fulfillment of requirements for graduation.

AIR COMMAND AND STAFF COLLEGE
AIR UNIVERSITY
MAXWELL AFB, AL 36112-5542
This report explores the use of linear goal programming as a military decision aid. Linear programming, as well as linear goal programming, is explained with a basic graphic example and definitions. The complicated weapon allocation problem facing the combined arms commanders in today's battlefield is defined in linear programming terms, developed into computerized linear programming format, and solved on an Apple IIe home computer. This solution provides an optimum weapon selection for the given situation. Additional capabilities, applications, and future potential are also discussed. Copyrighted software and printout are included.
This report explains how linear programming operates, highlights the differences between linear programming and linear goal programming, and develops a data set to apply linear goal programming as a decision aid for a combined arms commander in a tactical situation. These data and software are demonstrated in computing an optimum solution for a weapon selection problem. Additional information is presented on more powerful capabilities not demonstrated as well as additional proposed military applications.

Previous work on this subject was recently accomplished by the author in the University of Nebraska MBA program under the tutelage of Professor Marc Schnelderjans. Special thanks is due Dr. Schnelderjans for his patient instruction and generous copyright release.
About the Author

Major James F. Powell is a Electronic Warfare Officer with a broad range of Air Force experience. His initial assignment, after navigator and electronic warfare officer training, was as a combat certified electronic warfare officer aboard B-52's. After four years in this aircraft Major Powell cross trained into RC-135's where he flew the Cobra Ball aircraft. In this assignment he was selected as a Reconnaissance Crew Commander directing the entire mission aboard this aircraft and was additionally selected as the Standardization/Evaluation Elint Branch Chief. In this capacity he was responsible for all tasking guidance for this national platform. Following this assignment Major Powell was selected to serve on the Operations staff at HQ SAC, Offutt AFB NE. In this capacity he initially served as the B-1B Electronic Combat Support Branch Chief. Here he was responsible for all electronic combat support for the B-1B and spearheaded the development of several computer programs to provide this support. Subsequently, Major Powell was selected as the Special Technical Operations Division Chief. In this regard he served as the focal point for all HQ SAC Special Technical Operations initiatives. Major Powell is a 1973 graduate of the University of Houston with a BS degree in Electrical and Electronics Technology. He earned his commission in 1974 through OTS and subsequently earned an MBA from the University of Nebraska in 1987.
# TABLE OF CONTENTS

Preface ................................................................. 111  
List of Illustrations .................................................. vi  
Executive Summary .................................................. vii  

CHAPTER ONE - The Combined Arms Weapon Selection Problem 

CHAPTER TWO - What Linear Programming is and How it Works 
Linear Programming ................................................. 3  
A Graphic Example ................................................... 5  

CHAPTER THREE - Problem Statement in Linear Programming terms 
Definition Parameters .............................................. 11  
Decision Variables Defined ......................................... 11  
Constraint Definition .............................................. 12  

CHAPTER FOUR - Development, Manipulation, and Processing 
Data Development ................................................... 14  
Data Manipulation ................................................... 17  
Computer Data Processing ......................................... 19  

CHAPTER FIVE - Limits and Additional Capabilities 
Software Limits ..................................................... 22  
Additional Capabilities ............................................ 22  

CHAPTER SIX - Observations and Recommendation 
Observations .......................................................... 24  
Recommendation ...................................................... 24  

BIBLIOGRAPHY .......................................................... 26  

APPENDICES - 
Appendix A - Computer Software ................................. 29  
Appendix B - Computer Software ................................ 38  
Appendix C - Computer Printout ................................. 54  


LIST OF ILLUSTRATIONS

Figure 1: Fuel Constraint .................................................. 6
Figure 2: Maintenance Hour Constraint ............................. 7
Figure 3: Area of Feasible Solutions ................................. 8
Figure 4: Optimum Solution ............................................. 9
EXECUTIVE SUMMARY

Part of our College mission is distribution of the students' problem solving products to DOD sponsors and other interested agencies to enhance insight into contemporary, defense related issues. While the College has accepted this product as meeting academic requirements for graduation, the views and opinions expressed or implied are solely those of the author and should not be construed as carrying official sanction.

REPORT NUMBER 88-2155
AUTHOR(S) MAJOR JAMES F. POWELL
TITLE LINEAR GOAL PROGRAMMING AS A MILITARY DECISION AID

I. Purpose: To demonstrate the validity and utility of linear goal programming as a military decision aid tool.

II. Problem: The complexity of the modern battlefield, coupled with the speed, lethality, and vast range of weapon systems has surpassed the ability of the current joint planning staffs to insure optimum allocation of all weapon systems. Due to this complexity, the combined arms commander currently has no responsive method of ensuring optimum selection of weapons for application against various targets.

III. Method: Initially the report explains the basic operation of linear programming and how an optimum solution is derived. Additionally, representative, notional data sets for various weapon systems available to a combined arms commander are developed. Computerized linear goal programming processing of these data is demonstrated to provide an optimum weapon system selection.
IV. Conclusion: The type information required for linear programming to be applied is either available or can be derived. Computerized linear goal programming, operating on these data, offers a high degree of utility in the demonstrated combined arms case as well as other military applications.

V. Recommendation: HQ USAF conduct a detailed study of the utility of linear goal programming as a decision aid system.
Chapter One

THE COMBINED ARMS WEAPON SELECTION PROBLEM

The profession of military command has changed immensely throughout the years. This change is nowhere more apparent than in the weapons employed and the proliferation of targets in the conduct of warfare. In recent history the weapons available to the military commanders and the targets selected were relatively simple. The weapons consisted of hand held lances, bow and arrows, up to light artillery, while the targets consisted primarily of concentrations of enemy troops. The commander employing these weapons usually had firsthand knowledge of the application techniques and capabilities of the weapons he directed. Modern warfare, and its vast array of weapons and targets has changed this basic tenet of warfare.

Warfare has expanded to include weapons, even entire theaters of conflict, not dreamed of in most of recorded history. This, combined with the geographical expansion of theaters of warfare, the relative speed of war fighting, the range of weapons, and the proliferation of targets of all types, has exponentially increased the complexity of modern warfare. The historically recent additions of submarine warfare, air warfare, advanced armour warfare, the potential for space warfare, and many other advances have expanded the breadth of weapons employed to the point that no combined arms commander can be expected to have detailed knowledge of all the resources available to wage war.

Recognizing this eventuality, modern armies have adopted and expanded to include a staff organization to provide this basic knowledge of resources available so that these diverse systems might be properly employed in battle. These staff organizations often take the form of sizeable numbers of staff officers deployed with the headquarters elements to provide planning expertise for the weapons systems employed. This approach has worked well in ensuring that most weapons are employed properly but it has not allowed an overall capability to efficiently allocate all available weapon systems in relation to their individual strengths and weaknesses. In essence, there is no "big picture" plan of weapon allocation other than the mental evaluation of the commanders. The weapons involved have become so numerous and diverse in characteristics that this is rapidly becoming an impossible task.

Throughout history, these type calculations have made the difference between victory and defeat. Drawing from Sun Tzu, "Military tactics are like unto water; for water in its
natural course runs away from high places and hastens downward. So in war, the way is to avoid what is strong and to strike at what is weak. Water shapes its course according to the nature of the ground over which it flows; the soldier works out his victory in relation to the foe he is facing." (2:29) Applied to today's environment, this axiom might can be construed to say the application of forces must be finely tuned and tailored according to the enemy faced. To accomplish this there must be a method to ensure the overall optimization of all forces employed in the combined arms theaters of today.

This paper demonstrates an optimization technique, linear goal programming, which can provide the overall optimization guideline required and perform as a valuable, dynamic, decision aid for the combined arms commander. The paper will present the basics of linear programming but will concentrate on the application of computer based linear goal programming techniques to a representative mix of weapon systems and targets to demonstrate the capabilities and adaptability of this approach.
Chapter Two

WHAT LINEAR PROGRAMMING IS AND HOW IT WORKS

LINEAR PROGRAMMING

This chapter will lay the groundwork for the entire following project. Since linear programming is possibly a new topic for many readers, the logical starting point is a common, layman's definition of linear programming. This step will ensure we all start with a common terminology and frame of mind. Building on this definition, we will then discuss the requirements for linear programming applications, the parts of a linear programming problem, how linear programming works, the limits of linear programming, and finally, how linear goal programming relates to linear programming.

Linear programming is best described as a mathematical technique used to find the one best, or optimum, solution for a given situation from a set of feasible solutions. Linear indicates that the relationships among the elements, or variables, can be expressed as proportional mathematical functions. Programming simply refers to the type model and its usage to "program" elements of the solution. Linear programming, as an optimization technique, began in 1947 with G. B. Dantzig's Interactive process. (3:4) This technique has almost constantly been refined and grew to include linear goal programming beginning with a text written by A. Charnes and W. W. Cooper, Management Models and Industrial Applications of Linear Programming in 1961. (3:5) Since this time linear programming and linear goal programming have continued to gain acceptance as valuable management tools, and have been applied to many diverse management systems.

Linear programming is applicable to a wide range of management problems, however, there are four basic conditions which must exist before it can be considered the appropriate quantitative technique. (1:192-193)

1. The decision maker is attempting to achieve a specific objective. (objectives in the case of linear goal programming)
2. Alternative solutions are available. (several answers might "fit", but only one is optimum)
3. Resources are scarce.
4. The objective (objectives in linear goal programming) and resource limitations can be expressed as linear mathematical equations or inequalities.
If these conditions exist linear programming should be considered as a valid optimization technique and thought should be given to transforming the problem to a linear programming format.

In order to express a problem in linear programming form you must be familiar with the parts and terminology of a linear programming problem. The terminology presented here is that usually applied to computer formatted problems, and will be used throughout this project as it is geared to a computer derived solution. The following example illustrates the components of a linear programming minimization problem.

\[ \text{Min: } Z = C_1 X_1 + C_2 X_2 + \ldots + C_n X_n \]
subject to:
\[
A_{11} X_1 + A_{12} X_2 + \ldots + A_{1m} X_m < b_1 \\
A_{21} X_1 + A_{22} X_2 + \ldots + A_{2m} X_m < b_2 \\
A_{m1} X_1 + A_{m2} X_2 + \ldots + A_{mn} X_m < b_m
\]
and: \( X_1, X_2, \ldots, X_m > 0 \)

\( X = \) decision variables (the number of goods to be produced or resources allocated for the given solution).
\( C = \) contribution coefficient (how much each good or resource contributes to the given solution).
\( Z = \) unknown solved for (in a minimization problem it is usually an expression of combined resources required to provide the optimum solution).
\( b = \) side constraints (usually mathematical expressions of resource limitations, however may represent practically any limiting factors capable of being expressed as linear mathematical functions).

Theoretically, any problem is capable of being solved utilizing linear programming techniques if it can be expressed in these terms. It is often beneficial to think in graphic terms to better understand the processes and manipulations employed in linear programming to reach an optimum solution. The graphic solution presented will show the relationships of the elements and the manipulation required to reach an optimum solution. In this case the manipulation is done graphically, however utilizing the format presented above it is possible to convert this expression to a form readily adaptable to computer processing. This is what will be done with the stated problem facing the combined arms military commander later in this paper.
A GRAPHIC EXAMPLE OF HOW LINEAR PROGRAMMING WORKS

Note: This example is an adaptation of a problem from (7).

Problem: Consider two aircraft, the X1 and the X2. The X2 is slightly larger than the X1 and therefore can carry 5 bombs compared to 4 for the X1 (bombs of equal size). The X1, because it is faster requires 4000 lbs. of fuel per mission as compared to 2000 lbs. for the X2. The X2, because it is older, requires 6 hours maintenance preparation per mission as compared to 3 hours for the X1. Our squadron has 32000 lbs. of fuel and 36 maintenance hours available and is tasked with delivering the maximum bomb load for tomorrow's mission. How many of each aircraft should be utilized?

Problem Restatement:
Bomb load: X1 = 4
X2 = 5

<table>
<thead>
<tr>
<th>Aircraft</th>
<th>Fuel Required</th>
<th>Maintenance hrs. available</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>4000</td>
<td>3</td>
</tr>
<tr>
<td>X2</td>
<td>2000</td>
<td>6</td>
</tr>
<tr>
<td>Total</td>
<td>32000</td>
<td>36</td>
</tr>
</tbody>
</table>

Linear programming Formulation:
Maximize Z (bomb load)
X1 = number of aircraft X1 to utilize decision variables
X2 = number of aircraft X2 to utilize
Maximize: \( Z = 4X1 + 5X2 \) objective function
subject to:
4000X1 + 2000X2 < 32000 (fuel avail.) constraints
3X1 + 6X2 < 36 (maintenance hrs)

The first step in the graphic solution is to solve for the overall limits for each constraint. This is done by setting each decision variable equal to zero, solving for the remaining decision variable, and then graphing the resulting line.

Assume: 4000X1 + 2000X1 = 32000
let: X1 = 0
then: 4000(0) + 2000X2 = 32000
2000X2 = 32000
X2 = 16

let: X2 = 0
then: 4000X1 + 2000(0) = 32000
4000X1 = 32000
X1 = 8
This process is then repeated for the remaining decision constraint.

Assume: 3X1 + 6X2 = 36
    let: X1 = 0
    then: 3(0) + 6X2 = 36
          6X2 = 36
          X2 = 6

    let: X2 = 0
    then: 3X1 + 6(0) = 36
          3X1 = 36
          X1 = 12
FIGURE 2

maintenance hour constraint
The next step is to graph the area of feasible solutions. This area is defined as that area to the left of both constraint lines when they are combined on a graph. Theoretically, any point within this area is a feasible solution.

The next step is to determine the slope of a feasible solution. In this step we pick a convenient number for $Z$, solve the objective function, and graph the result.

Let $Z = 20$

Then: $4X_1 + 5X_2 = 20$

Let: $X_1 = 0$

then: $4(0) + 5X_2 = 20$

then: $4X_1 + 5(0) = 20$

$5X_2 = 20$

$X_2 = 4$

$4X_1 = 20$

$X_1 = 5$
The final step is to draw a line parallel to this line at the last point of tangency to the area of feasible solutions (see figure 4, optimum line). Lines perpendicular to the axis from this point of tangency will intercept the axis at the optimum values for the decision variables.

\[ Z = 40 \]

\[ X_1 = 6 \frac{2}{3} \]

\[ X_2 = 2 \frac{2}{3} \]

In the real world we know we cannot fly \( \frac{2}{3} \) of an aircraft. Consider though if you were planning a mission of hundreds of aircraft. This solution is valid as long as the linear relationships are maintained, that is hours of maintenance, fuel, and bombs per aircraft remain the same. Then the solution may be expanded linearly or simply resolved. This example illustrates graphically the same process that occurs when we solve linear programming problems on a computer, the major difference being the computer does the "number crunching".

The same type solution will work for a minimization solution as well as a maximization solution. In the minimization case, the area of feasible solutions shifts to the right of the two constraint lines instead of to the left. Many computer programs are coded to handle either type solution. Computers additionally have the capabilities of handling large numbers of constraints as well as numerous
decision variables thusly expanding the solution well beyond the two dimensional graphic solution capability. These capabilities will be demonstrated in the computer solution of the stated problem from chapter one.

Before we start this process it is important to recognize the limits of linear programming. As discussed to this point, linear programming is capable of considering and solving for only one objective. Recognizing this shortcoming, Y. IJiri in 1965 published Management Goals and Accounting for Control which described the use of preemptive priority factors to allow the modeling of multiple conflicting objectives in accordance with their ranked importance in the objective function. Simply put, this new technique allowed for the simultaneous solution for multiple goals in priority order. This is the fundamental difference between linear programming and linear goal programming. Other differences include the capability to attach priority weights to specific constraints to dictate which are considered first. These differences set linear goal programming apart as a powerful, more manipulable form of linear programming. To take advantage of these advances we will utilize linear goal programming to solve the combined arms weapon allocation problem.
Chapter Three

PROBLEM STATEMENT IN LINEAR PROGRAMMING TERMS

DEFINITION PARAMETERS

Recall from chapter one, the problem to be addressed in this paper is the optimization of the various weapon systems available to the combined arms commander. This portion of the project will deal with the development and detail of the elements necessary to manipulate the stated problem with linear goal programming techniques.

To provide an orderly flow to this development process a representative group of weapon systems will be defined and representative constraints developed. Note at this point that the actual values employed in this example are notional values to serve for demonstration purposes only. The relative values employed are a result of the authors experience and on occasion are referenced to sources such as Fast Track. The use of these notional values versus precise, validated values is employed to preclude any classification issues. Additionally, the aim of this project is to demonstrate the utility of linear goal programming as a technique, therefore, precise values are not required.

The limitations of the micro software used in this demonstration must be considered prior to element construction. This software (appendix B) is constructed to operate on the Apple II plus, IIE, and IIC family of home computers and is limited due to the limited capabilities of these systems. These limits, thirty-five goal constraints, ten decision variables, and nine priorities must be kept in mind while defining the demonstration components. It would be impractical to structure a problem with more elements than the demonstration software is capable of handling, and would not contribute to the validity of the demonstration. Additional software and processing capabilities are available to handle these larger applications.

DECISION VARIABLES DEFINED

With these limits in mind, the definition of weapon systems available, or decision variables is the necessary first step in the development of demonstration exercise elements. In this example, weapon systems available will include the following systems.
X1 - tactical aircraft (conventional arms)
X2 - strategic aircraft (conventional arms)
X3 - tactical missiles (conventional arms)
X4 - strategic missiles (nuclear arms)
X5 - ground assault force (infantry and armor, conventional arms)
X6 - chemical munitions
X7 - tactical aircraft (smart conventional munitions)
X8 - unconventional warfare assault (special forces, etc.)
X9 - tactical missiles (nuclear arms)
X10 - strategic aircraft (nuclear arms)

Referencing the example presented in chapter two, these weapon systems will represent decision variables, or X values in the objective function. It is important to recall that the decision variables for an actual problem could be defined to any required degree of accuracy. As an example, they could be defined as different weapon loads on the same type aircraft. The wide range of decision variables presented here were selected to demonstrate the overall flexibility of the linear goal programming technique as applied to the combined arms problem.

CONSTRAINT DEFINITION

The second part of the construction process will focus on the definition of constraints. These constraints can be thought of as the overall problem set facing the combined arms commander. For example, he may desire to destroy a target but hesitates to use his most effective weapon system because of the overriding fear of escalation. In this case preventing escalation is his highest priority while target destruction assumes a lower priority. In this demonstration, escalation, target destruction, and other decision factors will be defined and modeled as constraints. The following factors will be modeled in this demonstration.

b1 - Timeliness: the relative time interval from execution to weapon arrival at target.

b2 - Probability of detection: the relative probability that the weapon system will be detected, identified, and countered prior to arrival at target.

b3 - Probability of target destruction: the relative probability of target destruction after weapon arrival.

b4 - Escalation factor: relative likelihood that use of this weapon system will lead to escalation of the present scenario.

b5 - Probability of personnel loss: relative probability that allied personnel will be lost during the application of the various weapon systems.
Weapon system availability: the constraint established by the gross number of each weapon system available.

Recalling again the example presented in chapter two, these decision factors will be modeled as constraints, or \( b \), in this example. These constraints, like the decision variables, could be made as finite as desired in an actual application. The macro software at appendix A is capable of handling up to one hundred and fifty constraints and ten priorities providing a greatly increased capability to tailor an overall optimization model. (3:201)
Chapter Four

DATA DEVELOPMENT, MANIPULATION, AND PROCESSING

DATA DEVELOPMENT

This section of the report highlights one of the major problems facing the Department of Defense today. This is the problem of obtaining and utilizing accurate data when modeling systems from more than one command or service. Many of the services operating the weapons systems utilized in this model have well developed models and data for many of the constraints modeled here, but are hesitant to release this data outside the command or service. This hesitancy stems from the fear that once released, the data will be manipulated and used against the service or command in the PPBS cycle. We must overcome this hesitancy to provide the accurate, timely data to support modeling systems which reach across several organizations within the Department of Defense.

This chapter centers on further development of the linear goal programming elements defined in chapter three and culminates in their entry into a linear goal programming computer program. In order to manipulate the decision variables and constraints defined in chapter three with linear goal programming computer software it is necessary to assign numerical values to them. This section of the report will focus on the further development of these elements to allow their manipulation with a linear goal programming computer program. (appendix B) Numerical values will be assigned to each constraint as they are associated with a particular weapon system or decision variable. These values are simply the relative values associated with each weapon system for each constraint. The constraint relative value information is presented utilizing the prefixes assigned to each element in chapter three.

Timeliness is simply the relative time from execution to weapon arrival on target. Timeliness is minimum for the missiles in the example (they take the least time to reach the target) and is maximum for the ground forces. The required times are rated, the minimum equaling one, and the maximum equaling ten, with the intermediate values linearly expressed as values between one and ten in this example.
b1 - Timeliness: maximum value 10 (least timely)  
minimum value 1 (most timely)

| X1 | 3 |
| X2 | 5 |
| X3 | 1 |
| X4 | 2 |
| X5 | 9 |
| X6 | 2 |
| X7 | 3 |
| X8 | 1 |
| X9 | 1 |

Probability of detection is the relative probability that a weapon system will be detected and countered prior to reaching the target. Detection models already exist for most of the systems in this model. In the case of the strategic systems, the ROPES (Route Penetration Evaluation System) operated at Headquarters Strategic Air Command (SAC) provides probability of detection based on the type threats expected to be encountered. This value is expressed as a percentage value representing the actual probability of detection. The actual numeric values and algorithms employed in this model are classified and the values are not released outside the headquarters. This case is representative of many of the other systems employed in this model. Probability of detection models exist, but the data is classified or is not approved for release outside the command.

The values used here are representative values based on Fast Sick, a Tactical Air Forces Employment Feasibility Exercise, values and various other unclassified publications. The values were derived by reviewing the threat level expected from our prime potential adversary (the Soviet Union), to be faced by each weapon system. Those systems whose detection is assured (ground assault forces) were given a maximum value of approximately ten (9). Those systems whose detection is very unlikely (tactical missiles) were given minimum values approaching one. The other systems with intermediate probabilities of detection are linearly represented on the scale between two and nine.

b2 - Probability of detection:
maximum value 10 (detection assured)  
minimum value 1 (detection unlikely)

| X1 | 3 |
| X2 | 5 |
| X3 | 2 |
| X4 | 6 |
| X5 | 9 |
| X6 | 5 |
| X7 | 3 |
| X8 | 2 |
| X9 | 2 |
| X10 | 4 |

Probability of target destruction is simply the probability that the weapon system will destroy the intended target after arrival. The probabilities expressed here are
derived from Fast Stick, Army FM 100-5, and common sense. Those systems with the highest probability of target destruction (nuclear weapons) are awarded the minimum value of one. Those systems less likely to destroy the target are awarded linearly higher values ranging from one to ten. Keep in mind that these values assume arrival at the target. Inability to reach the target is modeled in probability of detection.

b3 - Probability of target destruction:
- maximum value 10 (destruction questionable)
- minimum value 1 (destruction assured)

\[
\begin{align*}
X_1 &= 4 \\
X_2 &= 3 \\
X_3 &= 4 \\
X_4 &= 1 \\
X_5 &= 2 \\
X_6 &= 3 \\
X_7 &= 3 \\
X_8 &= 4 \\
X_9 &= 1 \\
X_{10} &= 1 \\
\end{align*}
\]

Probability of escalation is the most subjective constraint in this demonstration. As used here it is meant to be a military/political judgement of what the escalation potential of each weapon system is. In this example it is especially pertinent to the nuclear weapons.

Those systems with a high escalation potential (nuclear weapons) are given maximum values of nine while those systems with no escalation potential are given values of one. The intermediate systems are given linear values between one and nine representing their potential for escalating a given situation.

b4 - Probability of escalation:
- maximum value 10 (escalation likely)
- minimum value 1 (escalation unlikely)

\[
\begin{align*}
X_1 &= 1 \\
X_2 &= 3 \\
X_3 &= 2 \\
X_4 &= 9 \\
X_5 &= 1 \\
X_6 &= 6 \\
X_7 &= 2 \\
X_8 &= 2 \\
X_9 &= 8 \\
X_{10} &= 9 \\
\end{align*}
\]

Probability of personnel loss is a restated constraint containing two major factors, the probability of detection combined with the number of personnel exposed to hostile fire to execute each weapon system. Those systems with a low probability of personnel loss (strategic and tactical missiles) are given minimum values. Those systems employing large numbers of personnel and a high probability of detection (ground assault force) are given maximum values. The systems employing fewer people and having a lower probability of detection are given linear values between one and nine.
b5 - Probability of personnel loss:
  maximum value 10 (loss likely)
  minimum value 1 (loss unlikely)

\[
\begin{align*}
X1 &- 3 & X6 &- 1 \\
X2 &- 3 & X7 &- 3 \\
X3 &- 1 & X8 &- 8 \\
X4 &- 1 & X9 &- 1 \\
X5 &- 9 & X10 &- 3
\end{align*}
\]

Weapon system availability is the only constraint that is not presented in the form of relative values. In this instance, the constraint values are structured to represent the numbers of actual weapon systems available for each decision variable. These constraints are necessary to preclude the modeling of more resources than what might actually be available.

b6 - Weapon system availability

\[
\begin{align*}
X1 &- 100 & X6 &- 2 \\
X2 &- 2 & X7 &- 8 \\
X3 &- 5 & X8 &- 5 \\
X4 &- 10 & X9 &- 2 \\
X5 &- 5 & X10 &- 2
\end{align*}
\]

DATA MANIPULATION

Following the attachment of numerical values demonstrated above, it becomes necessary to arrange the constraints into a matrix format. This step is necessary to allow the developed constraints to later be translated directly to linear goal programming format. The constraint matrix for the constraints developed above follows. In order to limit clutter in the matrix, only the previously assigned prefixes are used to denote the constraints and decision variables.

<table>
<thead>
<tr>
<th>Constraint Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
</tr>
<tr>
<td>b1</td>
</tr>
<tr>
<td>X1</td>
</tr>
<tr>
<td>X2</td>
</tr>
<tr>
<td>X3</td>
</tr>
<tr>
<td>X4</td>
</tr>
<tr>
<td>X5</td>
</tr>
<tr>
<td>X6</td>
</tr>
<tr>
<td>X7</td>
</tr>
<tr>
<td>X8</td>
</tr>
<tr>
<td>X9</td>
</tr>
<tr>
<td>X10</td>
</tr>
</tbody>
</table>
The next sequential step in translating the already developed constraints and decision variable into a computer understandable form requires that they be expressed in linear goal programming format. This step is a direct extension from the constraint format detailed above. Simply stated, the functions are summed, by columns to indicate the total contribution of all decision variables for each constraint.

\[ 3X_1 + 5X_2 + X_3 + 2X_4 + 9X_5 + 2X_6 + 3X_7 + X_8 + X_9 + 5X_{10} = \text{Timeliness} \]

\[ 3X_1 + 4X_2 + 2X_3 + 6X_4 + 9X_5 + 5X_6 + 3X_7 + 2X_8 + 2X_9 + 4X_{10} = \text{Probability of detection} \]

\[ 4X_1 + 3X_2 + 4X_3 + X_4 + 2X_5 + 3X_6 + 3X_7 + 4X_8 + X_9 + X_{10} = \text{Probability of target destruction} \]

\[ X_1 + 3X_2 + 2X_3 + 9X_4 + X_5 + 6X_6 + 2X_7 + 2X_8 + 8X_9 + 9X_{10} = \text{Probability of escalation} \]

\[ 3X_1 + 3X_2 + X_3 + X_4 + 9X_5 + X_6 + 3X_7 + 8X_8 + X_9 + 3X_{10} = \text{Probability of personnel loss} \]

Constraint b6, actual number of resources available is treated differently. As previously mentioned, these numbers represent the actual numbers of resources available instead of relative contribution values. Since they are individual resource values, they are required to be expressed as individual constraints applicable only to the decision variable they limit.

\[ X_1 = 100 \text{ (conventionally armed tactical aircraft available)} \]
\[ X_2 = 2 \text{ (conventionally armed strategic aircraft available)} \]
\[ X_3 = 5 \text{ (conventional tactical missiles available)} \]
\[ X_4 = 10 \text{ (nuclear strategic missiles available)} \]
\[ X_5 = 5 \text{ (ground assault forces available)} \]
\[ X_6 = 2 \text{ (chemical munitions available)} \]
\[ X_7 = 8 \text{ (smart munitions armed tactical aircraft available)} \]
\[ X_8 = 5 \text{ (unconventional warfare assault forces available)} \]
\[ X_9 = 2 \text{ (nuclear armed tactical missiles available)} \]
\[ X_{10} = 2 \text{ (nuclear armed strategic aircraft available)} \]

The next step in the development process requires some prior discussion of the actual software to be employed as well as linear goal programming software in general. The method of coding for most linear goal programming software operates against the deviation factor for each constraint. For example, if a minimization problem were being run the system would function to minimize positive deviation thusly forcing the affected variables to a value less than the maximum limit. Additionally, if priorities are assigned,
the process is structured to compute each constraint in the order of the priority assigned to it, priority one being first and so on to the last priority. This allows the linear goal programming software to optimize a solution based on several constraints in priority order.

In order to allow for this capability it is necessary to prioritize the constraints previously developed. The problem, as stated, is developed as a linear goal programming minimization problem. As such, the software to be employed will operate to minimize the use of resources (defined as decision variables in the problem) while minimizing the collateral constraints in priority order. To provide this capability in the demonstration problem the following notional priorities will be assigned:
1. Minimize probability of personnel loss
2. Minimize probability of detection
3. Minimize probability of escalation.

These priorities are for demonstration only. The selection of real priorities is the prime area where the combined arms commander can tailor the linear goal programming technique to fit his existing tactical or strategic situation. The commander can select priorities to fit the existing situation and easily adapt these choices to the situation as it changes. This adaptability is one of the key benefits of linear goal programming. It allows it to be rapidly adjusted to the dynamic situation.

The final step in developing the stated problem in linear goal programming format is the writing of the objective statement. The objective statement is a summation of all the constraints, in priority order, into one minimization statement. It may be compared to the overall statement of objectives. In the case of the example, it will be a minimization statement (minimize scarce resources) and will define the minimization of each constraint in priority order by minimizing the positive deviation associated with each constraint. For the developed example the objective function is as follows:

\[ \text{Min: } Z = P_1 c_5 + P_2 d_2 + P_3 d_4 + d_1 + d_3 + d_5 + d_7 + d_8 + d_9 + d_{10} + d_{11} + d_{12} + d_{13} + d_{14} + d_{15} \]

**COMPUTER DATA PROCESSING**

The statement above is the final manipulation required for input into the linear goal programming software to be used for this demonstration. (Appendix B) It is beyond the scope of this paper to give a detailed explanation of the software and the operations it performs to obtain an optimum
solution. The basic functioning of the software performs an operation similar to the graphic solution presented in chapter two only with additional dimensions and allowing for priorities.

With this in mind, the problem, as developed and defined is entered into the linear goal programming software on an Apple IIIE home computer. Directions for this operation, as well as an example, can be found in Linear Goal Programming, page 122-125. (3:122-125)

The results of this processing demonstration are included as Appendix C. The resultant computer printout is in three essential parts. The first part (all that area above coefficients in tableau) is a display of the inputed weights and priorities. This area serves as a record to compare and check the output values printed below.

The second portion, coefficients in tableau, is a printout of the actual computer manipulation process. The computer utilizes a tableau solution process as an alternative to the graphic process to compute the optimum solution. Since the graphic process, illustrated earlier, is limited to simple solutions and graphic depictions it is not suited to computer manipulation. The tableau process employed consists of arranging the developed information into numerical tableau format and then manipulating these tableaus to provide an optimum solution. The tableau process is very much like the graphic process only it is done strictly with numerical manipulation to take advantage of the computer capabilities.

The third and last portion of the printout is the solution variable and goal display. This portion of the printout displays the values for the optimum solution with the given input values. In other words, with the previously developed values and stated priorities these values represent the optimum numbers of weapons systems to employ. The only relevant values are those with decision variable prefixes, X3 and X8, in the left column and positive values in the right column. The optimum solution for this problem is to employ 5 conventionally armed tactical missiles and 5 unconventional warfare assault forces. The second portion of this area displays any unachieved goals. In this case we had only three stated priority goals: minimize personnel loss, detection probability, and escalation potential. The displayed value for P4 is present because all constraints other than those stated above are entered as fourth priority goals and this is a composite value for them. Since they are not priority goals for the purposes of this demonstration this value is superfluous.
With this problem, once all the values have been inputed, the Apple computer required approximately four minutes to provide the attached solution. The four minutes time included approximately two and one half minutes actual computing time and approximately one and one half minutes printing time.
Chapter Five

SOFTWARE LIMITS AND ADDITIONAL CAPABILITIES

SOFTWARE LIMITS

As is evident by this point the demonstrated software is severely limited. The previously mentioned limits of thirty-five goal constraints, ten decision variables, and nine priorities limit this software to only the simplest goal programming applications (3:201). This limit is evident in the rough groupings of constraints required in the demonstration problem. With a more powerful program it would be possible to greatly refine the constraints to provide a much more realistic representation. With more powerful software, constraints like probability of target destruction, might be broken down to provide probabilities for several different types of targets and the decision variables, like tactical aircraft, might be expanded to provide variables for each distinct model of tactical aircraft available. These expansions would provide a more realistic, refined and usable product.

The demonstration software is additionally limited in the method employed to print out the optimum solution. Without a significant level of prior knowledge of the software it is difficult to determine what the separate areas and lines of the printout represent. With a more powerful, better refined software package it would be possible to provide a printout easily readable without any prior knowledge.

ADDITIONAL CAPABILITIES

The shortcoming of this software are solved on linear goal programming software hosted on larger computers. The macro computer software included as attachment A has a much greater manipulation capability than that demonstrated. This software allows for 150 goal constraints, 150 decision variables, and 10 preemptive priority levels (3:114). These expanded capabilities would allow many of the expansions to constraints and variables mentioned above to provide a more refined product. This macro software provides a tenfold increase in variable definition capability and a three fold increase in constraint definition capability when compared to the demonstrated micro level software.

In addition to the expanded definition capability, the macro level software provides a much more understandable
printout. It provides the values of the decision variables as well as an analysis of the deviations and priority accomplishments. These analyses allow a much better presentation of the optimum solution values and the processing required to compute them.
Chapter Six

OBSERVATIONS AND RECOMMENDATIONS

OBSERVATIONS

This paper has demonstrated linear goal programming as a military decision aid system, however, linear programming has many other diverse uses, many applicable to the military. As a normative model linear goal programming may be applied to problems to minimize transportation costs, assign personnel to projects, and any type of problem dealing with the allocation of scarce resources (4:633). These type applications can provide utility to military maintenance organizations, supply organizations, or any other type military organization dealing with the allocation of any resource, including our most valuable resource, people. These applications, with the guaranteed optimum solution could save the military significant money as well as manpower hours in operating expenses.

The civilian business establishment has already recognized the power and potential of linear programming. In a recent survey (1984) almost three of every five respondent firms reported using linear programming in the production management area (1:193). Although not strictly in the production business, linear programming can be applied to many of the similar resource applications problems in the military.

Linear programming, in a different form is widely applied in electronics development and design. The basic Boolean algebra and Karnaugh map applications in electronic design led the way to the development of linear programming as a business tool (5:32). These efforts were initially developed to minimize electronic circuits in the design process and later led to the further development of linear programming as a business cost minimization tool. From this point they have developed as total decision aid systems applicable to a wide range of business and military applications.

The military services are presently making only limited use of linear programming. As presently employed, linear programming is used for some direct weapon allocation problems, limited parts control problems, and several other low order applications. In all of these cases linear programming is locally employed and does not cross command or service lines of responsibility. Also, to the authors
knowledge, linear goal programming is yet to be applied in the military environment.

RECOMMENDATION

The next step in the development of linear programming and linear goal programming should be their adoption by the military as standard decision aid tools. With the proper software development linear programming can provide utility to the joint forces commander, the maintenance officer, the supply organization, numerous other military organizations, and it can ultimately provide the rapid decision potential required in today's battlefield. This rapid decision potential is additionally applicable to the battlefield of tomorrow and might provide the a key input in any system requiring a quick, optimum decision when several alternatives are available.
A. REFERENCES CITED

Books


Official Documents


Other Sources

7. Powell, James F. Class notes from GBA-915, Decision Sciences, University of Nebraska, College of Business: Omaha, Nebraska, 1987.
CONTINUED

B. RELATED SOURCES

Books


Articles and Periodicals


APPENDIX A: Computer software code..........................29
APPENDIX B: Computer software code..........................38
APPENDIX C: Computer printout.................................54
Appendix A

LGP Macro Computer Program

This program is a dual simplex goal programming algorithm. It has been dimensioned for 150 decision variables, 150 constraints, and 10 goal priority levels.

Implicit real '8(A-H,P-Z)
Integer '2 ITIME(15)
Common NROW,NCOL,NVAR,NPRT,KTEST,ITER
Common /R1/ BASIS(150,300)
Common /R2/ VALC(11,300),VALB(11,150)
Common /R3/ PRHS(150),RHS(150)
Common /1/ IBASIC(150),JCOL(300)
Call TIMDAT (ITIME, INTS(15))
Write(6,888) ITIME(4),ITIME(5),ITIME(6)
   888 Format(' MIN',17,5X,'SEC',14,5X,'TICKS',15)
Write(6,889) ITIME(7),ITIME(8),ITIME(9),ITIME(10)
   889 Format(' CPU S',14,3X,'CPU T',15,5X,'10 S',14,3X,'10 T',15)
Call START
Call SIMPLX
Call FINISH
Call TIMDAT (ITIME,INTS(15))
Write(6,888) ITIME(4),ITIME(5),ITIME(6)
Write(6,889) ITIME(7),ITIME(8),ITIME(9),ITIME(10)
Stop
End

Extracted from Linear Goal Programming by Marc J. Schniederjans with permission.
SUBROUTINE START READS INPUT AND INITIATES WORKING MATRICES.

SUBROUTINE START
IMPLICIT REAL*8(A-H,P-Z)
INTEGER*4 POS,NEG,END
INTEGER*4 KSIGN
INTEGER*2 E,G,L,B
INTEGER*2 ISIGN
COMMON NROW,NCOL,NVAR,NPRT,KTEST,ITER
COMMON /R1/ BASIS(150,300)
COMMON /R2/ VALC(11,300),VALB(11,150)
COMMON /R3/ PRHS(150),RHS(150)
COMMON /I1/ IBASIC(150),JCOL(300)
DIMENSION ISIGN(300)
DATA POS,NEG,END, 'POS', 'NEG', 'END' /
DATA E,G,L,B,'E','G','L','B' /
READ(5,**) NROW
READ(5,**) NVAR
READ(5,**) NPRT
IF(NROW.LE.0) GO TO 91
IF(NVAR.LE.0) GO TO 91
IF(NPRT.LE.0) GO TO 91
NCOL = NROW + NVAR
DO 2 I = 1,NROW
   DO 1 J = 1,NCOL
      BASIS(I,J) = 0.0
      INDEX = J - NVAR
      IF(INDEX.EQ.I) BASIS(I,J) = 1.0
   1 CONTINUE
   IND = 1 + NCOL
   IBASIC(I) = IND
2 CONTINUE
DO 3 J = 1,NCOL
   JCOL(J) = J
3 CONTINUE
KEND = NPRT + 1
DO 6 K = 1,KEND
   DO 4 J = 1,NCOL
      VALC(K,J) = 0.0
   4 CONTINUE
   DO 5 I = 1,NROW
      VALB(K,I) = 0.0
5 CONTINUE
6 CONTINUE
KTEST = 0
READ(5,*) (ISIGN(I),I = 1,NROW)
DO 10 I = 1,NROW
   IF (ISIGN(I).EQ.E) GO TO 7
IF(SIG(I).EQ.G) GO TO 8
IF(SIG(I).EQ.L) GO TO 9
IF(SIG(I).EQ.B) GO TO 10
GO TO 92
7 KTEST = 1
INDEX = I + NVAR
VALB(I,I) = 1.0
VALC(I,INDEX) = 1.0
JCOL(INDEX) = 0
GO TO 10
8 INDEX = I + NVAR
KTEST = 1
VALC(I,INDEX) = 1.0
JCOL(INDEX) = 0
GO TO 10
9 KTEST = 1
VALB(I,I) = 1.0
10 CONTINUE
IF(KTEST.EQ.1) NPRT = NPRT + 1
11 READ(5,* ) KSIGN,I,K,WGT
IF(KSIGN.EQ.END) GO TO 13
IF(KTEST.EQ.1) K = K + 1
IF(KSIGN.EQ.POS) GO TO 12
IF(KSIGN.NE.NEG) GO TO 94
INDEX = I + NVAR
VALC(K,INDEX) = WGT
GO TO 11
12 CONTINUE
VALB(K,I) = WGT
GO TO 11
13 CONTINUE
15 READ(5,* ) I,J,AU
IF(I.EQ.0) GO TO 16
BASIS(I,J) = AU
GO TO 15
16 CONTINUE
READ(5,* ) (PRHS(I),I = 1,NROW)
DO 23 I = 1,NROW
IF(PRHS(I)) 20,21,22
21 PRHS(I) = 1.0E-12
22 RHS(I) = - PRHS(I)
23 CONTINUE
DO 31 J = 1,NCOL
IF(JCOL(J).NE.0) GO TO 31
DO 30 I = 1,NROW
BASIS(I,J) = 0.0
30 CONTINUE
LINEAR GOAL PROGRAMMING

31 CONTINUE
RETURN
91 WRITE(6,1091)
STOP
92 WRITE(6,1092)
STOP
94 WRITE(6,1094)
STOP
95 WRITE(6,1095)
STOP
1091 FORMAT (' NUMBER OF CONSTRAINTS, VARIABLES, OR PRIORITY LEVEL',/,' IMPROPERLY ENTERED. ')
1092 FORMAT (' SIGN SYMBOL SOMETHING OTHER THAN E, G, L, OR B. ')
1094 FORMAT (' DEVIATION TO BE MINIMIZED NOT POS OR NEG ')
1095 FORMAT (' THIS PROGRAM REQUIRES NON-NEGATIVE RIGHT HAND SIDES. ', ' MULTIPLY CONSTRAINT BY MINUS ONE. ')
END

C

THIS SUBROUTINE PERFORMS THE SIMPLEX OPERATION

SUBROUTINE SIMPLX
IMPLICIT REAL*8(A-H,P-Z)
COMMON NROW,NCOL,NVAR,NPRT,KTEST,ITER
COMMON /R1/ BASIS(150,300)
COMMON /R2/ VALC(11,300),VALB(11,150)
COMMON /R3/ PRHS(150),RHS(150)
COMMON /11/ IBASIC(150),JCOL(300)
DIMENSION JFAIL(150),JPICK(300),ZVAL(11,300)
KEND = NPRT + 1
DO 16 J = 1,NCOL
JPICK(J) = KEND
16 CONTINUE
DO 18 J = 1,NCOL
DO 17 K = 1,NPRT
IF(VALC(K,J),LE.1.0E-10) GO TO 17
JPICK(J) = K
17 CONTINUE
18 CONTINUE
ITER = 0
1 KEYROW = 0
KEYCOL = 0
KUNACH = 0
DO 2 I = 1,NROW
JFAIL(I) = 1
2 CONTINUE
IDENTIFY HIGHEST UNACHIEVED PRIORITY

DO 4 K = 1,NPRT
   DO 3 I = 1,NROW
      IF(VALB(K,I).LE.1.0E-10) GO TO 3
      KUNACH = K
      GO TO 11
   3 CONTINUE
   4 CONTINUE

IDENTIFY THE MOST NEGATIVE RHS

11 CONTINUE
   RMIN = -1.0E-10
   DO 12 I = 1,NROW
      IF(RHS(I).GE.RMIN) GO TO 12
      IF(JFAIL(I).EQ.0) GO TO 12
      KEYROW = I
      RMIN = RHS(I)
   12 CONTINUE

   IF KEYROW EQUALS 0, ALL RHS GREATER THAN OR EQUAL TO 0
   IF(KEYROW.EQ.0) GO TO 30

PATH FOR NEGATIVE RIGHT HAND SIDE

AU = 1.0E-8
   DO 25 M = 1,KEND
      L = KEND - M + 1
      DO 24 J = 1,NCOL
         IF(JCOL(J).EQ.0) GO TO 24
         IF(JPICK(J).LT.L) GO TO 24
         IF(BASIS(KEYROW,J).LE.AU) GO TO 24
         AU = BASIS(KEYROW,J)
         KEYCOL = J
      24 CONTINUE
      IF(KEYCOL.GT.0) GO TO 40
   25 CONTINUE
   IF(JFAIL(KEYROW) = 0)
      GO TO 11

PATH FOR NONNEGATIVE RIGHT HAND SIDE

30 CONTINUE
   IF(KUNACH.EQ.0) GO TO 96
   KFIN = KUNACH
LINEAR GOAL PROGRAMMING

The Zj matrix is developed. Since basis is negative of conventional, Zj calculated will be negative for favorable variables.

DO 33 K = KUNACH,NPRT
   DO 32 J = 1,NCOL
      ZVAL(K,J) = 0.0
      IF(JCOL(J).EQ.0) GO TO 32
      IF(JPICK(J).LT.KFIN) GO TO 32
      DO 31 I = 1,NROW
         IF(VALB(KQI).LE.-1.0E-10) GO TO 31
         IF(DABS(BASIS(I,J)).LE.1.0E-10) GO TO 31
         ZVAL(K,J) = ZVAL(K,J) + VALB(K)*BASIS(I,J)
      31 CONTINUE
      ZVAL(K,J) = ZVAL(K,J) + VALC(K,J)
   32 CONTINUE
   33 CONTINUE
   ZVALUE = -1.0E-8
   DO 36 K = KUNACH,NPRT
      DO 35 J = 1,NCOL
         IF(JCOL(J).EQ.0) GO TO 35
         IF(JPICK(J).LT.KFIN) GO TO 35
         IF(ZVAL(K,J).GE.ZVALUE) GO TO 35
         IF(K.LE.KUNACH) GO TO 39
         M = K - 1
         DO 34 L = 1,M
            IF(ZVAL(L,J).GE.1.0E-8) GO TO 35
         34 CONTINUE
      35 CONTINUE
   36 CONTINUE
   ZVALUE = ZVAL(K,J)
   KEYCOL = J
   37 CONTINUE
   IF(KEYCOL.GT.0) GO TO 37
   KFIN = KFIN + 1
   36 CONTINUE
   IF(KEYCOL.EQ.0) GO TO 97
   35 CONTINUE
   IF(KEYCOL.GT.0) GO TO 37
   37 CONTINUE
   THETA = 1.0E9
   DO 38 I = 1,NROW
      IF(BASIS(I,KEYCOL).GE.1.0E-10) GO TO 38
      IF(RHS(I).LE.1.0E-10) GO TO 38
      IF(RHS(I).LE.1.0E-10) RHS(I) = 1.0E-10
      ZETA = -RHS(I)/BASIS(I,KEYCOL)
      IF(ZETA.GE.THETA) GO TO 38
   38 CONTINUE
      THETA = ZETA
      KEYROW = I
   38 CONTINUE
   IF(KEYROW.GT.0) GO TO 40
   GO TO 97
SIMPLEX ROUTINE

40 CONTINUE
PIV = BASIS(KEYROW,KEYCOL)
DO 43 I = 1,NROW
  IF(I.EQ.KEYROW) GO TO 43
  IF(DBABS(BASIS(I,KEYCOL)).LE.1.0E-10) GO TO 43
  IF(DBABS(RHS(KEYROW)).LE.1.0E-10) GO TO 41
  RHS(I) = RHS(I) - (RHS(KEYROW)/PIV)*BASIS(I,KEYCOL)
41    DO 42 J = 1,NCOL
        IF(J.EQ.KEYCOL) GO TO 42
        IF(DBABS(BASIS(KEYROW,J)).LE.1.0E-10) GO TO 42
        BASIS(I,J) = BASIS(I,J) - (BASIS(I,KEYCOL)/PIV)*BASIS(KEYROW,J)
42    CONTINUE
BASIS(I,KEYCOL) = BASIS(I,KEYCOL)/PIV
43 CONTINUE
44 DO 45 J = 1,NCOL
        IF(J.EQ.KEYCOL) GO TO 45
        IF(DBABS(BASIS(KEYROW,J)).LE.1.0E-10) GO TO 45
        BASIS(KEYROW,J) = -BASIS(KEYROW,J)/PIV
45 CONTINUE
BASIS(KEYROW,KEYCOL) = 1/PIV
INDEX = JCOL(KEYCOL)
JCOL(KEYCOL) = IBASIC(KEYROW)
IBASIC(KEYROW) = INDEX
DO 46 K = 1,NPRT
  DUMMY = VALS(K,KEYROW)
  IF(DUMMY.GE.1.0E-8) JPICK(KEYCOL) = K
  VALB(K,KEYROW) = VALC(K,KEYCOL)
  VALC(K,KEYCOL) = DUMMY
46 CONTINUE
47 IF(KTEST.NE.1) GO TO 51
48 IF(VALC(1,KEYCOL).EQ.0.0) GO TO 51
49 JCOL(KEYCOL) = 0
50 DO 51 I = 1,NROW
      BASIS(I,KEYCOL) = 0.0
51 CONTINUE
52 CONTINUE
53 ITER = ITER + 1
54 GO TO 1
95 WRITE(6,1095)
96 RETURN
97 WRITE(6,1098)
STOP
1096 FORMAT( 'ALL GOALS ACHIEVED')
1098 FORMAT( 'THE MODEL IS INFEASIBLE')
END

C THIS SUBROUTINE REPORTS THE FINAL SOLUTION.

SUBROUTINE FINISH
IMPLICIT REAL*8(A-H,P-Z)
COMMON NROW,NCOL,NVAR,NPRT,KTEST,ITER
COMMON /R1/ BASIS(150,300)
COMMON /R2/ VALC(11,300),VALB(11,150)
COMMON /R3/ PRHS(150),RHS(150)
COMMON /I1/ IBASIC(150),JCOL(300)
DIMENSION X(150),POSD(150),RNEGD(150)

THIS SECTION IDENTIFIES AND REPORTS THE VALUES OF ALL MODEL VARIABLES. REAL VARIABLES ARE REPORTED FIRST, THEN DEVIATIONAL VARIABLES

DO 1 J = 1,NVAR
  X(J) = 0.0
1 CONTINUE
DO 2 I = 1,NROW
  POSD(I) = 0.0
  RNEGD(I) = 0.0
2 CONTINUE
DO 12 I = 1,NROW
  IVAR = IBASIC(I)
  IF(IVAR.GT.NCOL) GO TO 11
  IF(IVAR.GT.NVAR) GO TO 10
  X(IVAR) = RHS(I)
  GO TO 12
10 CONTINUE
  IND = IVAR - NVAR
  RNEGD(IND) = RHS(I)
  GO TO 12
11 CONTINUE
  IND = IVAR - NCOL
  POSD(IND) = RHS(I)
12 CONTINUE
WRITE(6,1000) ITER
WRITE(6,1001)
WRITE(6,1002)
DO 15 J = 1,NVAR
  WRITE(6,1003) J,X(J)
15 CONTINUE
WRITE(6,1004)
WRITE(6,1005)
DO 16 I = 1,NROW
WRITE(6,1006) I,PRHS(I),POSD(I),RNEGD(I)
16 CONTINUE

THIS SECTION PROVIDES A REPORT OF PRIORITY LEVEL
ACHIEVEMENT.
WRITE(6,1013)
KTOTAL = NPRT + 1
DO 52 K = 1,NPRT
KVAL = KTOTAL - K
M = KVAL
IF(KTEST.EQ.1) M = KVAL - 1
ZVALUE = 0.0
   DO 50 I = 1,NROW
      IF(VALB(KVAL,I).LE.1.OE-10) GO TO 50
      IF(DABS(RHS(I)).LE.1.OE-10) GO TO 50
      ZVALUE = ZVALUE + VALB(KVAL,I)*RHS(I)
50 CONTINUE
IF(KTEST.EQ.0) GO TO 51
IF(M.GT.0) GO TO 51
WRITE(6,1015) ZVALUE
GO TO 52
51 WRITE(6,1014) M,ZVALUE
52 CONTINUE
RETURN
1000 FORMAT (6, 'ITERATIONS')
1001 FORMAT ('DECISION VARIABLES')
1002 FORMAT ('VARIABLE VALUE')
1003 FORMAT (3X,15,3X,F15.5)
1004 FORMAT ('ANALYSIS OF DEVIATIONS FROM GOALS')
1005 FORMAT ('ROW',8X,'RHS-VALUE',10X,'POSITIVE DEVIATION',6X,'NEGATIVE DEVIATION')
1006 FORMAT (13,9X,F20.5)
1013 FORMAT ('ANALYSIS OF THE OBJECTIVE FUNCTION',//,'PRIORITY',9 X,'UNDERACHIEVEMENT')
1014 FORMAT (13,9X,F20.5)
1015 FORMAT ('ARTIFICIAL',F20.5)
END
Appendix B

LGP Micro Computer Program

10 REM 'SET UP PROBLEM AND FLAGS
20 HOME : CLEAR
30 D$ = CHR$ (4)
40 INPUT "DO YOU WANT INSTRUCTIONS? ";RP$
50 IF RP$ = "Y" THEN GOSUB 2150
60 IF RP$ = "Y" OR RP$ = "N" THEN GOTO 80
70 PRINT "Y OR N ONLY. TRY AGAIN."; GOTO 40
80 PRINT
90 INPUT "IS YOUR PROBLEM ALREADY ON FILE? ";RR$
100 IF RR$ = "Y" OR RR$ = "N" THEN GOTO 120
110 PRINT "Y OR N ONLY. TRY AGAIN."; GOTO 90
120 PRINT
130 INPUT "NAME YOUR PROBLEM. ";PR$
140 IF RR$ = "Y" THEN GOSUB 470
150 IF RR$ = "N" THEN GOSUB 790
160 IF RR$ = "N" THEN GOSUB 5740
170 IF RR$ = "N" THEN GOTO 230
180 IF RR$ = "Y" THEN INPUT "DO YOU WANT TO CHANGE IT? ";PF$
190 IF PF$ = "Y" THEN GOSUB 6020
200 IF PF$ = "Y" THEN GOTO 230
210 IF PF$ = "N" THEN GOTO 230
220 PRINT "Y OR N ONLY. TRY AGAIN."; GOTO 180
230 PRINT
240 INPUT "DO YOU WANT PRINTOUT? ";PO$
250 IF PO$ = "Y" THEN PRINT D$"PR#1": PRINT PR$: PRINT :
      PRINT D$"PR#0"

Extracted from Linear Goal Programming
by Marc J. Schniederjans with permission 211
LINEAR GOAL PROGRAMMING

260 IF POS = "Y" THEN INPUT "INCLUDING TABLEAU? "; TBS: IF
     TBS = "Y" THEN GOTO 300
270 IF POS = "Y" THEN GOTO 330
280 IF POS = "N" THEN GOTO 330
290 PRINT "Y OR N ONLY. TRY AGAIN."; GOTO 240
300 INPUT "(A)LL OR JUST (F)IRST? "; QQ$;
310 IF QQ$ = "A" OR QQ$ = "F" THEN GOTO 330
320 PRINT "A OR F ONLY. TRY AGAIN."; GOTO 300
330 PRINT
340 GOSUB 2070
350 GOSUB 5480
360 IF QQ$ = "F" AND TC > 0 THEN GOTO 420
370 IF QQ$ = "F" THEN GOSUB 5620
380 IF QQ$ = "F" THEN GOSUB 5020
390 IF QQ$ = "F" THEN GOTO 420
400 IF QQ$ = "A" AND TC = 0 THEN GOSUB 5620
410 IF QQ$ = "A" THEN GOSUB 5020
420 GOSUB 3210
430 GOSUB 3710
440 GOSUB 4350
450 GOSUB 4690
460 GOTO 350
470 REM READ FILE FROM DISK
480 PRINT D$"OPEN",PR$;","L300"
490 PRINT D$"READ",PR$;","R";0
500 INPUT NU: INPUT MC; INPUT P:
510 DIM A(MC,NU + (2 * MC) + 1),CZ(P,NU + (2 * MC) + 1),C(NU + (2 * MC)),B(MC), WC(NU + (2 * MC)),WB(MC)
520 DIM N$(NU)
530 DIM Y$(MC),DI(NU + 2 * MC)
540 DIM DB(NU + 2 * MC)
550 FOR I = 1 TO MC
560 PRINT D$"READ",PR$;","R";I
570 FOR J = 1 TO NU + (2 * MC) + 1
580 INPUT A(I,J)
590 NEXT J
600 NEXT I
610 I = MC + 1
620 PRINT D$"READ",PR$;","R";I
630 FOR J = 1 TO NU + (2 * MC)
640 INPUT C(J)
650 NEXT J
660 I = I + 1
670 PRINT D$"READ",PR$;","R";I
680 FOR J = 1 TO NU + (2 * MC)
690 INPUT WC(J)
700 NEXT J
710 IF N$ < > "Y" THEN GOTO 770
LGP Macro Computer Program

720 \( i = i + 1 \)
730 PRINT D$:"READ";PR$;"","R";i
740 FOR J = 1 TO NU
750 INPUT N$(J)
760 NEXT J
770 PRINT D$:"CLOSE";PR$;"
780 RETURN
790 REM DATA ENTRY ROUTINE
800 HOME: PRINT "READY TO ENTER DATA.";NU
810 PRINT
820 PRINT "WANT TO NAME VARIABLES? ";N$
830 PRINT
840 IF N$ = "N" THEN GOTO 920
850 IF N$ = "Y" THEN GOTO 880
860 PRINT "TRY AGAIN."; GOTO 830
870 PRINT "VARIABLE X";I;: INPUT "REPRESENTS ";N$(I)
880 DIM N$(NU): FOR I = 1 TO NU
890 PRINT "REMEMBER ONLY EQUATIONS WITH"
900 PRINT "DECISION VARIABLES COUNT IN"
910 PRINT "ANSWER NEXT QUESTION."
920 INPUT "NUMBER OF UNKNOWNS ";NU
930 PRINT
940 DIM N$(NU): FOR I = 1 TO NU
950 INPUT "NUMBER OF CONSTRAINTS ";MC
960 PRINT
970 INPUT "NO. OF DEVIATIONAL VARIABLES? ";DV
980 PRINT
990 INPUT "NUMBER OF PRIORITIES ";P
1000 PRINT
1010 DIM A(DV,NU + (2 * DV) + 1)
1020 DIM CZ(P,NU + 2 * DV +1)
1030 DIM C(NU + 2 * DV)
1040 DIM B(DV)
1050 DIM WC(NU + (2 * DV))
1060 DIM WB(DV)
1070 DIM Y$(DV),DI(NU + 2 * DV)
1080 FOR I = 1 TO MC
1090 HOME: PRINT "IF ONLY 3 OR 4 UNKNOWNS IN PROBLEM"
1100 PRINT "ANSWER NEXT QUESTION WITH 'A'."
1110 PRINT "IT WILL BE FASTER TO ENTER ALL."
1120 PRINT "TYPE IN NUMBER OF VARIABLES ";I
1130 PRINT "WHICH APPEAR IN EQUATION ";I
1140 INPUT "OR A FOR (A)LL ";AA$
1150 IF AA$ = "A" THEN GOTO 1240
1160 AA = VAL (AA$)
1170 FOR J = 1 TO AA
1180 PRINT "ENTER SUBSCRIPT OF UNKNOWN ";J
1190 PRINT "IN EQUATION ";J;": INPUT BB$
LINEAR GOAL PROGRAMMING

1200 PRINT "ENTER VALUE OF X":BB$;"": INPUT A$
1210 A(I, VAL (BB$)) = VAL (A$)
1220 NEXT J
1230 GOTO 1280
1240 FOR J = 1 TO NU
1250 PRINT "ENTER VALUE OF X";J;"": INPUT A$
1260 A(I,J) = VAL (A$)
1270 NEXT J
1280 PRINT
1290 PRINT "IS POS. DEV. VAR. ALLOWED IN"
1300 PRINT "EQUATION ";I; "": INPUT CC$
1310 IF CC$ = "N" THEN GOTO 1350
1320 IF CC$ = "Y" THEN GOTO 1340
1330 PRINT "Y OR N ONLY. TRY AGAIN." : GOTO 1280
1340 A(I,NU + I) = 1:A(I,NU + DV + I) = -1: GOTO 1360
1350 A(I,NU + I) = 1
1360 PRINT "RHS FOR EQUATION ";I; "": INPUT A$
1370 A(I,NU + 2 * DV + 1) = VAL (A$)
1380 IF A(I,NU + 2 * DV + 1) >= 0 THEN GOTO 1420
1390 FOR J = 1 TO NU + 2 * DV + 1
1400 A(I,J) = A(I,J) + 1
1410 NEXT J
1420 PRINT
1430 PRINT "EQUATION ";I;" READS:
1440 PRINT
1450 FOR J = 1 TO NU
1460 PRINT A(I,J);"X";J;" + ";
1470 NEXT J
1480 PRINT "D";I;"— ";
1490 IF CC$ = "Y" THEN PRINT " — D";I;" + ";
1500 PRINT " ";
1510 PRINT A(I,NU + 2 * MC + 1)
1520 PRINT "IS IT RIGHT?": INPUT A$
1530 IF A$ = "Y" THEN GOTO 1560
1540 IF A$ = "N" THEN HOME: PRINT "REENTER EQUATION ";I;"." : GOTO 1170
1550 PRINT "Y OR N ONLY. TRY AGAIN." : GOTO 1520
1560 NEXT I
1570 IF DV > MC THEN PRINT "YOUR DEV. VARS. NOT IN OTHER CONSTRAINTS ARE:" : GO TO 1590
1580 GOTO 1840
1590 PRINT "D";MC + 1;" TO D";DV
1600 FOR I = MC + 1 TO DV
1610 PRINT
1620 PRINT "HOW MANY OTHER DEV. VARS. APPEAR"
1630 PRINT "IN EQUATION FOR D";I;" : INPUT SS
1640 IF SS = 0 THEN NEXT I
1650 FOR J = 1 TO SS

214.
LGP Macro Computer Program

1660 PRINT "EQUATION NUMBER OF DEV. VAR. ";J; "IN D";I;:
      INPUT TT
1670 INPUT "(P)OS OR (N)EG DEV. VAR.? ",V$:
1680 IF V$ = "P" THEN GOTO 1690
1681 IF V$ = "N" THEN GOTO 1690
1682 GOTO 1670
1690 INPUT "(P)OS OR (N)EG VALUE? ",W$:
1700 IF W$ = "P" THEN GOTO 1690
1701 IF W$ = "N" THEN GOTO 1710
1702 GOTO 1670
1710 IF V$ = "P" AND W$ = "P" THEN A(I,NU + DV + TT) = 1
1720 IF V$ = "P" AND W$ = "N" THEN A(I,NU + DV + TT) = -1
1730 IF V$ = "N" AND W$ = "P" THEN A(I,NU + TT) = 1
1740 IF V$ = "N" AND W$ = "N" THEN A(I,NU + TT) = -1
1750 A(I,NU + I) = 1:A(I,NU + DV + I) = -1
1760 NEXT J
1770 PRINT "RHS FOR THIS EQUATION? ": INPUT A(I,NU + 2 * DV + 1)
1780 IF A(I,NU + 2 * DV + 1) > = 0 THEN GOTO 1820
1790 FOR J = 1 TO NU + 2 * MC + 1
1800 A(I,J) = A(I,J) * -1
1810 NEXT J
1820 NEXT I
1830 MC = DV
1840 L = 1
1850 PRINT
1860 FOR J = NU + 1 TO NU + MC
1870 PRINT "PRIORITY ASSOCIATED WITH D",L;"-": INPUT A$
1880 IF A$ = CHR$(13) THEN C(J) = 0: GOTO 1900
1890 C(J) = VAL (A$)
1900 INPUT "WEIGHT FOR THE DEV.VAR. ",A$
1910 IF A$ = CHR$(13) THEN WC(J) = 0: GOTO 1930
1920 WC(J) = VAL (A$)
1930 L = L + 1
1940 NEXT J
1950 L = 1
1960 FOR J = NU + MC + 1 TO NU + (2 * MC)
1970 PRINT "PRIORITY ASSOCIATED WITH D",L;"+": INPUT A$
1980 IF A$ = CHR$(13) THEN C(J) = 0: GOTO 2000
1990 C(J) = VAL (A$)
2000 INPUT "WEIGHT FOR THE DEV.VAR. ",A$
2010 IF A$ = CHR$(13) THEN WC(J) = 0: GOTO 2030
2020 WC(J) = VAL (A$)
2030 L = L + 1
2040 PRINT
2050 NEXT J
2060 RETURN
LINEAR GOAL PROGRAMMING

2070 REM CB IN INITIAL TABLEAU
2080 I = 1
2090 FOR J = NU + 1 TO NU + MC
2100 B(I) = C(J)
2110 WB(I) = WC(J): Y$(I) = "D" + STR$(J - NU) + "-"
2120 I = I + 1
2130 NEXT J
2140 RETURN

REM INSTRUCTIONS
2150 REM HOME
2160 HOME
2170 PRINT "THIS PROBLEM SOLVES A GOAL PROGRAMMING"
2180 PRINT "PROBLEM BUT NEEDS A LITTLE INTRO."
2190 PRINT
2200 PRINT "TO THIS END, 'SCHNIEDERJANS'"
2210 PRINT "EXAMPLE' HAS BEEN PROVIDED"
2220 PRINT "ON DISKETTE. IT IS PRETTY STRAIGHT—"
2230 PRINT "FORWARD EXCEPT FOR THE"
2240 PRINT "PRIORITIES WHERE, FOR MODELING"
2250 PRINT "REASONS, THE ARTIFICIAL, OR 0,
2260 PRINT "PRIORITY BECOMES PRIORITY 1 AND"
2270 PRINT "ALL THE OTHER, STATED, PRIORITIES"
2280 PRINT "SLIP DOWN ONE. INSTEAD OF 'FOUR'
2290 PRINT 'THE ANSWER TO THE QUESTION, 'NUMBER"
2300 PRINT 'OF PRIORITIES?' IS 'FIVE'.
2310 PRINT "THE OPERATOR MUST MAKE THIS"
2320 PRINT "CONVERSION."
2330 PRINT: PRINT "PRESS ANY KEY TO CONTINUE.";; GET Q$
2340 HOME
2350 PRINT "ANOTHER PROGRAM QUIRK OCCURS IF THERE"
2360 PRINT "ARE NO UNKNOWNS (JUST DEV. VARS.)"
2365 PRINT "IN AN EQUATION."
2370 PRINT "WHEN IT ASKS FOR NUMBER OF CONSTRAINTS"
2380 PRINT "ONLY ENTER THE NUMBER IN WHICH"
2390 PRINT "UNKNOWNS APPEAR."
2400 PRINT "DON'T TRY TO SUBSTITUTE DEVIATIONAL"
2410 PRINT "VARIABLES INTO CONSTRAINTS LIKE"
2420 PRINT "YOU MIGHT TRY TO DO TO LIMIT OVERTIME."
2430 PRINT "INSTEAD, PROGRAM WILL CREATE"
2440 PRINT "SEPARATE EQUATIONS FOR STAND ALONE"
2450 PRINT "DEVIATIONAL VARIABLES AS YOU"
2460 PRINT "ANSWER FURTHER QUESTIONS ABOUT"
2470 PRINT "DEV. VARS. IN EXCESS OF CONSTRAINTS."
2480 PRINT: PRINT "PRESS ANY KEY TO CONTINUE.";; GET Q$
2490 HOME
2500 PRINT "TABLEAUX ARE NOT NEAT. THEY"
2510 PRINT "ARE JAMMED TOGETHER IN THE ATTEMPT"
2520 PRINT "TO GET ALL ON THE FEWEST PRINTER"
2530 PRINT "LINES. THE OPTION TO PRINT"
PRINT "THEM IS PROVIDED FOR CHECKOUT"
PRINT "PURPOSES ONLY. READ STARTING WITH"
PRINT "RHS COLUMN ON LEFT IN A(I,J) PORTION"
PRINT "AND AMOUNT OF REMAINING PRIORITY"
PRINT "TO FILL ON LEFT IN ZJ-CZ PORTION."
PRINT "CB AND CJ ARE NOT PRINTED."
PRINT "YOU CAN DETERMINE WHAT THEY ARE AT"
PRINT "END OF RUN BY ASKING FOR PRINT"
PRINT "OF 'C(I)' AND 'WC(I)' FOR CJ"
PRINT "WHERE 'I' IS COLUMN NUMBER, 'C(I)'"
PRINT "IS THE SUBSCRIPT PRIORITY AND 'WC(I)'"
PRINT "IS THE WEIGHT ASSIGNED."
PRINT "THE SAME IS TRUE FOR 'CB' USING"
PRINT "'B(I)' AND 'WB(I)'
PRINT "WHERE 'I' IS THE ROW NUMBER."
PRINT "PRESS ANY KEY TO CONTINUE. ";: GET QS
PRINT "THE OTHER IMPORTANT VARIABLE NAMES"
PRINT "ARE 'A(I,J)' FOR A(I,J)"
PRINT "AND 'CZ(I,J)' FOR ZJ-CJ"
PRINT "WHERE 'I' IS 1 TO NUMBER OF DEV."
PRINT "VARS. IN A(I,J) AND 1 TO NUMBER"
PRINT "OF PRIORITIES IN CZ(I,J) AND"
PRINT "J IS 1 TO NUMBER OF CONSTRAINTS PLUS"
PRINT "TWICE THE NUMBER OF DEV. VARS. PLUS"
PRINT "ONE(TO INCLUDE RHS) IN BOTH CASES."
PRINT "PRESS ANY KEY TO CONTINUE. ";: GET QS
PRINT "TO RUN PROGRAM, THE FIRST QUESTION"
PRINT "GOT YOU HERE. THE NEXT WILL"
PRINT "ASK WHETHER OR NOT YOUR PROBLEM"
PRINT "IS ALREADY ON FILE (THE DISK)."
PRINT "ANSWER 'Y' OR 'N' AS APPROPRIATE."
PRINT "THE NEXT QUESTION ASKS YOU TO"
PRINT "NAME YOUR PROBLEM."
PRINT "BE CAREFUL NOT TO USE A NAME"
PRINT "OF A FILE ALREADY ON DISK."
PRINT: FLASH: PRINT "IT WILL GET WIPED OUT.": NORMAL
PRINT: PRINT "TO CHANGE A PROBLEM ON DISK ANSWER "
PRINT "YES"
PRINT "TO NEXT QUESTION. TO RERUN A PROBLEM"
PRINT "ALREADY ON DISK ANSWER NO TO"
PRINT "THIS QUESTION AND PROGRAM WILL"
LINEAR GOAL PROGRAMMING

3170 PRINT: PRINT "I THINK THAT SHOULD BE ENOUGH"
3180 PRINT "TO ALLOW YOU TO RUN PROGRAM."
3190 PRINT: PRINT "PRESS ANY KEY TO START": GET Q$  
3200 HOME: RETURN
3210 REM CHECK FOR DONE
3220 Z = 0
3230 FOR K = 1 TO P
3240 IF CZ(K,NU + 2 * MC + 1) > 0 THEN GOTO 3270
3250 NEXT K
3260 GOTO 5230
3270 IF K > 1 AND Z = 0 THEN GOTO 3550
3280 IF K > 1 THEN GOTO 3380
3290 FOR J = 1 TO NU + 2 * MC
3300 IF CZ(K,J) > 0 AND J < NU + 1 THEN RETURN
3310 IF CZ(K,J) > 0 AND K < C(J) THEN RETURN
3320 NEXT J
3330 NEXT K
3340 PRINT "NO POS VALUES IN PRIORITY 1."
3350 PRINT "PROBLEM IS INFEASIBLE."
3360 PRINT "PRESENT STATUS IS:"
3370 PRINT: GOTO 5230
3380 FOR J = 1 TO NU + 2 * MC
3390 IF CZ(K,J) > 0 THEN GOTO 3430
3400 NEXT J
3410 NEXT K
3420 GOTO 5230
3430 FORM = 1 TO K - 1
3440 FOR M = 1 TO K - 1
3450 IF CZ(M,J) < 0 THEN GOTO 3470
3460 GOTO 3500
3470 NEXT M
3480 NEXT K
3490 GOTO 5230
3500 IF J < NU + 1 THEN RETURN
3510 IF K < C(J) THEN RETURN
3520 NEXT J
3530 NEXT K
3540 GOTO 5230
3550 Z = Z + 1
3560 FOR I = 1 TO K - 1
3570 IF CZ(I,NU + 2 * MC + 1) = 0 THEN GOTO 3590
3580 NEXT I
3590 FOR J = 1 TO NU + 2 * MC
3600 IF CZ(I,J) > 0 THEN GOTO 3640
3610 NEXT J
3620 NEXT I
3630 GOTO 3380
3640 IF I = 1 THEN RETURN

218
3650 FOR L = 1 TO M - 1
3660 IF CZ(L,J) < 0 THEN GOTO 3610
3670 NEXT L
3680 IF J < NU + 1 THEN RETURN
3690 IF I < C(J) THEN RETURN
3700 GOTO 3610
3710 REM DETERMINE PIVOT COLUMN
3720 Z = 0: I = 0
3730 FOR K = 1 TO P
3740 FOR J = 1 TO NU + 2 * MC
3750 IF CZ(K,J) < = 0 THEN GOTO 3820
3760 IF CZ(K,J) < Z THEN GOTO 3820
3770 IF CZ(K,J) = Z THEN GOSUB 3830
3780 IF CV = 1 THEN CV = 0: GOTO 3820
3790 IF CZ(K,J) = 0 THEN GOTO 3810
3800 IF CZ(K,J) > Z THEN Z = CZ(K,J): I = 1: DI(1) = J: GOTO 3820
3810 I = I + 1: DI(I) = J
3820 NEXT J: GOTO 3920
3830 CV = 0
3840 IF K = 1 THEN RETURN
3850 IF K = 2 THEN GOTO 3900
3860 FOR JJ = K - 1 TO 1 STEP - 1
3870 IF CZ(JJ,J) < 0 THEN CV = 1: RETURN
3880 NEXT JJ
3890 RETURN
3900 IF CZ(1,J) < 0 THEN CV = 1
3910 RETURN
3920 IF I = 1 THEN M = DI(1): TB = K: GOTO 4300
3930 IF I > 1 THEN GOTO 3950
3940 NEXT K: GOTO 5230
3950 Z = 0: I = 0
3960 IF K >= P THEN GOSUB 4310
3970 M = DI(1 + CZ): TB = P
3980 FOR L = K + 1 TO P
3990 FOR J = 1 TO I
4000 IF CZ(L,DI(J)) < = 0 THEN GOTO 4070
4010 IF CZ(L,DI(J)) < Z THEN GOTO 4070
4020 IF CZ(L,DI(J)) = Z THEN GOSUB 4080
4030 IF CV = 1 THEN CV = 0: GOTO 4070
4040 IF CZ(L,DI(J)) = Z THEN GOTO 4060
4050 IF CZ(L,DI(J)) > Z THEN Z = CZ(L,DI(J)): II = 1: DB(II) = DI(J): GOTO 4070
4060 II = II + 1: DB(II) = DI(J)
4070 NEXT J: GOTO 4160
4080 CV = 0
4090 IF L = 2 THEN GOTO 4140
4100 FOR JJ = L - 1 TO L STEP - 1
LINEAR GOAL PROGRAMMING

4110 IF CZ(J,J,DI(J)) < 0 THEN CV = 1: RETURN
4120 NEXT J
4130 RETURN
4140 IF CZ(1,DI(J)) < 0 THEN CV = 1
4150 RETURN
4160 IF II > 0 THEN I = II
4170 IF Z = 0 THEN GOTO 4270
4180 IF II = 1 THEN M = DB(1):TB = L: GOTO 4300
4190 IF L > = P AND II = 0 THEN GOSUB 4310
4200 M = DI(QZ + 1):TB = P: RETURN
4210 IF L > = P THEN GOSUB 4330
4220 M = DB(1 + QZ):TB = P: RETURN
4230 FOR J = 1 TO II
4240 DI(J) = DB(J)
4250 NEXT J
4260 Z = 0:II = 0
4270 NEXT L
4280 GOSUB 4310
4290 M = DI(1):TB = 1: RETURN
4300 RETURN
4310 IF I = QZ THEN PRINT "ALL PIVOTS TRIED": PRINT "NO WAY OUT OF LOOP": PRINT "CURRENT STATUS IS:"; GOTO 5230
4320 RETURN
4330 IF II = QZ THEN PRINT "ALL PIVOTS TRIED": PRINT "NO WAY OUT OF LOOP": PRINT "CURRENT STATUS IS:"; GOTO 5230
4340 RETURN
4350 REM DETERMINE PIVOT ROW
4360 P2 = P1:P1 = PC:PC = M:M = 1
4370 IF QQ$ = "F" AND TC > 0 THEN GOTO 4390
4380 IF TB$ = "Y" THEN PRINT D$"PR#1"
4390 PRINT "PIVOT COLUMN = ";PC
4400 IF TB$ = "Y" THEN PRINT D$"PR#0"
4410 J = 0:M = 0:DR = 0
4420 FOR I = 1 TO MC
4430 IF A(I,PC) < = 0 THEN DI(I) = 0: GOTO 4500
4440 DI(I) = A(I,(NU + 2 * MC + 1)) / A(I,PC)
4450 IF DI(I) < 0 THEN GOTO 4500
4460 IF DR = 0 THEN GOTO 4480
4470 IF DI(I) > DR THEN GOTO 4500
4480 IF DI(I) < DR OR DR = 0 THEN M = I:DR = DI(I):J = 1:DB(J) = I: GOTO 4500
4490 IF DI(I) = DR AND DR > 0 THEN J = J + 1:DB(J) = I
4500 NEXT I
4510 IF J = 0 THEN PRINT "THE SOLUTION IS UNBOUNDED."
END
4520 IF J = 1 THEN GOTO 4680

220
4530 DR = O:K = 0
4540 FOR I = 1 TO J
4550 IF B(DB(I)) < 0 THEN GOTO 4600
4560 IF DR = 0 THEN GOTO 4580
4570 IF B(DB(I)) > DR THEN GOTO 4600
4580 IF B(DB(I)) < DR OR DR = 0 THEN M = DB(I):DR =
           B(DB(I)):K = 1:DI(K) = DB(I): GOTO 4600
4590 IF B(DB(I)) = DR AND DR > 0 THEN K = K + 1:DI(K) =
           DB(I)
4600 NEXT I
4610 IF K = 0 OR K = 1 THEN GOTO 4680
4620 FOR I = 1 TO K
4630 IF WB(DI(I)) < DR THEN GOTO 4670
4640 IF WB(DI(I)) > DR THEN M = DI(I):DR =
           WB(DI(I)):L =
           1:DB(L) = DI(I): GOTO 4670
4650 IF WB(DI(I)) = DR THEN M = DI(I):L = L + 1:DB(L) = DI(I)
4670 NEXT I
4680 RETURN
4690 REM DETERMINE COEFFICIENTS FOR NEXT TABLEAU
4700 IF QQ$ = "F" AND TC > 0 THEN GOTO 4720
4710 IF TB$ = "Y" THEN PRINT D$PR#1"
4720 P4 = P3:P3 = PR:PR = M:M = I:DI = A(PR,PC)
4730 PRINT "PIVOT ROW = ",PR: PRINT
4740 IF TB$ = "Y" THEN PRINT D$PR#O"
4750 IF PC = P2 AND PR = P4 THEN PRINT "IN A LOOP, TRYING
           AGAIN. ": GOTO 4960
4760 QZ = 0
4770 FOR J = 1 TO NU + (2 * MC) + 1
4780 A(PR,J) = A(PR,J) / DI
4790 NEXT J
4800 FOR I = 1 TO MC: GOTO 4810
4810 IF I = PR THEN NEXT I: GOTO 4870
4820 DI = A(I,PC)
4830 FOR J = 1 TO NU + (2 * MC) + 1
4840 A(I,J) = A(I,J) - (DI * A(PR,J))
4850 NEXT J
4860 NEXT I
4870 B(PR) = C(PC)
4880 WB(PR) = WC(PC)
4890 TC = TC + 1
4900 IF PC < NU THEN Y$(PR) = "X" + STR$(PC): RETURN
4910 IF PC > NU AND PC < NU + MC + 1 THEN Y$(PR) = "D" +
           STR$(PC - NU) + "-" :RETURN
4920 Y$(PR) = "D" + STR$(PC - NU - MC) + "+
4930 RETURN
4940 REM LAST THREE ROWS BROUGHT NEW VARIABLE
           NAMES
LINEAR GOAL PROGRAMMING

4950 REM AND WEIGHTS/PRIORITIES INTO BASIS
4960 QZ = QZ + 1
4970 FOR J = 1 TO NU + 2 * MC
4980 IF J = PC THEN GOTO 5000
4990 IF CZ(TB,J) = CZ(TB,PC) THEN GOTO 430
5000 NEXT J
5010 GOTO 5230
5020 REM TABLEAU PRINTOUT
5030 IF TB$ = "Y" THEN PRINT D$"PR#1"
5040 PRINT
5050 PRINT "COEFFICIENTS IN TABLEAU:"
5060 PRINT
5070 FOR I = 1 TO MC
5080 PRINT Y$(I);";A(I,NU + (2 * MC) + 1);"
5090 FOR J = 1 TO NU + (2 * MC)
5100 PRINT A(I,J);";";
5110 NEXT J: PRINT: NEXT I
5120 PRINT
5130 PRINT "VALUES IN ZJ-CJ:"
5140 PRINT
5150 FOR K = P TO 1 STEP - 1
5160 PRINT "P";K;";"
5170 PRINT CZ(K,NU + (2 * MC) + 1);"
5180 FOR J = 1 TO NU + (2 * MC)
5190 PRINT CZ(K,J);";";
5200 NEXT J: PRINT: NEXT K
5210 IF TB$ = "Y" THEN PRINT D$"PR#0"
5220 RETURN
5230 REM SOLUTION PRINTOUT
5240 IF PS$ = "Y" THEN PRINT D$"PR#1"
5250 PRINT
5260 PRINT "SOLUTION VARIABLES ARE:"
5270 PRINT
5280 FOR I = 1 TO MC
5290 QQ = LEN(Y$(I))
5300 IF N$ = "Y" AND LEFT$(Y$(I),1) = "X" THEN PP$ = RIGHT$(Y$(I),QQ - 1):PP = VAL(PP$): PRINT N$(PP); HTAB 20: PRINT A(I,NU + (2 * MC) + 1): GOTO 5320
5310 PRINT Y$(I): HTAB 20: PRINT A(I,NU + (2 * MC) + 1): GOTO 5320
5320 NEXT I
5330 PRINT
5340 Z = 0
5350 PRINT "UNACHIEVED GOALS ARE:"
5360 PRINT
5370 FOR K = 1 TO P
5380 IF CZ(K,NU + (2 * MC) + 1) = 0 THEN GOTO 5400
5390 PRINT "P";K; HTAB 20: PRINT CZ(K,NU + (2 * MC) + 1):Z = 

222
LGP Macro Computer Program

5400 NEXT K
5410 IF Z = 0 THEN PRINT "NONE": PRINT: PRINT: PRINT
5420 IF PO$ = "Y" THEN PRINT DS"PR#0"
5430 PRINT "PRESS 'R' TO DO ANOTHER PROBLEM"
5440 INPUT "OR PRESS 'Q' TO QUIT.";X$
5450 IF X$ = "Q" THEN END
5460 IF X$ = "R" THEN HOME: GOTO 10
5470 PRINT "YOU HIT A WRONG KEY.": GOTO 5430
5480 REM ZJ CALCULATION
5490 Z = 0
5500 FOR K = 1 TO P
5510 FOR J = 1 TO NU + (2 * MC) + 1
5520 FOR I = 1 TO MC
5530 IF B(I) = K THEN Z = Z + WB(I) * A(I,J)
5540 NEXT I
5550 CZ(K,J) = Z
5560 Z = 0
5570 IF J = NU + 2 * MC + 1 THEN GOTO 5590
5580 IF C(J) = K THEN CZ(K,J) = CZ(K,J) + WC(J)
5590 NEXT J
5600 NEXT K
5610 RETURN
5620 REM PRIORITY AND WEIGHT PRINTOUT
5630 IF TB$ = "Y" THEN PRINT DS"PR#1"
5640 IF TB$ = "N" THEN RETURN
5650 IF QQ$ = "F" AND TC > 0 THEN RETURN
5660 IF TC = 0 THEN I = 1: K = 1: PRINT : GOTO 5680
5670 RETURN
5680 FOR J = NU + 1 TO NU + MC: PRINT "D";I; "= ";C(J); HTAB 30: PRINT "WT ";WC(J)
5690 I = I + 1: NEXT J
5700 FOR J = NU + MC + 1 TO NU + (2 * MC): PRINT "D";K; "+ ";C(J); HTAB 30: PRINT "WT ";WC(J)
5710 K = K + 1: NEXT J
5720 IF TB$ = "Y" THEN PRINT DS"PR#0"
5730 RETURN
5740 REM WRITE SUBROUTINE
5750 PRINT DS"OPEN";PR$;"L300"
5760 FOR I = 1 TO MC
5770 PRINT DS"WRITE";PR$;"R";I
5780 FOR J = 1 TO NU + (MC * 2) + 1
5790 PRINT A(I,J)
5800 NEXT J
5810 NEXT I
5820 I = MC + 1
5830 PRINT DS"WRITE";PR$;"R";I
5840 FOR J = 1 TO NU + (2 * MC)
5850 PRINT C(J)
5860 NEXT J
LUIZA GOAL PROGRAM

5870  I = I + 1
5880  PRINT D$"WRITE";PR$;",";I
5890  FOR J = 1 TO NU + (2 * MC)
5900  PRINT WC(J)
5910  NEXT J
5920  I = I + 1
5930  IF N$ = "Y" THEN PRINT D$"WRITE";PR$;",";I: GOTO 5950
5940  GOTO 5980
5950  FOR J = 1 TO NU
5960  PRINT N$(J)
5970  NEXT J
5980  PRINT D$"CLOSE";PR$;
6000  PRINT N$: PRINT MC: PRINT P: PRINT N$
6010  RETURN
6020  REM CHANGE SUBROUTINE
6030  HOME: PRINT
6040  PRINT "IF YOU WANT TO CHANGE PRIORITIES"
6050  PRINT "THE ANSWER TO EQUATION NO. IS ";MC + 1
6060  PRINT "IF YOU WANT TO CHANGE WEIGHTS"
6070  PRINT "THE ANSWER TO EQUATION NO. IS ";MC + 2
6080  PRINT "IF YOU WANT TO ADD OR DELETE"
6090  PRINT "PRIORITIES OR CONSTRAINTS"
6100  PRINT "RESET AND RUN A NEW PROBLEM."
6110  PRINT
6120  PRINT "ANSWER QUESTIONS ABOUT WHAT"
6130  PRINT "UNKNOWN OR DEV. VAR. WITH THE"
6140  PRINT "FULL NAME (E.G. 'X1' OR 'D1+')."
6150  PRINT
6160  INPUT "WHAT EQUATION NUMBER? ";KK
6170  IF KK > MC + 2 THEN PRINT "NOT THAT MANY EQUATIONS. ONLY ";MC: GOTO 6190
6180  GOTO 6200
6190  PRINT "PLUS 2 FOR WEIGHTS AND PRIORITIES.": GOTO 6160
6200  PRINT
6210  IF KK = MC + 1 THEN GOTO 6530
6220  IF KK = MC + 2 THEN GOTO 6720
6230  IF KK < = MC THEN PRINT "IF YOU WANT TO CHANGE RHS."
6240  PRINT "THE ANSWER TO 'WHAT VARIABLE' IS"
6250  PRINT "RHS."
6260  PRINT
6270  INPUT "WHAT VARIABLE? ";JJ$
6280  PRINT "WHAT IS NEW VALUE OF ";JJ$: INPUT JJ: PRINT
6290  IF JJ$ = "RHS" THEN A(KK,NU + 2 * MC + 1) = JJ: GOTO 6910
6300  IF LEFT$(JJ$,1) = "X" THEN GOTO 6350

224
LGP Macro Computer Program

6310 IF LEFTS (JJS$1) = "D" AND RIGHTS (JJS$1) = "+" THEN
GOTO 6410
6320 IF LEFTS (JJS$1) = "D" AND RIGHTS (JJS$1) = "-" THEN
GOTO 6470
6330 PRINT "DIDN'T ANSWER WITH XI, DI+, DI- OR"
6340 PRINT "RHS. TRY AGAIN.": GOTO 6270
6350 IF LEN (JJS$) = 2 THEN JJS$ = RIGHTS (JJS$1)
6360 IF LEN (JJS$) = 3 THEN JJS$ = RIGHTS (JJS$2)
6370 J = VAL (JJS$)
6380 IF J > NU THEN PRINT "NO SUCH VARIABLE. TRY AGAIN.":
GOTO 6270
6390 A(KK,J) = JJS$
6400 GOTO 6910
6410 IF LEN (JJS$) = 4 THEN JJS$ = MID$ (JJS$2,2)
6420 IF LEN (JJS$) = 3 THEN JJS$ = MID$ (JJS$2,1)
6430 J = VAL (JJS$)
6440 IF J > MC THEN PRINT "NO SUCH DEV. VAR. TRY AGAIN.":
GOTO 6270
6450 A(KK,J + NU + MC) = JJS$
6460 GOTO 6910
6470 IF LEN (JJS$) = 4 THEN JJS$ = MID$ (JJS$2,2)
6480 IF LEN (JJS$) = 3 THEN JJS$ = MID$ (JJS$2,1)
6490 J = VAL (JJS$)
6500 IF J > MC THEN PRINT "NO SUCH DEV. VAR. TRY AGAIN.":
GOTO 6270
6510 A(KK,J + NU) + JJS$
6520 GOTO 6910
6530 INPUT "WHAT DEV. VAR.'S PRIORITY? ";JJS$
6540 PRINT "WHAT IS NEW PRIORITY FOR "JJS$": INPUT JJ
6550 IF LEFTS (JJS$1) < > "D" THEN PRINT "NEED A D TO PROCESS.";
GOTO 6530
6560 IF RIGHTS (JJS$1) = "+" THEN GOTO 6580
6570 IF RIGHTS (JJS$1) = "-" THEN GOTO 6630
6580 IF LEN (JJS$) = 4 THEN JJS$ = MID$ (JJ HGR ,2,2)
6590 IF LEN (JJS$) = 3 THEN JJS$ = MID$ (JJS$2,1)
6600 J = VAL (JJS$)
6610 C(J + NU + MC) = JJS$
6620 GOTO 6670
6630 IF LEN (JJS$) = 4 THEN JJS$ = MID$ (JJS$2,2)
6640 IF LEN (JJS$) = 3 THEN JJS$ = MID$ (JJS$2,1)
6650 J = VAL (JJS$)
6660 C(J + NU) = JJS$
6670 PRINT
6680 INPUT "ANOTHER PRIORITY? ";Q$
6690 IF Q$ = "Y" THEN GOTO 6530
6700 IF Q$ = "N" THEN GOTO 7030
6710 PRINT "Y OR N ONLY. TRY AGAIN.": GOTO 6680
6720 INPUT "WHAT DEV. VAR.'S WEIGHT? ";JJS$

225
LINEAR GOAL PROGRAMMING

6730 PRINT "WHAT IS NEW WEIGHT FOR "; JJ$; INPUT JJ
6740 IF LEFT$(JJ$,1) < > "D" THEN GOTO 6770
6750 IF RIGHT$(JJ$,1) = "+" THEN GOTO 6770
6760 IF RIGHT$(JJ$,1) = "-" THEN GOTO 6820
6770 IF LEN (JJ$) = 4 THEN JJ$ = MIDS (JJ$,2,2)
6780 IF LEN (JJ$) = 3 THEN JJ$ = MIDS (JJ$,2,1)
6790 J = VAL (JJ$)
6800 WC(J + NU + MC) = JJ
6810 GOTO 6860
6820 IF LEN (JJ$) = 4 THEN JJ$ = MIDS (JJ$,2,2)
6830 IF LEN (JJ$) = 3 THEN JJ$ = MIDS (JJ$,2,1)
6840 J = VAL (JJ$)
6850 WC(J + NU) = JJ
6860 PRINT
6870 INPUT "ANOTHER WEIGHT? ";Q$
6880 IF Q$ = "Y" THEN GOTO 6720
6890 IF Q$ = "N" THEN GOTO 7100
6900 PRINT "Y OR N ONLY. TRY AGAIN."; GOTO 6870
6910 PRINT
6920 INPUT "ANOTHER VALUE, SAME EQUATION? ";Q$
6930 IF Q$ = "Y" THEN GOTO 6270
6940 IF Q$ = "N" THEN GOTO 6960
6950 PRINT "Y OR N ONLY. TRY AGAIN."; GOTO 6920
6960 PRINT D$"OPEN";PR$;"L300"
6970 PRINT D$"WRITE";PR$;"R";KK
6980 FOR I = 1 TO NU + 2 * MC
6990 PRINT A(KK,I)
7000 NEXT I
7010 PRINT D$"CLOSE";PR$
7020 GOTO 7160
7030 PRINT D$"OPEN";PR$;"L300"
7040 PRINT D$"WRITE";PR$;"R";KK
7050 FOR I = 1 TO NU + 2 * MC
7060 PRINT C(I)
7070 NEXT I
7080 PRINT D$"CLOSE";PR$
7090 GOTO 7160
7100 PRINT D$"OPEN";PR$;"L300"
7110 PRINT D$"WRITE";PR$;"R";KK
7120 FOR I = 1 TO NU + 2 * MC
7130 PRINT WC(I)
7140 NEXT I
7150 PRINT D$"CLOSE";PR$
7160 PRINT
7170 INPUT "ANY MORE CHANGES? ";Q$
7180 IF Q$ = "Y" THEN GOTO 6160
7190 IF Q$ = "N" THEN RETURN
7200 PRINT "Y OR N ONLY. TRY AGAIN."; GOTO 7170

53
Appendix C - Computer Printout
<table>
<thead>
<tr>
<th>COEFFICIENTS IN TABLEAU</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1- 10 3 5 1 2 9 2 3 1 13</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>D2- 10 3 4 2 6 8 5 3 2 4 0 1 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>D3- 10 4 3 4 1 2 3 3 4 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>D4- 10 0 7 8 1 9 4 8 9 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

| 55 |
VALUES IN ZJ-CJ:

<table>
<thead>
<tr>
<th>P4</th>
<th>191</th>
<th>23</th>
<th>23</th>
<th>31</th>
<th>17</th>
<th>12</th>
<th>16</th>
<th>21</th>
<th>24</th>
<th>8</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>P2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>P1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

PIVOT COLUMN = 3
PIVOT ROW = 1

COEFFICIENTS IN TABLEAU:

<table>
<thead>
<tr>
<th>ZS</th>
<th>1.111111111</th>
<th>0.333333333</th>
<th>0.333333333</th>
<th>0.111111111</th>
<th>0.111111111</th>
<th>0.111111111</th>
<th>0.111111111</th>
<th>0.111111111</th>
</tr>
</thead>
<tbody>
<tr>
<td>D2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D5</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D7</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

(55)
<table>
<thead>
<tr>
<th>Row</th>
<th>Values in IJ-CJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1.33333333 8.2 0.66666667 3.66666667 10.1 1.33333333 3.33333333</td>
</tr>
<tr>
<td>P2</td>
<td>0.66666667 9.0 3.33333333</td>
</tr>
<tr>
<td>P3</td>
<td>0.0 0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td>P4</td>
<td>0.0 0.0 0.0 0.0 0.0</td>
</tr>
</tbody>
</table>

**Coefficients in Tableau:**

<table>
<thead>
<tr>
<th>Row</th>
<th>Values in IJ-CJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>X5</td>
<td>1.33333333 3.33333333</td>
</tr>
<tr>
<td>X1</td>
<td>0.0 0.0 0.0 0.0 0.0</td>
</tr>
</tbody>
</table>

**Values in IJ-CJ:**

<table>
<thead>
<tr>
<th>Row</th>
<th>Values in IJ-CJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>0.0 0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td>X5</td>
<td>1.33333333 3.33333333</td>
</tr>
</tbody>
</table>

...
**Coefficients in Tableau:**

<table>
<thead>
<tr>
<th></th>
<th>X3</th>
<th>X5</th>
<th>X2</th>
<th>X4</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
<th>X11</th>
<th>X12</th>
<th>X13</th>
<th>X14</th>
<th>X15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>4</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>5</td>
<td>-4</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>6</td>
<td>-5</td>
<td>-4</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>7</td>
<td>-6</td>
<td>-5</td>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>8</td>
<td>-7</td>
<td>-6</td>
<td>-7</td>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>9</td>
<td>-8</td>
<td>-7</td>
<td>-8</td>
<td>-7</td>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>10</td>
<td>-9</td>
<td>-8</td>
<td>-9</td>
<td>-8</td>
<td>-7</td>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>11</td>
<td>-10</td>
<td>-9</td>
<td>-10</td>
<td>-9</td>
<td>-8</td>
<td>-7</td>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>12</td>
<td>-11</td>
<td>-10</td>
<td>-11</td>
<td>-10</td>
<td>-9</td>
<td>-8</td>
<td>-7</td>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>13</td>
<td>-12</td>
<td>-11</td>
<td>-12</td>
<td>-11</td>
<td>-10</td>
<td>-9</td>
<td>-8</td>
<td>-7</td>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
</tr>
<tr>
<td>14</td>
<td>-13</td>
<td>-12</td>
<td>-13</td>
<td>-12</td>
<td>-11</td>
<td>-10</td>
<td>-9</td>
<td>-8</td>
<td>-7</td>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>15</td>
<td>-14</td>
<td>-13</td>
<td>-14</td>
<td>-13</td>
<td>-12</td>
<td>-11</td>
<td>-10</td>
<td>-9</td>
<td>-8</td>
<td>-7</td>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
</tr>
</tbody>
</table>

**Values in ZJ-Cj:**

<table>
<thead>
<tr>
<th></th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
<th>P8</th>
<th>P9</th>
<th>P10</th>
<th>P11</th>
<th>P12</th>
<th>P13</th>
<th>P14</th>
<th>P15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>14</td>
<td>28</td>
<td>62</td>
<td>0</td>
<td>22</td>
<td>114</td>
<td>18</td>
<td>-30</td>
<td>0</td>
<td>-9</td>
<td>-70</td>
<td>-70</td>
</tr>
<tr>
<td>1</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>58</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Solution Variables Are:**

<table>
<thead>
<tr>
<th></th>
<th>X3</th>
<th>X5</th>
<th>X2</th>
<th>X4</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
<th>X11</th>
<th>X12</th>
<th>X13</th>
<th>X14</th>
<th>X15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
</tr>
<tr>
<td>2</td>
<td>-30</td>
<td>-30</td>
<td>-30</td>
<td>-30</td>
<td>-30</td>
<td>-30</td>
<td>-30</td>
<td>-30</td>
<td>-30</td>
<td>-30</td>
<td>-30</td>
<td>-30</td>
<td>-30</td>
<td>-30</td>
</tr>
<tr>
<td>3</td>
<td>-70</td>
<td>-70</td>
<td>-70</td>
<td>-70</td>
<td>-70</td>
<td>-70</td>
<td>-70</td>
<td>-70</td>
<td>-70</td>
<td>-70</td>
<td>-70</td>
<td>-70</td>
<td>-70</td>
<td>-70</td>
</tr>
<tr>
<td>5</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
</tr>
</tbody>
</table>

58
UNACHIEVED GOALS ARE:

PA -14
END
DATED
FILM
8-88
D TIC