T-3991

A Geometric Programming Based Method for Determining the Required Constraints in a Class of Chemical Blending Problems

by

Owen S. Dull

T-3991

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Mineral Economics).

Golden, Colorado

Date 4 APR 91

Signed: Owen S. Dull
Owen S. Dull

Approved: R.E.D. Woolsey
Dr. R.E.D. Woolsey
Thesis Advisor

Golden, Colorado

~~Golden, Colorado~~ 4-4-91

Date 4-4-91

John E. Tilton
Dr. John E. Tilton
Professor and Head
Mineral Economics Department

ABSTRACT

An alternative algorithm to current linear programming methods is presented to solve petrochemical blending problems. The method, demonstrated on a class of such problems, shows that the solution time of these problems can be reduced significantly. Nonlinear chemical blending problems can also be solved with this new approach. It is shown that liquid fuel blending profits can be improved through use of the algorithm.

The method is evidence that geometric programming is a viable technique for solving specific linear programs. This means the method could be an alternative to the simplex algorithm. The algorithm can select from all possible constraints those which truly influence the problem, effectively reducing the size of these problems and decreasing the solution time. Use of the procedure assists the application of engineering to determine the optimal solutions to complex problems. We assert that this means the solution of these problems can often be reduced from a linear program to the inversion of a square matrix. In addition the technique is successfully extended to nonlinear blending problems of a type which have not been optimized before.

TABLE OF CONTENTS

page

ABSTRACT iii

ACKNOWLEDGMENTS vi

CHAPTER

1 INTRODUCTION 1

1.1 Operations Research in the Petroleum Industry 1
1.2 Solution of Linear Programs 4
1.3 Research at Colorado School of Mines 10
1.4 Geometric Programming Applied to Linear Programming 12

2 PROBLEM DEFINITION 15

2.1 Real Linear Programming Problems 15
2.2 Blending Problems 17
2.3 Nonlinear Blending Problems 20
2.4 A New Approach 21

3 THE TOOLS OF SOLUTION 22

3.1 The Calculus of Optimization 23
3.2 The Convexity of Linear Programs 27
3.2.1 Lines 27
3.2.2 Half-spaces 28
3.2.3 Intersections 29
3.2.4 Linear Equations 29
3.2.5 Weighted Averages 30
3.2.6 Convex Functions 31
3.2.7 Convex Programs 35
3.2.8 Linear Programs 40
3.3 The Rules of Geometric Programming 42
3.3.1 The General Arithmetic-Geometric Mean Inequality 43

3.3.2 Geometric Programming Without Constraints . . . 44
3.3.3 Geometric Programming With Constraints . . . 49
3.3.4 Condensation . . . 60
3.3.5 Surrogation . . . 61
3.3.6 The Advanced GP Sign Table . . . 62

4 THE PROCEDURE OF THE METHOD . . . 64

STEP I Examine the LP . . . 64
STEP II Formulate the GP . . . 65
STEP III Form the Advanced GP Sign Table . . . 66
STEP IV Determine Optimal Variable Values . . . 68
Flowchart of the Method . . . Pocket

5 EXAMPLE PROBLEMS . . . 70

5.1 Fertilizer Blending Problem . . . 70
5.2 Beer Blending Problem . . . 81
5.3 Aviation Gasoline Blending . . . 89
5.4 Refinery Optimization . . . 103
5.5 Nonlinear Gasoline Blending . . . 114

6 CONCLUSIONS AND AREAS FOR FURTHER RESEARCH . . . 133

6.1 Results and Conclusions . . . 133
6.2 Suggestions for Further Research . . . 134

REFERENCES CITED . . . 137

ACKNOWLEDGMENTS

I am grateful to the many people who kept faith with me throughout this project.

My parents probably wondered about the strange behavior of their boy but never let these concerns disturb the course of progress.

The relentless peer pressure of my fellow students kept me at the task. Their willingness to critically review each effort prevented many false starts.

Each member of my committee stood firm during my studies at Colorado School of Mines. Each in turn told me when I was wrong and found a way for me to recognize it. Prof. Donald I. Dickinson, Prof. William R. Astle, Dr. Thomas D. Kaufmann, Dr. Robert C. Drury, and Dr. Robert E.D. Woolsey, are teachers in the truest sense.

Finally, Dr. Woolsey's guild system provided the ready comradeship and environment I needed to accomplish this work.

Chapter 1

INTRODUCTION

This thesis is the result of a study to find new ways of solving existing problems. The expectation was that the new method would offer at least an alternative, if not an advantage, over existing methods. A serendipitous result to this successful effort is the ability to solve problems of a type that had previously not been solved by any known method. This success was achieved by developing a procedure to extend a part of the realm of nonlinear optimization to include a part of the realm of linear optimization.

1.1 Operations Research in the Petroleum Industry

Operations Research is defined as the application of scientific methods and techniques to decision-making problems. These problems result when there are two or more courses of action. The objective is to select the best alternative (Lesso 1982).

A common use of OR (Operations Research) is to maximize the utility of limited resources. The science was formally developed during the Second World War to fill this need. Because of its importance to the outcome of that conflict, the petroleum industry experienced some of these early efforts. The competitiveness of the world's Oil Industry has ensured that industries continued involvement in the development of the science ever since. From the start, the petroleum industry has been by far the most interested user of OR (Williams 1990).

Numerical optimization techniques are collectively known as **Mathematical Programming** (MP), which includes the fields of Linear Programming (LP), Nonlinear Programming (NLP), Integer Programming and Dynamic Programming. The practical use of MP began with two events, the development of the simplex algorithm by Dantzig in 1947 and commercial availability of electronic digital computers near the same time (Bodington and Baker 1990). The simplex algorithm is an efficient procedure for solving linear programs.

Algorithms for nonlinear problems came much later and are limited with regard to the types of problems they can handle. They are so limited that of necessity great efforts, and progress, have been made in the linearization of nonlinear problems in order to solve them by LP. These efforts continue today in the form of SLP (successive linear programming) and SQP (successive quadratic programming). These techniques replace the nonlinear functions being optimized with their Taylor series expansions, using the first two and three terms respectively (Baker and Lasdon 1985; Bodington 1979; Fletcher 1987; Gill, Murray, and Wright 1981; Prince and others 1983; Peressini, Sullivan, and Uhl 1988; Scales 1985).

A limitation of the available nonlinear techniques is their inability to handle *signomial* problems. Signomials are problems having negative coefficients in either or both the objective or constraint functions. These problems are of major concern to the petroleum industry. Many hydrocarbon components interact with nonlinear results. These results can be either gains or losses in various material properties. Examples of these properties are octane numbers, viscosities, flash points, aniline points, and vapor pressures (Gary and Handwerk 1984).

An important problem with both linear and nonlinear applications is blending. Blending problems occur when a number of components are mixed together to yield one or more products. Constraints on these problems are typically restrictions on the available quantities of raw materials, the quality of products, and the quantities of the products to be produced (Gass 1970). Particularly in the petroleum industry, these problems were the subject of some of the earliest efforts in the field of LP

(Symonds 1955). These problems become extremely large even when dealing with small to average sized facilities (Gass 1985; Wu and Coppins 1981; Woolsey 1987). Current efforts with these problems in the petroleum industry are concerned with the nonlinear type.

Most refinery feedstocks and intermediate streams are blended to a variety of finished products. Contracts and economics determine the desirability of the output product mix. Most economists are aware of the trade-offs between winter and summer when it comes to the gasoline or heating oil choices of refinery managers. The objective of product blending is to allocate the available blending components so that product specifications are achieved efficiently while producing a product mix that maximizes profit (Gary and Handwerk 1984). The volumes of refined products sold are so large that savings of even a fraction of a cent per gallon will result in substantial profit gains over a year (Klingman and others 1987).

1.2 Solution of Linear Programs

The purpose of LP is to find a maximum or a minimum value of a linear function. The subject is relevant only when the function is defined within a linear convex set (Gass 1985). The subject was essentially created in 1947. At that time Dantzig defined its scope and presented the first method for the practical solution of LP problems. This was termed the simplex algorithm. It is still the most versatile and widely used method (Shamir 1987).

An LP consists of an objective function and a set of constraints. Three possible outcomes are possible. The program can be infeasible, it can be feasible with an unbounded optimal value, or it can be feasible with an finite optimal value obtained at an optimal point. The simplex algorithm determines, in a finite number of iterations, which of these three cases holds. It will return an optimal point if one exists (Ecker and Kupferschmid 1988).

A matrix consisting of all the problem variables, the objective function and constraints is the starting point of the simplex algorithm. One artificial variable is added to the problem for each constraint. The simplex algorithm uses an initial solution consisting of all the artificial variables and none of the real variables. The current iterations working solution is known as the ***basis***. Each iteration consists of determining which real variable, by entering the basis, results in the best improvement of the objective function value. The constraint that most limits that variable determines which artificial or real variable leaves the basis. A ***basic infeasible solution*** means at least one artificial variable is still in the basis. A ***basic feasible solution*** means the current basis consists of real variables only. Each iteration results in an improvement or no change in the objective function. When no variables can enter the basis the algorithm is finished (Wessels 1988; Winston 1987).

Some LP problems exhibit special structures. Specific algorithms have been developed to take advantage of these structures. Examples of these special structures are the network, assignment, transportation, and transshipment problems. The specific algorithms for these problems are more efficient than simplex for their unique cases. They are not applicable to the general LP (Foulds 1984; Wu and Coppins 1981; Maurer 1990; Woolsey 1987).

The 1947 version of the simplex algorithm has been continuously improved. Current versions include routines to take advantage of concepts such as duality and generalized upper bounds. The newer routines either reduce the initial size of the problem or help the algorithm make greater improvements with each iteration. The basic procedure remains the same.

Typical LP problems contain thousands of variables and constraints. The upper limit on the number of basic feasible solutions the simplex algorithm will need to examine to solve a problem has been mathematically proven (Winston 1987). This limit is

$$\binom{n+m}{m} = \frac{(n+m)!}{n!m!}$$

where n and m are the number of real and artificial variables respectively. Normally the number of iterations required is a much smaller number than this upper limit. Typically from one to three times the number of constraints is a good approximation of the iterations required (Gass 1985; Ho and Luete 1983).

The proven upper limit on iterations required is a nonlinear function of the problem size. Valiant efforts have been directed at developing algorithms that would converge to a solution in a smaller number of iterations. The goal has been an algorithm with an upper limit on iterations required that would be a linear function of the problem size. Examples of these methods are

Khatchian's method and Karmarkar's method. Both methods can be shown to have an upper limit on iterations which is a linear function of problem size. Experience has shown that Khatchian's method nearly always requires the upper limit of iterations regardless of problem type. This makes it noncompetitive with simplex in typical problems (Bland, Goldfarb, and Todd 1981). The jury is still out on Karmarkar's method (Winston 1987). Efforts to develop these algorithms continue today (Dyer 1984; Packel 1988). The geometry of these solution techniques has led to them being referred to as interior point methods. In contrast, the simplex method interates along the outer edge of the feasible region.

The reason for these efforts are the limitations of computing equipment to solve useful problems. The limitation to LP progress has always been either the matrix size capacity of the computers, or the time to get a solution, or both. Typical problems form a matrix much larger than what can be stored in the active memory of any computer. Dantzig-Wolfe and Benders decomposition procedures, which are essentially each others dual, help expand the problem-solving capabilities of computers. They are an integral part of the idea of achieving overall optimization by coordinating several subproblem optimizations (Bodington and Baker 1990; Nazareth 1987). It is not unknown for a computing system's mean time between failures to be less than the expected solution time of these problems (Gass 1985; Klingman, Mote, and Phillips 1988; Woolsey 1989).

The mechanics of decomposition require that the problem matrix be broken down into several smaller matrices. Each smaller matrix is then optimized separately before they are all re-combined to see the final optimum solution. This means the greater part of the problem is kept on disk while a smaller subproblem is optimized in the active memory of the computer. This solution is not without its own drawbacks. Active computer memory works in nanoseconds: 10^{-9}sec, disk access across a bus works in microseconds: 10^{-6}sec, and disks work in milliseconds: 10^{-3}sec. This means there is a difference of six orders of magnitude in the speed at which the components of a computer system work on a specific problem. This difference in speed can be visualized by taking one second to reach across a desk for a table of information; six orders of magnitude greater would be waiting eleven and a half days for the same table to arrive. Current literature in this field discusses the application of decomposition techniques to solve large problems. Results are with few exceptions reported in cpu (central processing unit) seconds which are equivalent to active memory speeds. Typically, neither total solution or disk access time is not discussed (Aronson, Morton, and Thompson 1985; Ho and Luete 1983; Kingman, Mote, and Phillips 1988; Buchanan and others 1990).

Outside of matrix algebra techniques used to enhance computer performance when dealing with large matrices, little work has been done to reduce problem size (Gass 1985). The Operations Research Guild at the Colorado School of Mines has made progress in this area (Treece 1978). This thesis continues the effort.

1.3 Research at Colorado School of Mines

This thesis represents a continuation of the ongoing research program at Colorado School of Mines in the development and application of operations research techniques. Specifically in the area of this work, theses have been done showing the application of LP to blending problems in which it had not been previously used (Van Drew 1985; Gordon 1981). This work is the first application of nonlinear programming techniques to blending problems done at this school.

Under the directorship of Dr. Gene Woolsey, the Operations Research Guild at CSM has developed into a world center for the development of techniques in **Geometric Programming**. GP is a subject of nonlinear programming, particularly useful in dealing with problems of engineering design.

Examples of efforts in this area are the attacks on: the Simultaneous Nonlinear Equation Problem (Allen 1980; Baker 1980), Reliability Problems (Oatney 1987; Wilkinson 1987; Wilkinson 1989), Mineral Extraction (Taylor 1980), Bridge Design (Bailey 1989), Chemical Equilibrium (Wall 1984), Investment Decision (Grange 1977), and Reactor Design (Grange 1977).

Considerable work has been devoted to expanding the subject area and developing algorithms to handle general types of problems. These efforts are represented by: a proof of convergence of the condensation principle (Greening 1982), condensation techniques for posynomials (Ratliff 1986), condensation techniques for signomials (Thome 1988), preprocessing constrained signomials (Kirk 1988), condensation techniques for constrained signomials (Wessels 1989).

Each of these GP efforts has been directed at nonlinear problems exclusively. LP problems can be thought of as NLP problems with special structure; namely those NLP problems which are restricted to linear objective functions and linear constraints (Ecker and Kupferschmid 1988). This is one of the primary thrusts of this thesis.

1.4 Geometric Programming Applied to Linear Programming

In his work concerning simultaneous nonlinear equations Allen (1980) notes, under areas for further research, that the method of GP could work on LP problems. He believes this because GP can condense both negative and positive coefficients and thus approach the solution from both sides.

Early in the development of geometric programming the relationships between GP and LP problem formulations were defined using logarithms (Duffin, Peterson, and Zener 1967). These definitions showed that any primal LP could be transformed to a primal GP. The same problem could then be put in the dual GP form and transformed back to the correct dual LP form of the same problem. Two specific techniques of reformulating LP's into GP's are described in the mathematical appendix by Peterson of "Design Optimization" (Avriel, Rijckaert, and Wilde 1973). Peterson describes a third method (Peterson 1976), but notes none of the three has ever been applied to anything but teaching. A recent effort in mathematical programming placed emphasis on parallel processing (Han and Lou 1988). This particular work used **Quadratic Programming**, which can be done in parallel, to solve a small LP. It was earlier shown that QP (quadratic programming) could be useful in describing the sensitivity of LP solutions (Mangasarian and Meyer 1979). Another application (Jefferson 1985) showed that Quadratic Programs could be solved with GP. In combination these efforts show that the foundations for solving LP's with GP are in place.

The rules of GP (Duffin, Peterson, and Zener 1967; Woolsey 1969; Peressini, Sullivan, and Uhl 1988) are derived under mathematical conditions which require convex sets. This requirement is also present in any LP. The greatest obstacle to immediate application of GP to these problems is the feasible range of the variables. GP variables are defined over the range: $X_i > 0$. LP variables are defined over the range: $X_i \geq 0$. LP variables are allowed to take on the value of 0. This is not allowed for GP variables. This thesis presents a solution to this problem.

Thorough discussions of the techniques of geometric programming can be found in the literature (Avriel and Williams 1970; Ecker 1980; Rijckaert and Walraven 1985; Beightler and Phillips 1976; Peterson 1976; Beightler, Phillips, and Wilde 1979; Woolsey 1969; Woolsey, Kochenburger, and Linck 1971; Woolsey 1988). A particularly good treatment is contained in the thesis by Kirk (1988).

Chapter 2

PROBLEM DEFINITION

Mathematical programming is different from computer programming. It is a planning tool. The effective use of this tool returns an insight to the ***process*** being studied. Use of the tool requires the process be presented as a well formulated mathematical problem. Such a formulation is called a ***model***, defined as follows by Denn (1986):

> *A mathematical model of a process is a system of equations whose solution, given specified input data, is representative of the response of the process to a corresponding set of inputs. . . .*
>
> *The reason for obtaining a mathematical model is to enable computation of the expected behavior of the process for a range of inputs and conditions.*

2.1 Real Linear Programming Problems

Linear Programming refers to those situations which can be described with systems of linear equations. The purpose is to determine the best combination of controllable factors. Whatever limits more of a desirable factor is a constraint on the process.

Typical linear models are large, averaging about 2,500 constraints and 3,500 variables. Very large problems might have 10,000 to 20,000 constraints and 100,000 to a million variables. In practice, very large problems are not routinely solved because of the technology constraints previously described.

The very small examples used to teach LP do not impart the intricacies of solving a real problem. Effective problem formulation, data acquisition and input, algorithm execution, and analysis of real problems requires much more than the simplex algorithm. Consequently, ideas from the forefront of many fields, linear programming, computer programming, numerical analysis, data preparation and storage, to name a few, are implemented together in a **mathematical programming system** (MPS).

The key to the efficiency of a modern MPS lies in the structure of most real LP problems. They are ***sparse***. A matrix is termed sparse if the number of nonzero elements in it is small, less than 5%. In most of these problems a typical variable occurs, on average, in six constraints. In fact, most large-scale problems exhibit super-sparsity, which means they are less than 1% dense (Gass 1985). Sparseness allows the use of MPS features such as data packing, value tables, row and column indices, and pointers. These features allow storage of the problem in a much smaller area of memory.

When sparseness can be maintained throughout the simplex process, a relatively quick solution is achieved, assuming a true LP without any integer variables. The problem is in maintaining sparseness. Many problem structures exhibit what is termed "filling-in" while being processed by the simplex algorithm. This means the problem matrix becomes progressively more dense as the algorithm proceeds. The phenomena is similar in result to inverting a sparse matrix. The inverse may be anything but sparse. Typical causes of the problem are the relationships that link constraints in some classes of problems.

2.2 Blending Problems

A blending problem can always be recognized by its blending constraints. In its most basic form the problem will be constituted by blending constraints solely (see example no. 1).

Blending constraints are formed around a particular quality desired in the product. Each material input that can be used in the manufacture of the product will have this quality to some degree, but probably not the degree desired. Two or more different inputs can be combined in specific proportions to achieve the desired product quality. These proportions or ratios are what form blending constraints.

In a linear model the denominator of the ratio is multiplied through the constraint, one side of the equation is then subtracted from the other to form a standard LP blending constraint. The standard blending constraint can be an equality, or an equal to or greater than inequality, or an equal to or less than inequality. It will have a right hand side that is zero, and a left hand side that must have at least one positive coefficient term and one negative coefficient term.

In many engineering process operations it is possible to manufacture a product or products by several paths. In the petroleum industry a number of basic blending stocks are manufactured and a number of products are produced by blending them. The products will have material property specifications which must be met. Some of these will be maximum specs and some will be minimum specs. Each blending stock will have these properties in definite amounts. Each stock will also have a different cost from the other stocks. The problem is, how much of each blending stock should be mixed into the final product so that all the specifications are met? The key word is "met" because, since some specs are max and some min, the product can be produced in a number of ways at different costs (Amundson 1966).

In the terminology of mathematical programing these property specifications are known as quality constraints. As previously described there are input (blending stock) quantity constraints, and output (product) quantity constraints.

A well-formulated problem, meaning a good model of a refining operation, will contain "tie-in equations." Typically these equations are related by the same production facility or time period (Gass 1970). It is these relationships that lead to "filling-in." That is the bad news; the good news is, these same relationships are what allows the use of decomposition techniques.

Imbedded within these blending problems are several other well known types of LP's. Transhipment, transportation, and assignment types are among these. These imbedded problems, and their linkages to other parts of the overall problem, are the reason behind the large size of these problems.

2.3 Nonlinear Blending Problems

Constraints that require the use of a certain ingredient if a prescribed amount of another stock is used lead to integer variables. Once an LP contains integer variables it is termed a mixed integer problem and is no longer linear. It is sometimes still possible to solve these problems on an LP code using the simplex algorithm. The solution time of these problems is exponential in comparison to the true LP. Imbedded scheduling problems may also lead to integer variables within the overall optimization.

As noted in chapter 1, some hydrocarbon material properties do not blend linearly. Optimization of these problems has been carried out by linearizing segments of each material's blending functions. This has not always proven a satisfactory solution. Usually the reason is that the number of segments required makes the problem too large to handle.

The nonlinear techniques available have been limited by several factors. They are not as efficient as the simplex algorithm; they take longer to solve problems of a similar size. They are not global optimizers. This is because a nonlinear optimization problem has a fourth possibility besides those available to a linear problem. This other possibility is that the problem can be feasible with a bounded objective function value but no optimal point exists (Ecker and Kupferschmid 1988). This results in situations which present local optimal points. The nonlinear algorithms locate these local optima but cannot find their way out unless given a more fortunate starting condition.

2.4 A New Approach

The techniques of geometric programming have been highly successful when applied to problems of engineering design. Optimization of a process leads to the best design. Hence, it would seem, the techniques of GP are well suited to process optimization.

This is the case. GP can be used to identify the constraints which actively affect a problem at any given time in its solution. The technique has rules which allow solutions to be approached iteratively when necessary. GP formulations present a different and more illustrative presentation of the models typical to this class of problems. The greatest test is to identify when a variable is going to zero and to reformulate accordingly.

Chapter 3

THE TOOLS OF SOLUTION

The next three sections show the common ground of LP and GP. The mathematical basis for applying the techniques of GP to LP problems is defined. Readers not inclined to these proceedings should go on to section 3.4.

Inspiration for this tie-in of fields was a result of classes in LP and GP at Colorado School of Mines by G. J. Wessels and R. E. D. Woolsey in 1988. Much of the next two sections is taken directly from those courses. Other material is taken from two references in the field: *Geometric Programming: Theory & Application* by Duffin, Peterson, and Zener (1967); and *The Mathematics of Nonlinear Programming* by Peressini, Sullivan, and Uhl (1988).

It is important to know why these problems are so complex. The reason is that any feasible problem, except for the pure integer problem, has an infinite number of solutions that work. We are interested in the best solution. Once the active constraints have been identified, they define the feasible region of the problem's solutions. Any point within that region constitutes a viable situation, entirely legal. A student of these problems knows that the best solution lies on the border of the feasible region. We have to model the problem with an accuracy compatible with what can physically be done to implement the solution in practice.

The next three sections will show that LP and GP problems are congruent mathematical programs. As a part of this demonstration, the four fundamental rules of GP are derived. These rules are essential to any application of GP. The reason for this exercise is that no direct application of NLP applied to LP was found in our literature search. Therefore, these facts of optimization are presented together so that the reader may have direct access to the underlying principles of this method.

3.1 The Calculus of Optimization

Students of functional optimization should be familiar with the concepts of necessary and sufficient conditions. A necessary condition C_n for a statement S implies that C_n must be true for S to be true (S only if C_n). A sufficient condition C_s for a statement S implies that S is true when C_s is true (S if C_s). If a condition C_{ns} is both necessary and sufficient for S, then we can say S if and only if C_{ns}.

These conditions are sometimes referred to as first and second order conditions. This terminology comes from the concepts of calculus as applied to maxima and minima. Commonly a necessary condition involves the first-order derivatives. Also the sufficient conditions will involve the second-order derivatives. Do not treat these as synonyms. In particular the second-order conditions alone are not sufficient for an extremum (Koo 1977).

Stated formally, from calculus (this restricts us to functions for which the first and second derivative exists), for S to be a statement that $f(x)$ attains a local minimum at $x = x_0$, it is necessary that $f'(x) = 0$ at $x = x_0$. It is sufficient that $f'(x_0) = 0$ and $f''(x_0) > 0$. A local maximum requires the inversion of the inequality.

The following theorem from calculus is particularly useful to optimization theory:

(Extended Law of the Mean) Theorem. Given that $f(x), f'(x)$, *and* $f''(x)$ *exist on the closed interval* $[a,b] = \{x \in R : a \leq x \leq b\}$. *If* $\mathbf{x}_0$ *and* $\mathbf{x}$ *are any two points on the interval then there exists*

a point z *between* x_0 *and* x *such that*

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(z)}{2}(x - x_0)^2.$$

This is a special case of the Taylor Formula from calculus. Now if $f(x)$ is a function where $f''(x)$ is positive for all real x, and x_0 is a point where $f'(x_0) = 0$, then the above formula results in

$$f(x) = f(x_0) + 0 + a\ positive\ number$$

for all real numbers $x \neq x_0$. Hence $f(x) > f(x_0)$ for all $x \neq x_0$. Thus x_0 is the point which minimizes the value of $f(x)$ (Peressini, Sullivan, and Uhl 1988). This is essentially the second derivative test from calculus.

The following theorem is used to define terminology which will be used later in this section.

Theorem. Given that $f(x)$, $f'(x)$, and $f''(x)$ are all continuous on an interval I and that $x_0 \in I$ is a critical point of $f(x)$.

(a) If $f''(x) \geq 0$ for all $x \in I$, then x_0 is a global minimizer of $f(x)$ on I.

(b) If $f''(x) > 0$ for all $x \in I$ such that $x \neq x_0$, then x_0 is a strict global minimizer of $f(x)$ on I.

(c) If $f''(x_0) > 0$, then x_0 is a strict local minimizer of $f(x)$ (Peressini, Sullivan, and Uhl 1988).

These developments can be extended to functions of several variables by using vector notation and the following definitions. Given $f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n)$ is a function of n variables with continuous first and second partial derivatives on R^n. The *gradient* ∇f of $f(\mathbf{x})$ is the n-vector

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right).$$

The *Hessian* Hf of $f(\mathbf{x})$ is the symmetric $n \times n$-matrix

$$Hf = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}.$$

The multivariate Taylor's Formula follows:

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*) \cdot (\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*) \cdot Hf(\mathbf{z})(\mathbf{x} - \mathbf{x}^*).$$

Here $\mathbf{x}^*$ is used instead of x_0 to more impressively represent the optimum multivariate point we are searching for, rather than a zero point.

3.2 The Convexity of Linear Programs

The feasible region in all LP's and a large class of NLP's is defined by a ***convex set***. The definition of such sets is: If all two-point combinations within the boundary of an enclosed set can be connected by a straight line that is also entirely within the boundary, then that set is convex. All other cases are concave. Thus all individual points, all lines, and all triangles are convex.

The line segment [x,y] joining points x and y can be described mathematically by

$$[\mathbf{x},\mathbf{y}] = \{\mathbf{x} + \lambda(\mathbf{y} - \mathbf{x}) : 0 \leq \lambda \leq 1\}.$$

This set can also be described as

$$[\mathbf{x},\mathbf{y}] = \{\lambda\mathbf{y} + (1 - \lambda)\mathbf{x} : 0 \leq \lambda \leq 1\}.$$

Therefore, *a subset C of R^n is convex if and only if for every* x *and* y *in C and every* λ *with* $0 \leq \lambda \leq 1$, *the vector* $\lambda\mathbf{x} + (1 - \lambda)\mathbf{y}$ *is also in C* (Peressini, Sullivan, and Uhl 1988).

This is the mathematical equivalent of the definition given in the first paragraph of this subsection.

3.2.1 Lines

Lines in R^n can be described several different ways.

When x and v are vectors in R^n, *the line L through* x *in the* *direction of* $\mathbf{v}$ is described as $L = \{\mathbf{x} + \lambda\mathbf{v} : \lambda \in R\}$.

If $\mathbf{x}$ and $\mathbf{y}$ are vectors in R^n, the line L through $\mathbf{x}$ and $\mathbf{y}$ is the set $L = \{\lambda\mathbf{x} + (1-\lambda)\mathbf{y} : \lambda \in R\}$.

The important thing to notice is the similar structure of these lines and the mathematical definition of convexness. Clearly any line in R^n is a convex set (Peressini, Sullivan, and Uhl 1988).

3.2.2 Half-spaces

Given $\mathbf{x}^* \in R^n$ and $\alpha \in R$, then the *closed half-spaces*

$$F^+ = \{\mathbf{y} \in R^n : \mathbf{x}^* \cdot \mathbf{y} \geq \alpha\},$$

$$F^- = \{\mathbf{y} \in R^n : \mathbf{x}^* \cdot \mathbf{y} \leq \alpha\},$$

and the *open half-spaces*

$$G^+ = \{\mathbf{y} \in R^n : \mathbf{x}^* \cdot \mathbf{y} > \alpha\},$$

$$G^- = \{\mathbf{y} \in R^n : \mathbf{x}^* \cdot \mathbf{y} < \alpha\},$$

determined by $\mathbf{x}^*$ and α are all convex sets. Because $\mathbf{y}, \mathbf{z}$ are in F^+ and if $0 \leq \lambda \leq 1$, then

$$\mathbf{x}^* \cdot [\lambda\mathbf{y} + (1-\lambda)\mathbf{z}] = \lambda\mathbf{x}^* \cdot \mathbf{y} + (1-\lambda)\mathbf{x}^* \cdot \mathbf{z} \geq \lambda\alpha + (1-\lambda)\alpha = \alpha,$$

so that

$$\lambda\mathbf{y} + (1-\lambda)\mathbf{z} \in F^+.$$

The other cases can be shown the same way (Peressini, Sullivan, and Uhl 1988).

3.2.3 Intersections

If $C_1, C_2, \ldots, C_k, \ldots$ are convex sets in R^n, then the intersection $\cap C_i$ is also convex. To verify this: If $\mathbf{y}, \mathbf{z}$ belong to this intersection and if $0 \le \lambda \le 1$, then $\mathbf{y}, \mathbf{z}$ belong to each C_i, so $\lambda \mathbf{y} + (1-\lambda)\mathbf{z} \in C_i$ for each i since C_i is convex. But then $\lambda \mathbf{y} + (1-\lambda)\mathbf{z} \in \cap C_i$ so that $\cap C_i$ is convex.

This is important, it means the set formed by the intersection of convex sets is itself convex (Peressini, Sullivan, and Uhl 1988).

3.2.4 Linear Equations

Given $A = (a_{ij})$ is an $m \times n$-matrix and $\mathbf{b} \in R^m$, then the set S of all solutions $\mathbf{x} \in R^n$ of the system $A\mathbf{x} \le \mathbf{b}$ of linear inequalities:

$$\begin{pmatrix} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & \le & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & \le & b_2 \\ \cdot & & & & & & \cdot & & \cdot \\ \cdot & & & & & & \cdot & & \cdot \\ \cdot & & & & & & \cdot & & \cdot \\ a_{ml}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & \le & b_m \end{pmatrix}$$

is a convex set in R^n. This is fundamental to linear programming. In fact, if $\mathbf{a}^{(i)} = (a_{il}, a_{i2}, \ldots, a_{\in}) \in R^n$ is the ith row of A and if F_i^- is the half-space

$$F_i^- = \{\mathbf{x} \in R^n : \mathbf{a}^{(i)} \cdot \mathbf{x} \leq b_i\}$$

for $i = 1, \dots, m$, then the solution set S is the intersection of the half-spaces $F_1^-, F_2^-, \dots, F_m^-$. Thus the convexity of S follows from the above arguments concerning half-spaces and intersections (Peressini, Sullivan, and Uhl 1988).

3.2.5 Weighted Averages

If $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}$ are vectors in R^n, then a *weighted average* or *convex combination* of $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}$ is any vector

$$\lambda_1 \mathbf{x}^{(1)} + \lambda_2 \mathbf{x}^{(2)} + \cdots + \lambda_k \mathbf{x}^{(k)} = \sum_{i=1}^{k} \lambda_i \mathbf{x}^{(i)},$$

where $\lambda_1, \lambda_2, \dots, \lambda_k$ are nonnegative numbers which sum to one. If C is a convex set in R^n, then, by the definition of convexity, the set C contains any convex combination of *two* of its members. But note: If $\lambda_1, \lambda_2, \lambda_3$ are nonnegative numbers that sum to one and $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$ belong to C then,

$$\mathbf{x} = \lambda_1 \mathbf{x}^{(1)} + \lambda_2 \mathbf{x}^{(2)} + \lambda_3 \mathbf{x}^{(3)} = \sum_{i=1}^{3} \lambda_i \mathbf{x}^{(i)}$$

also belongs to C. This is clear when $\lambda_3 = 0$, since $\mathbf{x}$ is then a convex combination of *two* vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}$ in C. If $\lambda_3 \neq 0$, then

$$\mathbf{x} = \lambda_1 \mathbf{x}^{(1)} + (\lambda_2 + \lambda_3)\left[\frac{\lambda_2}{\lambda_2 + \lambda_3}\mathbf{x}^{(2)} + \frac{\lambda_3}{\lambda_2 + \lambda_3}\mathbf{x}^{(3)} \right].$$

Since

$$\frac{\lambda_2}{\lambda_2 + \lambda_3} + \frac{\lambda_3}{\lambda_2 + \lambda_3} = 1,$$

the expression in square brackets above is a convex combination of *two* vectors in C so it belongs to C as well. But then, since $\lambda_1 + (\lambda_2 + \lambda_3) = 1$, the same above equation shows that $\mathbf{x}$ is a convex combination of *two* vectors in C and so $\mathbf{x} \in C$.

The preceding argument shows that if C contains any convex combination of two of its vectors, then it must also contain any convex combination of three of its vectors (Pēressini, Sullivan, and Uhl 1988). This argument can be extended to any number of combinations required.

3.2.6 Convex Functions

A function $f(\mathbf{x})$ is defined as *convex* on C if

$$f(\lambda\mathbf{x} + [1-\lambda]\,\mathbf{y}) \le \lambda f(\mathbf{x}) + [1-\lambda] f(\mathbf{y})$$

for all $\mathbf{x}, \mathbf{y}$ in C and all λ with $0 \le \lambda \le 1$;

the function $f(\mathbf{x})$ is *strictly convex* on C if

$$f(\lambda\mathbf{x} + [1-\lambda]\,\mathbf{y}) < \lambda f(\mathbf{x}) + [1-\lambda] f(\mathbf{y})$$

for all $\mathbf{x}, \mathbf{y}$ in C with $\mathbf{x} \neq \mathbf{y}$ and all λ with $0 < \lambda < 1$. If the inequalities in the above definitions are reversed they define *concave* and *strictly concave* functions.

We now need some mathematical definitions for the theorems which follow.

Given that A is an $n \times n$-symmetric matrix and that $Q_A(\mathbf{y}) = \mathbf{y} \cdot A\,y$ is the quadratic form associated with A. The A and Q_A are called:

(a) positive semidefinite if $Q_A(\mathbf{y}) = \mathbf{y} \cdot A\,\mathbf{y} \geq 0$ for all $\mathbf{y} \in R^n$;

(b) positive definite if $Q_A(\mathbf{y}) = \mathbf{y} \cdot A\,\mathbf{y} > 0$ for all $\mathbf{y} \in R^n, \mathbf{y} \neq \mathbf{0}$;

(c) negative semidefinite if $Q_A(\mathbf{y}) = \mathbf{y} \cdot A\,\mathbf{y} \leq 0$ for all $\mathbf{y} \in R^n, \mathbf{y} \neq \mathbf{0}$;

(d) negative definite if $Q_A(\mathbf{y}) = \mathbf{y} \cdot A\,\mathbf{y} < 0$ for all $\mathbf{y} \in R^n, \mathbf{y} \neq \mathbf{0}$;

(e) indefinite if $Q_A(\mathbf{y}) = \mathbf{y} \cdot A\,\mathbf{y} > 0$ for some $\mathbf{y} \in R^n$ and $Q_A(\mathbf{y}) < 0$ for other $\mathbf{y} \in R^n$.

Theorem. *Given that* $\mathbf{x}^*$ *is a critical point of a function* $f(\mathbf{x})$ *with continuous first and second partial derivatives on* R^n *and that* $H f(\mathbf{x})$ *is the Hessian of* $f(\mathbf{x})$. *Then* $\mathbf{x}^*$ *is:*

(a) a global minimizer for $f(\mathbf{x})$ if $H f(\mathbf{x})$ is positive semidefinite on R^n;

(b) a strict global minimizer for $f(\mathbf{x})$ if $H f(\mathbf{x})$ is positive definite on R^n;

(c) a global maximizer for $f(\mathbf{x})$ if $H\,f(\mathbf{x})$ is negative semidefinite on R^n;

(d) a strict global maximizer for $f(\mathbf{x})$ if $H\,f(\mathbf{x})$ is negative definite on R^n.

Convex functions sum like convex sets intersect. Some useful theorems describing this follow:

(Convex 1) Theorem. *If $f(x)$ is a convex function defined on an open interval (a,b), then $f(x)$ is continuous on (a,b)* (Peressini, Sullivan, and Uhl 1988).

(Convex 2) Theorem. *Given that $f(\mathbf{x})$ is a convex function defined on a convex subset C of R^n. Given $\lambda_1, \lambda_2, \ldots, \lambda_k$ are nonnegative numbers which sum to one. Also given are $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(k)}$ points of C, then*

$$f\left(\sum_{i=1}^{k} \lambda_i \mathbf{x}^{(i)}\right) \le \sum_{i=1}^{k} \lambda_i f(\mathbf{x}^{(i)}).$$

(Peressini, Sullivan, and Uhl 1988).

(Convex 3) Theorem. *Any local minimizer of a convex function $f(\mathbf{x})$ defined on a convex subset C of R^n is also a global minimizer. Any local minimizer of a strictly convex function $f(\mathbf{x})$ defined on a convex set C in R^n is the unique strict global* *minimizer of $f(\mathbf{x})$ on C* (Peressini, Sullivan, and Uhl 1988).

(Convex 4) Theorem. *If $f(\mathbf{x})$ has continuous first partial derivatives on a convex set D in R^n. Then:*

(a) *the function $f(\mathbf{x})$ is convex if and only if*

$$f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot (\mathbf{y} - \mathbf{x}) \leq f(\mathbf{y})$$

for all $\mathbf{x}, \mathbf{y}$ in D:

(b) *the function $f(\mathbf{x})$ is strictly convex on D if and only if*

$$f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot (\mathbf{y} - \mathbf{x}) < f(\mathbf{y})$$

for all $\mathbf{x}, \mathbf{y}$ in D with $\mathbf{x} \neq \mathbf{y}$ (Peressini, Sullivan, and Uhl 1988).

(Convex 5) Theorem. *Given that $f(\mathbf{x})$ has continuous second partial derivatives on some open convex set C in R^n. If the Hessian $Hf(\mathbf{x})$ of $f(\mathbf{x})$ is positive semidefinite (respectively positive definite) on C, then $f(\mathbf{x})$ is convex (respectively strictly convex) on C* (Peressini, Sullivan, and Uhl 1988).

(Convex 6) Theorem.

(a) *Given $f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})$ are convex functions on a convex set C in R^n, then*

$$f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x}) + \cdots + f_k(\mathbf{x})$$

is convex. Also, if at least one $f_1(\mathbf{x})$ is strictly convex on C, then the sum $f(\mathbf{x})$ is strictly convex.

(b) *Given $f(\mathbf{x})$ is convex (respectively strictly convex) on a convex set C in R^n and in α is a positive number, then $\alpha f(\mathbf{x})$ is strictly convex.*

(c) *Given $f(\mathbf{x})$ is a convex (respectively strictly convex) function defined on a convex set C in R^n, and if $g(y)$ is an increasing (respectively strictly increasing) convex function defined on the range of $f(\mathbf{x})$ in R, then the composite function $g(f(\mathbf{x}))$ is convex (respectively strictly convex) on C* (Peressini, Sullivan, and Uhl 1988).

3.2.7 Convex Programs

The standard convex program is defined as follows: Given that $f(\mathbf{x}), g_1(\mathbf{x}), \dots, g_m(\mathbf{x})$ are real-valued functions defined on a subset C of R^n.

$$(P) \qquad \text{Minimize} \quad f(\mathbf{x})$$

$$\text{subject to}$$

$$g_1(\mathbf{x}) \le 0, \quad g_2(\mathbf{x}) \le 0, \dots, g_m(\mathbf{x}) \le 0,$$

$$\text{where} \quad x \in C \subset R^n.$$

The function $f(\mathbf{x})$ is called the *objective function of (P)* and the function inequalities $g_1(\mathbf{x}) \le 0, \dots, g_m(\mathbf{x}) \le 0$ are called the *inequality constraints for (P).* A point $\mathbf{x} \in C$ that satisfies all of the constraints of the program *(P)* is called a *feasible point for (P)*. The set F of all feasible points for *(P)* is the *feasibility region for (P)*. When the feasible region for *(P)* is not empty, we say that *(P)* is *consistent*. If there is a feasible point x for *(P)* such that $g_i(\mathbf{x}) < 0$ for $i = 1, \ldots, m$, then *(P)* is *superconsistent* and the point x is called a *Slater point* for *(P)*. Given *(P)* is a consistent program and $\mathbf{x}^*$ is a feasible point for *(P)* with $f(\mathbf{x}^*) \le f(\mathbf{x})$ for all feasible points x in *(P)*. Point $\mathbf{x}^*$ is then a *solution* to *(P)*

(P) is a convex program if the objective function, the constraint functions, and the underlying set C are all convex (Peressini, Sullivan, and Uhl 1988).

We now need two definitions to proceed. Given that $f(\mathbf{x})$ is a real-valued function defined on a subset C of R^n. If there is a smallest real number β such that $f(\mathbf{x}) \le \beta$ for all $\mathbf{x} \in C$, then β is called ***the supremum of*** $f(\mathbf{x})$ ***on*** C:

$$\sup_{\mathbf{x} \in C} f(\mathbf{x}) = \beta .$$

If there is a largest real number α such that $f(\mathbf{x}) \ge \alpha$ for all $\mathbf{x} \in C$, then α is called ***the infimum of*** $f(\mathbf{x})$ ***on*** C:

$$\inf_{\mathbf{x} \in C} f(\mathbf{x}) = \alpha .$$

Note that if $\mathbf{x}^*$ is a global maximizer of $f(\mathbf{x})$ on C;

$$\sup_{\mathbf{x} \in C} f(\mathbf{x}) = f(\mathbf{x}^*) .$$

Similarly, if $\mathbf{x}^*$ is a global minimizer of $f(\mathbf{x})$ on C;

$$\inf_{\mathbf{x} \in C} f(\mathbf{x}) = f(\mathbf{x}^*)$$

(Peressini, Sullivan, and Uhl 1988).

(Convex 7) Theorem. *If (P) is a convex program and if* $(P(\mathbf{z}))$ *is the perturbation of (P) by* $\mathbf{z} \in R^m$*, then the function* $MP(\mathbf{z})$ *is convex and its domain is a convex subset of* R^m*. If (P) is superconsistent, then* $\mathbf{0}$ *is an interior point of the domain of* $MP(\mathbf{z})$ (Peressini, Sullivan, and Uhl 1988).

(Convex 8) Theorem. *If (P) is a superconsistent convex program such that* $MP = MP(\mathbf{0})$ *is finite, then* $MP(\mathbf{z})$ *is finite on its entire domain and there exists a vector* $\vec{\lambda} \in R^m$ *such that* $\vec{\lambda} \geq \mathbf{0}$ *and*

$$MP(\mathbf{z}) \geq MP(\mathbf{0}) - \vec{\lambda} \cdot \mathbf{z}$$

for all $\mathbf{z}$ *in the domain of* $MP(\mathbf{z})$.

When this condition holds, $\vec{\lambda}$ is called the ***sensitivity vector*** for *(P)*. The theorem guarantees that superconsistent convex programs always have sensitivity vectors (Peressini, Sullivan, and Uhl 1988).

(Convex 9) Theorem. *Given that*

(*P*) $$\text{Minimize} \quad f(\mathbf{x})$$

$$\text{subject to}$$

$$g_1(\mathbf{x}) \le 0, \ldots, \quad g_m(\mathbf{x}) \le 0,$$

$$\text{where} \quad x \in C$$

is a convex program for which there is a sensitivity vector $\vec{\lambda}$. *Then*

$$MP = \inf_{\mathbf{x} \in C} \{ f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) \}.$$

The ***Lagrangian*** $L(\mathbf{x}, \vec{\lambda})$ of a convex program

(*P*) $$\text{Minimize} \quad f(\mathbf{x})$$

$$\text{subject to}$$

$$g_1(\mathbf{x}) \le 0, \ldots, \quad g_m(\mathbf{x}) \le 0,$$

$$\text{where} \quad x \in C$$

is the function defined by

$$L(\mathbf{x}, \vec{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x})$$

for $\mathbf{x} \in C$ and $\vec{\lambda} \ge \mathbf{0}$.

The Lagrangian $L(\mathbf{x}, \vec{\lambda})$ is a function of $m + n$ variables where m is the number of inequality constraints and n is the number of variables involved in the objective and constraint functions. If *(P)* is a superconsistent convex program such that *MP* is finite, then a sensitivity vector $\vec{\lambda}$ exists and

$$MP = \inf_{\mathbf{x} \in C} \{L(\mathbf{x}, \vec{\lambda})\}$$

by (Convex 9) (Peressini, Sullivan, and Uhl 1988).

(Convex 10) Theorem. *Given that (P) is a superconsistent convex program. Then* $\mathbf{x}^* \in C$ *is a solution of (P) if and only if there is a* $\vec{\lambda}^* \in R^m$ *such that:*

(1) $\vec{\lambda}^* \geq \mathbf{0}$;

(2) $L(\mathbf{x}^*, \vec{\lambda}) \leq L(\mathbf{x}^*, \vec{\lambda}^*)$ *for all* $\mathbf{x} \in C$ *and all* $\vec{\lambda} \geq \mathbf{0}$;

(3) $\lambda_i^* g_i(\mathbf{x}^*) = \mathbf{0}$ *for* $i = 1, 2, \ldots, m$.

This is the Saddle Point Form of the **Karush-Kuhn-Tucker** Theorem (Peressini, Sullivan, and Uhl 1988).

(Convex 11) Theorem. *Given that (P) is a superconsistent convex program such that the objective function* $f(\mathbf{x})$ *and the constraint functions* $g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})$ *have continuous first partial derivatives on the underlying set C for (P). If* $\mathbf{x}^*$ *is feasible for (P) and an interior point of C, then* $\mathbf{x}^*$ *is a solution of (P) if and only if there is a* $\vec{\lambda}^* \in R^m$ *such that:*

(1) $\lambda_i^* \geq 0$ *for* $i = 1, 2, \ldots, m$;

(2) $\lambda_i^* g_i(\mathbf{x}) = 0 \ i = 1, 2, \ldots, m$;

(3) $\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(\mathbf{x}^*) = \mathbf{0}$.

Which is the Gradient Form of the **Karush-Kuhn-Tucker Theorem** (Peressini, Sullivan, and Uhl 1988).

(Convex 12) Theorem. *Given that (P) is a convex program and that* $\mathbf{x}^*$ *is a solution of (P). If* $\vec{\lambda}^*$ *is a vector in* R^m *such that* $(\mathbf{x}^*, \vec{\lambda}^*)$ *satisfy the Karush-Kuhn-Tucker (K-K-T) Theorem conditions:*

(1) $\vec{\lambda}^* \geq \mathbf{0}$;

(2) $L(\mathbf{x}^*, \vec{\lambda}) \leq L(\mathbf{x}^*, \vec{\lambda}^*) \leq L(\mathbf{x}, \vec{\lambda}^*)$ *for all* $\mathbf{x} \in C$ *and all* $\vec{\lambda} \geq \mathbf{0}$;

(3) $\lambda_i^* g_i(\mathbf{x}^*) = 0$ *for* $i = 1, 2, \dots, m$;

then $\vec{\lambda}^*$ *is a sensitivity vector for (P); that is,*

$$MP(\mathbf{z}) \geq MP - \vec{\lambda}^* \cdot \mathbf{z}$$

for all $\mathbf{z}$ *in the domain of* $MP(\mathbf{z})$ (Peressini, Sullivan, and Uhl 1988).

3.2.8 Linear Programs

If we define that $\mathbf{u} \geq \mathbf{v}$ means $u_i \geq v_i$ for all i, then we can use vector notation to write the minimum standard form of a linear program in the following compact form: Given an $m \times n$-matrix A and vectors $\mathbf{b} \in R^n$, $\mathbf{c} \in R^m$:

$$(LP) \qquad \text{Minimize} \quad \mathbf{b} \cdot \mathbf{x}$$

subject to the constraints

$$A\mathbf{x} \geq \mathbf{c}, \quad \mathbf{x} \geq \mathbf{0}.$$

If the ith row vector of A is denoted by $\mathbf{a}^{(i)}$, then the constraints $A\mathbf{x} \geq \mathbf{c}$ can be written as

$$\mathbf{a}^{(i)} \cdot \mathbf{x} \geq c_i, \qquad i = 1, 2, \ldots, m.$$

The functions $f(\mathbf{x}) = \mathbf{b} \cdot \mathbf{x}$ and $g_i(\mathbf{x}) = c_i - \mathbf{a}^{(i)} \cdot \mathbf{x}$, $i = 1, 2, \ldots, m$, are linear and therefore convex. Also, the set $C = \{\mathbf{x} \in R^n : \mathbf{x} \geq 0\}$ is convex, so *(LP)* can be reformulated as a convex program as follows:

$$\text{Minimize} \quad f(\mathbf{x}) = \mathbf{b} \cdot \mathbf{x}$$

subject to the constraints

$$g_1(\mathbf{x}) = c_1 - \mathbf{a}^{(1)} \cdot \mathbf{x} \leq 0,$$

$$\cdot \quad \cdot$$

$$\cdot$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot$$

$$g_m(\mathbf{x}) = c_m - \mathbf{a}^{(m)} \cdot \mathbf{x} \leq 0,$$

$$\text{where} \qquad \mathbf{x} \in C = \{\mathbf{y} \in R : \mathbf{y} \geq \mathbf{0}\}.$$

Therefore every linear program is also a convex program (Peressini, Sullivan, and Uhl 1988).

3.3 The Rules of Geometric Programming

The basis of GP is the Arithmetic-Geometric (A-G) Mean Inequality. The most familiar form of this inequality is

$$\sqrt{x_1 x_2} \le \frac{1}{2}x_1 + \frac{1}{2}x_2,$$

where x_1 and x_2 are positive numbers. Equality exists in this equation only when $x_1 = x_2$. The left side of the above equation is the geometric mean and the right side is the arithmetic mean of x_1 and x_2.

A more general statement of this inequality asserts that

$$\sqrt[n]{x_1 x_2 \cdots x_n} \le \frac{1}{n}x_1 + \frac{1}{n}x_2 + \cdots + \frac{1}{n}x_n,$$

for $x_1, x_2, \ldots, x_n$ positive real numbers. Equality holds only when $x_1 = x_2 = \cdots = x_n$ (Peressini, Sullivan, and Uhl 1988).

Note that the exponents of the variables in the geometric mean are equal positive numbers that sum to one. These same exponents become the coefficients of each variable in the arithmetic mean. When referring to both means these same numbers are referred to as the weights of the associated variables.

3.3.1 The General Arithmetic-Geometric Mean Inequality

The general form of this inequality is found using the function

$$f(x) = -\ln x,$$

which is defined for $x > 0$ and is *strictly convex* since

$$f''(x) = 1/x^2 > 0.$$

If $x_1, x_2, \ldots, x_n$ and $\delta_1, \delta_2, \ldots, \delta_n$ are positive numbers such that

$$\delta_1 + \delta_2 + \cdots + \delta_n = 1.$$

The theorem on combining convex functions (Convex 2) implies that

$$-\ln\left(\sum_{i=1}^{n} \delta_i x_i\right) = f\left(\sum_{i=1}^{n} \delta_i x_i\right) \le \sum_{i=1}^{n} \delta_i f(x_i) = -\sum_{i=1}^{n} \delta_i \ln x_i.$$

This is equivalent to

$$\ln\left(\sum_{i=1}^{n} \delta_i x_i\right) \ge \sum_{i=1}^{n} \ln\left(x_i^{\delta_i}\right) = \ln\left(\prod_{i=1}^{n} x_i^{\delta_i}\right).$$

Since the logarithm function is strictly increasing, this becomes

$$\sum_{i=1}^{n} \delta_i x_i \ge \prod_{i=1}^{n} (x_i)^{\delta_i},$$

which is the general form of the A-G inequality (Duffin, Peterson, and Zener 1967). Note that it does not require that the weights be equivalent for equality, just the variables.

3.3.2 Geometric Programming Without Constraints

Geometric Programming is built around forcing equality in the Arithmetic-Geometric Inequality. GP is defined, and derived, for problems of a special structure. This structure requires a function $g(\mathbf{t})$ defined for all $\mathbf{t} = (t_1, \ldots, t_m)$ in R^m with $t_i > 0$ for all $i = 1, \ldots, m$. This structure is called a ***posynomial*** if $g(\mathbf{t})$ is of the form

$$g(\mathbf{t}) = \sum_{i=1}^{n} c_i \prod_{j=1}^{m} (t_j)^{\alpha_{ij}},$$

where the c_i's are positive constants and the α_{ij} are arbitrary real exponents. If any of the c_i's are negative the structure is termed a ***signomial*** (Peressini, Sullivan, and Uhl 1988). These are much more difficult problems. Sometimes they can be handled through the Geometric sign table. An example is included in the next chapter.

The goal of unconstrained GP is to solve the *primal geometric program*. This means:

(GP) Minimize the posynomial

$$g(\mathbf{t}) = \sum_{i=1}^{n} c_i \prod_{j=1}^{m} (t_j)^{\alpha_{ij}},$$

where $t_1 > 0, \ldots, t_m > 0.$

A *solution* to *(GP)* means a *global* minimizer $\mathbf{t}^*$ for $g(\mathbf{t})$ on the set of vectors $\mathbf{t}$ in R^m with positive components.

Note that $g(\mathbf{t})$ can be reformulated

$$g(\mathbf{t}) = \sum_{i=1}^{n} \delta_i \left(\frac{c_i \prod_{j=1}^{m} t_j^{\alpha_{ij}}}{\delta_i} \right),$$

where each δ_i is a positive number, the positivity condition (Duffin, Peterson, and Zener 1967). Now add the restriction (**Rule 2a**: Woolsey, 1969)

$$\sum_{i=1}^{n} \delta_i = 1 \qquad \text{(the Normality Condition)},$$

and apply the A-G Inequality to this expression to obtain:

$$g(\mathbf{t}) \overset{(A-G)}{\geq} \prod_{i=1}^{n} \left(\frac{c_i \prod_{j=1}^{m} t_j^{\alpha_{ij}}}{\delta_i} \right)^{\delta_i}$$

$$= \prod_{i=1}^{n} \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \left(\prod_{i=1}^{n} \prod_{j=1}^{m} t_j^{\alpha_{ij}\delta_i} \right)$$

$$= \prod_{i=1}^{n} \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \prod_{j=1}^{m} t_j^{\sum_{i=1}^{n} \alpha_{ij}\delta_i} .$$

At this point another restriction (**Rule 2b**: Woolsey 1969) is applied:

$$\sum_{i=1}^{n} \alpha_{ij} \delta_i = 0 ; \qquad j = 1 , \dots , m \qquad \text{(the Orthogonality Condition)},$$

and the preceding inequality becomes

$$g(\mathbf{t}) \geq \prod_{i=1}^{n} \left(\frac{c_i}{\delta_i} \right)^{\delta_i} .$$

Now define

$$v(\vec{\delta}) = \prod_{i=1}^{n} \left(\frac{c_i}{\delta_i} \right)^{\delta_i} .$$

The preceding work reveals

$$g(\mathbf{t}) \geq v(\vec{\delta}) \qquad \text{(the \textbf{Primal-Dual Inequality})}$$

for any $\mathbf{t} \in R^m$ with positive components and any $\vec{\delta} \in R^n$ that satisfies the Positivity, Normality, and Orthogonality Conditions (Duffin, Peterson, and Zener 1967).

This brings us to the *dual geometric program* (**Rule 1**: Woolsey 1969):

$$(DGP) \qquad \text{Maximize} \quad v(\vec{\delta}) = \prod_{i=1}^{n} \left(\frac{c_i}{\delta_i} \right)^{\delta_i},$$

subject to

$$\delta_1 > 0, \ldots, \delta_n > 0 \qquad \text{(the Positivity Condition)},$$

$$\sum_{i=1}^{n} \delta_i = 1 \qquad \text{(the Normality Condition)},$$

$$\sum_{i=1}^{n} \alpha_{ij}\delta_i = 0; \quad \text{for all } j \qquad \text{(the Orthogonality Condition)}.$$

A vector $\vec{\delta} \in R^n$ that satisfies the Positivity, Normality, and Orthogonality Conditions is a *feasible* vector for *(DGP)*. The dual program is *consistent* if the set of feasible vectors is not empty. A *solution* to the *(DGP)* is a global maximizer for $v(\vec{\delta})$ on the set of feasible vectors for *(DGP)*.

If **t** is a solution to *(GP)* and $\vec{\delta}$ is a solution to *(DGP)*, then

$$g(\mathbf{t}) \geq v(\vec{\delta})$$

by the Primal-Dual Inequality (Duffin, Peterson, and Zener 1967).

A typical term of $g(\mathbf{t})$ is

$$u_i(\mathbf{t}) = c_i t_1^{\alpha_{i1}} \cdots t_m^{\alpha_{im}}.$$

Note:

$$t_j \frac{\partial u_i}{\partial t_j} = \alpha_{ij} u_i .$$

Because $\mathbf{t}^* = (t_1^*, \ldots, t_m^*)$ is a minimizer for $g(\mathbf{t})$, we see

$$0 = \frac{\partial g}{\partial t_j}(\mathbf{t}^*) = \sum_{i=1}^{n} \frac{\partial u_i}{\partial t_j}(\mathbf{t}^*), \qquad j = 1, 2, \ldots, m.$$

$$0 = \sum_{i=1}^{n} \alpha_{ij} u_i(\mathbf{t}^*)$$

Since $g(\mathbf{t}^*) > 0$, divide both sides by $g(\mathbf{t}^*)$.

$$0 = \sum_{i=1}^{n} \alpha_{ij} \left(\frac{u_i(\mathbf{t}^*)}{g(\mathbf{t}^*)} \right).$$

Now set:

$$\delta_i^* = \frac{u_i(\mathbf{t}^*)}{g(\mathbf{t}^*)}, \qquad i = 1, \ldots, n,$$

then $\vec{\delta}^* = (\delta_1^*, \ldots, \delta_n^*)$ satisfies the Orthogonality Condition for the *(DGP)*. Because $\delta_i^* > 0$ for $i = 1, \ldots, n$ the Positivity Condition is satisfied. Next,

$$\sum_{i=1}^{n} \delta_i^* = \sum_{i=1}^{n} \left(\frac{u_i(\mathbf{t}^*)}{g(\mathbf{t}^*)} \right) = \frac{g(\mathbf{t}^*)}{g(\mathbf{t}^*)} = 1,$$

so the Normality Condition holds. Thus the vector $\vec{\delta}^*$ is feasible for the dual program so *(DGP)* is consistent. Also

$$g(\mathbf{t}^*) = g(\mathbf{t}^*)^{\delta_1^* + \cdots + \delta_n^*} = (g(\mathbf{t}^*))^{\delta_1^*} \cdots (g(\mathbf{t}^*))^{\delta_n^*}$$

$$= \left(\frac{u_1(\mathbf{t}^*)}{\delta_1^*} \right)^{\delta_1^*} \cdots \left(\frac{u_n(\mathbf{t}^*)}{\delta_n^*} \right)^{\delta_n^*}$$

$$= \left(\frac{c_1}{\delta_1^*} \right)^{\delta_1^*} \cdots \left(\frac{c_n}{\delta_n^*} \right)^{\delta_n^*} = v(\vec{\delta}^*)$$

(**Rule 3**: Woolsey 1969)

so equality holds in the Primal-Dual Inequality. This implies that $\vec{\delta}^*$ is a solution of *(DGP)* (Duffin, Peterson, and Zener 1967). Since

$$\delta_i^* = \frac{u_i(\mathbf{t}^*)}{g(\mathbf{t}^*)}.$$

3.3.3 Geometric Programming With Constraints

When dealing with constrained geometric programs, the problem is to minimize a posynomial objective function subject to posynomial constraints. The Arithmetic-Geometric Inequality is still the central concept, but K-K-T theory provides the theoretical justification for the technique (Peressini, Sullivan, and Uhl 1988). This gets back to the fourth possibility of nonlinear optimization where local optimal points can exist.

Consider a sequence of δ_i's which are all positive and sum to some number λ. Note that δ_i/λ is positive for $i = 1, 2, \ldots, n$ and that

$$\frac{\delta_1}{\lambda}+\frac{\delta_2}{\lambda}+\cdots+\frac{\delta_n}{\lambda}=1.$$

Now use the Arithmetic-Geometric Mean Inequality with the above sequences to obtain

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{n}\left(\frac{\delta_i}{\lambda}\right)\left(\frac{\lambda x_i}{\delta_i}\right) \geq \prod_{i=1}^{n}\left(\frac{\lambda x_i}{\delta_i}\right)^{\left(\frac{\delta_i}{\lambda}\right)},$$

with equality holding, if and only if

$$\frac{\lambda x_1}{\delta_1}=\frac{\lambda x_2}{\delta_2}=\cdots=\frac{\lambda x_n}{\delta_n}.$$

Consequently,

$$\left(\sum_{i=1}^{n} x_i\right)^{\lambda} \geq \prod_{i=1}^{n}\left(\frac{\lambda x_i}{\delta_i}\right)^{\delta_i} = \lambda^{\lambda}\prod_{i=1}^{n}\left(\frac{x_i}{\delta_i}\right)^{\delta_i}.$$

Now, if $\lambda x_i/\delta_i = \mathrm{M}$ for $i = 1, 2, \ldots, n$, then

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{n}\frac{\mathrm{M}\delta_i}{\lambda} = \frac{\mathrm{M}}{\lambda}\sum_{i=1}^{n}\delta_i = \mathrm{M},$$

which means,

$$x_i = \frac{\mathrm{M}\delta_i}{\lambda} = \frac{\delta_i}{\lambda}\left(\sum_{i=1}^{n} x_i\right),$$

so that, the equality condition holds.

Define the conventions $0^0 = 1$ and $(x_i / 0)^0 = 1$. Note if all the δ_i's are equal to zero, then both sides of the inequality are equal to 1 (Duffin, Peterson, and Zener 1967).

We can now define the standard constrained geometric program. Suppose that $g_0(t), g_1(t), \ldots, g_k(t)$ are posynomials in m positive real variables $t = (t_1, t_2, \ldots, t_m)$. Then the program

$$(GP) \qquad \text{Minimize} \quad g_0(t)$$

subject to the constraints

$$g_1(t) \leq 1, \quad g_2(t) \leq 1, \ldots, \quad g_k(t) \leq 1,$$

$$\text{where} \quad t_1 > 0 \quad t_2 > 0, \ldots, \quad t_m > 0$$

is called a *constrained geometric program.*

Because each of the $k+1$ posynomials in the above standard program *(GP)* may consist of several terms, we need a consistent system of notation. A way of doing this is to begin by counting the terms of the objective function $g_0(t)$ from 1 to n_0. Then the first constraint $g_1(t)$ from $n_0 + 1$ to n_1. And so on until the last constraint $g_k(t)$ from $n_{k-1} + 1$ to $n_k = p$ (Peressini, Sullivan, and Uhl 1988). The jth term in this scheme is then:

$$u_j(t) = c_m t_1^{\alpha_{j1}} t_2^{\alpha_{j2}} \cdots t_m^{\alpha_{jm}}.$$

The standard constrained geometric program *(GP)* can then be rewritten:

$$(GP) \qquad \text{Minimize} \qquad g_0(\mathbf{t}) = u_1(\mathbf{t}) + \cdots + u_{n_0}(\mathbf{t})$$

subject to the constraints

$$g_1(\mathbf{t}) = u_{n_0+1}(\mathbf{t}) + \cdots + u_{n_1}(\mathbf{t}) \le 1,$$

$$\vdots$$

$$g_k(\mathbf{t}) = u_{n_{k-1}+1}(\mathbf{t}) + \cdots + u_{n_k}(\mathbf{t}) \le 1,$$

$$\text{where} \quad t_1 > 0, \quad t_2 > 0, \ldots, \quad t_m > 0, \quad \text{and} \quad n_k = p.$$

Application of the A-G Inequality and rules 2a and 2b allows us to see the standard dual geometric program *(DGP)*:

$$(DGP) \qquad \text{Maximize} \qquad v(\vec{\delta}) = \left(\prod_{j=1}^{p} \left(\frac{c_j}{\delta_j} \right)^{\delta_j} \right) \prod_{i=1}^{k} \lambda_i(\vec{\delta})^{\lambda_i(\vec{\delta})}$$

subject to the constraints

$$\begin{pmatrix} \delta_1 & + & \cdots & + & \delta_{n_0} & = & 1 \\ \alpha_{11}\delta_1 & + & \cdots & + & \alpha_{p1}\delta_p & = & 0 \\ \vdots & & & & \vdots & & \vdots \\ \alpha_{1m}\delta_1 & + & \cdots & + & \alpha_{pm}\delta_p & = & 0 \end{pmatrix}$$

where $\delta_i > 0$ for $i = 1, \ldots, n_0$ and for each $k \ge 1$, either $\delta_i > 0$ for all i

with $n_{k-1}+1 \le i \le n_k$ or $\delta_i = 0$ for all i with $n_{k-1}+1 \le i \le n_k$. Here, $\lambda_i(\vec{\delta}) = \delta_{n_{i-1}+1} + \cdots + \delta_{n_i}$.

The program *(DGP)* is called the *dual program* of *(GP)*, the function $v(\vec{\delta})$ is the *dual objective function* and the constraints in *(DGP)* are the *dual constraints*. If $\delta_1, \delta_2, \ldots, \delta_p$ are numbers that satisfy the dual constraints in *(DGP)*, then $\vec{\delta} = (\delta_1, \ldots, \delta_p)$ is a *feasible vector* for *(DGP)* which is said to be consistent. A vector $\vec{\delta}^* = (\delta_1^*, \ldots, \delta_p^*)$ that maximizes $v(\vec{\delta})$ on the set of feasible vectors for *(DGP)* is a *solution* to *(DGP)* (Duffin, Peterson, and Zener 1967).

In general, the number of dual variables in *(DGP)* is equal to the total number $p\ (= n_k)$ of terms in the objective and constraint functions for *(GP)*. The constraint equations in *(DGP)* are linear, so that the problem of finding feasible vectors for *(DGP)* is reduced to the feasible region of a system of linear equations. The term $\lambda_i(\vec{\delta})$ is the sum of the components of $\vec{\delta}$ which correspond to the terms of the ith constraint function $g_i(\mathbf{t})$ for $i = 1, 2, \ldots, k$ (Peressini, Sullivan, and Uhl 1988).

Since posynomials are not necessarily convex the applicability of K-K-T theory to GP is not obvious. But, any posynomial $g(\mathbf{t})$ can be transformed into a convex function $h(\mathbf{x})$ by the change of variables

$$t_j = e^{x_j}, \qquad j = 1, 2, \ldots, m.$$

Using the above change of variables on the posynomial

$$g(\mathbf{t}) = \sum_{i=1}^{n} c_i t_1^{\alpha_{i1}} t_2^{\alpha_{i2}} \cdots t_m^{\alpha_{im}}, \qquad c_i > 0,$$

results in the following function of $\mathbf{x} = (x_1, x_2, \ldots, x_m)$

$$h(\mathbf{x}) = \sum_{i=1}^{n} c_i e^{\sum_{j=1}^{m} \alpha_{ij} x_j},$$

where $h(\mathbf{x})$ is convex on R^m by (Convex 6).

This allows the transformation of the standard constrained geometric program

(GP) Minimize $g_0(\mathbf{t})$

subject to the constraints

$$g_1(\mathbf{t}) \le 1, \qquad g_2(\mathbf{t}) \le 1, \ldots, \qquad g_k(\mathbf{t}) \le 1,$$

$$\text{where} \qquad t_1 > 0, \qquad t_2 > 0, \ldots, \qquad t_m > 0$$

into the *associated convex program*

$(GP)^*$ Minimize $h_0(\mathbf{x})$

subject to the constraints

$$h_1(\mathbf{x}) - 1 \le 0, \qquad h_2(\mathbf{x}) - 1 \le 0, \ldots, \qquad h_k(\mathbf{x}) - 1 \le 0,$$

$$\text{where} \qquad \mathbf{x} \in R^m.$$

Because $t = e^x$ is a strictly increasing function, the programs *(GP)* and $(GP)^*$ are equivalent in that $\mathbf{t}^* = (t_1^*, t_2^*, \ldots, t_m^*)$ is a solution to *(GP)* if and only if $\mathbf{x}^* = (x_1^*, \ldots, x_m^*)$ is a solution to $(GP)^*$ where $t_i^* = e^{x_i^*}$ for $i = 1, 2, \ldots, m$ (Duffin, Peterson, and Zener 1967).

Given that $\mathbf{t}$ is a feasible vector for *(GP)* and that $\vec{\delta}$ is a feasible vector for *(DGP)* then, the definitions of the A-G Inequality and the *(DGP)* show

$$g_0(\mathbf{t}) \geq v\,(\vec{\delta}) \qquad \text{(the Primal-Dual Inequality)}.$$

Because *(GP)* is superconsistent, so is the associated convex program $(GP)^*$. Since *(GP)* has a solution $\mathbf{t}^* = (t_1^*, t_2^*, \ldots, t_m^*)$, $(GP)^*$ has a solution $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_m^*)$ given by

$$x_i^* = \ln t_i^*, \qquad i = 1, 2, \ldots, m,$$

(Duffin, Peterson, and Zener 1967).

K-K-T theory requires a vector $\vec{\lambda}^* = (\lambda_1^*, \ldots, \lambda_k^*)$ such that

(a) $\lambda_i^* \geq 0 \quad \text{for} \quad i = 1, 2, \ldots, k;$

(b) $\lambda_i^* (h_i(\mathbf{x}^*) - 1) = 0 \quad \text{for} \quad i = 1, 2, \ldots, k;$

(c) $\dfrac{\partial h_0}{\partial x_j}(\mathbf{x}^*) + \sum_{i=1}^{k} \lambda_i^* \dfrac{\partial h_i}{\partial x_j}(\mathbf{x}^*) = 0 \quad \text{for} \quad j = 1, \ldots, m.$

Because $t_i = e^{x_i}$ for $i = 1, \ldots, m$, it follows that for $i = 0, 1, \ldots, k$

$$\frac{\partial h_i}{\partial x_j} = \frac{\partial h_i}{\partial t_j}\frac{\delta t_j}{\delta x_j} = \frac{\partial g_i}{\partial t_j} e^{x_j},$$

so that condition (c) is equivalent to

$$(c') \qquad \frac{\partial g_0}{\partial t_j}(\mathbf{t}^*) + \sum_{i=1}^{k} \lambda_i^* \frac{\partial g_i}{\partial t_j}(\mathbf{t}^*) = 0, \qquad j = 1, 2, \ldots, m$$

since $e^{x_j} > 0$ for $j = 1, \ldots, m$. But $t_j^* > 0$ for $j = 1, 2, \ldots, m$, so (c') is equivalent to

$$(c'') \qquad t_j^* \frac{\partial g_0}{\partial t_j}(\mathbf{t}^*) + \sum_{i=1}^{k} \lambda_i^* t_j^* \frac{\partial g_i}{\partial t_j}(\mathbf{t}^*) = 0, \qquad j = 1, \ldots, m.$$

Because the terms of $g_i(\mathbf{t})$ are of the form

$$u_q(\mathbf{t}) = c_q t_1^{\alpha_{q1}} t_2^{\alpha_{q2}} \cdots t_m^{\alpha_{qm}},$$

it is clear that

$$t_j^* \frac{\partial g_i}{\partial t_j}(\mathbf{t}^*) = \sum_{q=n_{i-1}+1}^{n_i} \alpha_{qj} u_q(\mathbf{t}^*), \qquad j = 1, \ldots, m,$$

so (c'') implies

$$0 = \sum_{q=1}^{n_0} \alpha_{qj} u_q(\mathbf{t}^*) + \sum_{r=1}^{k} \sum_{q=n_{r-1}+1}^{n_r} \lambda_r^* \alpha_{qj} u_q(\mathbf{t}^*),$$

(Peressini, Sullivan, and Uhl 1988).

Dividing the last equation by

$$g_0(\mathbf{t}^*) = \sum_{q=1}^{n_0} u_q(\mathbf{t}^*),$$

results in

$$0 = \sum_{q=1}^{n_0} \alpha_{qj}\left(\frac{u_q(\mathbf{t}^*)}{q_0(\mathbf{t}^*)}\right) + \sum_{r=1}^{k} \sum_{q=n_{r-1}+1}^{n_r} \alpha_{qj}\left(\frac{\lambda_r^* u_q(\mathbf{t}^*)}{g_0(\mathbf{t}^*)}\right).$$

Define the vector $\vec{\delta}_q^*$ by

$$\vec{\delta}_q^* = \begin{cases} \dfrac{u_q(\mathbf{t}^*)}{g_0(\mathbf{t}^*)}; & q = 1, 2, \ldots, n_0, \\[2ex] \dfrac{\lambda_r^* u_q(\mathbf{t}^*)}{g_0(\mathbf{t}^*)}; & q = n_{r-1}+1, \ldots, n_r, \quad r = 1, \ldots, k. \end{cases}$$

Note that $\delta_q^* > 0$ for $q = 1, 2, \ldots, n_0$ and that, for each $r \geq 1$, either $\delta_i^* > 0$ for all i with $n_{r-1} + 1 \leq i \leq n_r$ or $\delta_i^* = 0$ for all over the same range. Consequently the corresponding K-K-T multiplier λ_r^* is positive or zero. Further note the vector $\vec{\delta}^*$ satisfies all the m exponent constraint equations in (DGP) as well as the constraint

$$\sum_{q=1}^{n_0} \delta_q^* = \sum_{q=1}^{n_0} \frac{u_q(\mathbf{t}^*)}{g_0(\mathbf{t}^*)} = 1.$$

Accordingly, $\vec{\delta}^* = (\delta_1^*, \ldots, \delta_p^*)$ is a feasible vector for *(DGP)* (Duffin, Peterson, and Zener 1967).

The K-K-T multipliers λ_r^* are related to the corresponding

$\lambda_r(\vec{\delta}^*)$ in *(DGP)* as follows (Duffin, Peterson, and Zener 1967):

$$\lambda_r(\vec{\delta}^*) = \sum_{q=n_{r-1}+1}^{n_r} \delta_q^* = \sum_{q=n_{r-1}+1}^{n_r} \lambda_r^* \frac{u_q(\mathbf{t}^*)}{g_0(\mathbf{t}^*)} = \lambda_r^* \frac{g_r(\mathbf{t}^*)}{g_0(\mathbf{t}^*)}$$

for $r = 1, \ldots, k$. The K-K-T condition (b) yields

$$(b') \qquad \lambda_r^*(g_r(\mathbf{t}^*) - 1) = 0, \qquad r = 1, \ldots, k,$$

so $\lambda_r^* g_r(\mathbf{t}^*) = \lambda_r^*$ for $r = 1, \ldots, k$. Therefore, for $r = 1, \ldots, k$ and $q = n_{r-1} + 1, \ldots, n_r$, we see

$$\delta_q^* = \frac{\lambda_r^* u_q(\mathbf{t}^*)}{g_0(\mathbf{t}^*)} = \frac{\lambda_r^* g_r(\mathbf{t}^*) u_q(\mathbf{t}^*)}{g_0(\mathbf{t}^*)} = \lambda_r(\vec{\delta}^*) u_q(\mathbf{t}^*).$$

(**Rule 4:** Woolsey 1969)

Equation (b') shows that either $g_r(\mathbf{t}^*) = 1$ or $\lambda_r^* = 0$ for $r = 1, 2, \ldots, k$ and the equation above (Rule 4) shows that $\lambda_r^* = 0$ if and only if $\lambda_r(\vec{\delta}^*) = 0$ for $r = 1, \ldots, k$. This means that the values of δ_q^* actually force equality in the Primal-Dual Inequality (Duffin, Peterson, and Zener 1967)

$$g_0(\mathbf{t}^*) = v(\vec{\delta}^*).$$

The above result has important consequences. First the theoretical existence of the K-K-T multipliers is all that is needed to prove that the calculation of $\vec{\delta}^*$ of *(DGP)* and hence the calculation of t^* of *(GP)* can be made. Second, the K-K-T multipliers λ_r^* are related to the sum of each constraints δ's, that is the term: $\lambda_r(\vec{\delta}^*)$, prescribed in *(DGP)* by the formula

$$\lambda_r(\vec{\delta}^*) = \frac{\lambda_r^*}{g_0(t^*)}.$$

Since the K-K-T multiplier λ_r^* measures the sensitivity of the solution of the associated convex program to changes in the *r*th constraint, the formula above shows that the size of $\lambda_r(\vec{\delta}^*)$ gives a qualitative measure of the sensitivity of the solution of the constrained geometric program to its *r*th constraint. The larger the value of $\lambda_r(\vec{\delta}^*)$, the greater the decrease of the minimum value of the geometric program that results from relaxation of the *k*th constraint (Peressini, Sullivan, and Uhl 1988).

The Primal-Dual Inequality can also be used to bound geometric programs as follows:

$$g_0(t) \geq \min(GP) \geq \max(DGP) \geq v(\vec{\delta}).$$

The result of all this is twofold. First, the use of the techniques of GP on LP problems is shown to be legitimate. This follows from the demonstrations that LP and GP programs are both convex programs. Second, the four rules of GP are derived.

They will be used extensively later in this thesis.

Note the demonstration that both LP's and GP's are convex programs means that any LP can be formulated as a posynomial GP. The previous theory on posynomial GP's tells us that if the program can be solved, the solution will be a global optimum equivalent to the LP optimum solution.

Since the LP blending problem is a proper subset of Linear Programming, it can be formulated as a posynomial GP. As such it will be a proper subset of the set of all posynomial GP's. Therefore if the GP can be solved the answer will be a global optimum equivalent to the LP global optimum.

3.3.4 Condensation

Condensation is the use of the A-G Inequality to approximate a multiterm posynomial with a single-term function called a monomial. At optimality the approximation is exact.

Partial condensation is the use of condensation to reduce at least two terms of a multiterm problem to a monomial form.

By exploiting both left and right sides of constraints, the technique can be expanded to include signomial problems. Signomial objective functions can sometimes be handled by reformulating and inducing an additional constraint.

The technique has two prime functions in GP. One is to reduce the degree of difficulty of multiterm problems. The other is to assist in obtaining the required formulation for GP techniques.

Condensation has proven to be an extremely efficient procedure when applicable. Its use tends to develop problems requiring iterative solutions. These are developed using rules 3 and 4 which hold at optimality. Prior to the achievement of optimality they result in progressively better current solutions.

3.3.5 Surrogation

Surrogation is a technique for aggregating constraints. Weights are assigned to each constraint. The weights always sum to one. These weights are carried by each term and all similar terms are linearly combined. This allows for reformulation of the problem and hopefully a zero degree of difficulty *(DGP)* problem to be solved.

If the variable values calculated result in equality between the *(GP)* and *(DGP)* programs the weights selected are correct. If this is not the case, the weights need to be adjusted until equality is obtained.

Surrogation has proven useful with certain types of problems but can be a lengthy procedure.

3.3.6 The Advanced GP Sign Table

The ***GP sign*** table is a concise representation of the GP problem being undertaken. One axis is used to represent each term in the problem. The other axis is used to represent each variable in the problem. The advanced GP, or weighted, sign of a variable is the product of the exponent on an individual variable in a term and the sign of that terms coefficient. If neither or both of these is positive, the sign will be positive. If this is not the case, the sign will be negative.

A term is a combination of variables with a coefficient separated from the rest of the function by either a binary (+/-) or relational (= / ≤ / ≥ / < / >) operator.

A well formulated LP or GP problem requires that each variable in that problem be upper and lower bounded, a condition defined as the ***balance*** of the variable. The sign table allows the user to quickly determine if this is the case. Every variable on the table must have at least one positive and one negative sign, among all the terms of the problem, to be balanced. A variable need not have a sign for each term on the table, because not all variables may be in all terms. We are looking for a variable that has only one positive or one negative sign in the table. When this occurs, the term containing that sign is essential to the problem. Terms cannot be split off from their respective function. Thus the constraint containing that term is essential to the problem. The objective function is always essential.

A result of condensation can be variables with several exponents in a single term. Some of these exponents are geometric weights which are made up of combinations of the variables in that term. When an iterative solution is required, these weights (δ'*s*) change with each iteration. Thus a solution (hopefully close to feasible) is required to evaluate the advanced sign table.

Chapter 4

THE PROCEDURE OF THE METHOD

Step I Examine the LP

(a) Look for pairs of constraints where each of the individual terms has the same functional form. This means if the pair of constraints were both equalities they could be added together. If the constraints are not equal, one will be ***dominant***. Dominance between two constraints means any solution which satisfies the first of the constraints always satisfies the second but, solutions which satisfy the second do not necessarily satisfy the first. Since the first constraint is more difficult to satisfy, it is said to dominate the second constraint. Naturally both constraints contain the same variables in the same functional form. Retain the dominant constraint and eliminate the other from the problem. Compare all such pairs, eliminate one in each case possible. Continue until no feasible pairs remain to be considered.

(b) Use each equality constraint to eliminate a variable from the problem.

(c) Repeat steps (a) and (b) until the problem stabilizes.

(d) If (b) does not simplify the problem, replace the equality with two inequality ($\leq / \geq$) constraints.

Step II Formulate the GP

(a) Put the objective function into Minimization form:

(1) Use as is if you are dealing with the minimization of a posynomial, unless condensation is required to reduce the problem degree of difficulty.

(2) If you are dealing with the maximization of a posynomial, condense it and invert to form the minimization problem.

(3) For a signomial objective function, group the terms according to the sign of their coefficients. The function will need to be reformulated to group the positive terms separately from the negative. It will require inversion if it is a maximization problem. The new objective function will be a posynomial.

(4) Either of substeps 2 or 3 (above) will require the formulation of an induced constraint. This constraint relates the new objective function to the original. It is written in GP constraint form but in reality is always tight.

(b) Put the constraints into proper form:

(1) Separate positive and negative terms to opposite sides of the inequality, all terms become positive.

(2) Condense the terms on each side of the inequality as separate problems. Condense only as required to achieve zero degrees of difficulty.

(3) Divide through for proper form.

Step III Form the Advanced GP Sign Table

(a) Recall section 3.3.6 in order to build the GP Sign Table. If all the signs are obvious, that is positive or negative, note which constraints alone balance one or more variables of the problem.

(a') If some signs are not obvious, meaning you cannot tell which of the exponents being summed will define the sign, evaluate the advanced sign table to determine the sign. It is advisable to use a known feasible solution for this evaluation. If no feasible solution is known, at least use feasible variable values based on their primary constraints. This is the same as generalized upper bounding. As above, note which constraints alone balance one or more variables, and retain these constraints.

(b) Note which combinations of those constraints retained above completely balance all problem variables. This means every variable in the problem is contained in balanced form between the objective function and retained constraints. Thus if we built another sign table to represent the retained problem, every variable would have at least one positive and one negative entry in the table. If there is only one combination, all constraints in it are essential. If there is more than one combination, one of them represents the optimal intersection of constraints. This event is a candidate for the technique of surrogation. Typical blending problems exhibit one combination of constraints as essential.

(c) Each essential constraint is now an equality. Use each equality to eliminate a variable by returning to the LP and repeating the above procedure starting with I(b). If not using surrogation in III(b) above, each combination should be considered essential in turn.

(d) Take advantage of engineering knowledge of the process and the natural sparcity of the matrix. Look for blending stocks which alone can bring in a product specification. This is the same as balancing a variable, except in this case we are examining the transpose and balancing a constraint. Think of what we have been required to do as balancing horizontally. Now we can and should, if possible, balance vertically.

Step IV Determine Optimal Variable Values

(a) When the problem has been reduced to zero degrees of difficulty it is ready for solution.

(b) If the problem can be directly solved using the *(DGP)*, rules 3 and 4, do so.

(c) If (b) is not the case, use rules 3 and/or 4 to set up recursive relationships between the variables and geometric weights. Proceed to iterate to a solution.

(d) If any iterating variable is decreasing, note its effect on the objective function. If it is a positive contributor (meaning it contributes in the direction we want to go) to either a max or a min problem, turn it off. If it is a negative contributor, see if it is decreasing faster than the objective function is decreasing (min problem). If so, turn it off. In the case of a max problem, it is obvious that this variable should be turned off. If any iterating variable iterates beyond its upper or lower limit, assign that variable its limit.

(e) By turning off we mean assigning a value of zero to that variable. When this is done the problem needs to be reformulated. This is no more difficult than the reformulations required when using equality constraints to eliminate variables.

Chapter 5

EXAMPLE PROBLEMS

The following examples are progressively more difficult problems which illustrate the use of the theory and method presented in chapter 3. No single example uses all of the steps described in the method. In total they present a graphic demonstration of the power of the method.

5.1 Fertilizer Blending Problem

This model (Shogan 1988) represents the amounts of available fertilizer blending stocks, containing different percentages of nitrogen, phosphorus, and potassium required to produce a specific fertilizer product. Each available blending stock has a different manufacturing cost which will contribute to the product cost in proportion to the amount of that stock used. It is desired to keep the total product cost as low as possible.

The percentage compositions of the blending stocks and the desired product are given in the table below:

	Percentage Composition of Blending Stocks			Minimum % Req'd
	x_1 50 – 20 – 5	x_2 0 – 15 – 20	x_3 10 – 10 – 10	Z 17 – 14 – 10
Pounds of Nitrogen/ 100 lb of stock	50	0	10	17
Pounds of Phosphorus/ 100 lb of stock	20	15	10	14
Pounds of Potassium/ 100 lb of stock	5	20	10	10
Cost per 100 lb	\$90	\$20	\$30	

Find the mixture of blending stocks which achieves the product requirements at the lowest possible cost.

The LP model, for 1000 lb of product, is

Minimize $90x_1 + 20x_2 + 30x_3$

Subject to

$$\begin{pmatrix} 50x_1 & + & & + & 10x_3 & \geq & 170 \\ 20x_1 & + & 15x_2 & + & 10x_3 & \geq & 140 \\ 5x_1 & + & 20x_2 & + & 10x_3 & \geq & 100 \end{pmatrix}.$$

Using the first step in the method, it can be seen that all three constraints have the same form. However, dominance between them is not obvious. Reformulating the constraints

$$\begin{pmatrix} \frac{5}{17}x_1 & + & & + & \frac{1}{17}x_3 & \geq & 1 \\ \frac{1}{7}x_1 & + & \frac{3}{28}x_2 & + & \frac{1}{14}x_3 & \geq & 1 \\ \frac{1}{20}x_1 & + & \frac{1}{5}x_2 & + & \frac{1}{10}x_3 & \geq & 1 \end{pmatrix}$$

is helpful but still not conclusive. Note that in this form the problem is not only in LP form but also signomial GP form. Therefore step II(a) is accomplished. However, for purposes of illustration, we can use condensation on the constraints, step II(b), to put the problem into the following posynomial GP form.

$$170\left(\frac{\delta_{a1}}{50x_1}\right)^{\delta_{a1}}\left(\frac{\delta_{a2}}{10x_3}\right)^{\delta_{a2}} \leq 1$$

$$140\left(\frac{\delta_{b1}}{20x_1}\right)^{\delta_{b1}}\left(\frac{\delta_{b2}}{15x_2}\right)^{\delta_{b2}}\left(\frac{\delta_{b3}}{10x_3}\right)^{\delta_{b3}} \leq 1$$

$$100\left(\frac{\delta_{c1}}{5x_1}\right)^{\delta_{c1}}\left(\frac{\delta_{c2}}{20x_2}\right)^{\delta_{c2}}\left(\frac{\delta_{c3}}{10x_3}\right)^{\delta_{c3}} \leq 1$$

Either form of the constraints fulfills the requirements of the second step of the method. For the third step, we will use the latter form to build the advanced sign table.

	Objective Function			Constraints		
Variable				(a)	(b)	(c)
	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6
x_1	$+1$			$-\delta_{a1}$	$-\delta_{b1}$	$-\delta_{c1}$
x_2		$+1$			$-\delta_{b2}$	$-\delta_{c2}$
x_3			$+1$	$-\delta_{c1}$	$-\delta_{c2}$	$-\delta_{c3}$

The table points out the obvious. The objective function, as always, is required in the well-formulated problem. Either (b) or (c) could balance the problem alone. Or they might balance it in combination with another constraint. If constraint (a) is required at least one of the other two constraints is required to balance the x_2 variable.

A person might think that all three constraints could be required. Since there are only three variables, this event coincides with three equations and three unknowns. Inverting this matrix returns the following optimal values:

$$\mathbf{x} = (x_1^*, x_2^*, x_3^*) = (6, 10, -13).$$

Variable values less than zero $[\mathbf{x} \in R^n : \mathbf{x} \geq \mathbf{0}]$ are not possible in the class of problems being studied here. This restriction rules out the three tight constraints solution.

This means there are five possible combinations of the constraints, any of which could be the optimum. We could calculate the solution to each of these possibilities and then pick the minimum. This is feasible and would result in the true optimum answer. But the purpose here is to demonstrate a method which can be used on much bigger problems. Step III(b) is applicable to this case.

The surrogated constraints are shown below.

$$\left(\frac{5}{17}\lambda_1 + \frac{1}{7}\lambda_2 + \frac{1}{20}\lambda_3\right)x_1 + \left(\frac{3}{28}\lambda_2 + \frac{1}{5}\lambda_3\right)x_2 + \left(\frac{1}{17}\lambda_1 + \frac{1}{14}\lambda_2 + \frac{1}{10}\lambda_3\right)x_3 \geq 1$$

For a given $\overline{\lambda}$, the above equation can be represented by

$$A\,x_1 + B\,x_2 + C\,x_3 \geq 1$$

which is the signomial form of the surrogated constraints. Putting this into posynomial GP form produces the following GP problem:

$$\text{Minimize} \qquad 90x_1 + 20x_2 + 30x_3$$

$$\text{Subject to} \qquad \left(\frac{\delta_1}{A\,x_1}\right)^{\delta_1}\left(\frac{\delta_2}{B\,x_2}\right)^{\delta_2}\left(\frac{\delta_3}{C\,x_3}\right)^{\delta_3} \le 1$$

which is a zero degree of difficulty exercise in elementary GP techniques. However the point at this level of the solution is to determine which combination of constraints is completely tight.

Given a feasible starting solution a GP based iterative solution process should return a feasible working solution from which to proceed. The techniques of surrogation show how to re-weight the constraints to get feasible return solutions. A consequence of this is that the required, or tight, constraints get progressively higher weights ($\vec{\lambda}$'s). This is valuable information. Constraints which continuously get decreasing weights are not required.

Now apply step IV. Form the DGP (Rule 1) as follows:

$$Z^* = (90)^{\delta_1}(20)^{\delta_2}(30)^{\delta_3}\left(\frac{1}{A}\right)^{\delta_1}\left(\frac{1}{B}\right)^{\delta_2}\left(\frac{1}{C}\right)^{\delta_3}$$

Subject to the following constraints (rules 2a and 2b):

$$\begin{pmatrix} \omega_1 & + & \omega_2 & + & \omega_3 & + & 0 & = & 1 \\ \omega_1 & + & 0 & + & 0 & - & \delta_1\omega_4 & = & 0 \\ 0 & + & \omega_2 & + & 0 & - & \delta_2\omega_4 & = & 0 \\ 0 & + & 0 & + & \omega_3 & - & \delta_3\omega_4 & = & 0 \end{pmatrix}$$

The simultaneous solution to this set of equations is

$$\vec{\omega} = (\omega_1, \omega_2, \omega_3, \omega_4) = (\delta_1, \delta_2, \delta_3, 1).$$

Using an initial feasible solution $\mathbf{x} = (3, 3, 3.5)$and weighting all three constraints equally $\vec{\lambda} = (1/3, 1/3, 1/3)$, returns the following values for the $\vec{\delta}$'s and constants:

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} .1623 \\ .1024 \\ .0768 \end{bmatrix} \qquad \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix} = \begin{bmatrix} .4582 \\ .2890 \\ .2528 \end{bmatrix}$$

Note that we used several more significant figures than those reported here and throughout the remainder of this chapter. In checking these calculations, do not round off !

The DGP can now be calculated ($Z^* = 375.4465$). Rule 3 allows the incremental calculation of the new variable vector.

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} 1.9115 \\ 5.4254 \\ 3.1634 \end{bmatrix}$$

Checking the constraints with the new solution returns:

$$\begin{bmatrix}(a)\\(b)\\(c)\end{bmatrix}=\begin{bmatrix}0.7483\\1.0803\\1.4970\end{bmatrix}$$

Because constraint (a) is violated, this is not a feasible solution. Constraint (c) is the most secure, so lighten it and favor (a). The new weights are $\vec{\lambda}=(1/2,1/3,1/6)$. Using the original feasible solution and repeating the above calculations returns the following:

$$\begin{bmatrix}A\\B\\C\end{bmatrix}=\begin{bmatrix}.2030\\.0690\\.0699\end{bmatrix} \qquad \begin{bmatrix}\delta_1\\\delta_2\\\delta_3\end{bmatrix}=\begin{bmatrix}.5741\\.1953\\.2306\end{bmatrix}$$

$$Z^* = 404.9476$$

$$\begin{bmatrix}x_1'\\x_2'\\x_3'\end{bmatrix}=\begin{bmatrix}2.5833\\3.9538\\3.1126\end{bmatrix} \qquad \begin{bmatrix}(a)\\(b)\\(c)\end{bmatrix}=\begin{bmatrix}0.9429\\1.0150\\1.2312\end{bmatrix}$$

Again constraint (a) is violated and (c) is still the most secure. We now suspect that (c) is not required. Assign it a weight of zero and weight (a) and (b) equally, $\vec{\lambda}=(1/2,1/2,0)$. Once more using the known feasible solution produces the next result.

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} .1828 \\ .0536 \\ .0651 \end{bmatrix} \qquad \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix} = \begin{bmatrix} .5852 \\ .1715 \\ .2433 \end{bmatrix}$$

$Z^* = 462.0204$

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} 3.0042 \\ 3.9624 \\ 3.7466 \end{bmatrix} \qquad \begin{bmatrix} (a) \\ (b) \\ (c) \end{bmatrix} = \begin{bmatrix} 1.1040 \\ 1.1213 \\ 1.3173 \end{bmatrix}$$

All constraints are feasible. There are now only two possibilities, either (a) and (b) are tight or (b) alone is tight. Evaluating the last possibility results in the next GP.

Minimize $\quad 90x_1 + 20x_2 + 30x_3$

Subject to

$$140\left(\frac{\delta_{b1}}{20x_1}\right)^{\delta_{b1}}\left(\frac{\delta_{b2}}{15x_2}\right)^{\delta_{b2}}\left(\frac{\delta_{b3}}{10x_3}\right)^{\delta_{b3}} \leq 1$$

The new DGP is

$$Z^* = (90)^{\delta_{b1}}(20)^{\delta_{b2}}(30)^{\delta_{b3}}\left[140\left(\frac{1}{20}\right)^{\delta_{b1}}\left(\frac{1}{15}\right)^{\delta_{b2}}\left(\frac{1}{10}\right)^{\delta_{b3}}\right]$$

subject to rules 2a and 2b as are all DGP's. This calculation returns the following:

$Z^* = 385.0485$

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} 1.8336 \\ 6.1883 \\ 3.2087 \end{bmatrix} \qquad \begin{bmatrix} (a) \\ (b) \\ (c) \end{bmatrix} = \begin{bmatrix} 0.7280 \\ 1.1542 \\ 1.6502 \end{bmatrix}$$

Again constraint (a) is violated. We now know that constraints (a) and (b) must both be tight. A fundamental rule from LP is: *The number of real variables in the basis may not exceed the number of constraints.* Since the problem has been reduced to two constraints, no more than two variables may be active. One of the problem's three variables must be assigned a value of zero. Examination of the two remaining constraints reveals that x_3 contributes the least to them until x_3 is five times larger than x_1 and three and a half times larger than the total contribution of x_1 and x_2. Any advantage of x_3 being this large is lost when evaluating the objective function. Therefore x_3 should be assigned a value of zero.

With that bit of analysis done, constraint (a) defines $x_1 = 3.4$, and constraint (b) then defines $x_2 = 4.8$. This allows determination of $Z^* = 402$, all of which agrees with the LP solution. What the answer means is that we should use 3.4 units of stock x_1, 4.8 units of stock x_2, and none of stock x_3 in order to produce product Z at minimum cost.

Note, we have not allowed for the normal complications which would arise in this type of problem. Typically we would need to produce a specific amount of product Z. Also the blending stocks would normally be available in specific size increments. These added complications would have turned this problem into a mixed integer LP. The problem was simplified for this example in order to present a basic example of the method's application.

Note that the GP-based method identified the tight constraints before any effort toward determining variable values was needed. The simplex method of LP, in contrast, would have started off by determining the best value of the most promising variable first and continued from there. Because of the manner in which the simplex method iterates around the feasible region, the essential constraint set is the last determination made.

5.2 Beer Blending Problem

This model (Woolsey 1987) represents the blending of five ingredients, each with varying amounts of four different properties and different prices. The objective is to obtain one product, with certain specifications concerning each of these properties, at the lowest price possible.

The ingredients are specified as follows:

	Ingredient					Product
	x_1	x_2	x_3	x_4	x_5	Z
Alcohol Vol. %	0	2.5	3.7	4.5	5.8	$z = 3.1$
Specific Gravity	1	1.030	1.043	1.050	1.064	$1.034 \leq z \leq 1.040$
Color (EBC units)	0	11	9	8	7	$8 \leq z \leq 10$
Hop Resin mg/l	0	30	20	28	30	$20 \leq z \leq 25$
Price Danish Kr	0	44	50	64	90	

It is desired to produce 100 hl. of the product. Prices are given in Danish krones. This simple problem results in the five variable eight constraint LP which follows:

Minimize $$44x_2 + 50x_3 + 64x_4 + 90x_5$$

Subject to

$$\begin{pmatrix} x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & = & 100 \\ & & 2.5x_2 & + & 3.7x_3 & + & 4.5x_4 & + & 5.8x_5 & = & 310 \\ x_1 & + & 1.03x_2 & + & 1.043x_3 & + & 1.05x_4 & + & 1.064x_5 & \geq & 103.4 \\ x_1 & + & 1.03x_2 & + & 1.043x_3 & + & 1.05x_4 & + & 1.064x_5 & \leq & 104.0 \\ & & 11x_2 & + & 9x_3 & + & 8x_4 & + & 7x_5 & \geq & 800 \\ & & 11x_2 & + & 9x_3 & + & 8x_4 & + & 7x_5 & \leq & 1000 \\ & & 30x_2 & + & 20x_3 & + & 28x_4 & + & 30x_5 & \geq & 2000 \\ & & 30x_2 & + & 20x_3 & + & 28x_4 & + & 30x_5 & \leq & 2500 \end{pmatrix}$$

The first equality is the volume constraint, and the second is the alcohol content constraint. The third constraint is the lower bound on specific gravity. The fourth constraint is the upper bound on specific gravity. The fifth constraint is the lower bound on color, and the sixth is the upper bound. The seventh and eighth constraint are lower and upper bounds on the hop resin concentration.

There are no pairs from which to judge dominance. Proceeding with step I(b) and using the two equalities to eliminate the x_1 and x_2 variables results in the next LP.

Minimize

$$Z = -15.12x_3 - 15.2x_4 - 12.08x_5 + 5456$$

Subject to

$$\begin{pmatrix} 0.0014x_3 & + & 0.004x_4 & + & 0.0056x_5 & \leq & 0.32 \\ -0.0014x_3 & - & 0.004x_4 & - & 0.0056x_5 & \geq & 0.28 \\ 7.28x_3 & + & 11.8x_4 & + & 18.52x_5 & \leq & 564 \\ 7.28x_3 & + & 11.8x_4 & + & 18.52x_5 & \geq & 364 \\ 24.4x_3 & + & 26.0x_4 & + & 39.6x_5 & \leq & 1720 \\ 24.4x_3 & + & 26.0x_4 & + & 39.6x_5 & \geq & 1220 \end{pmatrix}$$

Continuing with step II of the method results in the GP formulation.

Minimize Z

Subject to

$$i) \quad \frac{1}{\left(K_1 Z^{\delta_1} x_3^{\delta_2} x_4^{\delta_3} x_5^{\delta_4}\right)} \leq 1$$

$$\text{where} \quad K_1 = \left[\left(\frac{1}{\delta_1}\right)^{\delta_1}\left(\frac{15.12}{\delta_2}\right)^{\delta_2}\left(\frac{15.2}{\delta_3}\right)^{\delta_3}\left(\frac{12.08}{\delta_4}\right)^{\delta_4}\right]$$

Constraint (i) above is an example of the induced constraint described in step II(a)(4).

Now proceed to the original constraints and their GP formulations.

a) $$\left(\frac{1}{.32}\right)\left(\frac{.0014x_3}{\delta_{a1}}\right)^{\delta_{a1}}\left(\frac{.004x_4}{\delta_{a2}}\right)^{\delta_{a2}}\left(\frac{.0056x_5}{\delta_{a3}}\right)^{\delta_{a3}} \leq 1$$

the lower bound on specific gravity.

b) $$-\frac{x_3}{200} \; -\frac{x_4}{70} \; -\frac{x_5}{50} \leq 1$$

the upper bound on specific gravity.

c) $$\left(\frac{1}{564}\right)\left(\frac{7.28x_3}{\delta_{b1}}\right)^{\delta_{b1}}\left(\frac{11.8x_4}{\delta_{b2}}\right)^{\delta_{b2}}\left(\frac{18.52x_5}{\delta_{b3}}\right)^{\delta_{b3}} \leq 1$$

the lower bound on color.

d) $$(364)\left(\frac{\delta_{b1}}{7.28x_3}\right)^{\delta_{b1}}\left(\frac{\delta_{b2}}{11.8x_4}\right)^{\delta_{b2}}\left(\frac{\delta_{b3}}{18.52x_5}\right)^{\delta_{b3}} \leq 1$$

the upper bound on color.

e) $$\left(\frac{1}{1720}\right)\left(\frac{24.4x_3}{\delta_{c1}}\right)^{\delta_{c1}}\left(\frac{26x_4}{\delta_{c2}}\right)^{\delta_{c2}}\left(\frac{39.6x_5}{\delta_{c3}}\right)^{\delta_{c3}} \leq 1$$

the lower bound on hop resin content.

f) $$(1220)\left(\frac{\delta_{c1}}{24.4x_3}\right)^{\delta_{c1}}\left(\frac{\delta_{c2}}{26x_4}\right)^{\delta_{c2}}\left(\frac{\delta_{c3}}{39.6x_5}\right)^{\delta_{c3}} \leq 1$$

the upper bound on hop resin content.

Proceed to step III and form the advanced GP sign table.

Var	Obj	i	a	b_1	b_2	b_3	c	d	e	f
Z	1	$-\delta_1$								
x_3		$-\delta_2$	δ_{a1}	-1			δ_{b1}	$-\delta_{b1}$	δ_{c1}	$-\delta_{c1}$
x_4		$-\delta_3$	δ_{a2}		-1		δ_{b2}	$-\delta_{b2}$	δ_{c2}	$-\delta_{c2}$
x_5		$-\delta_4$	δ_{a3}			-1	δ_{b3}	$-\delta_{b3}$	δ_{c3}	$-\delta_{c3}$

From the table it is obvious that besides the objective function, the induced constraint (i) is required to balance Z. This is always the case when using an induced constraint because of its linkage to the objective function. We can also see that at least one of constraints (a), (c), or (e) is required to balance the problem. Alternatively a combination of any two of the three remaining constraints could balance the problem. A quick check of the "three tight constraints" condition reveals that $(x_3^*, x_4^*, x_5^*) = (24.17, 390.84, -228.07)$ is not feasible. Note that at this point we have eliminated half of the original constraints from the problem.

We can now proceed to step IV. Constraints (a), (c), and (e)

can be generalized to the form below:

$$a)\quad d_1 x_3^{\delta_{a1}} x_4^{\delta_{a2}} x_5^{\delta_{a3}} = 1$$

$$b)\quad d_2 x_3^{\delta_{b1}} x_4^{\delta_{b2}} x_5^{\delta_{b3}} = 1$$

$$c)\quad d_3 x_3^{\delta_{c1}} x_4^{\delta_{c2}} x_5^{\delta_{c3}} = 1$$

Here we are using rule 4 on each constraint in order to apply algebra in determining which constraints are tight. Now take logarithms:

$$a)\quad \delta_{a1} \ln(x_3) + \delta_{a2} \ln(x_4) + \delta_{a3} \ln(x_5) = \ln\left(\frac{1}{d_1}\right)$$

$$c)\quad \delta_{b1} \ln(x_3) + \delta_{b2} \ln(x_4) + \delta_{b3} \ln(x_5) = \ln\left(\frac{1}{d_2}\right)$$

$$e)\quad \delta_{c1} \ln(x_3) + \delta_{c2} \ln(x_4) + \delta_{c3} \ln(x_5) = \ln\left(\frac{1}{d_3}\right)$$

Make the substitutions $y_i = \ln(x_i)$ and $b_i = 1/\ln(d_i)$.

$$a)\quad \delta_{a1} y_3 + \delta_{a2} y_4 + \delta_{a3} y_5 = b_1$$

$$c)\quad \delta_{b1} y_3 + \delta_{b2} y_4 + \delta_{b3} y_5 = b_2$$

$$e)\quad \delta_{c1} y_3 + \delta_{c2} y_4 + \delta_{c3} y_5 = b_3$$

This is a well known situation from linear algebra,

$$A\overrightarrow{X_i} = b$$

which, in this case, is equivalent to

$$\delta \overrightarrow{y_i} = b\,.$$

Using an initial solution of $(x_3, x_4, x_5) = (14, 14, 14)$ and making one iteration determines the following results:

$$\begin{bmatrix} \delta_{a1} \\ \delta_{a2} \\ \delta_{a3} \end{bmatrix} = \begin{bmatrix} .1273 \\ .3636 \\ .5091 \end{bmatrix} \quad \begin{bmatrix} \delta_{b1} \\ \delta_{b2} \\ \delta_{b3} \end{bmatrix} = \begin{bmatrix} .1936 \\ .3138 \\ .4926 \end{bmatrix} \quad \begin{bmatrix} \delta_{c1} \\ \delta_{c2} \\ \delta_{c3} \end{bmatrix} = \begin{bmatrix} .2711 \\ .2889 \\ .4400 \end{bmatrix}$$

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} .0344 \\ .0667 \\ .0523 \end{bmatrix} \quad \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 3.3704 \\ 2.7081 \\ 2.9503 \end{bmatrix}$$

$$\begin{bmatrix} y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} .0079 \\ .0467 \\ -9.3945 \end{bmatrix} \quad \begin{bmatrix} x_3' \\ x_4' \\ x_5' \end{bmatrix} = \begin{bmatrix} 35.4711 \\ 1.566 \times 10^9 \\ .0001 \end{bmatrix}.$$

It is obvious that x_5 is going to zero. Set $x_5 = 0$ and reformulate the problem. Since there are now only two potential non-zero variables there can be only two active constraints. Check the value of each constraint using the initial solution:

$$\begin{bmatrix} (a) \\ (b) \\ (c) \end{bmatrix} = \begin{bmatrix} .4381 \\ .9333 \\ .7326 \end{bmatrix} \le \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Since constraint (a) is the least tight eliminate it. The problem is now:

Minimize $$Z = -15.12x_3 - 15.2x_4 + 5456$$

Subject to

$$c)\quad 7.28x_3 + 11.8x_4 \le 564$$

$$e)\quad 24.4x_3 + 26.0x_4 \le 1720.$$

If both of these constraints were tight the solution would be

$$x_3 = 57.0965 \qquad x_4 = 12.5710 \qquad Z = 4401.6217$$

But if they are not both tight, one of the variables is not needed. Checking the extreme possibilities, either $x_3 = 0$ or $x_4 = 0$.

$$x_3 = 0 \qquad x_4 = 47.7996 \qquad Z = 4729.4915$$

$$x_3 = 70.4918 \qquad x_4 = 0 \qquad Z = 4390.1639$$

he problem is now solved. $x_3 = 70.4918$, $x_4 = 0$, $x_5 = 0$, and $Z = 4390.1639$. Use the equalities in the original problem to determine $x_2 = 19.6721$, and $x_1 = 9.8361$. This solution agrees with the LP solution.

GP techniques eliminated four of the problem's six inequality constraints. The problem was reduced to a point where algebra could be used to solve it.

5.3 Aviation Gasoline Blending

This problem (Symonds 1955) is a description of an actual refining process. Do not be deceived by the size of this problem. At the time of its publication, 1955, the largest problems being solved were 25 by 50 element matrices. They required 30 minutes per iteration on the best computers available. This is a real world example of the petrochemical blending problem.

It is desired to blend five refinery blending streams to the requirements of Grade 100/130 and Grade 115/145 aviation gasoline and obtain maximum profit.

Aircraft engines (piston type) are run under two in flight conditions: full power (at a rich mixture) for takeoff or climbing, and cruise power (at a lean mixture) for the most fuel efficient operation (Foster, T.R.V. 1978). The grade descriptions refer to octane rating tests run under severe conditions versus normal conditions (Leffler, W.L. 1985). It is known that three inspections are critical for each grade: 1) 1C + 4.6 lean octane PN (performance number), the normal (cruise) condition test; 2) 3C + 4.6 rich octane PN, the severe (full power) condition test; and 3) RVP (Reid vapor pressure).

The inspections used in this study are linear blending values (BL) including tolerances. The critical inspections, cost, and availability of the refinery streams are shown as follows:

Stream	BL PN 1C+4.6	BL PN 3C+4.6	BL RVP	Cost cpg	Avail bpd
No. 1	122.0	158.3	3.0	17.7	750
No. 2	109.0	145.0	4.0	16.0	800
No. 3	97.7	110.0	5.0	14.5	857
No. 4	90.0	125.0	7.0	13.0	1000
No. 5	114.0	145.0	19.0	11.0	Any

The specifications, refinery prices, and saleable volumes of the products are as follows:

Product	BL PN 1C+4.6	BL PN 3C+4.6	BL RVP	Price cpg	Amount bpd
M	100	130	7	16.0	2000
N	115	145	7	18.0	2000

In view of its low cost and high quality, it is obvious that the maximum amount of stream No. 5 should be used in the blends. Its use will be limited by the RVP specification of 7 on each grade. Thus, the RVP inspection and stream No. 5 can be eliminated from the problem by blending streams 1 thru 4 up to 7 RVP with stream 5. This results in the following preliminary blends:

Input Stream	BL PN 1C+4.6	BL PN 3C+4.6	BL RVP	Cost cpg	Avail bpd
A	120.0	155.0	7	16.0	1000
B	110.0	145.0	7	15.0	1000
C	100.0	115.0	7	14.0	1000
D	90.0	125.0	7	13.0	1000

This combination of inputs and products results in the following LP:

Maximize

$$Z = 2x_{an} + x_{bm} + 3x_{bn} + 2x_{cm} + 4x_{cn} + 3x_{dm} + 5x_{dn}$$

Subject to

$a)\quad x_{am} + x_{bm} + x_{cm} + x_{dm} \leq 2000$

the upper limit on the M-product stream.

$b)\quad x_{an} + x_{bn} + x_{cn} + x_{dn} \leq 2000$

the upper limit on the N-product stream.

$c)\quad x_{am} + x_{an} \leq 1000$

the upper limit on the A-blending stock.

$d)\quad x_{bm} + x_{bn} \leq 1000$

the upper limit on the B-blending stock.

e) $x_{cm} + x_{cn} \leq 1000$

the upper limit on the C-blending stock.

f) $x_{dm} + x_{dn} \leq 1000$

the upper limit on the D-blending stock.

g) $20x_{am} + 10x_{bm} - 10x_{dm} \geq 0$

the M-product severe condition octane rating.

h) $25x_{am} + 15x_{bm} - 15x_{cm} - 5x_{dm} \geq 0$

the M-product normal condition octane rating.

i) $5x_{an} - 5x_{bn} - 15x_{cn} - 25x_{dn} \geq 0$

the N-product severe condition octane rating.

j) $10x_{an} - 30x_{cn} - 20x_{dn} \geq 0$

the N-product normal condition octane rating.

It is illustrative to examine this problem in standard LP format.

Variable	x_{am}	x_{an}	x_{bm}	x_{bn}	x_{cm}	x_{cn}	x_{dm}	x_{dn}		
Maximize:		2	1	3	2	4	3	5		
a	1		1		1		1		≤	2000
b		1		1		1		1	≤	2000
c	1	1							≤	1000
d			1	1					≤	1000
e					1	1			≤	1000
f							1	1	≤	1000
g	20		10				-10		≥	0
h	25		15		-15		-5		≥	0
i		5		-5		-15		-25	≥	0
j		10				-30		-20	≥	0

The problem's corresponding advanced GP sign table is shown below:

		Constraints										
Var	Obj	I	a	b	c	d	e	f	g	h	i	j
Z	-1	1										
x_{am}			δ_{a1}		δ_{c1}				$-\delta_{g1}$	$-\delta_{h1}$		
x_{an}		$-\delta_1$		δ_{b1}	δ_{c2}						-1	-1
x_{bm}		$-\delta_2$	δ_{a2}			δ_{d1}			$-\delta_{g2}$	$-\delta_{h2}$		
x_{bn}		$-\delta_3$		δ_{b2}		δ_{d2}					δ_{i1}	
x_{cm}		$-\delta_4$	δ_{a3}				δ_{e1}			δ_{h3}		
x_{cn}		$-\delta_5$		δ_{b3}			δ_{e2}				δ_{i2}	δ_{j1}
x_{dm}		$-\delta_6$	δ_{a4}					δ_{f1}	1	δ_{h4}		
x_{dn}		$-\delta_7$		δ_{b4}				δ_{f2}			δ_{i3}	δ_{j2}

Looking first at the LP format of the problem, we use step I(a) of the method, to note that there is no obvious dominance. We further note that there are no equalities to use for step I(b).

As in the two previous examples, step II is used to formulate the GP. We then use step III of the method to build the Advanced GP sign table presented above. Checking step III(a) we note that the sign of each entry is obvious. Looking at this table we now check that every row has at least one positive and one negative entry. As this is true we conclude that all variables are balanced and, as is always the case, the objective function and its induced constraint are required. This fulfills the requirements of step III(b). Since there are no required constraints, we can skip step III(c).

We now proceed with step III(d) and apply engineering knowledge, and mathematical fact, to the problem. All the constraints which remain at this point are potentially required (none that balance the same side of a variable as the objective function remain). Note, in this example we have not eliminated any constraints yet. The knowledge and fact we are referring to is stated simply as: the tighter all blending constraints are held, the closer the solution is to the optimum.

To accomplish step III(d) we examine the blending constraints in the sign table. In this case the rightmost four columns. We are looking for either only one positive or one negative entry in a column. If any group of constraints share this entry (the variable's GP sign) and all other variables including the product, they must be compared as equalities to establish dominance. We will now proceed with the example.

Using step III(d), note constraint (i), x_{an} is the one blending stock that can bring in the (blending constraint) $1C+4.6$ lean octane specification (i) for the N-product stream. The x_{bn} variable can meet constraint (j) by itself but can not bring in any other blending stock to the product specification. Note, in blending constraint (i), one unit of x_{an} balances nine units of other inputs to this constraint. In blending constraint (j), one unit of x_{an} balances five units of other inputs. Constraint (i) is harder to satisfy than constraint (j), which is our definition of dominance. Constraint (i) is therefore retained and constraint (j) is dropped from further analysis. Because the variable x_{am} does not help our objective function but does inversely effect x_{an} we know that to maximize the objective function and x_{an} we need to minimize x_{am}. Therefore constraint (i) is tight and the quantity of x_{an} used should be maximized to make constraint (c) tight with $x_{an} = 1000$ bpd and $x_{am} = 0$ bpd.

The problem has been reduced by three constraints and two variables at this point. The remaining problem's advanced GP sign table is shown below:

		Constraints							
Var	Obj	I	a	b	d	e	f	g	h
Z	-1	1							
x_{bm}		$-\delta_2$	δ_{a2}		δ_{d1}			$-\delta_{g2}$	$-\delta_{h2}$
x_{bn}		$-\delta_3$		δ_{b2}	δ_{d2}				
x_{cm}		$-\delta_4$	δ_{a3}			δ_{e1}			δ_{h3}
x_{cn}		$-\delta_5$		δ_{b3}		δ_{e2}			
x_{dm}		$-\delta_6$	δ_{a4}				δ_{f1}	1	δ_{h4}
x_{dn}		$-\delta_7$		δ_{b4}			δ_{f2}		

Note constraint (g), x_{dm} is the one product stream that can bring in the blending constraint $1C + 4.6$ lean octane specification for the M-product stream. Therefore constraint (g) is tight and since $x_{am} = 0$ bpd, $x_{bm} = x_{dm}$. The last statement means that x_{cm} is the one product which can bring in constraint (h) making it tight and $x_{cm} = 2/3\, x_{bm}$.

This reduces the problem by two constraints, and defines the M-product stream components in specific ratios. The remaining GP advanced sign table follows:

		Constraints					
Var	Obj	I	*a*	*b*	*d*	*e*	*f*
Z	-1	1					
x_{bm}		$-\delta_2$	δ_{a2}		δ_{d1}		
x_{bn}		$-\delta_3$		δ_{b2}	δ_{d2}		
x_{cm}		$-\delta_4$	δ_{a3}			δ_{e1}	
x_{cn}		$-\delta_5$		δ_{b3}		δ_{e2}	
x_{dm}		$-\delta_6$	δ_{a4}				δ_{f1}
x_{dn}		$-\delta_7$		δ_{b4}			δ_{f2}

Because the variables x_{bm}, x_{cm}, and x_{dm} are specified by the above ratios, we can maximize the M-product stream by making constraint (a) tight. This defines $M_{max} = 2000$ bpd at $x_{am} = 0$ bpd, $x_{bm} = 750$ bpd, $x_{cm} = 500$ bpd, and $x_{dm} = 750$ bpd.

To now maximize the N-product stream under these conditions we use the previously determined equality in constraint (i). This equality tells us that to get the most volume on the N-product stream we should use as much x_{bn} as possible followed by as much x_{cn} as possible and if any x_{an} is left to be balanced to use x_{dn} as a last resort to balance it. This prioritization of variables makes constraint (d) tight which defines $x_{bn} = 250$ bpd. The equality (i) is fulfilled when $x_{cn} = 250$ bpd, and $x_{dn} = 0$ bpd. This makes the N-product stream a maximum at $N_{max} = 1500$ bpd. The problem is now fully solved. The objective function: $Z = 7750$ bpd. This is equivalent to the LP solution.

Aside from the above solution, independent LP solutions were run at each iteration of constraint elimination. These runs correspond to each of the GP sign tables given above. The results of these runs are presented below.

	LP solution run			
	1	2	3	4
Constraints	10	7	5	3
Variables	8	6	6	4
LP iterations	11	7	5	2
LP solution time (sec)	3	2.5	2	1.5
Constraints eliminated	c, i, j	g, h	a, d	
Variables eliminated	x_{am}, x_{an}	none	x_{dn}, x_{cn}	

Our results confirm a monotonically decreasing number of LP iterations and running time for this problem. With this problem we have demonstrated the fundamental task of this dissertation, given in the abstract as: *The algorithm can select from all possible constraints those which truly influence the problem, effectively reducing the size of these problems and decreasing the solution time.*

A comparison of the original LP and the LP consisting of just the tight constraints determined above was run. It resulted in 11 iterations versus 9 iterations respectively. However, by throwing in the zero variables, the LP is reduced to a set of six simultaneous equations in six variables. This can be inverted quite efficiently by most modern scientific calculators. More importantly, a similar result on a much larger problem could be just as easily inverted with the aid of a mathematical programming system. This example also demonstrates the second fundamental task of this dissertation, which was also given in the abstract as: *We assert that this means the solution of these problems can often be reduced from a linear program to the inversion of a square matrix.*

5.4 Refinery Optimization

This example is a simplified linear model of refinery operation (Winston 1987). Three products are produced: heating oil, gasoline, and jet fuel. Average octane levels must be at least 4.5 for heating oil, 8.5 for gasoline, and 7.0 for jet fuel. Two types of oil are used as feed stock: crude #1 ($12/Bbl), and crude #2 ($10/Bbl). Up to 10,000 Bpd of each crude is available.

The crude must be distilled before it can be used to produce product. Distillation capacity is 15,000 Bpd ($0.10/Bbl). After distillation crude #1 yields 0.6 Bbl of naptha, 0.3 Bbl of distilled #1, and 0.1 Bbl of distilled #2. Crude #2 yields 0.4 Bbl of naptha, 0.2 Bbl of distilled #1, and 0.4 Bbl of distilled #2.

Naptha can only be used to produce gasoline or jet fuel. Distilled oil can be used to produce heating oil, or it can be processed by the catalytic cracker ($0.15/Bbl). Cat cracker capacity is 5,000 Bpd. After cracking distilled #1 yields 0.8 Bbl of cracked #1, and 0.2 Bbl of cracked #2. Distilled #2 yields 0.7 Bbl of cracked #1, and 0.3 Bbl of cracked #2. Cracked oil can be used to produce gasoline or jet fuel but not heating oil.

The octane level of each type of oil is naptha - 8, distilled #1 - 4, distilled #2 - 5, cracked #1 - 9, cracked #2 - 6.

The heating oil can be sold at \$14/Bbl; gasoline, \$18/Bbl; and jet fuel, \$16/Bbl. Contracts require that at least 3,000 Bpd of each product be produced. Our goal is to maximize the refiner's profit.

This extremely simplified description of a refinery operation results in the following model.

Maximize

$$Z = -12.1x_1 - 10.1x_2 - .15x_3 - .15x_4 + 14x_5 + 18x_6 + 16x_7$$

Subject to

1) $x_1 + x_2 \leq 15000$

the distillation capacity.

2) $x_3 + x_4 \leq 5000$

the catalytic convertor capacity.

3) $x_6 \geq 3000$

the gasoline contract requirement.

4) $x_5 \geq 3000$

the heating oil contract requirement.

5) $x_7 \geq 3000$

the jet fuel contract requirement.

6) $x_1 \leq 10000$

the crude 1 supply.

7) $x_2 \leq 10000$

the crude 2 supply.

8) $-0.5x_8 + 0.5x_9 - 2.5x_{10} \geq 0$

the gasoline octane requirement.

9) $-0.5x_{11} + 0.5x_{12} \geq 0$

the heating oil octane requirement.

10) $x_{13} + 2x_{14} - x_{15} > 0$

the jet fuel octane requirement.

11) $x_7 - x_{13} - x_{14} - x_{15} = 0$

the jet fuel mass balance.

12) $x_6 - x_8 - x_9 - x_{10} = 0$

the gasoline mass balance.

13) $x_5 - x_{11} - x_{12} = 0$

the heating oil mass balance.

14) $-0.6x_1 - 0.4x_2 + x_8 + x_{13} = 0$

the naptha mass balance.

15) $-0.8x_3 - 0.7x_4 + x_9 + x_{14} = 0$

the cracked 1 mass balance.

16) $-0.2x_3 - 0.3x_4 + x_{10} + x_{15} = 0$

the cracked 2 mass balance.

17) $-0.3x_1 - 0.2x_2 + x_3 + x_{11} = 0$

the distillate 1 mass balance.

18) $-0.1x_1 - 0.4x_2 + x_4 + x_{12} = 0$

the distillate 2 mass balance.

Eliminating variables by using the equalities is not efficient for this problem. Instead use step I(d) and replace them with inequalities. Now form the advanced GP sign table.

The Advanced GP Sign Table (a)

	i	1	2	3	4	5	6	7
x_1	1	δ_{a1}					1	
x_2	1	δ_{a2}						1
x_3	1		δ_{b1}					
x_4	1		δ_{b2}					
x_5	$-\delta_1$				-1			
x_6	$-\delta_2$			-1				
x_7	$-\delta_3$					-1		
x_8								
x_9								
x_{10}								
x_{11}								
x_{12}								
x_{13}								
x_{14}								
x_{15}								

The Advanced GP Sign Table (b)

	8	9	10	11*a*	11*b*	12*a*	12*b*	13*a*	13*b*
x_1									
x_2									
x_3									
x_4									
x_5								-1	1
x_6						-1	1		
x_7				-1	1				
x_8	δ_{h1}					δ_{l1}	$-\delta_{l1}$		
x_9	-1					δ_{l2}	$-\delta_{l2}$		
x_{10}	δ_{h2}					δ_{l3}	$-\delta_{l3}$		
x_{11}		1						δ_{m1}	$-\delta_{m1}$
x_{12}		-1						δ_{m2}	$-\delta_{m2}$
x_{13}			$-\delta_{j1}$	δ_{k1}	$-\delta_{k1}$				
x_{14}			$-\delta_{j2}$	δ_{k2}	$-\delta_{k2}$				
x_{15}			1	δ_{k3}	$-\delta_{k3}$				

The Advanced GP Sign Table (c)

	$14a$	$14b$	$15a$	$15b$	$16a$	$16b$	$17a$	$17b$	$18a$	$18b$
x_1	δ_{n3}	$-\delta_{n3}$					δ_{q3}	$-\delta_{q3}$	δ_{r3}	$-\delta_{r3}$
x_2	δ_{n4}	$-\delta_{n4}$					δ_{q4}	$-\delta_{q4}$	δ_{r4}	$-\delta_{r4}$
x_3			δ_{o3}	$-\delta_{o3}$	δ_{p3}	$-\delta_{p3}$	$-\delta_{q1}$	δ_{q1}		
x_4			δ_{o4}	$-\delta_{o4}$	δ_{p4}	$-\delta_{p4}$			$-\delta_{r1}$	δ_{r1}
x_5										
x_6										
x_7										
x_8	$-\delta_{n1}$	δ_{n1}								
x_9			$-\delta_{o1}$	δ_{o1}						
x_{10}					$-\delta_{p1}$	δ_{p1}				
x_{11}							$-\delta_{q2}$	δ_{q2}		
x_{12}									$-\delta_{r2}$	δ_{r2}
x_{13}	$-\delta_{n2}$	δ_{n2}								
x_{14}			$-\delta_{o2}$	δ_{o2}						
x_{15}					$-\delta_{p2}$	δ_{p2}				

Several points can be noted from the sign table.

1. Conditions (a) or (b) of constraints 11-18 (the equalities) must be tight !
2. Terms x_{13}, x_{14}, and x_{15} are balanced between equalities 11, 14, 15, and 16, or constraint 10 is tight.
3. Terms x_{11}, and x_{12} are balanced between equalities 13, 17, and 18, or constraint 9 is tight.
4. Terms x_8, x_9, and x_{10} are balanced between equalities 12, 14, 15, and 16, or constraint 8 is tight.

We will now solve the problem using the advanced sign table and the algorithm.

A. Examining the last statement calls attention to the variable x_9 (cracker #1 gas). It is the one stream which can bring in the gasoline octane specification. The supply of x_9 then upper bounds (limits) the amount of x_6, gasoline, the most profitable product of the problem. Constraint 8 (the gasoline octane requirement) is therefore tight ! It also implies that x_{10} (cracker #2 gas) should be minimized. Therefore $x_{10} = 0$. We can maximize x_9 by minimizing $x_{14} = 0$ (cracker #1 jet) in equality 15 (cracker #1 balance).

B. Equality 15 (cracker #1 balance) then lower bounds the x_9

(cracker #1 gas) volume and upper bounds variables x_3, and x_4 (distillate #1 & #2) volumes. This makes constraint 2 (cracker capacity) tight !

C. Equalities 17 and 18 (distillate #1 & #2 balances) then lower bound the x_3, and x_4 (distillate #1 & #2) volumes and upper bound the x_1, and x_2 (crude #1 & #2) volumes. This forces constraint 1 (distillation capacity) to be tight !

D. Crude #2 which is cheaper than crude #1 and renders more volume to cracker #1 (which produces the most gasoline) is the preferred raw material. Constraint 7 (crude #2 available) is therefore tight !

E. Equalities 17 and 18 (distillation column 1 & 2 balances) also lower bound x_{11}, and x_{12} (distillate #1 heating oil, and distillate #2 heating oil) volumes. Equality 13 (heating oil balance) then upper bounds x_{11}, and x_{12} volumes by lower bounding x_5 (heating oil) volume. This makes constraint 4, (the heating oil contract) an upper bound and tight !

The problem can now be solved from the above determinations:

crude #2: $x_2 = 10000$

cracked #2 gas: $x_{10} = 0$

crude #1: $x_1 = 15000 - x_2 = 5000$

heating oil: $x_5 = 3000$

distillate #1: $x_3 = .3x_1 + .2x_2 = 3500$

distillate #2: $x_4 = 5000 - x_3 = 1500$

cracked #1 jet: $x_{14} = 0$

cracked #1 gas: $x_9 = .8x_3 + .7x_4 - 0 = 3850$

naptha gas: $x_8 = x_9 = 3850$

gasoline: $x_6 = x_8 + x_9 + x_{10} = 7700$

naptha jet: $x_{13} = .6x_1 + .4x_2 - x_8 = 3150$

cracked #2 jet: $x_{15} = .2x_3 + .3x_4 - x_{10} = 1150$

distillate #1 heating oil: $x_{11} = .3x_1 + .2x_2 - x_3 = 0$

distillate #2 heating oil: $x_{12} = .1x_1 + .4x_2 - x_4 = 3000$

jet fuel: $x_7 = x_{13} + x_{14} + x_{15} = 4300$

These results agree with the LP results. Another method of solution would be to return to the LP code with just the tight constraints we identified and the equalities. The advantage is that the number of inequality constraints (10) has been reduced to 5 equalities. The number of iterations required is reduced from 23 to 16 (30%). The GP technique of the advanced sign table delivered a systematic identification of the required constraints to solve the problem.

This problem is an excellent example of the power of the method. Although it is a highly simplified example of a refinery, the problem presents a sophisticated model to the student. This model has 15 variables and, after reformulating as a GP, 19 terms. This structure is termed a three degree of difficulty problem. Mathematically, one degree of difficulty is an order of magnitude greater problem to solve than a zero degree of difficulty problem. A zero degree of difficulty linear problem would be a set of simultaneous equations with an equal number of variables.

Using only the A-G Inequality and the GP sign table (steps II and III) it is possible to solve the problem by inspection.

5.5 Nonlinear Gasoline Blending

In this subsection we will, as promised in the abstract, extend the previous methodology to a nonlinear refining problem. To our knowledge, this type of problem has no defined methodology of solution in the literature.

As noted in chapters 1 and 2, some hydrocarbon properties do not blend linearly. One of these properties is the octane rating used when blending gasoline. This example is real, the gasoline produced using the resulting recipe is a usable blend any of us could use in our own vehicles, diesels excepted. The example demonstrates the concepts of this method's application to these types of problems. This example was supplied by the Lake Charles, Louisiana refinery of Citgo Petroleum Corp.

A refinery produces the following streams:

Input Stream	Octane	RVP	Value cpg	Volume KBpd
Cat Gasoline	86	9.5	90	46.0
Reformate	95	6.0	93	50.0
Alkylate	92	8.5	87	15.0
Natural Gasoline	72	11.0	75	8.5
Normal Butane	84	51.0	60	20.0

These five streams are to be used in blending two types of gasoline at 87 and 93 octane. The characteristics of these products are given below.

Output Stream	Octane	RVP	Value cpg
87 Unleaded	87	9.0	95
93 Unleaded	93	9.0	100

Define the following variables:

x_1: Cat/93	x_2: Cat/87
x_3: Ref/93	x_4: Ref/87
x_5: Alk/93	x_6: Alk/87
x_7: Nat/93	x_8: Nat/87
x_9: But/93	x_{10}: But/87

The model (NLP) is formulated as the previous LP examples were, with the exception of the cross product terms found in the octane constraints. The coefficients on these terms (interaction coefficients) are determined experimentally and are, in some cases, proprietary (Baird, C.T. 1989).

The nonlinear blending problem is defined as follows:

Maximize

$$Z = 10x_1 + 5x_2 + 7x_3 + 2x_4 + 13x_5 + 8x_6 + 25x_7 + 20x_8 + 40x_9 + 35x_{10}$$

Subject to

Constraint (1): 93 Unleaded Octane Balance

$$\begin{pmatrix} -7x_1 & + & 2x_3 & - & x_5 & - & 21x_7 \\ -9x_9 & + & .67x_1x_3 & - & .98x_1x_5 & + & 8.3x_1x_7 \\ +4.2x_1x_9 & + & .32x_3x_5 & + & 10.2x_3x_7 & + & 5.7x_3x_9 \\ -1.7x_5x_7 & - & .9x_5x_9 & + & .05x_7x_9 & & \end{pmatrix} \geq 0$$

Constraint (2): 87 Unleaded Octane Balance

$$\begin{pmatrix} -x_2 & + & 8x_4 & + & 5x_6 & - & 15x_8 \\ -3x_{10} & + & .67x_2x_4 & - & .98x_2x_6 & + & 8.3x_2x_8 \\ +4.2x_2x_{10} & + & .32x_4x_6 & + & 10.2x_4x_8 & + & 5.7x_4x_{10} \\ -1.7x_6x_8 & - & .9x_6x_{10} & + & .05x_8x_{10} & & \end{pmatrix} \geq 0$$

Constraint (3): 93 Unleaded Reid Vapor Pressure Balance

$$.5x_1 - 3x_3 - .5x_5 + 2x_7 + 42x_9 \leq 0$$

Constraint (4): 87 Unleaded Reid Vapor Pressure Balance

$$.5x_2 - 3x_4 - .5x_6 + 2x_8 + 42x_{10} \leq 0$$

Volume Constraints:

(5) $x_1 + x_2 = 46$ 1000 bpd

(6) $x_3 + x_4 = 50$ 1000 bpd

(7) $x_5+x_6=15$ 1000 bpd

(8) $x_7+x_8 \leq 8.5$ 1000 bpd

(9) $x_9+x_{10} \leq 20$ 1000 bpd

A feasible solution to this program can be found using LP. In doing this, the cross product terms in the nonlinear constraints are ignored.

This results in the reduced problem

Maximize

$$Z = 10x_1+5x_2+7x_3+2x_4+13x_5+8x_6+25x_7+20x_8+40x_9+35x_{10}$$

Subject to

Constraint (1): 93 Unleaded Octane Balance

$$-7x_1+2x_3-x_5-21x_7-9x_9 \geq 0$$

Constraint (2): 87 Unleaded Octane Balance

$$-x_2+8x_4+5x_6-15x_8-3x_{10} \geq 0$$

Constraint (3): 93 Unleaded Reid Vapor Pressure Balance

$$.5x_1-3x_3-.5x_5+2x_7+42x_9 \leq 0$$

Constraint (4): 87 Unleaded Reid Vapor Pressure Balance

$$.5x_2-3x_4-.5x_6+2x_8+42x_{10} \leq 0$$

Volume Constraints:

(5) $x_1 + x_2 = 46$ 1000 bpd

(6) $x_3 + x_4 = 50$ 1000 bpd

(7) $x_5 + x_6 = 15$ 1000 bpd

(8) $x_7 + x_8 \leq 8.5$ 1000 bpd

(9) $x_9 + x_{10} \leq 20$ 1000 bpd

The problem is identical to the NLP with the exception of the octane constraints which contain only linear terms. This solution (all volumes are in 1000 bpd) is

$x_1 =$	3.1979	$x_2 =$	42.8021
$x_3 =$	28.4786	$x_4 =$	21.5214
$x_5 =$	15.0000	$x_6 =$	0.0000
$x_7 =$	0.0000	$x_8 =$	8.5000
$x_9 =$	2.1747	$x_{10} =$	0.6229

This solution is feasible for the NLP and produces an objective function value of $Z = 962.1726$. Multiplied by a factor of 42000 $gpd/100\ cp\$$, supplied by N. Goddard of Citgo, gives a predicted daily profit of $404,112. This technique is currently used in designing this type of blending process. An extensive search of the available petroleum industry literature showed that current available NLP codes have not been demonstrated capable of optimizing this type of NLP. Refinery operators know that better results are possible owing to the nonlinearities of the real operation. The proof of this is that these operators have consistently produced higher profits that those predicted by the LP solution. They use field measurements to aid in improving their operations of this type. We will now demonstrate that this method can optimize this type of NLP problem.

There are no pairs of constraints from which to judge dominance, so proceed with step I(b) and eliminate the x_2, x_4, and x_6 variables from the problem. The reformulated problem is as follows:

Maximize

$$5x_1+5x_3+5x_5+25x_7+20x_8+40x_9+35x_{10}+450$$

Subject to

(1)
$$\begin{pmatrix} -7x_1 & + & 2x_3 & - & x_5 & - & 21x_7 \\ -9x_9 & + & .67x_1x_3 & - & .98x_1x_5 & + & 8.3x_1x_7 \\ +4.2x_1x_9 & + & .32x_3x_5 & + & 10.2x_3x_7 & + & 5.7x_3x_9 \\ -1.7x_5x_7 & - & .9x_5x_9 & + & .05x_7x_9 & & \end{pmatrix} \geq 0$$

(2) $$\begin{pmatrix} -47.2x_1 & - & 43.62x_3 & - & 66.08x_5 & + & 851.3x_8 \\ +461.7x_{10} & + & .67x_1x_3 & - & .98x_1x_5 & - & 8.3x_1x_8 \\ -4.2x_1x_{10} & + & .32x_3x_5 & - & 10.2x_3x_8 & - & 5.7x_3x_{10} \\ +1.7x_5x_8 & + & .9x_5x_{10} & + & .05x_8x_{10} & + & 1533.8 \end{pmatrix} \geq 0$$

(3) $$.5x_1 - 3x_3 - .5x_5 + 2x_7 + 42x_9 \leq 0$$

(4) $$-.5x_1 + 3x_3 + .5x_5 + 2x_8 + 42x_{10} - 134.5 \leq 0$$

(8) $$x_7 + x_8 \leq 8.5$$

(9) $$x_9 + x_{10} \leq 20$$

Apply step II to reformulate the problem as a GP. Next apply step III and form the advanced GP sign table.

Var	3	4	5	6	10	11
x_1	$\begin{pmatrix} \delta_{a1} & + & \delta_{a5} \\ -\delta_{a9} & - & \delta_{a10} \\ -\delta_{a11} & & \end{pmatrix}$	$\begin{pmatrix} \delta_{b1} & + & \delta_{b4} \\ +\delta_{b5} & + & \delta_{b6} \\ -\delta_{b11} & & \end{pmatrix}$	δ_{c1}			
x_3	$\begin{pmatrix} -\delta_{a8} & - & \delta_{a9} \\ -\delta_{a12} & - & \delta_{a13} \\ -\delta_{a14} & & \end{pmatrix}$	$\begin{pmatrix} \delta_{b2} & + & \delta_{b7} \\ +\delta_{b8} & - & \delta_{b11} \\ -\delta_{b12} & & \end{pmatrix}$	$-\delta_{c4}$	δ_{d1}		
x_5	$\begin{pmatrix} \delta_{a2} & + & \delta_{a5} \\ +\delta_{a6} & + & \delta_{a7} \\ -\delta_{a12} & & \end{pmatrix}$	$\begin{pmatrix} \delta_{b3} & + & \delta_{b4} \\ -\delta_{b12} & - & \delta_{b13} \\ -\delta_{b14} & & \end{pmatrix}$	$-\delta_{c5}$	δ_{d2}		
x_7	$\begin{pmatrix} \delta_{a3} & + & \delta_{a6} \\ -\delta_{a10} & - & \delta_{a13} \\ -\delta_{a15} & & \end{pmatrix}$		δ_{c2}		δ_{h1}	
x_9	$\begin{pmatrix} \delta_{a4} & + & \delta_{a7} \\ -\delta_{a11} & - & \delta_{a14} \\ -\delta_{a15} & & \end{pmatrix}$		δ_{c3}			δ_{i1}
x_8		$\begin{pmatrix} \delta_{b5} & + & \delta_{b7} \\ -\delta_{b9} & - & \delta_{b13} \\ -\delta_{b15} & & \end{pmatrix}$		δ_{d3}	δ_{h2}	
x_{10}		$\begin{pmatrix} \delta_{b6} & + & \delta_{b8} \\ -\delta_{b10} & - & \delta_{b14} \\ -\delta_{b15} & & \end{pmatrix}$		δ_{d4}		δ_{i2}

Because there are multiple bipolar exponents operating on individual variables within terms, a solution (not necessarily feasible), see step III(a'), is necessary to evaluate the advanced GP sign table. Using the previously described linear solution (not optimal) to the problem for this purpose shows that term 6 (on x_3) is essential. Term 6 is constraint 4, thus constraint 4 is tight !

Using the equality $x_1 = 6x_3 + x_5 + 4x_8 + 84x_{10} - 269$, results in the following reformulated problem:

Maximize

$$35x_3 + 10x_5 + 25x_7 + 40x_8 + 40x_9 + 455x_{10} - 895$$

Subject to

(1)

$$\begin{pmatrix} -136.23x_3 & + & 269.62x_5 & - & 2253.7x_7 & + & 28x_8 \\ -1138.8x_9 & + & 588x_{10} & + & 4.02x_3^2 & - & .98x_5^2 \\ -4.89x_3x_5 & + & 60x_3x_7 & + & 2.68x_3x_8 & + & 30.9x_3x_9 \\ +56.28x_3x_{10} & + & 6.6x_5x_7 & - & 3.92x_5x_9 & + & 3.3x_5x_9 \\ -82.32x_5x_{10} & + & 33.2x_7x_8 & + & .05x_7x_9 & + & 697.2x_7x_{10} \\ +16.8x_8x_9 & + & 352.8x_9x_{10} & - & 1883.0 & & \end{pmatrix} \geq 0$$

(2)

$$\begin{pmatrix} 14230.6 & - & 507.05x_3 & + & 150.34x_5 & + & 2895.2x_8 \\ -2373.3x_{10} & + & 4.02x_3^2 & - & .98x_5^2 & - & 33.2x_8^2 \\ -352.8x_{10}^2 & - & 4.89x_3x_5 & - & 57.32x_3x_8 & + & 25.38x_3x_{10} \\ -10.52x_5x_8 & - & 85.62x_5x_{10} & - & 713.95x_8x_{10} & & \end{pmatrix} \geq 0$$

(3) $2x_7 + 2x_8 + 42x_9 + 42x_{10} - 134.5 \leq 0$

(8) $x_7 + x_8 \leq 8.5$

(9) $x_9 + x_{10} \leq 20$

There are no dominant constraints or equalities so proceed with steps II and III again. The reformulated GP produces the next advanced sign table:

	3	4	5	10	11
x_3	$\begin{pmatrix} \delta_{a1} & + & \delta_{a4} & & \\ -2\delta_{a11} & - & \delta_{a12} & & \\ -\delta_{a13} & - & \delta_{a14} & - & \delta_{a15} \end{pmatrix}$	$\begin{pmatrix} \delta_{b1} & + & \delta_{b3} \\ +\delta_{b4} & - & 2\delta_{b13} \\ -\delta_{b14} & & \end{pmatrix}$			
x_5	$\begin{pmatrix} \delta_{a4} & + & 2\delta_{a5} & & \\ +\delta_{a6} & + & \delta_{a7} & & \\ -\delta_{a8} & - & \delta_{a15} & - & \delta_{a19} \end{pmatrix}$	$\begin{pmatrix} \delta_{b3} & + & 2\delta_{b5} \\ +\delta_{b6} & + & \delta_{b7} \\ -\delta_{b11} & & \end{pmatrix}$			
x_7	$\begin{pmatrix} \delta_{a2} & - & \delta_{a14} \\ -\delta_{a15} & - & \delta_{a16} \\ -\delta_{a17} & - & \delta_{a22} \end{pmatrix}$		δ_{c1}	δ_{h1}	
x_8	$\begin{pmatrix} \delta_{a6} & - & \delta_{a9} \\ -\delta_{a12} & - & \delta_{a16} \\ -\delta_{a20} & & \end{pmatrix}$	$\begin{pmatrix} \delta_{b4} & + & \delta_{b6} \\ +2\delta_{b8} & + & \delta_{b9} \\ -\delta_{b12} & & \end{pmatrix}$	δ_{c2}	δ_{h2}	
x_9	$\begin{pmatrix} \delta_{a3} & - & \delta_{a18} \\ -\delta_{a19} & - & \delta_{a20} \\ -\delta_{a21} & - & \delta_{a22} \end{pmatrix}$		δ_{c3}		δ_{i1}
x_{10}	$\begin{pmatrix} \delta_{a7} & - & \delta_{a10} \\ -\delta_{a13} & - & \delta_{a17} \\ -\delta_{a21} & & \end{pmatrix}$	$\begin{pmatrix} \delta_{b2} & + & \delta_{b7} \\ +\delta_{b9} & + & 2\delta_{b10} \\ -\delta_{b14} & & \end{pmatrix}$	δ_{c4}		δ_{i2}

Examination of this table again requires a solution to evaluate the multiple bipolar exponents affecting all of the variables. Using the previous linear solution to do this shows that x_7 needs term 3 and either 5 or 10 to balance. Term 10, which is constraint (8), is by far easier to work with (term 10 has 2 variables versus the 4 in term 5) so use it. This means constraint (8) is tight ! Use the equality $x_7 = 8.5 - x_8$ to reformulate the problem as follows:

Maximize

$$35x_3 + 10x_5 + 15x_8 + 40x_9 + 455x_{10} - 682.5$$

Subject to

(1)

$$\begin{pmatrix} 373.77x_3 & + & 325.72x_5 & + & 2563.9x_8 & - & 1138.375x_9 \\ +6514.2x_{10} & + & 4.02x_3^2 & - & .98x_5^2 & - & 33.2x_8^2 \\ -4.89x_3x_5 & - & 57.32x_3x_8 & + & 30.9x_3x_9 & + & 56.28x_3x_{10} \\ -10.52x_5x_8 & + & 3.3x_5x_9 & - & 82.32x_5x_{10} & + & 16.75x_8x_9 \\ -697.2x_8x_{10} & + & 352.8x_9x_{10} & - & 21039.45 & & \end{pmatrix} \geq 0$$

(2)

$$\begin{pmatrix} 14230.6 & - & 507.05x_3 & + & 150.34x_5 & + & 2895.2x_8 \\ -2373.3x_{10} & + & 4.02x_3^2 & - & .98x_5^2 & - & 33.2x_8^2 \\ -352.8x_{10}^2 & - & 4.89x_3x_5 & - & 57.32x_3x_8 & + & 25.38x_3x_{10} \\ -10.52x_5x_8 & - & 85.62x_5x_{10} & - & 713.95x_8x_{10} & & \end{pmatrix} \geq 0$$

(3) $42x_9 + 42x_{10} - 117.5 \leq 0$

(9) $x_9 + x_{10} \leq 20$

Returning to step I(a), we find that constraint (3) dominates constraint (9). We can drop constraint (9) from the problem. Continuing with steps II and III, we obtain the next advanced sign table:

	3	4	5
x_3	$\left(\begin{array}{ccc} \delta_{a4} & + & \delta_{a5} \\ -\delta_{a8} & - & 2\delta_{a12} \\ -\delta_{a13} & - & \delta_{a14} \end{array}\right)$	$\left(\begin{array}{ccc} \delta_{b1} & + & \delta_{b3} \\ +\delta_{b4} & + & \delta_{b5} \\ -2\delta_{b14} & - & \delta_{b15} \end{array}\right)$	
x_5	$\left(\begin{array}{ccc} 2\delta_{a2} & + & \delta_{a4} \\ +\delta_{a6} & + & \delta_{a18} \\ -\delta_{a9} & - & \delta_{a15} \end{array}\right)$	$\left(\begin{array}{ccc} \delta_{b3} & + & 2\delta_{b6} \\ +\delta_{b7} & + & \delta_{b8} \\ -\delta_{b12} & & \end{array}\right)$	
x_8	$\left(\begin{array}{ccc} 2\delta_{a3} & + & \delta_{a5} \\ +\delta_{a6} & + & \delta_{a7} \\ -\delta_{a10} & - & \delta_{a16} \end{array}\right)$	$\left(\begin{array}{ccc} \delta_{b4} & + & \delta_{b5} \\ +\delta_{b7} & + & 2\delta_{b9} \\ +\delta_{b10} & - & \delta_{b13} \end{array}\right)$	
x_9	$\left(\begin{array}{ccc} \delta_{a1} & - & \delta_{a14} \\ -\delta_{a15} & - & \delta_{a16} \\ -\delta_{a17} & & \end{array}\right)$		δ_{c1}
x_{10}	$\left(\begin{array}{ccc} \delta_{a7} & + & \delta_{a18} \\ -\delta_{a11} & - & \delta_{a13} \\ -\delta_{a17} & & \end{array}\right)$	$\left(\begin{array}{ccc} \delta_{b2} & + & \delta_{b8} \\ +\delta_{b10} & + & 2\delta_{b11} \\ -\delta_{b15} & & \end{array}\right)$	δ_{c2}

As in the previous iterations the advanced sign table must be evaluated to determine the true signs. Using the known feasible solution to do this shows variable x_9 must have term 5 to balance. This development requires that constraint 3 be tight ! It also means our previous choice of term 10 over term 5 really did not matter and we were justified in reducing our work load.

Using the equality $x_9 = 2.7976 - x_{10}$ results in the following reformulation of the problem:

Maximize

$$35x_3 + 10x_5 + 15x_8 + 415x_{10} - 570.5952$$

Subject to

(1)

$$\begin{pmatrix} 460.2164x_3 & + & 334.9521x_5 & + & 2610.7601x_8 & + & 8639.575x_{10} \\ +4.02x_3^2 & - & .98x_5^2 & - & 33.2x_8^2 & - & 352.8x_{10}^2 \\ -4.89x_3x_5 & - & 57.32x_3x_8 & + & 25.38x_3x_{10} & - & 10.52x_5x_8 \\ -85.62x_5x_{10} & - & 713.95x_8x_{10} & - & 24224.1896 & & \end{pmatrix} \geq 0$$

(2)

$$\begin{pmatrix} 14230.6 & - & 507.05x_3 & + & 150.34x_5 & + & 2895.2x_8 \\ -2373.3x_{10} & + & 4.02x_3^2 & - & .98x_5^2 & - & 33.2x_8^2 \\ -352.8x_{10}^2 & - & 4.89x_3x_5 & - & 57.32x_3x_8 & + & 25.38x_3x_{10} \\ -10.52x_5x_8 & - & 85.62x_5x_{10} & - & 713.95x_8x_{10} & & \end{pmatrix} \geq 0$$

The last formulation readily allows another solution to be determined. This solution is half of the greatest upper bound on each variable.

$x_1 =$	23.0000	$x_2 =$	23.0000
$x_3 =$	25.0000	$x_4 =$	25.0000
$x_5 =$	7.5000	$x_6 =$	7.5000
$x_7 =$	4.2500	$x_8 =$	4.2500
$x_9 =$	1.3988	$x_{10} =$	1.3988

This solution returns an objective function value of $Z = 1023.6608$, which is 6.4% better than the linear solution. There are good engineering reasons for not preferring this solution to the linear one, but for now it has the advantage of keeping all of the variables in the basis.

Using step I(a), and either known feasible solution to evaluate the remaining constraints, shows that constraint (2) is closer to being tight than (1). It therefore dominates (1) which is now dropped from the problem. The problem is now reduced to the following:

Maximize

$$35x_3 + 10x_5 + 15x_8 + 415x_{10} - 570.5952$$

Subject to

(2)

$$\begin{pmatrix} 14230.6 & - & 507.05x_3 & + & 150.34x_5 & + & 2895.2x_8 \\ -2373.3x_{10} & + & 4.02x_3^2 & - & .98x_5^2 & - & 33.2x_8^2 \\ -352.8x_{10}^2 & - & 4.89x_3x_5 & - & 57.32x_3x_8 & + & 25.38x_3x_{10} \\ -10.52x_5x_8 & - & 85.62x_5x_{10} & - & 713.95x_8x_{10} & & \end{pmatrix} \geq 0$$

Proceed to step IV. The problem can now be formulated as a zero degree of difficulty GP. Rule 4 was used to set up recursive relationships between the variables and the remaining constraint. These relationships were then used to obtain the following progressive solutions:

Iteration	x_3	x_5	x_8	x_{10}
1	25.0000	7.5000	4.2500	1.3988
2	26.6025	8.2488	5.7742	1.4111
3	27.1455	8.9820	7.3551	1.3265
4	27.5263	10.3180	8.5000	1.2272

Because variable x_8 is at its upper limit and, hence, x_7 is at its lower limit, these variables have been determined. The problem is now reformulated:

Maximize

$$35x_3 + 10x_5 + 415x_{10} - 443.0952$$

Subject to

$$\begin{pmatrix} 36441.1 & - & 994.27x_3 & + & 60.92x_5 & - & 8441.875x_{10} \\ +4.02x_3^2 & - & .98x_5^2 & - & 352.8x_{10}^2 & - & 4.89x_3x_5 \\ +25.38x_3x_{10} & - & 85.62x_5x_{10} & & & & \end{pmatrix} \geq 0$$

Continuing with step IV. The next iteration, using rule 4, returns the following solution which determines the x_5 and x_6 variables:

Iteration	x_3	x_5	x_{10}
1	27.2360	15.0000	1.1500

The problem is now:

Maximize

$$35x_3 + 415x_{10} - 293.0952$$

Subject to

(2) $$\begin{pmatrix} 37134.4 & - & 1067.62x_3 & - & 9726.175x_{10} \\ +4.02x_3^2 & - & 352.8x_{10}^2 & + & 25.38x_3x_{10} \end{pmatrix} \geq 0$$

Continuing with step IV and rule 4 returns the following solutions:

Iteration	x_3	x_{10}
1	28.4622	1.0708
2	29.6469	0.9740
3	30.6931	0.8859
4	31.6042	0.8089
5	32.3953	0.7419

6	33.0819	0.6838
7	33.6774	0.6333

Because our goal is to maximize the objective function which contains a positive x_{10} term and because the above GP iterations show this is done by reducing the contribution of this term, it is clear that x_{10} is going to zero. This determines the values of both x_9 and x_{10} to be 2.7976 and 0 respectively.

The reformulated problem is

Maximize

$$35x_3 - 293.0952$$

Subject to

(2) $$37134.4 - 1067.62x_3 + 4.02x_3^2 \geq 0$$

The iterated solution of this problem is given below:

Iteration	x_3	Z^*
1	40.7060	1131.6362
2	41.1603	1147.5382
3	41.1622	1147.5821

The optimum solution to the problem is now determined and presented below:

Cat/93:	$x_1 =$	26.9732	Cat/87:	$x_2 =$	19.0268
Ref/93:	$x_3 =$	41.1622	Ref/87:	$x_4 =$	8.8378
Alk/93:	$x_5 =$	15.0000	Alk/87:	$x_6 =$	0.0000
Nat/93:	$x_7 =$	0.0000	Nat/87:	$x_8 =$	8.5000
But/93:	$x_9 =$	2.7976	But/87:	$x_{10} =$	0.0000

Prior to this application of GP, no method of numerical optimization has been shown to solve this type of blending problem. The optimized objective function value of 1,147.5821 is a 19.3% improvement over the linear method in use. The refinery operator we worked with on this problem believes that he is recovering most of this by adjusting his blending operation on site until all stocks are used in the process. He estimates that his gasoline blending operations are about 18.3% better than the LP design. However, he is interested in knowing if further improvement is possible and how to go about achieving it.

N. Goddard of Citgo Petroleum Corp. has confirmed that, for this particular case, the profit increase would be not less than \$3,836 per day over the actual refinery operation. Implementation of this method on a daily basis, given similar problems, would result in a profit increase of not less than \$1,400,000 per year.

Chapter 6

CONCLUSIONS AND AREAS FOR FURTHER RESEARCH

The goal of this dissertation was to develop a method of simplifying linear blending problems by eliminating constraints prior to processing with the existing simplex algorithm. This objective has been achieved and exceeded.

6.1 Results and Conclusions

The method, on this class of problems, has been shown, to be capable of not just reducing the problems, but solving them. The method does this by eliminating constraints which are not influencing the problem. If it is not desired to pursue the method to solution, it can be stopped after the relevant constraints are selected. The problem is then returned to a conventional LP code. Based on these examples we assert that this can result in a significantly reduced problem which is solved faster than the full sized problem. The results and information gained are identical.

Because the technique can solve these problems it is a feasible alternative to the simplex algorithm for this class of problems. The underlying geometry of Geometric Programming defines the method as an interior point technique.

We have shown that the method can solve a real-world nonlinear chemical blending problem. This is currently beyond the ability of the simplex algorithm. A library search revealed no references to nonlinear codes solving this class of problems. In short, our contribution has three areas of significance. It simplifies chemical blending problems of this class. It also provides an alternative to existing methods for solving this class of Linear Programming problems. Furthermore, it gives forecasters and managers a tool for optimizing nonlinear blending problems.

6.2 Suggestions for Further Research

Several areas for further research spring from this work. A number of them concern providing a complete alternative to the simplex algorithm.

1. Test the technique on other basic types of LP problems. Production process, multiperiod work scheduling, and inventory models are all feasible problem types for this method. Many other types of LP problems (consider the network problem) could avail themselves of the technique as well. Any one of them is suitable as a dissertation topic.

2. The technique demonstrated in this dissertation was able to iterate in on optimal integer solutions in some cases. This ability needs to be tested on pure integer and mixed integer problems. The technique may prove a viable method for solving classes of Integer Programming problems.

3. It should be possible to codify rules by which to formulate a sign table applicable to LP problems exclusively. This would allow nonbinding constraints to be eliminated from the problem before processing with an LP code.

4. Some problems may present multiple combinations of feasible constraint sets which cannot be reduced by the methods we have used. In this case these combinations need to be separately evaluated. This is an excellent application for parallel processing.

5. The same problem described above could be considered an Integer Programming problem. An IP based approach to it would make it a fine topic.

Several other problems occurred to me during this work. They have a broader application than those listed above.

6. Little work has been done in the field of surrogation. A serious effort at compiling and demonstrating what has been done is needed as a starting point for further efforts.

7. Many GP problems are overspecified (more constraint equations than variables). These problems can be reformulated as a DGP subject to rules 2a and 2b. It should be possible to apply LP to these constraints as a means of identifying terms (and hence constraints) worthy of investigation prior to others.

8. Mathematical Statistics could be used on the same problem described above.

9. Considering the advanced sign table as described in the nonlinear gasoline blending problem presented in this dissertation. Some signomial problems may not have any known feasible solutions. These same problems also may have limited useful nonfeasible starting solutions. It should be possible to use Mathematical Statistics on the advanced sign table to provide best estimates of the GP signs within the table.

10. The A-G Mean Inequality was derived on the basis of convexity. Several other inequalities such as Young's, the Cauchy-Schwarz, Hölder's, and Minkowski's, can be derived on the same basis. Any one of these could be used to form a new arena of Mathematical Programming.

REFERENCES CITED

Allen, P.D., 1980, "Unbounded Weights: A New Approach to the Solution of Nonlinear Optimization Problems with Applications to Mineral Economics." Ph.D. dissertation, Colorado School of Mines, Golden, Colorado.

Amundson, N.R., 1966, *Mathematical Methods in Chemical Engineering*. Englewood Cliffs, N.J.: Prentice-Hall.

Aronson, J.E., Morton, T.E., and Thompson, G.L., 1985, "A Forward Simplex Method for Staircase Linear Programs." *Management Science*. v 31, n 6.

Avriel, M., and Williams, A.C., 1970, "Complementary Geometric Programming." *SIAM Journal of Applied Mathematics*. v 19, n 1.

Avriel, M., Rijckaert, M.J., and Welde, D.J., 1973, *Optimization and Design*. Englewood Cliffs, N.J.: Prentice-Hall.

Bailey, S.S., 1989, "An Algorithm for the Solution of a Class of Economic Models for the Design of Prestressed Concrete Bridges." Ph.D. dissertation, Colorado School of Mines, Golden, Colorado.

Baird, C.T. IV., 1989, *Guide to Petroleum Product Blending*. Austin, TX.: HPI Consultants, Inc.

Baker, D.W., 1980, "Application of the Geometric Inequality to the Solution of Systems of Nonlinear Equations." Ph.D. dissertation, Colorado School of Mines, Golden, Colorado.

Baker, T.E., and Lason, L.S., 1985, "Successive Linear Programming at Exxon." *Management Science*. v 31, n 3.

Beightler, C.S., and Phillips, D.T., 1976, *Applied Geometric Programming*. New York: Wiley.

Beightler, C.S., Phillips, D.T., and Wilde, D.J., 1979, *Foundations of Optimization* 2nd ed. Englewood Cliffs, N.J.: Prentice-Hall.

Bland, R.G., Goldfarb, D., and Todd, M.J., 1981, "The Ellipsoid Method: A Survey." *Operations Research*. v 29, n 6.

Bodington, C.E., 1979, "Nonlinear Programs for Product Blending." Joint National TIMS/ORSA Meeting, April/May. New Orleans, LA.

Bodington, C.E., and Baker, T.E., 1990, "A History of Mathematical Programming in the Petroleum Industry." *Interfaces*. v 20, n 4.

Buchanan, J.E., and others, 1990, "A Multi - Refinery, Multi - Period Modeling System for the Turkish Petroleum Refining Industry." *Interfaces* v 20, n 4.

Denn, M.M., 1986, *Process Modeling*. White Plains, N.Y.: Longman.

Duffin, R.J., Peterson, E.L., and Zener, C., 1967, *Geometric Programming: Theory and Application*. New York: Wiley.

Dyer, M.E., 1984, "Linear Time Algorithms for Two- and Three-Variable Linear Programs." *SIAM Journal of Computing*. v 13, n 1.

Ecker, J.G., 1980, "Geometric Programming: Methods, Computations and Applications." *SIAM Review*. v 22, n 3.

Ecker, J.G., and Kupferschmid, M., 1988, *Introduction to Operations Research*. New York: Wiley.

Fletcher, R., 1987, *Practical Methods of Optimization* 2nd ed. Chichester, England: Wiley.

Foster, T.R.V., 1978, *The Aircraft Owner's Handbook*. New York: Van Nostrand Reinhold.

Foulds, L.R., 1984, *Combinatorial Optimization for Undergraduates*. New York: Springer-Verlag.

Gary, J.H., and Handwerk, G.E., 1984, *Petroleum Refining Technology and Economics* 2nd ed.. New York: Dekker.

Gass, S.I., 1970, *An Illustrated Guide to Linear Programming*. New York: Dover.

Gass, S.I., 1985, *Linear Programming* 5th ed.. New York: McGraw-Hill.

Gill, P.E., Murray, W., and Wright, M.H., 1981, *Practical Optimization*. London: Academic Press.

Gordon, R.C. III., 1981, "Optimization of Phosphate Ore Blending: A Linear Programming Approach." M.S. thesis, Colorado School of Mines, Golden, Colorado.

Grange, F.E. II., 1977, "The Solution and Analysis of the Geometric Mean Portfolio Problem with Methods of Geometric Programming." Ph.D. dissertation, Colorado School of Mines, Golden, Colorado.

Grange, J.K., 1977, "Design of Distillate Desulfurization Reactors with Geometric Programming." M.S. thesis, Colorado School of Mines, Golden, Colorado.

Greening, D.R., 1982, "A Proof of Convergence for a Condensation Approach to the Solution of Systems of Nonlinear Equations." M.S. thesis, Colorado School of Mines, Golden, Colorado.

Han, S., and Lou, G., 1988, "A Parallel Algorithm for a Class of Convex Programs." *SIAM Journal of Control and Optimization*. v 26, n 2.

Ho, J.K., and Loute, E., 1983, "Computational Experience with Advanced Implementation of Decomposition Algorithms for Linear Programming." *Mathematical Programming*. v 27, n 3.

Jefferson, T.R., and Scott, C.H., 1985, "Quadratic Geometric Programming with Application to Machining Economics." *Mathematical Programming*. v 31, n 2.

Kirk, J.B., 1988, "A Geometric Programming Based Algorithm for Preprocessing Nonlinear Signomial Optimization Programs." Ph.D. dissertation, Colorado School of Mines, Golden, Colorado.

Klingman, D., and others, 1987, "An Optimization Based Integrated Short-Term Refined Petroleum Product Planning System." *Management Science*. v 33, n 7.

Klingman, D., Mote, J., and Phillips, N.V., 1988, "A Logistics Planning System at W.R. Grace." *Operations Research*. v 36, n 6.

Koo, D., 1977, *Elements of Optimization*. Heidelberg: Springer - Verlag.

Leffler, W.L., 1985, *Petroleum Refining for the Non-Technical Person*. Tulsa, OK.: Penn Well.

Lesso, W.G., 1982, "Operations Research." In *McGraw-Hill Encyclopedia of Science and Technology* 5th ed..

Mangasarian, O.L., and Meyer, R.R., 1979, "Nonlinear Perturbation of Linear Programs." *SIAM Journal of Control and Optimization*. v 17, n 6.

Maurer, R., 1990, "Class Notes: Mathematics Course 582: Networks and Decision Theory." Colorado School of Mines, Golden, Colorado.

Nazareth, J.L., 1987, *Computer Solution of Linear Programs*. New York: Oxford.

Oatney, C.K., 1987, "Solution to a Class of Economic Reliability Problems with Geometric Programming." M.S. thesis, Colorado School of Mines, Golden, Colorado.

Packel, E.W., 1988, "Do Linear Problems Have Linear Optimal Algorithms?" *SIAM Review*. v 30, n 3.

Peressini, A.L., Sullivan, F.E., and Uhl, J.J. Jr., 1988, *The Mathematics of Nonlinear Programming*. New York, N.Y.: Springer - Verlag.

Peterson, E.L., 1972, "Mathematical Foundations of Geometric Programming." "Appendix to Geometric Programming and Some of its Extensions;" see Avriel, M., Rijckaert, M.J., and Wilde, D.J., 1972, *Optimization and Design*. Englewood Cliffs, N.J.: Prentice-Hall.

Peterson, E.L., 1976, Geometric Programming. *SIAM Review*. v 18, n 1.

Prince, L., and others, 1983, "Gasoline Blending at Texaco Using Nonlinear Programming." Joint National TIMS/ORSA Meeting, April. Chicago, IL.

Ratliff, R.M., 1986, "A Generalized Condensation Algorithm for the Solution of Unconstrained Balanced Posynomial Problems." M.S. thesis, Colorado School of Mines, Golden, Colorado.

Rijckaert, M.J., and Walraven, E.J.C., 1985, "Geometric Programming: Estimation of Lagrange Multipliers." *Operations Research*. v 33, n 1.

Scales, L.E., 1985, *Introduction to Non-Linear Optimization*. New York: Springer - Verlag.

Shamir, R., 1987, "The Efficiency of the Simplex Method: A Survey." *Management Science*. v 33, n 3.

Shogan, A.W., 1988, *Management Science*. Englewood Cliffs, N.J.: Prentice-Hall.

Symonds, G.H., 1955, *Linear Programming: The Solution of Refinery Problems*. New York: Esso Standard Oil Co.

Taylor, D.L., 1980, "Some Aspects of Optimal Design and Operation of Mineral Floation Circuits." Ph.D. dissertation, Colorado School of Mines, Golden, Colorado.

Thome, J.J. Sr., 1988, "An Algorithm for Solving a Class of Nonlinear, Unconstrained, Signomial Economic Models Using the Greening Technique." Ph.D. dissertation, Colorado School of Mines, Golden, Colorado.

Treece, P.T., 1978, "The Application of Pre-Processing Techniques to the Solution of Reliability Models in Linear Programming." M.S. thesis, Colorado School of Mines, Golden, Colorado.

Van Drew, S.L., 1985, "A Linear Programming - Based Coal Preparation and Blending Technique." M.S. thesis, Colorado School of Mines, Golden, Colorado.

Wall, T.W., 1984, "A Numerical Algorithm for the Solution of Chemical Equilibrium Problems." M.S. thesis, Colorado School of Mines, Golden, Colorado.

Wessels, G.J., 1988, "Class Notes: Mineral Economics Course 555: Linear Programming." Colorado School of Mines, Golden, Colorado.

Wessels, G.J., 1989, "A Geometric Programming Algorithm for Solving a Class of Nonlinear, Signomial Optimization Problems." Ph.D. dissertation, Colorado School of Mines, Golden, Colorado.

Wilkinson, J.G., 1987, "An Algorithm for Solving a Class of Multiple Goal Reliability Problems." M.S. thesis, Colorado School of Mines, Golden, Colorado.

Wilkinson, J.G., 1989, "Optimization of a Class of Nonlinear Reliability Problems Using a Structure Analysis Algorithm." Ph.D. dissertation Colorado School of Mines, Golden, Colorado.

Williams, H.P., 1990, *Model Building in Mathematical Programming* 3rd ed.. Chicheste, England: Wiley.

Winston, W.L., 1987, *Operations Research: Applications and Algorithms*. Boston: Duxbury.

Woolsey, R.E.D., and Swanson, H.S., 1969, *Operations Research for Immediate Application: A Quick and Dirty Manual*. New York: Harper & Row.

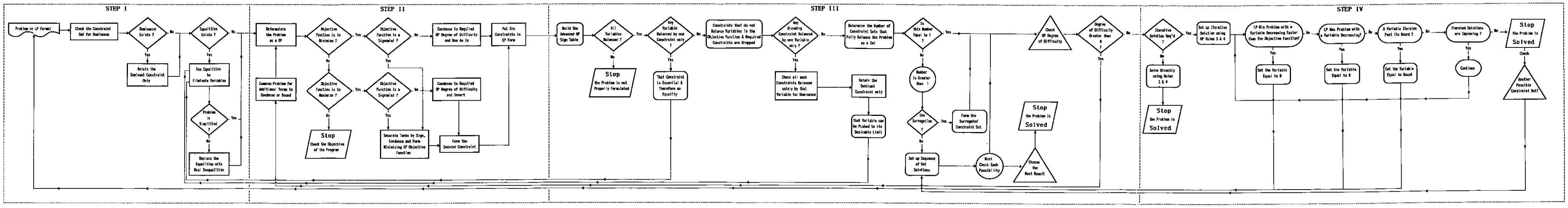
Woolsey, R.E.D., Kochenburger, G.A., and Linck, K.R., 1971, "Can Engineers Find Happiness Without Calculus?" *Hydrocarbon Processing*. Aug.

Woolsey, R.E.D., 1987, "Class Notes: Mineral Economics Course 525: Operations Research." Colorado School of Mines.

Woolsey, R.E.D., 1988, "Class Notes: Mineral Economics Course 558: Geometric Programming." Colorado School of Mines.

Woolsey, R.E.D., 1989, "Class Notes: Mineral Economics Course 554: Integer Programming." Colorado School of Mines.

Wu, N., and Coppins, R., 1981, *Linear Programming and Extensions*. New York: McGraw-Hill.



STEP I
Problem in LP Format
Check the Constraint Set for Dominance
Dominance Exists ?
Retain the Dominant Constraint Only
Equalities Exists ?
Use Equalities to Eliminate Variables
Problem is Simplified ?
Replace the Equalities with Dual Inequalities
STEP II
Reformulate the Problem as a GP
Examine Problem For Additional Terms to Condense or Bound
Objective Function is to Minimize ?
Objective Function is to Maximize ?
Stop
Check the Objective of the Program
Objective Function is a Signomial ?
Separate Terms by Sign, Condense and Form Minimizing GP Objective Function
Condense to Required GP Degree of Difficulty and Use As Is
Condense to Required GP Degree of Difficulty and Invert
Form the Induced Constraint
Put the Constraints in GP Form
STEP III
Build the Advanced GP Sign Table
All Variables Balanced ?
Stop
the Problem is not Properly Formulated
Any Variable Balanced by one Constraint only ?
That Constraint is Essential & Therefore an Equality
Constraints that do not Balance Variables in the Objective Function & Required Constraints are Dropped
Any Blending Constraint Balanced by one Variable only ?
Check all such Constraints Balanced solely by that Variable for Dominance
Retain the Dominant Constraint only
That Variable can be Pushed to its Desirable Limit
Determine the Number of Constraint Sets that Fully Balance the Problem as a Set
Is this Number Equal to 1 ?
Number is Greater than 1
Use Surrogation ?
Form the Surrogated Constraint Set
Set up Sequence of Set Solutions
Must Check Each Possibility
Choose the Best Result
Stop
the Problem is
Solved
Check GP Degree of Difficulty
Degree of Difficulty Greater than 0 ?
STEP IV
Iterative Solution Req'd ?
Solve Directly using Rules 3 & 4
Stop
the Problem is
Solved
Set up Iterative Solution using GP Rules 3 & 4
LP Min Problem with a Variable Decreasing Faster than the Objective Function?
Set the Variable Equal to 0
LP Max Problem with a Variable Decreasing?
Set the Variable Equal to 0
A Variable Iterates Past Its Bound ?
Set the Variable Equal to Bound
Iterated Solutions are Improving ?
Continue
Stop
the Problem is
Solved
Check
Another Possible Constraint Set?
Yes
No