QUADRATIC MINIMIZATION
AND LEAST DISTANCE PROGRAMMING

by

Syed Z. Shariq

Dissertation submitted to the Graduate Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY
in
Industrial Engineering and Operations Research

APPROVED:

Dr. V. Chachra, Chairman

Dr. M. H. Agee

Dr. I. M. Besieris

Dr. P. M. Ghare

Dr. W. C. Turner

Blacksburg, Virginia

September 1974

ACKNOWLEDGMENTS

The author is grateful for this opportunity to express his appreciation to the following individuals for their aid and encouragement in the completion of this study.

Dr. V. Chachra, the author's major advisor, for his encouragement, understanding, help and guidance both within and outside the realm of this research.

Dr. P. M. Ghare, member of the graduate committee, for his valuable suggestions concerning the mathematical developments of this study.

Dr. M. H. Agee, member of the graduate committee, for his valuable guidance and support during this study.

Dr. W. C. Turner and Dr. I. M. Besieris, members of the graduate committee, for their advice, encouragement and active participation in this research.

The wife of the author, Doris, for her patient effort in reading and editing this manuscript.

Mr. G. Dayal, Mr. V. Garg and Mrs. D. Barnes, friends of the author, for their faith, optimism and encouragement.

Mrs. Margie Strickler, for her excellent typing--a job well done despite many obstacles.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

NOTATIONS

$\rightarrow$	implication, the statement on left implies the statement on the right.
$\leftrightarrow$	equivalence, the statement on left is equivalent to the statement on the right.
$\{x\}$	the set consisting of x.
$\{x\|u(x)\}$	the set of elements x for which u(x) is true.
$\emptyset$	the empty set.
ε, C	respectively, element and set containment.
$\cup, \cap, -$	respectively, set union, intersection and difference.
E^n	the n-dimensional Euclidean vector space.
E_+^n	the non-negative orthant in E^n.
$\oplus$	the direct sum of relative subspaces U and V, that is, if $E^n = U + V$, then any vector $x \varepsilon E^n$ may be written in the form $x = x_1 + x_2$, with $x_1 \varepsilon U$, $x_2 \varepsilon V$ in one and only one way.

For $x, y \varepsilon E^n$, let:

$x \geq y$ denote $x - y \varepsilon E_+^n$.

$(x,y) = \sum_{i=1}^{n} x_i y_i$ denote the inner product of x, y.

$|x|_2 = \sqrt{(x,x)}$, the Euclidean norm of x.

$x \perp y$ = denote $(x,y) = 0$

$|x|$ denote $(|x|)_i = x_i$, $i = 1, \ldots, n$.

For a subspace L in E^n let:

$x \perp L$ denote $x \perp y$ for all $y \varepsilon L$.

$L^{\perp} = \{x | x \varepsilon E_n, x \perp L\}$, the orthogonal complement of L.

P_L = the perpendicular projection on L, i.e., $P_L = P_L^2 = P_L'$, $L = \{x | x \varepsilon E^n, P_L x = x\}$

For a set $Q \subset E^n$, let:

int(Q) denote the interior of Q.

bdry(Q) denote the boundary of Q.

#(Q) denote the number of elements in Q.

I the identify matrix.

For an mxn matrix A let:

A' denote the transpose of A.

$N(A) = \{x | x \varepsilon E_n, Ax = 0\}$, the null space of A.

$R(A') = \{y | y \varepsilon E_n, y = A'u \text{ for some } u \varepsilon E^m\}$, the range space of A'.

rank(A) denote rank of A.

$A^\dagger$ denote the generalized inverse of A.

A^D denote the Darzin psuedoinverse of A.

For a square matrix B, let:

det(B) denote the determinant of B.

B^{-1} denote the inverse of B.

For $x \varepsilon E^n$, let:

$$x_+ \text{ denote } \begin{cases} (x_+)_i = x_i & \text{if } x_i > 0. \\ (x_+)_i = 0 & \text{if } x_i \leq 0. \end{cases}$$

Chapter I

INTRODUCTION

1.1 Overview Background

The scientific approach to the decision making process of complex systems generally consists of two steps. The first step consists of creating a model for the problem. If the model is in mathematical terms, then the problem is said to have a mathematical model associated with it. The second step consists of obtaining an optimal solution to the model. The term optimization is used for those procedures and techniques which enable us to arrive at the optimal solution of these models. Many of the mathematical models in real life are generally nonlinear in nature. The area of study that deals with these nonlinear models is called nonlinear programming.

The study of optimization procedures can be traced as far back as Cauchy and Gauss. An excellent account of fundamental concepts associated with optimization is presented by Hancock [59]. Since the end of the Second World War, there has been a rapidly expanding interest throughout the business and scientific world in the ideas and methods of mathematical programming [e.g., 4,10,31,35,125]. The development of the powerful simplex method of linear programming and the advent of high speed computers gave a large impetus to interest in this area [30,43,61,62,71,122]. This great interest in linear programming was followed by an even greater interest in the area of nonlinear programming.

A general nonlinear programming problem involves the maximization or minimization of a real valued nonlinear objective function, subject to nonlinear constraints. Unfortunately, this general problem is extremely difficult to solve. The relatively few solution procedures to the general nonlinear programming problem are very complex [e.g., 55,102]. If in the general nonlinear programming problem, the constraints are linear, then the problem becomes easier to solve. Compared to the general problem, there are relatively efficient methods to solve the nonlinear programming problem with linear constraints [e.g., 51,101].

There are two commonly used approaches to the solution of nonlinear programming problems. These are:

(1) The use of a transformation technique to convert a constrained nonlinear programming problem to an unconstrained nonlinear programming problem. Several transformations may be used to accomplish this. The use of penalty functions [40] or the Lagrange multipliers [67, p. 30] are good examples of the transformation technique.

(2) The nonlinear programming problem may be approximated by a sequence of linear or quadratic programming problems [67, p. 86]. Successive solutions to the linear or quadratic programming problems would lead to an approximate solution of the general nonlinear programming problem.

The quadratic programming problem plays an important role in both the above approaches. Either the quadratic programming problem is used to successively approximate the general nonlinear programming problem or the quadratic programming problem is used to test and establish the effectiveness of the algorithm devised for the general nonlinear programming problem. The quadratic programming problem is also important in its own right as it is inherent in many of the economic models. A few examples of the numerous applications of quadratic programming are: problems of profit maximization when demand is a linear function of price [32,33]; portfolio selection for a stock market requiring a certain profit while maintaining a minimum variance [82]; finding the best least squares fit to a given data where certain parameters must satisfy linear constraints [53,108]. In view of the importance of the problem, the central theme of this dissertation will be directed towards the solution of the quadratic programming problem. The quadratic programming problem will be solved through the use of the least distance programming problem. This is done because least distance programming problems are easier to visualize and understand. In solving the least distance programming problem, importance is given to the property of numerical stability.

The term "numerical instability" refers to a high tendency to cumulate and propogate computational errors during a computational procedure. Many of the existing approaches to the solution of the nonlinear programming problem suffer from numerical instability. Attempts to invert a nearly singular matrix results in numerical instability. Another example of numerical instability arises in the use of the steepest descent method for minimizing an unconstrained quadratic function with the Hessian matrix having a high ratio of the largest eigenvalue to the smallest eigenvalue. An investigation of numerical instability in mathematical programming algorithms was started by Bartles, et al. [5,6,7,8] and Müller-Merbach [87]. Recently, it has been found that the Cholesky factorization when used in conjunction with Householder's transformation provides a numerically stable procedure for solving the linear least squares problem [53]. This property of Cholesky factorization has motivated its use in nonlinear programming techniques. For example, Saunders [103] has proposed a stable version of the Simplex method for linear programming using Cholesky factorization [17,42,46,50].

The research of this dissertation extends the use of Cholesky factorization in determining a unique transformation for solving quadratic minimization problems. Since the procedure presented here is based on Cholesky factorization, it provides a numerically stable method. The procedure is fully described in Chapter 2. Briefly, the procedure consists of transforming the quadratic programming problem to a least distance programming problem through the use of linear least distance transformation.

Since the linear least distance transformation reduces the quadratic programming problem to a least distance programming problem, we need an effective method for solving the latter. Chapter 3 considers some special cases in the solution of the least distance programming problem. Primarily, these special cases enhance the understanding of the geometry of the least distance programming problem.

Also, the special cases show simple solution procedures for problems meeting certain assumptions about the orthogonality of the constraint matrix.

An alternate approach to the solution of the least distance programming problem would be through its dual. Chapter 4 defines a "generalized dual" of the least distance programming problem. This dual problem is shown to be convex and uniquely solvable by the Newton procedure. The use of partially isometric matrices is also considered in the solution of the generalized dual problems.

In a sense, Chapters 3 and 4 present solutions to special cases of the least distance programming problem. Chapter 5 defines a computational procedure for solving the general least distance programming problem. The algorithm presented in Chapter 5 uses a cutting plane strategy for successively diminishing the feasible region. The proposed procedure is characterized by the few computations it requires compared to other existing algorithms. A few alternative approaches to certain computations in the algorithm are also presented.

1.2 Definitions

Before stating the objectives of this research in Section 1.3, it will be of help to have the following definitions.

Definition 1.1:

A quadratic programming problem (QP) is defined as:

$$q^* = q^*(x^*) = \min_{x \varepsilon Q} [q(x)] = \min_{x \varepsilon Q} [x'Ax+2b'x] \tag{1.1}$$

where

$Q = \{x | Gx \leq h\} \subset E^n$

$A = [n,n]$ positive definite matrix

$G = [m,n]$ matrix

and if $Q = E^n$, then the unconstrained minimum is given by

$$x^*_u \equiv x^* \quad \text{and} \quad q^*_u \equiv q(x^*_u) .$$

Definition 1.2:

A least distance programming problem (LDP) is defined as:

$$\ell^* = \ell(y^*) = \min_{y \varepsilon L} [\ell(y)] = \min_{y \varepsilon L} [|y|_2{}^2] = \min_{y \varepsilon L} [y'y] \tag{1.2}$$

where

$L = \{y | By \leq c\} \subset E^n$

$B = [m,n]$ matrix

and if $L = E^n$, then the unconstrained minimum is given by

$$y^*_u = y^* = 0 \quad \text{and} \quad \ell^*_u = \ell(y^*_u) = 0 .$$

Furthermore, if in a quadratic programming problem the linear and interaction terms are absent from the objective function then it reduces to a least distance programming problem.

1.3 Research Objectives

This research is primarily directed to the solution of quadratic programming problems. It is felt that the least distance programming problem is simpler to visualize and solve than the quadratic programming problem; therefore, the research is directed toward solving the quadratic programming problem through the use of the least distance programming problem. Specifically, the objectives of the research are:

(1) To identify a transformation that reduces the quadratic programming problem to the least distance programming problem.

(2) To determine a numerically stable algorithm to accomplish the transformation obtained in (1).

(3) To study the various approaches to the solution of the least distance programming problem, the primary objective being to use the special properties of least distance programs to obtain easy solutions.

(4) To develop a computational algorithm to solve the least distance programming problem. In view of (1) and (2), this would yield a solution to the quadratic programming problem.

1.4 Literature Review

During the last two decades, several techniques for solving convex quadratic programs have been developed [18,75], the majority of them based on Kuhn-Tucker optimality conditions [74]. Some of the better known quadratic programming algorithms relevant to this category are those of Beale [9], Frank-Wolf [15, p. 147] and Wolf [119]. Recent research has been concerned with the development of non-convex quadratic programming algorithms [e.g., 26,48,70,84,89,98], to which pivotal algebra has also made successful contributions [92,111].

There has also been considerable interest in unconstrained quadratic minimization methods, due primarily to their wide applicability in solving general nonlinear programming problems [40,63] and linear least squares problems [53,72,107]. A solution to the unconstrained quadratic minimization problem is usually obtained by solving a system of linear equations. However, if this system of linear equations is ill-conditioned the well-known solution techniques [e.g., 46,54,105,110, 118] are found to perform poorly [116,117].

Householder's transformation [50,53,66,107] and elimination method [24] are examples of new techniques developed to solve ill-conditioned quadratic minimization problems. Despite the ill-conditioning, these techniques have been found reasonably stable and accurate [91,115]. Furthermore, their superior computational properties have made them useful in quadratic programming [53,89]. A recent survey of these and other related developments in quadratic minimization techniques may be found in Leotsma [79] and Murray [90].

The least distance programming problem is a special case of the quadratic programming problem, originally introduced by Tucker [112,113], who also presented a combinatorial solution to the problem. In the approximation theory the least distance programming problem occurs as a special case of the general minimum norm problem. The general minimum norm problem (i.e., p-norm, $p > 0$) reduces to a least distance programming problem if the norm is taken to be Euclidean (i.e., $p = 2$).

The general minimum norm problem has been used as a subproblem to obtain solutions of linear and nonlinear inequalities [28,29,99]. Since minimum norm problems with $p = 2$ (equivalent to the least distance programming problem) are rather difficult to solve, common practice is to solve for $p = 1$ and $p = \infty$ norms. Problems with $p = 1$ and $p = \infty$ norms can be solved by the Simplex method of linear programming [97]. A theoretical convergent algorithm for a special class of the general minimum norm problem is proposed by Karlovitz [69]. In addition, the exchange algorithm of convex analysis [77] can be modified to solve minimum norm problems.

The least distance programming problem can be solved by any of the known convex quadratic programming techniques [18,75]. Mitchell, et al. [86] have devised an algorithm for this purpose. This is perhaps the only algorithm designed specifically to solve least distance programming problems. However, since all the above procedures including that by Mitchell et al. are complex, it is felt that a simpler solution procedure should be made available. This research is therefore undertaken to study some new approaches in the solution of least distance programming problems.

Chapter II

THE LINEAR LEAST DISTANCE TRANSFORMATION

2.1 Introduction

The use of transformations to reduce the complexity of mathematical programming problems is by no means a new idea [63, p. 7]. The use of a transformation is justified only if it is simple in nature (e.g., linear) and if it can be accurately computed with a minimum amount of computational effort. The transformation of a quadratic function

$$f(x) = (Ax,x) \ , \tag{2.1}$$

to a sum of squares

$$g(y) = (y,y) \ , \tag{2.2}$$

where A is a positive definite matrix is well known in matrix theory [85, p. 398]. A computational procedure to determine such a transformation is presented by Schwarz, et al. [105, p. 20]. This procedure uses the Cholesky factorization of matrix A. The Cholesky factorization is numerically stable and has been recently used by several investigators [7,49,50,53,89,103] in mathematical programming techniques.

It will be shown that the general quadratic function

$$q(x) = (Ax,x) + 2(x,b) \tag{2.3}$$

can be reduced to a sum of squares

$$\ell(y) = (y,y) + t \tag{2.4}$$

by means of the Cholesky factorization, where t is a constant.

Since the Cholesky factorization is uniquely defined for positive definite matrices, the transformation from equation (2.1) to equation (2.2) is already available. Also, equation (2.2) and equation (2.4) are similar. The transformation from equation (2.3) to equation (2.4), therefore, would be achieved if it were possible to represent equation (2.3) in the form of equation (2.1). The transformation that is proposed here provides a method of going from equation (2.3) to a form similar to equation (2.1) while maintaining the positive definite requirements. This used in conjunction with the Cholesky factorization will define a transformation from equation (2.3) to equation (2.4). This transformation from equation (2.3) to equation (2.4) will be called the linear least distance transformation (LLDT).

The following Lemmas are used in proving the main theorems in the derivation of the linear least distance transformation.

Lemma 2.1

If matrix A is positive definite, then the matrix A^{-1} is also positive definite.

Proof:

The proof may be found in Fadeev and Fadeeva [38, Theorem 11.10, p. 91].

Lemma 2.2

Let the square matrix S be partitioned

$$S = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{2.5}$$

and suppose A_{11} is nonsingular. Then

$$\det(S) = \det(A_{11})\cdot\det(A_{22}-A_{21}A_{11}{}^{-1}A_{12}) \ . \tag{2.6}$$

Proof:

The proof may be found in Householder [66, p. 17].

Lemma 2.3 (Cholesky Factorization)

If matrix S is positive definite, then there is a unique upper triangular matrix T with positive diagonal elements such that

$$S = T'T \ . \tag{2.7}$$

Proof:

The proof may be found in Stewart [107, Theorem 3.8, p. 140].

Subsequently, Section 2.2 gives the actual derivation of the LLDT. The computational algorithm to obtain the transformation is presented in Section 2.3. Possible applications are outlined in Section 2.4. Section 2.5 shows a numerical example of the technique.

2.2 Linear Least Distance Transformation (LLDT)

The quadratic function

$$q(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}x_i x_j + 2\sum_{j=1}^{n} b_j x_j$$

$$= (Ax,x) + 2(x,b) \tag{2.7a}$$

consists of a real symmetric [n,n] positive definite matrix A and the vector $b \neq 0$. If $b = 0$ then the function q(x) reduces to the function in equation (2.1). The transformation procedure for such a function already exists [105, p. 20]. Thus, it will be assumed that $b \neq 0$. If the feasible region Q of the quadratic programming problem defined in equation (1.1) is the euclidean space E^n, then we have an unconstrained quadratic programming problem. The optimal solution of an unconstrained quadratic programming problem x^*_u is defined by,

$$\min_{x \in E^n} [q(x)] = q(x^*_u) = q^*_u \ . \tag{2.7b}$$

The derivation of LLDT follows directly from Theorems 2.1, 2.2 and 2.3.

Theorem 2.1

If the quadratic programming problem is defined by equation (2.7b) with the matrix A as a real, symmetric and positive definite matrix and also the vector $b \neq 0$, then,

$$q^*_u = -(A^{-1}b,\ b) < 0 \ . \tag{2.8}$$

Proof:

The positive definiteness of matrix A implies strict-convexity of the function $q(x)$. Thus the necessary conditions defining the minimum of the function $q(x)$ are also sufficient conditions. Hence, setting the gradient equal to zero gives

$$x^*u = -A^{-1}b \ . \tag{2.9}$$

Substituting from equation (2.9) in equation (2.7a) results in

$$\begin{aligned} q^*_u = q(x^*_u) &= (Ax^*_u, x^*_u) + 2(x^*_u, b) \\ &= (-AA^{-1}b, A^{-1}b) + 2(-A^{-1}b, b) \\ &= (b, A^{-1}b) - 2(A^{-1}b, b) \ . \end{aligned}$$

Therefore,

$$q^*_u = -(A^{-1}b,b) \ . \tag{2.10}$$

From Lemma 2.1 matrix A^{-1} is positive definite. Therefore,

$$(A^{-1}b,b) > 0 \ , \text{ for all } b \neq 0 \ . \tag{2.11}$$

Using equation (2.10) and equation (2.11) gives

$$q^*_u = -(A^{-1}b,b) < 0 \ . \tag{2.12}$$

This concludes the proof of Theorem 2.1.

Theorem 2.1 establishes that the optimal solution of the quadratic program defined by equation (2.7b) is always negative. Now, for some positive scalar $\delta > 0$, the quadratic function $q(x)$ can be rewritten as,

$$q(x) = q(x) + \delta - \delta \tag{2.13}$$

$$= (Ax,x) + 2(x,b) + \delta - \delta$$

$$= [x,1]\begin{bmatrix} A & b \\ b' & \delta \end{bmatrix}\begin{bmatrix} x \\ 1 \end{bmatrix} - \delta \tag{2.14}$$

Or, that

$$q(x) = [x,1] \; S \; [{}^{x}_{1}] - \delta \ , \tag{2.15}$$

where

$$S = \begin{bmatrix} A & b \\ b' & \delta \end{bmatrix} \tag{2.16}$$

is a [n+1,n+1] matrix.

Examination of equation (2.15) shows that the negatative δ plays no part in the minimization of the function $q(x)$, as δ is a scalar constant. The rest of equation (2.15) is very similar to equation (2.1.). Hence, if it can be established that S is a positive definite matrix then the Cholesky factorization can be directly used for equation (2.15). Equation (2.16) defines the matrix S. Since A is positive definite matrix, it is clear that for certain values of δ the matrix S will be positive definite. Theorem 2.2 establishes a criterion for selecting an appropriate value of δ which makes the matrix S a positive definite matrix.

Theorem 2.2

Given a quadratic function q(x),

$$q(x) = (Ax,x) + 2(x,b) ,$$

where A is a positive definite matrix and q^*_u is such that $q^*_u = \min_{x\in E^n}[q(x)]$, then the matrix S defined by (δ is a scalar)

$$S = \begin{bmatrix} A & b \\ b' & \delta \end{bmatrix}$$

is a positive definite matrix if

$$\delta > -q^*_u > 0 . \tag{2.17}$$

In particular, S is positive definite if

$$\delta = -q^*_u + 1 . \tag{2.18}$$

Proof:

From Lemma 2.2 and equation (2.16) we have

$$\det(S) = \det(\delta - b'A^{-1}b)\cdot\det(A) . \tag{2.19}$$

Since the matrix A is positive definite the determinant of A is greater than zero, i.e.,

$$\det(A) > 0 \ , \tag{2.20}$$

as are the determinants of all the principal minors of A. Therefore the matrix S will be positive definite if

$$\det(S) > 0 \ . \tag{2.21}$$

Using equation (2.19), det(S) may be rewritten as

$$\det(S) = \det(\delta - b'A^{-1}b)\cdot\det(A) \ .$$

Since $\det(A) > 0$, we only need to establish that

$$\det(\delta - b'A^{-1}b) > 0 \ .$$

But, $(\delta - b'A^{-1}b)$ is a scalar, therefore we need only show that

$$(\delta - b'A^{-1}b) > 0 \ .$$

Or that

$$\delta > b'A^{-1}b = (A^{-1}b,b) \ .$$

Using equation (2.8) gives,

$$\delta > -q^*_u > 0 \ . \tag{2.22}$$

Equation (2.22) shows that for any $\delta > -q^*_u$ the matrix S is positive definite. In particular, the value of $\delta = (-q^*+1) > 0$ ensures the positive definiteness of S. This concludes the proof of Theorem 2.2.

Theorem 2.2 establishes that the matrix S can be made to be a positive definite matrix. In order to perform the linear least distance transformation it is necessary to represent the matrix S in a unique form. Theorem 2.3 shows the representation and establishes certain important properties of this representation. The results of Theorem 2.3 are then used to identify the linear least distance transformation.

Theorem 2.3

Any [n+1,n+1] positive definite matrix S has a unique representation

$$S = T'T \tag{2.23}$$

where

$$T = \begin{bmatrix} \tilde{T} & \Pi \\ 0' & \alpha \end{bmatrix} \tag{2.24}$$

with

$$\tilde{T} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ & t_{22} & \cdots & t_{2n} \\ & & \ddots & \\ 0 & & & t_{nn} \end{bmatrix}, \quad \Pi = \begin{bmatrix} t_{1,n+1} \\ t_{2,n+1} \\ \vdots \\ t_{n,n+1} \end{bmatrix}, \tag{2.25}$$

and

$$\alpha = t_{n+1,n+1} > 0 . \tag{2.26}$$

For the above representation, the following holds:

(a) $\alpha = (\delta - |\Pi|_2^{\,2})^{1/2}$ (2.27)

(b) $x^*_u = -\tilde{T}^{-1}(\Pi)$ (2.28)

and $q^*_u = -|\Pi|_2^{\,2}$ (2.29)

Furthermore, if in particular $\delta = (-q^*_u+1) > 0$ (as suggested in Theorem 2.2), then

(c) $\alpha = 1$. (2.30)

Proof:

(a) $S = T'T$

From equations (2.16) and (2.24) we get

$$\begin{bmatrix} A & b \\ b' & \delta \end{bmatrix} = \begin{bmatrix} \tilde{T} & \Pi \\ 0' & \alpha \end{bmatrix}' \begin{bmatrix} \tilde{T} & \Pi \\ 0' & \alpha \end{bmatrix} . \tag{2.31}$$

Equating the (n+1,n+1) element from both sides of equation (2.31) results in

$$\delta = \Pi'\Pi + \alpha^2 ,$$

or

$$\alpha = (\delta - |\Pi|_2{}^2)^{1/2} . \tag{2.32}$$

(b) Using equations (2.14) and (2.31) the function q(x) can be written as

$$q(x) = [x,1]\begin{bmatrix} A & b \\ b' & \delta \end{bmatrix}\begin{bmatrix} x \\ 1 \end{bmatrix} - \delta$$

or

$$q(x) = [x,1]\begin{bmatrix} \tilde{T} & \Pi \\ 0' & \alpha \end{bmatrix}'\begin{bmatrix} \tilde{T} & \Pi \\ 0' & \alpha \end{bmatrix}\begin{bmatrix} x \\ 1 \end{bmatrix} - \delta$$

or

$$q(x) = \begin{bmatrix} \tilde{T}x + \Pi \\ \alpha \end{bmatrix}' \begin{bmatrix} \tilde{T}x + \Pi \\ \alpha \end{bmatrix} - \delta . \tag{2.33}$$

Let

$$y = \tilde{T}x + \Pi \ . \tag{2.34}$$

Substituting in equation (2.33) the value of α^2 from equation (2.32) and the value of $\tilde{T}x + \Pi$ from equation (2.34) results in,

$$q(y) = \begin{bmatrix} y \\ \alpha \end{bmatrix}' \begin{bmatrix} y \\ \alpha \end{bmatrix} - \delta$$

$$q(y) = y'y + \alpha^2 - \delta$$

$$q(y) = y'y + \delta - |\Pi|_2^{\,2} - \delta$$

or

$$q(y) = y'y - |\Pi|_2^{\,2} = |y|_2^{\,2} - |\Pi|_2^{\,2} \ . \tag{2.35}$$

Now the minimum of the function $q(y)$ is obtained when the gradient of $q(y) \equiv \nabla\, q(y)$ is zero. Hence

$$\nabla\, q(y) = 0 \ . \tag{2.36}$$

That is,

$$y = 0 \ . \tag{2.37}$$

The minimum point x^*_u can now be obtained by back substitution in equations (2.34) and (2.37). Thus,

$$\tilde{T}x^*_u + \Pi = 0 \ .$$

Therefore we must have

$$x^*_u = -\tilde{T}^{-1}(\Pi) \ . \tag{2.38}$$

The corresponding minimum value of $q(x)$ is obtained by substituting from equation (2.37) in equation (2.35)

$$q^*_u = -|\Pi|_2^{\,2} \ . \tag{2.39}$$

(c) From part (a) $\alpha = (\delta - |\Pi|_2^2)^{1/2}$ and if $\delta = (-q^*_u + 1)$, then using equation (2.39), α can be written as

$$\begin{aligned}\alpha &= (\delta - |\Pi|_2^2)^{1/2} \\ &= (-q^*_u + 1 - |\Pi|_2^2)^{1/2} \\ &= (|\Pi|_2^2 + 1 - |\Pi|_2^2)^{1/2} \ .\end{aligned}$$

Therefore $\alpha = 1$. This concludes the proof of Theorem 2.3.

Equation (2.35) shows that $q(x)$ can be expressed in terms of the variable y and the constant vector Π. The form of equation (2.35), except for a constant, is identical to equation (2.2). Thus the determination of y accomplishes the linear least distance transformation. Further, y is defined by equation (2.34) in terms of $\tilde{T}$ and Π. Thus the LLDT is completely defined if the values of $\tilde{T}$ and Π can be computed. Knowing $\tilde{T}$ and Π will give the values of x as

$$x = \tilde{T}^{-1}(y - \Pi) = \tilde{T}^{-1}y + x^*_u \ . \tag{2.40}$$

The following algorithm shows a computational procedure to obtain Π and $\tilde{T}$. The algorithm performs the Cholesky factorization of the first n rows of the matrix S, yielding $\tilde{T}$ and Π. The values of x^*_u and q^*_u are then obtained from equations (2.28) and (2.29).

2.3 Algorithm

$\tilde{T}$ and Π are defined in equation (2.25) and the matrix S in equation (2.16).

(0) Set i = 1.

(1) Compute

$$t_{ii} = (S_{ii} - \sum_{k=1}^{i-1} |t_{ki}|^2)^{1/2} > 0 \; . \tag{2.41}$$

(2) For values of j = i+1, i+2, ..., n+1, compute

$$t_{ij} = t_{ii}^{-1}(S_{ij} - \sum_{k=1}^{i-1} t_{ki} \cdot t_{kj}) \; . \tag{2.42}$$

(3) If i = n, then go to step 4. Otherwise, set i = i + 1 and return to step 1.

(4) Recursively compute the elements of vector x^*_u,

(a) $$(x^*_u)_i = -t_{ii}^{-1}(t_{i,n+1} + \sum_{j=i+1}^{n} t_{ij} x_j) \tag{2.43}$$

(b) If i = 1, then go to step 5. Otherwise, set i = 1 - 1 and return to step 4a.

(5) Compute

$$q^*_u = -\sum_{k=1}^{n} t^2_{k,n+1} \; . \tag{2.44}$$

2.4 Possible Applications

With the development of the LLDT algorithm above the following applications show its importance and versatility.

(1) Equations (2.43) and (2.44) can be directly used in computing the unconstrained minimum of a strictly convex quadratic function.

(2) The use of the algorithm also serves as a test of the positive definiteness of the matrix A. If in equation (2.41)

t_{ii} is zero or a square root of a negative quantity then the matrix A is not positive definite.

(3) Using the LLDT the quadratic programming problem in equation (1.1) can be reduced to a least distance programming problem in equation (1.2). A numerical example of such an application is presented in section 2.5.

(4) This algorithm can also be efficiently used in solving the linear least squares problems [53].

2.5 Example [75, p. 131]

Kunzi, in his text [75], has taken a problem and used it as an illustrative example for all the quadratic programming algorithms described in that text. In order to afford a base for comparison, the same example is chosen here to demonstrate the application of the linear least distance transformation. In Chapter 5, the result of the transformation obtained here is used as an example for the algorithm for the least distance programming problem. Thus a part of Kunzi's problem is solved here and the rest of it in section 5.5.

Using equation (1.1) Kunzi's quadratic programming problem is written as

$$\min_{x\varepsilon Q}[q(x)] = \min_{x\varepsilon Q}[x'Ax+2b'x] \tag{2.45}$$

where $Q = \{x|Gx \leq h\} \subset E^n$. Now in this problem

$$A = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}, \quad b = \begin{bmatrix} -1/2 \\ -1 \end{bmatrix}, \tag{2.46}$$

$$G = \begin{bmatrix} 2 & 3 \\ 1 & 4 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad h = \begin{bmatrix} 6 \\ 5 \\ 0 \\ 0 \end{bmatrix} . \tag{2.47}$$

Using the algorithm described in section 2.3, the following results are obtained.

$$S = \begin{bmatrix} 1/2 & 0 & -1/2 \\ 0 & 1/2 & -1 \\ -1/2 & -1 & \delta \end{bmatrix} , \qquad T = \begin{bmatrix} \tilde{T} & \Pi \\ 0' & \alpha \end{bmatrix} , \tag{2.48}$$

where

$$\tilde{T} = \begin{bmatrix} 1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} \end{bmatrix} , \qquad \Pi = \begin{bmatrix} -1/\sqrt{2} \\ -\sqrt{2} \end{bmatrix} , \tag{2.49}$$

$$x^*_u = -\tilde{T}^{-1}(\underset{\sim}{\Pi}) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} , \tag{2.50}$$

$$q^*_u = -|\Pi|_2^{\,2} = -\frac{5}{2} . \tag{2.51}$$

In order to get the problem in the form of a least distance programming problem, a change of variables is performed using equation (2.40). The required form is

$$\min_{y\varepsilon L} \ell(y) = \min_{y\varepsilon L}[|y|_2^{\,2}+q^*_u] \tag{2.52}$$

where $L = \{y|By \leq C\} \subset E^n$. In view of equation (2.40) $Gx \leq h$ in equation (2.45) will reduce to

$$G[\tilde{T}^{-1}(y)-x^*_u] \leq h ,$$

or

$$G\tilde{T}^{-1}(y) \leq Gx^*_u + h . \tag{2.53}$$

Therefore

$$B = G\tilde{T}^{-1} \quad \text{and} \quad c = Gx^*_u + h \; . \tag{2.54}$$

Computing B and c from equations (2.49), (2.50) and (2.54) gives

$$B = \begin{bmatrix} 2\sqrt{2} & 3\sqrt{2} \\ \sqrt{2} & 4\sqrt{2} \\ -\sqrt{2} & 0 \\ 0 & -\sqrt{2} \end{bmatrix} \quad \text{and} \quad c = \begin{bmatrix} -2 \\ -4 \\ 1 \\ 2 \end{bmatrix} . \tag{2.55}$$

Thus, Kunzi's original problem

$$\min \{x'\begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}x + 2\begin{bmatrix} -1/2 \\ -1 \end{bmatrix}'x\} ,$$

subject to

$$\begin{bmatrix} 2 & 3 \\ 1 & 4 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}[x] \leq \begin{bmatrix} 6 \\ 5 \\ 0 \\ 0 \end{bmatrix}$$

can be rewritten as

$$\min[|y|_2^{\,2}+q^*_u] = \min[y'y-(5/2)] = [\min(y'y)] - 5/2$$

subject to

$$\begin{bmatrix} 2\sqrt{2} & 3\sqrt{2} \\ \sqrt{2} & 4\sqrt{2} \\ -\sqrt{2} & 0 \\ 0 & -\sqrt{2} \end{bmatrix}[y] \leq \begin{bmatrix} -2 \\ -4 \\ 1 \\ 2 \end{bmatrix} .$$

A solution to the above least distance programming problem is obtained in Section 5.5.

Chapter III

SPECIAL CASES IN LEAST DISTANCE PROGRAMMING

3.1 The Problem

The least distance programming problem was introduced in Chapter I, equation (1.2). In Chapter II a linear least distance transformation was derived. The need for the study of least distance programming problem is obvious, since one of the uses of the transformation may be to reduce a quadratic programming problem to a least distance programming problem. This has been illustrated by an example in Section 2.5. A brief geometrical interpretation of the least distance programming problem is in order.

In n-dimensional euclidean space E^n, the objective function of a least distance programming problem can be visualized as a n-dimensional sphere centered at the origin. The value of the objective function at any point in E^n is equal to the square of the radius of the sphere passing through it. The feasible region L in equation (1.2) is a convex polyhedron defined by m half-spaces [60, 61, 73]. The spherical objective function is strictly convex and the unique optimal solution to the least distance programming problem $y^* \varepsilon L$ lies on the sphere of the smallest possible radius.

Geometrically, the least distance programming problem appears to be one of the simplest types of constrained nonlinear programming problems. Nevertheless, the solution procedures for the least distance programming problem are non-trivial [86, 112, 113].

Further insight into the geometry of the least distance programming problem is obtained through the study of special cases considered in this chapter. Some of these special cases have simple solution procedures. This chapter essentially deals with the analysis of the special cases and their limitations.

In Section 3.2 some useful fundamental results are presented. The special problem of solving a least distance programming problem under equality constraints and full row rank assumption is considered in Section 3.3. In Sections 3.4 and 3.5 we investigate other special cases of the least distance programming problem, invoking the orthogonality property of the constraint matrix.

3.2 Some Useful Results

The following Lemmas will be useful in the development of the main results presented in this chapter, these being Theorem 3.1, 3.2 and 3.3.

Lemma 3.1

The necessary and sufficient conditions for optimality of a LDP defined in equation (1.2) are given by

(1) $2y + B'\lambda = 0$ (3.1)

(2) $By \leq c$ (3.2)

(3) $\lambda'(By-c) = 0$ (3.3)

and

(4) $\lambda \geq 0$. (3.4)

Proof:

The proof may be obtained through the well known Kuhn-Tucker Theorem [74, 75, pp. 67-75]. Independent proofs, considering specifically the least distance programming problem and not directly involving the Kuhn-Tucker Theorem, are presented by Fan [39, p. 129] and Luenberger [80, p. 71].

Lemma 3.2

The dual problems associated with the LDP defined in equation (1.2) can be stated as,

$$\max_{\lambda \geq 0} [\min_{y} (|y|_2^2 + \lambda'(By-c))] \tag{3.5}$$

or

$$\min_{\lambda \geq 0} [(1/4)\lambda' BB'\lambda + c'\lambda] . \tag{3.6}$$

Proof:

The proofs of (3.5) and (3.6) may be found in Geoffrion [47, p. 88] and Kunzi [75, p. 79], respectively.

Lemma 3.3

If P is a symmetric positive definite matrix then P^{-1} exists; furthermore, if B is a matrix with full row rank such that the product $(BP^{-1}B')$ is defined, then $(BP^{-1}B')^{-1}$ exists. Both the matrices P^{-1} and $(BP^{-1}B')^{-1}$ are also positive definite.

Proof:

The proof may be found in Boot [18, p. 17].

Lemma 3.4

If P is a positive definite matrix and $p_i > 0$ is one of its eigenvalues corresponding to the eigenvector a_i, then a_i is also an eigenvector of matrix P^{-1} corresponding to the eigenvalue $p_i^{-1} > 0$.

Proof:

By the definition of an eigenvector, we have

$$Pa_i = p_i a_i \ .$$

Or, equivalently,

$$\begin{aligned} Pa_i &= p_i P P^{-1} a_i \\ a_i &= p_i P^{-1} a_i \\ p_i^{-1} a_i &= P^{-1} a_i \ . \end{aligned}$$

Which, by definition, is the required result. This completes the proof.

Lemma 3.5

For any vector $p \varepsilon E^n$ and matrix $D = \text{diag}\{d_{11}, d_{22}, \ldots d_{nn}\}$, where $d_{ii} > 0$ for $i = 1, \ldots, n$,

$$(p_+)'D(p_+ - p) = 0 \ . \tag{3.7}$$

Proof:

Using the definition

$$(p_+)_i = \begin{cases} p_i & \text{if } p_i > 0 \\ 0 & \text{if } p_i \leq 0 \end{cases} \ , \text{ for } i = 1, \ldots, n,$$

the inner product on the left of equation (3.7) can be written as,

$$(p_+)'D(p_+ - p) = \sum_{i=1}^{n} (p_+)_i (d_{ii}) [(p_+)_i - p_i]$$

or equivalently as,

$$(p_+)'D(p_+-p) = \sum_{\forall i \ni p_i>0} (p_+)_i(d_{ii})(0) + \sum_{\forall i \ni p_i\leq 0} (0)(d_{ii})(-p_i)$$

$$= 0 \ .$$

This concludes the proof.

Lemma 3.6

A square matrix has orthonormal columns if and only if it has orthonormal rows.

Proof:

The proof may be found in Franklin [42, p. 88].

Lemma 3.7 (QR - Factorization)

If B is a [m,n] matrix with full row rank and Q is unitary matrix partitioned into Q_1 and Q_2, then B can be written uniquely as

$$B = [R|0]Q = [R|0]\left[\frac{Q_1}{Q_2}\right] = RQ_1 \ , \tag{3.8}$$

where Q_1 and Q_2 have orthonormal rows and R is a lower triangular matrix with positive diagonal elements. Furthermore,

(1) Q'_1 is a unitary basis for $R(B')$

(2) Q'_2 is a unitary basis for $N(B) = [R(B')]^{\perp}$

(3) $R(B') = R(Q'_1)$.

Proof:

The proofs may be found in Stewart [107, pp. 214-215].

3.3 Case 1: LDP with Equality Constraints

The special problem considered here can be defined as,

$$\min_{y \varepsilon L_e} [|y|_2^2] = |y^*_e|_2^2 \ , \tag{3.9}$$

where

$$L_e = \{y | By=c, \ y \varepsilon E^n, \ c \varepsilon E^m\} \ . \tag{3.9a}$$

The above problem is a special case of a least distance programming problem. The optimal solution y^*_e to this problem is simply the projection of the zero vector on the linear manifold defined by L_e in equation (3.9a).

The optimal solution to the special problem in equation (3.9) can be obtained from the results in Corollary 3.1. Corollary 3.1 is a particular case of Theorem 3.1, which determines the oblique projection of the zero vector on a linear manifold. These results also define a computational procedure.

Theorem 3.1

The optimal solution to the minimization problem,

$$\min_{y \varepsilon L_e} [|Py|_2^2] = |Py^*_e|_2^2 \tag{3.10}$$

where

$$L_e = \{y | By=c, \ \mathrm{rank}(B)=m\} \ ,$$

$$\mathrm{rank}(P) = n \ , \tag{3.11}$$

$$P \text{ is } [k,n], \ n \leq k$$

and $\quad B \text{ is } [m,n]$,

is given by

$$y^*_e = (P'P)^{-1}B'[B(P'P)^{-1}B']^{-1}c \ . \tag{3.12}$$

Proof:

The Kuhn-Tucker optimality criteria for the optimal solution y^*_e of the problem in equation (3.10) are

$$By = c \tag{3.13}$$

and

$$2P'Py + B'\lambda = 0 \ . \tag{3.14}$$

Now $(P'P)$ is a positive definite matrix since rank$(P) = n$. Thus, from Lemma 3.3, $(P'P)^{-1}$ and $[B(P'P)^{-1}B']^{-1}$ exist. Substituting the value of y from equation (3.14) into equation (3.13) the following equality is obtained:

$$[B(P'P)^{-1}B']\lambda = -2c \ ,$$

or equivalently,

$$\lambda = -2[B(P'P)^{-1}B']^{-1}c \ . \tag{3.15}$$

Now substituting for λ from equation (3.15) into equation (3.14) we have,

$$2(P'P)y - 2B'[B(P'P)^{-1}B']^{-1}c = 0 \ ,$$

which is the same as

$$y^*_e = (P'P)^{-1} \ B'[B(P'P)^{-1}B']^{-1}c \ .$$

This completes the proof of Theorem 3.1.

Corollary 3.1

The optimal solution to the minimization problem in equation (3.9) is

$$y^*_e = B'(BB')^{-1}c = B'D^{-1}c \ , \tag{3.16}$$

where $D = BB'$ and the matrix P is an identity matrix.

If the matrix B is non-singular, then equation (3.16) simply states that the optimal solution to the problem in equation (3.9) is

$$y^*_e = B^{-1}c \ . \qquad (3.17)$$

This is to be expected since in this case there is only one feasible solution and therefore it must be the optimal solution. If the matrix B satisfies a weaker condition that the rank(B) = m, then equation (3.16) represents the optimal solution to the problem in equation (3.9).

3.4 Case 2: LDP with Row Orthogonality

This section capitalizes on the special characteristics of the least distance programming problems in which the matrix B in equation (1.2) has orthogonal rows. Problems with this property will be referred to as "row-orthogonal least distance programming problems."

Geometrically, this would imply that each of the hyperplanes forming the constraint set is orthogonal to the rest. An analysis resulting in the optimal solution under the above assumption is presented in Section 3.4.1. Section 3.4.2 presents a further generalization of the results and Section 3.4.3 explores the potential of the results from Section 3.4.1 in solving a least distance programming problem through row orthogonalization of matrix B.

3.4.1 An Analysis

Theorem 3.2 shows the derivation of the optimal solution for the least distance programming problem under the row-orthogonality assumption. The results derived here may also be used directly for computational purposes.

Theorem 3.2

The optimal solution to the row-orthogonal least distance programming problem is given by

$$y^* = -B'd_+ \tag{3.18}$$

where

$$d = -D^{-1}c \ , \tag{3.19}$$

$$D = BB' \tag{3.20}$$

and d_+ is obtained by replacing the negative elements of d by zero.

Proof:

Because the rows of matrix B are orthogonal, the matrix D = BB' has positive diagonal elements and the off-diagonal elements are zero. The Kuhn-Tucker optimality conditions for the least distance programming problem are given by Lemma 3.1. Those are,

$$2y + B'\lambda = 0 \tag{3.1}$$

$$By \leq c \tag{3.2}$$

$$\lambda'(By-c) = 0 \tag{3.3}$$

and

$$\lambda \geq 0 \ . \tag{3.4}$$

Equations (3.1) and (3.2) can be combined into

$$D\lambda \geq -2c \ . \tag{3.21}$$

Since D is a diagonal matrix with positive elements, equation (3.21) can now be written as

$$\lambda \geq -2D^{-1}c \ . \tag{3.22}$$

Let d be defined by equation (3.19). Now if λ_1 satisfies equations (3.22) and (3.4), then in view of equation (3.19), it can be written as

$$\lambda_1 = 2d_+ \ . \tag{3.23}$$

Substituting the value of λ_1 from equation (3.23) into equation (3.1) yields the desired solution,

$$y_1 = -B'd_+ \ . \tag{3.24}$$

The above solution y_1 will be optimal if λ_1 and y_1 also satisfy the last of the Kuhn-Tucker conditions in Lemma 3.1, that is, equation (3.3),

$$(\lambda_1)'(By_1-c) = 0 \ . \tag{3.25}$$

Substituting λ_1 from equation (3.23), y_1 from equation (3.24) and d from equation (3.19) into the left side of equation (3.25), the following steps illustrate that equation (3.25) holds:

$$\begin{aligned}(\lambda_1)'(By_1-c) &= (2d_+)'(-BB'd_+-c) = -2(d_+)'(Dd_++c) \\ &= -2(d_+)'D(d_++D^{-1}c) \\ &= -2(d_+)'D(d_+-d) = 0 \ . \qquad \text{(from Lemma 3.5)}\end{aligned}$$

Thus,

$$y^* = y_1 = -B'd_+ \ .$$

This concludes the proof of Theorem 3.2.

Corollary 3.2

The optimal solution to the row-orthogonal least distance programming problem is given by

$$y^* = -B'd_+ \tag{3.26}$$

where

$$d = -c \quad . \tag{3.27}$$

Proof:

The proof follows directly from Theorem 3.2 as D = BB' = I.

3.4.2 Generalization of Results

The results from Theorem 3.2 which apply to the least distance programming problem can be extended further to solve the more general problem,

$$\min_{y \in L} [|Py|_2^2] = |Py^*|_2^2 \quad , \tag{3.28}$$

where

$$L = \{y | By \leq c, \ \text{rank}(B) = m\} \tag{3.29}$$

and A = P'P is positive definite.

If we let P = I, then the problem in equation (3.28) reduces to a least distance programming problem.

Geometrically, the contours of the function

$$f(y) = |Py|_2^2 \tag{3.30}$$

are hyperellipsoids in E^n with the directions of their principal axes being the orthonormal set of eigenvectors of matrix A. Theorem 3.3 uses a relationship between the principal axes of the hyperellipsoids and the rows of the matrix B to obtain an optimal solution to the special problem in equation (3.28).

Theorem 3.3

If the rows of matrix B constitute an orthogonal set of vectors in E^n pointing in the direction of eigenvectors of matrix A = P'P, then the solution to the problem in equation (3.28) is given by

$$y^* = -A^{-1}B'e_+ \tag{3.31}$$

where

$$e = -[BA^{-1}B']^{-1}c \ , \tag{3.32}$$

and e_+ is obtained by replacing the negative elements of e by zeros.

Proof:

The Kuhn-Tucker optimality criteria for the problem in equation (3.28) can be written as

$$2Ay + B'\lambda = 0 \tag{3.33}$$

$$By \leq c \tag{3.34}$$

$$\lambda \geq 0 \tag{3.35}$$

and

$$\lambda'(By-c) = 0 \ . \tag{3.36}$$

Equation (3.33) and (3.34) can be combined into

$$BA^{-1}B'\lambda \geq -2c \ . \tag{3.37}$$

Let $\{a_1,a_2,\ldots,a_m\}$ be the subset of eigenvalues of matrix A corresponding to the eigenvectors forming the columns of matrix B'. Now using the definition of eigenvectors and Lemma 3.4 we obtain

$$A^{-1}B' = B'\begin{bmatrix} 1/a_1 & & 0 \\ & \ddots & \\ 0 & & 1/a_m \end{bmatrix} . \tag{3.37a}$$

Thus from equation (3.37a), equation (3.37) can be reduced to

$$BB'\begin{bmatrix} 1/a_1 & & 0 \\ & \ddots & \\ 0 & & 1/a_m \end{bmatrix} \lambda \geq -2c \ . \tag{3.38}$$

Or equivalently,

$$E\lambda \geq -2c \ , \tag{3.39}$$

where

$$E = BA^{-1}B' = BB' \begin{bmatrix} 1/a_1 & & 0 \\ & \ddots & \\ 0 & & 1/a_m \end{bmatrix} . \tag{3.40}$$

The matrix E is a diagonal matrix with positive elements, since D = BB' is a diagonal matrix with positive elements and $a_i > 0$, $i = 1, \ldots, m$. Hence the inverse of E exists and equation (3.39) can now be written as

$$\lambda \geq -2E^{-1}c \ . \tag{3.41}$$

Let e be defined by equation (3.32). If λ_1 satisfies equations (3.41) and (3.35), then in view of equation (3.32), it can be written as

$$\lambda_1 = 2e_+ \ . \tag{3.42}$$

Substituting the value of λ_1 from equation (3.42) into equation (3.33) yields the desired solution,

$$y_1 = -A^{-1}B'e_+ \ . \tag{3.43}$$

The above solution y_1 will be optimal if λ_1 and y_1 also satisfy the last of the Kuhn-Tucker conditions in equation (3.36), i.e.,

$$(\lambda_1)'(By_1-c) = 0 \ . \tag{3.44}$$

Substituting λ_1 from equation (3.42), y_1 from equation (3.43), E from equation (3.40) and e from equation (3.32) into the left side of equation (3.44), the following sequence of steps shows that equation (3.44) holds:

$$(\lambda_1)'(By_1-c) = (2e_+)'[-BA^{-1}B'e_+-c]$$
$$= -2(e_+)'(Ee_++c) = -2(e_+)'E(e_++E^{-1}c)$$
$$= -2(e_+)'E(e_+-e) = 0 \ . \qquad \text{(from Lemma 3.5)}$$

Thus,

$$y^* = y_1 = -A^{-1}B'e_+ \ .$$

This concludes the proof of Theorem 3.3.

The results in Theorem 3.2 can be obtained as a special case of Theorem 3.3. If we let $A = I$ in Theorem 3.3, then Theorem 3.2 follows directly from it. Thus the row-orthogonal least distance programming problem is a special case of the problem considered in Theorem 3.3.

3.4.3 Orthogonality Transformation

In this section the application of Theorem 3.2 in solving a least distance programming problem is considered. This discussion brings out the limitations of Theorem 3.2.

Since Theorem 3.2 gives a direct solution for a row-orthogonal least distance programming problem and since a general least distance programming problem does not necessarily possess this property, the natural procedure would be to first use an orthogonalization transformation on the rows of matrix B and then apply Theorem 3.2 to solve the transformed problem. The following discussion shows that such a transformation for a system of linear inequalities does not exist, hence the use of Theorem 3.2 is limited to only those problems that naturally possess the row orthogonality property.

Let us consider the least distance programming problem defined in equation (1.2),

$$\min[|y|_2^{\,2}] \qquad (3.45)$$

subject to

$$By \leq c \ , \qquad (3.46)$$

assuming that the constraint matrix B has full row rank, that is, rank(B) = m. Because of this assumption the rows of the matrix B form an independent set of vectors. However, if these vectors are also orthogonal then Theorem 3.2 can be used to solve the above problem. In the case that these vectors are not orthogonal then one may use the Gram-Schmidt procedure to orthogonalize these vectors prior to the application of Theorem 3.2. But, due to the presence of linear inequalities in equation (3.46) this orthogonalization approach does not yield the desired result. This conclusion is derived from the following reasoning.

Firstly, we observe the well-known fact that the Gram-Schmidt procedure is nothing but a linear coordinate (basis) transformation. Furthermore, an analysis of the Gram-Schmidt orthogonalization procedure shows its equivalence to the well-known QR-transformation of matrices [53]. Therefore, instead of the Gram-Schmidt procedure we shall use the QR-transformation for the orthogonalization of the rows of matrix B. The relevant theoretical structure of the QR-transformation is presented in Lemma 3.7.

Now, using equation (3.8) from Lemma 3.7, the least distance programming problem in equation (3.45) and equation (3.46) can be reduced to

$$\min[|y|_2^{\,2}] \tag{3.47}$$

subject to

$$(RQ_1)y \leq c \ , \tag{3.48}$$

where RQ_1 is the QR-transformation of the matrix B. The matrix Q_1 has orthogonal rows and R is a lower-triangular, nonsingular matrix. The desired orthogonalization of the least distance programming problem in equations (3.45) and (3.46) to a form usable in Theorem 3.2 would have been achieved from equations (3.47) and (3.48) if it were only possible to represent equation (3.48) as

$$Q_1 y \leq R^{-1}c \ . \tag{3.49}$$

Unfortunately, since equation (3.48) is an inequality, in general it is not equivalent to equation (3.49). However, if equation (3.48) were an equality, then we would have reduced it to

$$Q_1 y = R^{-1}c \ , \tag{3.50}$$

and this would have been the required form to be used in Theorem 3.2. But, the special case of the least distance programming problem under equality constraints has already been considered in Theorem 3.1.

Therefore, in general, due to the presence of inequalities in equation (3.48) an orthogonalized form of the least distance programming problem usable in Theorem 3.2 cannot be obtained.

3.5 Case 3: LDP with Column Orthogonality

A least distance programming problem with the columns of matrix B being orthogonal will be referred to as a "column-orthogonal least distance programming problem." As established in Theorem 3.2, the solution of a row-orthogonality least distance programming problem can be obtained in a straightforward manner. Now, the natural question becomes: Does there exist a simple procedure for solving a column-orthogonal least distance programming problem? The answer to this question is no. An explanation is as follows.

Using Lemma 3.1 the Kuhn-Tucker optimality condition for a least distance programming problem can be written as

$$2y + B'\lambda = 0$$

$$By \leq c$$

$$\lambda \geq 0$$

and $$\lambda(By-c) = 0 \ ,$$

or equivalently, as

$$-2BB'\lambda \geq c \tag{3.51}$$

$$\lambda \geq 0 \tag{3.52}$$

and

$$\lambda'(By-c) = 0 \ . \tag{3.53}$$

In Theorem 3.2, because of row-orthogonality of matrix B, the matrix D = BB' was a diagonal matrix with positive elements. Thus it was possible to write equation (3.51) as

$$\lambda \geq -(D)^{-1}c \ . \tag{3.54}$$

In general, the above results cannot be obtained under the assumption of column-orthogonality of the matrix B. This prevents a development similar to Section 3.4 using the column-orthogonality property of constrained matrices of the least distance programming problem.

However, it is interesting to note that, if matrix B is non-singular, then from Lemma 3.6 the column-orthogonality of matrix B also implies its row-orthogonality and vice versa. Thus the knowledge of column-orthogonality can only be used in the above sense to establish row-orthogonality.

Chapter IV

A DUAL LEAST DISTANCE PROGRAM

4.1 Introduction

The concept of duality was first introduced by Gale, Kuhn and Tucker [71, p. 317] in the theory of linear programming and games. It plays an important role both in theoretical and computational aspects of linear programming [30]. In recent years there has been an increasing interest in the duality theory of nonlinear programming [47,80]. This can be attributed to the fact that a dual problem often provides better understanding of the original problem and sometimes is easier to solve. A great majority of duality relationships involve the use of either the classical Lagrange multiplier theory or Fenchel's duality theory [100, Sec. 31].

The primal and dual problems are related in the sense that if one of these problems is a minimization problem, the other is generally, though not necessarily, a maximization problem. The two problems are equivalent in that the optimal values of the objective function are equal and a solution of either problem leads to solution of the other.

The dual of the following least distance programming problem (primal)

$$\min\ [|y|_2^2] \tag{4.1}$$

subject to

$$By \leq c\ , \tag{4.2}$$

is given by

$$\min \; [(1/4)\lambda' BB' \lambda + c' \lambda] \tag{4.3}$$

subject to

$$\lambda \geq 0 \; , \tag{4.4}$$

where λ is a vector of Lagrange multipliers. A dual of the least distance programming problem using Fenchel's conjugate duality theory is given in Luenberger [80, p. 209]. A solution procedure specifically for the dual problem in equation (4.3) has been developed by Hildreth and D'Esopo [75, p. 78].

In this chapter a new form of the dual least distance programming problem is presented. This dual least distance programming problem makes use of the concept of generalized inverses of rectangular matrices [15]. In this connection some basic results involving generalized inverses and partial isometries are included in Section 4.2. The dual problem, referred to as "generalized dual," is derived in Section 4.3. Section 4.4 presents Newton's method for the solution of the generalized dual problem. The relationship of the generalized dual problem to the partial isometries is also included in this section.

4.2 Generalized Inverses and Partial Isometries

Definition 4.1

A [n,m] matrix $A^{\dagger}$ is said to be the unique generalized inverse (pseudoinverse) of a [m,n] matrix A if $A^{\dagger}$ satisfies the following four properties [15]:

$$(1)\quad AA^{\dagger}A = A \tag{4.5}$$

$$(2)\quad A^{\dagger}AA^{\dagger} = A^{\dagger} \tag{4.6}$$

$$(3)\quad (AA^{\dagger})' = AA^{\dagger} \tag{4.7}$$

$$(4)\quad (A^{\dagger}A)' = A^{\dagger}A \ , \tag{4.8}$$

where

$$AA^{\dagger} = P_{R(A)} = P_{N(A')^{\perp}} = I - P_{N(A')} \tag{4.9}$$

and

$$A^{\dagger}A = P_{R(A')} = P_{N(A)^{\perp}} = I - P_{N(A)} \ . \tag{4.10}$$

The perpendicular projections on the range space of matrix A and on the orthogonal complement of null space of A' are denoted by $P_{R(A)}$ and $P_{N(A')^{\perp}}$ respectively.

Definition 4.2

A [m,n] matrix V is called a partial-isometry if the linear transformation

$$Vx = z \quad \text{for all } x \varepsilon N(V)^{\perp} \tag{4.11}$$

preserve distances, i.e.,

$$|Vx_1 - Vx_2|_2 = |x_1 - x_2|_2 \quad \text{for all } x_1, x_2 \varepsilon N(V)^{\perp} \ . \tag{4.12}$$

Or equivalently,

$$|Vx|_2 = |x|_2 \quad \text{for all } x \varepsilon N(V)^{\perp} \ , \tag{4.13}$$

or

or

$$(Vx_1, Vx_2) = (x_1, x_2) \quad \text{for all } x_1 x_2 \varepsilon N(V)^{\perp} . \tag{4.14}$$

A partial isometry becomes unitary if

$$N(V) = \{0\} \tag{4.15a}$$

or that

$$N(V)^{\perp} = E^n . \tag{4.15b}$$

Lemma 4.1

If A is a [m,n] matrix such that

$$Ax = z \quad \text{for all } x\varepsilon E^n , \tag{4.16}$$

then

$$E^n = [N(A) \oplus N(A)^{\perp}] = [N(A) \oplus R(A')] \tag{4.17}$$

and

$$E^m = [R(A) \oplus R(A)^{\perp}] = [R(A) \oplus N(A')] , \tag{4.18}$$

where

$$R(A) = \{z | z=Ax, z\varepsilon E^m, x\varepsilon E^n\} \subset E^m \tag{4.19}$$

and

$$N(A) = \{x | 0=Ax, 0\varepsilon E^m, x\varepsilon E^n\} \subset E^n . \tag{4.20}$$

Proof:

The proof may be found in Halmos [57, p. 88], Luenberger [80, p. 157] and Ben-Israel and Charnes [15, p. 682].

Lemma 4.2

A $[m,n]$ matrix V is a partial-isometry if and only if any one of the following two statements holds:

$$(1) \quad V^{\dagger} = V' \tag{4.21}$$

$$(2) \quad V'Vx = x \quad \text{for all } x \in N(V)^{\perp} . \tag{4.22}$$

Proof:

The proof may be found in Erdelyi [36, p. 456].

4.3 Generalized Dual Problem

The motivation for the generalized dual problem is obtained from the following observation. Assuming that the matrix B of constraints of the primal least distance programming problem in equation (4.2) is non-singular and

$$w = By , \tag{4.23}$$

equation (4.2) can be written as,

$$y = B^{-1}w \tag{4.24}$$

and

$$w \leq c . \tag{4.25}$$

Thus, using equations (4.24) and (4.25) the primal least distance programming problem in equations (4.1) and (4.2) can be reduced to the problem

$$\min \, [|B^{-1}w|_2{}^2] \tag{4.26}$$

subject to

$$w \leq c , \tag{4.27}$$

where equations (4.23) and (4.24) uniquely relate the variable w to y. The problem obtained in equations (4.26) and (4.27) in the presence of a non-singular B matrix is a dual to the least distance programming problem in equations (4.1) and (4.2).

The above result was only possible due to the non-singularity assumption of matrix B. The natural question now arises: Is it possible to obtain a generalization of these results without the assumption of non-singularity of the B matrix? The following theorem and its proof presents the generalized dual problem for the case when B is not necessarily non-singular.

Theorem 4.1

The primal least distance programming problem,

$$\min_{y \varepsilon L_y} [|y|_2^2] \tag{4.28}$$

where

$$L_y = \{y|By \leq c,\ c \not\geq 0\}\ , \tag{4.29}$$

is equivalent to the generalized dual problem,

$$\min_{w \varepsilon L_w} [|B^{\dagger}w|_2^2] \tag{4.30}$$

where

$$L_w = \{w|w \leq c,\ c \not\geq 0\}\ . \tag{4.31}$$

Furthermore, for every $y \varepsilon L_y$ we have,

$$y = B^{\dagger}w\ , \tag{4.32a}$$

where we define the variable w as,

$$w = By \ . \tag{4.32b}$$

The restriction $c \not\perp 0$ in the definition of sets L_y and L_w only insures that the zero vector is not contained in either of the two sets. Without the restriction the optimal solution is trivially obtained as $y^* = 0$.

Proof:

Because $c \not\perp 0$ the null space of the matrix B,

$$N(B) = \{y | By = 0\}$$

is infeasible with respect to the region L_y of the primal least distance programming problem. That is,

$$y \epsilon N(B) \longrightarrow y \not\epsilon L_y \ , \tag{4.33}$$

or equivalently,

$$y \epsilon L_y \longrightarrow y \not\epsilon N(B) \ . \tag{4.34}$$

Thus from equation (4.17) in Lemma 4.1 and equation (4.34) it follows that

$$y \epsilon L_y \longrightarrow y \epsilon N(B)^{\perp} \ . \tag{4.35}$$

Now let

$$By = w \ . \tag{4.36}$$

The general solution of equation (4.36), when solvable, is given by [15, p. 669],

$$y = B^{\dagger} w + (I - B^{\dagger} B) z \ ; \ z \text{ arbitrary } \epsilon E^{n} \ , \tag{4.37}$$

where $B^\dagger$ is the generalized inverse of the matrix B. Multiplying both sides of equation (4.37) by $B^\dagger B$ we obtain

$$B^\dagger By = B^\dagger BB^\dagger w + (B^\dagger B - B^\dagger BB^\dagger B)z \;;\; z \text{ arbitrary } \varepsilon E^n \,. \qquad (4.38)$$

Using equations (4.6) and (4.10), equation (4.38) can be shown equivalent to

$$P_{N(B)^\perp}(y) = B^\dagger w + (B^\dagger B - B^\dagger B)z \;;\; z \text{ arbitrary } \varepsilon E^n \,,$$

which is the same as

$$P_{N(B)^\perp}(y) = B^\dagger w \,. \qquad (4.39)$$

Now from equation (4.35) we have

$$y \varepsilon L_y \subset N(B)^\perp \,. \qquad (4.40)$$

Therefore every feasible vector $y \varepsilon L_y$ must also satisfy

$$P_{N(B)^\perp}(y) = y \,. \qquad (4.41)$$

Thus equation (4.39) for every vector $y \varepsilon L_y$ reduces to

$$y = B^\dagger w \,. \qquad (4.42)$$

Substituting from equations (4.36) and (4.42) into equations (4.28) and (4.29) the generalized dual problem is given by,

$$\min_{w \varepsilon L_w} [|B^\dagger w|_2^{\,2}] \,,$$

where

$$L_w = \{w \mid w \leq c,\; c \not\perp 0\} \,.$$

Finally, the results in equations (4.32a) and (4.32b) follow directly from equations (4.42) and (4.36). This completes the proof of Theorem 4.1.

Having completed the derivation of the generalized dual problem we will now prove that the objective function of the generalized dual problem in equation (4.30) is convex.

Theorem 4.2

The function $f\colon E^m \to E^1$,

$$f^2(w) = |B^\dagger w|_2^{\,2} \tag{4.43}$$

is convex. Furthermore, if the matrix B has full row rank then the function $f^2(w)$ is strictly convex.

Proof:

Since the hessian matrix $(B^\dagger)'B^\dagger$ of the function

$$f^2(w) = |B^\dagger w|_2^{\,2} = w'(B^\dagger)'B^\dagger w \tag{4.44}$$

is positive semidefinite, the function $f^2(w)$ is convex by definition. Furthermore, since $\text{rank}(B) = \text{rank}(B^\dagger)$ [15, p. 669] and if $\text{rank}(B) = m$ then the hessian matrix $(B^\dagger)'B^\dagger$ will be positive definite and the function $f^2(w)$ strictly convex. This completes the proof of Theorem 4.2.

In this section we have derived a generalized dual least distance programming problem. Like the Lagrangian dual least distance programming problem in equations (4.3) and (4.4), the generalized dual problem can be solved by any one of the many convex quadratic programming techniques.

If the constraint matrix B has full row rank then both the Lagrangian dual problem and the generalized dual problem have strictly convex objective functions. In this case, these problems can be solved by the strictly-convex quadratic programming technique of Theil and Van de Panne [75, p. 85]. Additionally, the simple form of the constraints in these problems considerably reduces the computations in each iteration of the Theil and Van de Panne procedure.

The objective function of the generalized dual problem is expressed in terms of $B^{\dagger}$, the generalized inverse of a $[m,n]$ matrix B. The computation of $B^{\dagger}$ is not difficult. There are various procedures available to accomplish this. One such procedure, complete with a computer program, may be found in Wilkinson [118]. This procedure originally proposed by G. H. Golub uses the concept of singular value decomposition [3].

4.4 Solving the Generalized Dual Problem

It has been indicated earlier that the generalized dual problem can be solved by any one of the many convex quadratic programming techniques. In this section we will consider two additional procedures for solving the generalized dual least distance programming problem. The first procedure is an application of the modified Newton method developed by Ben-Israel [11,12,13,14]. Ben-Israel developed the modified Newton method for the solution of constrained non-linear least squares problems. Ben-Israel's procedure is not applicable to either the primal least distance programming problem or the Lagrangian dual of the problem. However, it will be shown here that Ben-Israel's procedure can in fact be used for the generalized dual problem.

The second approach to the solution of generalized dual problems uses the concept of partial isometries. Theorem 4.3 establishes the conditions under which the solution of the generalized dual problem is obtained almost by inspection.

4.4.1 Newton's Method:

The modified Newton method considered here is due to Ben-Israel [14]. Ben-Israel extended Newton's method for solving linear and nonlinear equations [12] to apply to the nonlinear least squares problem over convex sets [13]. The application of Ben-Israel's procedure to a class of nonlinear programming problems, of which the generalized dual problem is a special case, is presented below.

Let S be a closed convex set in E^n and f a function such that f^2 is convex and differentiable. Consider the problem

$$\min_{w \in S} f^2(w) \; . \tag{4.45}$$

The optimal solution to the above problem w* is a stationary point of a convex function g(w) [14, p. 349], where

$$g(w) = f^2(w) + |P_{S\perp}[w - \tfrac{1}{2}\nabla f^2(w^*)]|_2^{\,2} \; . \tag{4.46}$$

The optimal solution to the above problem can be obtained by iteratively solving a sequence of subproblems [14, p. 350]. Newton's method is used to solve these subproblems. Thus, starting at some known vector w^0 the optimal solution w* is obtained as a limit of the solution w^k of the sequence of subproblems. The following iteration defines Newton's method used by Ben-Israel to obtain w^{k+1} from w^k,

$$w^{k+1} = w^k - ([J_f(w^k)]'f(w^k)+I)^{-1} ([J_f(w^k)]'f(w^k) + P_{S^\perp} [w^k - \frac{1}{2} \nabla f^2(w^k)]) , \tag{4.47}$$

where $[J_f(w^k)]'$ is the transpose of Jacobian matrix at w^k. If the sequence $\{w^k | k = 0, 1, \ldots; \; w^0 \text{ given}\}$ so obtained converges to w^*, then w^* is a stationary point of function $g(w)$ in equation (4.46) and thus solves the minimization problem in equation (4.45).

The method presented above cannot be directly used to solve the Lagrangian dual least distance programming problem, since it is not in the form of the minimization problem given in equation (4.45). However, the difficulty in solving the primal least distance programming problem, by Newton's method, is quite different. Comparing equations (4.1) and (4.2) with equation (4.45) we have,

$$f(y) = |y|_2 \tag{4.48}$$

and

$$S = \{y | By \leq c\} . \tag{4.49}$$

Thus in the iterative scheme in equation (4.47) all terms except for the perpendicular projection on $S^\perp$,

$$P_{S^\perp} [y^k - \frac{1}{2} \nabla f^2(y^k)] , \tag{4.50}$$

are easily obtained. The computation of this projection will require the solution of a least distance programming problem at each iteration. This is quite undesirable since the subproblem is itself the main problem.

Now we consider the generalized dual problem in equations (4.30) and (4.31). Comparing it with the problem in equation (4.45) we have,

$$f(w) = |B^{\dagger}w|_2 \tag{4.51}$$

and

$$S = L_W = \{w | w \leq c,\ c \ngtr 0\} .$$

In this case the iterative scheme in equation (4.47) is easily computable, since the orthogonal projection of any $z \varepsilon E^m$ on $S^{\perp}$ is simply

$$P_{S^{\perp}}(z) = P_{L_W^{\perp}}(z) = \begin{cases} z_i & \text{if } z_i > c_i \\ & \\ c_i & \text{if } z_i \leq c_i \end{cases}, \quad i = 1, \ldots, m. \tag{4.52}$$

Thus Ben-Israel's version of Newton's method is applicable to the generalized dual form of the least distance programming problem, whereas it is not applicable to the primal and Lagrangian dual forms of the least distance programming problem. The only computational disadvantage in using the generalized dual problem may be in computing $B^{\dagger}$. This concludes the discussion of Newton's method.

4.4.2 Partial Isometries:

In Lemma 4.2 the partial isometries were shown to be a class of norm-preserving linear transformations in Euclidean space [36,37]. The relevance of this transformation to the solution of a generalized dual least distance programming problem is obvious from the following theorem.

Theorem 4.3

If the matrix $B^{\dagger}$ is a partial isometry then the optimal solution w^* of the minimization problem

$$\min_{w \varepsilon L_w} [|B^{\dagger}w|_2^{\,2}] , \tag{4.53}$$

where

$$L_w = \{w | w \leq c, \; c \not\geq 0\} \tag{4.54}$$

is given by

$$w^* = -(-c)_+ \; , \tag{4.55}$$

where $(-c)_+$ is obtained by replacing the negative elements of $(-c)$ by zeros.

Proof:

From equation (4.13) in Definition 4.2, for every vector $w \varepsilon N(B^{\dagger})^{\perp}$ the following holds:

$$|B^{\dagger}w|_2^{\,2} = |w|_2^{\,2} \; . \tag{4.56}$$

Now, from Theorem 4.1, and because $0 \not\in L_y$, we have

$$\min_{w \varepsilon L_w} [|B^{\dagger}w|_2^{\,2}] = \min_{y \varepsilon L_y} [|y|_2^{\,2}] \neq 0 \; , \tag{4.57}$$

or

$$\min_{w \varepsilon L_w} [|B^{\dagger}w|_2^{\,2}] \neq 0 \; . \tag{4.57a}$$

Now, from equation (4.57a) it follows that

$$L_w \subset N(B^\dagger)^\perp . \qquad (4.58)$$

Using equations (4.56) and (4.58), the minimization problem in equation (4.53) can be rewritten as

$$\min_{w \in L_w} [|w|_2^{\,2}] . \qquad (4.59)$$

The optimal solution of the above problem is obtained directly from Corollary 3.2 as

$$w^* = -(-c)_+ . \qquad (4.60)$$

This concludes the proof of Theorem 4.3.

If the generalized inverse of the constrained matrix B is a partial isometry, then the optimal solution y* to the primal least distance programming problem

$$\min_{y \in L_y} [|y|_2^{\,2}] , \qquad (4.61)$$

where

$$L_y = \{y | By \leq c, c \not\perp 0 , \qquad (4.62)$$

can be obtained from Theorem 4.3 and equations (4.32a) and (4.32b) as,

$$y^* = B^\dagger w^* = -B^\dagger(-c)_+ . \qquad (4.63)$$

It is clear that the results of Theorem 4.3 can be applied to the solution of only those problems having the generalized inverse of the constraint matrix to be a partial isometry. It is indeed difficult to identify the set of matrices that qualify. The preliminary findings of

Erdelyi [19, Theorem 3, p. 212] identifies a group of matrices that do belong to the above set. This group identified by Erdelyi contains any square matrix A whose Darzin pseudoinverse A^D [19, p. 205] is normal and the matrix A satisfies the following condition,

$$A^{m-1}(A')^m = A^m(A')^{m+1}, \; m = 1, 2, \ldots . \tag{4.64}$$

In view of this, one must conclude that Theorem 4.3 has only a very limited application in solving a least distance programming problem.

4.5 Example

This is an example of the solution of a least distance programming problem using the results of Theorem 4.3. Consider the following primal least distance programming problem,

$$\min_{y \in L_y} [|y|_2^2] \tag{4.65}$$

where

$$L_y = \{y | By \leq c, \; c \nleq 0\} . \tag{4.66}$$

Given

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}+1}{4} & -\frac{\sqrt{3}-1}{4} \\ 0 & -\frac{\sqrt{3}+1}{4} & \frac{\sqrt{3}-1}{4} \end{bmatrix} \tag{4.67}$$

and

$$c = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} \tag{4.68}$$

Computing $B^\dagger$ we obtain,

$$B^\dagger = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}+1}{4} & -\frac{\sqrt{3}+1}{4} \\ 0 & -\frac{\sqrt{3}-1}{4} & \frac{\sqrt{3}-1}{4} \end{bmatrix} . \tag{4.69}$$

Now since equation (4.21) holds for $B^\dagger$ and B, the generalized inverse $B^\dagger$ is a partial isometry. The solution to the problem in equations (4.65) through (4.68) is given by equation (4.63) as

$$y^* = -B^\dagger(-c)_+ = -B^\dagger(1,0,0)' , \tag{4.70}$$

or equivalently

$$y^* = (-1,0,0)' . \tag{4.71}$$

Chapter V

A CUTTING PLANE ALGORITHM FOR THE LEAST DISTANCE PROGRAMMING PROBLEM

5.1 Introduction

In examining the various algorithms for solving a linearly constrained non-linear programming problem, two broad and important classes of algorithms can be distinguished. The first class of algorithms consists of those based only on the consideration of the active constraints at each iteration, while the second class of algorithms consists of those based on the consideration of the entire feasible region at each iteration. In general, the convergence of the latter class of algorithms is easier to establish but the subproblem at each iteration is of a much higher degree of complexity than the algorithms of the former class.

Techniques such as those of Goldfarb [51], Rosen [101], Theil and Van de Panne [75, p. 85], and Zountendijk [124] all belong to the former class of the algorithms, while the algorithms based on conditional gradient, gradient projection and Newton's method as discussed in Levitin and Polyak [78] are examples of the latter class of algorithms.

The cutting plane algorithm for the least distance programming problem presented here can be considered as a combination of the above two types. It is expected to incorporate the good characteristics of both classes of algorithms. The main objective of the algorithm is to minimize computational effort. The algorithm is designed to iteratively reduce the feasible region of the least distance programming problem through the addition of cutting plane(s). These iterations finally converge to a feasible region containing only the optimal point.

The cutting plane algorithm for the least distance programming is presented in Section 5.2. Section 5.3 shows the details of computational procedure. Section 5.4 presents convergence of the cutting plane algorithm and Section 5.5 contains numerical examples of least distance programming problems.

5.2 The Cutting Plane Algorithm

The least distance programming can be redefined as:

$$\min_{y \in L_I} |y|_2^{\,2} = |y^*|_2^{\,2} \tag{5.1}$$

where

$$L_I = \{y \mid (b_i)'y \leq c_i,\ i \in I\}\ , \tag{5.2}$$

$$I\ = \{1,\ 2,\ \ldots,\ m\}\ , \tag{5.3}$$

and

$$0 \notin L_I\ . \tag{5.4}$$

Note that if the set L_I contains the origin then the origin itself is the optimal point, i.e.,

$$y^* = 0\ . \tag{5.5}$$

Thus the assumption in equation (5.4) eliminates the above trivial case.

To solve the least distance programming problem in equation (5.1) the cutting plane algorithm starts with a known initial feasible interior point of the set L_I. The optimal solution is then obtained through an

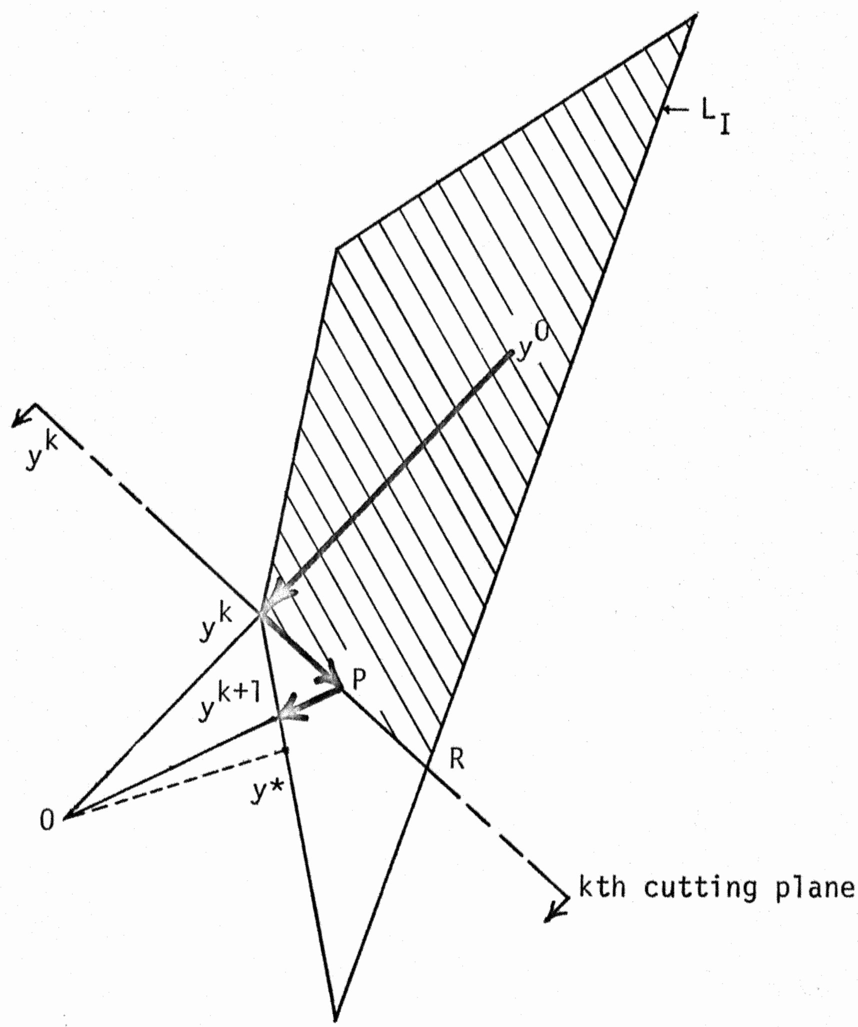

Figure 1

The Cutting Plane Algorithm

iterative procedure. The following discussion explains this iterative procedure.

For the kth iteration of the algorithm let y^0 represent the known initial feasible interior point. The first step of the algorithm involves moving along the ray emanating from y^0 and pointing towards the origin. Due to the nature of the objective function this direction is also the gradient direction (Figure 1). Since the origin is infeasible, there will exist a point y^k on this ray such that it is on the boundary of region L_I. If y^k is optimal then the algorithm terminates. If y^k is non-optimal then a cutting hyperplane orthogonal to vector y^k is generated. This cutting plane is then added to the constraint hyperplanes in an inequality form such that the half space containing y^0 is infeasible. The shaded area in Figure 1 indicates the portion of the feasible region eliminated by introducing this cutting plane.

The algorithm now attempts to find a feasible direction d^k from the point y^k such that d^k lies on the newly introduced cutting plane. Such a direction will always exist,for the non-existance of such a direction implies the optimality of y^k (Theorem 5.1). Once the direction d^k is found,a step in this direction is taken such that the point

$$z^k = y^k + \delta^k \lambda^k d^k \tag{5.6}$$

is in the interior of the original feasible region. (Note that z^k is on the boundary of the newly created constraint.) This is done by manipulating the parameters δ^k and λ^k. The new point z^k is redefined as y^0 (point P in Figure 1) and serves as a starting point for the (k+1)st iteration of the cutting plane algorithm. The mathematical details of the algorithm are presented below.

Step (0): Obtain an initial feasible interior point: $y^0 \varepsilon L_I$. Set $k = 1$ and $m = \#(I)$. Choose δ^k such that $0 < \delta^k < 1$ and $\varepsilon > 0$.

Step (1): Starting from y^0 proceed in the direction $(-y^0)$ until a boundary point y^k of the feasible region L_I is reached. Compute,

$$\rho^k = \max\{\rho_i | 0 < \rho_i = \frac{c_i}{(b_i)'y^0} < 1;\ i \varepsilon I\} \tag{5.7}$$

and

$$y^k = \rho^k y^0 \ . \tag{5.8}$$

Also define the set V_k,

$$V_k = \{i | (b_i)'y^k = c_i;\ i \varepsilon I\} \subset I \ , \tag{5.9}$$

and the kth cutting plane at y^k,

$$(y^k)'y = |y^k|_2^2 \ . \tag{5.10}$$

Step (2): Using the procedure(s) from Section 5.3 determine a direction d^k into the interior of the original feasible region, along the kth cutting plane. Or in other words, determine d^k such that,

$$(y^k)'d^k = 0 \tag{5.11}$$

and

$$(b_i)'d^k < 0,\ i \varepsilon V_k \ . \tag{5.12}$$

If either

$$\left| 1 - \frac{|y^{k-1}|_2^2}{|y^k|_2^2} \right| < \varepsilon \ , \tag{5.13}$$

for some tolerance factor $\varepsilon > 0$, or no such direction d^k satisfying equation (5.11) and equation (5.12) exists, then set

$$y^* = y^k \tag{5.14}$$

and stop; otherwise, continue to Step (3).

Step (3): Add the kth cutting plane from equation (5.10) as the (m+k)th constraint:

$$(y^k)'y \leq |y^k|_2^2 \tag{5.15}$$

to the constraints in L_I. This is done by including (m+k) in set I and setting,

$$C_{m+k} = |y^k|_2^2 \; , \tag{5.16}$$

and

$$b_{m+k} = y^k \; . \tag{5.17}$$

Step (4): If k = 1 then go to Step (5); otherwise, continue. Using the following procedure drop those previously established cutting planes which may now have become redundant. For any $j > 0$ and $j \neq k$ exclude (m+j) from set I, if

$$(m+j) \varepsilon I \tag{5.18}$$

and

$$(b_{m+k})'y^j > C_{m+k} \; . \tag{5.19}$$

Step (5): Move in the direction of d^k (obtained from Step (2)) until a constraint boundary is reached. Compute

$$z^k = y^k + \delta^k(\lambda^k d^k) \; , \tag{5.20}$$

where

$$\lambda^k = \min\{\lambda_i \,|\, 0 < \lambda_i = -\frac{(b_i)'y^k - C_i}{(b_i)'d^k} < \infty;\ i\varepsilon\{I - V_k - (m+k)\}\} \,. \quad (5.21)$$

If either

$$\lambda_i = -\frac{(b_i)'y^k - C_i}{(b_i)'d^k} < 0 \text{ for all } i\varepsilon\{I - V_k - (m+k)\} \quad (5.22)$$

or the set

$$\{I - V_k - (m+k)\} = \{\emptyset\} \,, \quad (5.23)$$

then an arbitrary value of λ^k (say $\lambda^k = 1$) can be selected. Now set k equal to k+1, y^0 equal to z^k and choose δ^k such that

$$0 < \delta^k < 1 \,.$$

Return to Step (1).

The above algorithm does not require vector normalizations. However, it is recommended since this will help in reducing the magnitude of the computational error. The value of δ^k can be selected arbitrarily, $\delta^k = 0.5$, for $k = 1, 2, \ldots$, is usually appropriate.

Except for Step (2), where we need to prove the optimality criterion used and establish a procedure for computing an interior feasible direction d^k, the algorithm is completely defined. The following section provides the necessary material for the completion of Step (2) of the algorithm.

5.3 The Feasible Direction

A feasible direction d^k at y^k along the kth cutting plane can be found by a procedure defined by Zountendijk [124]. Zountendijk's procedure requires that

$$(y^k)'d^k = 0 \qquad (5.24)$$

$$(b_i)'d^k \le 0,\ i \in V_k\ . \qquad (5.25)$$

Or equivalently, that

$$(y^k)'d^k = 0 \qquad (5.26)$$

$$B_k d^k \le 0\ , \qquad (5.27)$$

where

$$B_k = \begin{bmatrix} (b_1)' \\ (b_2)' \\ \vdots \\ (b_i)' \\ \vdots \end{bmatrix} ,\ i \in V_k\ . \qquad (5.28)$$

However, an <u>interior</u> feasible direction d^k at y^k along the kth cutting plane is defined by

$$(y^k)'d^k = 0 \qquad (5.29)$$

$$B_k d^k < 0\ . \qquad (5.30)$$

In Step (2) of the algorithm an interior feasible direction satisfying equations (5.29) and (5.30) is required and if no such direction exists then y^k is the optimal solution to the least distance programming problem. This claim is proved in Theorem 5.1.

<u>Theorem 5.1</u>

If there exists no interior feasible direction d^k at y^k such that

$$(y^k)'d^k = 0 \qquad (5.31)$$

$$(b_i)'d^k < 0\ ,\ i \in V_k \qquad (5.32)$$

then

$$y^* = y_k \tag{5.33}$$

is the optimal solution to the least distance programming problem defined in equation (5.1).

Proof: The theorem can be established by proving: If

$$y^* = y_k \tag{5.34}$$

then the vector d^k satisfying one of the following two equations violates the other;

$$(y^k)'d^k = 0 \ , \tag{5.35}$$

$$(b_i)'d^k < 0 \ , \ i \varepsilon V_k \ . \tag{5.36}$$

From the Kuhn-Tucker optimality conditions [75, p. 68] the $y^k \varepsilon L_I$ is an optimal solution to the least distance programming problem if and only if

$$-y_k = \sum_{i \varepsilon V_k} \alpha_i b_i \tag{5.37}$$

and

$$\alpha_i \geq 0, \ i \varepsilon V_k \ . \tag{5.38}$$

Now from equation (5.4) the optimal solution $y^* \neq 0$. Thus if y^k is an optimal solution, then $y^k \neq 0$. Hence in equation (5.37) and equation (5.38) we must have

$$\alpha_i > 0 \ , \text{ for at least one } i \varepsilon V_k \ . \tag{5.39}$$

Let the index set V_k' be defined such that

$$V_k' = \{i | \alpha_i > 0, \ i \varepsilon V_k\} \subset V_k \ . \tag{5.40}$$

Using equation (5.40) equation (5.37) can be rewritten as

$$-y_k = \sum_{i \varepsilon V_k'} \alpha_i b_i \ . \tag{5.41}$$

Multiplying both sides of equation (5.41) by some vector d^k we obtain

$$-(y^k)'d^k = \sum_{i\varepsilon V_k'} \alpha_i(b_i)'d^k \tag{5.42}$$

From the above equation observe that if d^k satisfies equation (5.35) then we have

$$0 = \sum_{i\varepsilon V_k'} \alpha_i(b_i)'d^k ,$$

which implies

$$0 = (b_i)'d^k \text{ , for all } i\varepsilon V_k' ,$$

since $\alpha_i > 0$ for all $i\varepsilon V_k' \subset V_k$. Thus d^k violates equation (5.36). Conversely, if d^k satisfies equation (5.36) then from equation (5.42) we have

$$-(y^k)'d^k = \sum_{i\varepsilon V_k'} \alpha_i(b_i)'d^k < 0 ,$$

since $\alpha_i > 0$ for all $i\varepsilon V_k' \subset V_k$. Thus d^k violates equation (5.35).

This completes the proof of Theorem 5.1.

The above theorem validates the stopping criterion used in Step (2) of the algorithm. The only phase now left in the completion of Step (2) of the algorithm is to establish a procedure for computing the direction d^k at y^k such that equation (5.29) and equation (5.30) are satisfied. In the following paragraphs the computational procedures to determine the direction d^k are presented.

Several existing procedures can be used in computing the feasible direction d^k at y^k satisfying equation (5.29) and equation (5.30). Linear programming, Ho and Kashyap algorithm [64,65] and the methods discussed in Zountendijk [124] are examples of these procedures. However, these procedures require considerable computational effort. Since the cutting plane algorithm requires an iterative use of these procedures, a short cut algorithm "e-method" for computing d^k is proposed. The e-method is developed so as to require a minimum computational effort. In some cases the e-method may not yield the required direction d^k. If this occurs then one must resort to one of the several existing procedures. Amongst these existing procedures the Ho and Kashyap algorithm appears to be more suitable since equation (5.30) involves strict inequalities and the Ho and Kashyap algorithm is designed for the problems with strict inequalities. The e-method is presented and is followed by an explanation of the Ho and Kashyap algorithm. The notations used here are the same as in Section 5-2.

<u>e-Method</u>:

In order to determine an interior feasible direction d^k at y^k, consider any known point e such that it is feasible with respect to the kth cutting hyperplane, that is,

$$(y^k)'e \leq C_{m+k} = (y^k)'y^k \; . \tag{5.43}$$

Now since y^0 is infeasible with respect to the kth cutting hyperplane, that is,

$$(y^k)'y^0 > C_{m+k} = (y^k)'y^k \; , \tag{5.44}$$

there must exist a point u on the cutting hyperplane such that the line joining y^0 and e intersects the cutting plane at u. The point u can be written as

$$u = y^0 + \lambda(e-y^0) \ , \tag{5.45}$$

where

$$0 < \lambda = \frac{(y^k)'y^0-(y^k)'y^k}{(y^k)'(y^0-e)} = \frac{(y^k)'(y^0-y^k)}{(y^k)'(y^0-e)} \leq 1 \ . \tag{5.46}$$

Using equation (5.8) equation (5.46) can be rewritten as

$$0 < \lambda = \frac{(1-\rho^k)(y^k)'y^k}{(y^k)'(y^k-\rho^k e)} \leq 1 \ . \tag{5.47}$$

Now for some vector d along the cutting plane

$$u = y^k + d \ , \tag{5.48}$$

since y^k and u both lie on the cutting plane. From equation (5.45) and equation (5.48) the vector d can be written as

$$d = y^0 - y^k + \lambda(e-y^0) \ . \tag{5.49}$$

Using equation (5.8) and equation (5.47) we have

$$d = \lambda[\frac{(1-\rho^k)y^k}{\lambda\rho^k} + e - \frac{y^k}{\rho^k}] \ ,$$

or

$$d = \lambda[e - \frac{y^k}{\rho^k} + \frac{y^k(y^k)'(y^k-\rho^k e)}{\rho^k(y^k)'y^k}] \ ,$$

and finally,

$$d = \lambda[e - \frac{y^k(y^k)'e}{(y^k)'y^k}] \ . \tag{5.50}$$

The direction d in equation (5.50) is obtained from equation (5.48) and satisfies equation (5.29). In order to be an interior feasible direction, d must also satisfy equation (5.30), that is,

$$B_k d < 0 ,$$

or, by using equation (5.50) we reduce it to

$$\lambda\ B_k[I - \frac{y^k(y^k)'}{(y^k)'y^k}]\ e < 0 . \tag{5.51}$$

Since $\lambda > 0$, the above equation can be rewritten as

$$B_k[I - \frac{y^k(y^k)'}{(y^k)'y^k}]\ e < 0 . \tag{5.52}$$

Furthermore, substituting for λ from equation (5.47) in equation (5.50) the vector d is obtained as

$$d = [\frac{(1-\rho^k)(y^k)'y^k}{(y^k)'(y^k-\rho^k e)}]\ [e - \frac{y^k(y^k)'\rho}{(y^k)'y^k}] . \tag{5.53}$$

Equations (5.52) and (5.53) define the requirements for determining the direction d^k. The first step in the procedure is to find an arbitrary point e such that equation (5.52) is satisfied. The point e satisfying equation (5.52) is then substituted into equation (5.53) to give the required direction d ($=d^k$). Under some conditions the vector e is readily found. In general, a simple trial and error procedure can be used to identify the vector e. Both of these procedures are outlined below.

If $\#(V_k) = 1$, say $V_k = \{i\}$, then the task of finding vector e is greatly simplified. In this case $e = -b_i$ and this vector always satisfies equation (5.52). This is true because the matrix

$$[I - \frac{y^k(y^k)'}{(y^k)'y^k}] = M_k \tag{5.54}$$

in equation (5.52) is an elementary reflector [107] and hence a positive-semidefinite matrix. Therefore we select

$$e = -b_i \ , \tag{5.55}$$

which satisfies equation (5.52) in slightly weaker form, that is,

$$(b_i)'(M_k)(-b_i) \leq 0 \ . \tag{5.56}$$

In general, however, a finite number of vectors e are generated according to the following sequence:

$$\begin{array}{l} (\eta,\ 0,\ 0,\ \ldots,\ 0,\ 0) \\ (-\eta,\ 0,\ 0,\ \ldots,\ 0,\ 0) \\ (0,\ \eta,\ 0,\ \ldots,\ 0,\ 0) \\ (0,\ -\eta,\ 0,\ \ldots,\ 0,\ 0) \\ \qquad\quad \vdots \\ (0,\ 0,\ 0,\ \ldots,\ 0,\ \eta) \\ (0,\ 0,\ 0,\ \ldots,\ 0,\ -\eta)\ , \end{array} \tag{5.57}$$

where

$$\eta > 0 \ .$$

These vectors are sequentially tested to see if equation (5.52) is satisfied. The first vector satisfying (5.52) is chosen as e.

Once a vector e satisfying equation (5.43) and equation (5.52) is found, then the required direction d^k is obtained from equation (5.53). If none of the generated e-vectors satisfy equation (5.52) then the e-method is abandoned and the technique of Ho and Kashyap is used. The e-method is summarized in the next paragraph. It is executed within Step (2) of the cutting plane algorithm.

e-Algorithm:

(1) If $\#(V_k) = 1$, say $V_k = \{i\}$, then $e = -b_i$ satisfies equation (5.52), go to (1a); otherwise go to (2) below.

(a) If e violates equation (5.43), then select σ such that $0 < \sigma < 1$, set $e = \sigma e$ and return to (1a); otherwise continue to (1b).

(b) Compute d^k using equation (5.53).

(c) If $d^k = 0$, then set $y^* = y^k$ and stop; otherwise return to Step (2) of the cutting plane algorithm.

(2) Let
$$e_{sj} = (0, 0, \ldots, 0, \underset{\substack{\uparrow \\ j\text{th element}}}{s\eta}, 0, \ldots, 0) \tag{5.58}$$

$$\eta > 0 \tag{5.59}$$

and
$$s = \begin{cases} +1 \\ \text{or} \\ -1 \end{cases} . \tag{5.60}$$

Set η, $s = +1$ and $j = 1$.

(a) If e violates equation (5.43) then select σ such that $0 < \sigma < 1$, set $e = \sigma e$ and return to (2a); otherwise continue to (2b).

(b) If e_{sj} satisfies equation (5.52), then substitute $e = e_{sj}$ in equation (5.53) and obtain d^k, now return to Step (2) of the cutting plane algorithm; otherwise continue.

(c) If $j = n$, then "use Ho and Kashyap procedure"; otherwise set $j = j + (\frac{1+(-1)s}{2})$, $s = (-1)s$, and return to (2a).

Ho and Kashyap Algorithm:

The algorithm developed by Ho and Kashyap [65] is exponentially convergent and finite. This algorithm is used here to determine vector e such that equation (5.52) is satisfied. Then equation (5.53) is used to determine the direction d^k. Equation (5.52) is simplified by substitutiting

$$A = -B_k \left[I - \frac{y^k(y^k)'}{(y^k)'y^k}\right] , \tag{5.61}$$

and rewritten as

$$Ae > 0 . \tag{5.62}$$

Let

$$f > 0 \tag{5.63}$$

and

$$g = Ae - f . \tag{5.64}$$

The Ho and Kashyap algorithm determines a vector e, if one exists, such that equation (5.62) is satisfied. The following two steps define the iterations in the algorithm:

(1) $e_{i+1} = e_i + \rho_i \; SA'|g_i| \; , \; e_0$ arbitrary , (5.65)

(2) $f_{i+1} = f_i + (g_i+|g_i|) \; , \; f_0 > 0$ but arbitrary , (5.66)

where

$$S = (2|A'A|_2 I - A'A) \Big/ |A'A|_2^{\,2} , \tag{5.67}$$

and

$$0 < \rho_i = \frac{|g_i|'ASA'|g_i|}{|g_i|'ASA'ASA|g_i|} < 2 . \tag{5.68}$$

The spectral norm $|A'A|_2$ can be easily computed by the power method [46,54].

The occurrence of a nonpositive vector g_i at any stage terminates the algorithm and indicates the inconsistency in equation (5.62).

Thus if a vector e satisfying equation (5.62) is found by the use of the Ho and Kashyap algorithm then equation (5.53) is used to compute the direction d^k; otherwise the algorithm terminates prematurely indicating a non-existance of the direction d^k. In either case this completes the computations under Step (2) of the cutting plane algorithm.

5.4 Convergence

The cutting plane algorithm presented in Section 5.2 is convergent. At each iteration a new cutting plane at y^k is constructed, with respect to which there is at least one point y^0 of the feasible region lying on the infeasible side of the cutting plane. This point also remains infeasible with respect to all future cutting planes. Thus the above elimination process in limit will converge to a feasible region containing only the optimal solution.

The cutting plane algorithm does not cummulate computational error, since the calculations in each iteration are independent to those in other iterations.

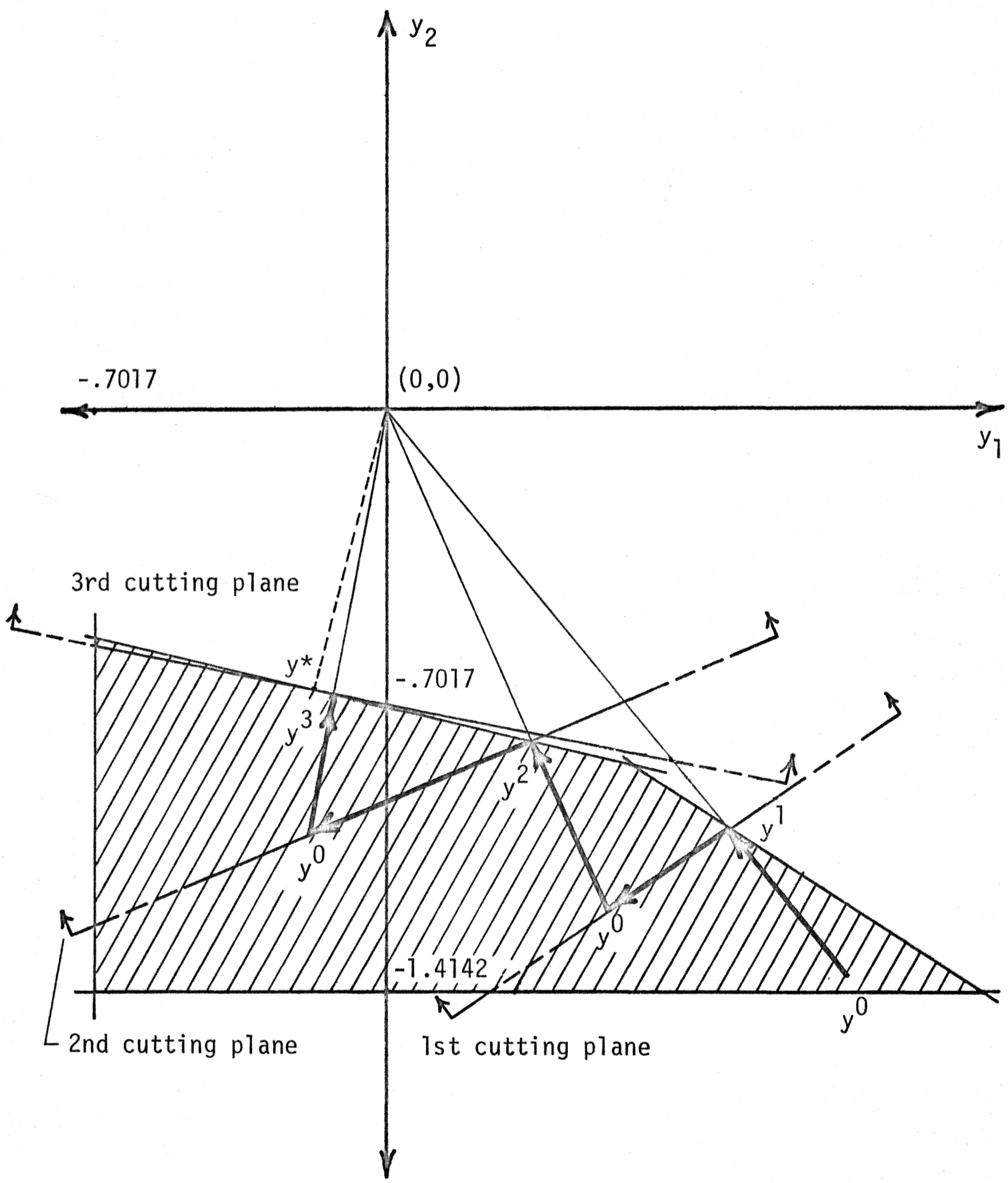

Figure 2

Iterations in the Solution of Example 1

5.5 Example

5.5.1 First Example

This problem is a continuation of the example in Section 2.5, where we reduced Kunzi's [75] quadratic programming problem to the following least distance programming problem:

$$\min[|y|_2^2 - 2.5] \tag{5.69}$$

subject to

$$\begin{bmatrix} 2\sqrt{2} & 3\sqrt{2} \\ \sqrt{2} & 4\sqrt{2} \\ -\sqrt{2} & 0 \\ 0 & -\sqrt{2} \end{bmatrix} [y] \le \begin{bmatrix} -2 \\ -4 \\ 1 \\ 2 \end{bmatrix} , \tag{5.70}$$

or equivalently,

$$\min[|y\ |_2^2 - 2.5] \tag{5.69}$$

subject to

$$\begin{bmatrix} 2 & 3 \\ 1 & 4 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} [y] \le \begin{bmatrix} -1.4142 \\ -2.6284 \\ 0.7071 \\ 1,4142 \end{bmatrix} . \tag{5.70}$$

The following solution includes the first three iterations of the cutting plane algorithm (Figure 2).

Iteration 1:

(0) $y^0 = (1.1,-1.4)'$.

$k = 1$

$\delta = 0.5$

$m = 4$

$\varepsilon = 0.0001$

(1) $\rho_1 = 0.7071$, $\rho_2 = 0.6285$, $\rho_3 = -0.6428$, $\rho_4 = 1.0101$,

$\rho^1 = \max\{0.7071, 0.6285\} = 0.7071$.

Therefore, $y^1 = 0.7071\ (1.1,-1.4)' = (0.7778,-0.9899)'$,

$V_1 = \{1\}$,

and the 1st cutting plane is $(y^1)'y = |y^1|_2^2 = 1.5849$.

(2) Using Step (1) of the e-algorithm:

$e = -(2,3)' = (-2,-3)'$,

$(y^1)'e = 1.4141 < 1.5849$,

thus $d^1 = (-2.1370, -1.6797)'$.

The point y^1 is non-optimal.

(3) Add 1st constraint: $(0.7778,-0.9899)y \leq 1.584$.

$b_5' = (0.7778,-0.9899)$

$c_5 = 1.5849$

$I = I + \{5\} = \{1,2,3,4,5\}$

(4) Not executed.

(5) $\{I-V_k-(m+k)\} = \{2,3,4\}$,

$\lambda_2 = -0.0399$, $\lambda_3 = 0.6948$, $\lambda_4 = 0.2526$.

Therefore $\lambda^1 = \min\{0.6948, 0.2526\} = 0.2526$,

and $z^1 = y^1 + (0.5)(0.2526)d^1$

$= (0.5079,-1.2020)'$.

$k = 2,$

$\delta^2 = 0.5,$

$y^0 = (0.5079,-1.2020)'.$

Iteration 2:

(1) $\rho_1 = 0.5460,\ \rho_2 = 0.6577,\ \rho_3 < 0,\ \rho_4 > 1.$

Thus $\rho^2 = \max\{0.5460,0.6577\} = 0.6577,$

$y^2 = 0.6577(0.5079,-1.2020)' = (0.3340,-0.7906)',$

$V_2 = \{2\},$

and the 2nd cutting plane is $(y^2)'y = |y^2|_2^{\,2} = 0.7366.$

(2) Using Step (1) of the e-algorithm:

$e = -(1,4)' = (-1,-4)',$

$(y^2)'e = 2.8284 \not< 0.7366.$

Let $\sigma = 1/4,\ e = (-0.25,-1),$

$(y^2)'e = 0.7071 < 0.7366.$

Thus $d^2 = (-0.48973,-0.20688).$

The point y^2 is non-optimal.

(3) Add 2nd constraint $(0.3340,-0.7906)y \leq 0.7366.$

$b_6' = (0.3340,-0.7906),$

$c_6 = 0.7366$

$I = I + \{6\} = \{1,2,3,4,5,6\}.$

(4) $(y^2)'y^1 = (0.3340,-0.7906)(0.7778,-0.9899)'$

$= 1.0424 > 0.7366 = c_6.$

Drop the first cutting plane (or the 5th constraint),

$I = I - \{5\} = \{1,2,3,4,6\}.$

(5) $\{I-V_k-(m+k)\} = \{1,3,4\}$,

$\lambda_1 < 0$, $\lambda_3 = 2.1259$, $\lambda_4 = 3.0143$.

Therefore $\lambda^2 = \min\{2.1259, 3.0143\} = 2.1259$,

$z^2 = y^2 + (0.5)(2.1259)d^2$

$= (-0.1865, -1.0105)'$,

$k = 3$,

$\delta^3 = 0.5$,

$y^0 = z^2 = (-0.1865, -1.0105)'$.

Iteration 3:

(1) $\rho_1 = 0.4145$, $\rho_2 = 0.6689$, $\rho_3 > 1$, $\rho_4 > 1$.

Thus $\rho^3 = \max\{0.4154, 0.6689\} = 0.6689$,

$y^3 = 0.6689(-0.1865, -1.0105)' = (-0.1247, -0.6759)'$,

$V_3 = \{2\}$

and the third cutting plane is $(y^3)'y = |y^3|_2^2 = 0.4724$.

(2) Using Step (1) of the e-algorithm:

$e = -(1,4)' = (-1,-4)'$,

$(y^3)'e = 2.8283 \not\leq 0.4724$.

Let $\sigma = 0.1$, $e = (-0.1, -0.4)$,

$(y^3)'e = 0.2828 < 0.4724$.

Now direction d^3 can be computed and the algorithm may be continued further. The point y^3 is non-optimal.

(3) Add 3rd constraint $(-0.1247, -0.6759)y \leq 0.4724$.

$b_7' = (-0.1247, -0.6759)$,

$c_7 = 0.4724$,

$I = I + \{7\} = \{1,2,3,4,6,7\}$.

(4) $(y^3)'y^2 = (-0.1247,-0.6759)(0.3340,-0.7906)'$

$= 0.4927 > 0.4724 = C_7.$

Drop the second cutting plane (or the 6th constraint),

$I = I - \{6\} = \{1,2,3,4,7\}.$

(5) This may be continued further.

The result after three iterations is

$$y^3 = (-0.1247,-0.6759),$$

as compared to the optimal solution

$$y^* = (-0.16638,-0.6655).$$

Now using y^3 and the inverse transformation the approximate solution to Kunzi's original quadratic programming problem, x^3, is given by

$$x^3 = \tilde{T}^{-1}(y^3) + x^*_u \ ,$$

that is,

$$x^3 = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix} y^3 + \begin{bmatrix} 1 \\ 2 \end{bmatrix} .$$

Substituting for y^3 in the above equation yields,

$$x^3 = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} -0.1247 \\ -0.6759 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} ,$$

or, equivalently

$$x^3 = \begin{bmatrix} 0.8234 \\ 1.0441 \end{bmatrix} ,$$

as compared to the optimal solution obtained by Kunzi [75],

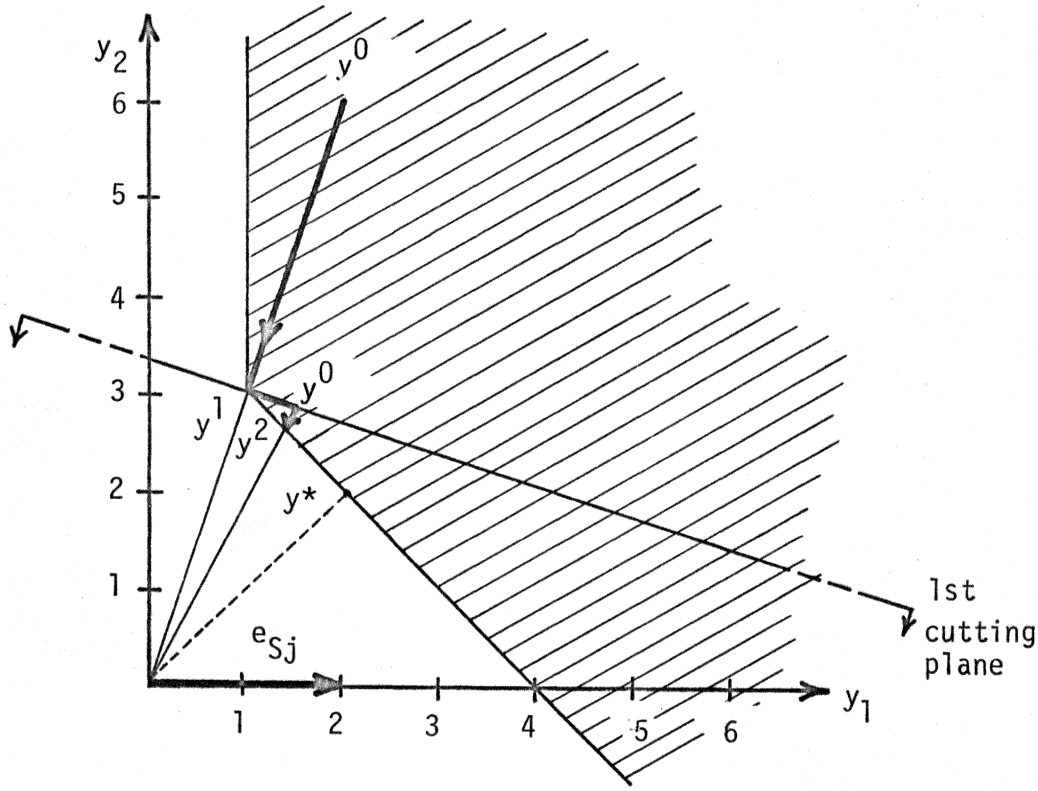

Figure 3

Iterations in the Solution of Example 2

$$x^* = \begin{bmatrix} 0.7647 \\ 1.0588 \end{bmatrix} .$$

The above example shows that in three iterations a solution very close to the optimum is obtained. Since this example considers only one active constraint at each iteration, the following example is designed to illustrate the use of the e-algorithm in finding an interior feasible direction when there are more than one active constraints involved.

5.5.2 Second Example

Consider the following least distance programming problem:

$$\min[|y|_2^{\,2}]$$

subject to

$$\begin{bmatrix} -1 & 0 \\ -1 & -1 \end{bmatrix} [y] \leq \begin{bmatrix} -1 \\ -4 \end{bmatrix} .$$

The cutting plane algorithm is used to solve this problem (Figure 3).

Iteration 1:

(0) $y^0 = (2,6)'$

$k = 1$

$\delta = 0.5$

$m = 2$

$\varepsilon = 0.0001$

(1) $\rho_1 = 0.5$, $\rho_2 = 0.5$,

$\rho^1 = \max\{0.5, 0.5\} = 0.5$.

Therefore $y^1 = 0.5(2,6)' = (1,3)'$.

$V_1 = \{1,2\}$

and the first cutting plane is $(y^1)'y = |y^1|_2^2 = 10$.

(2) Using step (2) of the e-algorithm:

$\eta = 2$, $j = 1$, $s = +1$,

$e_{sj} = (2,0)$,

e_{sj} satisfies equation (5.52).

Therefore $d_k = (1,-1/3)'$.

The point y' is found to be non-optimal.

(3) Add 1st constraint: $(1,3)y \leq 10$.

$b_3' = (1,3)$,

$c_3 = 10$,

$I = I + \{3\} = \{1,2,3\}$.

(4) Not executed.

(5) $\{I - V_k - (m+k)\} = \{\emptyset\}$.

Therefore $\lambda^1 = 1$ (arbitrary),

and $z^1 = y^1 + (0.5)(1)d^1$

$= (3/2, 17/6)'$.

$k = 2$,

$\delta^2 = 0.5$,

$y^0 = z^1 = (3/2, 17/6)'$.

The procedure may now be continued further. However, this example is designed to demonstrate the use of the e-algorithm in finding an interior feasible direction when there are more than one active constraints involved.

Chapter VI

CONCLUSION

6.1 Summary

Quadratic minimization problems constitute a significant and important subset of optimization problems. Research has shown that a convex quadratic programming problem can be reduced to a least distance programming problem using a unique and non-singular "linear least distance transformation." A computational algorithm for determining this linear least distance transformation is presented. This algorithm uses a variation of the Cholesky factorization technique for positive definite matrices, since Cholesky factorization is known to be numerically stable.

In solving the least distance programming problem it is found that certain special cases, dealing with the orthogonality property of the constraint matrix, lend themselves to a simple solution procedure. Theorems establishing the optimality of such solutions are presented. It is also found that solutions for least distance programming problems having a row-orthogonal constraint matrix are easily obtained. A generalization of this result for solving a special case of quadratic programming is also presented. A QR-transformation is found to be of no use in reducing a general least distance programming problem to a row-orthogonal least distance programming problem. It is found that the column orthogonality property of the constraint matrix of a least distance programming problem does not contribute to the solution of the problem. These special cases, though useful in their own right, have limited application to the solution of a general least distance programming problem.

Using generalized inverses, a new "generalized dual" of the least distance programming problem is derived. The dual problem is solvable by existing quadratic programming techniques. In addition, it is shown that the generalized dual problem can be solved by Ben-Israel's modified Newton method. A special case of the generalized dual problem is considered. It is found that the least distance programming problem, having a generalized inverse of the constraint matrix as a partial isometry, is easily solved. Since it is difficult to identify this latter class of matrices, the results have rather limited applications.

Finally, a convergent iterative algorithm for solving the least distance programming problem is presented. This algorithm has wide applications. It is based on a cutting plane technique, and requires the computation of an interior feasible direction at each iteration. The method of Ho and Kashyap may be used to establish this direction. The "e-method" proposed here gives an alternative for computing the feasible directions and generally requires much less computation. A few numerical examples are presented to illustrate these algorithms.

6.2 Suggestions for Further Study

The following three suggestions for further study are the logical consequences of this research.

(1) With reference to Section 4.4.2 and the work of Halmos [56] and Halmos and McLaughlin [58] we make the following observations. If a partial isometry B is not zero then its spectral norm $|A|_2 = 1$. Since by definition any matrix B is a contraction if $|B|_2 \leq 1$, it follows that a partial isometry is a contraction. Furthermore, Halmos and McLaughlin [58] and Erdelyi [37] show that every contraction on E^n can be extended to a partial isometry on E^{2n}. In view of this and Theorem 4.3 it may be possible to consider a solution technique in the E^{2n} space for the generalized dual problem presented in Chapter 4.

(2) The following conjecture may be worthy of consideration in solving the least distance programming problem given by

$$\min_{y \varepsilon L_y} [|y|_2{}^2] \tag{6.1}$$

where

$$L_y = \{y | By \leq c,\ c \nleq 0\} . \tag{6.2}$$

<u>Conjecture</u>: Let y^* be the optimal solution to the above problem. Since $y^* \neq 0$ is unique and L_y is a convex polyhedron in E^n, there are supporting hyperplanes to set L_y at the point $y^* \varepsilon L_y$. Let these hyperplanes be denoted by set H^*. Note that y^* is always situated on the boundary of L_y. Now a normal η^h drawn from the origin to a hyperplane $h \varepsilon H \subseteq H^*$ will penetrate the feasible region L_y. Let $L^*_y = \bigcup_{\forall h \varepsilon H} \{ \eta^h \cap L_y \} \subseteq L_y$. Consider the linear programming problem,

$$\min_{y \varepsilon L_y} (y,a) \tag{6.3}$$

where

$$a \varepsilon L^*_y \ . \tag{6.4}$$

If $y^* > 0$, then the solution to the above linear programming problem is also an optimum solution for the least distance programming problem in equations (6.1) and (6.2).

If the above conjecture can be proven then it may help in obtaining a suitable technique for solving a least distance programming problem. It may even be possible to consider an extension of the above conjecture to a linearly constrained convex nonlinear programming problem,where the optimal solution is not contained in the interior of the feasible region.

(3) The cutting plane algorithm presented in Chapter 5 along with the linear least distance transformation from Chapter 2 should be computationally compared with the other well known methods of quadratic programming presented in Kunzi [75].

BIBLIOGRAPHY

[1] Adachi, N. (1973), "On the Convergence of Variable-Metric Methods," Computing, 11, p. 111.

[2] Akaike, H. (1959), "On a Successive Transformation of Probability Distribution and Its Application to the Analysis of the Optimum Gradient Method," Am. Inst. Stat. Math., Tokyo, 11, p. 1.

[3] Amir-Moez, A. R. (1968), "Singular Values of Linear Transformations," Mathematical Series No. II, Texas Technological College, Lubbock, Texas.

[4] Arrow, K. J., et al. (1958), "Studies in Linear and Non-Linear Programming," Stanford University Press, California.

[5] Bartles, R. H. (1968), "A Numerical Investigation of the Simplex Method," Technical Report No. CS104, Computer Science Department, Stanford University, California.

[6] Bartles, R. H. (1971), "A Stablization of the Simplex Method," Numer. Math., 16, p. 414.

[7] Bartles, R. H., and G. H. Golub (1969), "The Simplex Method of Linear Programming Using LU Decomposition," Comm. Acm., 12, p. 266.

[8] Bartles, R. H., et al. (1970), "Numerical Techniques in Mathematical Programming," Technical Report No. 162, Computer Science Department, Stanford University, California.

[9] Beale, E. M. L. (1959), "On Quadratic Programming," Naval Research Logistics Quarterly, 6, p. 227.

[10] Beltrami, E. J. (1968), "A Comparison of Some Recent Iterative Methods for the Numerical Solution of Nonlinear Programs," paper presented at the Second International Conference on "Computing Methods in Optimization Problem," San Remo, Italy.

[11] Ben-Israel, A. (1964), "Notes on Linear Inequalities, I: The Intersection of the Nonnegative Orthant with Complementary Orthogenal Subspaces," J. Math. Anal. Aplcs., 9, p. 303.

[12] Ben-Israel, A. (1966), "A Newton-Ralphson Method for the Solution of Equations," J. Math. Anal. Apl., 15, p. 243.

[13] Ben-Israel, A. (1967), "On Iterative Methods for Solving Nonlinear Least Squares Problems Over Convex Sets," Israel J. Math., 5, p. 211.

[14] Ben-Israel, A. (1967), "On Newton's Method in Nonlinear Programming," Proceedings of Sixth International Symposium on Mathematical Programming, Princeton, New Jersey, p. 339.

[15] Ben-Israel, A., and A. Charnes (1963), "Contributions to the Theory of Generalized Inverses," J. SIAM, Vol. 11, No. 3, p. 667.

[16] Ben-Israel, A., and A. Charnes (1968), "An Explicit Solution of a Special Class of Linear Programming Problems," Opns. Res., 16, p. 1167.

[17] Bodewig, E. (1959), "Matrix Calculus," North-Holland Publishing Company, Amsterdam.

[18] Boot, J. C. G. (1964), "Quadratic Programming," North-Holland Publishing Company, Amsterdam.

[19] Boullion, T. L., and P. L. Odell (eds.) (1968), "Proceedings of the Symposium on Theory and Applications of Generalized Inverses of Matrices," Mathematics Series No. 4, Texas Technological College, Lubbock, Texas.

[20] Boullion, T. L., and P. L. Odell (1971), "Generalized Inverse Matrices," Wiley-Interscience, New York.

[21] Charnes, A., and W. W. Cooper (1961), "Management Models and Industrial Applications of Linear Programming," Vol. I and Vol. II, John Wiley and Sons, New York.

[22] Cheney, E. W. (1966), "Introduction to Approximation Theory," McGraw-Hill Book Company, New York.

[23] Cheney, W., and A. A. Goldstein (1959), "Proximity Maps for Convex Sets," Trans. Am. Math. Soc., p. 448.

[24] Cline, A. K. (1973), "An Elimination Method for the Solution of Linear Least Squares Problems," SIAM J. Numer. Anal., Vol. 10, No. 3, p. 283.

[25] Cottle, R. W., et al. (1970), "On Class of Copositive Matrices," Lin. Algebra Aplcs., 3, p. 295.

[26] Cottle, R. W., and W. C. Mylander (1970), "Ritter's Cutting Plane Method for Nonconvex Quadratic Programming," p. 257, in "Integer and Non-Linear Programming," ed. by J. Abadie, North-Holland, Amsterdam.

[27] Cottle, R. W., and J. A. Ferland (1972), "Matrix-Theoretic Criteria for the Quasi-Convexity and Psuedo-Convexity of Quadratic Functions," Lin. Algebra Aplcs., 5, p. 123.

[28] Daniel, J. W. (1973a), "On Perturbations in Systems of Linear Inequalities," SIAM J. Numer. Anal., Vol. 10, No. 2, p. 299.

[29] Daniel, J. W. (1973b), "Newton's Method for Nonlinear Inequalities," Numer. Math., 21, p. 381.

[30] Dantzig, G. B. (1963), "Linear Programming and Extensions," Princeton University Press, New Jersey.

[31] Dantzig, G. B., and A. F. Veinott (eds.),(1968), "Mathematics of the Decision Sciences, Part I," Lectures in Applied Mathematics, Vol. 11, American Mathematical Society, Rhode Island.

[32] Dorfman, R. (1951), "Application of Linear Programming to the Theory of the Firm," University of California Press, California.

[33] Dorfman, R., P. A. Samuelson, and R. M. Solow (1958), "Linear Programming and Economic Analysis," McGraw-Hill, New York.

[34] Dorn, W. S. (1960), "Duality in Quadratic Programming," Quart. Appl. Math., Vol. 18, p. 155.

[35] Dorn, W. S. (1963), "Nonlinear Programming - A Survey," Management Sci., Vol. 9, No. 2, p. 171.

[36] Erdelyi, E. (1966), "On Partial Isometries in Finite-Dimensional Euclidean Spaces," J. SIAM Appl. Math., Vol. 14, No. 3, p. 453.

[37] Erdelyi, I. (1968), "Partial Isometries Closed Under Multiplication on Hilbert Spaces," J. Math. Anal. Apl., 22, p. 546.

[38] Fadeev, D. K., and V. N. Fadeeva (1963), "Computational Methods of Linear Algebra," W. H. Freeman and Company, California.

[39] Fan, K. (1956), "On Systems of Linear Inequalities," Annals of Math. Studies, No. 38, p. 99.

[40] Fiacco, A. V., and G. P. McCormic (1968), "Nonlinear Programming: Sequential Unconstrained Minimization Techniques," John Wiley and Sons, New York.

[41] Fleming, W. H. (1965), "Functions of Several Variables," Addison-Wesley Publishing Company, Massachusetts.

[42] Franklin, J. N. (1968), "Matrix Theory," Prentice-Hall, New Jersey.

[43] Gale, D. (1960), "The Theory of Linear Economic Models," McGraw-Hill Book Company, New York.

[44] Gantmacher, F. R. (1959), "The Theory of Matrices," Chelsa Publishing Company, New York.

[45] Gass, S. I. (1970), "An Illustrated Guide to Linear Programming," McGraw-Hill, New York, pp. 153-168.

[46] Gastinel, N. (1970), "Linear Numerical Analysis," Hermann, Paris, France.

[47] Geoffrion, A. M. (ed.) (1972), "Perspectives on Optimization: A Collection of Expository Articles," Addison-Wesley, Massachusetts.

[48] Giannessi, F. (1972), "A Gradual Algorithm for the Solution of Non-Convex Quadratic Programming Problems," p. 61, in "Minimization Algorithms, Mathematical Theories and Computer Results," ed. by G. P. Szego, Academic Press, New York.

[49] Gill, P. E. and W. Murray (1972), "Quasi-Newton Methods for Unconstrained Optimization," <u>J. Inst. Math. Applcs.</u>, p. 91.

[50] Gill, P. E., et al. (1974), "Methods for Modifying Matrix Factorizations," <u>Math. Comp.</u>, Vol. 28, No. 126, p. 505.

[51] Goldfarb, D. (1969), "Extension of Davidon's Variable Metric Method to Maximization Under Linear Inequality and Equality Constraints," <u>SIAM J. Appl. Math.</u>, Vol. 17, No. 4, p. 739.

[52] Goldstein, A. A. (1967), "Constructive Real Analysis," Harper and Row, New York.

[53] Golub, G. H., and M. A. Saunders (1969), "Linear Least Squares and Quadratic Programming," Technical Report No. 134, Computer Science Department, Stanford University, California.

[54] Gourlay, A. R., and G. A. Watson (1973), "Computational Methods for Matrix Eigenproblems," John Wiley and Sons, London.

[55] Greenstadt, J. L. (1966), "A Ricocheting Gradient Method for Nonlinear Optimization," <u>SIAM Appl. Math.</u>, Vol. 14, No. 3, p. 429.

[56] Halmos, P. (1957), "Introduction to Hilbert Space," Chelsa Publishing Company, New York.

[57] Halmos, P. (1958), "Finite-Dimensional Vector Spaces," D. Van Nostrand Company, New Jersey.

[58] Halmos, P. R., and J. E. McLaughlin (1963), "Partial Isometries," <u>Pacific J. Math.</u>, 13, p. 585.

[59] Hancock, H. (1917), "Theory of Maxima and Minima," Dover Publications, New York.

[60] Hadley, G. (1961), "Linear Algebra," Addison-Wesley Publishing Company, Massachusetts.

[61] Hadley, G. (1962), "Linear Programming," Addison-Wesley Publishing Company, Massachusetts.

[62] Hadley, G. (1964), "Nonlinear and Dynamic Programming," Addison-Wesley Publishing Company, Massachusetts.

[63] Himmelblau, D. M. (1972), "Applied Nonlinear Programming," McGraw-Hill, New York.

[64] Ho, Y., and R. L. Kashyap (1965), "An Algorithm for Linear Inequalities and Its Applications," IEEE Trans. Comp., Vol. EC-14, No. 5, p. 683.

[65] Ho, Y., and R. L. Kashyap (1966), "A Class of Iterative Procedures for Linear Inequalities," J. SIAM Control, Vol. 4, No. 4, p. 112.

[66] Householder, A. S. (1964), "The Theory of Matrices in Numerical Analysis," Blaisdell Publishing Company, New York.

[67] Jacoby, S. L. S., et al. (1972), "Iterative Methods for Nonlinear Optimization Problems," Prentice-Hall, New Jersey.

[68] Kantorovich, L. V., and G. P. Akilov (1964), "Functional Analysis in Normed Spaces," The Macmillan Company, New York.

[69] Karlovitz, L. A. (1970), "Construction of Nearest Points in the L^p, p Even, and L^∞ Norms. I," J. Approx. Theory, 3, p. 123.

[70] Keller, E. L. (1973), "The General Quadratic Optimization Problem," Math. Prog., Vol. 5, p. 311.

[71] Koopmans, T. C. (ed.) (1951), "Activity Analysis of Production and Allocation," John Wiley and Sons, New York.

[72] Kowalik, J., and M. R. Osborne (1968), "Methods for Unconstrained Optimization Problems," American Elsevier Publishing Company, New York.

[73] Kreko, B. (1968), "Linear Programming," American Elsevier Publishing Company, New York.

[74] Kuhn, H. W., and A. W. Tucker (1951), "Nonlinear Programming," Proceedings of the Second Berkley Symposium on Mathematical Statistics and Probability, p. 481.

[75] Kunzi, H. P., and W. Krelle (1966), "Nonlinear Programming," Blaisdell Publishing Company, Massachusetts.

[76] Lancaster, P. (1969), "Theory of Matrices," Academic Press, New York.

[77] Laurent, P. J. (1973), "Exchange Algorithm in Convex Analysis," in "Approximation Theory," edited by G. G. Lorentz, Academic Press, New York.

[78] Levitin, E. S. and B. T. Polyak (1966), "Constrained Minimization Methods," Zh. Uychisl. Mat. mat. Fiz., 6, 5, p. 787.

[79] Lootsma, F. A. (ed.) (1973), "Numerical Methods for Non-linear Optimization," Academic Press, London.

[80] Luenberger, D. G. (1969), "Optimization by Vector Space Methods," John Wiley and Sons, New York.

[81] Luenberger, D. G. (1973), "Introduction to Linear and Nonlinear Programming," Addison-Wesley Publishing Company, Massachusetts.

[82] Markowitz, H. (1959), "Portfolio Selection: Efficient Diversification of Investment," John Wiley, New York.

[83] Martos, B. (1969), "Subdefinite Matrices and Quadratic Forms," SIAM J. Appl. Math., Vol. 17, No. 6, p. 1215.

[84] Martos, B. (1971), "Quadratic Programming with a Quasi-Convex Objective Function," Opns. Res., 19, p. 87.

[85] Mirsky, L. (1963), "An Introduction to Linear Algebra," Oxford University Press, London.

[86] Mitchell, B. F., et al. (1974), "Finding the Point of a Polyhedron Closest to the Origin," SIAM J. Control, Vol. 12, No. 1, p. 19.

[87] Muller-Merbach, H. (1970), "On Round-Off Errors in Linear Programming," Springer, Berlin.

[88] Mills, H. D. (1967), "Extending Newton's Method to Systems of Linear Inequalities," Proceedings of Sixth International Symposium on Mathematical Programming, Princeton, New Jersey, p. 353.

[89] Murray, W. (1971), "An Algorithm for Indefinite Quadratic Programming," National Physical Laboratory, DNAC Report No. 1. Teddington, England.

[90] Murray, W. (ed.) (1972), "Numerical Methods for Unconstrained Optimization," Academic Press, London.

[91] Paige, C. C. (1973), "An Error Analysis of a Method for Solving Matrix Equations," Math. Comp., Vol. 27, No. 122, p. 355.

[92] Parsons, T. D. (1966), "A Combinatorial Approach to Convex Quadratic Programming," Ph.D. Thesis, Princeton University, New Jersey.

[93] Powell, M. J. D. (1972), "Quadratic Termination Properties of Minimization Algorithms I: Statement and Discussion of Results," J. Inst. Maths. Applcs., 10, p. 333.

[94] Pshenichnyi, B. N. (1970), "Newton's Method for the Solution of Systems of Equalities and Inequalities," English translation: Math. Notes, 8, p. 827.

[95] Ramez, E. J. (1965), "On the Determination of Boundary Solutions of a System of Linear Inequalities and the Method of Equalizing Descents," Soviet Math., Vol. 6, No. 1, p. 310.

[96] Robers, P. D. and A. Ben-Israel (1970), "A Suboptimization Method for Interval Linear Programming: A New Method of Linear Programming," Lin. Algebra Applcs., 3, p. 383

[97] Rabinowitz, P. (1970), "Mathematical Programming and Approximation," in "Approximation Theory" ed. by A. Talbot, Academic Press, London, p. 217.

[98] Ritter, K. (1966), "A Method for Solving Maximum-Problem with a Nonconcave Quadratic Objective Function," Z. Wahrscheinlicj-keitztheorie, Vrew. Geb. 4, p. 340.

[99] Robinson, S. M. (1972), "Extension of Newton's Method to Nonlinear Functions with Values in a Cone," Numer. Math., 19, p. 341.

[100] Rockafellar, R. T. (1970), "Convex Analysis," Princeton University Press, Princeton, New Jersey.

[101] Rosen, J. B. (1960), "The Gradient Projection Method for Nonlinear Programming, Part I: Linear Constraints," J. SIAM, Vol. 8, No. 1, p. 181.

[102] Rosen, J. B. (1961), "The Gradient Projection Method for Nonlinear Programming, Part II: Nonlinear Constraints," J. SIAM, Vol. 9, No. 4, p. 514.

[103] Saunders, M. A. (1972), "Large-Scale Linear Programming Using the Cholesky Factorization," Technical Report No. 252, Computer Science Department, Stanford University.

[104] Schultz, H. K. (1973), "A Kuhn Tucker Algorithm," SIAM J. Control, Vol. 11, No. 3, p. 438.

[105] Schwarz, H. R. (1973), "Numerical Analysis of Symmetric Matrices," Prentice-Hall, New Jersey.

[106] Sposito, V. A. (1973), "Solutions of a Special Class of Linear Programming Problems," Opns. Res., 21, p. 386.

[107] Stewart, G. W. (1973), "Introduction to Matrix Computations," Academic Press, New York.

[108] Stoer, J. (1971), "On the Numerical Solutions of Constrained Least-Squares Problems," SIAM J. Numer. Anal., Vol. 8, No. 2, p. 382.

[109] Szego, G. P. (1972), "Minimization Algorithms, Mathematical Theories and Computer Results," Academic Press, New York.

[110] Taussky, O. (ed.) (1954), "Contribution to the Solution of Systems of Linear Equations and the Determination of Eigenvalues," National Bureau of Standards, Applied Mathematics Series, #39.

[111] Tucker, A. W. (1963), "Principal Pivot Transform of Square Matrices," SIAM Review, 5, p. 305 (Abstract).

[112] Tucker, A. W. (1967), "Least Distance Programming," Proceedings of Sixth International Symposium on Mathematical Programming, Princeton, New Jersey.

[113] Tucker, A. W. (1968), "A Least-Distance Approach to Quadratic Programming," in "Mathematics of the Decision Sciences, Part I" ed. by G. B. Dantzig and A. F. Veinott, Lectures in Applied Mathematics, Vol. 11, American Mathematical Society, Rhode Island.

[114] Varaiya, P. P. (1972), "Notes on Optimization," Van Nostrand Reinhold, New York.

[115] Wampler, R. H. (1969), "An Evaluation of Linear Least Squares Computer Programs," J. Res. Nat. Bur. Stan., Vol. 73B, No. 2, p. 59.

[116] Wilkinson, J. H. (1961), "Error Analysis of Direct Methods of Matrix Inversions," J. Assoc. Comp. Mach., p. 281.

[117] Wilkinson, J. H. (1963), "Rounding Errors in Algebraic Processes," Prentice-Hall, New Jersey.

[118] Wilkinson, J. H., and C. Reinsch (1971), "Handbook for Automatic Computation, Vol. II: Linear Algebra," Springer-Verlag, New York.

[119] Wolfe, P. (1959), "The Simplex Method for Quadratic Programming," Econometrica, 27, p. 382.

[120] Wolfe, P. (1967), "Methods of Nonlinear Programming," Ch. 6 in "Nonlinear Programming" ed. by J. Abadie, North-Holland, Amsterdam.

[121] Zangwill, W. I. (1969), "Nonlinear Programming: A Unified Approach," Prentice-Hall, New Jersey.

[122] Zionts, S. (1974), "Linear and Integer Programming," Prentice-Hall, New Jersey.

[123] Zlobec, S., and A. Ben-Israel (1970), "On Explicit Solutions of Interval Linear Programs," Israel J. Math., 8, p. 12.

[124] Zountendijk, G. (1960), "Methods of Feasible Directions," Elsevier Publishing Company, Amsterdam, Ch. 7.

[125] Zountendijk, G. (1966), "Nonlinear Programming: A Numerical Survey," SIAM J. Control, Vol. 4, No. 1, p. 194.

[126] Zukhovitskiy, S. I., and L. I. Avdeyeva (1966), "Linear and Convex Programming," W. B. Saunders Company, Philadelphia.

The vita has been removed from the scanned document

QUADRATIC MINIMIZATION

AND LEAST DISTANCE PROGRAMMING

by

Syed Z. Shariq

(ABSTRACT)

Quadratic minimization problems constitute a significant and important subset of optimization problems. Research has shown that a convex quadratic programming problem can be reduced to a least distance programming problem using a unique and non-singular "linear least distance transformation." A computational algorithm for determining this linear least distance transformation is presented. This algorithm uses a variation of the Cholesky factorization technique for positive definite matrices, since Cholesky factorization is known to be numerically stable.

In solving the least distance programming problem it is found that certain special cases, dealing with the orthogonality property of the constraint matrix, lend themselves to a simple solution procedure. Theorems establishing the optimality of such solutions are presented. It is also found that solutions for least distance programming problems having a row orthogonal constraint matrix are easily obtained. A generalization of this result for solving a special case of quadratic programming is also presented. A QR-transformation is found to be of no use in reducing a general least distance programming problem to a row-orthogonal least distance programming problem. It is found that the column orthogonality property of the constraint matrix of a least

distance programming problem does not contribute to the solution of the problem. These special cases, though useful in their own right, have limited application to the solution of a general least distance programming problem.

Using generalized inverses, a new "generalized dual" of the least distance programming problem is derived. The dual problem is solvable by existing quadratic programming techniques. In addition it is shown that the generalized dual problem can be solved by Ben-Israel's modified Newton method. A special case of the generalized dual problem is considered. It is found that the least distance programming problem, having a generalized inverse of the constraint matrix as a partial isometry, is easily solved. Since it is difficult to identify this latter class of matrices, the results have rather limited applications.

Finally, a convergent iterative algorithm for solving the least distance programming problem is presented. This algorithm has wide applications. It is based on a cutting plane technique, and requires the computation of an interior feasible direction at each iteration. The method of Ho and Kashyap may be used to establish this direction. The "e-method" proposed here gives an alternative for computing the feasible directions and generally requires much less computation. A few numerical examples are presented to illustrate these algorithms.