INTERIOR ALGORITHMS FOR LINEAR, QUADRATIC, AND LINEARLY CONSTRAINED CONVEX PROGRAMMING

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ENGINEERING-ECONOMIC SYSTEMS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By

Yinyu Ye

October 1987
INTERIOR ALGORITHMS FOR LINEAR, QUADRATIC, AND LINEARLY CONSTRAINED CONVEX PROGRAMMING

Yinyu Ye, Ph.D.
Stanford University, 1987

Abstract

This dissertation develops two interior algorithms, extensions of Karmarkar's projective and the "center" path-following linear programming algorithms, for solving more general groups of optimization problems: convex quadratic and linearly constrained convex programs. These algorithms are polynomial-time algorithms if the objective function is convex quadratic. More precisely, the number of iterations of the projective algorithm is bounded by $O(Ln)$, and each iteration can be computed in $O(n^4)$ or $O(Ln^3)$ arithmetic operations; the number of the path-following algorithm is bounded by $O(Ln^{0.5})$, and each iteration can be computed in $O(n^3)$ arithmetic operations; where $n$ is the number of variables and $L$ is the binary-encoding length of the input data. Both algorithms create a sequence of interior feasible points that converge to the optimal solution point.

This dissertation also discusses the relation between the ellipsoid method and these interior algorithms. The potential function, which is used to measure polynomial convergence of the primal solutions in the projective algorithm, correctly represents the logarithmic volume of a dual ellipsoid containing all the optimal dual solutions. Like in the ellipsoid method, the volume of this dual ellipsoid uniformly shrinks to zero as the potential function monotonically declines. This resemblance illustrates the insight equivalence and difference between the ellipsoid method and Karmarkar's algorithm, and leads to a strong eliminating theorem in determining the optimal basis for linear programming.

Based on these theoretical developments, this dissertation proposes practical approaches to efficiently implementing the interior algorithms—approaches that bypass the difficulties found in Karmarkar's original algorithm for linear programs, relax the small-step-size restriction in the path-following algorithm, and solve more general linearly constrained convex programs. In addition, the computational results and applications of the projective algorithm are reported.
Preface

Organization of This Dissertation

This dissertation is divided into seven chapters.

Chapter 0 overviews optimization problems, algorithms, and the criteria for judging the efficiency of optimization algorithms. In particular, I discuss the combinatorial nature and the major difficulties in inequality constrained optimization, and previous approaches to solving these optimization problems.

Chapter 1 introduces a general framework, the interior ellipsoid (IE) method, for solving linearly constrained convex programming (CP). The solution strategy, geometrical interpretation, and local convergence rate are explored. I show how the IE method overcomes the difficulties in CP via constructing an interior ellipsoid centered about the current solution point, followed by optimizing the objective function over the ellipsoid.

Chapter 2 applies the projective transformation (PT) to the IE method, and presents a new primal-dual method for linear programming (LP). The techniques of using duality and cutting objective are combined to obtain both the primal and dual optimal solutions simultaneously without solving them together. This method requires neither prior knowledge of the optimal objective value, nor explicit transformation of the standard LP form into Karmarkar’s canonical form. It maintains polynomial-time complexity and achieves practical efficiency. I also discuss the relation between the ellipsoid method and this primal-dual method at the end of this chapter.

Chapter 3 extends the PT and IE approaches (PTIE) to solving linearly constrained convex programming. First, the PTIE method, coupled with an objective augmentation technique, is discussed. In this method, the solution iteration is bounded by $O(Ln)$. If the objective function is convex quadratic, then the total solution time is $O(Ln^5)$ or $O(L^3n^4)$, depending on how one implements the algorithm. Later in this chapter, I discuss the primal-dual method for solving general CP problems to achieve practical efficiency.
Chapter 4 adds the p-projective transformations and the tangent-plane move procedure to the PTIE method. This addition results in an approximation method for solving each iteration with efficient performance in both theory and practice.

Chapter 5 applies the “center” path-following (PF) scheme to the IE method, and develops a primal-dual algorithm (PFIE) for convex quadratic programming (QP). This algorithm creates a sequence of primal and dual interior feasible points converging to the optimal solution. At each iteration, the complementary slackness value, i.e., the objective gap between the primal and dual is reduced at a global ratio \((1 - \frac{1}{4\sqrt{n}})\), and each iteration solves a system of linear equations in \(O(n^3)\) arithmetic operations. Therefore, the algorithm solves QP in total \(O(Ln^{3.5})\) arithmetic operations. I also discuss a line search technique to improve the algorithm’s practical efficiency.

Chapter 6 outlines the major contributions of this work and summarizes my final conclusions.

A program of the PTIE method has been coded in FORTRAN and applied to solve real problems. The computational results for this program are reported and displayed within this dissertation.

Acknowledgements

I would like to thank the many people who contributed to my success in this undertaking.

First, my sincere thanks are due my advisor, Professor Edison Tse, for his guidance and understanding in helping me focus my interests on algorithm development for mathematical programming and systems optimization, and for his support throughout my four years in the Engineering-Economic Systems department at Stanford.

This work could not have been done without Professor David Luenberger. It is he who brought me to the frontiers of optimization theory. I am very grateful for many discussions with him about this work, especially with regard to nonlinear programming.
I have also been very fortunate to benefit from the constant encouragement and patient advice of Professor George Dantzig, who spent innumerable hours teaching me the fundamentals of linear programming, both individually and through his courses and seminars. It has been an honor to be his student.

I am deeply indebted to Professor Samuel Chiu, a member of my dissertation committee, for his invaluable suggestions, support, and supervision on most aspects of this work. He spent much time reading and commenting on my papers.

I am extremely appreciative of Christiane Liu and Lynn Cannon for proofreading my dissertation during the whole process of my writing.

My sincere thanks are also due other professors—Michael Todd, Michael Saunders, Ronald Howard, Christos Papadimitrou and Ross Shachter—for their excellent teaching and/or help during this work. For many reasons I wish to give special thanks to my officemates and classmates—John Breese, Miguel Perez, Brion Feinberg, Joseph Tatman, Scott Strait, and Seok Hui Ng—from whom I have benefited much. I am also grateful to my American friends, homestay family, English partner, and writing tutor, who made life in the United States so pleasant and enjoyable for me and my family. In addition, I would like to acknowledge Nancy Florence, Susan Clement, Mako Bockholt, and Alma Gaylord for their invaluable help throughout my studies at Stanford.

Finally, I owe many thanks to my wife, Daisun, for her patience during the long hours alone, for her help in coding these algorithms, and for her constant physical and emotional support.
Notation

The point in Euclidean $n$-space ($R^n$) with components $(x_1, x_2, \ldots, x_n)$ is denoted by the column vector $x$, and the function $f(x_1, x_2, \ldots, x_n)$ is denoted by $f(x)$. $\text{diag}(x)$ always denotes the diagonal matrix of vector $x$. The gradient row-vector of $f(x)$ is designated as $\nabla f(x)$. $\ln(x)$ is the natural logarithmic function, and $\exp(x)$ is the exponential function.

Vectors are printed in lower case. Matrices are printed in upper case. Superscript $T$ denotes the transpose operation. $e$ is the vector of all one’s and 0 is the vector of all zero’s, with their dimensions matching the other vectors in expressions unless otherwise stated. Vector inequalities should be interpreted component by component.

The summation symbols $\sum_{i=1}^{n}$ and the multiplication symbols $\prod_{i=1}^{n}$ mean that the operations are executed from 1 to $n$ over index $i$. $\|\cdot\|$ is the $L_2$ norm in Euclidean space.

Linearly constrained convex programming is abbreviated as CP, linear programming as LP, and convex quadratic programming as QP.
# Table of Contents

**Chapter 0. Introduction**

0.1 Optimization Problems and Algorithms ............................................. 1
0.2 Previous Approaches to Solving CP ...................................................... 2
0.3 Primal and Dual Models of CP .............................................................. 6
0.4 What Makes CP Hard to Solve? ............................................................. 9

**Chapter 1. Interior Ellipsoid Method**

1.1 Introduction ................................................................. 12
1.2 Interior Ellipsoid: A Geometric Expression ............................................ 12
1.3 Weighted Gradient-Projection: An Algebraic Representation ...................... 14
1.4 Interior Ellipsoid (IE) Method for CP ............................................... 18
1.5 Interior Ellipsoid Method in CP General Form ....................................... 26
1.6 Summary ................................................................. 27

**Chapter 2. Projective Transformation and Interior Ellipsoid Method: Linear Programming**

2.1 Introduction ................................................................. 28
2.2 Potential Function and Projective Transformation ................................... 29
2.3 Cutting Objective Lower-Bound Technique .......................................... 34
2.4 Cutting Dual-Objective Value Technique:
   A New Primal-Dual Method for LP ....................................................... 41
2.5 Cutting Primal-Objective Value Technique:
   A Heuristic Approach ................................................................. 45
2.6 Computational Results ............................................................... 47
2.7 Relation to the Ellipsoid Method ....................................................... 50
2.8 Summary ................................................................. 56
Chapter 3. Projective Transformation and Interior Ellipsoid Method: Convex Quadratic Programming

3.1 Introduction .......................................................... 59
3.2 Convexity Invariance in Projective Transformation .................. 60
3.3 A Polynomial-Time Algorithm for QP ................................ 64
3.4 Cutting Dual-Objective Value Technique:
   The Primal-Dual Method for CP .................................... 73
3.5 Computational Results ................................................. 77
3.6 Summary ................................................................. 79

Chapter 4. P-Simplex: Improving PTIE Efficiency

4.1 Introduction .......................................................... 81
4.2 P-Simplex and P-Projective Transformation ......................... 81
4.3 Barrier-Simplex in PTIE ............................................. 87
4.4 Computational Results ................................................. 90
4.5 Summary ................................................................. 91

Chapter 5. Path-Following and Interior Ellipsoid Method: Convex Quadratic Programming

5.1 Introduction .......................................................... 92
5.2 “Center” and Barrier Function ...................................... 92
5.3 The “Center” Path-Following Algorithm for QP .................... 94
5.4 Setting Initial Solution Pair ........................................ 99
5.5 A Safeguard Line Search Technique ............................... 101
5.6 Summary ................................................................. 102

Chapter 6. Conclusions ..................................................... 103

References  ........................................................................ ix
Chapter 0 Introduction

0.1 Optimization Problems and Algorithms

Optimization has been an important objective of operations research and management science ever since this approach to decision-making was originally postulated. However, continual increases in the power of optimization have been possible only through the recent development of the high-speed digital computer. The goal of today's researchers in this field is to provide more efficient algorithms so that computer users can solve a wider range of optimization problems than ever before.

Optimization problems can generally be divided into two categories: combinatorial optimization problems and continuous optimization problems. Combinatorial optimization problems have discrete variables while continuous optimization problems have continuous variables. In continuous problems, we usually look for real numbers or even a function; in combinatorial problems, we look for an object from a discontinuous set—typically an integer, permutation, or graph. These two categories generally have quite different characteristics, and the algorithms for solving them are very divergent. Therefore, quadratic programming (QP) plays a unique role in optimization theory: in one sense, QP is a continuous optimization that includes linear programming (LP) and provides a fundamental approach for nonlinear programming, but it also may be considered one of the most challenging combinatorial optimization problems in computer science.

Most optimization algorithms are iterative in nature. Several criteria for judging the efficiency of optimization algorithms have been proposed in the past. One is the result of convergence analysis, explored in operations research and system science. Another criterion is the computational complexity studied by computer scientists over the past eighteen years. A third is, of course, practical efficiency in solving a large number of real problems.
Convergence analysis is well-rooted as a principle that both verifies whether or not a given algorithm will in fact generate a sequence that converges to the optimal solution point, and also evaluates how fast this convergence will be (Powell [54], Zangwill [69]). It has two aspects—one is global convergence analysis and the other is local convergence analysis. Global convergence analysis answers whether the sequence converges or not; local convergence analysis crudely measures the local (asymptotic) convergence rate and ratio. The local convergence rate and ratio have certain application limits since they are only valid for solving continuous optimization problems in local cases such as equality constrained optimization (ECO) when the points in the sequence are close to the optimal solution. It may take an exorbitant amount of time to bring the points close to the optimal solution.

Whereas convergence analysis evaluates the local speed of the convergence, computational complexity considers the global speed of the convergence. Edmonds proposed a useful criterion for judging the efficiency of algorithms for solving optimization problems [17]. He calls an algorithm for a class of problems “good” if the solution time for solving a problem in the class that requires binary encoding of length $L$ (also called the size of the problem) is bounded by a polynomial in $L$. This polynomial-time bound is valid globally, i.e., no matter where the solution starts and no matter whether the problem is the “best case” or the “worst case.”

In addition to these two criteria, practical performance in solving a large number of real problems tests the efficiency of optimization algorithms. For example, even though the simplex (pivot) method is not a polynomial-time algorithm using the criterion of computational complexity (it requires an exponential number of steps in the worst case), it solves most real problems quite well in practice.

0.2 Previous Approaches to Solving CP

Let us briefly outline the previous approaches to solving QP and CP and see how they meet, or do not meet, the above criteria. These approaches can be divided into two categories: those that search along the boundary of the feasible polytope, and those that move in the interior of the polytope.
The pivot-type methods, described by Beale [6], Cottle and Dantzig [11], Lemke [42] and Wolfe [66], belong to the first category. They move from vertex to vertex of the feasible polytope and converge in a finite number of steps for QP. They do not have a convergence ratio and they are not polynomial-time algorithms, although on the average they perform well in practice.

The active set methods (Fletcher [19], Gill and Murray [26]), including the gradient-projection method (Rosen [56]), the variable metric method (Davidon [14], Goldfarb [27]), and the reduced-gradient method (Wolfe [65]), also belong to the first category. They move along the faces or edges of the feasible polytope and converge in a finite number of steps for QP. They have a local convergence rate or ratio restricted to the active constraints set, and they are not polynomial-time algorithms.

The penalty or barrier function methods (Bartels [4], Carroll [8], Conn and Sinclair [10], Fiacco and McCormick [18], Frisch [20], Han and Mangasarian [30], Luenberger [44], Zangwill [69]) belong to the second category. They move to the optimal solution from the interior of the polytope and they have a local convergence rate or ratio. However, numerical problems and unfavorable computational structures may occur when they are implemented. They are not polynomial-time algorithms either.

Given that none of the above approaches are polynomial-time algorithms, the question arises: does there exist a polynomial-time algorithm for QP or for LP?

Attempting to answer this question, in 1979, L. Khachiyan proved that a certain LP algorithm, called the ellipsoid method, is polynomial [37]. This method proved to be a polynomial-time algorithm for convex quadratic programming as well (Kozlov et al. [40], Papadimitrou and Steiglitz [52]). In the method, there exists a global convergence ratio with respect to an error function defined by the volume of an ellipsoid containing the optimal solution point. Khachiyan's results are based on the work of other mathematicians (Shor [58]) in nonlinear programming in that they almost completely disregard the combinatorial nature of the problem. The ellipsoid
method does not deal with LP directly, but deals with an equivalent problem that requires the optimal objective value in advance. Although polynomial-time status is not affected by this requirement, the algorithm takes considerably more steps to converge if the value is unknown. Unfortunately, the ellipsoid method does not compete with the simplex method in practice, and its significance remains theoretical.

In another attempt, N. Karmarkar of Bell Labs proposed a projective algorithm to solve LP [35]. His algorithm converges to the optimal solution from the interior of the feasible polytope and reduces a cleverly constructed potential function at a global convergence ratio. Karmarkar’s efficient polynomial-time algorithm, which has sparked enormous interest, centers on a certain canonical form of linear programming. In each iteration, the algorithm projective transforms the current solution into the center of a simplex. In practice, his algorithm has competed with the simplex method. However, there are two drawbacks to his algorithm: (1) it does not generate the optimal dual solution, which has significant value in sensitivity analysis, and (2) it requires prior knowledge of the optimal objective value, as does the ellipsoid method.

Since Karmarkar’s algorithm was aimed at solving LP, the question of whether or not an efficient polynomial-time algorithm exists for solving QP remains open. In addition, as we have seen, none of the previous algorithms meet all of the criteria discussed above. For example, the pivot and active set methods globally converge in finite iterations and perform well in practice; however, they are not polynomial-time algorithms. The ellipsoid method is a polynomial-time algorithm, but, unfortunately, it cannot compete with the pivot method in solving most real problems. Karmarkar’s LP polynomial-time algorithm can be implemented efficiently if the optimal objective value is known in advance; otherwise, his “primal-dual” method doubles the size of the problem [35] and his “practical approach” is not a polynomial-time algorithm [36].

From this review, I conclude that:
(1) An efficient polynomial-time algorithm for QP has not yet been developed and implemented.

(2) The combinatorial nature of LP is discarded by all of the LP polynomial-time algorithms where an error or potential function is reduced at a fixed global convergence ratio.

(3) The LP polynomial-time algorithms have a number of practical difficulties, such as requiring prior knowledge of the optimal objective value.

In addition to the above review of constrained optimization, we will also look at a well-developed method for solving unconstrained optimization—the trust region (TR) method. In the TR method, a new iterative solution is obtained by optimizing a local quadratic model of the objective function over a trust ellipsoidal region centered about the current solution point. This approach is different from other “step-length-based methods” that line search along the descending direction of the objective function. The idea of defining a trust region was suggested for nonlinear least-squares problems by Levenberg [43] and Marquardt [47]. The applications of the technique to general nonlinear problems were considered by Goldfeld, Quandt, and Trotter [28]. Various schemes for updating the size of the trust region were proposed by Fletcher [19], Gay [23], Hebden [31] and Sorenson [60]. This development in solving unconstrained optimization problems has a certain impact on generating new techniques for solving constrained optimization problems, which will be further discussed in Chapter 1.

In this dissertation, I present a polynomial-time algorithm, based on previous developments and especially on Karmarkar’s algorithm for linear programming and the trust region method for unconstrained optimization, to meet all of the criteria for solving LP and QP. When implemented, the algorithm bypasses the difficulties found in existing LP polynomial-time algorithms. More significantly, I have developed a practical approach to solving more general optimization problems: linearly constrained convex programming (CP) problems.
0.3 Primal and Dual Models of CP

Many optimization problems can be modeled in terms of CP, such as portfolio investment, maximum-entropy, power-pooling, and nonlinear network optimization. CP is also important because many existing modern algorithms for general nonlinear programming are based on CP subroutines. The mathematical models of CP are described below.

Primal and Dual Models

Let the CP problem have the following standard form:

\begin{align*}
\text{CP} & \quad \text{minimize} \quad f(x) \\
& \quad \text{subject to} \quad x \in X = \{x \in R^n : Ax = b \text{ and } x \geq 0\}
\end{align*}

where \( A \in R^{m \times n} \) and \( b \in R^m \), and \( f(.) \) is convex and twice-continuous differentiable in the feasible polytope. Then, CP has a strong dual model with the following form:

\begin{align*}
\text{CD} & \quad \text{maximize} \quad d(x, y) = yb - \nabla f(x)x + f(x) \\
& \quad \text{subject to} \quad (x, y) \in Y = \{(x, y) : yA \leq \nabla f(x), \text{ and } x \in X\}
\end{align*}

where the row-vector \( y \in R^m \). It is well-known that for all \((x, y) \in Y\)

\begin{equation}
d(x, y) \leq z^* \leq f(x), \quad (0.1)
\end{equation}

where \( z^* \) designates the optimal objective value of CP. Usually CP is called the primal program and CD the dual program. The coherent relation between CP and CD plays an invaluable role in my approach. In addition, all coefficients and variables in the CP and CD models have conceptual interpretations, such as \( x \): the production vector, \( f(x) \): the production cost function, \( y \): the price vector, \( \nabla f(x) \): the marginal cost vector, and \( \nabla f(x) - yA \): the reduced-cost vector. Overall, CP is a problem of minimizing total production cost while meeting certain production demands. If \( f(x) \) is a linear function, such as

\begin{equation}
f(x) = cx \quad \text{where} \quad c \in R^n, \quad (0.2)
\end{equation}

then CP is reduced to a linear program (LP); if \( f(x) \) is convex quadratic such as
\[ f(x) = \frac{x^T Q x}{2} + cx \] (0.3)

where \( Q \) is a \( n \times n \) and positive semi-definite matrix, then CP is called a convex quadratic program (QP). Therefore, LP is a simple case of QP (\( Q = 0 \)), which itself is a special case of CP. In this dissertation, I focus exclusively on the standard forms of CP and CD instead of on other assumed canonical forms. Consequently, my approach provides a direct correspondence between the original coefficients and the resulting variables.

**Optimality Conditions**

Kuhn and Tucker [41] obtained a very important theoretical result in optimization. Their theorem concludes that \( x^* \) is an optimal solution for CP if and only if the following optimality conditions hold:

1) Primal feasibility: \( x^* \in X \)

2) Dual feasibility: \( \exists y^* \), such that \( x^*, y^* \) are feasible for CD: \( (x^*, y^*) \in Y \)

3) Complementary slackness:

\[
(\nabla f(x^*) - y^* A) \text{diag}(x^*) = 0.
\]

As a result of the above conditions, if an optimal feasible solution exists for QP, then via (0.3) there exists a basic optimal feasible solution such that

\[
\begin{pmatrix}
Q & -A^T \\
A & 0
\end{pmatrix}
\begin{pmatrix}
x^* \\
y^* T
\end{pmatrix}
=
\begin{pmatrix}
-c^T \\
b
\end{pmatrix}
\] (0.4)

where \( x_i^* = 0 \) if \( i \notin I_B \), an index subset of \( \{1, 2, \ldots, n\} \). Generally, the nonzero components of a basic feasible solution correspond to solutions of the linear system equations with \( d \) as the right-hand vector and \( B \) as the left-hand matrix, where \( d \) is a subvector of

\[
\begin{pmatrix}
-c^T \\
b
\end{pmatrix},
\]
and $B$ is a principal submatrix of

$$\begin{pmatrix} Q & -A^T \\ A & 0 \end{pmatrix}.$$ 

**Combinatorial Nature**

Therefore, the combinatorial properties of QP are similar to that of LP (Gács and Lovász [21], Papadimitriou and Steiglitz [52]):

1) Let the coefficients in $Q$, $A$, $c$, and $b$ all be integers. Then the basic feasible solutions of (0.4) are vectors of rational numbers, both the numerator and denominator of which are bounded by $2^L$. In other words, for any basic feasible solution, $x$,

$$x_i \leq 2^L,$$ 

for $i = 1, 2, \ldots, n,$

and either $x_i = 0$ or $x_i \geq 2^{-L}$, where $L$ is the number of bits in the input, which is also called the size of the optimization problem. For quadratic programming,

$$L = n^2 + mn + \lfloor \log |P| \rfloor,$$

and $P$ is the product of the nonzero integer coefficients appearing in $Q$, $c$, $A$, and $b$. This fact guarantees that all data in iterative computations can be represented by rational numbers with finite length $O(L)$.

2) Let $x^1$ be a basic feasible solution. Then, due to the quadratic formula, the objective value is bounded:

$$-2^{2L} \leq f(x^1) \leq 2^{3L},$$

where the denominator of $f(x^1)$ is bounded by $2^{2L}$, too.

3) Let two basic feasible solutions, $x^1$ and $x^2$, have different objective values. Then

$$|f(x^1) - f(x^2)| \geq 2^{-4L}.$$
Assumptions

We now assume that there exists an interior feasible solution $x^0$ for CP with

$A1 \quad x^0 \geq 2^{-L}e.$

We further make an implicit assumption that the optimal solution can be found in a bounded polytope, i.e.,

$A2 \quad x \leq 2^L e,$

where $e$ is the vector of all one’s, and, therefore, for all $x \in X$ in QP:

$$-2^{2L} \leq z^* \leq f(x) \leq 2^{3L}. \quad (0.5)$$

In general, we say that CP is solved if and only if an $x \in X$ has been found such that

$$f(x) - z^* \leq M^{-1}, \quad (0.6)$$

where $M = 2^{4L}$. Due to the above three facts and two assumptions, if the input data are all integers, then the exact optimal feasible solution of QP can be obtained by rounding the error from $x$, as is done in linear programming [35][37].

0.4 What Makes CP Hard to Solve?

In this section, I analyze what makes CP hard to solve, i.e., what the combinatorial nature of CP is. Basically, the answers to these questions lie in the existence of the inequality constraints: $x \geq 0$.

Actually, CP and an equality constrained convex program (ECP):

$ECP \quad \begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad Ax = b \quad \text{and} \quad x_i = 0 \quad \text{for} \quad i \notin I_B
\end{align*}$
where \( I_B \) is a subset of \( \{1, 2, \ldots, n\} \), can be made to share the same optimal solution by correctly partitioning the inequality constraints of CP into two sets: an active constraint set and a passive one. Essentially, ECP is an unconstrained optimization that can be solved much more efficiently than CP. As (0.4) indicated, if \( f(x) \) is quadratic, then the optimal solution for ECP can be obtained by solving a system of linear equations in \( O(n^3) \) arithmetic operations, i.e., in a polynomial of \( n \). Most of the previously mentioned algorithms, such as the pivot and active set methods, approach the optimal solution of CP by solving a sequence of ECP problems. The shortcoming of this is that if there are \( n \) inequality constraints in CP, then there are \( 2^n \) such ECP problems each of which could be a candidate for providing the optimal solution. According to the complexity criterion, neither the pivot method nor the active set method is a polynomial-time algorithm since the number of ECP problems that they solve is an exponential of \( n \) in the worst case. This illustrates why some of the previous approaches have failed to meet the polynomial-time criterion.

According to the convergence criterion, we need to know the local convergence ratio, \( \rho \), in a given algorithm [44]. \( \rho \) is defined as

\[
\frac{E(x^{k+1})}{E(x^k)} \leq \rho < 1 \quad \text{for a large enough} \quad k,
\]

where \( x^k \) is the sequence of iterative solutions generated by the algorithm, and \( E(x) \) is an error function with \( E(x) > 0 \) for all feasible \( x \neq x^* \) and \( E(x^*) = 0 \). A typical choice of error functions is

\[
E(x) = f(x) - z^*.
\]  

(0.7)

Obviously, the smaller the value of \( \rho \), the faster the algorithm. It is somewhat surprising that a useful local convergence ratio does not exist for the pivot methods. This is due to the "zigzagging" phenomenon, which occurs on the boundary of the feasible polytope during the course of these methods, and which is caused by the inequality constraints. Even if a local convergence ratio for these methods did exist, this ratio would not be sufficient to judge which method is superior since it only measures the speed of the algorithm asymptotically; there is no guarantee that this reduced ratio is valid for all \( k \), unless \( x^k \) is close to the optimal solution. Bringing \( x^k \) close to the optimal solution may require an exorbitant number of steps.
Overall, to develop an algorithm for QP that meets all of the criteria, we need to avoid the combinatorial nature of the inequality constraints and the "zigzagging" phenomenon on the boundary of the feasible polytope. We also need an error or potential function that is reduced at a global convergence ratio. These conditions are met in this dissertation by introducing the projective transformation and interior ellipsoid (PTIE) method.
Chapter 1 Interior Ellipsoid Method

1.1 Introduction

In this chapter, I first introduce the interior ellipsoid, a geometric expression that avoids both the combinatorial nature of the inequality constraints and the "zigzagging" phenomenon on the boundary of the feasible polytope. Next, the structure of weighted gradient-projection, an algebraic representation, is discussed. Under this structure, the simplex method, the gradient-projection method, and Karmarkar's method can be unified. The interior ellipsoid (IE) method for solving CP, along with its local convergence ratio and solution strategy, are then analyzed.

1.2 Interior Ellipsoid: A Geometric Expression

To illustrate the basic concept of the IE method, I shall use a linear objective function. Figure 1.1 displays a feasible polytope with the arrow pointing in the descent direction of the objective function. In the pivot method, the solution moves from vertex to vertex, i.e., from A to B, ..., converging to the optimal solution point P.

![Figure 1.1. Objective Contour and Feasible Polytope](image-url)
In the gradient-projection (GP) method, the solution may start from the interior of the polytope. As illustrated, if the starting point happened to be at point $Q$, then the method could reach point $P$ in one step. However, if we started from point $R$, the method would generate the boundary point $C$, and then would move along the edges of the polytope. As soon as the iterative solution reached the boundary, a “zigzagging” phenomenon would occur and a combinatorial decision would have to be made to reform the active constraint set at each step. In the worst case, the optimal solution would be reached in an exponential number of steps.

Naturally, the question arises: can we avoid hitting the “wrong” boundary? In other words, can we develop a mechanism to move the solution in the direction $R - S$ instead of $R - C$?

One way to accomplish this objective is to apply a geometric expression from the concept of the trust region method, which draws an ellipsoidal region around the starting point. Here, the trust ellipsoidal region has to be in the interior of the feasible polytope, as shown in Figure 1.2. The objective function can then be minimized over this interior ellipsoid to generate the next interior solution point $P'$.

\[\text{Figure 1.2. The Interior Ellipsoid Approach}\]
A series of such ellipsoids can be constructed to make the iterative solutions converge to the optimal solution point that sits on the boundary. If the optimal solution point itself is an interior solution (this can happen if the objective is a non-linear function), then the series terminates as soon as the optimal point is encircled by the newest ellipsoid.

1.3 Weighted Gradient-Projection: An Algebraic Representation

Geometrically, we can easily draw an ellipsoid around the starting solution point; algebraically, how to represent the ellipsoid in our standard CP form is not clear. In this section, I start with the simplex and the gradient-projection methods to see how their geometric characteristics are algebraically represented by a weighted gradient-projection structure.

For convenience, I use a linear objective function (0.2) to illustrate the algebraic structure. Let the feasible solution at step $k$ be $x^k$. Given nonnegative weights $w_1, w_2, \ldots, w_n$, the weighted gradient-projection vector, $p$, at $x^k$ is defined as:

$$p = Wc^T - WAT(AW^2A^T)^{-1}AW^2c^T$$

$$= Wr,$$

where

$$W = \text{diag}(w_1, w_2, \ldots, w_n), \quad r = (c - yA)^T,$$

and

$$y = cW^2A^T(AW^2A^T)^{-1}.$$

$W$ weights the column of $A$ and $c$ as

$$AW = (w_1a_1 \quad w_2a_2 \quad \ldots \quad w_na_n)$$

and

$$cW = (w_1c_1 \quad w_2c_2 \quad \ldots \quad w_nc_n)$$
where $a_i$ is the $i^{th}$ column of $A$. Thus, $p$ is the projection from $Wc^T$ to the null space of $AW$. I will show that the simplex method, the gradient-projection method, and even Karmarkar-type algorithms all fit into this algebraic structure, and that each chooses $W$ using the information from the current feasible solution $x^k$.

**Simplex**

Note that any current feasible solution $x^k$ for the simplex method is a vertex associated with a basis $B$. If $x^k$ is nondegenerate, $W$ can be chosen as

$$w_i = \begin{cases} 1, & \text{if } x^k_i > 0; \\ 0, & \text{otherwise.} \end{cases}$$

(1.2)

Therefore, there are only $m$ nonzero $w_i$'s. Hence, the simplex method weights the column discriminatively by choosing $m$ columns. Without losing generality, we may assume that

$$B = (a_1 \ a_2 \ldots \ a_m),$$

i.e., $w_i > 0$ for $i = 1, \ldots, m$ and $w_i = 0$ for $i = m+1, \ldots, n$. When the above choice for $W$ is followed, $y$ corresponds to the price vector at basis $B$, i.e.,

$$y = c_BB^{-1},$$

where $c_B$ is the vector of the objective function coefficients corresponding to the current basic variables. Likewise, $r$ corresponds exactly to the reduced cost vector at $B$,

$$r = c - c_BB^{-1}A;$$

and $p$ corresponds to the complementary slackness vector at $B$, which is always zero in the simplex method:

$$\text{diag}(x^k)r = Wr = p = 0.$$

Since $p$ is a zero vector at base $B$, or the current active constraint set $\{m+1, \ldots, n\}$, no progress can be made by moving in the direction of $-p$, and the simplex method has to select a new base at each iteration. If $r \geq 0$, then $x^k$ is optimal; otherwise, a new base, which corresponds to an adjacent vertex of $x^k$, must be formed by evaluating the sign of $r$ and using a feasible ratio test. Since the number of all possible vertices is finite, the simplex method has a finite convergence.
Gradient-Projection

Note that the gradient-projection method starts with any feasible point. Let $W$ be chosen as in (1.2). The difference in this case is that $x^k$ may not be an extreme solution, i.e., $W$ may have more than $m$ non-zero elements. However, $W$ still weights $A$ discriminatively. $y$ is the corresponding price vector, $r$ is the reduced cost vector with the set $\{i : x_i = 0\}$ of active constraints, and $p$ represents the gradient-projection to the null space of current active constraints:

$$p = W r = \begin{cases} r_i, & \text{if } x_i^k > 0; \\ 0, & \text{otherwise.} \end{cases}$$

If $r \geq 0$, then $x^k$ is the optimal solution. Otherwise, either an improvement along a face of the polytope can be made in the form of:

$$x^{k+1} = x^k - \beta p$$

where

$$\beta = \min_{1 \leq i \leq n} \left\{ \frac{x_i^k}{p_i} : p_i > 0 \right\},$$

or the algorithm "jams" at this set of active constraints if $p = 0$. In the latter case, a new set of active constraints must be formed by dropping and adding indices from and to the set using the sign of $r$ and a ratio test. From now on the gradient-projection method merges into the simplex method.

Scaled Gradient-Projection

Ignoring the canonical form of Karmarkar's algorithm, this projection chooses $W$ as the current strict positive interior solution itself:

$$W = \text{diag}(x^k).$$

Therefore,

$$p = \text{diag}(x^k) (c - y A)^T.$$
\( p \) represents the complementary slackness vector and \( r \) is the reduced cost vector, both associated with \( y \) as the price vector. The next iterate solution is obtained via

\[
x^{k+1} = W(e - \beta \frac{p}{\|p\|}) = x^k - \beta \frac{Wp}{\|p\|},
\]

and \( 0 < \beta < 1 \). In this case, \( x^k (> 0) \) is an interior point for all \( k \). Therefore, \( W \) always weights the columns of \( A \) smoothly and continuously. Intuitively, if \( x^k \) converges to an optimal extreme solution, \( W \) will eventually select the right basis in the limiting sense. In other words, each column of \( A \) is not eliminated from the basis until \( x^k \) converges to a vertex. It can be verified that \( x^{k+1} \) is the optimal solution for the following sub-optimization problem starting from \( x^k \):

\[
\text{LP1.1} \quad \begin{align*}
\text{minimize} & \quad cx \\
\text{subject to} & \quad Ax = b \\
& \quad \|W^{-1}(x - x^k)\|_2^2 \leq \beta^2 < 1.
\end{align*}
\]

Noting (1.3), the last constraint, \( \|W^{-1}(x - x^k)\|_2^2 \leq \beta^2 \), can be rewritten as

\[
\sum_{i=1}^{n} \left( \frac{x_i - x_i^k}{x_i^k} \right)^2 \leq \beta^2,
\]

which corresponds to an interior ellipsoid in the positive orthant \( R^+_n = \{ x \in R^n : x \geq 0 \} \). Therefore, \( \{ x : Ax = b \text{ and } \|W^{-1}(x - x^k)\|_2^2 \leq \beta^2 \} \) is a successive algebraic representation of the interior ellipsoid centered at \( x^k \) in the feasible polytope of CP.

Overall, for a general convex objective function, the sub-optimization problem solved over the interior ellipsoid at each step has the following algebraic form:

\[
\text{CP1.1} \quad \begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad Ax = b \\
& \quad \|W^{-1}(x - x^k)\|_2^2 \leq \beta^2,
\end{align*}
\]

where \( W \) is chosen to maintain \( x > 0 \). The parameter \( \beta \) characterizes the size of the ellipsoid, and \( W \) affects its orientation and shape. In this dissertation, I choose \( W \) as in (1.3).
1.4 Interior Ellipsoid (IE) Method for CP

It is somehow more convenient to reformulate CP1.1 in terms of the variables

\[ x' = W^{-1} x. \]

Thereby, CP1.1 becomes

\[
\begin{align*}
\text{CP1.2} & \quad \text{minimize} \quad f'(x') = f(Wx') \\
\text{subject to} \quad A'x' &= b' \\
\|x' - e\|^2 &\leq \beta^2,
\end{align*}
\]

where \( A' = AW, \ b' = b, \) and \( e \) is the current feasible solution. Then the interior ellipsoid method can be described as follows.

**Algorithm 1.1**

At the \( k^{th} \) iteration do

begin

\[ W = \text{diag}(x^k); \]

let \( a \) be the minimal solution for CP1.2;

\[ x^{k+1} = W a; \]

\[ k = k + 1; \]

end.

If the optimal solution for CP is a boundary solution point, i.e., at least one component of \( x^* \) is zero, then \( a \) will be on the boundary of the ellipsoid (sphere) as well. In other words, the last inequality constraint for CP1.2 will be binding so that CP1.2 is in effect an equality constrained (convex) optimization. Consequently, the combinatorial nature is eliminated from this approach. The optimality conditions for CP1.2 are given by

\[
\begin{align*}
\nabla f'(a) - yA' + \mu(a - e)^T &= 0; \quad \text{(1.5a)} \\
A' a &= b'; \quad \text{(1.5b)}
\end{align*}
\]
\[ \|a - e\|^2 \leq \beta^2 \quad \text{and} \quad \mu \geq 0; \]  

and

\[ \mu(\beta - \|a - e\|) = 0. \]  

(1.5c)

(1.5d)

Multiplying (1.5a) by \( A'^T \) from the right,

\[ \nabla f'(a)A'^T - yA'A'^T + \mu(a - e)A'^T = 0. \]

Since both \( a \) and \( e \) satisfy the equality constraints,

\[ A'(a - e) = b' - b' = 0 \]

and

\[ yA'A'^T = \nabla f'(a)A'^T, \]

(1.6)

let

\[ p^k = (\nabla f'(a) - yA')^T. \]

(1.7a)

Then if \( p^k \neq 0 \), from (1.5a) and (1.5c),

\[ a = e - \beta \frac{p^k}{\|p^k\|} \]

(1.7b)

and

\[ \mu = \frac{\|p^k\|}{\beta}. \]

(1.7c)

With respect to the original variables and coefficients of CP1.1, we derive the following expressions:

\[ x^{k+1} = Wa = x^k - \beta \frac{Wp^k}{\|p^k\|}, \]

(1.8a)

\[ p^k = W(\nabla f(x^{k+1}) - yA)^T, \]

(1.8b)

and

\[ \nabla f(x^{k+1})Wp^k = \nabla f'(a)p^k = \|p^k\|^2. \]  

(1.8c)
It is obvious that $x^{k+1} > 0$ if $x^k > 0$, i.e., the iterative solution remains an interior (positive) feasible solution.

In order to analyze convergence of the method, the following lemmas are introduced.

**Lemma 1.1** If $p^k = 0$ (i.e., $\mu = 0$) for $k < \infty$, then $x^{k+1}$ and $y$ are optimal for CP and CD.

**Proof.** Note that

$$p^k = 0$$

implies

$$(\nabla f(x^{k+1}) - yA)W = 0,$$

which implies

$$(\nabla f(x^{k+1}) - yA) = 0,$$  \hspace{1cm} (1.9)

and

$$(\nabla f(x^{k+1}) - yA)\text{diag}(x^{k+1}) = 0.$$  \hspace{1cm} (1.10)

Therefore, the conclusion in Lemma 1.1 follows from the optimality conditions that 1) $x^{k+1}$ is feasible for CP, 2) $x^{k+1}$ and $y$ are feasible for CD from (1.9), and 3) complementary slackness is satisfied from (1.10). Q.E.D.

Essentially, Lemma 1.1 claims that the IE method will never "jam" unless the optimal solution for CP is generated in a finite number of iterations. If $\|p^k\| > 0$ for all finite $k$, the second lemma claims that $\|p^k\| \to 0$, where $\to$ designates "converges to."

**Lemma 1.2**

Let $\|p^k\| > 0$ for all $k < \infty$ and the optimal objective value of CP be bounded from below. Then $\|p^k\| \to 0$ (i.e., $\mu \to 0$). Furthermore, if the Hessian of $f(.)$ is positive-definite, then $x^k$ converges.
Proof. Using (1.8a) and (1.8c), we have

\[
f(x^k) = f(x^{k+1} + \beta \frac{Wp^k}{\|p^k\|})
\]

\[
= f(x^{k+1}) + \frac{\beta}{\|p^k\|} \nabla f(x^{k+1}) Wp^k + \frac{\beta^2}{2\|p^k\|^2} (p^k)^T W \nabla^2 f(.) W p^k
\]

\[= f(x^{k+1}) + \beta \|p^k\| + \frac{\beta^2}{2\|p^k\|^2} (p^k)^T W \nabla^2 f(.) W p^k,
\]

where the Hessian \( \nabla^2 f(.) \) is at least positive semi-definite. Therefore,

\[
\beta \|p^k\| \leq f(x^k) - f(x^{k+1}).
\]

Since \( f(x^k) \) is monotonically decreasing and is bounded from below, \( f(x^k) \) must converge and \( f(x^k) - f(x^{k+1}) \to 0 \), which implies \( \|p^k\| \to 0 \). Moreover, via (1.11),

\[
\frac{\beta^2}{2\|p^k\|^2} (p^k)^T W \nabla^2 f(.) W p^k \leq f(x^k) - f(x^{k+1}).
\]

Hence, if \( \nabla^2 f(.) \) is positive definite, it follows that

\[
\frac{\|Wp^k\|}{\|p^k\|} \to 0,
\]

which implies from (1.8a) that

\[
\|x^k - x^{k+1}\| \to 0.
\]

As the feasible polytope is bounded and closed (compact), \( \{x^k\} \) must have a subsequence converging to \( x^\infty \). This fact and (1.12) imply that the whole sequence \( \{x^k\} \) converges to \( x^\infty \).

Q.E.D.

Let \( y^{k+1} \) be the \( y \) that appeared in (1.5a) at the \( k^{th} \) iteration. \( y^{k+1} \) always exists as a solution of (1.6), even though it may not be unique [12]. Therefore, if \( x^k \) converges, \( y^k \) does not necessarily converge. This minor flaw can be fixed by adding a nondegenerate assumption for \( x^\infty \) (with at least \( m \) nonzero components). Since there exists a unique \( y \) that satisfies (1.6), we conclude that \( y^k \) converges if \( x^k \) does. This leads to
Lemma 1.3

If \( \|p^k\| > 0 \) for all \( k < \infty \), \( x^k \to x^\infty \), \( y^k \to y^\infty \) and \( \|p^k\| \to 0 \), then \( x^\infty \) and \( y^\infty \) are feasible for CD.

Proof. We have

\[
p_i^\infty = x_i^\infty (\nabla f(x^\infty) - y^\infty A)_i = 0 \quad \text{for} \quad i = 1, 2, \ldots, n. \quad (1.13)
\]

Suppose \( y^\infty \) is not feasible for CD, i.e. \( \exists \varepsilon > 0 \) and \( 1 \leq j \leq n \), such that

\[
(\nabla f(x^\infty) - y^\infty A)_j \leq -\varepsilon < 0;
\]

then \( \exists K > 0 \) such that for all \( \infty > k > K \)

\[
(\nabla f(x^{k+1}) - y^{k+1} A)_j < -\frac{\varepsilon}{2}.
\]

At the \( k^{th} (k > K) \) iteration of the algorithm,

\[
x_{j}^{k+1} = x_{j}^{k} (1 - \beta \frac{x_{j}^{k}(\nabla f(x^{k+1}) - y^{k+1} A)_j}{\|p^k\|}) > x_{j}^{k};
\]

hence,

\[
x_{j}^{k+1} > x_{j}^{k} > x_{j}^{K} > 0 \quad \text{for all} \quad k > K.
\]

Thus, \( \{x_{j}^{k}\} \) is a strictly increasing positive series for \( k > K \). Since neither \( x_{j}^{k} \) nor \( (\nabla f(x^{k}) - y^{k} A)_j \) converges to 0, \( x_{j}^{k}(\nabla f(x^{k}) - y^{k} A)_j \) does not converge to 0. This contradicts (1.13). Therefore, it must be true that

\[
\nabla f(x^\infty) - y^\infty A \geq 0,
\]

i.e., \( x^\infty \) and \( y^\infty \) are feasible for CD. Q.E.D

With the above three lemmas, we derive
Theorem 1.1

Let the optimal objective value of CP be bounded from below and let $x^k$ and $y^k$ converge. Then, Algorithm 1.1 generates solution sequences $x^k$ and $y^k$ that converge to the optimal solutions for both CP and CD.

Proof. If $p^k = 0$ for $k < \infty$, the conclusion of Theorem 1.1 follows from Lemma 1.1; otherwise, from Lemma 1.3, $x^\infty$ and $y^\infty$ are feasible for CP and CD, and from Lemma 1.2, $\|p^\infty\| = 0$, which implies that complementary slackness is satisfied. Thus, $x^\infty$ and $y^\infty$ are optimal for CP and CD. Q.E.D.

Here we see that $\|p^k\|$ can be used to develop a stopping criterion for Algorithm 1.1. Moreover, if $f(.)$ is strictly convex, then the Hessian $\nabla^2 f(.)$ is positive definite and the following corollary holds.

Corollary 1.1

Let the optimal objective value of CP be bounded from below and let $f(.)$ be strictly convex. Then Algorithm 1.1 generates solution sequences that converge to the optimal solutions for both CP and CD.

To evaluate the convergence ratio of the IE method, we have

Theorem 1.2

Let $x^{k+1}$ and $y^{k+1}$ be feasible for CD. Then

$$f(x^{k+1}) - z^* \leq (1 - \frac{\beta}{\sqrt{n}})(f(x^k) - z^*).$$

Proof. Since $f(x)$ is a convex function,

$$f(x^k) - f(x^{k+1}) \geq \nabla f(x^{k+1})(x^k - x^{k+1}). \quad (1.14)$$

We are also given that $y^{k+1}$ is feasible for CD; so from (0.1)

$$d(x^{k+1}, y^{k+1}) - z^* = f(x^{k+1}) - \nabla f(x^{k+1})x^{k+1} + y^{k+1}b - z^* \leq 0. \quad (1.15)$$

23
From (1.14) and (1.15),

\[ \nabla f(x^{k+1})x^k - y^{k+1}b + d(x^{k+1}, y^{k+1}) - z^* \leq f(x^k) - z^*. \]  

(1.16)

According to (1.8a) and (1.8c),

\[ \nabla f(x^{k+1})x^{k+1} = \nabla f(x^{k+1})x^k - \beta \|p^k\|. \]  

(1.17)

Using Hölder's inequality and (1.8b) and noting \( p^k \geq 0 \),

\[ \|p^k\| \geq \frac{1}{\sqrt{n}}\left(\sum_{i=1}^{n} |p_i^k|\right) \]

\[ = \frac{1}{\sqrt{n}}\left(\sum_{i=1}^{n} p_i^k\right) \]

\[ = \frac{1}{\sqrt{n}}(\nabla f(x^{k+1})x^k - y^{k+1}b). \]  

(1.18)

Due to (1.15), (1.16), (1.17), and (1.18),

\[ f(x^{k+1}) - z^* = \nabla f(x^{k+1})x^{k+1} - y^{k+1}b + d(x^{k+1}, y^{k+1}) - z^* \]

\[ \leq \nabla f(x^{k+1})x^k - \beta \|p^k\| - y^{k+1}b + (1 - \frac{\beta}{\sqrt{n}})(d(x^{k+1}, y^{k+1}) - z^*) \]

\[ \leq (1 - \frac{\beta}{\sqrt{n}})(\nabla f(x^{k+1})x^k - y^{k+1}b + d(x^{k+1}, y^{k+1}) - z^*) \]

\[ \leq (1 - \frac{\beta}{\sqrt{n}})(f(x^k) - z^*). \]  

\[ Q.E.D. \]

According to Lemma 1.3, \( y^k \) does converge to a dual feasible solution. Therefore, Theorem 1.2 essentially claims that the local (asymptotic) convergence ratio of the IE method is \( (1 - O(\frac{1}{\sqrt{n}})) \), which is a polynomial in \( n \). As the local convergence ratio counts the number of iterations in the course of the algorithm, the question arises: what is the computational complexity in each iteration, i.e., in solving CP1.2 (or CP1.1)? By solving CP1.2, we can determine whether or not CP has a positive interior optimal solution (PIOS). To do so, we need only solve the following ECP:
CP1.3 \[ \text{minimize} \quad f(x) \]
subject to \[ Ax = b. \]

If an optimal solution for CP1.3 exists and if the solution is nonnegative, then CP shares the same optimal solution with CP1.3; otherwise, CP has no PIOS, and the sphere constraint will be binding in CP1.2. Therefore, we face an ECP at each iteration of the IE method. Since ECP is a continuous optimization, it can be solved efficiently. For example, if \( f(.) \) is a linear function, then CP1.1 or CP1.2 can be solved in \( O(n^3) \) arithmetic operations; if \( f(.) \) is convex quadratic given by (0.3), then the optimality conditions (1.5a) and (1.5b) can be written in a matrix form:

\[
\begin{pmatrix}
Q + \mu W^{-2} & -A^T \\
A & 0
\end{pmatrix}
\begin{pmatrix}
x^{k+1} \\
y^{k+1}
\end{pmatrix}
= \begin{pmatrix}
-e^T + \mu W^{-1} e \\
b
\end{pmatrix}.
\]

Therefore, like the trust region method, CP1.1 or CP1.2 can be solved by approximating the multiplier \( \mu \) in \( O(Ln^3) \) operations. Essentially, \( \mu \) characterizes the radius \( \beta \) of the interior sphere from (1.7c). Consequently, searching for \( \mu \) is equivalent to searching for the right size of the interior trust ellipsoid (sphere) region.

Even if CP1.2 could be solved in polynomial time, the IE method is not a polynomial algorithm because there is no guarantee that \( (f(x) - z^*) \) will be reduced globally at the above convergence ratio. This leads me to propose a modified IE method for CP in the next chapter—an error function is reduced at a global ratio whose ratio is polynomial in \( L \) and \( n \), where each iteration can be computed in polynomial time if the objective function is convex quadratic.

1.5 IE Method in CP General Form

In the preceding section, the IE method was discussed in the standard CP form in which all variables are subject to nonnegative constraints. If some of the variables are "free," say, \( x_i \) can be either negative or nonnegative, then we usually substitute two nonnegative variables for \( x_i \), i.e., let \( x_i = x'_i - x''_i \) where \( x'_i \geq 0 \) and \( x''_i \geq 0 \). Therefore, any CP problem can be transformed into the standard form while the variables are doubled. In order to handle general CP problems efficiently, I introduce a variant of the IE method in CP general form:
**Algorithm 1.2**

At the $k^{th}$ iteration do

begin

\[ D = \text{diag}(Ax^k - b); \]

let $x^{k+1}$ be the minimal solution for the problem GCP.1;

\[ k = k + 1; \]

end

where problem GCP.1 is given by

**GCP.1**

minimize \( f(x) \)

subject to \( (x - x^k)^T A^T D^{-2} A(x - x^k) \leq \beta^2. \)

Overall, both Theorem 1.1 and Theorem 1.2 hold for Algorithm 1.2. Specifically, the set \( \{ x : (x - x^k)^T A^T D^{-2} A(x - x^k) \leq \beta^2 \} \) represents the interior ellipsoid centered about $x^k$, \( \{ x^k \} \) will remain as interior solutions (i.e., $Ax^k > b$) converging to the optimal solution, and the local convergence ratio is identical to the one in Algorithm 1.1.
1.6 Summary

The interior ellipsoid method has been introduced to solve linearly constrained convex programming. Because the IE method eliminates the combinatorial nature embedded in inequality constraints, it avoids the “zigzagging” phenomenon on the boundary of the feasible polytope. In addition, it can be used to analyze the local convergence rate. In terms of optimality conditions, the differences among the iterative solution strategies of the primal simplex (pivot) method (PS), the dual simplex method (DS), the gradient-projection method (GP), and the IE method are summarized in Figure 1.3.

<table>
<thead>
<tr>
<th>OPTIMALITY CONDITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>primal feasibility</td>
</tr>
<tr>
<td>PS</td>
</tr>
<tr>
<td>DS</td>
</tr>
<tr>
<td>GP</td>
</tr>
<tr>
<td>IE</td>
</tr>
</tbody>
</table>

Figure 1.3. Solution Strategies of the Major CP Algorithms

From Figure 1.3, we can see that for the IE method complementary slackness is not satisfied at each iteration but is only satisfied in the final stage, which basically explains why the IE method avoids the “zigzagging” phenomenon during the course of the algorithm.
Chapter 2  PTIE Method: Linear Programming

2.1 Introduction

The birth of linear programming (LP) is usually identified with the development of the simplex method in 1947 by George B. Dantzig [13]. It says much for the algorithm’s originator that the simplex method, which is not polynomial, remains a major algorithm used in optimization systems. Due to its simple linear structure, LP has sparked tremendous interest among scientists in developing polynomial algorithms. Khachiyan proved the first polynomial-time algorithm in 1979 [37], and Karmarkar published another one in 1984 [35].

Two major difficulties found in Karmarkar’s algorithm are: 1) it requires prior knowledge of the exact minimal objective value, which is usually unknown in advance; and 2) it does not generate the optimal dual solution that is significant in sensitivity analysis.

If the minimal objective value is unknown, a binary-search technique is employed in Karmarkar’s sliding objective function method, or the primal and dual problems are adjoined and solved together [35]. While both of the above approaches achieve polynomial complexity, the former method dramatically increases the computational time and the latter doubles the problem size. Todd and Burrell [62] proposed a polynomial method using the dual variables in Karmarkar’s LP canonical form. Their method generates not only a sequence of interior feasible solutions, but also a sequence of objective lower bounds that converges to the minimal objective value from below. Their method uses a pre-scaling technique to force a “predictable effect” constraint in the transformation from the LP standard form to Karmarkar’s canonical form. In another attempt to overcome the difficulties, Karmarkar recently presented the “two-dimensional search” technique [36]. However, this approach has not been proved to be a polynomial-time algorithm.
In order to achieve both polynomial complexity and practical efficiency in overcoming the two difficulties in Karmarkar's original algorithm, I developed a polynomial variant of Karmarkar's algorithm in the LP standard form. The variant does not require prior knowledge of the minimal objective value, and it simultaneously generates both optimal primal and dual solutions without adjoining the two programs together. The variant does not require explicit transformation of the LP standard form into Karmarkar's canonical form; therefore, it avoids some numerical problems in using the pre-scaling technique.

In addition to describing the polynomial variant, I will also discuss the relation between the ellipsoid method [37] and this variant, and will report my initial computational results that suggest the usefulness of this variant in practice.

2.2 Potential Function and Projective Transformation

I begin by introducing a new potential function, which is similar to the one in Karmarkar's algorithm, to measure the convergence rate of the optimization process.

Potential Function

Let \( x \) be an interior feasible solution for CP, and \( z \leq z^* \). The potential function associated with the CP standard form is defined by

\[
P(x, z) = (n + 1)\ln(f(x) - z) - \sum_{i=1}^{n} \ln(x_i)
\]

where \( 0 < z \in X \) and \( f(x) > z^* \geq z \). By a simple calculation, we have the following equality

\[
\frac{f(x) - z}{f(x^0) - z^0} = \gamma(x)\exp\left(\frac{P(x, z) - P(x^0, z^0)}{n + 1}\right)
\]

where \( x^0 \leq z^* \), and

\[
\gamma(x) = \exp\left(\frac{\sum_{i=1}^{n} \ln(x_i) - \sum_{i=1}^{n} \ln(x_i^0)}{n + 1}\right).
\]
By the assumptions A1 and A2, \( \gamma(x) \leq 2^2L \) in the feasible region \( X \). If \( P(x^k, z^k) \) tends to \(-\infty\) along some sequences \( \{0 < x^k \in X\} \) and \( \{-\infty < z^k \leq z^*\} \), then \( f(x^k) - z^k \) converges to zero. This implies that the minimization of the potential function \( P(x, z) \), subject to the constraints \( 0 < x \in X \) and \( z \leq z^* < f(x) \), leads to the minimal solution of CP. The algorithm that is described in this section generates a sequence \( \{0 < x^k \in X\} \) and \( \{-\infty < z^k \leq z^*\} \) such that

\[
P(x^{k+1}, z^{k+1}) \leq P(x^k, z^k) - \alpha \quad \text{for} \quad k = 1, 2, \ldots , \tag{2.1}
\]

where \( \alpha \geq 0.2 \). Hence,

\[
P(x^k, z^k) \leq P(x^0, z^0) - k\alpha \quad \text{for} \quad k = 1, 2, \ldots .
\]

Thus,

\[
f(x^k) - z^k \leq 2^{2L}(f(x^0) - z^0)\exp(-\frac{k\alpha}{n + 1}) \leq 2^{5L}\exp(-\frac{k\alpha}{n + 1}).
\]

Therefore,

\[
f(x^k) - z^k \leq 2^{-4L} \quad \text{for} \quad k \geq 45L(n + 1).
\]

In this approach, I will implicitly use the following projective transformation.

**Projective Transformation**

Let \( z^k \) be the interior feasible solution of CP and \( W \) be given by (1.3). The projective transformation \( T : \mathbb{R}^n \rightarrow \mathbb{R}^{n+1} \) is defined by

\[
x'[n] = \frac{(n + 1)W^{-1}x}{e^TW^{-1}x + 1}
\]

and

\[
x'_{n+1} = \frac{n + 1}{e^TW^{-1}x + 1},
\]

where \( x' = T(x) \in \mathbb{R}^{n+1} \), and \( x'[n] \) denotes the \( n \) vector of the first \( n \) components of \( x' \). Via \( T \), the feasible region \( X \) is mapped onto

\[
X' = \{x' \in \mathbb{R}^{n+1} : A' = b', x' \geq 0 \text{ and } x'_{n+1} > 0\},
\]

30
where $A' \in R^{(m+1) \times (n+1)}$

$$A' = \begin{pmatrix} AW, \ e^T & -b \end{pmatrix},$$

and $b' \in R^{n+1}$

$$b' = \begin{pmatrix} 0 \\ n + 1 \end{pmatrix}.$$

The new feasible region $X'$ is contained in the simplex

$$S = \{ x' \in R^{n+1} : e^T x' = n + 1 \text{ and } x' \geq 0 \}$$

where $e = T(x^k)$ is the center shown in Figure 2.1. Conversely, for any $x' \in X'$, an

$x \in X$ can be obtained via $T^{-1} : R^{n+1} \rightarrow R^n$

$$x = T^{-1}(x') = \frac{W x'[n]}{x'_{n+1}}.$$  

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure2.1.png}
\caption{The Simplex, Inscribing, and Circumscribing Spheres}
\end{figure}

One key observation regarding $S$ is as follows. Let $r$ be the radius of the

sphere centered at $e$ that inscribes $S$, and $R$ be the radius of the sphere centered

at $e$ that circumscribes $S$. Then it can be verified that $r = 1$, $R = \sqrt{n(n+1)}$, and the ratio of the two radii

$$\frac{r}{R} = \frac{1}{\sqrt{n(n+1)}} > \frac{1}{n + 1}. \quad (2.2)$$
I will presently show how to transform the following LP standard form into the canonical form by means of the projective transformation.

**LP Standard Form**

Let $f(x)$ be given by (0.2) on page 6. Then CP becomes

\[
\begin{align*}
\text{minimize} & \quad cx \\
\text{subject to} & \quad x \in X = \{x \in \mathbb{R}^n : Ax = b \text{ and } x \geq 0\},
\end{align*}
\]

which we call the LP standard form. Accordingly, CD reduces to

\[
\begin{align*}
\text{maximize} & \quad yb \\
\text{subject to} & \quad y \in Y = \{y \in \mathbb{R}^m : yA \leq c\}.
\end{align*}
\]

By the well-known duality theorem, we see that

\[
yb \leq z^* \leq cx \quad \text{for all } x \in X \text{ and } y \in Y.
\]

Using the projective transformation $T$, LP is related to

\[
\begin{align*}
\text{minimize} & \quad c'x' \\
\text{subject to} & \quad x' \in X'
\end{align*}
\]

where $c' = (cW, -z)$ and $z \leq z^*$. LP$(z)$ is equivalent to Karmarkar’s canonical form except for the objective function, in which $z$, called the “cut” here, is a lower bound of the minimal objective value for LP. Hence, in the interior feasible region of LP$(z)$, the objective function has a nonnegative value. (Otherwise, via $T^{-1}$, we will obtain an $x \in X$ such that $cx < z^*$.)

Applying the IE method to LP$(z)$ with $e$ as the starting point, we are concerned with the following problem at each iteration:

\[
\begin{align*}
\text{minimize} & \quad c'x' \\
\text{subject to} & \quad x' \in X' \\
& \quad \|x' - e\|^2 \leq \beta^2 < 1.
\end{align*}
\]
Since $e$ is the center of the simplex $S$, the interior ellipsoid reduces to the inscribing sphere (with radius $\beta$). Obviously, $x' > 0$ whenever $\|x' - e\| < 1$. This implies that the nonnegativity constraints in $X'$ are redundant. Let $a$ be the minimal solution of LP2.1($z^*$), i.e., $z = z^*$, and note that there exists an $x^{**} = T(x^*) \in X'$ such that $c^* x^{**} = 0$. Then, due to the inequality (2.2),

$$\frac{c' a}{c' e} \leq 1 - \frac{\beta}{n + 1}. \quad (2.3)$$

In addition, the following inequality was proved by Karmarkar [35]:

$$\sum_{i=1}^{n+1} \ln(a_i) \geq \frac{-\beta^2}{2(1 - \beta)^2}. \quad (2.4)$$

Consequently, by letting $x^{k+1} = T^{-1}(a)$, we have

$$P(x^{k+1}, z^*) - P(x^k, z^*) = (n + 1) \ln \left( \frac{c' a}{c' e} \right) - \sum_{i=1}^{n+1} \ln(a'_i)$$

$$\leq (n + 1) \ln(1 - \frac{\beta}{n + 1}) + \frac{\beta^2}{2(1 - \beta)^2}$$

$$\leq -\beta + \frac{\beta^2}{2(1 - \beta)^2}. \quad (2.5)$$

Let

$$\alpha = \beta - \frac{\beta^2}{2(1 - \beta)^2}. \quad$$

Then, $\alpha \geq 0.2$ for $\beta = 0.27 - 0.36$, which leads to a polynomial-time algorithm by setting $z^k = z^*$, as discussed on page 30. But this approach still relies on prior knowledge of the minimal objective value. Therefore, the question arises: if this value is unknown, how can we solve linear programs in polynomial time in terms of practical efficiency?
2.3 Cutting Objective Lower Bound Technique

In this section, I modify the approach discussed at the end of the last section to present a polynomial-time algorithm in which the minimal objective value is not required. The modification consists of cutting well-updated lower bounds for the minimal objective value similar to Todd and Burrell’s method [62]. Unlike their method, the lower bound will be updated neither by forcing a “predictable effect” constraint nor by explicitly transforming the LP standard form.

Let $p^k$ be the projection vector from $c^T$ to the null space of $A'$ in LP2.1, i.e.,

$$p^k = (I - A'^T(A'A'^T)^{-1}A')c^T.$$

Then, in terms of the original coefficients that appeared in LP standard form,

$$p^k = \begin{pmatrix} W(c - y(z)A) \end{pmatrix}^T \frac{cx^k - z}{n + 1} e$$

(2.6)

where $y(z)$ solves the system of linear equations

$$(AW^2A^T + bb^T)y(z)^T = AW^2c^T + zb.$$ 

$y(z)$ can be further written as

$$y(z) = y_2 + zy_1$$

(2.7a)

where $y_1$ and $y_2$ satisfy

$$(AW^2A^T + bb^T)y_1^T = AWe = b$$

(2.7b)

and

$$(AW^2A^T + bb^T)y_2^T = AW^2c^T.$$ 

(2.7c)

Let

$$u^T = (cW - y_2 AW, y_2 b)$$

(2.7d)

34
\( v^T = (y_1 AW, \ 1 - y_1 b). \) \hspace{2cm} (2.7e)

Then we denote

\[
\begin{align*}
    r(z) &= \left( W(c - y(z)A) \right)^T \\
          &= u - zv,
\end{align*}
\]

and

\[
\phi(z) = \min_{1 \leq i \leq n+1} \{ r(z)_i \} \\
       = \min_{1 \leq i \leq n+1} \{ u_i - zv_i \}. \hspace{2cm} (2.9)
\]

We now derive the following lemma regarding \( \phi(z) \):

**Lemma 2.1**

\( \phi(z) \) is a concave piece-wise linear function, and there is \( Z \in R \) such that \( \phi(z) \leq 0, \forall \ z \geq Z \).

**Proof.** It is obvious that \( \phi(z) \) is a piece-wise linear and concave function from (2.9).

Note that

\[
u_{n+1} = y_2 b = b^T AW^2 c^T
\]

and

\[
v_{n+1} = 1 - y_1 b = 1 - b^T (AW^2 A^T + bb^T)^{-1} b \\
       = \frac{1}{1 + b^T (AW^2 A^T)^{-1} b} > 0.
\]

Hence, \( \forall \ z \geq Z = u_{n+1}/v_{n+1}, \)

\[
\phi(z) \leq u_{n+1} - zv_{n+1} \leq 0. \hspace{2cm} Q.E.D.
\]

Based on Lemma 2.1, we can draw \( \phi(z) \) as in Figure 2.2, where the critical point is

\[
z_1 = \min_{1 \leq i \leq n+1} \left\{ \frac{u_i}{v_i} : v_i > 0 \right\}.
\]
Figure 2.2. $\phi(z)$: Piece-Wise Linear and Concave Function

I now introduce the cutting objective lower bound technique to obtain a polynomial-time algorithm. Suppose we have obtained $0 < z^k \in X$ and $z^k \leq z^*$ at the $k^{th}$ iteration. Replacing $z$ with $z^k$ in the above expressions, we have two possibilities:

1) $\phi(z^k) \leq 0$, i.e., $\exists 1 \leq j \leq n + 1$, such that $r_j(z^k) \leq 0$. In this case,

\[
\|p^k\| = \|r(z^k) - \frac{cx^k - z^k}{n + 1} e\|
\geq |r_j(z^k) - \frac{cx^k - z^k}{n + 1}|
\geq \frac{cx^k - z^k}{n + 1}
= \frac{c' e}{n + 1}.
\]

From (1.7b) and (1.8c),

\[
c'a = c' e - \beta \|p^k\|. \tag{2.10}
\]

Therefore, inequality (2.3) holds, which enables (2.5) to hold where $z^*$ is replaced by $z^k$. Thus, by letting $z^{k+1} = z^k$, we have

\[
p(x^{k+1}, z^{k+1}) \leq p(x^k, z^k) - \alpha.
\]

2) $\phi(z^k) > 0$, i.e., $\forall 1 \leq j \leq n + 1$, $r_j(z^k) > 0$. One can see that via (2.8), $y(z^k)$ is feasible for LD and $y(z^k)b - z^k > 0$, showing $z^k$ to be strictly below $z^*$. In this case, we have
Lemma 2.2
Let $z^k \in \{z : \phi(z) \geq 0\} \neq \emptyset$. Then $z^k \leq z_1 \leq z^*$. 

Proof. From Lemma 2.1 and Figure 2.2,

$$z_1 = \sup \{z : \phi(z) \geq 0\}. \quad \text{(2.11)}$$

Hence,

$$z^k \leq z_1.$$ 

In addition,

$$\phi(z_1) = 0, \quad \text{i.e.,} \quad r(z_1) \geq 0.$$ 

From (2.8), the first $n$ inequalities imply that $y(z_1)$ is feasible for LD, and the last inequality implies that

$$z_1 \leq y(z_1)b \leq z^*.$$ 

Q.E.D.

Let $z^{k+1} = z_1$. Then, based on Lemma 2.2,

$$\|p^k\| \geq \frac{cx^k - z^{k+1}}{n + 1},$$

which leads to

$$p(x^{k+1}, z^{k+1}) \leq p(x^k, z^{k+1}) - \alpha \leq p(x^k, z^k) - \alpha.$$ 

Therefore, in both cases the potential function is reduced by a fixed amount $\alpha$. Moreover, if $z^*$ exists for LP, then $z^* \geq -2^L$ due to (0.5). Thus, we can set the initial lower bound to $z^0 = -2^L - 1$. Formally, I state the algorithm as follows.

Algorithm 2.1
Let $-2Le < x^0 \in X$, $z^0 = -2^L - 1$, and $k = 0$.

while $(cx^k - z^k) > M^{-1}$ do

begin

$W = \text{diag}(x^k)$;

end
obtain \(y_1, y_2, u\) and \(v\) via (2.7b)–(2.7e);

1° \quad \text{if } \phi(z^k) > 0 \text{ then}

\begin{align*}
  z^{k+1} &= z_1 \quad \text{else} \\
  z^{k+1} &= z^k;
\end{align*}

\begin{align*}
  p^k &= u - z^{k+1}v - \frac{c x_k - z^{k+1}}{n+1} e; \\
  a &= e - \frac{\beta p^k}{\|p^k\|}, \text{ where } \beta = 0.27–0.36; \\
  x^{k+1} &= T^{-1}(a); \\
  k &= k + 1;
\end{align*}

end.

The steps from 1° to 2° are the lower bound updating scheme. If \(z^0 = z^*\) or \(\text{LP}\) is unbounded, then \(z^0\) will never be updated since \(\phi(z^0) \leq 0\) for all \(k\). Overall, the following theorem can be proved.

**Theorem 2.1**

In \(O(Ln)\) iterations of Algorithm 2.1

\[ cx^k - z^k \leq M^{-1}, \quad \text{where } z^k \leq z^* \leq cx^k. \]

**Proof.** The proof directly follows from the discussion on pages 30–31 and the inequality

\[ p(x^{k+1}, z^{k+1}) \leq p(x^k, z^k) - \alpha \quad \text{for} \quad k = 1, 2, \ldots \]

where \(\alpha \geq 0.2\).

Q.E.D.

The following refined version of Algorithm 2.1, which appears to be simpler, is introduced by using the formula

\[ (Q + bb^T)^{-1} = Q^{-1} - \frac{(Q^{-1}b)(Q^{-1}b)^T}{1 + b^T Q^{-1} b} \]

and by noting

\[ AWc = Ax^k = b. \]
Refined Algorithm 2.1

At the $k^{th}$ step do

begin

\[ W = \text{diag}(x^k); \]

let $y'$ and $y''$ solve

\[ AW^2 A^T y'^T = AW e \quad \text{and} \quad AW^2 A^T y''^T = AW^2 c^T; \quad (2.12) \]

\[ y_1 = y'/ (1 + \omega_1) \quad \text{and} \quad y_2 = y'' - \omega_2 y_1, \] where $\omega_1 = y' b$ and $\omega_2 = y'' b$;

\[ p^k = u - z^{k+1} v - \frac{s x_k - z_k}{n+1} e; \]

the rest are the same as in Algorithm 2.1;

end.

According to (2.12), $y'$ and $y''$ always exist, so Theorem 2.1 holds whether $AW^2 A^T$ is singular or not. Nevertheless, if $AW^2 A^T$ is ill-conditioned, numerical problems may result. One of the advantages of the refined Algorithm 2.1 is that we solve the lower-dimension least squares problems at each step. Another advantage is that, in practice, $A$ is usually sparse and $b$ is usually dense so that $AW^2 A^T$ is much sparser than $AW^2 A^T + bb^T$. Hence, in solving (2.12), we can take advantage of the sparseness to make computations more efficient.

Table 2.1 on page 40 shows three computational experiments for Algorithm 2.1, where $z^0 = z^*$, $-10^6$, and $-10^9$, respectively. Even though the three solution times are at the same level, the initial setting of $z^0$ still appears sensitive to the convergence speed. When $z^k = z^0 = z^*$, the number of iterations is 13. When $z^0 = -10^6$ and $z^0 = -10^9$, the numbers of iterations increase to 21 and 28, respectively. In the latter two cases, the lower bound $(z^k)$ did not update during the first several iterations, and the primal objective value $(c x^k)$ went up. In order to further speed up the algorithm and to obtain the optimal dual solution, I present the cutting dual-objective technique, also called the new primal-dual method in the next section.
Table 2.1 Cutting Objective Lower-Bound Technique for TOY1
(optimal objective value $z^* = -90.15951$)

<table>
<thead>
<tr>
<th>iteration $(k)$</th>
<th>$z^k = z^0 = z^*$</th>
<th>$z^0 = -10^6$</th>
<th>$z^k = -10^9$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$cx^k$</td>
<td>$cx^k$</td>
<td>$cx^k$</td>
</tr>
<tr>
<td>0</td>
<td>-42.28979</td>
<td>-42.28979</td>
<td>-42.28979</td>
</tr>
<tr>
<td>1</td>
<td>-69.91632</td>
<td>-33.53101</td>
<td>-33.52689</td>
</tr>
<tr>
<td>2</td>
<td>-80.59710</td>
<td>-9.358616</td>
<td>-9.359189</td>
</tr>
<tr>
<td>3</td>
<td>-85.58926</td>
<td>95.58571</td>
<td>95.60847</td>
</tr>
<tr>
<td>4</td>
<td>-89.08782</td>
<td>419.8374</td>
<td>420.0740</td>
</tr>
<tr>
<td>5</td>
<td>-89.61885</td>
<td>140.7111</td>
<td>141.7355</td>
</tr>
<tr>
<td>6</td>
<td>-90.01867</td>
<td>40.68949</td>
<td>89.92520</td>
</tr>
<tr>
<td>7</td>
<td>-90.08872</td>
<td>8.935287</td>
<td>275.2121</td>
</tr>
<tr>
<td>8</td>
<td>-90.14657</td>
<td>-28.34679</td>
<td>113.2207</td>
</tr>
<tr>
<td>9</td>
<td>-90.15481</td>
<td>-59.16423</td>
<td>5805.871</td>
</tr>
<tr>
<td>10</td>
<td>-90.15889</td>
<td>-75.65789</td>
<td>2522.758</td>
</tr>
<tr>
<td>11</td>
<td>-90.15927</td>
<td>-85.72628</td>
<td>143.8895</td>
</tr>
<tr>
<td>12</td>
<td>-90.15943</td>
<td>-87.97033</td>
<td>69.28692</td>
</tr>
<tr>
<td>13</td>
<td>-90.15949</td>
<td>-89.24955</td>
<td>-15.49746</td>
</tr>
<tr>
<td>14</td>
<td>-90.15949</td>
<td>-89.75756</td>
<td>0.099997</td>
</tr>
<tr>
<td>15</td>
<td>-89.75756</td>
<td>-90.01867</td>
<td>-49.5138</td>
</tr>
<tr>
<td>16</td>
<td>-90.10005</td>
<td>-90.01709</td>
<td>-54.38325</td>
</tr>
<tr>
<td>17</td>
<td>-90.14409</td>
<td>-90.16386</td>
<td>-75.19707</td>
</tr>
<tr>
<td>18</td>
<td>-90.15541</td>
<td>-90.15977</td>
<td>-84.13189</td>
</tr>
<tr>
<td>19</td>
<td>-90.15874</td>
<td>-90.15953</td>
<td>-87.98166</td>
</tr>
<tr>
<td>20</td>
<td>-90.15920</td>
<td>-90.15951</td>
<td>-88.9664</td>
</tr>
<tr>
<td>21</td>
<td>-90.15944</td>
<td>-90.15951</td>
<td>-89.72953</td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td>-89.92754</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td>-90.09386</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td>-90.15391</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td>-90.15803</td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td>-90.15017</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td>-90.15942</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td>-90.15951</td>
</tr>
</tbody>
</table>
2.4 Cutting Dual-Objective Value Technique:

A New Primal-Dual Algorithm for LP

There are two drawbacks to Algorithm 2.1: the initial setting of the lower bound appears to be sensitive to the convergence speed in some problems, and, like Karmarkar’s algorithm, the optimal dual feasible solution has not been generated. I will show how to generate simultaneously a lower bound closer to the optimal objective value to achieve further efficiency, and how to recover the optimal dual feasible solution in the algorithm.

In Algorithm 2.1, we see that the lower bound \( z^k \) is updated if and only if \( \phi(z^k) > 0 \).

![Figure 2.3](image)

**Figure 2.3.** Case of \( \phi(z^k) < 0 \) but \( \{ z : \phi(z) \geq 0 \} \neq \emptyset \)

Hence, in the case of Figure 2.3, \( z^k \) will not be updated, i.e., \( z^{k+1} = z^k \). While cutting out \( z^{k+1} \) preserves polynomial complexity, \( z^{k+1} \) is underestimated. This is because, based on Lemma 2.2, \( \phi(z_1) = 0 \) and \( z_1 \leq z^* \) whenever \( \{ z : \phi(z) \geq 0 \} \neq \emptyset \), which is a weaker condition than \( \phi(z^k) > 0 \). If we let \( z^{k+1} = z_1 \) in this case, then \( z^k < z^{k+1} < z^* \) and \( \phi(z^{k+1}) = 0 \). Therefore, cutting out this greater \( z^{k+1} \) also preserves the polynomial complexity and brings \( z^{k+1} \) closer to \( z^* \). Thus, we can modify the lower bound updating criterion in Algorithm 2.1 from \( \phi(z^k) > 0 \) to \( \{ z : \phi(z) \geq 0 \} \neq \emptyset \) and \( z^k < z_1 \). To see if \( \{ z : \phi(z) \geq 0 \} \neq \emptyset \), one can check to see if \( \max_{z \in \mathbb{R}} \phi(z) \geq 0 \) in \( O(n) \) arithmetic operations from Lemma 2.1. Since \( \phi(z^k) > 0 \) implies that \( \{ z : \phi(z) \geq 0 \} \neq \emptyset \) and \( z^k < z_1 \), the lower bound \( (z_1) \) generated under the new criterion is at least as tight as the lower bound generated under the former criterion. Consequently, the modification will make the algorithm converge faster.
The remaining questions are: how can we recover the optimal dual solution if LD is feasible; and how can we find a lower bound even closer than $z_1$ from the new criterion?

So far we have updated the lower bound of the objective function without using dual feasibility. Actually, if LP is unbounded, then no feasible solution exists for LD. $z^k$, in the above approaches, does not associate with any dual variable; it is the pure lower bound of the objective function in LP standard form. As long as $\phi(z^k) \leq 0$, then $z^k$ is a valid "cut" to maintain polynomial complexity. However, at each updating of $z^k$, we can certainly construct a dual solution like $y(z_1)$ via (2.1a). Not only is $y(z_1)$ feasible for LD, but $y(z_1)b$ is also greater than $z_1$ due to the proof of Lemma 2.2. Thus, we can simultaneously update the primal and dual feasible solutions when we raise the lower bound. The initial dual solution $y^0$ is symbolically set to be infeasible. When we write $y^k = y^0$, we mean that $y^k$ remains infeasible. Therefore, we can modify the steps between 1° and 2° in Algorithm 2.1 with the following procedure:

Modification 2.1: Cutting dual-objective value

$1°$ if $\{z: \phi(z) \geq 0\} \neq \emptyset$ and $z^k < y(z_1)b$ then

$y^{k+1} = y(z_1)$ and $z^{k+1} = y(z_1)b$ else

$2°$ $y^{k+1} = y^k$ and $z^{k+1} = z^k$.

Lemma 2.3

At each update of $z^k$ in Modification 2.1, $y(z_1)$ is feasible for LD, $y(z_1)b \geq z_1$, and $\phi(y(z_1)b) \leq 0$.

Proof. It is a direct result from $\phi(z_1) = 0$ that $y(z_1)$ is feasible for LD and $y(z_1)b \geq z_1$ (see proof of Lemma 2.2). Then, noting (2.11),

$$z_1 = \sup\{z: \phi(z) \geq 0\},$$

which implies

$$\phi(y(z_1)b) \leq 0.$$

Q.E.D.
Due to Lemma 2.3, we can derive

**Theorem 2.2**

In $O(Ln)$ iterations of Modification 2.1, either

$$cx^k - z^0 \leq M^{-1},$$

and $y^k$ is infeasible for LD or

$$cx^k - y^k b \leq M^{-1},$$

and $y^k$ is feasible for LD.

**Proof.** If the algorithm never updates the lower bound, then $\phi(z^0) \leq 0$ for all $k$, and $c x^k$ converges to $z^0$ in $O(Ln)$ iterations from Theorem 2.1. In this case, $y^k = y^0$ for all $k$, i.e., $y^k$ is infeasible for LD.

If the update criterion is first satisfied at the $K^{th}$ iteration, then $\forall k > K$, $y^k$ is feasible for LD and

$$z^* \geq y^k b = z^k \geq z^{k-1}.$$ 

Therefore, in $O(Ln)$ iterations,

$$(cx^k - z^k) \leq M^{-1}$$

implies that

$$(cx^k - y^k b) \leq M^{-1}.$$  

$Q.E.D.$

Table 2.2 on page 44 provides computational results based on this modification. Even though $z^0 = -10^{30}$, the number of iterations in Table 2.2 remains at $15$—much lower than the number of iterations in Table 2.1. Therefore, the initial setting of $z^0$ is no longer sensitive to the convergence speed.
Table 2.2 Cutting Dual-Objective Value Technique for TOY1
(optimal objective value \( z^* = -90.15951 \))

<table>
<thead>
<tr>
<th>iteration ((k))</th>
<th>( z^0 = -10^9 )</th>
<th>( z^0 = -10^{30} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( cx^k )</td>
<td>( z^k(y^k b) )</td>
<td>( cx^k )</td>
</tr>
<tr>
<td>0</td>
<td>-42.28979</td>
<td>-42.28979</td>
</tr>
<tr>
<td>1</td>
<td>-33.53101</td>
<td>-33.52689</td>
</tr>
<tr>
<td>2</td>
<td>-55.16278</td>
<td>-55.15740</td>
</tr>
<tr>
<td>3</td>
<td>-66.38482</td>
<td>-66.38009</td>
</tr>
<tr>
<td>4</td>
<td>-82.09950</td>
<td>-82.09729</td>
</tr>
<tr>
<td>5</td>
<td>-86.61590</td>
<td>-86.61603</td>
</tr>
<tr>
<td>6</td>
<td>-88.58283</td>
<td>-88.58340</td>
</tr>
<tr>
<td>7</td>
<td>-89.39706</td>
<td>-89.39661</td>
</tr>
<tr>
<td>8</td>
<td>-89.88962</td>
<td>-89.88951</td>
</tr>
<tr>
<td>9</td>
<td>-90.04104</td>
<td>-90.04129</td>
</tr>
<tr>
<td>10</td>
<td>-90.12323</td>
<td>-90.12325</td>
</tr>
<tr>
<td>11</td>
<td>-90.14952</td>
<td>-90.14951</td>
</tr>
<tr>
<td>12</td>
<td>-90.15737</td>
<td>-90.15763</td>
</tr>
<tr>
<td>13</td>
<td>-90.15889</td>
<td>-90.15889</td>
</tr>
<tr>
<td>14</td>
<td>-90.15935</td>
<td>-90.15935</td>
</tr>
<tr>
<td>15</td>
<td>-90.15947</td>
<td>-90.15947</td>
</tr>
</tbody>
</table>

If \( \{ z : \phi(z) \geq 0 \} = \emptyset \) in all of the iterations, then \( z^0 \) will never be updated.

This essentially corresponds to the case of an unbounded primal objective function, i.e., LD does not have a feasible solution, and \( \{ z : \phi(z) \geq 0 \} = \emptyset \). Thus, the algorithm drives \( cx^k \) to \( z^0 \), showing that the primal objective is unbounded. Practically, we can check to see if the \((n+1)^{th}\) component of \( p^k \) blocks the feasible solution. If it does, and \( cx^{k+1} < 0 \), then the primal objective is unbounded.

Furthermore, in Phase 1, Karmarkar’s algorithm checks primal infeasibility by evaluating the reduction of the potential function [35]. This criterion is valid only if the step size, \( \beta \), is less than 0.36. However, we usually take large-sized steps in the practical approach. To overcome this difficulty, we can check for infeasibility by seeing whether \( z^0 \) is updated, where \( z^0 = 0 \) in Phase 1. If \( z^0 \) is updated, then 0 is not the optimal objective value for Phase 1, so that the primal program is infeasible. This criterion is used in place of computing the logarithmic potential function and is valid for any step size.
2.5 Cutting Primal-Objective Value: A Heuristic Approach

So far, in Algorithm 2.1 and Modification 2.1, we have cut out $z^k (\leq z^*)$ from the objective function. In these algorithms, the objective function of LP does not monotonically decrease in contrast to Algorithm 1.1. In this section, I present the cutting primal-objective value method in which I fix the "cut" at the $k^{th}$ iteration as the current primal objective value, $cx^k$. The significant difference is that $cx^k$ is an upper bound of the optimal objective value. The proof for the convergence of the cutting primal-objective method is almost identical to the proof for Algorithm 1.1 in Chapter 1. Therefore, I focus on its local convergence ratio, which is proved to be $(1 - \frac{1}{\sqrt{n}})$, i.e., it asymptotically reduces the complexity by the factor $\sqrt{n}$. Accordingly, the objective function of LP monotonically decreases in this method. Although it is not a polynomial-time algorithm, the cutting primal-objective method is worthy of presentation since it performs quite well in most experimental problems and it involves less computational work.

Modification 2.2: Cutting primal-objective value

1° $z^{k+1} = cx^k$;

2°

By substituting $z$ with $cx^k$ in (2.6), $p^k$ is simplified as

$$p^k = \begin{pmatrix} (D(c - y(z)A)^T) \\ y(z)b - z \end{pmatrix},$$

(2.13)

and $c'e = 0$. Thus, (1.10) becomes

$$c'a = -\beta\|p^k\|.$$ 

With the inverse projective transformation $T^{-1}$,

$$(cx^{k+1} - cx^k)a_{n+1} = -\beta\|p^k\|$$

or

$$cx^{k+1} = cx^k - \frac{\beta\|p^k\|}{a_{n+1}}.$$  

(2.14)
Therefore, the objective monotonically decreases, and we can derive the following
corollary, which is similar to Theorem 1.2, to evaluate the local convergence ratio
in Modification 2.2.

**Corollary 2.1**

Let \( y(z^{k+1}) \) be feasible for LD. Then

\[
\begin{align*}
    cx^{k+1} - z^* &\leq (1 - \frac{2\beta}{(1 + \beta)\sqrt{n + 1}})(cx^k - z^*).
\end{align*}
\]

**Proof.** Since \( y(z^{k+1}) \) is feasible for LD, via (2.13)

\[
p^k[n] \geq 0,
\]

and

\[
p^k_{n+1} = y(z^{k+1})b - cx^k < 0.
\]

Using Hölder's inequality,

\[
\|p^k\| \geq \frac{1}{\sqrt{n + 1}} \left( \sum_{i=1}^{n+1} |p_i^k| \right)
\]

\[
= \frac{1}{\sqrt{n + 1}} \left( \sum_{i=1}^{n} (p_i^k - p_{n+1}^k) \right)
\]

\[
= \frac{2}{\sqrt{n + 1}}(cx^k - y(z^{k+1})b)
\]

\[
\geq \frac{2}{\sqrt{n + 1}}(cx^k - z^*).
\]

Due to (2.14) and \( 1 - \beta \leq a_{n+1} \leq 1 + \beta, \)

\[
\begin{align*}
    cx^{k+1} - z^* &\leq cx^k - z^* - \frac{\beta}{1 + \beta}\|p^k\|
\end{align*}
\]

\[
\leq (1 - \frac{2\beta}{(1 + \beta)\sqrt{n + 1}})(cx^k - z^*). \quad Q.E.D.
\]
Since \( y(z^{k+1}) \) converges to a dual feasible solution, Corollary 3.1 shows that the asymptotic convergence ratio is \( (1 - \frac{1}{\sqrt{n}}) \). Recall that the convergence ratio of Algorithms 2.1 is \( (1 - \frac{1}{n}) \); the cutting primal-objective method will reduce the complexity by the factor \( \sqrt{n} \) asymptotically. Corollary 2.1 offers the possibility of shifting between the cutting primal-objective value and the cutting dual-objective value method. In other words, we can control the “cut” to take advantage of both the global convergence ratio of the cutting dual method and the local efficiency of the cutting primal method. Table 2.3 presents the computational results using the cutting primal method for the same problem that is shown in Table 2.2. As we can see, they generally converge at the same speed.

### Table 2.3 Cutting Primal-Objective Value Technique for TOY1

(optimal objective value \( z^* = -90.15951 \))

<table>
<thead>
<tr>
<th>iteration ( k )</th>
<th>( z^k = cx_{k-1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-42.28979</td>
</tr>
<tr>
<td>1</td>
<td>-69.76170</td>
</tr>
<tr>
<td>2</td>
<td>-81.16894</td>
</tr>
<tr>
<td>3</td>
<td>-84.63674</td>
</tr>
<tr>
<td>4</td>
<td>-87.77828</td>
</tr>
<tr>
<td>5</td>
<td>-89.28146</td>
</tr>
<tr>
<td>6</td>
<td>-89.79235</td>
</tr>
<tr>
<td>7</td>
<td>-90.03871</td>
</tr>
<tr>
<td>8</td>
<td>-90.10666</td>
</tr>
<tr>
<td>9</td>
<td>-90.13373</td>
</tr>
<tr>
<td>10</td>
<td>-90.15529</td>
</tr>
<tr>
<td>11</td>
<td>-90.15687</td>
</tr>
<tr>
<td>12</td>
<td>-90.15875</td>
</tr>
<tr>
<td>13</td>
<td>-90.15945</td>
</tr>
</tbody>
</table>

### 2.6 Computational Results

The computational results that appear in Table 2.4 on page 48 summarize the numerical experiments in using the three major cutting-objective methods:
Algorithm 2.1 with \( z^k = z^* \), Algorithm 2.1 with Modification 2.1, and Algorithm 2.1 with Modification 2.2. The problems are solved by the same Phase 1 procedure. The stopping tolerance is

\[
\frac{cx^k - y^k b}{1 + |cx^k|} < 10^{-6}
\]

for the cutting dual method. The step size, \( \beta \), was chosen to be 0.9.

Table 2.4 Solution Iterations of Three Cutting Objective Techniques

<table>
<thead>
<tr>
<th>Problem name</th>
<th>Size ( m \times n )</th>
<th>( z^* )</th>
<th>Total Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAIR</td>
<td>5 \times 11</td>
<td>-3905</td>
<td>( z^k = y^k b )</td>
</tr>
<tr>
<td>EXP1</td>
<td>10 \times 17</td>
<td>-18354</td>
<td></td>
</tr>
<tr>
<td>TIRE</td>
<td>12 \times 24</td>
<td>19173.3</td>
<td></td>
</tr>
<tr>
<td>AFIRO</td>
<td>27 \times 51</td>
<td>-4.647531</td>
<td></td>
</tr>
<tr>
<td>TOY1</td>
<td>47 \times 80</td>
<td>-90.15951</td>
<td></td>
</tr>
<tr>
<td>TOY2</td>
<td>47 \times 80</td>
<td>-90.07216</td>
<td></td>
</tr>
<tr>
<td>ASSI</td>
<td>100 \times 2500</td>
<td>-2675</td>
<td>( z^k = z^0 = z^* )</td>
</tr>
</tbody>
</table>

BLAIR and EXP1 problems require many pivots using the simplex method, but they are solved by the PTIE method in a small number of iterations. The AFIRO problem is degenerate, and the computation is unstable: at the above tolerance the primal feasibility is violated at the final stages. This is because in AFIRO I use the norm equation to solve least squares, which is unstable if \( A^TA \) is ill-conditioned. The ASSI problem is highly degenerate. However, in contrast to AFIRO, the solution for ASSI is stable since the code for solving least squares is numerically stable, as is shown in Table 2.5 on page 49. By contrast, the simplex method solves AFIRO using only 6 pivots. TOY1 and TOY2 have the same structure but with slightly different objective coefficients. By using the simplex method, TOY1 requires 56 pivots and TOY2 requires 107 pivots. They require the same number of iterations using the PTIE method.
Table 2.5 Cutting Dual-Objective Value Technique for ASSI

(optimai objective value $z^* = -2675$)

<table>
<thead>
<tr>
<th>iteration $(k)$</th>
<th>$z^0 = -10^{30}$</th>
<th>$cx^k$</th>
<th>$y^kb$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1496.720001</td>
<td>-1.0E30</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-1695.918526</td>
<td>-3501.440000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-2005.349575</td>
<td>-3167.012842</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-2104.850021</td>
<td>-2794.898947</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-2397.273718</td>
<td>-2721.291142</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-2407.309533</td>
<td>-2695.818242</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-2526.212707</td>
<td>-2683.018124</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-2572.993991</td>
<td>-2677.933121</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-2635.087018</td>
<td>-2677.118992</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-2650.194006</td>
<td>-2675.769222</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-2671.755077</td>
<td>-2675.115275</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>-2672.858629</td>
<td>-2675.012196</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>-2674.412174</td>
<td>-2675.003201</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>-2674.853429</td>
<td>-2675.000193</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>-2674.958109</td>
<td>-2675.000022</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>-2674.991845</td>
<td>-2675.000001</td>
<td></td>
</tr>
</tbody>
</table>

Overall, the computational results can be summarized as follows:

1. The convergence speed of the PTIE method is insensitive to the size of the optimization problem.

2. In the PTIE method prior knowledge of the optimal objective value is no longer significant.

3. The data structure has less effect on the convergence of the PTIE method than on the convergence of the simplex method.

4. A numerical problem may result when the code for solving least squares is unstable.
2.7 Relation to the Ellipsoid Method

As we briefly discussed in Chapter 0, the ellipsoid method (the first polynomial-time algorithm for LP) behaves similarly to the worst case complexity bound in practice, and its value remains theoretical. However, our computational results show that the polynomial variant of Karmarkar’s algorithm behaves very well in practice. The questions arise: what is the significant difference between Karmarkar’s algorithm and the ellipsoid method? Are these two algorithms totally unrelated?

The answer to the second question is “no.” The relation between the ellipsoid method and Karmarkar’s algorithm was first studied by Todd. In a recent paper, he has shown how to obtain an ellipsoid that contains all the optimal dual solutions from Karmarkar’s algorithm [61]. He proved that under certain conditions the volume of this ellipsoid shrinks to zero as the primal-dual “gap” converges to zero. The volume determinant of the ellipsoid proposed in his approach is apparently dependent on the nondegeneracy of the optimal primal solution. Unlike the ellipsoid method, the volume does not necessarily shrink monotonically.

In this section, based on Todd’s analysis, I propose a new ellipsoid in Algorithm 2.1. The new ellipsoid is contained in the complementary dual-slack space and centered at the current dual-slack solution, and it contains all the optimal dual feasible solutions. Surprisingly, the “magic” potential function, which is used to measure polynomial convergence of Algorithm 2.1, turns out to correctly represent the logarithmic volume of this ellipsoid. Therefore, as the potential function monotonically declines, the volume of the ellipsoid uniformly shrinks to zero as it does in the ellipsoid method. Since the volume of the new ellipsoid is uniquely characterized by the potential function, it shrinks to zero whether or not the optimal primal solution is degenerate.

The Ellipsoid Method

The ellipsoid method was originally used for solving problems of linear inequalities (LI), i.e., either finding \( y \in R^m \) for the system of inequalities

\[ yA \leq c, \]

50
or reporting that the system is inconsistent. Let \( R(A) \) be the row space of \( A \), i.e.,

\[
R(A) = \{ x \in \mathbb{R}^n : x = yA, y \in \mathbb{R}^m \}
\]

that is a subspace of \( \mathbb{R}^n \), and let

\[
C(c) = \{ x \in \mathbb{R}^n : x \leq c \}
\]

that is a cone pointed at \( c \). Then the geometric interpretation of LI is either to find

a point in \( R(A) \cap C(c) \) or to detect \( R(A) \cap C(c) = \emptyset \).

Let \( Q \in \mathbb{R}^{n \times n} \) be a positive definite matrix, and the ellipsoid be defined by

\[
E = \{ u \in R(A) : (u - \bar{u})Q(u - \bar{u})^T \leq 1 \},
\]

which is embedded in an \( n \) dimensional ellipsoid centered at \( \bar{u} \in R(A) \). Then the

ellipsoid method starts with a worst case of \( Q^0 \) and \( \bar{u}^0 \), and generates a sequence

of ellipsoids \( \{E^k\} \)

\[
E^k = \{ u \in R(A) : (u - \bar{u}^k)Q^k(u - \bar{u}^k)^T \leq 1 \},
\]

which centers at \( \bar{u}^k \in R(A) \) and contains \( R(A) \cap C(c) \). The volume of \( E^k \) reduces

globally at a fixed ratio. More precisely,

\[
\frac{V(E^{k+1})}{V(E^k)} \leq O(2^{-1/2n}). \quad (2.15)
\]

Therefore, after enough iterations either we must discoversolution, or else we

must be certain that through successive shrinkings the ellipsoid has become too

small to contain \( R(A) \cap C(c) \) and conclude \( R(A) \cap C(c) = \emptyset \). Each iteration of the

ellipsoid method requires \( O(Ln^2) \) arithmetic operations. One can analyze that the

overall complexity is \( O(L^2n^4) \).

The ellipsoid method can be used naturally for solving LD by finding a feas-

ible solution \( y \) for the system of linear inequalities

\[
yA \leq c \quad \text{and} \quad yb \geq z^*,
\]

51
$$yA \leq \bar{c}(z^*)$$,

or by finding $u \in \mathbb{R}^{n+1}$ such that

$$u \in R(\bar{A}) \cap C(\bar{c}(z^*))$$ \hspace{1cm} (2.16)

where

$$\bar{A} = (A, -b) \quad \text{and} \quad \bar{c}(z^*) = (c, -z^*)$$.

We now derive the dual ellipsoid. Let

$$\bar{D} = \begin{pmatrix} \text{diag}(x^k) & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} W & 0 \\ 0 & 1 \end{pmatrix}.$$ \hspace{1cm} (2.16)

Then for all optimal dual feasible solution $y^*$ of LD, we must have

$$y^*\bar{A}\bar{D} \leq \bar{c}(z^*)\bar{D} \leq \bar{c}(z^{k+1})\bar{D}.$$ \hspace{1cm} (2.17)

(2.17) further implies that

$$\|(\bar{c}(z^{k+1}) - y^*\bar{A})\bar{D}\|^2 \leq ((\bar{c}(z^{k+1}) - y^*\bar{A})\bar{D}e)^2 = (cx^k - z^{k+1})^2.$$ \hspace{1cm} (2.18)

Moreover, via (2.8), $r(z^{k+1}) \in N(\bar{A}\bar{D})$ where $N(\bar{A}\bar{D})$ stands for the null space of $\bar{A}\bar{D}$. Thus,

$$\|(\bar{c}(z^{k+1}) - y^*\bar{A})\bar{D}\|^2 = \|(\bar{c}(z^{k+1}) - y(z^{k+1})\bar{A})\bar{D} + (y(z^{k+1})\bar{A} - y^*\bar{A})\bar{D}\|^2$$

$$= \|r(z^{k+1}) + (y(z^{k+1}) - y^*)\bar{A}\bar{D}\|^2$$

$$= \|r(z^{k+1})\|^2 + \|(y(z^{k+1}) - y^*)\bar{A}\bar{D}\|^2.$$ \hspace{1cm} (2.19)

Furthermore, let

$$\epsilon = \sqrt{(cx^k - z^{k+1})^2 - \|r(z^{k+1})\|^2}$$

$$\epsilon^k = cx^k - z^k;$$
then (2.18) and (2.19) imply that
\[ \|(y(z^{k+1}) - y^*)\bar{A}\bar{D}\|^2 = \|(\bar{c}(z^{k+1}) - y^*\bar{A})\bar{D}\|^2 - \|r(z^{k+1})\|^2 \leq (ex^k - z^{k+1})^2 - \|r(z^{k+1})\|^2 = \epsilon^2 \]

\[ \|(y(z^{k+1})\bar{A} - y^*\bar{A})\bar{D}\| \leq \epsilon \leq \epsilon^k. \quad (2.20) \]

Let \( \bar{u}^k = y(z^{k+1})\bar{A} \), and \( S^k \) be the ellipsoid
\[ S^k = \{ u \in R^{n+1} : \|(u - \bar{u}^k)\bar{D}\| \leq \epsilon^k \}. \]

Then we must have

**Lemma 2.4**
\[ R(\bar{A}) \cap C(\bar{c}(z^*)) \subset S^k. \]

**Proof.** Since (2.17) holds for all \( y^* \) that are the optimal feasible solutions for LD, \( S^k \) contains all \( y^*\bar{A} \in R(\bar{A}) \) such that \( y^*\bar{A} \leq \bar{c}(z^*) \). This proves
\[ R(\bar{A}) \cap C(\bar{c}) \subset S^k. \]

**Q.E.D.**

Interestingly, the volume of \( S^k \) is
\[ V(S^k) = \pi(\epsilon^k)^{n+1} / \det(\bar{D}) = \pi(e^{x^k} - z^k)^{n+1} / \prod_{i=1}^{n} x_i^k. \]
In other words,
\[ \ln V(S^k) = P(x^k, z^k) + \ln \pi, \quad (2.21) \]

i.e., the potential function value correctly represents the logarithmic volume of the ellipsoid \( S^k \). Using (2.1) and Lemma 2.4, we directly derive
Theorem 2.3

Let

\[ V(S^0) = \frac{\pi (cx^0 - z^0)^n}{\prod_{i=1}^{n} x_i^0} (\leq O(2^{Ln})) \]

Then, for all \( k \) in Algorithm 2.1,

\[ R(\overline{A}) \cap C(\overline{z}^*) \subset S^k \]

and

\[ \frac{V(S^{k+1})}{V(S^k)} \leq \exp(-\alpha) \leq O(2^{-1}) \]

The theorem indicates that \( S^k \) contains all the optimal dual-slack solutions \((y^*, A)\) and the volume of \( S^k \) shrinks at the ratio \( O(2^{-1}) \), compared to the shrinking ratio \( O(2^{-1/n}) \) for the ellipsoid method given in (2.15). We can also see other minor differences between the ellipsoid method and Algorithm 2.1:

1) The initial volume of the ellipsoid method has to take the worst case bound \( n^2 2^{Ln} \), but the initial volume of Karmarkar’s algorithm is naturally bounded by the initial potential function value, which is much less than \( n^2 2^{Ln} \).

2) The moving step in the ellipsoid method is limited to a certain size, therefore, the theoretical shrinking ratio is strictly true for the ellipsoid method. However, a larger step size \((\beta)\) in Algorithm 2.1 can be taken by incorporating a line search technique to minimize the potential function (hence, the volume of the dual ellipsoid), resulting in much greater volume reduction than the theoretical shrinking ratio.

Practically, these dual ellipsoids can be used to determine the optimal non-basic variables. Let the optimal dual-slacks be defined as

\[ s^* = (c - y^* A, y^* b - z^*) \]

and let

\[ s^k = (c - y(z^{k+1}) A, y(z^{k+1}) b - z^*) \]
Obviously,
\[ s^* - s^k = (y(s^{k+1}) - y^*)\overline{A} \in R(\overline{A}). \] (2.22)

Hence, (2.20) induces a dual-slack ellipsoid
\[ \{ s : \| (s - s^k)\overline{D} \| \leq \epsilon \} \] (2.23)

that contains all the optimal dual slacks \( s^* \). From the complementary slackness condition,
\[ s^*_i x^*_i = 0 \quad \text{for all} \quad 1 \leq i \leq n, \]

i.e., \( s^*_i > 0 \) implies that \( x^*_i = 0 \) or \( x_i \) is an optimal nonbasic variable. Therefore, we can calculate the minimal value for \( s^*_i \) subject to constraints (2.22) and (2.23). Indeed, we can solve the following dual-slack ellipsoidal optimization problem:

\[ \text{minimize} \quad s_i \]
\[ \text{subject to} \quad s - s^k \in R(\overline{A}) \]
\[ \| (s - s^k)\overline{D} \|^2 \leq \epsilon^2. \]

The minimal values of DEO for \( 1 \leq i \leq n \) are given by the following proposition.

**Proposition 2.1**

The minimal objective value for the DEO problem is
\[ s_i^k - \frac{\epsilon}{d_i} \sqrt{q_i} \]

where \( d_i \) is the \( i \)th diagonal element of positive diagonal matrix \( \overline{D} \) and \( q_i \) is the \( i \)th diagonal element of the projection matrix
\[ \overline{D}A^T (\overline{A}^2 A^T)^{-1} \overline{A}D. \]

**Proof.** The proof directly results from the Kuhn-Tucker optimality conditions for DEO.
If the minimal value of DEO is positive, then \( s_i^* > 0 \), showing that \( x_i \) must be an optimal nonbasic variable. Surprisingly,

\[
DA^T (AD^2A^T)^{-1}A = (AW, -b)^T (AW^2A^T + bb^T)^{-1} (AW, -b),
\]
which was already obtained while solving least-squares (2.7) in Algorithm 2.1. For example, if \((AW, -b)\) is factorized by the QR method such that

\[(AW, -b)^T = QR\]

where \( Q \in R^{(n+1) \times m} \) and \( R \in R^{m \times m} \), then

\[(AW, -b)^T (AW^2A^T + bb^T)^{-1} (AW, -b) = QQ^T.\]

Thus, \( q_i \) is the L₂ norm square of the \( i \)th row of \( Q \), and the minimal value of DEO can be calculated for all \( i \) with \( O(n) \) arithmetic operations.

In case \( x^k \) has some zero elements like in the simplex method, Proposition 2.2 can be described as

**Proposition 2.2**

The minimal objective value of EOP is

\[
s_i^k = \epsilon \sqrt{q_i}
\]

where \( q_i \) is the \( i \)th diagonal element of the positive semi-definite matrix

\[
A^T (AD^2A^T)^{-1}A.
\]

Overall, we can derive the following strong column eliminating theorem for linear programming.

**Theorem 2.4**

Let \( x^k \in X, \epsilon \) be defined above, and \( q_i \) be the \( i \)th diagonal element of matrix

\[
A^T (AD^2A^T)^{-1}A.
\]
\[ s_i^k - \epsilon \sqrt{q_i} > 0, \]

then the \( i \)th column is not in any of the optimal bases for linear programming.

Note that \( s^k \) is the reduced cost vector in the simplex method; hence, the new pricing rule subtracts the second term \( \epsilon \sqrt{q_i} \) from the reduced cost vector to obtain a strong column eliminating conclusion in Theorem 2.4. Moreover, the algorithm makes \( \epsilon \) converge to zero in a polynomial number of steps. Therefore, if \( q_i \) is bounded from above for all \( 1 \leq i \leq n \), all optimal nonbasic columns will be eliminated from the candidate set in a polynomial number of steps.

2.8 Summary

In this chapter, I modified the IE method to overcome the difficulties found in other polynomial-time algorithms. Algorithm 2.1 and Modification 2.1 do not require prior knowledge of the optimal objective value, and neither of them destroys the order of polynomial complexity. In addition, they can be efficiently implemented in practice.

Using the cutting dual-objective value technique, I obtained the optimal feasible solutions for both primal and dual programs while neither doubling the size of the original LP problem as is done in Karmarkar's original algorithm [35], nor forcing a "predictable constraint" as is done in Todd and Burrell's method [62]. Mathematically, the conclusions in this chapter hold without the assumption of non-degeneracy since the least squares solutions in (2.12) always exist even though \( AW^{2}A^{T} \) is singular.

A dual ellipsoid of Algorithm 2.1 was also discussed in this chapter. The potential function, which is used to measure polynomial convergence of the primal solutions in the algorithm, correctly represents the logarithmic volume of the ellipsoid containing all the optimal dual-slack solutions. Like in the ellipsoid method, the volume of this dual-slack ellipsoid uniformly shrinks as the algorithm iterates. This resemblance leads to a strong column eliminating theorem to determine the optimal basis for linear programming.
The framework of the PTIE technique is extended to linearly constrained convex programming in the next chapter. This extension leads to a polynomial-time algorithm for convex quadratic programming.

Remark

While this dissertation has been in process, several developments have been made to the growing literature on Karmarkar’s LP algorithm. Adler, Karmarkar, Iliesene and Veiga [1] obtained a large number of computational results for Karmarkar’s algorithm. Anstreicher [2] proposed a lower bound estimation technique in Karmarkar’s LP canonical form, similar to Todd and Burrell’s work. Barnes [3], Cavalier and Soyster [9], Kortanek and Shi [39], and Vanderbei, Meketon, and Freedman [64] developed the “affine scaling” method that is identical to the IE method for solving LP problems in standard forms. Bayer and Lagarias [5], Megiddo [48], and Megiddo and Shub [49] studied the solution trajectories and boundary behavior of Karmarkar-type algorithms. The “cutting optimal-objective value” technique has been described and analyzed by Gay [22] and Lustig [46] using the potential function in the LP standard form. Gay [22] and Ghellinck and Vial [24] proposed a variant similar to Algorithm 2.1. Gill, Murray, Saunders, Wright, and Tomlin [25], Iri and Imai [33], and Kojima, Mizuno and Yoshise [38] analyzed the Newton barrier method for LP and its similarity to Karmarkar’s algorithm. Renegar [55] and Gonzaga [29] improved the solution time for LP by introducing the interior centering method. Ye and Chiu [68] showed how to recover the shadow price of LP in Karmarkar’s algorithm.
Chapter 3  PTIE Method: Quadratic and Convex Programming

3.1 Introduction

Linear programming is a subset of convex quadratic programming (QP), which arises in such varied disciplines as engineering, economics, the physical sciences, mathematics, or in any other area where decisions must be made in a complex situation. QP and its dual were intensively discussed and analyzed by Cottle and Dantzig [11][13], Eaves [16], and Murty [51]. Several significant developments for solving QP problems were introduced by Hildreth [32], Lemke [42] (implemented by Tomlin [63]), and Wolfe [66]. As far as I know, none of these approaches is a polynomial algorithm. Kozlov, Tarazov and Khachiyan [40] proved that the ellipsoid method solves the convex QP in polynomial-time. But, as we discussed in Chapter 0, the implementational result of the ellipsoid method is not as attractive as its theoretical result. Overall, an efficient polynomial-time algorithm for QP has not yet been developed and implemented.

In this chapter, I apply the projective transformation technique to CP. I show that an extension of the PTIE method solves CP in \( O(Ln) \) iterations. In addition, if \( f(x) \) is quadratic, the sub-optimization problem in each iteration can be solved in polynomial-time. Therefore, the PTIE method is a polynomial-time algorithm for QP. In Section 3.4, I extend the primal-dual algorithm (discussed in the last chapter) to solving CP, computing each iteration approximately and updating the dual objective value iteratively. Section 3.5 summarizes the computational results, which show that the behavior of the algorithm is similar to the procedure used in solving LP problems.

3.2 Convexity Invariance in Projective Transformation

In Chapter 2, we saw that the polynomial complexity of the PTIE method for linear programming is based on the inequality of (2.3), which resulted from solving
LP2.1($z^*$). This inequality makes the linear objective function reduce at the ratio of \((1 - O(\frac{1}{n}))\), and the algorithm converge in \(O(ln)\) iterations. However, this reduction should also be true even when the objective is a general convex function in LP2.1($z^*$). More specifically, let \(a\) be the minimal solution for

\[
\text{CP3.1}(z^*) \quad \begin{array}{ll}
\text{minimize} & f'(x') \\
\text{subject to} & x' \in X' \\
& \|x' - e\|^2 \beta^2 < 1
\end{array}
\]

where \(f'(x') \geq 0 \forall x' \in X'\), and \(\exists x'^* \in X'\) such that \(f'(x'^*) = 0\). Then, due to the inequality (2.2) and the convexity of \(f'(x')\),

\[
\frac{f'(a)}{f'(e)} \leq 1 - \frac{\beta}{n + 1},
\]

which resembles (2.3).

In this section, I show how to projective-transform CP into the canonical form with a new augmented convex objective function, and then how to apply the IE method to solving CP3.1($z^*$). To begin, I assume that the optimal objective value \(z^*\) is known in advance for CP (this assumption will be released later). By using the projective transformation \(T\) and the following objective augmenting technique, CP can be related to the following problem:

\[
\text{CP3}(z^*) \quad \begin{array}{ll}
\text{minimize} & f'(x') = x'_{n+1}(f(T^{-1}x') - z^*) \\
\text{subject to} & x' \in X'.
\end{array}
\]

The augmented objective function in CP3($z^*$) plays a key role in my approach. Note that \(f'(x')\) is the product of the error function \(f(x) - z^*\) multiplied by \(x'_{n+1} \geq 0\). Hence, \(f'(x') \geq 0\) in \(X'\). Generally, we have

\[
f'(x') = x'_{n+1}(f(x) - z^*),
\]

where \(x' = T(x)\). Especially,

\[
f'(e) = f(x^k) - z^*
\]
and
\[ f'(x'^*) = x'^*_{n+1}(f(x^*) - z^*) = 0, \]

where \( x'^* = T(x^*) \). If \( f(x) \) is a linear function, then \( f'(x') \) is also a linear function; otherwise, from
\[
\nabla f'(x') = (\nabla f(x)W, f(x) - z^* - \nabla f(x)x)
\]
we have
\[
\nabla f'(x')x' = f'(x'),
\]
which, interestingly, indicates that \( f'(x') \) still behaves like a linear function. However, perhaps the most important characteristic of the new objective function is due to the following lemma.

**Lemma 3.1**

Let \( f(x) \) be a convex function in \( X \). Then \( f'(x') \) is a convex function in \( X' \).

**Proof.** Let \( \zeta, \eta \geq 0, \zeta + \eta = 1, \) and \( u = T^{-1}(u') \) and \( v = T^{-1}(v') \). Then, for any \( u', v' \in X' \)
\[
f'(\zeta u' + \eta v') = (\zeta u'_{n+1} + \eta v'_{n+1})(f(\frac{W(\zeta u'[n] + \eta v'[n])}{\zeta u'_{n+1} + \eta v'_{n+1}}) - z^*)
\]
\[
= (\zeta u'_{n+1} + \eta v'_{n+1})(f(\frac{\zeta u'_{n+1}u + \eta v'_{n+1}v}{\zeta u'_{n+1} + \eta v'_{n+1}}) - z^*)
\]
\[
\leq (\zeta u'_{n+1})(f(u) - z^*) + (\eta v'_{n+1})(f(v) - z^*)
\]
\[
= \zeta f'(u') + \eta f'(v').
\]

**Q.E.D.**

Lemma 3.1 leads to a very important conclusion: the convexity of the objective function remains invariant in the objective augmentation and the projective transformation. This invariance enables us to use the PTIE method to solve CP3.1. As a result, the following algorithm is introduced:
Algorithm 3.1

while $f(x^k) - z^* > M^{-1}$ do

begin

$W = \text{diag}(x^k)$;

let $a$ be the optimal solution of CP3.1($z^*$);

$x^{k+1} = T^{-1}(a)$;

$k = k + 1$;

end.

The next lemma is used to prove a theorem for Algorithm 3.1. The lemma confirms that inequality (3.1) holds, i.e., a fixed objective reduction can be made at each iteration.

Lemma 3.2

$$\frac{f'(a)}{f'(e)} \leq (1 - \frac{\beta}{n+1}).$$

Proof. If $x'^* \in \{x' \in X' : \|x' - e\| \leq b\}$, then

$$0 \leq f'(a) \leq f'(x'^*) = 0,$$

and so Lemma 3.2 holds. Otherwise, since $X'$ is a convex polytope, the intersection point of the boundary of $\{x' \in X' : \|x' - e\| \leq b\}$ and the line segment between $e$ and $x'^*$, should be feasible for CP3.1. Let $a'$ be the intersection point; then $a'$ satisfies

$$\|a' - e\| = \beta,$$  \hspace{1cm} (3.3)

and

$$a' = \theta x'^* + (1 - \theta)e \quad \text{for some} \quad 0 < \theta < 1.$$  \hspace{1cm} (3.4)

Substituting $a'$ in (3.4) for $a'$ in (3.3),

$$\|\theta x'^* + (1 - \theta)e - e\| = \beta,$$
then
\[ \theta \| x^* - e \| = \beta. \]

Note that
\[ \| x^* - e \|^2 = \| x^* \|^2 - 2e^T x^* + \| e \|^2, \]
\[ e^T x^* = n + 1, \]
and
\[ \| x^* \|^2 \leq (e^T x^*)^2 = (n + 1)^2. \]

Hence,
\[ \theta \geq \frac{\beta}{\sqrt{n(n + 1)}} > \frac{\beta}{n + 1}. \]  \hspace{1cm} (3.5)

In addition, due to the convexity of \( f'(.) \),
\[ f'(a') \leq \theta f'(x^*) + (1 - \theta) f'(e) \]
\[ = (1 - \theta) f'(e). \]

Since \( a \) is the optimal feasible solution and \( a' \) is a feasible solution for CP3.1, we have
\[ f'(a) \leq f'(a') \leq (1 - \theta) f'(e). \]  \hspace{1cm} (3.6)

Lemma 3.2 thus follows from (3.5) and (3.6).

As a result of (2.4) and Lemma 3.2, we can derive

**Theorem 3.1**

\[ P(x^{k+1}, z^*) \leq P(x^k, z^*) - \alpha, \quad \text{where} \quad \alpha = \beta - \frac{\beta^2}{2(1 - \beta)^2}. \]

Proof. The proof directly results from substituting \( c'a \) and \( c'e \) in the derivation of (2.5).

Thus, we have the following corollary.
Corollary 3.1

In $O(Ln)$ iterations of Algorithm 3.1

$$f(x^k) - z^* \leq M^{-1}.$$ 

Proof. This proof is parallel to the proof for Theorem 2.1. Q.E.D.

3.3 A Polynomial-Time Algorithm for QP

Corollary 3.1 concludes that Algorithm 3.1 generates the optimal feasible solution in a polynomial number of iterations. To develop a polynomial-time algorithm for QP, we need to prove that CP3.1 can be solved polynomially. Since nonnegativity constraints in CP3.1 are redundant, CP3.1 shares the same form as CP1.2. In other words, we need to prove that the optimality conditions (1.5a)–(1.5d) can be computed in a polynomial of $L$ and $n$. Again, as discussed in Section 1.4, we can distinguish between two cases: in the first case, CP has a POFS; in the second, it does not. In the latter case, a unique solution must exist for CP3.1. This is because: 1) the optimal solution $x''^*$ of CP3 is not contained in the feasible region of CP3.1, 2) the objective of CP3.1 is a convex function, and 3) the feasible region of CP3.1 is a strict convex set. By the well-known separating theorem, a unique minimal solution of CP3.1 occurs at the boundary of the feasible region. In other words, there exists a unique fixed point $(a, y, \mu)$ that satisfies (1.5a)–(1.5d), where $y$ is the Lagrange multiplier vector for the linear equality constraints and $\mu$ is the nonnegative multiplier for the ellipsoid (sphere) inequality constraint.

Let $f(x)$ be the quadratic function given in (0.3). Then

$$f'(x') = \frac{x'[n]^TQ'x'[n]}{2x'_{n+1}} + c'x'[n] - z^*x'_n$$

where

$$Q' = WQW, \quad \text{and} \quad c' = cW.$$
One can see that the augmented objective function of (3.7) is almost quadratic, except that the last variable appeared in the denominator of the quadratic term. This still makes it possible to solve CP3.1 by using the multiplier $\mu > 0$ as a parameter like Equation (1.19) discussed in Section 1.4. Here I introduce two approaches: the first approach is to let $\mu$ be fixed, and then solve CP3.1 in $O(n^3)$ operations and guarantee that the radius $\beta$ of the constraining sphere is bounded by

$$O\left(\frac{1}{n+1}\right) < \beta < 0.46,$$

in which case the overall complexity is $O(Ln^5)$. The second approach is to let $\beta = 0.26-0.36$; then we solve CP3.1 by binary-searching $\mu$ in $O(Ln^3)$ operations, in which case the overall complexity is $O(L^2n^4)$. In both approaches, we count $O(n^3)$ operations for solving a system of $n$ linear equations. To be more specific, I first provide a bound for $\mu$. Using (3.7),

$$\nabla f'(a) = \left(\frac{a[n]^TQ'}{a_{n+1}} + c', \quad \frac{-a[n]^TQ'a[n]}{2(a_{n+1})^2} - z^*\right).$$

Again, we notice linear-similarity

$$\nabla f'(a)a = f'(a).$$

Multiplying both sides of (1.7b) by $\nabla f'(a)$ from the left,

$$\nabla f'(a)a = \nabla f'(a)e - \beta\|p^k\|$$

or

$$\beta\|p^k\| = \nabla f'(a)(e - a).$$

This leads to

**Lemma 3.3**

$$\frac{f'(a)}{\beta(n + 1)} \leq \mu \leq \frac{f'(e) - f'(a)}{\beta^2}.$$
Proof. Due to the convexity of \( f'(.) \) and the linear-similarity of (3.9),
\[
    f'(x') \geq f'(a) + \nabla f'(a)(x' - a) = \nabla f'(a)x' \quad \forall \ x' \in X'.
\]

In addition, the optimality conditions imply that \( a \) is also the optimal solution in minimizing the linear objective function, \( \nabla f'(a)x' \), and subjecting \( x' \) to the constraints of CP3.1. Hence, similar to the proof of Lemma 3.2,
\[
\begin{align*}
    \nabla f'(a)a &\leq \theta \nabla f'(a)x'^* + (1 - \theta)\nabla f'(a)e \\
    &\leq \theta f'(x'^*) + (1 - \theta)\nabla f'(a)e \\
    &= (1 - \theta)\nabla f'(a)e \\
    &\leq (1 - \frac{\beta}{n + 1})\nabla f'(a)e.
\end{align*}
\]

Therefore, from (3.9), (3.10), and the above inequality
\[
\frac{\beta f'(a)}{n + 1} \leq \frac{\beta\nabla f'(a)e}{n + 1} \leq \beta\|p^k\| = \nabla f'(a)(e - a) \leq f'(e) - f'(a).
\]

By combining this inequality with (1.7c) in Section 1.4, we derive the conclusion in Lemma 3.3. Q.E.D.

Now we split (1.5a) into two groups: the first one includes the first \( n \) equations of (1.5a)
\[
\frac{Q'a[n]}{a_{n+1}} + c'T - A'[n]T yT + \mu(a[n] - e) = 0,
\]
and the second one is the last equation of (1.5a)
\[
\frac{-aT[n]Q'a[n]}{2(a_{n+1})^2} - z^* - yA'_{n+1} + \mu(a_{n+1} - 1) = 0,
\]
where \( A'[n] \) is the matrix of the first \( n \) columns of \( A' \) and \( A'_{n+1} \) is the last column of \( A' \). Let
\[
y' = a_{n+1}y
\]
and
\[
\lambda = a_{n+1}\mu.
\]
Then, since \((1 - \beta) \leq a_{n+1} \leq (1 + \beta)\), Lemma 3.3 imposes a bound for \(\lambda\):
\[
0 \leq \frac{(1 - \beta)f'(a)}{\beta(n + 1)} \leq \lambda \leq \frac{(1 + \beta)(f'(e) - f'(a))}{\beta^2} \leq \lambda_{max},
\]
(3.12)
where
\[
\lambda_{max} = \frac{(1 + \beta)f'(e)}{\beta^2}.
\]
Particularly, (3.11a) and (3.11b) become
\[
(Q' + \lambda I)a[n] - A'[n]T y'T = \lambda e - a_{n+1}c'T
\]
(3.13a)
and
\[
\frac{aT[n]Q'a[n]}{2} + (\lambda - z^*)(a_{n+1})^2 - (y'A_{n+1} + \lambda)a_{n+1} = 0,
\]
(3.13b)
respectively. In addition, (1.5b) can be rewritten as
\[
A'[n]a[n] = \begin{pmatrix} a_{n+1}b \\ n + 1 - a_{n+1} \end{pmatrix}.
\]
(3.13c)
Then (3.13a) and (3.13c) form a system of linear equations similar to (1.19):
\[
P \begin{pmatrix} a[n] \\ y'T \end{pmatrix} = a_{n+1}b^1 + b^2
\]
where
\[
P = \begin{pmatrix} Q' + \lambda I & -A'[n]T \\ A'[n] & 0 \end{pmatrix}
\]
\[
b^1 = \begin{pmatrix} -c'T \\ b \\ -1 \end{pmatrix} \quad \text{and} \quad b^2 = \begin{pmatrix} \lambda e \\ 0 \\ n + 1 \end{pmatrix}.
\]
For any given \(\lambda \in [0, \lambda_{max}]\), we can compute \(P^{-1}b^1\) and \(P^{-1}b^2\), and let
\[
\begin{pmatrix} a[n] \\ y'T \end{pmatrix} = a_{n+1}P^{-1}b^1 + P^{-1}b^2.
\]
(3.14)
Then we substitute (3.14) for (3.13b) to compute \(a_{n+1}\) by solving a single quadratic equation. Thus, \(a\) will be precisely determined via (3.14). Denote \(a\) as a function of \(\lambda : a(\lambda)\). Then the following two approaches are introduced for solving CP3.1.
Approach 1

The first approach fixes $\lambda$ such that $\lambda = 7 f'(e) = 7(f(x^k) - z^*)$.

Procedure 3.1

Begin

1. Let $\lambda = 7 f'(e)$.

2. Compute $P^{-1} b^1$ and $P^{-1} b^2$ in $O(n^3)$ arithmetic operations.

3. Return $a$ from (3.13b) and (3.14).

end.

As a result of Procedure 3.1, we have

Theorem 3.2

Let $a$ be returned from Procedure 3.1, and $x^{k+1} = T^{-1}(a)$. Then

$$P(x^{k+1}, z^*) \leq P(x^k, z^*) - \mathcal{O}(\frac{1}{n+1}) \text{ for } n > 3.$$  

Proof. Due to the right inequality of (3.12),

$$7 \beta^2 \leq 1 + \beta \quad \text{i.e., } \beta < 0.46. \quad (3.15)$$

Let $\rho = \frac{f'(e)}{f'(e)}$; then due to the left inequality of (3.12),

$$\beta \geq \frac{\rho}{7(n+1) + \rho}.$$  

If $\rho \leq 1 - \frac{0.46}{n+1}$, then via (3.15) and the proof in Theorem 3.1,

$$P(x^{k+1}, z^*) \leq P(x^k, z^*) - 0.46 + \frac{0.46^2}{2(1 - 0.46)^2}$$

$$\leq P(x^k, z^*) - 0.1$$

$$\leq P(x^k, z^*) - \frac{1}{8(n+1)}$$
for $n \geq 1$. Otherwise, 

$$\rho > 1 - \frac{0.46}{n+1},$$

which implies

$$\beta \geq \frac{\rho}{\frac{1}{\ell(n+1)} + \rho}$$

$$> \frac{\rho}{\frac{1}{\ell(n+1)}}$$

$$> \frac{1}{\ell(n+1)}(1 - \frac{0.46}{n+1})$$

$$\geq \frac{1}{8(n+1)}$$

for $n > 3$. Therefore, directly from Theorem 3.1,

$$P(x^{k+1}, z^*) \leq P(x^k, z^*) - O(\beta)$$

$$\leq P(x^k, z^*) - O\left(\frac{1}{n+1}\right)$$

for $n > 3$. Q.E.D.

Thus, we can further derive

**Corollary 3.2**

In $O(Ln^2)$ iterations of Algorithm 3.1 with Procedure 3.1

$$f(x^k) - z^* \leq M^{-1}.$$ 

Proof. This proof is parallel to the proof for Theorem 2.1. Q.E.D.

Therefore, the computational complexity of Algorithm 3.1 for QP, coupled with Procedure 3.1, is $O(Ln^5)$ with $O(n^3)$ arithmetic operations in each iteration and $O(Ln^2)$ total iterations. This concludes Approach 1.
Approach 2

The second approach lets $\beta = 0.27 - 0.36$, and then uses a line-search technique for $\lambda$ until $\|a(\lambda) - e\|$ lands in the range of $[0.26, 0.36]$. Let $h(\lambda) = \|a(\lambda) - e\| - \beta$; then $h(\lambda)$ has a unique zero bounded in (3.12). In addition, let

$$\lambda_{\min} = \frac{(1 - \beta)^2 M^{-1}}{\beta(n + 1)}.$$

Obviously, $h(\infty) = \|e - e\| - \beta = -\beta < 0$, which implies $h(\lambda_{\max}) \leq 0$. On the other hand, if $h(\lambda_{\min}) \leq 0$, then we obtain a positive (interior) optimal feasible solution for QP from inequality (3.12), since

$$f(T^{-1}(a(\lambda_{\min}))) \leq M^{-1}.$$

Generally, we expect $h(\lambda_{\min}) > 0$. Overall, the above process can be implemented by applying the well-known bisection technique to determine the zero $\lambda^*$ of $h(\lambda)$ in the interval $[\lambda_{\min}, \lambda_{\max}]$, as shown in the following procedure:

**Procedure 3.2**

*Begin*

1. Set $\beta = 0.31$, $\lambda_1 = \lambda_{\min}$, $\lambda_3 = \lambda_{\max}$, and $\lambda_2 = (\lambda_1 + \lambda_3)/2$.

2. Let $\lambda = \lambda_2$; then compute two vectors $P^{-1} b^1$ and $P^{-1} b^2$.

3. Determine $a$ from (3.13b) and (3.14).

4. Check $h(\lambda)$ to see if $|h(\lambda)| \leq 0.05$. If "yes", then stop and return $a$;

   else if $\lambda_3 - \lambda_1 \leq O(2^{-L})$, then stop and return $a$;

   else use the bisection method to update the three points $\lambda_1$, $\lambda_2$ and $\lambda_3$, and go to 2.

*end.*
Remark 3.1

The only issue that needs clarifying in Procedure 3.2 is what to do if (3.13b) has non-real solutions. In fact, from the uniqueness of the optimal solution for CP3.1 and the boundedness of $\lambda$ in (3.12), for any given $\lambda \in [\lambda^*, \infty)$, there must exist a solution to $a_{n+1}$ in (3.13b), such that $a_{n+1}$ is real and $|a_{n+1} - 1| \leq \beta$. Therefore, if (3.13b) has no real solutions, it must be true that $\lambda < \lambda^*$. Essentially, $\lambda$ characterizes the radius ($\beta$) of the interior ellipsoid (sphere) from (1.7c). Consequently, similar to the trust region method, searching for $\lambda^*$ is equivalent to searching for the correct size of the interior ellipsoid region.

Regarding Procedure 3.2, we conclude the following:

Theorem 3.3

In $O(Ln^3)$ arithmetical operations of Procedure 3.2,

$$|h(\lambda_2)| \leq O(2^{-L}).$$

Proof. First, the bisection of Procedure 3.2 is performed within the bounded interval $[O(2^{-L}, O(2^L)]$; second, it can be verified that norms of $P, b^1, b^2$ are all bounded by $O(2^L)$. Therefore, in $O(L)$ steps of Procedure 3.2,

$$|\lambda_2 - \lambda^*| \leq \lambda_3 - \lambda_1 \leq O(2^{-L}),$$

and

$$||a(\lambda_2) - a(\lambda^*)|| \leq O(2^{-L}).$$

Each step of Procedure 3.2 costs $O(n^3)$ arithmetic operations. Overall, in $O(Ln^3)$ operations,

$$||a(\lambda_2) - e|| - \beta = ||a(\lambda_2) - e|| - ||a(\lambda^*) - e||$$

$$\leq ||a(\lambda_2) - a(\lambda^*)|| \leq O(2^{-L}),$$

i.e.,

$$|h(\lambda_2)| \leq O(2^{-L}).$$

Q.E.D.

71
Theorem 3.3 establishes that a resulting from Procedure 3.2 is a valid solution to hold Theorem 3.1, since the range of \( \beta \) can be tolerated from 0.27 to 0.36. Actually, we can use a looser bisection tolerance to complete Procedure 3.2. This is why we set \( |h(\lambda)| \leq 0.05 \) as another criterion to terminate Procedure 3.2. In practice, \( \beta \) has been tolerated from 0 to any positive number as long as solution \( a(\lambda) > 0 \). My computational experience indicates that Procedure 3.2 is always terminated in a few steps using a reasonable estimation on \( \lambda \).

Directly from Theorem 3.1, Theorem 3.2, and Theorem 3.3, we obtain

**Corollary 3.3**

Let the optimal objective value of QP be known in advance. Then Algorithm 3.1 either solves QP in \( O(Ln^5) \) operations when coupled with Procedure 3.1, or it solves QP in \( O(L^2n^4) \) operations when coupled with Procedure 3.2.

Furthermore, with the quadratic objective function given by (0.3), CD can be merged into CP to form

**QPD**

\[
\begin{align*}
\text{minimize} & \quad F(x, y) = f(x) - d(x, y) = x^T Qx + cx - yb \\
\text{subject to} & \quad (x, y) \in Y = \{(x, y) : Ax = b, yA \leq x^T Q + c, x \geq 0\}.
\end{align*}
\]

In QPD, the optimal objective value is known to be zero (except when QPD is infeasible). In addition, the objective function of QPD remains convex quadratic and the constraints of QPD remain linear. Applying Algorithm 3.1 and Procedure 3.2 to QPD, we derive

**Corollary 3.4**

Convex quadratic programming can be correctly solved by Algorithm 3.1 and Procedure 3.1 (or Procedure 3.2) in \( O(Ln^5) \) (or \( O(L^2n^4) \)) arithmetical operations.
3.4 Cutting Dual-Objective Value Technique:

The New Primal-Dual Method for CP

In the preceding section, we achieved a significant theoretical result: convex quadratic programming can be computed in polynomial-time using the PTIE method. The complexity of the algorithm for QP is $O(Ln^5)$ or $O(L^2n^4)$, in contrast to $O(Ln^4)$ for LP, where the extra factor of $n$ or $L$ results from solving CP3.1. Like Karmarkar's original algorithm for LP, Algorithm 3.1 requires prior knowledge of the optimal objective value. If the value is unknown, then we solve the primal and dual programs together, thereby doubling the problem size. In order to improve practical efficiency, I extended the primal-dual method in Section 2.4 to solving CP. With this technique, I use $z$, a dual objective value, to replace $z^*$ as the "cut" while the primal-dual gap is narrowed at each step. Additionally, CP3.1 is solved approximately based on the first-order information of the objective function. Therefore, we do not need to solve the primal and dual programs together, and save the extra complexity factor $n$ in practical implementation.

Replacing $z^*$ with parameter $z(\leq z^*)$, the augmented objective function can be written as

$$f'(x', z) = x'_{n+1}(f(T^{-1}x') - z)$$

and the gradient vector of $f'(x', z)$ becomes

$$\nabla f'(x', z) = (\nabla f(x)w, f(x) - \nabla f(x)x - z).$$

The projection vector from $\nabla f'(x', z)$ to the null space of $A'$ is given by

$$p = r(z) - \frac{f(x) - \nabla f(x)(x - x^k) - z}{n + 1}e,$$

where

$$r(z) = \begin{pmatrix} W(\nabla f(x) - y(z)A)^T \\ d(x, y(z)) - z \end{pmatrix} = u - (z - f(x) + \nabla f(x)x)v,$$

$$y(z) = y_2 + (z - f(x) + \nabla f(x)x)y_1;$$
\(d(.)\) is the dual objective function of CD; and \(y_1, y_2, u, \) and \(v\) are given in (2.7b)–(2.7e) by replacing \(c\) with \(\nabla f(x)\), respectively. Particularly, when \(x = x^k\),

\[
p = \left( W(\nabla f(x^k) - y(z)A)^T \right) - \frac{f(x^k) - z}{n + 1} e. \tag{3.20}
\]

These forms are identical to those discussed in Sections 2.3 and 2.4. The following method for solving CP resembles the new primal-dual method discussed in Section 2.4: at the \(k^{th}\) iteration we increase \(z^k\) to \(z^{k+1}(\leq z^*)\) such that

\[
\phi(z^{k+1}) \leq 0
\]

where

\[
\phi(z) = \min_{1 \leq i \leq n+1} \{r(z)_i\}.
\]

Then, from (3.20),

\[
\|p\| \geq \frac{f(x^k) - z^{k+1}}{n + 1}.
\]

This will guarantee that a fixed improvement can be made in minimizing the augmented objective function. In the meantime, a dual feasible solution \(y(z^{k+1})\) of (3.19) can be generated and updated as was done in Section 2.4. This approach is described as follows.

**Algorithm 3.2**

Let \(-2^ke < x^0 \in \mathcal{X}, z^0 = -M - 1, \) and \(k = 0.\)

\[
\text{while } (f(x^k) - z^k) > M^{-1} \text{ do }
\]

\[
\text{begin }
\]

\[
W = \text{diag}(x^k) \text{ and } c = \nabla f(x^k);
\]

obtain \(y_1, y_2, u, \) and \(v\) via (2.7b)–(2.7e);

\[
z_1 = \min_{1 \leq i \leq n+1} \{\frac{w_i}{v_i} : v_i > 0\} + f(x^k) - cx^k;
\]

\[
\text{if } \{z : \phi(z) \geq 0\} \neq \emptyset \text{ and } z^k < d(x^k, y(z_1)) \text{ then }
\]

\[
y^{k+1} = y(z_1) \text{ and } z^{k+1} = d(x^k, y(z_1)) \text{ else}
\]

74
\[ y^{k+1} = y^k \text{ and } z^{k+1} = z^k \text{ end if; } \]
\[ p^k = u - (z^{k+1} + f(x^k) - cx^k)v - \frac{f(x^k) - z^{k+1}}{n+1}c; \]

let \( \beta \) minimize \( P(T^{-1}(e - \frac{\beta p}{\|p\|}, z^{k+1}); \)

\[ a = e - \frac{\beta p}{\|p\|}; \]
\[ x^{k+1} = T^{-1}(a); \]
\[ k = k + 1; \]

end.

The performance of Algorithm 3.2 results from the following corollary.

**Corollary 3.5**

\[ P(x^{k+1}, z^{k+1}) \leq P(x^k, z^k) - O(\beta). \]

**Proof.** It can be verified that \( \phi(z^{k+1}) \leq 0. \) Therefore,

\[ \|p\| \geq \frac{f(x^k) - z^{k+1}}{n+1}, \]

and

\[ z^k \leq z^{k+1} \leq z^*. \]

In addition, \( a \) is obtained along the steepest descent direction of \( f'(x', z^{k+1}) \) starting with \( e. \) Then, for \( \beta \) small enough,

\[ f'(a, z^{k+1}) = f'(e, z^{k+1}) - O(\beta)\|p\| \]
\[ \leq (1 - \frac{O(\beta)}{n+1})f'(e, z^{k+1}). \]

Thus, similar to the derivation of (2.5),

\[ P(x^{k+1}, z^{k+1}) - P(x^k, z^k) \leq -O(\beta) + O(\beta^2). \]

Since \( \beta \) is chosen such that it minimizes \( P(T^{-1}(a), z^{k+1}); \) it must be true that

\[ P(x^{k+1}, z^{k+1}) - P(x^k, z^k) \leq -O(\beta). \quad Q.E.D. \]
However, there is no guarantee that \( \beta \) will be a fixed number as it is in linear programming. It is possible that \( \beta << 1 \), which severely affects the efficiency of the algorithm since such a small improvement takes \( O(n^3) \) arithmetic operations. But, if \( \beta << 1 \), then \( a \) will be very close to \( e \), and \( x^{k+1} \) will be very close to \( x^k \). In this case, we do not need to update \( W \) to perform the next iteration. We keep solving CP3.1, but this time starting from \( a \) instead of \( e \), to generate the next solution \( a' \) in \( O(n^2) \) operations using the existing inverse matrix \( AW^2A^T \). These steps, called the move procedure, can repeated until the solutions reach the boundary of the interior sphere \( \|a' - e\| \leq 0.36 \), which is illustrated in Figure 3.1.

![Geometric Expression of the Move Procedure](image)

**Figure 3.1.** Geometric Expression of the Move Procedure

In Figure 3.1, essentially we want to minimize the objective function \( f(x) \) within the sphere. At \( e \), we go along the descent direction \( e - a \) to generate \( a \) in the strict interior of the sphere. At \( a \), we can move along the updated descent direction, in either the new gradient direction of \( a - p \) or the conjugate direction of \( a - q \), to generate the next point. These moves last until a solution point hits the boundary of the sphere at \( a' \); then, \( a' \) is the optimal approximation solution for CP3.1. This move procedure, which will be further discussed in the next chapter, essentially results from the steepest direction method, or conjugate direction method, for solving CP3.1 iteratively. The significance of the move procedure is that it uses \( O(n^2) \) operations to achieve the improvement that usually takes \( O(n^3) \) operations.
3.5 Computational Results

Table 3.1 and Table 3.2 are numerical samples for solving QP problems, and Table 3.3 and Table 3.4 are the numerical samples for solving CP problems using Algorithm 3.2 coupled with the move procedure.

Table 3.1 Solution Iterations of the PTIE Method for QP
( obtained objective value within 5-8 digits of the optimal one)

<table>
<thead>
<tr>
<th>Problem index</th>
<th>Size $m \times n$</th>
<th>Optimal value $z^*$</th>
<th>Total iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4 $\times$ 7</td>
<td>0.33333</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>4 $\times$ 7</td>
<td>1.50000</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>4 $\times$ 7</td>
<td>3.00000</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>15 $\times$ 30</td>
<td>0.00000</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>20 $\times$ 40</td>
<td>6.83333</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>20 $\times$ 40</td>
<td>6.53916</td>
<td>23</td>
</tr>
</tbody>
</table>

The QP problems chosen are those containing all possible solution locations—at an extreme point, on the boundary, or in the interior of the feasible polytope. They also represent problems of various sizes.

Table 3.2 Cutting Dual-Objective Value Technique for QP
(optimal objective value $z^* = 3.0$)

<table>
<thead>
<tr>
<th>iteration $(k)$</th>
<th>$x^0 = 0$</th>
<th>$f(x^k)$</th>
<th>$z^k = d(x^k, y^k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>6.83333</td>
<td>0.0000000000</td>
</tr>
<tr>
<td>1</td>
<td>4.96221319</td>
<td>0.0000000000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3.832025134</td>
<td>1.686757358</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3.164552546</td>
<td>2.455500358</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3.067877849</td>
<td>2.966584796</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3.028262273</td>
<td>2.984553149</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.013247304</td>
<td>2.988930474</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3.002031503</td>
<td>2.999617867</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3.000313612</td>
<td>2.999994136</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3.000141008</td>
<td>2.999996807</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3.000030729</td>
<td>2.999998813</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>3.000007471</td>
<td>2.999999992</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>3.000003794</td>
<td>2.999999992</td>
<td></td>
</tr>
</tbody>
</table>
The primal objective value \( f(x^k) \) and dual objective value \( d(x^k, y^k) \) at each iteration for one of the above QP problems are listed in Table 3.2. As we can see, the primal-dual gap converges to zero at the same speed as it does for LP problems (Table 2.5 on page 49). This numerical result agrees with my theoretical result in this chapter.

CP problems are well-known maximum entropy problems. The objective functions of the four problems in Table 3.3 represent a number of widely-used entropy functions in economics, physics, and engineering, such as \( \sum -\ln x_i, \sum -\ln x_i, \sum x_i \ln x_i, \) and \( \sum x_i^2 \). The primal objective value \( f(x^k) \) and dual objective value \( d(x^k, y^k) \) at each iteration for one of the above CP problems are listed in Table 3.4 on page 79.

### Table 3.3 Solution Iterations of the PTIE Method for CP
(Obtained objective value within 4-5 digits of the optimal one)

<table>
<thead>
<tr>
<th>Problem index</th>
<th>Size ( m \times n )</th>
<th>Total iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 4 \times 100 )</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>( 4 \times 100 )</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>( 4 \times 100 )</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>( 4 \times 100 )</td>
<td>21</td>
</tr>
</tbody>
</table>

As we can see in these tables, the sample problems all take considerably fewer iterations to converge (each iteration takes several moves). Note that the amount of work in each iteration is similar to that in each iteration for LP. From my computational experiences, the number of moves increases as the primal-dual gap narrows. Overall, the numerical results can be summarized as follows:

1. The convergence speed in my approach is insensitive to the size of an optimization problem.

2. The convergence speed for CP is indeed at the same level as that for LP, which matches the theoretical results.
3. The move procedure performs extremely well if numerical accuracy is not too restrictive. This is valuable since most real problems are solved on approximation bases.

Table 3.4 Cutting Dual-Objective Value Technique for CP

<table>
<thead>
<tr>
<th>iteration (k)</th>
<th>( z^0 = -10^6 )</th>
<th>( f(x^k) )</th>
<th>( z^k = d(x^k, y^k) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>485.9495862</td>
<td>-1000000.000</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>482.5438613</td>
<td>-10006.13552</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>479.3217646</td>
<td>-8066.392454</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>476.2810144</td>
<td>-6440.027796</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>473.4210118</td>
<td>-5081.828454</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>470.7417206</td>
<td>-3952.660379</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>468.249198</td>
<td>-3018.828036</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>465.9238051</td>
<td>-2251.370440</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>463.7829981</td>
<td>-1625.348786</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>461.8190032</td>
<td>-1119.174899</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>460.0310025</td>
<td>-714.0395455</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>458.4196651</td>
<td>-393.4930199</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>456.9876717</td>
<td>-143.1785356</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>455.7402450</td>
<td>49.36113788</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>454.6869652</td>
<td>194.91484041</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>453.8465381</td>
<td>302.90296217</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>453.2546948</td>
<td>381.64339531</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>452.9722017</td>
<td>438.57409101</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>452.9594872</td>
<td>452.91113528</td>
<td></td>
</tr>
</tbody>
</table>

3.6 Summary

In this chapter, I extended the PTIE method to solving linearly constrained convex programs in order to achieve global convergence within a "worst" case bound. The number of iterations of the algorithm is bounded by \( O(Ln) \), and each iteration requires solving a continuous optimization problem, where \( n \) is the dimension of the problem and \( L \) is the number of bits in the input. Particularly, if the objective is a quadratic function, two approaches were introduced: the first approach uses \( O(Ln^2) \) iterations, and each iteration can be completed in \( O(n^3) \) arithmetic operations; the second approach uses \( O(Ln) \) iterations, and each iteration can be computed in \( O(Ln^3) \) arithmetic operations. The exact
optimal solution can be found by rounding the error. This extension is based on the fact that the convexity of the objective function is projective-invariant in the objective augmentation and the projective transformation.

I have recently learned that Kapoor and Vaidya [34] also devised independently a similar extension of Karmarkar's algorithm for convex quadratic programming. Both my and Kapoor and Vaidya's methods generate a sequence of interior solution points. However, in addition, my approach avoids a line search procedure proposed in their method and saves a factor $O(\log(n))$ in the complexity bound. Consequently, it converges faster, according to my computational experience.

An implementation of this algorithm, the new primal-dual method, was also discussed in this chapter (not by Kapoor and Vaidya). The method does not require prior knowledge of the optimal objective value and, as a result, is more efficient in practice. The computational results show that each iteration takes about $O(n^3)$ operations, and regardless of the size of the problem, the total number of iterations required to achieve 6-digit accuracy is about 15 to 20.

Moreover, since there is no need to solve CP3.1 exactly, we can use many existing iterative algorithms, such as the quasi-Newton method (Buckley [7], Fletcher [19]), the conjugate direction method (Dixon [15], Shanno [57]), and other computational methods (Polak [53], Powell [54]), to achieve further practical efficiency.
Chapter 4  P-Simplex: Improving PTIE Efficiency

4.1 Introduction

One difficulty with the PTIE method is that each column of A is active when CP3.1 is solved. Therefore, each iteration needs $O(n^3)$ operations. In this chapter, I introduce a large group of simplices and discuss a related tangent-plane move (TPM) procedure. Like the move procedure discussed at the end of Section 3.5, this sub-procedure takes only $O(n^2)$ operations to reduce the potential function. The reason for using the TPM procedure is to further improve the efficiency of the PTIE method.

In Section 4.2, I discuss a general class of simplices, p-simplex, and their basic geometric properties. Then I introduce the TPM procedure for solving CP3.1 within the p-simplex constraint. Section 4.3 discusses barrier, or 0-simplex, specifically. Here, the TPM procedure guarantees a reduction with respect to the potential function, but takes considerably less time to compute. Section 4.4 presents the computational results.

4.2 p-Simplex and p-Projective Transformation

As discussed in Section 2.2, via the projective transformation $T$, the new feasible region is contained in the simplex $S$, in which the ratio of the two radii of inscribing and circumscribing spheres is greater than $\frac{1}{n+1}$. This ratio plays a fundamental role in polynomial convergence of the PTIE method. Likewise, the other geometrical regions may also achieve this ratio. A general class of such geometrical regions is the p-simplex given by

$$S_p = \{x \in \mathbb{R}^{n+1} : \|x\|_p = n + 1 \text{ and } x \geq 0\}.$$  \tag{4.1}

where $\|\cdot\|_p$ is the classical p-norm defined by

$$\|x\|_p = \left(\sum_{i=1}^{n+1} x_i^p\right)^{\frac{1}{p}}, \text{ for } p = 1, 2, \ldots$$  \tag{4.2}
Another important simplex, called the barrier-simplex or 0-simplex, is

\[ S_0 = \{ x \in \mathbb{R}^{n+1} : \sum_{i=1}^{n+1} \ln(x_i) = 0 \text{ and } x \geq 0 \}. \]  \hspace{1cm} (4.3)

The geometric surfaces of \( p \)-simplex are illustrated in Figure 4.1, where \( n = 1 \).

![Figure 4.1. p-Simplex](image)

In Figure 4.1, the line \( x_1-x_2 \) (hyperplane) represents \( S_1 \), the circle (sphere) represents \( S_2 \), the hyperbolic curve represents \( S_0 \), and the surface of the square represents \( S^\infty \). As we can see, they all contain \( e \) as the geometrical center. In general,

\[ \frac{r}{R} = \frac{1}{\sqrt{n + ((n + 1)^{1/p} - 1)^2}} \]

where \( r \) is the radius of the inscribing sphere and \( R \) is the radius of the circumscribing sphere. As we can see, the ratio of the two radii are closer for \( p \geq 2 \). Let \( W \) be given by (1.3). Then, the nonnegative \( n \)-dimensional orthant can be mapped into \( S_p \) via the following \( p \)-projective transformation \( T_p \), which is defined by

for \( p \geq 1 \):

\[ x'[n] = \frac{(n + 1)^{1/p} W^{-1} x}{F(x)} \quad \text{and} \quad x'_{n+1} = \frac{(n + 1)^{1/p}}{F(x)} \]

where

\[ F(x) = \left( \sum_{i=1}^{n} \left( \frac{x_i}{w_i} \right)^p + 1 \right)^{\frac{1}{p}}; \]
and for $p = 0$:

$$
x'[n] = \frac{W^{-1}x}{F(x)} \quad \text{and} \quad x'_{n+1} = \frac{1}{F(x)}
$$

where

$$
F(x) = \left( \prod_{i=1}^{n} \frac{x_i}{w_i} \right)^{\frac{1}{n+1}}.
$$

The inverse transformation $T_p^{-1}$ remains unchanged, as is given in Section 2.2.

Replacing the 1-simplex constraint with the $p$-simplex constraint, CP3 in Section 3.2 becomes

$$
\text{CP4}(z^*) \quad \text{minimize} \quad f'(x')
$$

subject to

$$
(AW, -b)x' = 0
$$

$$
x' \in S_p.
$$

The difficulty in CP4 is that we face a nonlinear constraint surface. Hence, we are concerned with the tangent planes to these surfaces. It can be verified that the tangent-plane to $S_p$ at $a \in S_p$ is given by

$$
\{x' \in R^{n+1} : \sum_{i=1}^{n+1} a_i^{p-1} x_i = n + 1 \}.
$$

Therefore, in CP4 we can replace the nonlinear surface $S_p$ with the tangent-plane to $S_p$ at $a$. Let $r = \min_{1 \leq i \leq n+1} a_i$, and let $B(a, \beta r) = \{x' : \|x' - a\| \leq \beta r\}$, which is an interior sphere centered at $a$ (not necessarily at the center $e$) and which inscribes the positive orthant, and let $a^p$ designate the row-vector with each component as $a_i^p$. Then, as we did for CP3, we solve CP4.1 instead of solving CP4.

$$
\text{CP4.1}(z^*) \quad \text{minimize} \quad f'(x')
$$

subject to

$$
A'x' = b'
$$

$$
\|x' - a\| \leq \beta r
$$

where

$$
A' = \left( \begin{array}{c} AW, -b \\ a^{p-1} \end{array} \right)
$$
and

\[ b' = \begin{pmatrix} 0 \\ n + 1 \end{pmatrix}. \]

If \( a = e \) or \( p = 1 \), then CP4.1 merges into CP3.1. The next lemma guarantees that there exists a point on the tangent-plane to represent the optimal solution.

**Lemma 4.1**

There exists \( a^* \geq 0 \) such that \( a^* \) satisfies the equality constraints in CP4.1 and \( T^{-1}(a^*) \) is the optimal solution of CP.

**Proof.** Let \( x'^* = T_1(x^*) \) via 1-projective transformation, where \( x^* \) is the optimal solution for CP. Note the homogeneity of \( T_p^{-1} \), i.e.,

\[ T_p^{-1}(tx') = T_p^{-1}(x') = x \quad \text{for} \quad t > 0. \]

Now choose \( t \) such that

\[ t = \frac{n + 1}{a^{p-1}x'^*} \]

and

\[ a^* = tx'^*. \]

Then it can be verified that \( a^* \) satisfies the equality constraints in CP4.1, and

\[ T_p^{-1}(a^*) = T_p^{-1}(tx'^*) = x^*. \quad Q.E.D. \]

The following lemma measures the radius of the circumscribing sphere. Note that \( ||\cdot|| \) (no subscript) designates the \( L_2 \)-norm in Euclidean space.

**Lemma 4.2**

Let \( R \) be the radius of the smallest sphere centered at \( a \) and circumscribing \( \{x' : a^{p-1}x' = n + 1 \text{ and } x' \geq 0\} \). Then, for \( 0 \leq p \leq 2 \),

\[ R \leq \frac{\max_{1 \leq i \leq n+1}(a_i^{1-p/2})^2}{\min_{1 \leq i \leq n+1}(a_i^{p/2})}(n + 1). \]
Proof. For any $x' \geq 0$ on the tangent-plane, we have

$$\|x' - a\|^2 = \sum_{i=1}^{n+1} (x'_i - a_i)^2$$

$$= \sum_{i=1}^{n+1} a_i^{2-p}(a_i^{p/2-1}x'_i - a_i^{p/2})^2$$

$$\leq \max(a_i^{2-p}) \sum_{i=1}^{n+1} (a_i^{p/2-1}x'_i - a_i^{p/2})^2$$

$$\leq \max(a_i^{2-p})(\sum_{i=1}^{n+1} a_i^{p-2}x_i'^2 - n - 1)$$

$$\leq \max(a_i^{2-p}) \sum_{i=1}^{n+1} \frac{a_i^{2p-2}x_i'^2}{a_i^p}$$

$$\leq \frac{\max(a_i^{2-p})}{\min(a_i^p)} \sum_{i=1}^{n+1} (a_i^{p-1}x_i')^2$$

$$\leq \frac{\max(a_i^{2-p})}{\min(a_i^p)} (n + 1)^2.$$  

Therefore,

$$R \leq \frac{\max(a_i^{1-p/2})}{\min(a_i^{p/2})} (n + 1).$$  

Q.E.D.

Let

$$\gamma = \frac{\min(a_i^{1+p/2})}{\max(a_i^{-p/2})}.$$  

(4.5)

Then, since $r$ represents the radius of the inscribing sphere $B(a, r)$,

$$\frac{r}{R} \geq \frac{\gamma}{n + 1}.$$  

This leads to the next theorem that evaluates the convergence ratio of each tangent-plane move.
Theorem 4.1

Let $a'$ be the minimal solution for CP4.1. Then

$$\frac{f'(a')}{f'(a)} \leq 1 - \frac{\beta \gamma}{n + 1}.$$ 

Proof. Connecting $a^*$ and $a$, we have a feasible point $a''$ for CP4.1 such that

$$a'' = \theta a^* + (1 - \theta)a \quad \text{for some} \quad 0 < \theta < 1.$$ 

This implies that

$$\theta \|a^* - a\| = \|a'' - a\| = \beta r.$$ 

Since $\|a^* - a\| \leq R$ from Lemma 4.2,

$$\theta \geq \frac{\beta r}{R} \geq \frac{\beta \gamma}{n + 1}.$$ 

Following the same arguments in Lemma 3.2,

$$f'(a') \leq f'(a'')$$

$$\leq \theta f'(a^*) + (1 - \theta)f'(a)$$

$$= (1 - \theta)f'(a).$$

$$\leq (1 - \frac{\beta \gamma}{n + 1})f'(a).$$

Q.E.D.

Figure 4.2 on page 87 illustrates the basic idea of the TPM procedure in the case of $p = 2$. We start from $e$ to generate the next point $\bar{a}$. At this point, if no further TPM applies, then we projective-transform $\bar{a}$ to the center $e$ of a new simplex and solve a new CP4.1 starting from $e$. This step, called a major iteration, costs $O(n^3)$. Applying the TPM procedure, we update the tangent-plane to $S_p$ at the solution $\bar{a}$ and then continue solving CP4.1 starting from $\bar{a}$. This step, called a TPM, costs $O(n^2)$ since only the last row of $A'$ changed and the new moving direction can be updated by exploiting previous computations. Theorem 4.1 claims that if $\gamma$ is close to 1, then each TPM guarantees the same convergence ratio of $(1 - O(\frac{\theta}{n}))$ as obtained in the major iteration. Therefore, under certain circumstances, the “cheaper” TPM achieves the same result that is usually achieved by the “expensive” major iteration.
4.3 Barrier-Simplex in PTIE

In this section, I discuss the special case of \( p = 0 \) in detail. In barrier-simplex, the term \( \sum \ln(a_i) \) is fixed at zero. To evaluate the reduction of the potential function \( P(x^k, z^*) \) in each TPM, I derived the following lemma.

**Lemma 4.3**

Let \( a' \) be on the sphere \( B(a, \beta r) \) and the tangent-plane \( \{ x' : a^{-1} x' = n+1 \} \).

Then

\[
\sum_{i=1}^{n+1} \ln\left(\frac{a'_i}{a_i}\right) \geq \frac{-\beta^2}{2(1-\beta)^2}.
\]

**Proof.** Let

\[
x'_i = \frac{a'_i}{a_i} \quad \text{for} \quad i = 1, 2, \ldots, n+1.
\]

Then

\[
e^T x' = a^{-1} a' = n + 1,
\]

87
and
\[
\|x' - e\|^2 = \sum_{i=1}^{n+1} \left( \frac{a'_i}{a_i} - 1 \right)^2
\]
\[
= \sum_{i=1}^{n+1} a_i^{-2} (a'_i - a_i)^2
\]
\[
\leq r^{-2} \|a' - a\|^2
\]
\[
= r^{-2} (\beta r)^2
\]
\[
= \beta^2.
\]
Therefore, from (2.4)
\[
\sum_{i=1}^{n+1} \ln \left( \frac{a'_i}{a_i} \right) = \sum_{i=1}^{n+1} \ln (x'_i) \geq \frac{-\beta^2}{2(1 - \beta)^2}.
\]
Q.E.D.

Based on Theorem 4.1 and Lemma 4.3, we have

**Theorem 4.2**

Let \( p = 0 \) and \( a' \) be the minimal solution of CP4.1. Then
\[
P(T^{-1}(a'), z^*) \leq P(T^{-1}(a), z^*) - \gamma \beta + \frac{\beta^2}{2(1 - \beta)^2}.
\]

Proof.
\[
P(T^{-1}(a'), z^*) - P(T^{-1}(a), z^*) = (n + 1) \ln \left( \frac{f'(a')}{f'(a)} \right) - \sum_{i=1}^{n+1} \ln \left( \frac{a'_i}{a_i} \right)
\]
\[
\leq (n + 1) \ln \left( 1 - \frac{\gamma \beta}{n + 1} \right) + \frac{\beta^2}{2(1 - \beta)^2}
\]
\[
\leq -\gamma \beta + \frac{\beta^2}{2(1 - \beta)^2}.
\]
Q.E.D.

Thus, from Theorem 4.2 we can derive
Corollary 4.1

Let $\beta = 1/8$. Then the first seven TPM's to barrier-simplex in solving CP4.1 reduce the potential function by at least 0.4.

Based on the result of Corollary 4.1, the TPM procedure makes the PTIE method twice as fast as before: each iteration of the PTIE method with TPM reduces the potential function by 0.4, but each iteration without TPM reduces the potential function by only 0.2, while both of these approaches use the same number of computations. It can also be verified that if the iterative solutions are close to the optimal solution, i.e., $\gamma$ is close to 1, then several tangent-plane moves reduce the potential function by $O(\sqrt{n})$ [67].

From a computational standpoint, TPM is an approximation method for solving least squares problems. Suppose that, without loss of generality, we have solved the following LP optimization problem, with $0 < \beta < 1/3$, at the $k^{th}$ iteration:

\[
\begin{align*}
\text{LP}(k) & \quad \text{minimize} \quad cW^k x'[n] - z^* x'_{n+1} \\
& \quad \text{subject to} \quad AW^k x'[n] - x'_{n+1} b = 0 \\
& \quad \quad \quad \quad \quad e^T x' = n + 1 \\
& \quad \quad \quad \quad \quad \|x' - e\|^2 \leq \beta^2,
\end{align*}
\]

where $W^k = \text{diag}(x^k)$ and $e$ is the feasible starting solution.

Then, at the $(k+1)^{th}$ iteration of the PTIE method we need to solve another LP optimization problem with a step size of $0 < \beta < 1/3$ as LP$(k+1)$ with

\[
\begin{align*}
W^{k+1} &= \text{diag}(x^{k+1}), \\
x^{k+1} &= W^k a[n]/a_{n+1},
\end{align*}
\]

and $a$ being the optimal solution for LP$(k)$. Instead of solving LP$(k+1)$, we introduce a transformation

\[
x''_i = a_i x'_i \quad \text{for} \quad i = 1, 2, \ldots, n + 1.
\]
Hence, LP(k + 1) becomes

\[
\begin{align*}
\text{LP}'(k + 1) \quad & \text{minimize} \quad c W^k x''[n] - z'' x''_{n+1} \\
\text{subject to} \quad & A W^k x''[n] - x''_{n+1} b = 0 \\
& a^{-1} x'' = n + 1 \\
& \| \text{diag}(a^{-1}) x'' - e \| ^2 \leq \beta ^2.
\end{align*}
\]

Replacing the ellipsoid constraint in LP'(k + 1) with \( \|x'' - e\| \leq \beta \), we see that LP'(k + 1) becomes identical to LP(k), except in the last linear-equality constraint. The gradient-projection vector, \( p \), of LP'(k + 1) can be updated by solving two triangular systems of linear equations in \( O(n^2) \) operations, where the triangular matrix is obtained in the previous computations for solving LP(k). Overall, the algorithm could consist of major iterations for solving LP(k) exactly, and sub-iterations for solving LP'(k + 1) approximately (i.e., solving LP'(k + 1)) between every two major iterations. Comparing LP'(k + 1) to CP4.1, the sub-iteration is exactly the same as the tangent-plane move in barrier-simplex. Since \( a \) is usually very close to \( e \), the optimal solution for LP'(k + 1) is quite a good approximation of the solution for LP(k + 1). Even when \( a \) is far from \( e \), the tangent-plane move seems to work well in my computational experience.

4.4 Computational Results

Table 4.1 on page 91 presents the PTIE method, coupled with TPM to barrier-simplex, for solving LP problems. There are eight problems listed in Table 4.1. I will explain each column in the table by using the second problem, called EXP1, as an example. The dimension of matrix \( A \) of EXP1 is 10 \( \times \) 17, which is shown in the first column. In Phase 1, i.e., finding the initial feasible starting point, the PTIE method (with TPM) takes a total of 3 major iterations and 5 moves; Karmarkar’s algorithm takes 6 major iterations. In Phase 2, PTIE takes a total of 5 major iterations and 12 moves whereas Karmarkar’s algorithm takes 11 major iterations. The last column of Table 4.1 shows that the simplex method (Murtagh and Saunders [50]) solves EXP1 in 56 pivots.
I emphasize that each major iteration of both the PTIE method and Karmarkar's algorithm uses $O(Ln^3)$ arithmetic operations, and each move uses only $O(Ln^2)$ operations. We can see from the table that the number of major iterations in the PTIE method is significantly reduced by applying TPM's.

Table 4.1 Computational Results of Tangent-Plane Moves with Barrier-Projection (optimal objective value used)

<table>
<thead>
<tr>
<th>Problem size $(m \times n)$</th>
<th>Barrier-Projection</th>
<th>Karmarkar's</th>
<th>Minos</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of itns</td>
<td># of moves</td>
<td># of itns</td>
</tr>
<tr>
<td>$4 \times 6$</td>
<td>2</td>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>$10 \times 17$</td>
<td>8</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>$12 \times 24$</td>
<td>6</td>
<td>19</td>
<td>15</td>
</tr>
<tr>
<td>$27 \times 51$</td>
<td>7</td>
<td>27</td>
<td>16</td>
</tr>
<tr>
<td>$47 \times 80$</td>
<td>8</td>
<td>37</td>
<td>16</td>
</tr>
<tr>
<td>$47 \times 80$</td>
<td>9</td>
<td>39</td>
<td>18</td>
</tr>
<tr>
<td>$56 \times 138$</td>
<td>11</td>
<td>47</td>
<td>32</td>
</tr>
<tr>
<td>$96 \times 162$</td>
<td>9</td>
<td>38</td>
<td>27</td>
</tr>
</tbody>
</table>

4.5 Summary

In this chapter, I introduced a large class of simplices and the associated tangent-plane move procedure. The motivation for adding these is to further improve the efficiency of the PTIE method by exploiting the computational results from the previous steps. Each of the moves in barrier-simplex, having a mathematically-guaranteed performance, uses only $O(Ln^2)$ arithmetic operations, whereas each major iteration costs $O(Ln^3)$ arithmetic operations.

By applying the tangent-plane moves, I reduce the number of major iterations significantly in my computational results; thus, the PTIE method for both LP and CP converges much faster than before.
Chapter 5  PFIE Method: Convex Quadratic Programming

5.1 Introduction

Since Karmarkar proposed the projective interior algorithm, another popular polynomial-time interior algorithm, the new “centering” method, has been developed by Renegar [55] and Sonnevend [59]. Renegar obtained a convergence ratio \((1 - O(\frac{1}{n}))\), comparing to the one \((1 - O(\frac{1}{n}))\) in Karmarkar’s algorithm. Recently, using the rank-one updating scheme, Gonzaga [29] further reduced the solution time for LP by a factor \(n^{0.5}\).

Based on this “centering” idea, I incorporate the work of the barrier pathway to the optimal set of Megiddo [48], the primal-dual interior algorithm of Kojima, Mizuno and Yoshise [38], and the interior ellipsoid method in Chapter 1 to develop a “center” path-following and interior ellipsoid (PFIE) algorithm for QP. This algorithm creates a sequence of primal and dual interior feasible points converging to the optimal solution. At each iteration, the complementary slackness value, i.e., the objective gap between the primal and dual, is reduced at a global ratio \((1 - \frac{1}{4\sqrt{n}})\). Therefore, the algorithm solves QP in \(O(Ln^{3.5})\) arithmetic operations. I also discuss how to use a line search technique to achieve practical efficiency for PFIE algorithm.

5.2 “Center” and Barrier Function

In the interior ellipsoid method of Chapter 1, if the optimal solutions \(x^{k+1}\) and \(y^{k+1}\) of CP1.1 on page 18 are feasible for CD, then we have the following convergence ratio from Theorem 1.2,

\[
\frac{f(x^{k+1}) - z^*}{f(x^k) - z^*} \leq (1 - \frac{\beta}{\sqrt{n}})\frac{f(x^k) - z^*}{f(x) - z^*}.
\]

Unfortunately, this feasibility condition is not necessarily satisfied at each iteration. Geometrically, the problem is that the starting point \(R\) of Figure 1.2 can be very close to a nonoptimal vertex, so that the interior ellipsoid centered at \(R\) will be too small to improve the solution. On the other hand, if \(R\) is close to the “center”—the “most interior” point of the feasible polyhedron—we can draw a large interior ellipsoid centered at \(R\) to further reduce the objective function.
The questions are:

1) How to define the “center” path in a feasible polyhedron for QP?

2) How to force the solution points follow the “center” path to the optimal solution set?

3) How to generate the initial point that is close to the “center” path?

To answer the first question, let $x$ and $y$ be an interior feasible solution pair for QD, i.e.,

$$Ax = b, x > 0 \text{ and } s = Qx + c^T - A^Ty > 0.$$  \hspace{1cm} (5.1)

Megiddo analyzed a “center” path to the optimal set

$$P_{center} = \{(x, s) : Xs = ze, \ z = \frac{e^T Xs}{n}, \ 0 \leq z < \infty\},$$

where $X = \text{diag}(x)$, and

$$z = \frac{e^T Xs}{n} = \frac{x^T Qx - b^T y}{n} = \frac{f(x) - d(x, y)}{n}. \hspace{1cm} (5.2)$$

In this chapter, we say that $x$ and $s$ are $\alpha$-close to the “center” path $P_{center}$ if and only if

$$\|Xs - ze\| \leq \alpha z, \hspace{1cm} (5.3)$$

As we can see, $z$ is the mean value and $\|Xs - ze\|$ is the standard deviation of the complementary slackness vector $Xs$ at $x$ and $y$.

In addressing the second question, a motivation from the barrier function method is to solve the following suboptimization problem at the $k$th iteration instead of CP1.1.

**QP5.1** \[ \begin{align*}
\text{minimize} & \quad \frac{x^T Qx}{2} + cx - \lambda e^T (X^k)^{-1}(x - x^k) \\
\text{subject to} & \quad Ax = b \\
& \quad \|(X^k)^{-1}(x - x^k)\|^2 \leq \beta^2,
\end{align*} \]

where $X^k = W = \text{diag}(x^k)$, $e^T (X^k)^{-1}(x - x^k)$ is the linear approximation of the barrier function $\sum_{i=1}^n \ln(x_i)$ at $x^k$, and $\lambda > 0$ is the barrier parameter. Let $x^{k+1}$ be the optimal solution for QP5.1. Then again, $x^{k+1}$ and $y^{k+1}$ meet the following optimality condition:
\[
\begin{pmatrix}
Q + \mu(X^k)^{-2} & - A^T \\
A & 0
\end{pmatrix}
\begin{pmatrix}
x^{k+1} \\
y^{k+1}
\end{pmatrix} =
\begin{pmatrix}
\lambda(X^k)^{-1}e - c^T + \mu(X^k)^{-1}e \\
b
\end{pmatrix}.
\]

(5.4)

The above equation is identical to (1.19) except for the term \(\lambda(X^k)^{-1}e\) on the right-hand side. Suppose that (5.1) and (5.3) hold for \(x^k\) and \(y^k\), i.e.,

\[
\|X^k s^k - z^k e\| \leq \alpha z^k,
\]

(5.5)

in the following section I show that by appropriately selecting \(\mu\), \(\lambda\), and \(\alpha\), \(x^{k+1}\) and \(y^{k+1}\) still satisfy equations (5.1) and (5.3), or \(x^{k+1}\) and \(y^{k+1}\) are still close to the “center” path. But additionally, we have

\[
z^{k+1} \leq (1 - \frac{1}{4\sqrt{n}})z^k.
\]

This inequality shows that the objective gap between the primal and dual, or the value of the complementary slackness vector is reduced at a global ratio \(1 - \frac{1}{4\sqrt{n}} < 1\).

5.3 The “Center” Path-Following Algorithm

Let

\[
\Delta x = x^{k+1} - x^k, \quad \Delta y = y^{k+1} - y^k, \quad \text{and} \quad \Delta s = Q\Delta x - A^T\Delta y
\]

and

\[
S^k = \text{diag}(s^k) \quad \text{and} \quad \Delta X = \text{diag}(\Delta x).
\]

Then (5.4) can be rewritten as

\[
\begin{pmatrix}
Q + \mu(X^k)^{-2} & - A^T \\
A & 0
\end{pmatrix}
\begin{pmatrix}
\Delta x \\
\Delta y
\end{pmatrix} =
\begin{pmatrix}
\lambda(X^k)^{-1}e - s^k \\
0
\end{pmatrix},
\]

(5.6)

or

\[
X^k \Delta s + \mu(X^k)^{-1} \Delta x = \lambda e - X^k s^k
\]

(5.7)

and

\[
A \Delta x = 0.
\]

(5.8)
We now select that
\[ \mu = z^k \quad \text{and} \quad \lambda = (1 - \frac{\alpha}{\sqrt{n}})z^k. \]  
(5.9)

Then we can establish the following three lemmas.

**Lemma 5.1**

\[
\|(X^k)^{-1} \Delta x\| \leq \sqrt{2} \alpha; \\
\|(S^k)^{-1} \Delta s\| \leq \frac{\sqrt{2} \alpha}{1 - \alpha}; \\
\|\Delta X \Delta s\| \leq \alpha^2 z^k. \]  
(5.10)

**Proof.** Since \( Q \) is positive-semi definite, 
\[
\Delta x^T \Delta s = \Delta x^T Q \Delta x - \Delta x^T A^T \Delta y \\
= \Delta x^T Q \Delta x \geq 0.
\]

Thus,
\[
\|\lambda e - X^k s^k\|^2 = \|X^k \Delta s + \mu(X^k)^{-1} \Delta x\|^2 \\
= \|X^k \Delta s\|^2 + 2 \mu \Delta x^T \Delta s + \|\mu(X^k)^{-1} \Delta x\|^2 \\
\geq \|X^k \Delta s\|^2 + \|\mu(X^k)^{-1} \Delta x\|^2. \]  
(5.11)

Hence,
\[
\|\mu(X^k)^{-1} \Delta x\| \leq \|\lambda e - X^k s^k\| \]  
(5.12)
\[
\|X^k \Delta s\| \leq \|\lambda e - X^k s^k\|. \]  
(5.13)

However, using (5.5) and (5.9), we have
\[
\|\lambda e - X^k s^k\|^2 = \|z^k e - X^k s^k - \frac{\alpha z^k}{\sqrt{n}} e\|^2 \\
= \|z^k e - X^k s^k\|^2 + \|\frac{\alpha z^k}{\sqrt{n}} e\|^2 \\
\leq (\alpha z^k)^2 + (\alpha z^k)^2 \\
= 2(\alpha z^k)^2. \]  
(5.14)
Therefore, from (5.12) and (5.14)
\[ \|(X^k)^{-1} \Delta x\| \leq \frac{\sqrt{2\alpha \varepsilon_k}}{\mu} = \sqrt{2\alpha}; \]
from (5.5), (5.13) and (5.14)
\[ \|(S^k)^{-1} \Delta s\| = \|(S^k X^k)^{-1} X^k \Delta s\| \]
\[ \leq \frac{\|X^k \Delta s\|}{(1 - \alpha)\varepsilon_k} \]
\[ = \frac{\sqrt{2\alpha}}{1 - \alpha}; \]
and from (5.11) and (5.14)
\[ \|\Delta X \Delta s\| = \|\Delta X(X^k)^{-1} X^k \Delta s\| \]
\[ \leq \frac{1}{\mu} \|\mu(X^k)^{-1} \Delta x\|\|X^k \Delta s\| \]
\[ \leq \frac{1}{\mu} \|\mu(X^k)^{-1} \Delta x\|^2 + \|X^k \Delta s\|^2 \]
\[ \leq \frac{1}{2\mu} \|\lambda e - x^{k+1} s\|^2 \]
\[ \leq \frac{1}{2\varepsilon_k} (\sqrt{2\alpha \varepsilon_k})^2 = \alpha^2 \varepsilon_k. \]

Lemma 5.1 essentially claims that \( x^{k+1} \) and \( y^{k+1} \) remain interior solutions for QP and QD and that the second order term \( \|\Delta X \Delta s\| \) of the new complementary slackness vector \( X^{k+1} s^{k+1} \) is relatively small if \( \alpha \) is small enough.

The second lemma establishes a global convergence ratio in minimizing the primal-dual objective gap.

Lemma 5.2
\[
(1 - \frac{\alpha}{\sqrt{n}} - \frac{\sqrt{2\alpha^2}}{n}) \varepsilon_k \leq z^{k+1} \leq (1 - \frac{\alpha}{\sqrt{n}} + \frac{\alpha^2}{4n}) \varepsilon_k.
\]

Proof. Note from (5.7) that
\[
X^{k+1} s^{k+1} = (X^k + \Delta X)(s^k + \Delta s)
= X^k s^k + \Delta X s^k + X^k \Delta s + \Delta X \Delta s
= X^k s^k + X^k \Delta s + \mu(X^k)^{-1} \Delta x - \mu(X^k)^{-1} \Delta x + X^k s^k + \Delta X \Delta s
= \lambda e + \Delta X(s^k + \Delta s - \mu(X^k)^{-1} e). \tag{5.15}
\]
From (5.7), (5.15) can also be written by

\[ X^{k+1}s^{k+1} = \lambda e + \Delta X(X^k)^{-1}(\lambda e - \mu(X^k)^{-1}\Delta x - \mu e). \]  

(5.16)

From (5.16),

\[ nz^{k+1} = e^T X^{k+1}s^{k+1} \]

\[ = n\lambda + (\lambda - \mu)e^T(X^k)^{-1}\Delta x - \mu\|(X^k)^{-1}\Delta x\|^2 \]

\[ = n\lambda - \frac{\alpha z^k}{\sqrt{n}}e^T(X^k)^{-1}\Delta x - \mu\|(X^k)^{-1}\Delta x\|^2 \]

\[ \leq n\lambda + \frac{\alpha z^k}{\sqrt{n}}|e^T(X^k)^{-1}\Delta x| - \mu\|(X^k)^{-1}\Delta x\|^2 \]

\[ \leq n\lambda + \frac{\alpha z^k}{\sqrt{n}}\|e^T\|\|(X^k)^{-1}\Delta x\| - \mu\|(X^k)^{-1}\Delta x\|^2 \]

\[ = n\lambda + \alpha z^k\|(X^k)^{-1}\Delta x\| - \mu\|(X^k)^{-1}\Delta x\|^2 \]

\[ = n\lambda + z^k(\alpha\|(X^k)^{-1}\Delta x\| - \|(X^k)^{-1}\Delta x\|^2) \]

\[ \leq n\lambda + \frac{\alpha^2z^k}{4}. \]

The last inequality holds since the quadratic term achieves maximum at \(\|(X^k)^{-1}\Delta x\| = \alpha/2\). Thus, via (5.9)

\[ z^{k+1} \leq (1 - \frac{\alpha}{\sqrt{n}} + \frac{\alpha^2}{4n})z^k. \]

From (5.15)

\[ nz^{k+1} = e^T X^{k+1}s^{k+1} \]

\[ = n\lambda + \Delta x^T(X^k)^{-1}(X^k s^k - \mu e) + \Delta x^T\Delta s \]

\[ \geq n\lambda + \Delta x^T(X^k)^{-1}(X^k s^k - \mu e) \]

\[ \geq n\lambda - |\Delta x^T(X^k)^{-1}(X^k s^k - \mu e)| \]

\[ \geq n\lambda - \|\Delta x^T(X^k)^{-1}\|\|(X^k s^k - \mu e)\| \]

\[ \geq n\lambda - \sqrt{2}\alpha\alpha z^k. \]

Thus, via (5.9)

\[ z^{k+1} \geq (1 - \frac{\alpha}{\sqrt{n}} - \frac{\sqrt{2}\alpha^2}{n})z^k. \]

Q.E.D.
The third lemma confirms that \( x^{k+1} \) and \( y^{k+1} \) are still close to the "center" path.

**Lemma 5.3**

\[
\|X^{k+1}s^{k+1} - z^{k+1}e\| \leq (1 + \sqrt{2})\alpha^2 z^k
\]

**Proof.**

\[
\|X^{k+1}s^{k+1} - z^{k+1}e\| = \|X^{k+1}s^{k+1} - \lambda e + \lambda e - z^{k+1}e\|
\]

\[
= \|((X^{k+1}s^{k+1} - \lambda e) - (z^{k+1} - \lambda)e\|
\]

\[
\leq \|((X^{k+1}s^{k+1} - \lambda e)\|.
\]

The above inequality hold since the standard deviation is less than the square-root of the second order moment. From (5.7), (5.10), and (5.15) and the above inequality,

\[
\|X^{k+1}s^{k+1} - z^{k+1}e\| \leq \|((X^{k+1}s^{k+1} - \lambda e)\|
\]

\[
= \|\Delta X(s^k + \Delta s - \mu(X^k)^{-1}e)\|
\]

\[
\leq \|\Delta X(s^k - \mu(X^k)^{-1}e)\| + \|X\Delta s\|
\]

\[
= \|(X^k)^{-1}\Delta x\|\|(X^k)s^k - \mu e)\| + \|X\Delta s\|
\]

\[
\leq \sqrt{2\alpha\alpha z^k + \alpha^2 z^k}
\]

\[
= (1 + \sqrt{2})\alpha^2 z^k.
\]

**Q.E.D.**

Based on the above three lemmas, we derive

**Theorem 5.1**

Let \( \alpha = 1 - \frac{\sqrt{2}}{2} \) and \( n \geq 2 \). Then

\[
Ax^{k+1} = b, x^{k+1} > 0, \quad \text{and} \quad s^{k+1} = Qx^{k+1} + c^T - A^T y^{k+1} > 0;
\]

\[
\|X^{k+1}s^{k+1} - z^{k+1}e\| \leq \alpha z^{k+1};
\]

and

\[
z^{k+1} \leq (1 - \frac{1}{4\sqrt{n}})z^k.
\]
Proof. The first two inequalities hold due to (5.8) and Lemma 5.1; the third inequality holds due to the left inequality of Lemma 5.2 and Lemma 5.3; and the fourth inequality is true due to the right inequality of Lemma 5.2. Q.E.D.

Now the “center” path-following algorithm can be described as follows.

**Algorithm 5.1:**

Given $Ax^0 = b$, $x^0 > 0$, $s^0 = Qx^0 + c^T - A^Ty^0 > 0$,

$\|X^0s^0 - z^0e\| \leq \alpha z^0$, and $z^0 = \frac{f(z^0) - d(z^0,y^0)}{n}$, where $\alpha = 1 - \frac{\sqrt{2}}{2}$;

set $k = 0$;

while $z^k \geq \epsilon$ do

begin

- let $\mu = z^k$ and $\lambda = (1 - \frac{\alpha}{\sqrt{n}})z^k$;
- let $\Delta x$ and $\Delta y$ solve (5.6);
- let $x^{k+1} = x^k + \Delta x$ and $y^{k+1} = y^k + \Delta y$;
- $k = k + 1$;

end.

The performance of Algorithm 5.1 results from the following corollary.

**Corollary**

The Algorithm terminates in $4n^{0.5}|\log(\frac{\epsilon}{\delta})|$ iterations and each iteration uses $n^3$ arithmetic operations.

Now we need to focus on addressing the remaining third question: how to obtain the initial solution pair?

**5.4. Setting Initial Solution Pair**

We noticed that several methods to obtain the “analytical center” [59] of a polyhedron are well illustrated in Todd and Burrell [62], Gonzaga [29], and Renegar [55]. Via those methods, QP can always be augmented to a related new QP problem.
with known "analytical center". Therefore, without losing generality, let the center $x^0$ of the feasible region of QP be known. Then $x^0$ should be positive and feasible for QP. However, the most attractive property of $x^0$ is that there exists a $y$ such that

$$X^0 A^T y = -e. \quad (5.17)$$

Now let

$$y^0 = \eta \bar{y} \quad \text{for some } \eta,$$

then from (5.17)

$$s^0 = Qx^0 + c^T - A^T y^0,$$

$$z^0 = \frac{e^T X^0 (Qx^0 + c^T)}{n} + \eta,$$

and

$$X^0 s^0 - z^0 e = X^0 (Qx^0 + c^T) + \eta e - z^0 e$$

$$= X^0 (Qx^0 + c^T) - \frac{e^T X^0 (Qx^0 + c^T)}{n} e.$$

Therefore, choose $\eta$ such that

$$\eta e > -X^0 (Qx^0 + c^T)$$

and

$$\eta \geq \frac{\|X^0 (Qx^0 + c^T) - (e^T X^0 (Qx^0 + c^T) / n) e\|}{\alpha} - \frac{e^T X^0 (Qx^0 + c^T)}{n}.$$

Then it can be verified that

$$\|X^0 s^0 - z^0 e\| \leq \alpha z^0, \quad \text{and } s^0 > 0.$$

Therefore, $x^0$ and $y^0$ can be used to initialize Algorithm 5.1.
5.5 A Safeguard Line Search Technique

Theoretically, both the Karmarkar’s projective and the path-following algorithms allow solutions moving in a small step-size. This restriction severely slows down the convergence of these algorithms in practice. In Karmarkar’s projective algorithm, one can use a line search technique to minimize the potential function, which significantly improves the practical efficiency of the projective algorithm [62]. Similarly, I propose a safeguard line search technique to overcome the small step-size difficulty in the path-following algorithm.

From the derivation of Lemma 5.2, we can see that the smaller the $\lambda$, the faster the solution convergence. Moreover, $\lambda$ only appears on the right-hand side of the system equation (5.6). These observations lead to the following safeguard line search technique to minimize $z^{k+1}$ at each iteration of Algorithm 5.1.

Algorithm 5.2:

Given $Ax^0 = b, x^0 > 0, s^0 = Qx^0 + e^T - A^Ty^0 > 0,$

$$\|X^0s^0 - z^0e\| \leq \alpha z^0, \text{ and } z^0 = f(x^0) - d(x^0, y^0), \text{ where } \alpha = 1 - \frac{\sqrt{2}}{2};$$

set $k = 0;$

while $z^k \geq \epsilon$ do

begin

let $\mu = z^k;$

via system (5.6), select $\lambda^* \in [0 \ (1 - \alpha/\sqrt{n})z^k]$ to minimize $z^{k+1}$

subject to the safeguard inequalities:

$$x^{k+1} > 0, s^{k+1} > 0, \text{ and } \|X^{k+1}s^{k+1} - z^{k+1}e\| \leq \alpha z^{k+1};$$

$k = k + 1;$

end.
Each step of the line search costs at most $O(n)$ arithmetic operations after solving system (5.6) against two right-hand vectors $(X^k)^{-1}e$ and $s^k$. More interestingly, since $\lambda^*$ makes the safeguard inequality $\|X^{k+1}s^{k+1} - z^{k+1}e\| \leq \alpha z^{k+1}$ binding, $\lambda^*$ can \textit{analytically} be obtained by solving a quadratic equation

$$\|X^{k+1}s^{k+1} - z^{k+1}e\| = \alpha z^{k+1}.$$  

Obviously, Algorithm 5.2 remains a polynomial-time algorithm with the same worst case bound of Algorithm 5.1. However, from my computational experience the line search technique significantly improves performance of the “center” path-following algorithm in practice. $\lambda$ can be chosen as a constant less than 1, resulting a constant convergence ratio in solving most of convex QP problems.

\textbf{5.6 Summary}

The interior ellipsoid algorithm for convex quadratic programming is further enhanced. This development is motivated by the recent “center” path-following algorithm for linear programming. The enhanced IE method for QP creates a sequence of dual as well as primal interior feasible points converging to the optimal solution point. At each iteration, the gap between the primal and dual objective values (or the complementary slackness value) is reduced at a global convergence ratio $(1 - \frac{1}{4\sqrt{n}})$, where $n$ is the number of variables in the convex QP problem. A line search technique is also incorporated into this algorithm to achieve practical efficiency.
Chapter 6 Conclusions

By extending Karmarkar's projective and the path-following polynomial-time algorithms for linear programming, I have developed two interior algorithms, the projective (PTIE) and the "center" path-following (PFIE) methods for solving convex quadratic and linearly constrained convex programming. The major theoretical features of these methods are:

1) The projective algorithm is a polynomial-time algorithm, with the computational complexity $O(L^2n^4)$ or $O(Ln^5)$, for convex quadratic programming. The path-following algorithm is also a polynomial-time algorithm, with the computational complexity $O(Ln^{3.5})$, for convex quadratic programming. Both algorithms are related to the trust region method for unconstrained optimization problems.

2) The projective algorithm solves linearly constrained convex programming in $O(Ln)$ iterations, and each iteration solves an equality constrained convex program—a continuous optimization problem. This polynomial bound results from the convexity invariance in projective transformation.

3) The projective algorithm is an extension of Karmarkar's polynomial-time algorithm for LP. My approach overcomes the difficulties found in Karmarkar's and Khachiyan's original polynomial algorithms, such as requiring prior knowledge of the optimal objective value and generating no optimal dual solutions unless the primal and dual programs are adjoined. The PTIE method also works exclusively in the LP standard form, thus bypassing explicit transformation to Karmarkar's LP canonical form.

4) The projective algorithm is similar to the ellipsoid method. At each iteration the potential function used to measure convergence of the primal solutions in the algorithm correctly represents the logarithmic volume of a dual ellipsoid that contains all the optimal dual solutions. As the potential
function declines, the volume of the dual ellipsoid monotonically shrinks to zero. This resemblance leads to a strong column eliminating theorem in determining the optimal basis for linear programming.

5) The path-following algorithm incorporates the “centering” method, the barrier function method, and the interior ellipsoid method. This algorithm creates a sequence of primal and dual interior feasible points converging to the optimal solution. At each iteration, the complementary slackness value, i.e., the objective gap between the primal and dual, is reduced at a global convergence ratio \( (1 - \frac{1}{4\sqrt{n}}) \), where \( n \) is the number of variables in the convex quadratic program.

6) A safeguard line search technique is developed in the path-following algorithm to relax the small-step-size restriction. This technique results faster convergence speed that is insensitive to the size of the QP problem.

7) While the theoretical results of the local convergence ratio for continuous optimization are well-developed, the state of the art of global convergence analysis is rather unsatisfactory. The global convergence ratio introduced in my approach brings two widely-used algorithm criteria—convergence ratio and computational complexity—into one volume. This ratio can be used to develop a new criterion for judging the efficiency of optimization algorithms globally.

In this dissertation, I also discussed how to efficiently implement these interior methods in practice. The new primal-dual method is implemented in solving linearly constrained convex programming as well as linear programming. The program generates the optimal solutions for both primal and dual problems. My approach is to narrow the primal-dual “gap” to the user-required tolerance in contrast to the traditional stopping criterion of the first-order condition. In addition, I introduced a large class of simplices and the associated tangent-plane move procedure. These moves solve each iteration approximately, and, in my numerical experiments, reduce the solution time significantly.
However, perhaps the most important feature of this method is that it provides a different framework for attacking mathematical programming that may lead to a family of algorithms for a variety of programming situations. In particular, the method offers an approach to general nonlinear programming. Further research can be initiated within this framework in areas such as nonlinearly constrained programming, large structured optimization, combinatorial optimization, and parallel computational processing.
REFERENCES


