Interior-Point Algorithm: Theory and Analysis

Yinyu Ye

June 1996, Revised January 1997

To my family and parents

Contents

Preface

On a sunny afternoon in 1984, one of my officemates told me that there would be a seminar given by N. Karmarkar, an AT&T scientist, on linear programming. At the time, my knowledge of linear programming was limited to one optimization course and one research project with Prof. David Luenberger in the Department of Engineering-Economic Systems, Stanford University. As a second-year Ph.D. student, I was familiar with economic terms like cost, supply, demand, and price, rather than mathematical terms like vertex, polyhedron, inequality, and duality.

That afternoon the Terman auditorium was packed. I could not find a seat and had to sit on the floor for about one and a half hours, where I saw Prof. George Dantzig and many other faculty members. I was not particular enthusiastic about the statement from the speaker that a new interior-point method would be 40 times faster than the simplex method, but I was amazed by the richness and applicability of linear programming as a whole. That was how and when I determined to devote my Ph.D. study to mathematical programming.

I immediately took a series of courses from Prof. Dantzig. In those courses I was fortunate to learn from many distinguished researchers. I also went to Cornell to work under the guidance of Prof. Michael Todd. Since then, my interest in linear programming has become stronger and my knowledge of interior-point algorithms has grown broader.

I decided to write a monograph about my understanding of interior-point algorithms and their complexities. I chose to highlight the underlying interior-point geometry, combinatorica, and potential theory for convex inequalities. I did not intend to cover the entire progress of linear programming and interior-point algorithms during the last decade in this write-up. For a complete survey, I refer the reader to several excellent articles or books by Goldfarb and Todd [151], Gonzaga [163], den Hertog [182], Nesterov and Nemirovskii [327], Roos, Terlaky and Vial [366], Terlaky [404], Todd [406, 412], Vanderbei [443], and Wright [457].

Information, supporting materials, and computer programs related to this book may be found at the following address on the World-Wide Web:

http://dollar.biz.uiowa.edu/col/ye/book.html

Please report any question, comment and error to the address:

yinyu-ye@uiowa.edu

I am specially indebted to Steve Benson and Andrew Prince, who made a careful and thorough reading of the manuscript and provided me numerous comments and suggestions on organizing the materials presented in this book. Special thanks also go to Erling Andersen, for giving me detailed feedback from a course he taught in Odense University using an early version of the manuscript, and to Nadine Castellano, for her proofreading of the manuscript. I wish to thank my colleagues Kurt Anstreicher, Dingzhu Du, Ken Kortanek, Tom Luo, Panos Pardalos, Florian Potra, Mike Todd, Steve Vavasis, and Guoliang Xue, for their support and advice on the process of writing the manuscript. I am grateful to the University of Iowa, for the environment and resources which made this effort possible. For their contribution to this effort, I acknowledge the students and other participants in the course 6K287 at the Department of Management Science. I also wish to thank the Institute of Applied Mathematics of Chinese Academy and Huazhong University of Science and Technology, for their support and assistance during my many visits there. It has been a pleasure to work with Jessica Downey and her colleagues of John Wiley & Sons involved in the production of this book.

My research efforts have been supported by the following grants and awards at various institutes: National Science Foundation Grants DDM-8922636, DDM-9207347 and DMI-9522507; the summer grants of the Iowa College of Business of Administration, the Obermann Fellowship, and the Center for Advanced Studies Interdisciplinary Research Grant at the University of Iowa; the Cornell Center for Applied Mathematics and the Advanced Computing Research Institute, a unit of the Cornell Theory Center, which receives major funding from the NSF and IBM Corporation, with additional support from New York State and members of its Corporate Research Institute; NSF Coop. Agr. No. CCR-8809615 at the Center for Research in Parallel Computation of Rice University; NSF Grant OSR-9350540 at the University of Vermont; the award of K. C. WONG Education Foundation at Hong Kong; and Dutch Organization for Scientific Research (NWO).

Finally, I thank my wife Daisun Zhou, and daughter Fei, for their constant love and support, and my parents Xienshu Ye and Chunxuan Wang, for their inspiration.

YINYU YE

Iowa City, 1996

List of Figures

Chapter 1

Introduction and Preliminaries

1.1 Introduction

Complexity theory is the foundation of computer algorithms. The goal of the theory is to develop criteria for measuring the effectiveness of various algorithms and the difficulty of various problems. The term "complexity" refers to the amount of resources required by a computation. In this book, running time or number of arithmetic operations is the major resource of interest.

Linear programming, hereafter LP, plays a very important role in complexity analysis. In one sense it is a continuous optimization problem in minimizing a linear objective function over a convex polyhedron; but it is also a combinatorial problem involving selecting an extreme point among a finite set of possible vertices. Businesses, large and small, use linear programming models to optimize communication systems, to schedule transportation networks, to control inventories, to plan investments, and to maximize productivity.

Linear inequalities define a polyhedron, properties of which have been studied by mathematicians for centuries. Ancient Chinese and Greeks studied calculating volumes of simple polyhedra in three-dimensional space. Fourier's fundamental research connecting optimization and inequalities dates back to the early 1800s. At the end of 19th century, Farkas and Minkowski began basic work on algebraic aspects of linear inequalities. In 1910 De La Vallée Poussin developed an algebraic technique for minimizing the infinity-norm of $b - Ax$ that can be viewed as a precursor of the simplex method. Beginning in the 1930s, such notable mathematicians as von Neumann, Kantorovich, and Koopmans studied mathematical economics based on linear inequalities. During World War II, it was observed that decisions involving the best movement of personnel and optimal allocation of resources could be posed and solved as linear programs. Linear programming began to assume its current popularity.

An optimal solution of a linear program always lies at a vertex of the feasible region, which itself is a polyhedron. Unfortunately, the number of vertices associated with a set of n inequalities in m variables can be exponential in the dimensions—in this case, up to $n!/m!(n-m)!$. Except for small values of m and n, this number is so large as to prevent examining all possible vertices for searching an optimal vertex.

The simplex method, invented in the mid-1940s by George Dantzig, is a procedure for examining optimal candidate vertices in an intelligent fashion. It constructs a sequence of adjacent vertices with improving values of the objective function. Thus, the method travels along edges of the polyhedron until it hits an optimal vertex. Improved in various way in the intervening four decades, the simplex method continues to be the workhorse algorithm for solving linear programming problems. On average, the number of vertices or iterations visited by the simplex method seems to be roughly linear in m and perhaps logarithmic n.

Although it performs well on average, the simplex method will indeed examine every vertex when applied to certain linear programs. Klee and Minty in 1972 gave such an example. These examples confirm that, in the worst case, the simplex method needs an exponential number of iterations to find the optimal solution. As interest in complexity theory grew, many researchers believed that a good algorithm should be polynomial—i.e., broadly speaking, the running time required to compute the solution should be bounded above by a polynomial in the "size," or the total data length, of the problem. Thus, the simplex method is not a polynomial algorithm.

In 1979, a new approach to linear programming, Khachiyan's ellipsoid method, received dramatic and widespread coverage in the international press. Khachiyan proved that the ellipsoid method, developed during the 1970s by other mathematicians, is a polynomial algorithm for linear programming under a certain computational model. It constructs a sequence of shrinking ellipsoids with two properties: the current ellipsoid always contains the optimal solution set, and each member of the sequence undergoes a guaranteed reduction in volume, so that the solution set is squeezed more tightly at each iteration.

The ellipsoid method was studied intensively by practitioners as well as theoreticians. Based on the expectation that a polynomial linear programming algorithm would be faster than the simplex method, it was a great disappointment that the best implementations of the ellipsoid method were not even close to being competitive. In contrast to the simplex method, the number of steps required for the ellipsoid method to terminate was almost always close to the worst case bound—whose value, although defined by a polynomial, is typically very large. Thus, after the dust eventually settled, the prevalent view among linear programming researchers was that Khachiyan had answered a major open question on the polynomiality of solving linear programs, but the simplex method remained the clear winner in practice.

This contradiction, the fact that an algorithm with the desirable theoretical property of polynomiality might nonetheless compare unfavorably with the (worst-case exponential) simplex method, set the stage for exciting new developments. It was no wonder, then, that the announcement by Karmarkar in 1984 of a new polynomial interior-point algorithm with the potential to dramatically improve the practical effectiveness of the simplex method made front-page news in major newspapers and magazines throughout the world.

Interior-point algorithms are continuous iterative algorithms. Computation experience with sophisticated procedures suggests that the number of iterations necessarily grows much more slowly than the dimension grows. Furthermore, they have an established worst-case polynomial iteration bound, providing the potential for dramatic improvement in computation effectiveness. The success of interior-point algorithms also brought much attention to complexity theory itself.

The goal of the book is to describe some of these recent developments and to suggest a few directions in which future progress might be made. The book is organized as follows. In Chapter 1, we discuss some necessary mathematical preliminaries. We also present several decision and optimization problems, models of computation, and several basic numerical procedures used throughout the text.

Chapter 2 is devoted to studying the geometry of inequalities and interior-point algorithms. At first glance, interior-point algorithms seem less geometric than the simplex or the ellipsoid methods. Actually, they also possess many rich geometric concepts. These concepts, such as "center," "volume," and "potential" of a polytope, are generally "non-combinatorial." These geometries are always helpful for teaching, learning, and research.

In Chapter 3 we present some basic algorithms to compute a so-called analytic center, or, equivalently, to maximize a potential or minimize a barrier function for a polytope. They are key elements underlying interior-point algorithms. Then, we present several interior-point linear programming algorithms in Chapter 4. It is impossible to list all the literature in this field. Here, we select five algorithms: Karmarkar's projective algorithm, the path-following algorithm, the potential reduction algorithm, the primal-dual algorithm including the predictor-corrector algorithm, and the affine scaling algorithm,

We analyze the worst-case complexity bound for interior-point algorithms in Chapter 5. The main issues are arithmetic operation, termination, and initialization techniques. We will use the real number computation model in our analysis because of the continuous nature of interior-point algorithms. We also compare the complexity theory with the convergence rate used in numerical analysis.

The worst-case complexity bound alone hardly serves as a practical criterion for judging the efficiency of algorithms. We will discuss a common phenomenon arising from using interior-point algorithms for solving optimization problems. It is often observed that effectiveness of an algorithm is dependent on the dimension or size of a problem instance as well as a parameter, called the "condition number," inherited in the problem. This condition number represents the degree of difficulty of the problem instance. For two problems having the same dimension but different condition numbers, an algorithm may have drastically different performances. This classification will help us to understand algorithm efficiency and possibly improve the condition and, therefore, improve the complexity of the problem. We present some condition-based complexity results for LP interior-point algorithms in Chapter 5.

While most of research has been focused on the worst-case performance of interior-point algorithms, many other complexity results were quietly established during the past several years. We try to cover these less-noticeable but significant results. In particular, we present some average and probabilistic complexity results in Chapter 6 and some asymptotic complexity (local convergence) results in Chapter 7. Average complexity bounds have been successfully established for the simplex method, and asymptotic or local convergence rates have been widely accepted by the numerical and continuous optimization community as major criteria in judging efficiency of iterative procedures.

Not only has the complexity of LP algorithms been significantly improved during the last decade, but also the problem domain solvable by interior-point algorithms has dramatically widened. We present complexity results for multi-objective programming, non-smooth convex programming, positive semi-definite programming and non-polyhedron optimization, and monotone complementarity in Chapter 8. We also discuss some complexity results for fractional programming and non-monotone linear complementarity, and approximation results for solving nonconvex optimization problems in Chapter 9.

Finally, we discuss major implementation issues in Chapter 10. It is common to have a gap between a theoretical algorithm and its practical implementation: theoretical algorithm makes sure that it works for all instances and never fails, while practical implementation emphasizes average performance and uses many clever "tricks" and ingenious techniques. In this chapter we discuss several effective implementation techniques frequently used in interior-point linear programming software, such as the presolver process, the sparse linear system, the high-order predictor-corrector, the homogeneous and self-dual formulation, and the optimal basis identification. Our objective is to provide *theoretical* justification for these techniques and to explain their practical pros and cons.

1.2 Mathematical Preliminaries

This section summarizes mathematical background material for linear algebra, linear programming, and nonlinear optimization.

1.2.1 Basic notations

The notation described below will be followed in general. There may be some deviation where appropriate.

By $\mathcal{R}$ we denote the set of real numbers. $\mathcal{R}_+$ denotes the set of nonnegative real numbers, and $\overset{\circ}{\mathcal{R}}_+$ denotes the set of positive numbers. For a natural number n, the symbol $\mathcal{R}^n$ ($\mathcal{R}^n_+$, $\overset{\circ}{\mathcal{R}}{}^n_+$) denotes the set of vectors with n components in $\mathcal{R}$ ($\mathcal{R}_+$, $\overset{\circ}{\mathcal{R}}_+$). We call $\overset{\circ}{\mathcal{R}}{}^n_+$ the interior of $\mathcal{R}^n_+$.

Addition of vectors and multiplication of vectors with scalars are standard. The vector inequality $x \geq y$ means $x_j \geq y_j$ for $j = 1, 2, ..., n$. Zero represents a vector whose entries are all zeros and e represents a vector whose entries are all ones, where their dimensions may vary according to other vectors in expressions. A vector is always considered as a column vector, unless otherwise stated. Upper-case letters will be used to represent matrices. Greek letters will typically be used to represent scalars.

The superscript "T" denotes transpose operation. The inner product in $\mathcal{R}^n$ is defined as follows:

$$\langle x, y \rangle := x^T y = \sum_{j=1}^{n} x_j y_j \quad \text{for} \quad x, y \in \mathcal{R}^n.$$

The l_2 norm of a vector x is given by

$$\|x\|_2 = \sqrt{x^T x},$$

and the l_∞ norm is

$$\|x\|_\infty = \max\{|x_1|, |x_2|, ..., |x_n|\}.$$

In general, the p norm is

$$\|x\|_p = \left(\sum_1^n |x_j|^p\right)^{1/p}, \quad p = 1, 2, ...$$

The dual of the p norm, denoted by $\|.\|^*$, is the q norm, where

$$\frac{1}{p} + \frac{1}{q} = 1.$$

In this book, $\|.\|$ generally represents the l_2 norm.

For convenience, we sometime write a column vector x as

$$x = (x_1; x_2; \ldots; x_n)$$

and a row vector as

$$x = (x_1, x_2, \ldots, x_n).$$

For natural numbers m and n, $\mathcal{R}^{m\times n}$ denotes the set of real matrices with m rows and n columns. For $A \in \mathcal{R}^{m\times n}$, we assume that the row index set of A is $\{1, 2, ..., m\}$ and the column index set is $\{1, 2, ..., n\}$. The ith row of A is denoted by $a_{i.}$ and the jth column of A is denoted by $a_{.j}$; the i and jth component of A is denoted by a_{ij}. If I is a subset of the row index set and J is a subset of the column index set, then A_I denotes the submatrix of A whose rows belong to I, A_J denotes the submatrix of A whose columns belong to J, A_{IJ} denotes the submatrix of A induced by those components of A whose indices belong to I and J, respectively.

The identity matrix is denoted by I. The null space of A is denoted $\mathcal{N}(A)$ and the range of A is $\mathcal{R}(A)$. The determinant of an $n \times n$-matrix A is denoted by $\det(A)$. The trace of A, denoted by $\mathrm{tr}(A)$, is the sum of the diagonal entries in A. The operator norm of A, denoted by $\|A\|$, is

$$\|A\|^2 := \max_{0\neq x \in \mathcal{R}^n} \frac{\|Ax\|^2}{\|x\|^2}.$$

For a vector $x \in \mathcal{R}^n$, the upper-case X represents a diagonal matrix in $\mathcal{R}^{n\times n}$ whose diagonal entries are the entries of x, i.e.,

$$X = \mathrm{diag}(x).$$

A matrix $Q \in \mathcal{R}^{n\times n}$ is said to be positive definite (PD), denoted by $Q \succ 0$, if

$$x^T Qx > 0, \quad \text{for all} \quad x \neq 0,$$

and positive semi-definite (PSD), denoted by $Q \succeq 0$, if

$$x^T Qx \geq 0, \quad \text{for all} \quad x.$$

If $Q \succ 0$, then $-Q$ is called negative definite (ND), denoted by $Q \prec 0$; if $Q \succeq 0$, then $-Q$ is called negative semi-definite (NSD), denoted by $Q \preceq 0$. If Q is symmetric, then its eigenvalues are all real numbers; furthermore, Q is PSD if and only if all its eigenvalues are non-negative, and Q is PD if and only if all its eigenvalue are positive. Given a PD matrix Q we can define a Q-norm , $\|.\|_Q$, for vector x as

$$\|x\|_Q = \sqrt{x^T Qx} \ .$$

$\mathcal{M}^n$ denotes the space of symmetric matrices in $\mathcal{R}^{n\times n}$. The inner product in $\mathcal{M}^n$ is defined as follows:

$$\langle X, Y\rangle := X \bullet Y = \text{tr}X^T Y = \sum_{i,j} X_{i,j}Y_{i,j} \quad \text{for} \quad X, Y \in \mathcal{M}^n.$$

This is a generalization of the vector inner product to matrices. The matrix norm associated with the inner product is called *Frobenius norm*:

$$\|X\|_f = \sqrt{\text{tr}X^T X} \ .$$

$\mathcal{M}^n_+$ denote the set of positive semi-definite matrices in $\mathcal{M}^n$. $\overset{\circ}{\mathcal{M}}{}^n_+$ denotes the set of positive definite matrices in $\mathcal{M}^n$. We call $\overset{\circ}{\mathcal{M}}{}^n_+$ the interior of $\mathcal{M}^n_+$.

$\{x^k\}_0^\infty$ is an ordered sequence $x^0, x^1, x^2, ..., x^k, ...$ A sequence $\{x^k\}_0^\infty$ is convergent to $\bar{x}$, denoted $x^k \to \bar{x}$, if

$$\|x^k - \bar{x}\| \to 0.$$

A point x is a limit point of $\{x^k\}_0^\infty$ if there is a subsequence of $\{x^k\}$ convergent to x.

If $g(x) \geq 0$ is a real valued function of a real nonnegative variable, the notation $g(x) = O(x)$ means that $g(x) \leq \bar{c}x$ for some constant $\bar{c}$; the notation $g(x) = \Omega(x)$ means that $g(x) \geq \underline{c}x$ for some constant $\underline{c}$; the notation $g(x) = \theta(x)$ means that $\underline{c}x \leq g(x) \leq \bar{c}x$. Another notation is $g(x) = o(x)$, which means that $g(x)$ goes to zero faster than x does:

$$\lim_{x\to 0} \frac{g(x)}{x} = 0.$$

1.2.2 Convex sets

If x is a member of the set Ω, we write $x \in \Omega$; if y is not a member of Ω, we write $y \notin \Omega$. The union of two sets S and T is denoted $S \cup T$; the intersection of them is denoted $S \cap T$. A set can be specified in the form $\Omega = \{x : P(x)\}$ as the set of all elements satisfying property P.

For $y \in \mathcal{R}^n$ and $\epsilon > 0$, $B(y, \epsilon) = \{x : \|x - y\| \leq \epsilon\}$ is the ball or sphere of radius ϵ with center y. In addition, for a positive definite matrix Q of dimension n, $E(y, Q) = \{x : (x - y)^T Q(x - y) \leq 1\}$ is called an *ellipsoid.* The vector y is the center of $E(y, Q)$. When Q is diagonal, $E(y, Q)$ is called a *coordinate-aligned ellipsoid* (Figure 1.1).

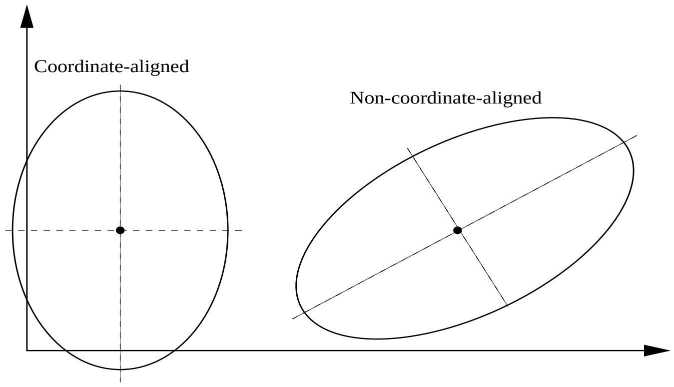

Figure 1.1. Coordinate-aligned and non-coordinate-aligned ellipsoids.

A set Ω is closed if $x^k \to x$, where $x^k \in \Omega$, implies $x \in \Omega$. A set Ω is open if around every point $y \in \Omega$ there is a ball that is contained in Ω, i.e., there is an $\epsilon > 0$ such that $B(y, \epsilon) \subset \Omega$. A set is bounded if it is contained within a ball with finite radius. A set is compact if it is both closed and bounded. The (topological) interior of any set Ω, denoted $\overset{\circ}{\Omega}$, is the set of points in Ω which are the centers of some balls contained in Ω. The closure of Ω, denoted $\hat{\Omega}$, is the smallest closed set containing Ω. The boundary of Ω is the part of $\hat{\Omega}$ that is not in $\overset{\circ}{\Omega}$.

A set C is said to be convex if for every $x^1, x^2 \in C$ and every real number α, $0 < \alpha < 1$, the point $\alpha x^1 + (1 - \alpha)x^2 \in C$. The convex hull of a set Ω is the intersection of all convex sets containing Ω.

A set C is a cone if $x \in C$ implies $\alpha x \in C$ for all $\alpha > 0$. A cone that is also convex is a convex cone. For a cone $C \subset \mathcal{E}$, the dual of C is the cone

$$C^* := \{y : \langle x, y \rangle \geq 0 \quad \text{for all} \quad x \in C\},$$

where $\langle \cdot, \cdot \rangle$ is an inner product operation for space $\mathcal{E}$.

Example 1.1 *The n-dimensional non-negative orthant, $\mathcal{R}^n_+ = \{x \in \mathcal{R}^n : x \geq 0\}$, is a convex cone. The dual of the cone is also $\mathcal{R}^n_+$; it is self-dual.*

Example 1.2 *The set of all positive semi-definite matrices in $\mathcal{M}^n$, $\mathcal{M}^n_+$, is a convex cone, called the* positive semi-definite matrix cone*. The dual of the cone is also $\mathcal{M}^n_+$; it is self-dual.*

Example 1.3 *The set $\{(t; x) \in \mathcal{R}^{n+1} : t \geq \|x\|\}$ is a convex cone in $\mathcal{R}^{n+1}$, called the* second-order cone*. The dual of the cone is also the second-order cone in $\mathcal{R}^{n+1}$; it is self-dual.*

A cone C is (convex) polyhedral if C can be represented by

$$C = \{x : Ax \leq 0\}$$

for some matrix A (Figure 1.2).

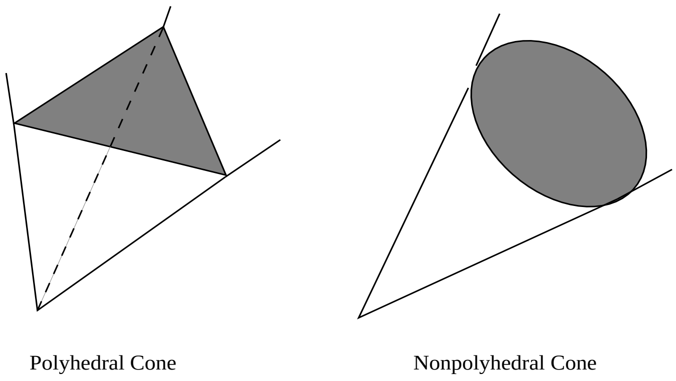

Figure 1.2. Polyhedral and nonpolyhedral cones.

Example 1.4 *The non-negative orthant is a polyhedral cone, and neither the positive semi-definite matrix cone nor the second-order cone is polyhedral.*

It has been proved that for cones the concepts of "polyhedral" and "finitely generated" are equivalent according to the following theorem.

Theorem 1.1 *A convex cone C is polyhedral if and only if it is finitely generated, that is, the cone is generated by a finite number of vectors $b_1,...,b_m$:*

$$C = cone(b_1, ..., b_m) := \left\{ \sum_{i=1}^{m} b_i y_i : \ y_i \geq 0, \ i = 1, ..., m \right\}.$$

The following theorem states that a polyhedral cone can be generated by a set of basic directional vectors.

Theorem 1.2 *(Carathéodory's theorem) Let convex polyhedral cone $C = cone(b_1, ..., b_m)$ and $x \in C$. Then, $x \in cone(b_{i_1}, ..., b_{i_d})$ for some linearly independent vectors $b_{i_1},...,b_{i_d}$ chosen from $b_1,...,b_m$.*

Example 1.5 *The following polyhedral cone of $\mathcal{R}^2$ in Figure 1.3 can be represented as either*

$$C = \left\{ \begin{pmatrix} 1 \\ 2 \end{pmatrix} y_1 + \begin{pmatrix} 2 \\ 1 \end{pmatrix} y_2 : \ y_1, y_2 \geq 0 \right\}$$

or

$$C = \left\{ x \in \mathcal{R}^2 : \ \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix} x \leq 0 \right\}.$$

The two vectors in the former representation are two basic directional vectors.

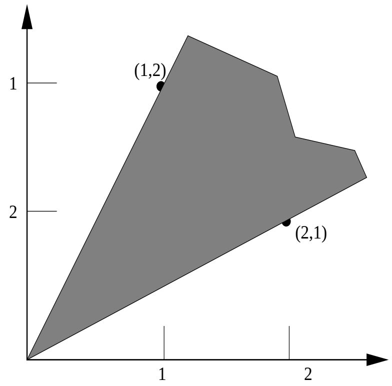

Figure 1.3. Representations of a polyhedral cone.

The most important type of convex set is a hyperplane. Hyperplanes dominate the entire theory of optimization. Let a be a nonzero n-dimensional vector, and let b be a real number. The set

$$H = \{x \in \mathcal{R}^n : a^T x = b\}$$

is a hyperplane in $\mathcal{R}^n$ (Figure 1.4). Relating to hyperplane, positive and negative closed half spaces are given by

$$H_+ = \{x : a^T x \geq b\}$$

$$H_- = \{x : a^T x \leq b\}.$$

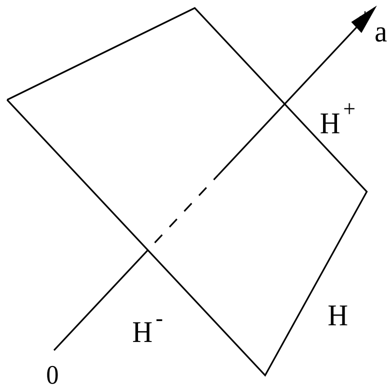

Figure 1.4. A hyperplane and half-spaces.

A set which can be expressed as the intersection of a finite number of closed half spaces is said to be a convex *polyhedron*:

$$P = \{x : Ax \leq b\}.$$

A bounded polyhedron is called *polytope*.

Let P be a polyhedron in $\mathcal{R}^n$, F is a face of P if and only if there is a vector c for which F is the set of points attaining max $\{c^T x : \ x \in P\}$ provided the this maximum is finite. A polyhedron has only finite many faces; each face is a nonempty polyhedron.

The most important theorem about the convex set is the following separating theorem (Figure 1.5).

Theorem 1.3 *(Separating hyperplane theorem) Let $C \subset \mathcal{E}$, where $\mathcal{E}$ is either $\mathcal{R}^n$ or $\mathcal{M}^n$, be a closed convex set and let y be a point exterior to C. Then there is a vector $a \in \mathcal{E}$ such that*

$$\langle a, y \rangle < \inf_{x \in C} \langle a, x \rangle.$$

The geometric interpretation of the theorem is that, given a convex set C and a point y outside of C, there is a hyperplane containing y that contains C in one of its open half spaces.

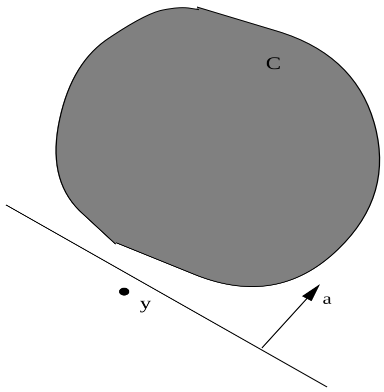

Figure 1.5. Illustration of the separating hyperplane theorem; an exterior point y is separated by a hyperplane from a convex set C.

Example 1.6 *Let C be a unit circle centered at the point $(1;1)$. That is, $C = \{x \in \mathcal{R}^2 : (x_1 - 1)^2 + (x_2 - 1)^2 \leq 1\}$. If $y = (2;0)$, $a = (-1;1)$ is a separating hyperplane vector. If $y = (0;-1)$, $a = (0;1)$ is a separating hyperplane vector. It is worth noting that these separating hyperplanes are not unique.*

We use the notation $\mathcal{E}$ to represent either $\mathcal{R}^n$ or $\mathcal{M}^n$, depending on the context, throughout this book, because all our decision and optimization problems take variables from one or both of these two vector spaces.

1.2.3 Real functions

The real function $f(x)$ is said to be continuous at x if $x^k \to x$ implies $f(x^k) \to f(x)$. The real function $f(x)$ is said to be continuous on set $\Omega \subset \mathcal{E}$, where recall that $\mathcal{E}$ is either $\mathcal{R}^n$ or $\mathcal{M}^n$, if $f(x)$ is continuous at x for every $x \in \Omega$. A continuous function f defined on a compact set (bounded and closed) $\Omega \subset \mathcal{E}$ has a minimizer in Ω; that is, there is an $x^* \in \Omega$ such that for all $x \in \Omega$, $f(x) \geq f(x^*)$. This result is called the *Weierstrass theorem.*

A function $f(x)$ is called homogeneous of degree k if $f(\alpha x) = \alpha^k f(x)$ for all $\alpha \geq 0$.

Example 1.7 *Let $c \in \mathcal{R}^n$ be given and $x \in \overset{\circ}{\mathcal{R}}{}^n_+$. Then $c^T x$ is homogeneous of degree 1 and*

$$\mathcal{P}(x) = n \log(c^T x) - \sum_{j=1}^{n} \log x_j$$

is homogeneous of degree 0*, where* $\log$ *is the natural logarithmic function. Let* $C \in \mathcal{M}^n$ *be given and* $X \in \overset{\circ}{\mathcal{M}}{}^n_+$. *Then* $x^T C x$ *is homogeneous of degree* 2, $C \bullet X$ *and* $\det(X)$ *are homogeneous of degree* 1*, and*

$$\mathcal{P}(X) = n \log(C \bullet X) - \log \det(X)$$

is homogeneous of degree 0.

A set of real-valued function $f_1, f_2, ..., f_m$ defined on $\mathcal{R}^n$ can be written as a single vector function $f = (f_1, f_2, ..., f_m)^T \in \mathcal{R}^m$. If f has continuous partial derivatives of order p, we say $f \in C^p$. The gradient vector of a real-valued function $f \in C^1$ is a row vector

$$\nabla f(x) = \{\partial f / \partial x_i\}, \quad \text{for} \quad i = 1, ..., n.$$

If $f \in C^2$, we define the Hessian of f to be the n-dimensional symmetric matrix

$$\nabla^2 f(x) = \left\{ \frac{\partial^2 f}{\partial x_i \partial x_j} \right\} \quad \text{for} \quad i = 1, ..., n; \; j = 1, ..., n.$$

If $f = (f_1, f_2, ..., f_m)^T \in \mathcal{R}^m$, the Jacobian matrix of f is

$$\nabla f(x) = \begin{pmatrix} \nabla f_1(x) \\ ... \\ \nabla f_m(x) \end{pmatrix}.$$

f is a (continuous) convex function if and only if for $0 \leq \alpha \leq 1$,

$$f(\alpha x + (1 - \alpha) y) \leq \alpha f(x) + (1 - \alpha) f(y).$$

f is a (continuous) quasi-convex function if and only if for $0 \leq \alpha \leq 1$,

$$f(\alpha x + (1 - \alpha) y) \leq \max[f(x), f(y)].$$

Thus, a convex function is a quasi-convex function. The level set of f is given by

$$L(z) = \{x : \quad f(x) \leq z\}.$$

f is a quasi-convex function implies that the level set of f is convex for any given z (see Exercise 1.9).

A group of results that are used frequently in analysis are under the heading of *Taylor's theorem* or the mean-value theorem. The theorem establishes the linear and quadratic approximations of a function.

Theorem 1.4 *(Taylor expansion) Let $f \in C^1$ be in a region containing the line segment $[x, y]$. Then there is a α, $0 \leq \alpha \leq 1$, such that*

$$f(y) = f(x) + \nabla f(\alpha x + (1-\alpha)y)(y - x).$$

Furthermore, if $f \in C^2$ then there is a α, $0 \leq \alpha \leq 1$, such that

$$f(y) = f(x) + \nabla f(x)(y-x) + (1/2)(y-x)^T \nabla^2 f(\alpha x + (1-\alpha)y)(y-x).$$

We also have several propositions for real functions. The first indicates that the linear approximation of a convex function is a under-estimate.

Proposition 1.5 *Let $f \in C^1$. Then f is convex over a convex set Ω if and only if*

$$f(y) \geq f(x) + \nabla f(x)(y-x)$$

for all x, $y \in \Omega$.

The following proposition states that the Hessian of a convex function is positive semi-definite.

Proposition 1.6 *Let $f \in C^2$. Then f is convex over a convex set Ω if and only if the Hessian matrix of f is positive semi-definite throughout Ω.*

1.2.4 Inequalities

There are several important inequalities that are frequently used in algorithm design and complexity analysis.

Cauchy-Schwarz: given $x, y \in \mathcal{R}^n$, then

$$x^T y \leq \|x\| \|y\|.$$

Arithmetic-geometric mean: given $x \in \mathcal{R}^n_+$,

$$\frac{\sum x_j}{n} \geq \left(\prod x_j\right)^{1/n}.$$

Harmonic: given $x \in \overset{\circ}{\mathcal{R}}{}^n_+$,

$$\left(\sum x_j\right)\left(\sum 1/x_j\right) \geq n^2.$$

Hadamard: given $A \in \mathcal{R}^{m \times n}$ with columns $a_1, a_2, ..., a_n$, then

$$\sqrt{\det(A^T A)} \leq \prod \|a_j\| .$$

1.3 Decision and Optimization Problems

A decision or optimization problem has a form that is usually characterized by the decision variables and the constraints. A problem, $\mathcal{P}$, consists of two sets, data set $\mathcal{Z}_p$ and solution set $\mathcal{S}_p$. In general, $\mathcal{S}_p$ can be implicitly defined by the so-called optimality conditions. The solution set may be empty, i.e., problem $\mathcal{P}$ may have no solution.

In what follows, we list several decision and optimization problems. More problems will be listed later when we address them.

1.3.1 System of linear equations

Given $A \in \mathcal{R}^{m\times n}$ and $b \in \mathcal{R}^m$, the problem is to solve m linear equations for n unknowns:

$$Ax = b.$$

The data and solution sets are

$$\mathcal{Z}_p = \{A \in \mathcal{R}^{m\times n}, b \in \mathcal{R}^m\} \quad \text{and} \quad \mathcal{S}_p = \{x \in \mathcal{R}^n : Ax = b\}.$$

$\mathcal{S}_p$ in this case is an affine set. Given an x, one can easily check to see if x is in $\mathcal{S}_p$ by a matrix-vector multiplication and a vector-vector comparison. We say that a solution of this problem is easy to recognize.

To highlight the analogy with the theories of linear inequalities and linear programming, we list several well-known results of linear algebra. The first theorem provides two basic representations, the null and row spaces, of a linear subspaces.

Theorem 1.7 *Each linear subspace of $\mathcal{R}^n$ is generated by finitely many vectors, and is also the intersection of finitely many linear hyperplanes; that is, for each linear subspace of L of $\mathcal{R}^n$ there are matrices A and C such that $L = \mathcal{N}(A) = \mathcal{R}(C)$.*

The following theorem was observed by Gauss. It is sometimes called the *fundamental theorem* of linear algebra. It gives an example of a characterization in terms of necessary and sufficient conditions, where necessity is straightforward, and sufficiency is the key of the characterization.

Theorem 1.8 *Let $A \in \mathcal{R}^{m\times n}$ and $b \in \mathcal{R}^m$. The system $\{x : Ax = b\}$ has a solution if and only if that $A^T y = 0$ implies $b^T y = 0$.*

A vector y, with $A^T y = 0$ and $b^T y = 1$, is called an *infeasibility certificate* for the system $\{x : Ax = b\}$.

Example 1.8 *Let $A = (1; -1)$ and $b = (1; 1)$. Then, $y = (1/2; 1/2)$ is an* infeasibility certificate *for $\{x : Ax = b\}$.*

1.3.2 System of nonlinear equations

Given $f(x) : \mathcal{R}^n \to \mathcal{R}^m$, the problem is to solve m equations for n unknowns:

$$f(x) = 0.$$

The "data" and solution sets are

$$\mathcal{Z}_p = \{f, \nabla f, \ldots\} \quad \text{and} \quad \mathcal{S}_p = \{x \in \mathcal{R}^n : f(x) = 0\}.$$

In contrast to that of a linear system, $\mathcal{Z}_p$ is called an *oracle* or "black-box": for any given input x, it outputs the function values, the Jacobian matrix, and/or other finite numerical values about the function. In computer programs an oracle is a set of subroutines to perform function evaluation for a given input x. These subroutines might use a computer build-in function, a finite-element model, or a numerical partial-differential-equation procedure. In our computation models, we count one value evaluation one operation.

1.3.3 Linear least-squares problem

Given $A \in \mathcal{R}^{m\times n}$ and $c \in \mathcal{R}^n$, the system of equations $A^T y = c$ may be over-determined or have no solution. Such a case usually occurs when the number of equations is greater than the number of variables. Then, the problem is to find an $y \in \mathcal{R}^m$ or $s \in \mathcal{R}(A^T)$ such that $\|A^T y - c\|$ or $\|s - c\|$ is minimized. We can write the problem in the following format:

$$\begin{array}{lll} (LS) & \text{minimize} & \|A^T y - c\|^2 \\ & \text{subject to} & y \in \mathcal{R}^m, \end{array}$$

or

$$\begin{array}{lll} (LS) & \text{minimize} & \|s - c\|^2 \\ & \text{subject to} & s \in \mathcal{R}(A^T). \end{array}$$

In the former format, the term $\|A^T y - c\|^2$ is called the *objective function*, y is called the *decision variable*. Since y can be any point in $\mathcal{R}^m$, we say this (optimization) problem is *unconstrained*. The data and solution sets are

$$\mathcal{Z}_p = \{A \in \mathcal{R}^{m\times n}, c \in \mathcal{R}^n\}$$

and

$$\mathcal{S}_p = \{y \in \mathcal{R}^m : \|A^T y - c\|^2 \le \|A^T x - c\|^2 \quad \text{for every} \quad x \in \mathcal{R}^m\}.$$

Given a y, to see if $y \in \mathcal{S}_p$ is as the same as the original problem. However, from a projection theorem in linear algebra, the solution set can be characterized and represented as

$$\mathcal{S}_p = \{y \in \mathcal{R}^m : AA^T y = Ac\},$$

which becomes a system of linear equations and always has a solution. The vector $s = A^T y = A^T(AA^T)^+ Ac$ is the projection of c onto the range of A^T, where AA^T is called *normal matrix* and $(AA^T)^+$ is called *pseudo-inverse*. If A has full row rank then $(AA^T)^+ = (AA^T)^{-1}$, the standard inverse of full rank matrix AA^T. If A is not of full rank, neither is AA^T and $(AA^T)^+ AA^T x = x$ only for $x \in \mathcal{R}(A^T)$.

The vector $c - A^T y = (I - A^T(AA^T)^+ A)c$ is the projection of c onto the null space of A. It is the solution of the following least-squares problem:

$$\begin{array}{lll} (LS) & \text{minimize} & \|x - c\|^2 \\ & \text{subject to} & x \in \mathcal{N}(A). \end{array}$$

In the full rank case, both matrices $A^T(AA^T)^{-1}A$ and $I - A^T(AA^T)^{-1}A$ are called *projection matrices*. These symmetric matrices have several desired properties (see Exercise 1.14).

The linear least-squares problem is a basic problem solved on each iteration of any interior-point algorithm.

1.3.4 System of linear inequalities

Given $A \in \mathcal{R}^{m\times n}$ and $b \in \mathcal{R}^m$, the problem is to find a solution $x \in \mathcal{R}^n$ satisfying $Ax \le b$ or prove that the solution set is empty. The inequality problem includes other forms such as finding an x that satisfies the combination of linear equations $Ax = b$ and inequalities $x \ge 0$. The data and solution sets of the latter are

$$\mathcal{Z}_p = \{A \in \mathcal{R}^{m\times n}, b \in \mathcal{R}^m\} \quad \text{and} \quad \mathcal{S}_p = \{x \in \mathcal{R}^n : Ax = b,\ x \ge 0\}.$$

Traditionally, a point in $\mathcal{S}_p$ is called a *feasible solution*, and a strictly positive point in $\mathcal{S}_p$ is called a *strictly feasible* or *interior feasible solution*.

The following results are Farkas' lemma and its variants.

Theorem 1.9 *(Farkas' lemma) Let $A \in \mathcal{R}^{m\times n}$ and $b \in \mathcal{R}^m$. Then, the system $\{x : \ Ax = b,\ x \ge 0\}$ has a feasible solution x if and only if that $A^T y \le 0$ implies $b^T y \le 0$.*

A vector y, with $A^T y \le 0$ and $b^T y = 1$, is called a (*primal*) infeasibility certificate for the system $\{x : \ Ax = b,\ x \ge 0\}$. Geometrically, Farkas' lemma means that if a vector $b \in \mathcal{R}^m$ does not belong to the cone generated by $a_{.1}, ..., a_{.n}$, then there is a hyperplane separating b from cone$(a_{.1}, ..., a_{.n})$.

Example 1.9 *Let $A = (1, 1)$ and $b = -1$. Then, $y = -1$ is an infeasibility certificate for $\{x : \ Ax = b,\ x \ge 0\}$.*

Theorem 1.10 *(Farkas' lemma variant) Let $A \in \mathcal{R}^{m\times n}$ and $c \in \mathcal{R}^n$. Then, the system $\{y:\ A^Ty \leq c\}$ has a solution y if and only if that $Ax = 0$ and $x \geq 0$ imply $c^Tx \geq 0$.*

Again, a vector $x \geq 0$, with $Ax = 0$ and $c^Tx = -1$, is called a (*dual*) infeasibility certificate for the system $\{y:\ A^Ty \leq c\}$.

Example 1.10 *Let $A = (1;-1)$ and $c = (1;-2)$. Then, $x = (1;1)$ is an infeasibility certificate for $\{y:\ A^Ty \leq c\}$.*

We say $\{x:\ Ax = b,\ x \geq 0\}$ or $\{y:\ A^Ty \leq c\}$ is *approximately feasible* in the sense that we have an approximate solution to the equations and inequalities. In this case we can show that any certificate proving their infeasibility must have large norm. Conversely, if $\{x:\ Ax = b,\ x \geq 0\}$ or $\{y:\ A^Ty \leq c\}$ is "approximately infeasible" in the sense that we have an approximate certificate in Farkas' lemma, then any feasible solution must have large norm. The details can be found in Exercise 5.12.

Example 1.11 *Given $\epsilon > 0$ but small. Let $A = (1,1)$ and $b = -\epsilon$. Then, $x = (0;0)$ is approximately feasible for $\{x:\ Ax = b,\ x \geq 0\}$, and the infeasibility certificate $y = -1/\epsilon$ has a large norm.*

Let $A = (1;-1)$ and $c = (1;-1-\epsilon)$. Then, $y = 1$ is approximately feasible for $\{y:\ A^Ty \leq c\}$, and the infeasibility certificate $x = (1/\epsilon;1/\epsilon)$ has a large norm.

1.3.5 Linear programming (LP)

Given $A \in \mathcal{R}^{m\times n}$, $b \in \mathcal{R}^m$ and $c, l, u \in \mathcal{R}^n$, the linear programming (LP) problem is the following optimization problem:

$$\begin{array}{ll} \text{minimize} & c^Tx \\ \text{subject to} & Ax = b,\ l \leq x \leq u, \end{array}$$

where some elements in l may be $-\infty$ meaning that the associated variables are unbounded from below, and some elements in u may be ∞ meaning that the associated variables are unbounded from above. If a variable is unbounded either from below or above, then it is called a "*free*" variable

The standard form linear programming problem is given below, which we will use throughout this book:

$$\begin{array}{lll} (LP) & \text{minimize} & c^Tx \\ & \text{subject to} & Ax = b,\ x \geq 0. \end{array}$$

The linear function c^Tx is called the *objective function*, and x is called the *decision variables*. In this problem, $Ax = b$ and $x \geq 0$ enforce *constraints* on the selection of x. The set $\mathcal{F}_p = \{x : Ax = b, x \geq 0\}$ is called *feasible set* or *feasible region*. A point $x \in \mathcal{F}_p$ is called a *feasible point*, and a feasible point x^* is called an *optimal solution* if $c^T x^* \leq c^T x$ for all feasible points x. If there is a sequence $\{x^k\}$ such that x^k is feasible and $c^T x^k \to -\infty$, then (LP) is said to be *unbounded*.

The data and solution sets for (LP), respectively, are

$$\mathcal{Z}_p = \{A \in \mathcal{R}^{m \times n}, b \in \mathcal{R}^m, c \in \mathcal{R}^n\}$$

and

$$\mathcal{S}_p = \{x \in \mathcal{F}_p : c^T x \leq c^T y, \quad \text{for every} \quad y \in \mathcal{F}_p\}.$$

Again, given an x, to see if $x \in \mathcal{S}_p$ is as difficult as the original problem. However, due to the duality theorem, we can simplify the representation of the solution set significantly.

With every (LP), another linear program, called the dual (LD), is the following problem:

$$\begin{array}{lll} (LD) & \text{maximize} & b^T y \\ & \text{subject to} & A^T y + s = c, \; s \geq 0, \end{array}$$

where $y \in \mathcal{R}^m$ and $s \in \mathcal{R}^n$. The components of s are called *dual slacks*. Denote by $\mathcal{F}_d$ the sets of all (y, s) that are feasible for the dual. We see that (LD) is also a linear programming problem where y is a "free" vector.

The following theorems give us an important relation between the two problems.

Theorem 1.11 *(Weak duality theorem) Let $\mathcal{F}_p$ and $\mathcal{F}_d$ be non-empty. Then,*

$$c^T x \geq b^T y \quad \textit{where} \quad x \in \mathcal{F}_p, \; (y, s) \in \mathcal{F}_d.$$

This theorem shows that a feasible solution to either problem yields a bound on the value of the other problem. We call $c^T x - b^T y$ the *duality gap*. From this we have important results.

Theorem 1.12 *(Strong duality theorem) Let $\mathcal{F}_p$ and $\mathcal{F}_d$ be non-empty. Then, x^* is optimal for (LP) if and only if the following conditions hold:*

i) $x^* \in \mathcal{F}_p$;

ii) *there is* $(y^*, s^*) \in \mathcal{F}_d$;

iii) $c^T x^* = b^T y^*$.

Theorem 1.13 *(LP duality theorem) If (LP) and (LD) both have feasible solutions then both problems have optimal solutions and the optimal objective values of the objective functions are equal.*

If one of (LP) or (LD) has no feasible solution, then the other is either unbounded or has no feasible solution. If one of (LP) or (LD) is unbounded then the other has no feasible solution.

The above theorems show that if a pair of feasible solutions can be found to the primal and dual problems with equal objective values, then these are both optimal. The converse is also true; there is no "gap." From this condition, the solution set for (LP) and (LD) is

$$\mathcal{S}_p = \left\{ (x,y,s) \in (\mathcal{R}^n_+, \mathcal{R}^m, \mathcal{R}^n_+) : \begin{array}{rcl} c^T x - b^T y & = & 0 \\ Ax & = & b \\ -A^T y - s & = & -c \end{array} \right\}, \tag{1.1}$$

which is a system of linear inequalities and equations. Now it is easy to verify whether or not a pair (x, y, s) is optimal.

For feasible x and (y, s), $x^T s = x^T(c - A^T y) = c^T x - b^T y$ is called the *complementarity gap*. If $x^T s = 0$, then we say x and s are complementary to each other. Since both x and s are nonnegative, $x^T s = 0$ implies that $x_j s_j = 0$ for all $j = 1, \ldots, n$. Thus, one equation plus nonnegativity are transformed into n equations. Equations in (1.1) become

$$\begin{array}{rcl} Xs & = & 0 \\ Ax & = & b \\ -A^T y - s & = & -c. \end{array} \tag{1.2}$$

This system has total $2n + m$ unknowns and $2n + m$ equations including n nonlinear equations.

The following theorem plays an important role in analyzing LP interior-point algorithms. It give a unique partition of the LP variables in terms of complementarity.

Theorem 1.14 *(Strict complementarity theorem) If (LP) and (LD) both have feasible solutions then both problems have a pair of strictly complementary solutions $x^* \geq 0$ and $s^* \geq 0$ meaning*

$$X^* s^* = 0 \quad \textit{and} \quad x^* + s^* > 0.$$

Moreover, the supports

$$P^* = \{j : \ x^*_j > 0\} \quad \textit{and} \quad Z^* = \{j : \ s^*_j > 0\}$$

are invariant for all pairs of strictly complementary solutions.

Given (LP) or (LD), the pair of P^* and Z^* is called the (strict) *complementarity partition.* $\{x: \ A_{P^*}x_{P^*} = b, \ x_{P^*} \geq 0, \ x_{Z^*} = 0\}$ is called the *primal optimal face*, and $\{y: \ c_{Z^*} - A_{Z^*}^T y \geq 0, \ c_{P^*} - A_{P^*}^T y = 0\}$ is called the *dual optimal face.*

Select m linearly independent columns, denoted by the index set B, from A. Then matrix A_B is nonsingular and we may uniquely solve

$$A_B x_B = b$$

for the m-vector x_B. By setting the variables, x_N, of x corresponding to the remaining columns of A equal to zero, we obtain a solution x such that

$$Ax = b.$$

Then, x is said to be a (*primal*) *basic solution* to (LP) with respect to the *basis* A_B. The components of x_B are called *basic variables.* A dual vector y satisfying

$$A_B^T y = c_B$$

is said to be the corresponding *dual basic solution.* If a basic solution $x \geq 0$, then x is called a *basic feasible solution.* If the dual solution is also feasible, that is

$$s = c - A^T y \geq 0,$$

then x is called an *optimal basic solution* and A_B an *optimal basis.* A basic feasible solution is a vertex on the boundary of the feasible region. An optimal basic solution is an optimal vertex of the feasible region.

If one or more components in x_B has value zero, that basic solution x is said to be (*primal*) degenerate. Note that in a nondegenerate basic solution the basic variables and the basis can be immediately identified from the nonzero components of the basic solution. If all components, s_N, in the corresponding dual slack vector s, except for s_B, are non-zero, then y is said to be (*dual*) nondegenerate. If both primal and dual basic solutions are nondegenerate, A_B is called a *nondegenerate basis.*

Theorem 1.15 *(LP fundamental theorem) Given (LP) and (LD) where A has full row rank m,*

i) *if there is a feasible solution, there is a basic feasible solution;*

ii) *if there is an optimal solution, there is an optimal basic solution.*

The above theorem reduces the task of solving a linear program to that searching over basic feasible solutions. By expanding upon this result, the simplex method, a finite search procedure, is derived. The simplex method is to proceed from one basic feasible solution (an extreme point of the feasible region) to an adjacent one, in such a way as to continuously decrease the value of the objective function until a minimizer is reached. In contrast, interior-point algorithms will move in the interior of the feasible region and reduce the value of the objective function, hoping to by-pass many extreme points on the boundary of the region.

1.3.6 Quadratic programming (QP)

Given $Q \in \mathcal{R}^{n\times n}$, $A \in \mathcal{R}^{m\times n}$, $b \in \mathcal{R}^m$ and $c \in \mathcal{R}^n$, the quadratic programming (QP) problem is the following optimization problem:

$$\begin{array}{lll} (QP) & \text{minimize} & q(x) := (1/2)x^T Qx + c^T x \\ & \text{subject to} & Ax = b, \ x \geq 0. \end{array}$$

We may denote the feasible set by $\mathcal{F}_p$. The data and solution sets for (QP) are

$$\mathcal{Z}_p = \{Q \in \mathcal{R}^{n\times n}, A \in \mathcal{R}^{m\times n}, b \in \mathcal{R}^m, c \in \mathcal{R}^n\}$$

and

$$\mathcal{S}_p = \{x \in \mathcal{F}_p : q(x) \leq q(y) \quad \text{for every} \quad y \in \mathcal{F}_p\}.$$

A feasible point x^* is called a *KKT* point, where KKT stands for Karush-Kuhn-Tucker, if the following KKT conditions hold: there exists $(y^* \in \mathcal{R}^m, s^* \in \mathcal{R}^n)$ such that (x^*, y^*, s^*) is feasible for the following dual problem:

$$\begin{array}{lll} (QD) & \text{maximize} & d(x,y) := b^T y - (1/2)x^T Qx \\ & \text{subject to} & A^T y + s - Qx = c, \ x, \ s \geq 0, \end{array}$$

and satisfies the complementarity condition

$$(x^*)^T s^* = (1/2)(x^*)^T Qx^* + c^T x^* - (b^T y^* - (1/2)(x^*)^T Qx^* = 0.$$

Similar to LP, we can write the KKT condition as:

$$(x, y, s) \in (\mathcal{R}^n_+, \mathcal{R}^m, \mathcal{R}^n_+)$$

and

$$\begin{array}{rcl} Xs & = & 0 \\ Ax & = & b \\ -A^T y + Qx - s & = & -c. \end{array} \tag{1.3}$$

Again, this system has total $2n + m$ unknowns and $2n + m$ equations including n nonlinear equations.

The above condition is also called the *first-order necessary condition*. If Q is positive semi-definite, then x^* is an optimal solution for (QP) if and only if x^* is a KKT point for (QP). In this case, the solution set for (QP) is characterized by a system of linear inequalities and equations. One can see (LP) is a special case of (QP).

1.3.7 Linear complementarity problem (LCP)

Given $M \in \mathcal{R}^{n\times n}$ and $q \in \mathcal{R}^n$, the linear complementarity problem (LCP) is to find a pair $x, s \in \mathcal{R}^n$ such that

$$s = Mx + q, \quad (x, s) \geq 0 \quad \text{and} \quad x^T s = 0.$$

A pair $(x, s) \geq 0$ satisfying $s = Mx + q$ is called a *feasible pair*. The LCP problem can be written as an optimization problem:

$$\begin{array}{lll} (LCP) & \text{minimize} & x^T s \\ & \text{subject to} & Mx - s = -q, \;\; x, s \geq 0. \end{array}$$

The LCP is a fundamental decision and optimization problem. It also arises from economic equilibrium problems, noncooperative games, traffic assignment problems, and optimization problems.

The data and solution sets for LCP are

$$\mathcal{Z}_p = \{M \in \mathcal{R}^{n\times n}, q \in \mathcal{R}^n\}$$

and

$$\mathcal{S}_p = \left\{ (x, s) \in (\mathcal{R}^n_+, \mathcal{R}^n_+) : \begin{array}{rcl} Xs & = & 0 \\ Mx - s & = & -q \end{array} \right\}. \tag{1.4}$$

(Again, $x^T s = 0$ plus nonnegativity on x and s imply n equations $Xs = 0$.)

There are extensions from LCP. One extension is some variables can be "free" (unrestricted in sign) and their counter-parts are zeros:

$$\begin{array}{ll} \text{minimize} & x^T s \\ \text{subject to} & M \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} s \\ 0 \end{pmatrix} = -q, \;\; x, s \geq 0. \end{array} \tag{1.5}$$

One can verify that the solution set of (LP) and the KKT set of (QP) can be formulated as this problem, where

$$M = \begin{pmatrix} Q & -A^T \\ A & 0 \end{pmatrix} \quad \text{and} \quad q = \begin{pmatrix} c \\ -b \end{pmatrix}. \tag{1.6}$$

Another extension is called the *generalized linear complementarity problem*:

$$
\begin{array}{lll}
(GLCP) & \text{minimize} & x^T s \\
 & \text{subject to} & Ax + Bs + Cz = q,\ (x \in \mathcal{R}^n, s \in \mathcal{R}^n, z \in \mathcal{R}^d) \geq 0,
\end{array}
$$

where matrices $A, B \in \mathcal{R}^{m \times n}$, $C \in \mathcal{R}^{m \times d}$ and $q \in \mathcal{R}^m$. We will discuss on solving GLCP in Chapter 9.

If M (may not be symmetric) is monotone or positive semi-definite, i.e.,

$$x^T M x \geq 0, \quad \text{for all} \quad x,$$

then the LCP is called a *monotone LCP*. The following theorem characterizes the solution set of a monotone linear complementarity problem.

Theorem 1.16 *(Monotone linear complementarity theorem) Consider the monotone linear complementarity problem. We have*

i) *if the problem is feasible, then it has a solution with* $x^T s = 0$*;*

ii) *the solution set is convex;*

iii) *it has a maximal complementary solution pair* x^* *and* s^**, meaning that the number of the positive components in vector* $x^* + s^*$ *is maximized. Moreover, the supports*

$$P^* = \{j :\ x_j^* > 0\} \quad \textit{and} \quad Z^* = \{j :\ s_j^* > 0\}$$

are invariant for all pairs of maximal complementary solutions.

Note that, in general, a monotone complementarity problem, unlike linear programming, may not have a strictly complementary solution.

1.3.8 Positive semi-definite programming (PSP)

Given $C \in \mathcal{M}^n$, $A_i \in \mathcal{M}^n$, $i = 1, 2, ..., m$, and $b \in \mathcal{R}^m$, the positive semi-definite programming problem is to find a matrix $X \in \mathcal{M}^n$ for the optimization problem:

$$
\begin{array}{lll}
(PSP) & \inf & C \bullet X \\
 & \text{subject to} & A_i \bullet X = b_i, i = 1, 2, ..., m,\ X \succeq 0.
\end{array}
$$

Recall that the $\bullet$ operation is the matrix inner product

$$A \bullet B := \operatorname{tr} A^T B.$$

The notation $X \succeq 0$ means that X is a positive semi-definite matrix, and $X \succ 0$ means that X is a positive definite matrix. If a point $X \succ 0$ and satisfies all equations in (PSP), it is called a (*primal*) strictly or interior feasible solution. .

The data set of (PSP) is

$$\mathcal{Z}_p = \{A_i \in \mathcal{M}^n,\ i = 1, \ldots, m,\ b \in \mathcal{R}^m, c \in \mathcal{M}^n\}.$$

The dual problem to (PSP) can be written as:

$$\begin{array}{lll} (PSD) & \sup & b^T y \\ & \text{subject to} & \sum_i^m y_i A_i + S = C,\ S \succeq 0, \end{array}$$

which is analogous to the dual (LD) of LP. Here $y \in \mathcal{R}^m$ and $S \in \mathcal{M}^n$. If a point $(y, S \succ 0)$ satisfies all equations in (PSD), it is called a *dual interior feasible solution*.

Example 1.12 *Let $P(y \in \mathcal{R}^m) = -C + \sum_i^m y_i A_i$, where C and A_i, $i = 1, \ldots, m$, are given symmetric matrices. The problem of minimizing the max-eigenvalue of $P(y)$ can be cast as a (PSD) problem.*

In positive semi-definite programming, we minimize a linear function of a matrix in the positive semi-definite matrix cone subject to affine constraints. In contrast to the positive orthant cone of linear programming, the positive semi-definite matrix cone is non-polyhedral (or "non-linear"), but convex. So positive semi-definite programs are convex optimization problems. Positive semi-definite programming unifies several standard problems, such as linear programming, quadratic programming, and convex quadratic minimization with convex quadratic constraints, and finds many applications in engineering, control, and combinatorial optimization.

We have several theorems analogous to Farkas' lemma.

Theorem 1.17 *(Farkas' lemma in PSP) Let $A_i \in \mathcal{M}^n$, $i = 1, ..., m$, have rank m (i.e., $\sum_i^m y_i A_i = 0$ implies $y = 0$) and $b \in \mathcal{R}^m$. Then, there exists a symmetric matrix $X \succ 0$ with*

$$A_i \bullet X = b_i, \quad i = 1, ..., m,$$

if and only if $\sum_i^m y_i A_i \preceq 0$ and $\sum_i^m y_i A_i \neq 0$ implies $b^T y < 0$.

Note the difference between the above theorem and Theorem 1.9.

Theorem 1.18 *(Weak duality theorem in PSP) Let $\mathcal{F}_p$ and $\mathcal{F}_d$, the feasible sets for the primal and dual, be non-empty. Then,*

$$C \bullet X \geq b^T y \quad \text{where} \quad X \in \mathcal{F}_p,\ (y, S) \in \mathcal{F}_d.$$

The weak duality theorem is identical to that of (LP) and (LD).

Corollary 1.19 *(Strong duality theorem in PSP) Let $\mathcal{F}_p$ and $\mathcal{F}_d$ be non-empty and have an interior. Then, X is optimal for (PS) if and only if the following conditions hold:*

i) $X \in \mathcal{F}_p$;

ii) *there is* $(y, S) \in \mathcal{F}_d$;

iii) $C \bullet X = b^T y$ *or* $X \bullet S = 0$.

Again note the difference between the above theorem and the strong duality theorem for LP.

Two positive semi-definite matrices are complementary to each other, $X \bullet S = 0$, if and only if $XS = 0$ (Exercise 1.24). From the optimality conditions, the solution set for certain (PSP) and (PSD) is

$$\mathcal{S}_p = \{X \in \mathcal{F}_p,\ (y, S) \in \mathcal{F}_d : C \bullet X - b^T y = 0\},$$

or

$$\mathcal{S}_p = \{X \in \mathcal{F}_p,\ (y, S) \in \mathcal{F}_d : XS = 0\},$$

which is a system of linear matrix inequalities and equations.

In general, we have

Theorem 1.20 *(PSP duality theorem) If one of (PSP) or (PSD) has a strictly or interior feasible solution and its optimal value is finite, then the other is feasible and has the same optimal value. If one of (PSP) or (PSD) is unbounded then the other has no feasible solution.*

Note that a duality gap may exists if neither (PSP) nor (PSD) has a strictly feasible point. This is in contrast to (LP) and (LD) where no duality gap exists if both are feasible.

Although positive semi-definite programs are much more general than linear programs, they are not much harder to solve. It has turned out that most interior-point methods for LP have been generalized to positive semi-definite programs. As in LP, these algorithms possess polynomial worst-case complexity under certain computation models. They also perform well in practice. We will describe such extensions later in this book.

1.3.9 Nonlinear programming (NP)

Given $f : \mathcal{R}^n \to \mathcal{R}$, $h : \mathcal{R}^n \to \mathcal{R}^m$, $g : \mathcal{R}^n \to \mathcal{R}^d$, the nonlinear programming problem is the following optimization problem:

$$\begin{array}{lll} (NP) & \text{minimize} & f(x) \\ & \text{subject to} & h(x) = b, \;\; g(x) \geq 0. \end{array}$$

We may denote the feasible set by $\mathcal{F}$. The data and solution sets for (NP) are

$$\mathcal{Z}_p = \{f, h, g, \nabla f, \nabla h, \nabla g, \ldots\}$$

and

$$\mathcal{S}_p = \{x \in \mathcal{F} : f(x) \leq f(y) \text{ for every } y \in \mathcal{F}\}.$$

A feasible point x^* is called a *KKT point* if the following KKT conditions hold: there exists $(y^* \in \mathcal{R}^m, s^* \in \mathcal{R}^d)$ such that (x^*, y^*, s^*) are satisfying

$$\nabla^T f(x^*) - \nabla^T h(x^*)y^* - \nabla^T g(x^*)s^* = 0, \;\; s^* \geq 0,$$

and

$$g(x^*)^T s^* = 0.$$

Here, y^* and s^* are called the *Lagrange* or *dual multipliers*. We have the following necessary condition for a point to be a local minimizer.

Theorem 1.21 *(Karush-Kuhn-Tucker theorem) Let x^* be a relative (local) minimum solution for (NP) and suppose x^* is a regular point for the constraints, meaning that the Jacobian matrix $\nabla h(x^*)$ and the gradient vectors $\nabla g_j(x^*)$ for all j with $g_j(x^*) = 0$ (they called* active inequality constraints*) are linearly independent. Then, x^* must be a KKT point.*

If f is convex, h is affine, and g is concave, then x^* is optimal if and only if x^* is a KKT point for (NP). Thus, the necessary condition becomes sufficient.

1.3.10 Nonlinear complementarity problem (NCP)

Let $f(x) : \mathcal{R}^n_+ \to \mathcal{R}^n$ be a continuous real function. Then, the nonlinear complementarity problem (NCP) is to find a pair $x, s \in \mathcal{R}^n$ such that

$$s = f(x), \quad (x, s) \geq 0 \quad \text{and} \quad Xs = 0.$$

We say (NCP) is feasible if there exists an $x \geq 0$ and $s = f(x) \geq 0$.

If $f(x)$ is a monotone function over $\mathcal{R}^n_+$, that is,

$$(x^1 - x^2)^T (f(x^1) - f(x^2)) \geq 0 \quad \text{for all} \quad x^1, x^2 \in \mathcal{R}^n_+,$$

then the problem is called the *monotone complementarity problem*. It can be written as an optimization problem:

$$(MCP) \quad \begin{array}{ll} \text{minimize} & x^T s \\ \text{subject to} & s = f(x), \ (x, s) \geq 0. \end{array}$$

The Jacobian matrix of f, ∇f, is positive semi-definite in $\mathcal{R}^n_+$.

Denote the feasible set by $\mathcal{F}$ and the solution set by $\mathcal{S}$. Note that (MCP) being feasible does not imply that (MCP) has a solution. The following theorem characterizes the solution set of a monotone complementarity problem.

Theorem 1.22 *(Monotone complementarity theorem) Consider the monotone nonlinear complementarity problem. We have*

i) *If (MCP) has an interior feasible point, i.e., $x > 0$ and $s = f(x) > 0$, then it has a solution with $x^T s = 0$;*

ii) *the solution set is convex;*

iii) *it has a maximal complementary solution pair x^* and s^*, meaning that the number of the positive components in vector $x^* + s^*$ is maximized. Moreover, the supports*

$$P^* = \{j : \ x^*_j > 0\} \quad \text{and} \quad Z^* = \{j : \ s^*_j > 0\}$$

are invariant for all pairs of maximal complementary solutions.

Note that, in general, a monotone complementarity problem, unlike linear programming, may not have a strictly complementary solution.

Similar to our discussion on linear complementarity, finding a KKT point or proving infeasibility or unboundedness of a convex nonlinear programming problem can be reduced to solving a monotone complementarity problem with possible "free" variables.

1.4 Algorithms and Computation Models

An algorithm is a list of instructions to solve a problem. For every instance of problem $\mathcal{P}$, i.e., for every given data $Z \in \mathcal{Z}_p$, an algorithm for solving $\mathcal{P}$ either determines that $\mathcal{S}_p$ is empty or generates an output x such that $x \in \mathcal{S}_p$ or x is close to $\mathcal{S}_p$ in certain measure. The latter x is called an *approximate* solution.

Let us use $\mathcal{A}_p$ to denote the collection of all possible algorithm for solving every instance in $\mathcal{P}$. Then, the (operation) complexity of an algorithm

$A \in \mathcal{A}_p$ for solving an instance $Z \in \mathcal{Z}_p$ is defined as the total arithmetic operations: $+$, $-$, $*$, $/$, and comparison on real numbers. Denote it by $c_o(A, Z)$. Sometimes it is convenient to define the iteration complexity, denoted by $c_i(A, Z)$, where we assume that each iteration costs a polynomial number (in m and n) of arithmetic operations. In most iterative algorithms, each iteration can be performed efficiently both sequentially and in parallel, such as solving a system of linear equations, rank-one updating the inversion of a matrix, pivoting operation of a matrix, multiplying a matrix by a vector, etc.

We present four algorithm efficiency or complexity measures frequently used in algorithm analysis and optimization theory. The first two are concerned by the worst-case performance, the third one is on the "average" performance, and the fourth one is about the "local" performance. They will all be used in our analysis of interior-point algorithms.

1.4.1 Worst-case complexity

The worst-case complexity of algorithm A for problem $\mathcal{P}$ is defined as

$$c(A) := \sup_{Z \in \mathcal{Z}_p} c(A, Z).$$

It is better to distinguish the worst-case complexity of an algorithm, A, from that of a problem $\mathcal{P}$. The worst-case complexity of the problem is

$$c^p := \inf_{A \in \mathcal{A}_p} c(A).$$

Analyzing the worst-case complexity of a problem is challenging since $\mathcal{A}_p$ is an unknown domain, and the analysis of the complexity of the algorithm is equally difficult since $\mathcal{P}$ is also immense. However, the complexity theory does not directly attack the algorithm complexity for every instance. Instead, it classifies $\mathcal{P}$ using its data bit-size L, where the data are assumed rational. This is the Turing Machine Model for computation. We may call this type of complexity *size-based*. Then, we express an upper bound $f_A(m, n, L)$, in terms of the parameters m, n, and L, for the size-based complexity of algorithm A as

$$c(A, L) := \sup_{Z \in \mathcal{Z}_p,\ \mathrm{size}(Z) \le L} c(A, Z) \le f_A(m, n, L).$$

(Recall that integer m and n represent the problem dimensions.) Then, the size-based complexity of problem $\mathcal{P}$ has a relation

$$c^p(L) := \inf_{A \in \mathcal{A}_p} c(A, L) \le f_A(m, n, L).$$

We see that the complexity of algorithms is an upper bound for the complexity of the problem. Another active pursuit in computer science is the analysis of a lower bound for the problem's complexity, which is outside of the scope of this monograph.

If $f_A(m,n,L)$ is a polynomial in m, n, and L, then we say algorithm A is a *polynomial-time* or *polynomial* algorithm and problem $\mathcal{P}$ is polynomially solvable. If $f_A(m,n,L)$ is independent of L and polynomial in m and n, then we say algorithm A is a *strongly polynomial algorithm*.

In the real number model, the use of L is not suitable. We may use ϵ, the error for an approximate solution as a parameter. Let $c(A,Z,\epsilon)$ be the total number of operations of algorithm A for generating an ϵ-approximate solution, with a well-defined measure, to problem $\mathcal{P}$. Then,

$$c(A,\epsilon) := \sup_{Z\in\mathcal{Z}_p} c(A,Z,\epsilon) \le f_A(m,n,\epsilon) \quad \text{for any} \quad \epsilon > 0.$$

We call this complexity model *error-based*. One may also view an approximate solution an exact solution to a problem ϵ-near to $\mathcal{P}$ with a well-defined measure in the data space. This is the so-called *backward analysis* model in numerical analysis.

If $f_A(m,n,\epsilon)$ is a polynomial in m, n, and $\log(1/\epsilon)$, then algorithm A is a polynomial algorithm and problem $\mathcal{P}$ is polynomially solvable. Again, if $f_A(m,n,\epsilon)$ is independent of ϵ and polynomial in m and n, then we say algorithm A is a strongly polynomial algorithm. If $f_A(m,n,\epsilon)$ is a polynomial in m, n, and $(1/\epsilon)$, then algorithm A is a polynomial approximation scheme or pseudo-polynomial algorithm . For some optimization problems, the complexity theory can be applied to prove not only that they cannot be solved in polynomial-time, but also that they do not have polynomial approximation schemes. In practice, approximation algorithms are widely used and accepted in practice.

Example 1.13 *There is a strongly polynomial algorithm for sorting a vector in descending or ascending order, for matrix-vector multiplication, and for computing the norm of a vector.*

Example 1.14 *Consider the bisection method to locate a root of a continuous function $f(x) : \mathcal{R} \to \mathcal{R}$ within interval $[0,1]$, where $f(0) > 0$ and $f(1) < 0$. The method calls the oracle to evaluate $f(1/2)$ (counted as one operation). If $f(1/2) > 0$, we throw away $[0,1/2)$; if $f(1/2) < 0$, we throw away $(1/2,1]$. Then we repeat this process on the remaining half interval. Each step of the method halves the interval that contains the root. Thus, in $\log(1/\epsilon)$ steps, we must have an approximate root whose distance to the root is less than ϵ. Therefore, the bisection method is a polynomial algorithm.*

We have to admit that the criterion of polynomiality is somewhat controversial. Many algorithms may not be polynomial but work fine in practice. This is because polynomiality is built upon the worst-case analysis. However, this criterion generally provides a qualitative statement: if a problem is polynomial solvable, then the problem is indeed relatively easy to solve regardless of the algorithm used. Furthermore, it is ideal to develop an algorithm with both polynomiality and practical efficiency.

1.4.2 Condition-based complexity

As we discussed before, in the Turing Machine Model the parameters are selected as the number of variables, the number of constraints, and the bit-size of the data of an instance. In fact, expressing the algorithm complexity in terms of the size of the problem does not really measure the difficulty of an instance of the problem. Two instances with the same size may result in drastically different performances by the same algorithm.

Example 1.15 *Consider the steepest descent algorithm for solving*

$$\text{minimize} \quad q(x) = (1/2)x^T Qx + c^T x, \tag{1.7}$$

where $Q \in \mathcal{R}^{n\times n}$ is positive definite. Denote by $x^* = -Q^{-1}c \in \mathcal{R}^n$ the minimizer of the problem. Starting from an $x^0 \in \mathcal{R}^n$, the method uses iterative formula

$$x^{k+1} = x^k - \alpha^k d^k, \quad \text{where} \quad d^k = Qx^k + c$$

and

$$\alpha^k = \frac{\|d^k\|^2}{\|d^k\|_Q^2} \ ,$$

which minimizes $q(x^k - \alpha d^k)$. (Recall that the norm $\|d\|_Q = \sqrt{d^T Qd}$ for positive definite Q.) It is well known that the algorithm generates a sequence of $\{x^k\}$ such that

$$\frac{\|x^{k+1} - x^*\|_Q}{\|x^k - x^*\|_Q} \leq \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1},$$

where $\lambda_1 \leq \lambda_2, ... \leq \lambda_n$ are n eigenvalues of Q with ascending order. Let

$$\xi(Q) := \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} < 1.$$

Then, to reach $\frac{\|x^{k+1}-x^*\|_Q}{\|x^k-x^*\|_Q} \leq \epsilon$ we need no more than $\log(1/\epsilon)/\log(1/\xi(Q))$ *iterations; the smaller $\xi(Q)$, the faster of the algorithm.*

In this example two Q matrices with the same size but different eigenvalue structure will possess quite different convergence speed. This phenomenon is surprisingly common in optimization, due to mathematical bases upon which algorithms are designed. Thus, the upper bound for the complexity of an algorithm may be expressed as $f_A(m,n,\eta(Z))$ or $f_A(m,n,\epsilon,\eta(Z))$ in both the rational-number and the real number models, where $\eta(Z)$ can be viewed as a condition number for the instance Z. ($\eta(Z) = \frac{1}{\log(1/\xi(Q))}$ in this example.) The better the condition number, the less difficult the instance. It is our goal to study this phenomenon and to improve the condition number and, thereby, the performance of an algorithm.

1.4.3 Average complexity

Let $\mathcal{Z}_p$ be a random sample space, then one can define the average or expected complexity of the algorithm for the problem as

$$c^a(A) = E_{Z\in\mathcal{Z}_p}(c(A,Z)).$$

If we know the condition-based complexity of an algorithm for $\mathcal{P}$, then the average complexity of the algorithm is

$$c^a(A) \le E_{Z\in\mathcal{Z}}(f_A(m,n,\eta(Z))).$$

In many cases, $f_A(m,n,\eta(Z))$ can be expressed as

$$f_A(m,n,\eta(Z)) = f^1(m,n)f^2(\eta(Z)).$$

Thus,

$$c^a(A) \le f^1(m,n)E_{Z\in\mathcal{Z}}(f^2(\eta(Z))),$$

which will simplify analysis a great deal.

Example 1.16 *Let us randomly generate positive definite matrix Q in Example 1.15. Then, $\xi(Q)$ is a random number, and the expected number of iterations to reach $\frac{\|x^{k+1}-x^*\|_Q}{\|x^k-x^*\|_Q} \le \epsilon$ is bounded by*

$$\log(1/\epsilon)E(\frac{1}{\log(1/\xi(Q))}).$$

Another probabilistic model is called *high-probability analysis.* We say that a problem $\mathcal{P}$ can be solved by algorithm A in $f_A(m,n)$ time with high probability if

$$\Pr\{c(A,Z) \le f_A(m,n)\} \to 1$$

as $m, n \to \infty$. Again, if we have a condition-based complexity and if we have

$$\Pr\{f^2(\eta(Z)) \leq f^3(m,n)\} \to 1$$

as $m, n \to \infty$, then the algorithm solves P in

$$f_A(m,n) = f^1(m,n) f^3(m,n)$$

operations with high probability.

1.4.4 Asymptotic complexity

Most algorithms are iterative in nature. They generate a sequence of ever-improving points $x^0, x^1, ..., x^k, ...$ approaching the solution set. For many optimization problems and/or algorithms, the sequence will never exactly reach the solution set. One theory of iterative algorithms, referred to as local or asymptotic convergence analysis, is concerned with the rate at which the optimality error of the generated sequence converges to zero.

Obviously, if each iteration of competing algorithms requires the same amount of work, the speed of the convergence of the error reflects the speed of the algorithm. This convergence rate, although it holds locally or asymptotically, provides evaluation and comparison of different algorithms. It has been widely used by the nonlinear optimization and numerical analysis community as an efficiency criterion. In many cases, this criterion does explain practical behavior of iterative algorithms.

Consider a sequence of real numbers $\{r^k\}$ converging to zero. One can define several notions related to the speed of convergence of such a sequence.

Definition 1.1 . *Let the sequence $\{r^k\}$ converge to zero. The order of convergence of $\{r^k\}$ is defined as the supermum of the nonnegative numbers p satisfying*

$$0 \leq \limsup_{k \to \infty} \frac{|r^{k+1}|}{|r^k|^p} < \infty.$$

Definition 1.2 . *Let the sequence $\{r^k\}$ converge to zero such that*

$$\limsup_{k \to \infty} \frac{|r^{k+1}|}{|r^k|^2} < \infty.$$

Then, the sequence is said to converge quadratically to zero.

It should be noted that the order of convergence is determined only by the properties of the sequence that holds as $k \to \infty$. In this sense we might say that the order of convergence is a measure of how good the tail of $\{r^k\}$ is. Large values of p imply the faster convergence of the tail.

Definition 1.3 . *Let the sequence $\{r^k\}$ converge to zero such that*

$$\limsup_{k\to\infty} \frac{|r^{k+1}|}{|r^k|} = \beta < 1.$$

Then, the sequence is said to converge linearly *to zero with convergence ratio β.*

Linear convergence is the most important type of convergence behavior. A linearly convergence sequence, with convergence ratio β, can be said to have a tail that converges to zero at least as fast as the geometric sequence $c\beta^k$ for a fixed number c. Thus, we also call linear convergence *geometric* convergence.

As a rule, when comparing the relative effectiveness of two competing algorithms both of which produce linearly convergent sequences, the comparison is based on their corresponding convergence ratio—the smaller the ratio, the faster the algorithm. The ultimate case where $\beta = 0$ is referred to as *superlinear* convergence.

Example 1.17 *The steepest descent algorithm for solving problem (1.7) in Example 1.15 has a linear convergence with convergence ratio $\xi(Q)$. If Q has equal eigenvalues, then the algorithm is superlinearly convergent.*

Example 1.18 *Consider the conjugate gradient algorithm for solving problem (1.7). Starting from an $x^0 \in \mathcal{R}^n$ and $d^0 = Qx^0 + c$, the method uses iterative formula*

$$x^{k+1} = x^k - \alpha^k d^k$$

where

$$\alpha^k = \frac{(d^k)^T(Qx^k + c)}{\|d^k\|_Q^2},$$

and

$$d^{k+1} = Qx^{k+1} - \theta^k d^k$$

where

$$\theta^k = \frac{(d^k)^T Q(Qx^{k+1} + c)}{\|d^k\|_Q^2} .$$

This algorithm is superlinearly convergent (in fact, it converges in finite number of steps).

1.5 Basic Computational Procedures

There are several basic numerical problems frequently solved by interior-point algorithms.

1.5.1 Gaussian elimination method

Probably the best-known algorithm for solving a system of linear equations is the Gaussian elimination method. Suppose we want to solve

$$Ax = b.$$

We may assume $a_{11} \neq 0$ after some row switching, where a_{ij} is the component of A in row i and column j. Then we can subtract appropriate multiples of the first equation from the other equations so as to have an equivalent system:

$$\begin{pmatrix} a_{11} & A_{1.} \\ 0 & A' \end{pmatrix} \begin{pmatrix} x_1 \\ x' \end{pmatrix} = \begin{pmatrix} b_1 \\ b' \end{pmatrix}.$$

This is a pivot step, where a_{11} is called a *pivot*, and A' is called a *Schur complement.* Now, recursively, we solve the system of the last $m-1$ equations for x'. Substituting the solution x' found into the first equation yields a value for x_1. The last process is called *back-substitution.*

In matrix form, the Gaussian elimination method transforms A into the form

$$\begin{pmatrix} U & C \\ 0 & 0 \end{pmatrix}$$

where U is a nonsingular, upper-triangular matrix,

$$A = L \begin{pmatrix} U & C \\ 0 & 0 \end{pmatrix},$$

and L is a nonsingular, lower-triangular matrix. This is called the *LU-decomposition.*

Sometimes, the matrix is transformed further to a form

$$\begin{pmatrix} D & C \\ 0 & 0 \end{pmatrix}$$

where D is a nonsingular, diagonal matrix. This whole procedure uses about nm^2 arithmetic operations. Thus, it is a strong polynomial-time algorithm.

1.5.2 Choleski decomposition method

Another useful method is to solve the least squares problem:

$$(LS) \quad \text{minimize} \quad \|A^T y - c\|.$$

The theory says that y^* minimizes $\|A^Ty - c\|$ if and only if

$$AA^Ty^* = Ac.$$

So the problem is reduced to solving a system of linear equations with a symmetric semi-positive definite matrix.

One method is Choleski's decomposition. In matrix form, the method transforms AA^T into the form

$$AA^T = L\Lambda L^T,$$

where L is a lower-triangular matrix and Λ is a diagonal matrix. (Such a transformation can be done in about nm^2 arithmetic operations as indicated in the preceding section.) L is called the *Choleski factor* of AA^T. Thus, the above linear system becomes

$$L\Lambda L^Ty^* = Ac,$$

and y^* can be obtained by solving two triangle systems of linear equations.

1.5.3 The Newton method

The Newton method is used to solve a system of nonlinear equations: given $f(x) : \mathcal{R}^n \rightarrow \mathcal{R}^n$, the problem is to solve n equations for n unknowns such that

$$f(x) = 0.$$

The idea behind Newton's method is to use the Taylor linear approximation at the current iterate x^k and let the approximation be zero:

$$f(x) \simeq f(x^k) + \nabla f(x^k)(x - x^k) = 0.$$

The Newton method is thus defined by the following iterative formula:

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

where scalar $\alpha \geq 0$ is called *step-size*. Rarely, however, is the Jacobian matrix ∇f inverted. Generally the system of linear equations

$$\nabla f(x^k)d_x = -f(x^k)$$

is solved and $x^{k+1} = x^k + \alpha d_x$ is used. The direction vector d_x is called a *Newton step*, which can be carried out in strongly polynomial time.

A modified or quasi Newton method is defined by

$$x^{k+1} = x^k - \alpha M^k f(x^k),$$

where M^k is an $n \times n$ symmetric matrix. In particular, if $M^k = I$, the method is called the *gradient method*, where f is viewed as the gradient vector of a real function.

The Newton method has a superior asymptotic convergence order equal 2 for $\|f(x^k)\|$. It is frequently used in interior-point algorithms, and believed to be the key to their effectiveness.

1.5.4 Solving ball-constrained linear problem

The ball-constrained linear problem has the following form:

$$\begin{array}{lll} (BP) & \text{minimize} & c^T x \\ & \text{subject to} & Ax = 0, \ \|x\|^2 \le 1, \end{array}$$

or

$$\begin{array}{lll} (BD) & \text{minimize} & b^T y \\ & \text{subject to} & \|A^T y\|^2 \le 1. \end{array}$$

x^* minimizes (BP) if and only if there always exists a y such that they satisfy

$$AA^T y = Ac,$$

and if $c - A^T y \neq 0$ then

$$x^* = -(c - A^T y)/\|c - A^T y\|;$$

otherwise any feasible x is a solution. The solution y^* for (BD) is given as follows: Solve

$$AA^T \bar{y} = b,$$

and if $\bar{y} \neq 0$ then set

$$y^* = -\bar{y}/\|A^T \bar{y}\|;$$

otherwise any feasible y is a solution. So these two problems can be reduced to solving a system of linear equations.

1.5.5 Solving ball-constrained quadratic problem

The ball-constrained quadratic problem has the following form:

$$\begin{array}{lll} (BP) & \text{minimize} & (1/2)x^T Q x + c^T x \\ & \text{subject to} & Ax = 0, \ \|x\|^2 \le 1, \end{array}$$

or simply

$$\begin{array}{lll} (BD) & \text{minimize} & (1/2)y^T Q y + b^T y \\ & \text{subject to} & \|y\|^2 \le 1. \end{array}$$

This problem is used by the classical *trust region* method for nonlinear optimization. The optimality conditions for the minimizer y^* of (BD) are

$$(Q + \mu^* I)y^* = -b, \quad \mu^* \geq 0, \quad \|y^*\|^2 \leq 1, \quad \mu^*(1 - \|y^*\|^2) = 0,$$

and

$$(Q + \mu^* I) \succeq 0.$$

These conditions are necessary and sufficient. This problem can be solved in polynomial time $\log(1/\epsilon)$ and $\log(\log(1/\epsilon))$ by the bisection method or a hybrid of the bisection and Newton methods, respectively. In practice, several trust region procedures have been very effective in solving this problem.

The ball-constrained quadratic problem will be used an a sub-problem by several interior-point algorithms in solving complex optimization problems. We will discuss them later in the book.

1.6 Notes

The term "complexity" was introduced by Hartmanis and Stearns [180]. Also see Garey and Johnson [131] and Papadimitriou and Steiglitz [337]. The NP theory was due to Cook [88] and Karp [219]. The importance of P was observed by Edmonds [106].

Linear programming and the simplex method were introduced by Dantzig [93]. Other inequality problems and convexity theories can be seen in Gritzmann and Klee [169], Grötschel, Lovász and Schrijver [170], Grünbaum [171], Rockafellar [364], and Schrijver [373]. Various complementarity problems can be found found in Cottle, Pang and Stone [91]. The positive semi-definite programming, an optimization problem in nonpolyhedral cones, and its applications can be seen in Nesterov and Nemirovskii [327], Alizadeh [9], and Boyd, Ghaoui, Feron and Balakrishnan [72]. Recently, Goemans and Williamson [141] obtained several breakthrough results on approximation algorithms using positive semi-definite programming. The KKT condition for nonlinear programming was given by Karush, Kuhn and Tucker [240].

It was shown by Klee and Minty [223] that the simplex method is not a polynomial-time algorithm. The ellipsoid method, the first polynomial-time algorithm for linear programming with rational data, was proven by Khachiyan [221]; also see Bland, Goldfarb and Todd [63]. The method was devised independently by Shor [380] and by Nemirovskii and Yudin [321]. The interior-point method, another polynomial-time algorithm for linear programming, was developed by Karmarkar. It is related to the classical barrier-function method studied by Frisch [126] and Fiacco and McCormick [116]; see Gill, Murray, Saunders, Tomlin and Wright [139], and Anstreicher [23]. For a brief LP history, see the excellent article by Wright [457].

The real computation model was developed by Blum, Shub and Smale [66] and Nemirovskii and Yudin [321]. The average setting can be seen in Traub, Wasilkowski and Wozniakowski [420]. The asymptotic convergence rate and ratio can be seen in Luenberger [248], Ortega and Rheinboldt [334], and Traub [419]. Other complexity issues in numerical optimization were discussed in Vavasis [450].

Many basic numerical procedures listed in this chapter can be found in Golub and Van Loan [154]. The ball-constrained quadratic problem and its solution methods can be seen in Moré [308], Sorenson [387], and Dennis and Schnable [96]. The complexity result of the ball-constrained quadratic problem was proved by Vavasis [450] and Ye [469, 473].

1.7 Exercises

1.1 *Let $Q \in \mathcal{R}^{n\times n}$ be a given nonsingular matrix, and a and b be given $\mathcal{R}^n$ vectors. Show*

$$(Q+ab^T)^{-1} = Q^{-1} - \frac{1}{1+b^TQ^{-1}a}Q^{-1}ab^TQ^{-1}.$$

This formula is called the Sherman-Morrison-Woodbury formula.

1.2 *Prove that the eigenvalues of all matrices $Q \in \mathcal{M}^{n\times n}$ are real. Furthermore, show that Q is PSD if and only if all its eigenvalues are nonnegative, and Q is PD if and only if all its eigenvalues are positive.*

1.3 *Using the ellipsoid representation in Section 1.2.2, find the matrix Q and vector y that describes the following ellipsoids:*

1. *The 3-dimensional sphere of radius 2 centered at the origin;*

2. *The 2-dimensional ellipsoid centered at $(1;2)$ that passes the points $(0;2)$, $(1;0)$, $(2;2)$, and $(1;4)$;*

3. *The 2-dimensional ellipsoid centered at $(1;2)$ with axes parallel to the line $y=x$ and $y=-x$, and passing through $(-1;0)$, $(3;4)$, $(0;3)$, and $(2;1)$.*

1.4 *Show that the biggest coordinate-aligned ellipsoid that is entirely contained in $\mathcal{R}^n_+$ and has its center at $x^a \in \overset{\circ}{\mathcal{R}}{}^n_+$ can be written as:*

$$E(x^a) = \{x \in \mathcal{R}^n : \ \|(X^a)^{-1}(x-x^a)\| \le 1\}.$$

1.5 *Show that the non-negative orthant, the positive semi-definite cone, and the second-order cone are all self-dual.*

1.6 *Consider the convex set* $C = \{x \in \mathcal{R}^2 : \ (x_1 - 1)^2 + (x_2 - 1)^2 \leq 1\}$ *and let* $y \in \mathcal{R}^2$*. Assuming* $y \notin C$*,*

1. *Find the point in* C *that is closest to* y*;*

2. *Find a separating hyperplane vector as a function of* y*.*

1.7 *Using the idea of Exercise 1.6, prove the separating hyperplane theorem 1.3.*

1.8 *Given an* $m \times n$ *matrix* A *and a vector* $c \in \mathcal{R}^n$*, consider the function* $\mathcal{B}(y) = \sum_{j=1}^n \log s_j$ *where* $s = c - A^T y > 0$*. Find* $\nabla \mathcal{B}(y)$ *and* $\nabla^2 \mathcal{B}(y)$ *in terms of* s*.*

1.9 *Prove that the level set of a quasi-convex function is convex.*

1.10 *Prove Propositions 1.5 and 1.6 for convex functions in Section 1.2.3.*

1.11 *Prove the Harmonic inequality described in Section 1.2.4.*

1.12 *Prove Farkas' lemma 1.8 for linear equations.*

1.13 *Prove the linear least-squares problem always has a solution.*

1.14 *Let* $P = A^T(AA^T)^{-1}A$ *or* $P = I - A^T(AA^T)^{-1}A$*. Then prove*

1. $P = P^2$*.*

2. P *is positive semi-definite.*

3. *The eigenvalues of* P *are either* 0 *or* 1*.*

1.15 *Using the separating theorem, prove Farkas' lemmas 1.9 and 1.10.*

1.16 *If a system* $A^T y \leq c$ *of linear inequalities in* m *variables has no solution, show that* $A^T y \leq c$ *has a subsystem* $(A')^T y \leq c'$ *of at most* $m + 1$ *inequalities having no solution.*

1.17 *Prove the LP fundamental theorem 1.15.*

1.18 *If (LP) and (LD) have a nondegenerate optimal basis* A_B*, prove that the strict complementarity partition in Theorem 1.14 is*

$$P^* = B.$$

1.19 *If Q is positive semi-definite, prove that x^* is an optimal solution for (QP) if and only if x^* is a KKT point for (QP).*

1.20 *Show that M of (1.6) is monotone if and only if Q is positive semi-definite.*

1.21 *Prove the monotone linear complementarity theorem 1.16.*

1.22 *Let $P(y) = -C + \sum_i^m y_i A_i$, where C and A_i, $i = 1, \ldots, m$, are given symmetric matrices. Formulate the minimization of the max-eigenvalue of $P(y)$ as a (PSD) problem. What does its primal problem look like?*

1.23 *Prove $X \bullet S \geq 0$ if both X and S are positive semi-definite matrices.*

1.24 *Prove that two positive semi-definite matrices are complementary to each other, $X \bullet S = 0$, if and only if $XS = 0$.*

1.25 *Prove Farkas' lemma 1.17 for positive semi-definite programming.*

1.26 *Let both (LP) and (LD) for a given data set (A, b, c) have interior feasible points. Then consider the level set*

$$\Omega(z) = \{y : \; c - A^T y \geq 0, -z + b^T y \geq 0\}$$

where $z < z^*$ and z^* designates the optimal objective value. Prove that $\Omega(z)$ is bounded and has an interior for any finite $z < z^*$, even $\mathcal{F}_d$ is unbounded.

1.27 *Given an (LP) data set (A, b, c) and an interior feasible point x^0, find the feasible direction d_x ($Ad_x = 0$) that achieves the steepest decrease in the objective function.*

1.28 *Given an (LP) data set (A, b, c) and a feasible point $(x^0, y^0, s^0) \in (\mathcal{R}^n_+, \mathcal{R}^m, \mathcal{R}^n_+)$ for the primal and dual, and ignoring the nonnegativity condition, write the systems of linear equations used to calculate the Newton steps for finding points that satisfy the optimality equations (1.2), (1.3), and (1.4), respectively.*

1.29 *Similar to our discussion on quadratic programming and linear complementarity, demonstrate that finding a KKT point of a convex nonlinear programming problem can be reduced to solving a monotone complementarity problem with possible "free" variables.*

1.30 *Show that the ball-constrained linear problem (BP) can be written in the (BD) form and write the KKT conditions for both of them.*

1.31 *Given a scalar $\alpha > 0$, a positive diagonal matrix S, and an $m \times n$ matrix A, find the formula for $y \in \mathcal{R}^m$ such that*

$$\begin{array}{ll} \textit{minimize} & e^T S^{-1} A^T y \\ \textit{subject to} & \|S^{-1} A^T y\|^2 \leq \alpha^2. \end{array}$$

1.32 *Show the optimality conditions for the minimizer y^* of (BD) in Section 1.5.5:*

$$(Q + \mu^* I) y^* = -b, \quad \mu^* \geq 0, \quad \|y^*\| \leq 1, \quad \mu^*(1 - \|y^*\|) = 0,$$

and

$$(Q + \mu^* I) \succeq 0,$$

are necessary and sufficient.