Conic Linear Programming

Yinyu Ye

December 2004, revised October 2017

Preface

This monograph is developed for MS&E 314, "Conic Linear Programming", which I am teaching at Stanford. Information, lecture slides, supporting materials, and computer programs related to this book may be found at the following address on the World-Wide Web:

http://www.stanford.edu/class/msande314

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

yinyu-ye@stanford.edu

A little story in the development of semidefinite programming (SDP), a major subclass of conic linear programming. One day in 1990, I visited the Computer Science Department of the University of Minnesota and met a young graduate student, Farid Alizadeh. He, working then on combinatorial optimization, introduced me "semidefinite optimization" or linear programming over the positive definite matrix cone. We had a very extensive discussion that afternoon and concluded that interior-point linear programming algorithms could be applicable to solving SDPs. I suggested Farid to look at the linear programming (LP) interior-point algorithms and to develop an SDP (primal) potential reduction algorithm. He worked hard for several months, and one afternoon showed up in my office in Iowa City, about 300 miles from Minneapolis. He had everything worked out, including potential function, algorithm, complexity bound, and even a "dictionary" list between LP and SDP. But he was stuck on one problem that was on how to keep the symmetry of the scaled directional matrix. We went to a bar nearby on Clinton Street in Iowa City (I paid for him since I was a third-year professor then and eager to demonstrate that I could take care of my students). After chatting for a while, I suggested that he should use scaling $X^{-1/2}\Delta X^{-1/2}$ to compute symmetric directional matrix Δ, instead of $X^{-1}\Delta$ which he was using earlier, where X is the current symmetric positive definite matrix. This way, $X + \alpha\Delta$ would remain symmetric with a step-size scalar. He returned to Minneapolis and moved to Berkeley shortly after, and few weeks later sent me an e-mail message telling me that everything had worked out beautifully.

At the same time, Nesterov and Nemirovskii developed a more general and powerful theory in extending interior-point algorithms for solving convex programs, where SDP was a special case. Boyd and his group presented a wide range of SDP applications and formulations, many of which were incredibly novel and elegant. Then came the primal-dual algorithms of many authors, the

SDP approximation algorithm for Max-Cut, ... – SDP eventually established its full popularity.

To Daisun, Fei, Tim, Kaylee and Rylee

Contents

List of Figures

Chapter 1

Introduction and Preliminaries

1.1 Introduction

Conic Linear Programming, hereafter CLP, is a natural extension of classical Linear programming (LP) that is a central decision model in Management Science and Operations Research. LP plays an extremely important role in the theory and application of Optimization. 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 adjust investments, and to maximize productivity.

In LP, the variables form a vector which is required to be component-wise nonnegative ($\geq \mathbf{0}$), where in CLP they are components of a vector, matrix or tensor and constrained to be in a (pointed) convex cone. Both of them have linear objective function and linear equality constraints as well.

Example 1.1 *Consider the following two optimization problems with three variables:*

- *A classical LP problem in standard form:*

$$\begin{array}{ll} \text{minimize} & 2x_1 + x_2 + x_3 \\ \text{subject to} & x_1 + x_2 + x_3 = 1, \\ & (x_1; x_2; x_3) \geq \mathbf{0}. \end{array}$$

- *An Second-Order Cone LP problem (SOCP) in standard form:*

$$\begin{array}{ll} \text{minimize} & 2x_1 + x_2 + x_3 \\ \text{subject to} & x_1 + x_2 + x_3 = 1, \\ & x_1 - \sqrt{x_2^2 + x_3^2} \geq 0, \end{array}$$

where the bottom constraint makes variables lie in the so-called "ice-cream cone" – formally called second-order cone.

- *An Semidefinite Cone LP problem (SDP) in standard form:*

$$\begin{array}{ll} \text{minimize} & 2x_1 + x_2 + x_3 \\ \text{subject to} & x_1 + x_2 + x_3 = 1, \\ & \begin{pmatrix} x_1 & x_2 \\ x_2 & x_3 \end{pmatrix} \succeq \mathbf{0}, \end{array}$$

where symbol $\cdot \succeq \mathbf{0}$ *implies that the left-side symmetric matrix must be positive semidefinite. In this instance, the matrix dimension is two.*

One can see that, although the objective and constraint are identical, the last constraint of the problems represents a different restriction, so that they are really different optimization problems and models. For example, the simplex method for LP is hardly applicable to CLP.

However, one thing in common is that interior-point algorithms developed in past three decades for LP are naturally applied to solving SDP or CLP at large. 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 goal of the monograph is to provide a text book for teaching Semidefinite Programming, a modern Linear Programming decision model and its applications in other scientific and engineering fields. One theme of the monograph is the "mapping" between CLP and LP, so that the reader, with knowledge of LP, can understand CLP with little effort.

The monograph is organized as follows. In Chapter 1, we discuss some necessary mathematical preliminaries. We also present several decision and optimization problems and several basic numerical procedures used throughout the text.

Chapter 2 is devoted to studying the theories and geometries of linear and matrix inequalities, convexity, and semidefinite programming. Almost all interior-point methods exploit rich geometric properties of linear and matrix inequalities, such as "center," "volume," "potential," etc. These geometries are also helpful for teaching, learning, and research.

Chapter 3 focuses on interior-point algorithms. Here, we select two types algorithms: the path-following algorithm and the potential reduction algorithm. Each algorithm has three forms, the primal, the dual and the primal-dual form. We analyze the worst-case complexity bound for them, where 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.

Not only has the convergence speed of CLP algorithms been significantly improved during the last decade, but also the problem domain applicable by CLP,

especially SDP, has dramatically widened. Chapters 4, 5, 6 and 7 would describe some of SDP applications and new established results in Quadratic and Combinatory Optimization, Euclidean Geometry Computation, Robust Optimization, Quantum Computation, etc.

Finally, we discuss major computational issues in Chapter 8. We discuss several effective implementation techniques frequently used in interior-point SDP software, such as the sparse linear system, the predictor and corrector step, and the homogeneous and self-dual formulation. We also present major difficulties and challenges faced by SDP.

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. We write vectors in bold lower case through out this monograph. Upper-case letters will be used to represent matrices. Greek letters will typically be used to represent scalars.

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

A vector is always considered as a column vector, unless otherwise stated. For convenience, we sometime write a column vector $\mathbf{x}$ as

$$\mathbf{x} = (x_1; x_2; \ldots; x_n)$$

and a row vector as

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

A set of vectors $\mathbf{a}_1, ..., \mathbf{a}_m$ is said to be linearly dependent if there are scalars $\lambda_1, ..., \lambda_m$, not all zero, such that the linear combination

$$\sum_{i=1}^{m} \lambda_i \mathbf{a}_i = \mathbf{0}.$$

The vector inequality $\mathbf{x} \geq \mathbf{y}$ means $x_j \geq y_j$ for $j = 1, 2, ..., n$. $\mathbf{0}$ represents a vector whose entries are all zeros and $\mathbf{e}$ represents a vector whose entries are all ones, where their dimensions may vary according to other vectors in expressions.

Addition of vectors and multiplication of a vector with a scalar are standard. The superscript "T" denotes transpose operation. The inner product in $\mathcal{R}^n$ is defined as follows:

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

The l_2 norm of a vector $\mathbf{x}$ is given by

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

and the l_∞ norm is

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

In general, the p-norm is

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

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

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

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

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 $\mathbf{a}_{i\cdot}$ and the jth column of A is denoted by $\mathbf{a}_{\cdot j}$; 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. For a vector $\mathbf{x} \in \mathcal{R}^n$, $\Delta(\mathbf{x})$ represents a diagonal matrix in $\mathcal{R}^{n\times n}$ whose diagonal entries are the entries of $\mathbf{x}$ and every other entry is 0, i.e.,

$$\Delta(\mathbf{x}) = \mathrm{Diag}(\mathbf{x}).$$

Addition of matrices and multiplication of a matrix with a scalar are standard. The inner product in $\mathcal{R}^{m\times n}$ is defined as follows:

$$\langle A, B\rangle := A \bullet B = \mathrm{tr}A^T B = \sum_{i,j} a_{i,j}b_{i,j} \quad \text{for} \quad A, B \in \mathcal{R}^{m\times n}.$$

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

$$\|A\|_f = \sqrt{\mathrm{tr}A^T A}\ .$$

On the other hand, the operator norm of A, denoted by $\|A\|$, is

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

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

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

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

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

If $Q \succ \mathbf{0}$, then $-Q$ is called negative definite (ND), denoted by $Q \prec \mathbf{0}$; if $Q \succeq \mathbf{0}$, then $-Q$ is called negative semi-definite (NSD), denoted by $Q \preceq \mathbf{0}$. 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 symmetric PD matrix Q we can define a Q-norm , $\|.\|_Q$, for vector $\mathbf{x}$ as

$$\|\mathbf{x}\|_Q = \sqrt{\mathbf{x}^T Q\mathbf{x}}\ ;$$

for matrix X as

$$\|X\|_Q = \sqrt{X \bullet QX}\ .$$

$\mathcal{S}^n$ denotes the space of symmetric matrices in $\mathcal{R}^{n\times n}$. $\mathcal{S}^n_+$ denote the set of positive semi-definite matrices in $\mathcal{S}^n$. $\mathcal{S}^n_{++}$ denotes the set of positive definite matrices in $\mathcal{S}^n$. If $A \in \mathcal{S}^n_+$ and $B \in \mathcal{S}^n_+$, then we must have (see Exercise (1.4))

$$\langle A, B\rangle := A \bullet B \geq 0.$$

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

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

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

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 $\mathbf{x}$ does:

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

1.2.2 Convex sets and cones

If $\mathbf{x}$ is a member of the set Ω, we write $\mathbf{x} \in \Omega$; if $\mathbf{y}$ is not a member of Ω, we write $\mathbf{y} \notin \Omega$. 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 $\Omega = \{\mathbf{x} : \ P(\mathbf{x})\}$ as the set of all elements satisfying property P.

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

A set Ω is closed if $\mathbf{x}^k \to \mathbf{x}$, where $\mathbf{x}^k \in \Omega$, implies $\mathbf{x} \in \Omega$. A set Ω is open if around every point $\mathbf{y} \in \Omega$ there is a ball that is contained in Ω, i.e., there is an $\epsilon > 0$ such that $B(\mathbf{y}, \epsilon) \subset \Omega$. 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 affine if for any two points $\mathbf{x}, \mathbf{y} \in C$ and any real numbers α and β, the affine combination point $\alpha\mathbf{x} + \beta\mathbf{y} \in C$. A set C is said to be convex if for any $\mathbf{x}, \mathbf{y} \in C$ and any real number α, $0 < \alpha < 1$, the convex combination point $\alpha\mathbf{x} + (1 - \alpha)\mathbf{y} \in C$. The convex hull of a set Ω is the intersection of all convex sets containing Ω.

Proposition 1.1 *Let C_1 and C_2 be convex sets in a same space. Then,*

- *$C_1 \cap C_2$ is convex.*
- *$C_1 + C_2$ is convex, where $C_1 + C_2 = \{\mathbf{b}_1 + \mathbf{b}_2 : \ \mathbf{b}_1 \in C_1$ and $\mathbf{b}_2 \in C_2\}$.*
- *$C_1 \oplus C_2$ is convex, where $C_1 \oplus C_2 = \{(\mathbf{b}_1; \mathbf{b}_2) : \ \mathbf{b}_1 \in C_1$ and $\mathbf{b}_2 \in C_2\}$.*

Let us use the notation $\mathcal{E}$ to represent either $\mathcal{R}^n$ or $\mathcal{S}^n$, 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. A set $K \subset \mathcal{E}$ is a cone if $\mathbf{x} \in K$ implies $\alpha\mathbf{x} \in K$ for all $\alpha > 0$. A cone that is also convex is a convex cone. For a cone $K \subset \mathcal{E}$, the dual of K is the cone

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

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

Example 1.2 *The n-dimensional non-negative orthant, $\mathcal{R}^n_+ = \{\mathbf{x} \in \mathcal{R}^n : \ \mathbf{x} \geq \mathbf{0}\}$, is a convex cone. The dual of the cone is also $\mathcal{R}^n_+$ so that it's self-dual.*

Example 1.3 *The set of all positive semi-definite matrices in $\mathcal{S}^n$, $\mathcal{S}^n_+$, is a convex cone, called the* positive semi-definite matrix cone*. The dual of the cone is also $\mathcal{S}^n_+$ so that it is self-dual.*

Example 1.4 *The set $\{\mathbf{x} \in \mathcal{R}^n : \ x_1 \geq \|\mathbf{x}_{-1}\|\}$, where $\mathbf{x}_{-1} := (x_2; ...; x_n) \in \mathcal{R}^{n-1}$, is a convex cone in $\mathcal{R}^n$. It is called the* second-order cone*, denoted by $\mathcal{N}^n_2$. The dual of the cone is also the second-order cone in $\mathcal{R}^n$ so that it is self-dual.*

Example 1.5 *The set* $\{\mathbf{x} \in \mathcal{R}^n : \ x_1 \geq \|\mathbf{x}_{-1}\|_p\}$, $1 \leq p \leq \infty$, *is a convex cone in* $\mathcal{R}^n$, *called the* p-order cone, *denoted by* $\mathcal{N}_p^n$. *The dual of the cone is the* q*order cone in* $\mathcal{R}^n$ *where* $\frac{1}{p} + \frac{1}{q} = 1$.

Definition 1.1 *We call* $\mathbf{x}$ *an interior point of cone* K *if and only if, for any point* $\mathbf{y} \in K^*$, $\mathbf{y} \bullet \mathbf{x} = 0$ *implies* $\mathbf{y} = \mathbf{0}$.

The set of interior points of K is denoted by $\overset{\circ}{K}$.

Proposition 1.2 *The interior of the followings convex cones are given as follows.*

- *The interior of the non-negative orthant cone is the set of all vectors whose every entry is positive.*
- *The interior of the positive semidefinite cone is the set of all positive definite matrices.*
- *The interior of* p*-order cone is the set of* $\{\mathbf{x} \in \mathcal{R}^n : \ x_1 > \|\mathbf{x}_{-1}\|_p\}$.

We leave the proof of the following proposition as an exercise.

Proposition 1.3 *Let* $\mathbf{X} \in \overset{\circ}{K}$ *and* $\mathbf{Y} \in K^*$. *Then For any nonnegative constant* κ, $Y \bullet X \leq \kappa$ *implies that* $\mathbf{Y}$ *is bounded.*

One of the most important type of convex sets is a hyperplane. Hyperplanes dominate the entire theory of optimization. Let $\mathbf{a}$ be a nonzero n-dimensional (directional) vector, and let b be a real number. The set

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

is a hyperplane in $\mathcal{R}^n$ (Figure 1.1). Relating to hyperplane, positive and negative closed half spaces are given by

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

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

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

$$P = \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}\}.$$

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. In particular, a cone C is (convex) polyhedral if C can be represented by

$$C = \{\mathbf{x} : \ A\mathbf{x} \leq \mathbf{0}\}$$

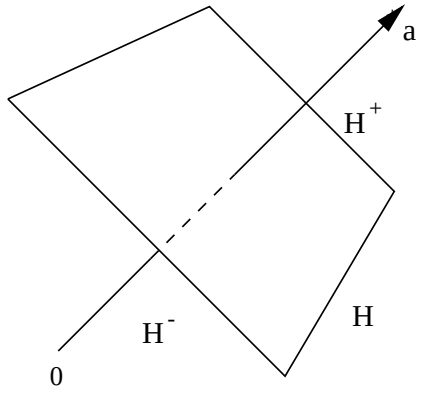


Figure 1.1: A hyperplane and half-spaces.

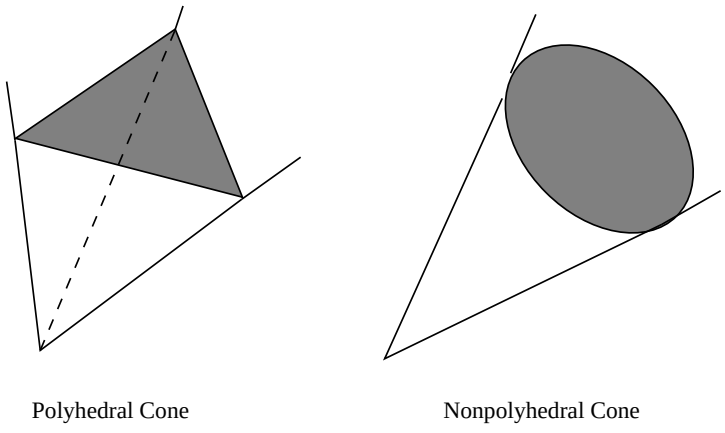


Figure 1.2: Polyhedral and nonpolyhedral cones.

or

$$C = \{A\mathbf{x} : \ \mathbf{x} \leq \mathbf{0}\}$$

for some matrix A (Figure 1.2). In the latter case, we say cone C is generated by the column vectors of A, written as cone(A).

Example 1.6 *The non-negative orthant is a polyhedral cone, and neither the positive semi-definite matrix cone nor the second-order cone is polyhedral.*

We have the following theorem on polyhedral cones:

Theorem 1.4 *(Carathéodory's theorem) Let point* $\mathbf{b} \in$ *cone*(A). *Then,* $\mathbf{b} \in$ *cone*$(\mathbf{a}_{i_1}, \mathbf{a}_{i_2}..., \mathbf{a}_{i_d})$ *for some linearly independent column vectors* $\mathbf{a}_{i_1}, \mathbf{a}_{i_2}..., \mathbf{a}_{i_d}$ *chosen from A.*

Proof.The proof is constructive and based on a null space reduction. Suppose

$$\mathbf{b} = \sum_{j=1}^{d} \bar{x}_j \mathbf{a}_{i_j}, \quad \bar{x_j} > 0, \ j = 1, ..., d$$

and generation vectors $\mathbf{a}_{i_1}, \mathbf{a}_{i_2}..., \mathbf{a}_{i_d}$ are not independent. Then, one can find a null space vector $\mathbf{0} \neq \bar{\mathbf{y}} \in \mathcal{R}^d$ such that

$$\mathbf{0} = \sum_{j=1}^{d} \bar{y}_j \mathbf{a}_{i_j}.$$

Let $\bar{\mathbf{y}}$ have at least one component positive (otherwise use $-\bar{\mathbf{y}}$), and consider the affine combination vector $\bar{\mathbf{x}} - \alpha\bar{\mathbf{y}}$. Then, there must be an $\bar{\alpha} > 0$ such that $\bar{\mathbf{x}} - \alpha\bar{\mathbf{y}} \geq \mathbf{0}$ and have at least one component equal 0; with out loss of generality, say the first one. Consequently,

$$\mathbf{b} = \sum_{j=1}^{d} (\bar{x}_j - \bar{\alpha}\bar{y}_j)\mathbf{a}_{i_j} = \sum_{j=2}^{d} (\bar{x}_j - \bar{\alpha}\bar{y}_j)\mathbf{a}_{i_j},$$

that is, $\mathbf{b} \in$ cone$(\mathbf{a}_{i_2}..., \mathbf{a}_{i_d})$. One can continue this reduction procedure as long as the remaining generation vectors are not independent, which gives the proof. ∎

The most important theorem about the convex set is the following separating theorem (Figure 1.3).

Theorem 1.5 *(Separating hyperplane theorem) Let* $C \subset \mathcal{E}$ *be a closed convex set and let* $\mathbf{b}$ *be a point exterior to C. Then there is a* $\mathbf{y} \in \mathcal{E}$ *such that*

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

The geometric interpretation of the theorem is that, given a convex set C and a point $\mathbf{b}$ outside of C, there is a hyperplane with norm-direction $\mathbf{y}$ which contains $\mathbf{b}$ in one of its open half spaces and C in the other.

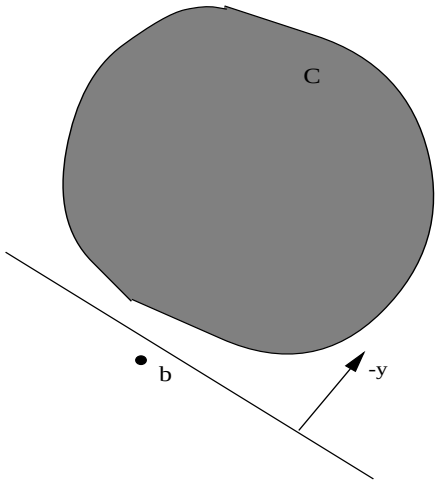


Figure 1.3: Illustration of the separating hyperplane theorem; an exterior point $\mathbf{b}$ is separated by a hyperplane from a convex set C.

Example 1.7 *Let C be a unit circle centered at the point $(1;1)$. That is, $C = \{\mathbf{x} \in \mathcal{R}^2 : (x_1 - 1)^2 + (x_2 - 1)^2 \leq 1\}$. If $\mathbf{b} = (2;0)$, $\mathbf{y} = (-1;1)$ is a separating hyperplane direction vector. If $\mathbf{b} = (0;-1)$, $\mathbf{y} = (0;1)$ is a separating hyperplane direction vector. It is worth noting that these separating hyperplanes are not unique.*

1.2.3 Real functions

The real function $f(\mathbf{x})$ is said to be continuous at $\mathbf{x}$ if $\mathbf{x}^k \to \mathbf{x}$ implies $f(\mathbf{x}^k) \to f(\mathbf{x})$. The real function $f(\mathbf{x})$ is said to be continuous on set $\Omega \subset \mathcal{E}$, where recall that $\mathcal{E}$ is either $\mathcal{R}^n$ or $\mathcal{S}^n$, if $f(\mathbf{x})$ is continuous at $\mathbf{x}$ for every $\mathbf{x} \in \Omega$.

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

Example 1.8 *Let $\mathbf{c} \in \mathcal{R}^n$ be given and $\mathbf{x} \in \mathcal{R}^n_{++}$. Then $\mathbf{c}^T\mathbf{x}$ is homogeneous of degree 1 and*

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

is homogeneous of degree 0, where $\ln$ is the natural logarithmic function. Let $C \in \mathcal{S}^n$ be given and $X \in \mathcal{S}^n_{++}$. Then $\mathbf{x}^T C\mathbf{x}$ is homogeneous of degree 2, $C \bullet X$ and $\det(X)$ are homogeneous of degree 1 and n, respectively, 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 $\mathcal{E}$ can be written as a single vector function $\mathbf{f} = (f_1, f_2, ..., f_m)^T \in \mathcal{R}^m$. If f_i has continuous partial derivatives of order p, we say $f_i \in C^p$. The gradient vector of a real-valued function $f_i \in C^1$ is a row vector in $\mathcal{R}^n$:

$$\nabla f_i(\mathbf{x}) = (\partial f/\partial x_1, \ ... \ , \partial f/\partial x_1).$$

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

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

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

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

Example 1.9 *Let $X \in \mathcal{S}^n_{++}$ and $f(X) = \ln \det(X)$. Then*

$$\nabla f(X) = X^{-1}$$

or

$$\frac{\partial f}{\partial x_{ij}} = (X^{-1})_{ij}, \ \forall i, j;$$

and

$$\nabla^2 f(X) = -X^{-1} \oplus X^{-1},$$

where $\oplus$ denotes the standard Kronecker product, or

$$\frac{\partial^2 f}{\partial x_{ij} \partial x_{kl}} = (X^{-1})_{ij} \cdot (X^{-1})_{kl}, \ \forall i, j, k, l.$$

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

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

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

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

Thus, a convex function is a quasi-convex function.

The epigraph set of f is given by

$$\{(t; \mathbf{x}) : \quad f(\mathbf{x}) \leq t\}.$$

f is a quasi-convex function implies that its epigraph set is convex. The z-level set of f is given by

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

f is a quasi-convex function implies that the level set of f is convex for any given z (see Exercise 1.14).

Several results that are used frequently in analyses 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 1.6 *(Taylor expansion) Let $f \in C^1$ be in a region containing the line segment $[\mathbf{x},\ \mathbf{y}]$. Then there is a α, $0 \leq \alpha \leq 1$, such that*

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

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

$$f(\mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x})(\mathbf{y} - \mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathbf{x})^T \nabla^2 f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y})(\mathbf{y} - \mathbf{x}).$$

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

Proposition 1.7 *Let $f \in C^1$. Then f is convex over a convex set Ω if and only if*

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

for all $\mathbf{x},\ \mathbf{y} \in \Omega$.

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

Proposition 1.8 *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 $\mathbf{x}, \mathbf{y} \in \mathcal{R}^n$, then

$$|\mathbf{x}^T\mathbf{y}| \leq \|x\|_p \|y\|_q, \text{ where } \frac{1}{p} + \frac{1}{p} = 1,\ p \geq 1.$$

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

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

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

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

Hadamard: given $A \in \mathcal{R}^{m\times n}$ with columns $\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n$, then

$$\sqrt{\det(A^T A)} \leq \prod \|\mathbf{a}_j\|\ .$$

1.3 Some Basic 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.

Theorem 1.9 *Weierstrass theorem 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 $\mathbf{x}^* \in \Omega$ such that for all $\mathbf{x} \in \Omega$, $f(\mathbf{x}) \geq f(\mathbf{x}^*)$.*

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

$$A\mathbf{x} = \mathbf{b}.$$

The data and solution sets are

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

$\mathcal{S}_p$ in this case is an affine set. Given an $\mathbf{x}$, one can easily check to see if $\mathbf{x}$ 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 1.10 *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 1.11 *Let $A \in \mathcal{R}^{m \times n}$ and $\mathbf{b} \in \mathcal{R}^m$. The system $A\mathbf{x} = \mathbf{b}$ has a solution if and only if there is no $\mathbf{y}$ such that $A^T\mathbf{y} = \mathbf{0}$ and $\mathbf{b}^T\mathbf{y} = 1$.*

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

Example 1.10 *Let $A = (1; -1)$ and $\mathbf{b} = (1; 1)$. Then, $\mathbf{y} = (1/2; 1/2)$ is an* infeasibility certificate *for $A\mathbf{x} = \mathbf{b}$.*

1.3.2 Linear least-squares problem

Given $A \in \mathcal{R}^{m\times n}$ and $\mathbf{c} \in \mathcal{R}^n$, the system of equations $A^T\mathbf{y} = \mathbf{c}$ 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 a $\mathbf{y} \in \mathcal{R}^m$ or $\mathbf{s} \in \mathcal{R}(A^T)$ such that $\|A^T\mathbf{y} - \mathbf{c}\|$ or $\|\mathbf{s} - \mathbf{c}\|$ is minimized. We can write the problem in the following format:

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

or

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

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

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

and

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

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

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

which becomes a system of linear equations and always has a solution. The vector $\mathbf{s} = A^T\mathbf{y} = A^T(AA^T)^+A\mathbf{c}$ is the projection of $\mathbf{c}$ 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 $(AA^T)^+AA^T\mathbf{x} = \mathbf{x}$ only for $\mathbf{x} \in \mathcal{R}(A^T)$.

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

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

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 1.19).

1.3.3 System of linear inequalities

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

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

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 1.12 *(Farkas' lemma) Let $A \in \mathcal{R}^{m\times n}$ and $\mathbf{b} \in \mathcal{R}^m$. Then, the system $\{\mathbf{x} :\ A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$ has a feasible solution $\mathbf{x}$ if and only if there is no $\mathbf{y}$ such that $A^T\mathbf{y} \leq \mathbf{0}$ and $\mathbf{b}^T\mathbf{y} = 1$.*

A vector $\mathbf{y}$, with $A^T\mathbf{y} \leq \mathbf{0}$ and $\mathbf{b}^T\mathbf{y} = 1$, is called a (*primal*) infeasibility certificate for the system $\{\mathbf{x} :\ A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$. Geometrically, Farkas' lemma means that if a vector $\mathbf{b} \in \mathcal{R}^m$ does not belong to the cone generated by column vectors $\mathbf{a}_1, ..., \mathbf{a}_n$, then there is a hyperplane separating $\mathbf{b}$ from cone$(\mathbf{a}_1, ..., \mathbf{a}_n)$.

Example 1.11 *Let $A = (1, 1)$ and $b = -1$. Then, $y = -1$ is an infeasibility certificate for $\{\mathbf{x} :\ A\mathbf{x} = b,\ \mathbf{x} \geq \mathbf{0}\}$.*

Theorem 1.13 *(Farkas' lemma variant) Let $A \in \mathcal{R}^{m\times n}$ and $\mathbf{c} \in \mathcal{R}^n$. Then, the system $\{\mathbf{y} :\ A^T\mathbf{y} \leq \mathbf{c}\}$ has a solution $\mathbf{y}$ if and only if there is no $\mathbf{x}$ such that $A\mathbf{x} = \mathbf{0}$, $\mathbf{x} \geq \mathbf{0}$ and $\mathbf{c}^T\mathbf{x} = -1$.*

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

Example 1.12 *Let $A = (1; -1)$ and $\mathbf{c} = (1; -2)$. Then, $\mathbf{x} = (1; 1)$ is an infeasibility certificate for $\{y :\ A^Ty \leq \mathbf{c}\}$.*

We say $\{\mathbf{x} :\ A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$ or $\{\mathbf{y} :\ A^T\mathbf{y} \leq \mathbf{c}\}$ 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 $\{\mathbf{x} :\ A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$ or $\{\mathbf{y} :\ A^T\mathbf{y} \leq \mathbf{c}\}$ is "approximately infeasible" in the sense that we have an approximate certificate in Farkas' lemma, then any feasible solution must have large norm.

Example 1.13 *Given $\epsilon > 0$ but small. Let $A = (1,\ 1)$ and $b = -\epsilon$. Then, $x = (0; 0)$ is approximately feasible for $\{\mathbf{x} :\ A\mathbf{x} = b,\ \mathbf{x} \geq \mathbf{0}\}$, and the infeasibility certificate $y = -1/\epsilon$ has a large norm.*

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

1.3.4 Linear programming (LP)

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

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b},\ \mathbf{l} \le \mathbf{x} \le \mathbf{u}, \end{array}$$

where some elements in $\mathbf{l}$ may be $-\infty$ meaning that the associated variables are unbounded from below, and some elements in $\mathbf{u}$ 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:

$$\begin{array}{lll} (LP) & \text{minimize} & \mathbf{c}^T\mathbf{x} \\ & \text{subject to} & A\mathbf{x} = \mathbf{b},\ \mathbf{x} \ge \mathbf{0}. \end{array}$$

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

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

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

and

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

Again, given an $\mathbf{x}$, to see if $\mathbf{x} \in \mathcal{S}_p$ 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:

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

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

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

Theorem 1.14 *(Weak duality theorem) Let $\mathcal{F}_p$ and $\mathcal{F}_d$ be non-empty. Then,*

$$\mathbf{c}^T\mathbf{x} \ge \mathbf{b}^T\mathbf{y} \quad \textit{for all} \quad \mathbf{x} \in \mathcal{F}_p,\ (\mathbf{y}, \mathbf{s}) \in \mathcal{F}_d.$$

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

Theorem 1.15 *(Strong duality theorem) Let $\mathcal{F}_p$ and $\mathcal{F}_d$ be non-empty. Then, $\mathbf{x}^*$ is optimal for (LP) if and only if the following conditions hold:*

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

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

iii) $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$.

Theorem 1.16 *(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

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

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

For feasible $\mathbf{x}$ and $(\mathbf{y}, \mathbf{s})$, $\mathbf{x}^T\mathbf{s} = \mathbf{x}^T(\mathbf{c} - A^T\mathbf{y}) = \mathbf{c}^T\mathbf{x} - \mathbf{b}^T\mathbf{y}$ is called the *complementarity gap*. If $\mathbf{x}^T\mathbf{s} = 0$, then we say $\mathbf{x}$ and $\mathbf{s}$ are complementary to each other. Since both $\mathbf{x}$ and $\mathbf{s}$ are nonnegative, $\mathbf{x}^T\mathbf{s} = 0$ 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

$$\begin{array}{rcl} \mathbf{x} \circ \mathbf{s} & = & \mathbf{0} \\ A\mathbf{x} & = & \mathbf{b} \\ -A^T\mathbf{y} - \mathbf{s} & = & -\mathbf{c}, \end{array} \tag{1.2}$$

where $\circ$ is the element-element or Hadamard product operator, that is,

$$\mathbf{x} \circ \mathbf{s} = \begin{pmatrix} x_1 s_1 \\ x_2 s_2 \\ \vdots \\ x_n s_n \end{pmatrix}.$$

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

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

Moreover, the support sets

$$P^* = \textit{supp}(\mathbf{x}^*) := \{j : \ x_j^* > 0\} \quad \textit{and} \quad Z^* = \textit{supp}(\mathbf{s}^*) := \{j : \ s_j^* > 0\}$$

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*. $\{\mathbf{x} : \ A_{P^*}\mathbf{x}_{P^*} = \mathbf{b}, \ \mathbf{x}_{P^*} \geq \mathbf{0}, \ \mathbf{x}_{Z^*} = \mathbf{0}\}$ is called the *primal optimal face*, and $\{\mathbf{y} : \ \mathbf{c}_{Z^*} - A_{Z^*}^T\mathbf{y} \geq \mathbf{0}, \ \mathbf{c}_{P^*} - A_{P^*}^T\mathbf{y} = \mathbf{0}\}$ 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

$$A_B\mathbf{x}_B = \mathbf{b}$$

for the m-vector $\mathbf{x}_B$. By setting the variables, $\mathbf{x}_N$, of $\mathbf{x}$ corresponding to the remaining columns of A equal to zero, we obtain a solution $\mathbf{x}$ such that

$$A\mathbf{x} = \mathbf{b}.$$

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

$$A_B^T\mathbf{y} = \mathbf{c}_B$$

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

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

then $\mathbf{x}$ 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 $\mathbf{x}_B$ has value zero, that basic solution $\mathbf{x}$ 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, $\mathbf{s}_N$, in the corresponding

dual slack vector $\mathbf{s}$, except for $\mathbf{s}_B$, are non-zero, then $\mathbf{y}$ is said to be (*dual*) nondegenerate. If both primal and dual basic solutions are nondegenerate, A_B is called a *nondegenerate basis*.

From Carathéodory's theorem, one can prove

Theorem 1.18 *(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.

1.3.5 Quadratic programming (QP)

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

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

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

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

and

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

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

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

and satisfies the complementarity condition

$$(\mathbf{x}^*)^T\mathbf{s}^* = \frac{1}{2}(\mathbf{x}^*)^T Q\mathbf{x}^* + \mathbf{c}^T\mathbf{x}^* - (\mathbf{b}^T\mathbf{y}^* - \frac{1}{2}(\mathbf{x}^*)^T Q\mathbf{x}^* = 0.$$

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

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

and

$$\begin{array}{rcl} \mathbf{x} \cdot \mathbf{s} & = & \mathbf{0} \\ A\mathbf{x} & = & \mathbf{b} \\ -A^T\mathbf{y} + Q\mathbf{x} - \mathbf{s} & = & -\mathbf{c}. \end{array} \tag{1.3}$$

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 $\mathbf{x}^*$ is an optimal solution for (QP) if and only if $\mathbf{x}^*$ 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).

1.4 Algorithms and Computations

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 $\mathbf{x}$ such that $x \in \mathcal{S}_p$ or $\mathbf{x}$ is close to $\mathcal{S}_p$ in certain measure. The latter $\mathbf{x}$ 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.

In the real number model, we introduce 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,

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

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.

1.4.1 Complexity of problems

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 1.14 *There is a strongly polynomial algorithm for sorting a vector in descending or ascending order, for multiplying a matrix by a vector, and for computing the norm of a vector.*

Example 1.15 *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* $f\frac{1}{2}$ *(counted as one operation). If* $f\frac{1}{2} > 0$*, we throw away* $[0, 1/2)$*; if* $f\frac{1}{2} < 0$*, 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 Convergence rate

Most algorithms are iterative in nature. They generate a sequence of ever-improving points $\mathbf{x}^0, \mathbf{x}^1, ..., \mathbf{x}^k, ...$ 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 may hold 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 1.2 . *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 1.3 . *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 1.4 . *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 *or* geometrically *to zero with convergence ratio* β.

Linear or geometric 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, the bisection method is linearly convergent and has a convergence ratio 0.5.

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 1.16 *Consider the conjugate gradient algorithm for minimizing* $\frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}$. *Starting from an* $\mathbf{x}^0 \in \mathcal{R}^n$ *and* $\mathbf{d}^0 = Q\mathbf{x}^0 + \mathbf{c}$, *the method uses iterative formula*

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

where

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

and

$$\mathbf{d}^{k+1} = Q\mathbf{x}^{k+1} - \theta^k \mathbf{d}^k$$

where

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

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

There is another convergence speed

Definition 1.5 . *Let the sequence* $\{r^k\}$ *converge to zero such that*

$$\frac{|r^k|}{|r^0|} \leq \frac{L}{k^p},$$

where L *is a fixed constant.Then, the sequence is said to converge* arithmetically *to zero with convergence order* $p > 0$.

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

$$A\mathbf{x} = \mathbf{b}.$$

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:

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

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 $\mathbf{x}'$. Substituting the solution $\mathbf{x}'$ 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:

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

The theory says that y^* minimizes $\|A^T\mathbf{y} - \mathbf{c}\|$ if and only if

$$AA^T\mathbf{y}^* = A\mathbf{c}.$$

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

$$L\Lambda L^T\mathbf{y}^* = A\mathbf{c},$$

and $\mathbf{y}^*$ 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 $\mathbf{f}(\mathbf{x}) : \mathcal{R}^n \to \mathcal{R}^n$, the problem is to solve n equations for n unknowns such that

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}.$$

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

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

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

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

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

$$\nabla \mathbf{f}(\mathbf{x}^k)\mathbf{d}_x = -\mathbf{f}(\mathbf{x}^k)$$

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

A modified or quasi Newton method is defined by

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

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

The Newton method has a superior asymptotic convergence order equal 2 for $\|\mathbf{f}(\mathbf{x}^k)\|$ when the sequence is convergent and $\nabla \mathbf{f}(\cdot)$ at the limit point is invertible. It is frequently used in interior-point algorithms, and believed to be the key to their effectiveness. On the other hand, the steepest descent method typically generates a sequence that converges to the limit point arithmetically.

1.5.4 Solving ball-constrained linear problem

The ball-constrained linear problem has the following form:

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

or

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

The minimizer $\mathbf{x}^*$ of (BP) is given as follows: Solve linear system

$$AA^T\bar{\mathbf{y}} = A\mathbf{c},$$

$\bar{\mathbf{y}}$; and if $\mathbf{c} - A^T\bar{\mathbf{y}} \neq \mathbf{0}$ then

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

otherwise any feasible $\mathbf{x}$ is a solution. The minimizer $\mathbf{y}^*$ of (BD) is given as follows: Solve linear system

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

for $\bar{\mathbf{y}}$; and if $A^T\bar{\mathbf{y}} \neq \mathbf{0}$ then set

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

otherwise any feasible $\mathbf{y}$ 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:

$$(BP) \quad \begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{0},\ \|\mathbf{x}\|^2 \leq 1, \end{array}$$

or simply

$$(BD) \quad \begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{y}^T Q\mathbf{y} + \mathbf{b}^T\mathbf{y} \\ \text{subject to} & \|\mathbf{y}\|^2 \leq 1. \end{array}$$

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

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

and

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

These conditions are necessary and sufficient. This problem can be solved in polynomial time $\log(1/\epsilon)$ or $\log(\log(1/\epsilon))$ 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

Most of the materials presented can be found from convex analysis, such as Rockeafellar [266].

The term "complexity" was introduced by Hartmanis and Stearns [157]. Also see Garey and Johnson [120] and Papadimitriou and Steiglitz [250]. The NP theory was due to Cook [72] and Karp [182]. The importance of P was observed by Edmonds [90].

Linear programming and the simplex method were introduced by Dantzig [75]. Other inequality problems and convexity theories can be seen in Gritzmann and Klee [143], Grötschel, Lovász and Schrijver [144], Grünbaum [145], Rockafellar [266], and Schrijver [273]. Various complementarity problems can be found found in Cottle, Pang and Stone [74]. The positive semi-definite programming, an optimization problem in nonpolyhedral cones, and its applications can be seen in Nesterov and Nemirovskii [243], Alizadeh [8], and Boyd, Ghaoui, Feron and Balakrishnan [58]. Recently, Goemans and Williamson [127] 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 [197].

It was shown by Klee and Minty [186] 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

[183]; also see Bland, Goldfarb and Todd [54]. The method was devised independently by Shor [279] and by Nemirovskii and Yudin [241]. 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 [111] and Fiacco and McCormick [106]; see Gill, Murray, Saunders, Tomlin and Wright [126], and Anstreicher [22]. For a brief LP history, see the excellent article by Wright [326].

The real computation model was developed by Blum, Shub and Smale [55] and Nemirovskii and Yudin [241]. Other complexity issues in numerical optimization were discussed in Vavasis [321].

Many basic numerical procedures listed in this chapter can be found in Golub and Van Loan [135]. The ball-constrained quadratic problem and its solution methods can be seen in Moré [231], Sorenson [284], and Dennis and Schnable [78]. The complexity result of the ball-constrained quadratic problem was proved by Vavasis [321] and Ye [335].

1.7 Exercises

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

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

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

1.2 *Prove that the eigenvalues of all symmetric matrices $X \in \mathcal{S}^n$ are real, and*

$$X = \sum_{i=1}^{r} \lambda_i \mathbf{v}_i \mathbf{v}_i^T,$$

where r is the rank of X, λ_i is an eigenvalue and $\mathbf{v}_i$ its eigenvector of X. Furthermore, show that X is PSD if and only if all its eigenvalues are nonnegative, and V is PD if and only if all its eigenvalues are positive.

1.3 *Let X be a positive semidefinite matrix of rank r, and A be any given symmetric matrix. Then, there is another decomposition of X*

$$X = \sum_{i=1}^{r} \mathbf{v}_i \mathbf{v}_i^T,$$

such that for all i,

$$\mathbf{v}_i^T A \mathbf{v}_i = A \bullet (\mathbf{v}_i \mathbf{v}_i^T) = \frac{1}{r}(A \bullet X).$$

1.4 *Prove $X \bullet S \geq 0$ if both X and S are positive semi-definite matrices. Moreover, prove that two positive semi-definite matrices are complementary to each other, $X \bullet S = 0$, if and only if $XS = \mathbf{0}$.*

1.5 *Using the ellipsoid representation in Section 1.2.2, find the matrix Q and vector $\mathbf{y}$ that describes the following ellipsoids:*

i) The 3-dimensional sphere of radius 2 centered at the origin;

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

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

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

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

1.7 *Prove Proposition 1.1.*

1.8 *Show that the non-negative orthant, the positive semi-definite cone, and the second-order cone are all self-dual. Also show that the dual cone of the p-order cone, $p = 1, ..., \infty$, is the q-order cone where $\frac{1}{q} + \frac{1}{p} = 1$.*

1.9 *When both K_1 and K_2 are closed convex cones. Show*

i) $(K_1^*)^* = K_1$.

ii) $K_1 \subset K_2 \Longrightarrow K_2^* \subset K_1^*$.

iii) $(K_1 \oplus K_2)^* = K_1^* \oplus K_2^*$.

iv) $(K_1 + K_2)^* = K_1^* \cap K_2^*$.

v) $(K_1 \cap K_2)^* = K_1^* + K_2^*$.

1.10 *Prove Proposition 1.3.*

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

i) Find the point in C that is closest to $\mathbf{y}$;

ii) Find a separating hyperplane vector as a function of $\mathbf{y}$.

iii) Using the idea of Exercise 1.11, prove the separating hyperplane theorem 1.5.

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

ii)/ Given $C \in \mathcal{S}^n$, $A_i \in \mathcal{S}^n, i = 1, \cdots, m$, *and* $\mathbf{b} \in \mathcal{R}^m$, *consider the function* $\mathcal{B}(\mathbf{y}) := -\ln det(S)$, *where* $S = C - \sum_{i=1}^{m} y_i A_i \succ \mathbf{0}$. *Find* $\nabla \mathcal{B}(\mathbf{y})$ *and* $\nabla^2 \mathcal{B}(\mathbf{y})$ *in terms of* S.

The best way to do this is to use the definition of the partial derivative

$$\nabla f(\mathbf{y})_i = \lim_{\delta \to 0} \frac{f(y_1, y_2, ..., y_i + \delta, ..., y_m) - f(y_1, y_2, ..., y_i, ..., y_m)}{\delta}.$$

1.13 *Let* $f(\mathbf{x}) : \mathcal{R}^n_{++} \to R$ *be a given convex function. Show that the function* $g : \mathcal{R}^{n+1}_{++} \to R$ *given by* $g(\tau; \mathbf{x}) = \tau \cdot f(\mathbf{x}/\tau)$ *(called the homogenized version of* f*) is also a convex function in the domain of* $(\tau; \mathbf{x}) \in R^{n+1}_{++}$. *Now, suppose that* $f(\mathbf{x})$ *is twice-differentiable. Write out the gradient vector and Hessian matrix of* g.

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

1.15 *Prove Propositions 1.7 and 1.8 for convex functions in Section 1.2.3.*

1.16 *Let* $f_1, \ldots, f_m$ *be convex functions. Then, the function* $\bar{f}(\mathbf{x})$ *defined below is also convex:*

i)

$$\max_{i=1,\ldots,m} f_i(\mathbf{x})$$

ii)

$$\sum_{i=1}^{m} f_i(\mathbf{x})$$

1.17 *Prove Farkas' lemma 1.11 for linear equations.*

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

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

i) $P = P^2$.

ii) P *is positive semi-definite.*

iii) The eigenvalues of P *are either* 0 *or* 1.

1.20 *Using the separating theorem, prove Farkas' lemmas 1.12 and 1.13.*

1.21 *Prove the LP fundamental theorem 1.18.*

1.22 *If (LP) and (LD) have a nondegenerate optimal basis* A_B, *prove that the strict complementarity partition in Theorem 1.17 is*

$$P^* = B.$$

1.23 *If Q is positive semi-definite, prove that $\mathbf{x}^*$ is an optimal solution for (QP) if and only if $\mathbf{x}^*$ is a KKT point for (QP).*

1.24 *Given an (LP) data set $(A, \mathbf{b}, \mathbf{c})$ and an interior feasible point $\mathbf{x}^0$, find the feasible direction $\mathbf{d}_x$ ($A\mathbf{d}_x = 0$) that achieves the steepest decrease in the objective function.*

1.25 *Given an (LP) data set $(A, \mathbf{b}, \mathbf{c})$ and a feasible point $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in (\mathcal{R}^n_+, \mathcal{R}^m, \mathcal{R}^n_+)$ 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) and (1.3), respectively.*

1.26 *Down load SEDUMI1.05, DSDP5.8, and/or CVX and install them in Matlab or R. Solve the SDP example in Example 1.1.*

Chapter 2

Conic Linear Programming

2.1 Conic Linear Programming and its Dual

Given a closed convex cone $K \subset \mathcal{E}$, $C \in \mathcal{E}$, $A_i \in \mathcal{E}$, $i = 1, 2, ..., m$, and $\mathbf{b} \in \mathcal{R}^m$, the conic linear programming problem is to find a matrix $X \in C$ for the optimization problem in a canonical form:

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

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

$$A \bullet B := \text{tr}(A^T B).$$

We put here "inf" instead of "minimize", since the minimal objective value may exist, but it cannot be attained at a finite solution. With this understanding, we will use "minimize" through out this monograph.

When $K = \mathcal{R}^n_+$, an element in K is conventionally written as $\mathbf{x} \geq \mathbf{0}$ or $\mathbf{x}$ is component-wise nonnegative; while when $K = \mathcal{S}^n_+$, an element in K is conventionally written as $X \succeq \mathbf{0}$ or X is a positive semi-definite matrix. Furthermore, $X \succ \mathbf{0}$ means that X is a positive definite matrix. If a point X is in the interior of K and satisfies all equations in (CLP), it is called a (*primal*) strictly or interior feasible solution. .

Note that the semidefinite programming example in Chapter 1

$$\begin{array}{ll} \text{minimize} & 2x_1 + x_2 + x_3 \\ \text{subject to} & x_1 + x_2 + x_3 = 1, \\ & \begin{pmatrix} x_1 & x_2 \\ x_2 & x_3 \end{pmatrix} \succeq \mathbf{0} \end{array}$$

can be written in the canonical form

$$\begin{array}{ll} \text{minimize} & C \bullet X \\ \text{subject to} & A_1 \bullet X = 1, \\ & X \succeq \mathbf{0}, \end{array}$$

where

$$C = \begin{pmatrix} 2 & .5 \\ .5 & 1 \end{pmatrix} \quad \text{and} \quad A_1 = \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix}.$$

For semidefinite programming, the coefficient matrices C and A_i are not necessarily symmetric in applications. However, since

$$A_i \bullet X = \frac{1}{2}(A_i + A_i^T) \bullet X,$$

we can use $\frac{1}{2}(A_i + A_i^T)$ to replace original A_i. Therefore, without loss of generality, we assume that C and A_i in the canonical SDP form are all symmetric.

Example 2.1 *Besides the two conic problems presented in Introduction of Chapter 1, the following example is a second-order cone programming (SOCP) problem:*

$$\begin{array}{ll} \text{minimize} & 2x_1 + x_2 + x_3 \\ \text{subject to} & x_1 + x_2 + x_3 = 1, \\ & \sqrt{x_2^2 + x_3^2} \leq x_1. \end{array}$$

Here, like linear programming, C and A_1 are vectors in $\mathcal{R}^3$:

$$C = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \quad \textit{and} \quad A_1^T = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

For convenience, we define an operator from space E to a vector:

$$\mathcal{A}X := \begin{pmatrix} A_1 \bullet X \\ A_2 \bullet X \\ \cdots \\ A_m \bullet X \end{pmatrix}. \tag{2.1}$$

Then, (CLP) can be written in a compact form:

$$\begin{array}{lll} (CLP) & \text{minimize} & C \bullet X \\ & \text{subjectto} & \mathcal{A}X = \mathbf{b}, \\ & & X \in K. \end{array} \tag{2.2}$$

Note that X may be decomposed as product several separate and mixed cones linked by the linear constraints such as

$$\begin{array}{lll} (CLP) & \text{minimize} & \sum_{l=1}^{p} C_k \bullet X_l \\ & \text{subjectto} & \sum_{l=1}^{p} \mathcal{A}_l X_l = \mathbf{b}, \\ & & X_l \in K_l,\ l = 1, ..., p, \end{array} \tag{2.3}$$

where K_l could be any closed convex cones. Recall that we can stack the variables and write it as

$$X = (X_1; X_2; ...; X_p) \in K_1 \oplus K_2 \oplus ... \oplus K_p.$$

2.1.1 Dual of conic linear programming

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

$$\begin{array}{lll} (CLD) & \text{maximize} & \mathbf{b}^T\mathbf{y} \\ & \text{subject to} & \sum_i^m y_i A_i + S = C,\ S \in K^*, \end{array}$$

which is analogous to the dual of linear programming. Here $\mathbf{y} \in \mathcal{R}^m$ and $S \in \mathcal{E}$. If a point $(\mathbf{y}, S)$ satisfies all equations in (SDD) and S is in the interior of K^*, it is called a *dual interior feasible solution.* Again, the maximal objective value of the dual may exist, but it cannot be attained at a finite solution. Just as in LP, the dual of the dual would be the primal problem.

Example 2.2 *Here are dual problems to the three examples where* $\mathbf{y}$ *is just a scalar.*

- *The dual to the LP example:*

$$\begin{array}{lll} & \text{maximize} & y \\ & \text{subject to} & y \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \mathbf{s} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \\ & & \mathbf{s} = (s_1;\ s_2;\ s_3) \in K^* = \mathcal{R}^3_+. \end{array}$$

- *The dual to SDP example:*

$$\begin{array}{lll} & \text{maximize} & y \\ & \text{subject to} & y \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix} + S = \begin{pmatrix} 2 & .5 \\ .5 & 1 \end{pmatrix}, \\ & & S \in K^* = \mathcal{S}^2_+. \end{array}$$

- *The dual to the SOCP example:*

$$\begin{array}{lll} & \text{maximize} & y \\ & \text{subject to} & y \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \mathbf{s} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \\ & & \mathbf{s} = (s_1;\ s_2;\ s_3) \in K^* = \mathcal{N}^3_2. \end{array}$$

Some computational problems can be directly written in the CLD form.

Example 2.3 *Let* $P(\mathbf{y} \in \mathcal{R}^m) = C - \sum_i^m y_i A_i$, *where* C *and* A_i, $i = 1, \ldots, m$, *are given symmetric matrices. The problem of maximizes the min-eigenvalue of* $P(\mathbf{y})$ *can be cast as a (CLD) problem:*

$$\begin{array}{lll} & \text{maximize} & y_0 \\ & \text{subject to} & y_0 I + \sum_i^m y_i A_i + S = C,\ S \succeq \mathbf{0}. \end{array}$$

Then, the dual problem would in CLP form:

$$\begin{array}{lll} & \text{minimize} & C \bullet X \\ & \text{subject to} & A_i \bullet X = 0,\ i = 1, \ldots, m, \\ & & I \bullet X = 1,\ X \succeq \mathbf{0}. \end{array}$$

Let us define the reverse operator of (2.1) from a vector to E:

$$\mathcal{A}^T\mathbf{y} = \sum_{i=1}^{m} y_i A_i. \tag{2.4}$$

Note that, by the definition, for any matrix $X \in E$

$$\mathcal{A}^T\mathbf{y} \bullet X = \mathbf{y}^T(\mathcal{A}X),$$

that is, the association property holds. Then, (CLD) can be written in a compact form:

$$\begin{array}{lll} (CLD) & \text{maximize} & \mathbf{b}^T\mathbf{y} \\ & \text{subjectto} & \mathcal{A}^T\mathbf{y} + S = C, \\ & & S \in K^*. \end{array} \tag{2.5}$$

The dual to the mixed conic problem 2.3 would be

$$\begin{array}{lll} (CLD) & \text{minimize} & \mathbf{b}^T\mathbf{y} \\ & \text{subjectto} & \mathcal{A}_l^T\mathbf{y} + S_l = C_l,\ l = 1, ..., p, \\ & & S_l \in K_l^*,\ l = 1, ..., p, \end{array} \tag{2.6}$$

where the last constraint can be written as

$$S = (S_1; S_2; ...; S_p) \in K_1^* \oplus K_2^* \oplus ... \oplus K_p^*.$$

Example 2.4 *(Euclidean Facility Location). This problem is to determine the location of a facility serving n clients placed in a Euclidean space, whose known locations are denoted by* $\mathbf{a}_l \in \mathcal{R}^d,\ l = 1, \ldots, n$. *The location of the facility would minimize the sum of the Euclidean distances from the facility to each of the clients. Let the location decision be vector* $\mathbf{f} \in \mathcal{R}^d$. *Then the problem is*

$$\text{minimize} \quad \textstyle\sum_{l=1}^{n} \|\mathbf{f} - \mathbf{a}_l\|. \ .$$

The problem can be reformulated as

$$\begin{array}{lll} \text{minimize} & \sum_{l=1}^{n} \delta_l & \\ \text{subjectto} & \mathbf{s}_j + \mathbf{f} = \mathbf{a}_l, & \forall l = 1, ..., n, \\ & \|\mathbf{s}_l\| \leq \delta_l, & \forall l = 1, ..., n. \end{array}$$

This is a conic formulation in the (CLD) form. To see it clearly, let $d = 2$ and $n = 3$ in the example, and let

$$A_1 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ A_2 = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ A_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \in \mathcal{R}^5,\ \mathbf{c}_1 = \begin{pmatrix} 0 \\ \mathbf{a}_1 \end{pmatrix} \in \mathcal{R}^3,\ \mathbf{c}_2 = \begin{pmatrix} 0 \\ \mathbf{a}_2 \end{pmatrix} \in \mathcal{R}^3,\ \mathbf{c}_2 = \begin{pmatrix} 0 \\ \mathbf{a}_3 \end{pmatrix} \in \mathcal{R}^3,$$

and variable vector

$$\mathbf{y} = (\delta_1;\ \delta_2;\ \delta_3;\ \mathbf{f}) \in \mathcal{R}^5.$$

Then, the facility location problem becomes

$$\begin{array}{ll} \text{minimize} & \mathbf{b}^T\mathbf{y} \\ \text{subject to} & A_l^T\mathbf{y} + \mathbf{s}_l = \mathbf{c}_l,\ i = 1, 2, 3, \\ & \mathbf{s}_l \in K_l = \mathcal{N}_2^3; \end{array}$$

that is, each K_l is the 3-dimensional second-order cone. The dual of the facility location problem would be in the (CLP) form:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{l=1}^{3} \mathbf{c}_l^T\mathbf{x}_l \\ \text{subject to} & \displaystyle\sum_{l=1}^{3} A_l\mathbf{x}_l = -\mathbf{b}, \\ & \mathbf{x}_l \in K_l^* = \mathcal{N}_2^3, \end{array}$$

where dual decision vector $\mathbf{x}_l \in \mathcal{R}^3$.

If one like to choose location $\mathbf{f}$ to

$$\text{minimize} \quad \textstyle\sum_{l=1}^{n} \|\mathbf{f} - \mathbf{a}_l\|_p, \quad p \geq 1\ ,$$

the same formulation hods but $K_l = \mathcal{N}_p^3$ and $K_l^* = \mathcal{N}_q^3$, where $\frac{1}{p} + \frac{1}{q} = 1$.

2.2 Farkas' Lemma and Duality Theorem of Conic Linear Programming

Let us consider the feasible region of (CLP) in (2.2):

$$\mathcal{F}_p := \{X :\ \ \mathcal{A}X = \mathbf{b},\ X \in K\};$$

where the interior of the feasible region is

$$\overset{\circ}{\mathcal{F}}_p := \{X :\ \ \mathcal{A}X = \mathbf{b},\ X \in \overset{\circ}{K}\}.$$

If $K = \mathcal{R}_+^n$ in linear programming and $\mathcal{F}_p$ is empty, from Farkas' lemma, a vector $\mathbf{y} \in \mathcal{R}^m$, with $-A^T\mathbf{y} \geq \mathbf{0}$ and $\mathbf{b}^T\mathbf{y} > 0$, always exists and it is an infeasibility certificate for $\mathcal{F}_p$. Does this alternative relations hold for K being a general closed convex one?

2.2.1 Alternative theorem for conic systems

Let us rigorousize the question: when $\mathcal{F}_p$ is empty, does there exist a vector $\mathbf{y} \in \mathcal{R}^m$ such that $-\mathcal{A}^T\mathbf{y} \in K^*$ and $\mathbf{b}^T\mathbf{y} > 0$? Similarly, one can ask: when set $\mathcal{F}_d := \{\mathbf{y} : \ C - \mathcal{A}^T\mathbf{y} \in K^*\}$ is empty, does there exist an $X \in K$ such that $\mathcal{A}X = \mathbf{0}$ and $C \bullet X < 0$? Note that the answer to the second question is also "yes" when $K = \mathcal{R}^n_+$.

Example 2.5 *The answer to either question is "not true in general"; see examples below.*

- *For the first question, consider $K = \mathcal{S}^2_+$ and*

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

and

$$\mathbf{b} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

- *For the second question, consider $K = \mathcal{S}^2_+$ and*

$$C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \textit{and} \quad A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

However, if the data set $\mathcal{A}$ satisfies additional conditions, the answer would be "yes"; see theorem below.

Theorem 2.1 *(Farkas' lemma for CLP) The following results hod.*

i) Consider set

$$\mathcal{F}_p := \{X : \ \mathcal{A}X = \mathbf{b}, \ X \in K\}.$$

Suppose that there exists a vector $\hat{\mathbf{y}}$ such that $-\mathcal{A}^T\hat{\mathbf{y}} \in \overset{\circ}{K}{}^*$. Then,

1. *Set $C := \{\mathcal{A}X \in \mathcal{R}^m : \ X \in K\}$ is a closed convex set;*
2. *$\mathcal{F}_p$ has a (feasible) solution if and only if set $\{\mathbf{y} : \ -\mathcal{A}^T\mathbf{y} \in K^*, \ \mathbf{y}^T\mathbf{b} > 0\}$ has no feasible solution.*

ii) Consider set

$$\mathcal{F}_d := \{\mathbf{y} : \ C - \mathcal{A}^T\mathbf{y} \in K\}.$$

Suppose that there exists a vector $\hat{X} \in \overset{\circ}{K}{}^*$ such that $\mathcal{A}\hat{X} = \mathbf{0}$. Then,

1. *Set $C := \{S - \mathcal{A}^T\mathbf{y} : \ S \in K\}$ is a closed convex set;*
2. *$\mathcal{F}_d$ has a (feasible) solution if and only if set $\{X : \ \mathcal{A}X = \mathbf{0}, \ X \in K^*, \ C \bullet X < 0\}$ has no feasible solution.*

Proof.We prove the first part of i) of the theorem. It is clear that C is a convex set. To prove that C is a closed set, we need to show that if $\mathbf{y}^k := \mathcal{A}X^k \in \mathcal{R}^m$ for $X^k \in K$, $k = 1, \ldots$, converges to a vector $\bar{\mathbf{y}}$, then $\bar{\mathbf{y}} \in C$, that is, there is $\bar{X} \in K$ such that $\bar{\mathbf{y}} := \mathcal{A}\bar{X}$. Without loss of generality, we assume that $\mathbf{y}^k$ is a bounded sequence. Then, there is a positive constant c such that

$$c \geq -\hat{\mathbf{y}}^T\mathbf{y}^k = -\hat{\mathbf{y}}^T(\mathcal{A}X^k) = -\mathcal{A}^T\hat{\mathbf{y}} \bullet X^k, \forall k.$$

Since $-\mathcal{A}^T\hat{\mathbf{y}} \in \overset{\circ}{K}{}^*$, from Proposition 1.3, the sequence of X^k is also bounded. Then there is at least one accumulate point $\bar{X} \in K$ for the sequence because K is a closed cone. Thus, we must have $\bar{\mathbf{y}} := \mathcal{A}\bar{X}$, which prove that C is closed.

We now prove the second part. Let $\mathbf{y}$ be $-\mathcal{A}^T\mathbf{y} \in K^*$. Then, if $\mathcal{F}_p$ has a feasible solution $\bar{X}$,

$$-\mathbf{y}^T\mathbf{b} = -\mathbf{y}^T(\mathcal{A}\bar{X}) = -\mathcal{A}^T\mathbf{y} \bullet \bar{X} \geq 0.$$

Thus, it must be true $\mathbf{b}^T\mathbf{y} \leq 0$, that is, $\{\mathbf{y} : \ -\mathcal{A}^T\mathbf{y} \in K^*, \ \mathbf{b}^T\mathbf{y} > 0\}$ must be empty.

On the other hand, let $\mathcal{F}_p$ has no feasible solution, or equivalently, $\mathbf{b} \notin C$. We now show that $\{\mathbf{y} : \ -\mathcal{A}^T\mathbf{y} \in K^*, \ \mathbf{y}^T\mathbf{b} > 0\}$ must be nonempty.

Since C is a closed convex set, from the separating hyperplane theorem, there must exist a $\bar{\mathbf{y}} \in \mathcal{R}^m$ such that

$$\bar{\mathbf{y}}^T\mathbf{b} > \bar{\mathbf{y}}^T\mathbf{y}, \ \forall \mathbf{y} \in C.$$

Or, from $\mathbf{y} = \mathcal{A}X, \ X \in K$, we have

$$\bar{\mathbf{y}}^T\mathbf{b} > \bar{\mathbf{y}}^T(\mathcal{A}X) = \mathcal{A}^T\bar{\mathbf{y}} \bullet X, \ \forall X \in K.$$

That is, $\mathcal{A}^T\bar{\mathbf{y}} \bullet X$ is bounded above for all $X \in K$.

Immediately, we see $\bar{\mathbf{y}}^T\mathbf{b} > 0$ since $\mathbf{0} \in K$. Next, it must be true $-\mathcal{A}^T\bar{\mathbf{y}} \in K^*$. Otherwise, we must be able to find an $\bar{X} \in K$ such that $-\mathcal{A}^T\bar{\mathbf{y}} \bullet \bar{X} < 0$ by the definition of K and its dual K^*. For any positive constant α we keep $\alpha\bar{X} \in K$ and let α go to ∞. Then, $\mathcal{A}^T\bar{\mathbf{y}} \bullet (\alpha\bar{X})$ goes to ∞, contradicting the fact that $\mathcal{A}^T\bar{\mathbf{y}} \bullet X$ is bounded above for all $X \in K$. Thus, $\bar{\mathbf{y}}$ is a feasible solution of $\{\mathbf{y} : \ -\mathcal{A}^T\mathbf{y} \in K^*, \ \mathbf{y}^T\mathbf{b} > 0\}$. ∎

Note that C may not be a closed set if the interior condition of Theorem 2.1 is not met. Consider A_1, A_2 and $\mathbf{b}$ in Example 2.5, and we have

$$C = \left\{ \mathcal{A}X = \begin{pmatrix} A_1 \bullet X \\ A_2 \bullet X \end{pmatrix} : \ X \in \mathcal{S}^2_+ \right\}.$$

Let

$$X^k = \begin{pmatrix} \frac{1}{k} & 1 \\ 1 & k \end{pmatrix} \in \mathcal{S}^2_+, \ \forall k = 1, \ldots.$$

Then we see

$$\mathbf{y}^k = \mathcal{A}X^k = \begin{pmatrix} \frac{1}{k} \\ 2 \end{pmatrix}.$$

As $k \to \infty$ we see $\mathbf{y}^k$ converges $\mathbf{b}$, but $\mathbf{b}$ is *not* in C.

2.2.2 Duality theorem for conic linear programming

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

Theorem 2.2 *(Weak duality theorem in CLP) Let $\mathcal{F}_p$ and $\mathcal{F}_d$ be non-empty. Then,*

$$C \bullet X \geq \mathbf{b}^T\mathbf{y} \quad \textit{for all} \quad X \in \mathcal{F}_p,\ (\mathbf{y}, S) \in \mathcal{F}_d.$$

Proof.By direct calculation

$$\begin{aligned} C \bullet X - \mathbf{b}^T\mathbf{y} &= \left(\sum_{i=1}^{m} y_i A_i + S\right) \bullet X - \mathbf{b}^T\mathbf{y} \\ &= \sum_{i=1}^{m} y_i (A_i \bullet X) + S \bullet X - \mathbf{b}^T\mathbf{y} \\ &= \sum_{i=1}^{m} y_i b_i + S \bullet X - \mathbf{y}^T\mathbf{b} \\ &= S \bullet X \geq 0, \end{aligned}$$

where the last inequality comes from $X \in K$ and $S \in K^*$. ∎

As in linear programming, $C \bullet X - \mathbf{b}^T\mathbf{y}$ is called the duality gap of (CLP) and (CLD). Linear programming admits a strong duality theorem: when both $\mathcal{F}_p$ and $\mathcal{F}_d$ are nonempty, then there is no gap at optimality. Does such strong duality theorem hold for conic linear programming in general? The answer is "not".

Example 2.6 *The following semidefinite program has a duality gap:*

$$C = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ A_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ A_2 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

and

$$\mathbf{b} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

The primal minimal objective value is 0 *achieved by*

$$X = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and the dual maximal objective value is -2 *achieved by* $\mathbf{y} = (0;\ -1)$*; so the duality gap is* 2*.*

This is in contrast to linear programming. However, under certain technical conditions, there would be no duality gap at optimality. One condition is related

to weather or not the primal feasible region $\mathcal{F}_p$ or dual feasible region has an interior feasible solution. Recall

$$\overset{\circ}{\mathcal{F}}_p := \{X : \ \mathcal{A}X = \mathbf{b}, \ X \in \overset{\circ}{K}\}$$

and

$$\overset{\circ}{\mathcal{F}}_d := \{(\mathbf{y}, S) : \ \mathcal{A}^T\mathbf{y} + S = C, \ S \in \overset{\circ}{K^*}\}.$$

We state here a version of the strong duality theorem for (CLP) and (CLD).

Theorem 2.3 *(Strong duality theorem in CLP)*

i) *Let (CLP) or (CLD) be infeasible, and furthermore the other be feasible and has an interior. Then the other is unbounded.*

ii) *Let (CLP) and (CLD) be both feasible, and furthermore one of them has an interior. Then there is no duality gap at optimality between (CLP) and (CLD).*

iii) *Let (CLP) and (CLD) be both feasible and have interior. Then, both have optimal solutions with no duality gap.*

Proof. We let cone $H = K \oplus R_+$ in the following proof. Note that $\overset{\circ}{H} = \overset{\circ}{K} \oplus R_{++}$.

i) Suppose $\mathcal{F}_d$ is empty and $\mathcal{F}_p$ is feasible and has an interior feasible solution. Then, we have an $\bar{X} \in \overset{\circ}{K}$ and $\bar{\tau} = 1$ that is an interior feasible solution to (homogeneous) conic system:

$$\mathcal{A}\bar{X} - \mathbf{b}\bar{\tau} = \mathbf{0}, \ (\bar{X}, \bar{\tau}) \in \overset{\circ}{H} .$$

Now, for any $z^* \in \mathcal{R}$, we form an alternative system pair based on Farkas' Lemma (see ii) of Theorem 2.1):

$$\{(X, \tau) : \ \mathcal{A}X - \mathbf{b}\tau = \mathbf{0}, \ C \bullet X - z^*\tau < 0, \ (X, \tau) \in H\},$$

and

$$\{(\mathbf{y}, S, \kappa) : \ \mathcal{A}^T\mathbf{y} + S = C, \ -\mathbf{b}^T\mathbf{y} + \kappa = -z^*, \ (S, \kappa) \in H^*\}.$$

But the latter is infeasible, so that the former has a feasible solution (X, τ). At such a feasible solution, if $\tau > 0$, we have $\mathcal{A}(X/\tau) = \mathbf{b}$, $(X/\tau) \in K$, and $C \bullet (X/\tau) < z^*$ for any z^*, which indicates there is feasible solution in $\mathcal{F})p$ whose objective value goes to $-\infty$. Otherwise, $\tau = 0$ implies that a new solution $\bar{X} + \alpha X$ is feasible for (CLP) for any positive α; and, as $\alpha \to \infty$, the objective value of the new solution also goes to $-\infty$. Hence, either way we have a feasible solution for (CLP) whose objective value is unbounded from below.

ii) Let $\mathcal{F}_p$ be feasible and have an interior feasible solution, and let z^* be its objective infimum. Again, we have an alternative system pair as listed in the proof of i). But now the former is infeasible, so that we have a solution for the latter. From the Weak Duality theorem $\mathbf{b}^T\mathbf{y} \leq z^*$, thus we must have $\kappa = 0$, that is, we have a solution $(\mathbf{y}, S)$ such that

$$\mathcal{A}^T\mathbf{y} + S = C,\ \mathbf{b}^T\mathbf{y} = z^*,\ S \in K^*.$$

iii) We only need to prove that there exist a solution $X \in \mathcal{F}_p$ such that $C \bullet X = z^*$, that is, the infimum of (CLP) is attainable. But this is just the other side of the proof given that $\mathcal{F}_d$ is feasible and has an interior feasible solution, and z^* is also the supremum of (CLD).

∎

Again, if one of (CLP) and (CLD) has no interior feasible solution, the common optimal objective value may not be attainable. For example,

$$C = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},\ A_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \text{and} \quad b_1 = 2.$$

The dual is feasible but has no interior, while the primal has an interior. The common objective value equals 0, but no primal solution attaining the infimum value.

2.2.3 Optimality conditions of conic linear programming

Most of these examples that make the strong duality failed are superficial, and a small perturbation would overcome the failure. Thus, in real applications and in the rest of the chapter, we may assume that both (CLP) and (CLD) have interior when they are feasible. Consequently, any primal and dual optimal solution pair must satisfy the optimality conditions:

$$\begin{array}{rcll} C \bullet X - \mathbf{b}^T\mathbf{y} & = & 0 & \\ \mathcal{A}X & = & \mathbf{b} & \\ -\mathcal{A}^T\mathbf{y} - S & = & -C & ; \\ X \in K, & & S \in K^* & \end{array} \tag{2.7}$$

which is a system of linear conic inequalities and equations.

These conditions can be rewritten in the complementarity gap form

$$\begin{array}{rcll} X \bullet S & = & \mathbf{0} & \\ \mathcal{A}X & = & \mathbf{b} & \\ -\mathcal{A}^T\mathbf{y} - S & = & -C & . \\ X \in K, & & S \in K^* & \end{array} \tag{2.8}$$

Example 2.7 *Here are optimal solutions to the three examples presented earlier.*

1) The LP primal and dual example:

$$\begin{array}{rl} \text{minimize} & \begin{pmatrix} 2 & 1 & 1 \end{pmatrix} \mathbf{x} \\ \text{subject to} & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \mathbf{x} = 1, \\ & \mathbf{x} \geq \mathbf{0} \end{array} \qquad \begin{array}{rl} \text{maximize} & y \\ \text{subject to} & y \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \mathbf{s} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \\ & \mathbf{s} \geq \mathbf{0}. \end{array}$$

A complementarity pair is at $y^* = 1$:

$$\mathbf{x}^* = \begin{pmatrix} 0 \\ 0.5 \\ 0.5 \end{pmatrix} \quad \textit{and} \quad \mathbf{s}^* = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Note that the primal has multiple optimal solutions.

2) The SDP primal and dual example:

$$\begin{array}{rl} \text{minimize} & \begin{pmatrix} 2 & .5 \\ .5 & 1 \end{pmatrix} \bullet X \\ \text{subject to} & \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix} \bullet X = 1, \\ & X \succeq \mathbf{0}. \end{array} \qquad \begin{array}{rl} \text{maximize} & y \\ \text{subject to} & y \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix} + S = \begin{pmatrix} 2 & .5 \\ .5 & 1 \end{pmatrix}, \\ & S \succeq \mathbf{0}. \end{array}$$

A complementarity pair is at $y^* = 1$:

$$X^* = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad \textit{and} \quad S^* = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

This SDP optimal solution pair is unique.

3) The SOCP primal and dual example:

$$\begin{array}{rl} \text{minimize} & \begin{pmatrix} 2 & 1 & 1 \end{pmatrix} \mathbf{x} \\ \text{subject to} & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \mathbf{x} = 1, \\ & \sqrt{x_2^2 + x_3^2} \leq x_1 \end{array} \qquad \begin{array}{rl} \text{maximize} & y \\ \text{subject to} & y \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \mathbf{s} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \\ & \sqrt{s_2^2 + s_3^2} \leq s_1. \end{array}$$

A complementarity pair is at $y^* = \sqrt{2}$:

$$\mathbf{x}^* = \begin{pmatrix} \sqrt{2} - 1 \\ 1 - \frac{1}{\sqrt{2}} \\ 1 - \frac{1}{\sqrt{2}} \end{pmatrix} \quad \textit{and} \quad \mathbf{s}^* = \begin{pmatrix} 2 - \sqrt{2} \\ 1 - \sqrt{2} \\ 1 - \sqrt{2} \end{pmatrix}.$$

This optimal solution pair is unique.

2.3 Exact Low-Rank SDP Solutions

In this section, we consider $K = K^* = \mathcal{S}^n_+$ and write (CLP) and (CLD) as (SDP) and (SDD), respectively.

In linear programming, since $\mathbf{x} \geq \mathbf{0}$ and $\mathbf{s} \geq \mathbf{0}$,

$$0 = \mathbf{x} \bullet \mathbf{s} = \mathbf{x}^T\mathbf{s} = \sum_{j=1}^{n} x_j s_j$$

implies that $x_j s_j = 0$ for all $j = 1, \ldots, n$. Now consider semidefinite cone $\mathcal{S}^n_+$. Since $X \succeq \mathbf{0}$ and $S \succeq \mathbf{0}$, $X \bullet S = 0$ implies $XS = \mathbf{0}$ (Exercise 1.4), that is, the regular matrix product of the two is a zero matrix. In other words, every column (or row) of X is orthogonal to every column (or row) of S at optimality. We also call such property complementarity. Thus, besides feasibility, an optimal semidefinite programming solution pair must satisfy complementarity, that is, condition 2.8 can be further represented by

$$\begin{array}{rcl} XS & = & \mathbf{0} \\ \mathcal{A}X & = & \mathbf{b} \\ -\mathcal{A}^T\mathbf{y} - S & = & -C \\ X \in K, & & S \in K^* \end{array} . \tag{2.9}$$

Therefore, we have

Proposition 2.4 *Let X^* and $(\mathbf{y}^*, S^*)$ be any optimal SDP solution pair with zero duality gap. Then complementarity of X^* and S^* implies*

$$\mathit{rank}(X^*) + \mathit{rank}(S^*) \leq n.$$

Furthermore, is there an optimal (dual) S^* such that $\mathit{rank}(S^*) \geq r$, then the rank of any optimal (primal) X^* is bounded above by $n - r$, where integer $0 \leq r \leq n$; and the converse is also true.

In certain SDP problems, one may be interested in finding an optimal solution whose rank is minimal, while the interior-point algorithm for SDP (developed later) typically generates solution whose rank is maximal for primal and dual, respectively. Thus, a rank reduction method sometimes is necessary to achieve the low-rank goal.

2.3.1 Exact low-rank theorem

For linear programming in the standard form, it is known from Carathéodory's theorem that if there is a feasible solution, then there is a *basic* feasible solution whose positive entries are at most m many, the number of linear equations; and if there an optimal solution, then there is an optimal *basic* solution. We now explore a similar sparse structure for SDP.

Theorem 2.5 *Consider (SDP), that is, (CLP) with $K = \mathcal{S}^n_+$.*

i) If there is a feasible solution for (SDP), then there is a feasible solution whose rank r satisfies $\frac{r(r+1)}{2} \leq m$.

ii) If there is an optimal solution for (SDP), then there is an optimal solution whose rank r satisfies $\frac{r(r+1)}{2} \leq m$.

Proof.We first work on the first part of the Theorem. Similar to the proof of Carathéodory's theorem, we use null-space reduction to construct a low-rank SDP solution. Let $X \succeq \mathbf{0}$ be a feasible solution, that is,

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

and its rank r makes $r(r+1)/2 > m$. Then, we can factorize X

$$X = V^T V, \quad V \in \mathcal{R}^{r \times n},$$

and consider variable symmetric matrix $U \in \mathcal{S}^r$ to satisfy the dimension-reduced system:

$$V A_i V^T \bullet U = b_i,\ i = 1, ..., m,\ U \succeq \mathbf{0}.$$

Note that any feasible U gives a feasible $X(U) = V^T U V$ for (SDP), since $V^T U V \succeq \mathbf{0}$ and

$$A_i \bullet X(U) = A_i \bullet V^T U V = V A_i V^T \bullet U = b_i,\ i = 1, ...m.$$

In particular $U = I$ is a feasible solution to the reduced system, since $X(I) = X$.

Now consider the system of homogeneous linear equations:

$$V A_i V^T \bullet W = 0,\ i = 1, ..., m.$$

where $W \in \mathcal{S}^r$ (i.e., a $r \times r$ symmetric matrices that does not need to be semidefinite). This system has $r(r+1)/2$ real variables and m equations. Thus, as long as $r(r+1)/2 > m$, we must be able to find a symmetric matrix $W \neq \mathbf{0}$ to satisfy all the m equations. Without loss of generality, let W be either indefinite or negative semidefinite (if it is positive semidefinite, we take $-W$ as W), that is, W have at least one negative eigenvalue. Then we consider

$$U(\alpha) = I + \alpha W.$$

Choosing a α^* sufficiently large such that $U(\alpha^*) \succeq \mathbf{0}$ and it has at least one 0 eigenvalue (or $\text{rank}(U(\alpha^*)) < r$). Note that

$$V A_i V^T \bullet U(\alpha^*) = V A_i V^T \bullet (I + \alpha^* W) = V A_i V^T \bullet I = b_i,\ i = 1, ..., m.$$

That is, $U(\alpha^*)$ is feasible for the reduced system. Thus, $X(U(\alpha^*))$ is feasible for (SDP) and its rank is strictly less than r. This process can be repeated till the system of homogeneous linear equations has only all-zero solution, which is true only when $r(r+1)/2 \leq m.$, which concludes the proof.

We now work on the second part. Let X^* be an optimal solution of (SDP) with rank r and S^* be any dual optimal solution. Again, we factorize X^*

$$X^* = (V^*)^T V^*, \quad V^* \in \mathcal{R}^{r\times n}.$$

Then we consider a dimension-reduced SDP problem

$$\begin{array}{lll} \text{minimize} & V^*C(V^*)^T \bullet U & \\ \text{subjectto} & V^*A_i(V^*)^T \bullet U & = b_i,\ i = 1, ..., m \\ & U \in \mathcal{S}^r_+. & \end{array} \tag{2.10}$$

Again, from any feasible solution of (2.10) one can construct a feasible solution for (SDP) using $X^*(U) = (V^*)^T U V^*$. Furthermore, $X^*(U)$ is actually optimal for (SDP), since

$$S^* \bullet X^*(U) = S^* \bullet (V^*)^T U V^* = V^* S^* (V^*)^T \bullet U = \mathbf{0} \bullet U = 0,$$

that is, $X^*(U)$ meets the complementarity so that it is optimal. Here, we have used the fact that $S^*X^* = \mathbf{0}$ implies $V^*S^*(V^*)^T = \mathbf{0}$. Now, we can apply the construct proof of the first part to prove the second part. ∎

The following example shows that the rank-bound in Theorem 2.5 is tight.

Example 2.8 *Consider SDP problem:*

$$\begin{array}{rl} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T \bullet X & = 1,\ \forall i < j = 1, 2, 3, 4, \\ X & \in \mathcal{S}^4_+, \end{array}$$

where $\mathbf{e}_i \in \mathcal{R}^4$ *has* 1 *at its* i*th entry and* 0 *everywhere else. The SDP the solution has rank exactly* 3*, which reaches the bound of Theorem 2.5 since it has* $m = 6$ *equations.*

To see an application of Theorem 2.5, consider a general quadratic minimization with sphere constraint

$$\begin{array}{rll} z^* \equiv & \text{minimize} & \mathbf{x}^T Q \mathbf{x} + 2\mathbf{c}^T \mathbf{x} \\ & \text{subjectto} & \|\mathbf{x}\|^2 = 1,\ \mathbf{x} \in \mathcal{R}^n, \end{array}$$

where Q is general. The problem has an SDP relaxation by let $X = \mathbf{x}\mathbf{x}^T$

$$\begin{array}{ll} \text{minimize} & Q \bullet X + 2\mathbf{c}^T \mathbf{x} \\ \text{subjectto} & I \bullet X = 1, \\ & X \succeq \mathbf{0}. \end{array}$$

But $X \succeq \mathbf{0}$ can be equivalently written as

$$\begin{pmatrix} X & \mathbf{x} \\ \mathbf{x}^T & 0 \end{pmatrix} \succeq \mathbf{0}.$$

Thus, the relaxation can be written in a standard form

$$\begin{array}{rll} z^{SDP} \equiv & \text{minimize} & \left(\begin{array}{cc} Q & \mathbf{c} \\ \mathbf{c}^T & 0 \end{array} \right) \bullet Y \\ & \text{subject to} & \left(\begin{array}{cc} I & \mathbf{0} \\ \mathbf{0}^R & 0 \end{array} \right) \bullet Y = 1, \\ & & \left(\begin{array}{cc} \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{array} \right) \bullet Y = 1, \\ & & Y \in \mathcal{S}_+^{n+1}. \end{array}$$

Note that the SDP relaxation and its dual both have interior so that the strong duality theorem holds, and it must have a rank-1 optimal SDP solution because $m = 2$. But a rank-1 optimal SDP solution would be optimal to the original quadratic minimization with sphere constraint. Thus, we must have $z^* = z^{SDP}$. Thus, the sphere constrained general quadratic minimization is a hidden convex SDP minimization, or its SDP relaxation is exact.

2.4 Approximate Low-Rank SDP Solutions

For simplicity, we again consider finding a feasible solution for (SDP), that is find $X \succeq \mathbf{0}$ that satisfies

$$A_i \bullet X = b_i \qquad \forall\ i = 1, \ldots, m, \tag{2.11}$$

where we assume that $A_1, \ldots, A_m$ are all symmetric and positive semidefinite matrices and scalars $\mathbf{b} = (b_1; \ldots; b_m) \geq \mathbf{0}$.

Instead of finding an exact low-rank SDP feasible solution, we are interested in finding an approximate low-rank feasible solution. More precisely, for a given positive integer d, we consider the problem of finding an $\hat{X} \succeq \mathbf{0}$ of rank at most d that satisfies the linear equations (2.11) *approximately*:

$$\beta \cdot b_i \leq A_i \bullet \hat{X} \leq \alpha \cdot b_i \qquad \forall\ i = 1, \ldots, m.$$

Here, $\alpha \geq 1$ and $\beta \in (0, 1]$ are called distortion factors. Clearly, the closer are both factors to 1, the better of the accuracy of $\hat{X}$.

2.4.1 Approximate low-rank theorem

We have the following low-rank approximate theorem:

Theorem 2.6 *Let $r = \max\{rank(A_i)\} \leq n$ and there be $X \succeq \mathbf{0}$ that satisfies (2.11) exactly. Then, for any integer $d \geq 1$, there exists an $\hat{X} \succeq \mathbf{0}$ with rank at most d such that*

$$\beta(m, d) \cdot b_i \leq A_i \bullet \hat{X} \leq \alpha(m, r, d) \cdot b_i \qquad \forall\ i = 1, \ldots, m,$$

where

$$\alpha(m,r,d) = \begin{cases} 1 + \dfrac{12\ln(4mr)}{d} & \text{for } 1 \le d \le 12\ln(4mr) \\ 1 + \sqrt{\dfrac{12\ln(4mr)}{d}} & \text{for } d > 12\ln(4mr) \end{cases}$$

and

$$\beta(m,d) = \begin{cases} \dfrac{1}{e(2m)^{2/d}} & \text{for } 1 \le d \le 4\ln(2m) \\ \max\left\{ \dfrac{1}{e(2m)^{2/d}},\, 1 - \sqrt{\dfrac{4\ln(2m)}{d}} \right\} & \text{for } d > 4\ln(2m) \end{cases}$$

Moreover, there exists an efficient randomized algorithm for finding such an $\hat{X}$.

Theorem 2.6 states that there exist an approximate rank-d SDP solution with respect to a bounded distortion if the system is feasible. As the allowable rank d increases, the distortions bounds become closer and closer to 1. When $d = \Omega(\ln(mr))$, then they become constants. Also, the lower distortion bound $\beta(m,d)$ is independent of r. This result contains as special cases several well-known results in the literature.

Without loss of generality, we can assume $X = I$ is such a feasible solution that satisfies (2.11) exactly. (Otherwise, as did in the last section, we can factorize X and introduce an equivalent system where I is a feasible solution.) With this assumption, we have $b_i = \mathrm{tr}(A_i) \ge 0$ for all i. Thus we would prove the following equivalent result:

Theorem 2.7 *Let* $A_1, \ldots, A_m \in \mathcal{S}^n$ *be symmetric positive semidefinite matrices and* $r = \max\{rank(A_i)\}$. *Then, for any integer* $d \ge 1$, *there exists an* $\hat{X} \succeq \mathbf{0}$ *with rank at most* d *such that*

$$\beta(m,d) \cdot tr(A_i) \le A_i \bullet \hat{X} \le \alpha(m,r,d) \cdot tr(A_i) \qquad \forall\ i = 1, \ldots, m,$$

where $\alpha(m,r,d)$ *and* $\beta(m,d)$ *are given in Theorem 2.6.*

To prove the theorem, we first introduce Markov's Inequality

Lemma 2.8 *Let* h *be any positive valued and non-decreasing function and let* ξ *be a random variable. Then, for any positive value* t, *the probability*

$$\Pr(\xi \ge t) \le \frac{\mathrm{E}[h(\xi)]}{h(t)},\ t > 0.$$

Proof.Let f_ξ be the density function of ξ. Then,

$$\begin{aligned}
\mathrm{E}[h(\xi)] &= \int_{-\infty}^{\infty} h(z) f_\xi(z) dz \\
&= \int_{-\infty}^{t} h(z) f_\xi(z) dz + \int_{t}^{\infty} h(z) f_\xi(z) dz \\
&\geq \int_{t}^{\infty} h(z) f_\xi(z) dz \quad (h \text{ is nonnegative}) \\
&\geq \int_{t}^{\infty} h(t) f_\xi(z) dz \quad (h \text{ is non-decreasing}) \\
&= h(t) \int_{t}^{\infty} f_\xi(z) dz = h(t) \mathrm{Pr}(\xi \geq t).
\end{aligned}$$

∎

2.4.2 A constructive proof

We now prove Theorem 2.7. We first apply a simple randomized procedure to construct an solution $\hat{X}$:

- Generate i.i.d. Gaussian random variables ξ_i^j with mean 0 and variance $1/d$, and define vectors $\boldsymbol{\xi}^j = (\xi_1^j; \ldots; \xi_n^j) \in \mathcal{R}^n$, where $i = 1, \ldots, n$; $j = 1, \ldots, d$.
- Return $\hat{X} = \sum_{j=1}^{d} \boldsymbol{\xi}^j \left(\boldsymbol{\xi}^j\right)^T \in \mathcal{S}_+^n$.

Clearly, the rank of random matrix $\hat{X}$ is no more than d and it is positive semidefinite with probability one.

Proposition 2.9 *Let $H \in \mathcal{S}^n$ be a symmetric positive semidefinite matrix with $r \equiv rank(H) \geq 1$. Then, for any $\beta \in (0, 1)$, we have:*

$$\mathrm{Pr}\left(H \bullet \hat{X} \leq \beta tr(H)\right) \leq \exp\left(\frac{d}{2}(1 - \beta + \ln \beta)\right) \leq \exp\left(\frac{d}{2}(1 + \ln \beta)\right)$$

Proof.Consider the spectral decomposition $H = \sum_{k=1}^{r} \lambda_k \mathbf{v}_k \mathbf{v}_k^T$, where eigenvalues in a descending order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$. Then, we have

$$H \bullet \hat{X} = \sum_{k=1}^{r} \sum_{j=1}^{d} \lambda_k \left(\mathbf{v}_k^T \boldsymbol{\xi}^j\right)^2.$$

Now, observe that $U = \left[\left(\mathbf{v}_k^T \boldsymbol{\xi}^j\right)_{k=1,\ldots,r, j=1,\ldots,d}\right] \sim \mathcal{N}(0, d^{-1} I_{rd})$, since $U_{kj} = \mathbf{v}_k^T \boldsymbol{\xi}^j$ is a Gaussian random variable with $E[\mathbf{v}_k^T \boldsymbol{\xi}^j] = 0$ and

$$\begin{aligned}
E\left[\left(\mathbf{v}_k^T \boldsymbol{\xi}^j\right)\left(\mathbf{v}_l^T \boldsymbol{\xi}^{j'}\right)\right] &= \frac{1}{d} \cdot \mathbf{v}_k^T \mathbf{v}_l \cdot \mathbf{1}_{\{j=j'\}} \\
&= \frac{1}{d} \cdot \mathbf{1}_{\{k=l, j=j'\}},
\end{aligned}$$

where $\mathbf{1}_{\{\cdot\}}$ is the indicator function either 1 or 0.

Since uncorrelated Gaussian random variables are independent, it follows that $H \bullet \hat{X}$ has the same distribution as $\sum_{k=1}^{r}\sum_{j=1}^{d}\lambda_k\tilde{\xi}_{kj}^2$, where $\tilde{\xi}_{kj}$ are i.i.d. Gaussian random variables with mean 0 and variance $1/d$. In particular, we have:

$$\begin{aligned}\Pr\left(H \bullet \hat{X} \leq \beta \mathrm{tr}(H)\right) &= \Pr\left(\sum_{k=1}^{r}\sum_{j=1}^{d}\lambda_k\tilde{\xi}_{kj}^2 \leq \beta\sum_{k=1}^{r}\lambda_k\right)\\ &= \Pr\left(\sum_{k=1}^{r}\sum_{j=1}^{d}\bar{\lambda}_k\tilde{\xi}_{kj}^2 \leq \beta\right)\end{aligned}$$

where $\bar{\lambda}_k = \lambda_k/(\lambda_1 + \cdots + \lambda_r)$ for $k = 1, \ldots, r$. Now, we compute

$$\begin{aligned}&\Pr\left(\sum_{k=1}^{r}\sum_{j=1}^{d}\bar{\lambda}_k\tilde{\xi}_{kj}^2 \leq \beta\right)\\ =\ &\Pr\left(\exp\left(-t\sum_{k=1}^{r}\sum_{j=1}^{d}\bar{\lambda}_k\tilde{\xi}_{kj}^2\right) \geq \exp(-t\beta)\right) \text{ (for all } t \geq 0)\\ \leq\ &\exp(t\beta)\cdot E\left[\exp\left(-t\sum_{k=1}^{r}\sum_{j=1}^{d}\bar{\lambda}_k\tilde{\xi}_{kj}^2\right)\right] \text{ (by Markov's inequality)}\\ =\ &\exp(t\beta)\cdot\prod_{k=1}^{r}E\left[\exp\left(-t\bar{\lambda}_k\tilde{\xi}_{11}^2\right)\right]^d \text{ (by independence)}\end{aligned}$$

Recall that for a standard Gaussian random variable ξ, we have $E[\exp\left(-t\xi^2\right)] = (1+2t)^{-1/2}$ for all $t \geq 0$. Thus, it follows that

$$\begin{aligned}\Pr\left(H \bullet \hat{X} \leq \beta \mathrm{tr}(H)\right) &\leq \exp(t\beta)\cdot\prod_{k=1}^{r}\left(1+\frac{2t\bar{\lambda}_k}{d}\right)^{-d/2}\\ &= \exp(t\beta)\cdot\exp\left(-\frac{d}{2}\sum_{k=1}^{r}\ln\left(1+\frac{2t\bar{\lambda}_k}{d}\right)\right).\end{aligned}$$

Now, note that for any fixed $t \geq 0$, the function $g_t : \mathbf{R}^r \to \mathbf{R}$ defined by $g_t(\mathbf{x}) = -(d/2)\sum_{k=1}^{r}\ln(1+2tx_k/d)$ is convex over the simplex $\sum_{k=1}^{r}x_k = 1$ and $\mathbf{x} \geq \mathbf{0}$. Hence, its maximum over a simplex is attained at a vertex, which leads to

$$\Pr\left(H \bullet \hat{X} \leq \beta \mathrm{tr}(H)\right) \leq \exp\left[t\beta - \frac{d}{2}\ln\left(1+\frac{2t}{d}\right)\right],\ \forall t > 0.$$

Furthermore, function $\exp(t\beta - (d/2)\ln(1+2t/d))$ is minimized at $t^* = d(1-\beta)/2\beta$, and $t^* > 0$ whenever $\beta \in (0,1)$, which gives the proof. ∎

Proposition 2.10 *Let $H \in \mathcal{S}^n$ be a symmetric positive semidefinite matrix with $r \equiv rank(H) \geq 1$. Then, for any $\alpha > 1$, we have:*

$$\Pr\left(H \bullet \hat{X} \geq \alpha tr(H)\right) \leq r \cdot \exp\left(\frac{d}{2}\left(1 - \alpha + \ln \alpha\right)\right)$$

Proof.We now consider factorization $H = \sum_{k=1}^{r} \mathbf{q}_k \mathbf{q}_k^T$. Thus, we have $H \bullet \hat{X} = \sum_{k=1}^{r} \sum_{j=1}^{d} \left(\mathbf{q}_k^T \boldsymbol{\xi}^j\right)^2$. Observe that $\mathbf{q}_k^T \boldsymbol{\xi}^j$ is a Gaussian random variable with zero mean and variance $\sigma_k^2 = \frac{1}{d}\|\mathbf{q}_k\|^2$. Then, we have

$$\sum_{k=1}^{r} \sigma_k^2 = \frac{1}{d} \sum_{k=1}^{r} \|\mathbf{q}_k\|^2 = \frac{1}{d} \cdot \mathrm{tr}(H).$$

It follows that:

$$\begin{array}{ll} \Pr\left(H \bullet \hat{X} \geq \alpha \mathrm{tr}(H)\right) & = \Pr\left(\sum_{k=1}^{r} \sum_{j=1}^{d} \left(\mathbf{q}_k^T \boldsymbol{\xi}^j\right)^2 \geq \alpha d \sum_{k=1}^{r} \sigma_k^2\right) \\ & \leq \sum_{k=1}^{r} \Pr\left(\sum_{j=1}^{d} \left(\mathbf{q}_k^T \boldsymbol{\xi}^j\right)^2 \geq \alpha d \sigma_k^2\right) \end{array} \tag{2.12}$$

To bound the last quantity, we first note that $E[\sum_{j=1}^{d} \left(\mathbf{q}_k^T \boldsymbol{\xi}^j\right)^2] = d \cdot \sigma_k^2$. Hence, for any $t \in [0, 1/2)$ and $k = 1, \ldots, r$, we have:

$$\begin{array}{rcl} \Pr\left(\sum_{j=1}^{d} \left(\mathbf{q}_k^T \boldsymbol{\xi}^j\right)^2 \geq \alpha d \sigma_k^2\right) & = & \Pr\left(\exp\left(t \sum_{j=1}^{d} \sigma_k^{-2} \left(\mathbf{q}_k^T \boldsymbol{\xi}^j\right)^2\right) \geq \exp(t\alpha d)\right) \\ & \leq & \exp(-t\alpha d) \cdot (1 - 2t)^{-d/2} \end{array}$$

Now, the function $\exp\left(-t\alpha d\right) \cdot (1 - 2t)^{-d/2}$ is minimized at $t^* = (\alpha - 1)/2\alpha$. Moreover, we have $t^* \in (0, 1/2)$ whenever $\alpha \in (1, \infty)$. It follows that:

$$\begin{array}{ll} \Pr\left(\sum_{j=1}^{d} \left(q_k^T \xi^j\right)^2 \geq \alpha d \sigma_k^2\right) & \leq \alpha^{d/2} \cdot \exp\left(-\frac{d(\alpha-1)}{2}\right) \\ & = \exp\left(\frac{d}{2}(1 - \alpha + \ln \alpha)\right). \end{array} \tag{2.13}$$

Upon combining (2.12) and (2.13), we obtain:

$$\Pr\left(H \bullet \hat{X} \geq \alpha \mathrm{tr}(H)\right) \leq r \cdot \exp\left(\frac{d}{2}\left(1 - \alpha + \ln \alpha\right)\right).$$

∎

From these two Propositions, we are now ready to prove the theorem. We first establish the lower bound. Let $\beta = \left(e(2m)^{2/d}\right)^{-1}$. Note that $\beta \in (0, 1)$ for all $d \geq 1$. Hence, by Proposition 2.9, we have:

$$\Pr\left(A_i \bullet \hat{X} \leq \beta \mathrm{tr}(A_i)\right) \leq \exp\left[\frac{d \ln(e\beta)}{2}\right] = \frac{1}{2m} \quad \text{for } i = 1, \ldots, m$$

which implies that:

$$\Pr\left(A_i \bullet \hat{X} \geq \frac{1}{e(2m)^{2/d}} \cdot \mathrm{tr}(A_i) \text{ for all } i = 1, \ldots, m\right) \geq \frac{1}{2} \tag{2.14}$$

On the other hand, if $d > 4\ln(2m)$, then we can obtain an alternative bound as follows. Write $\beta = 1 - \beta'$ for some $\beta' \in (0, 1)$. Using the inequality $\ln(1 - x) \leq -x - x^2/2$, which is valid for all $x \in [0, 1]$, we have:

$$1 - \beta + \ln\beta = \beta' + \ln(1 - \beta') \leq -\frac{\beta'^2}{2}.$$

Now, let $\beta' = \sqrt{\frac{4\ln(2m)}{d}}$. Since $d > 4\ln(2m)$, we have $\beta' \in (0, 1)$. It then follows from Proposition 2.9 that:

$$\Pr\left(A_i \bullet \hat{X} \leq \beta \mathrm{tr}(A_i)\right) \leq \exp\left(-\frac{d\beta'^2}{4}\right) = \frac{1}{2m} \quad \text{for } i = 1, \ldots, m$$

which in turn implies that:

$$\Pr\left(A_i \bullet \hat{X} \geq \left(1 - \sqrt{\frac{4\ln(2m)}{d}}\right) \cdot \mathrm{tr}(A_i) \text{ for all } i = 1, \ldots, m\right) \geq \frac{1}{2} \tag{2.15}$$

Upon combining (2.14) and (2.15), we obtain:

$$\Pr\left(A_i \bullet \hat{X} \geq \beta(m, d) \cdot \mathrm{tr}(A_i) \text{ for all } i = 1, \ldots, m\right) \geq \frac{1}{2} \tag{2.16}$$

where $\beta(m, d)$ is given by the theorem 2.6.

Next, we establish the upper bound. We write $\alpha = 1 + \alpha'$ for some $\alpha' > 0$. Using the inequality $\ln(1 + x) \leq x - x^2/2 + x^3/3$, which is valid for all $x > 0$, it is easy to show that:

$$1 - \alpha + \ln\alpha = -\alpha' + \ln(1 + \alpha') \leq \begin{cases} -\dfrac{\alpha'}{6} & \text{for } \alpha' \geq 1 \\ -\dfrac{\alpha'^2}{6} & \text{for } 0 < \alpha' < 1 \end{cases} \tag{2.17}$$

Let $T = \frac{12\ln(4mr)}{d}$. If $T \geq 1$, then set $\alpha' = T$; otherwise, set $\alpha' = \sqrt{T}$. In the former case, we have $\alpha' \geq 1$, and hence by Proposition 2.10 and the bound in (2.17), for $i = 1, \ldots, m$, we have:

$$\Pr\left(A_i \bullet \hat{X} \geq \alpha \mathrm{tr}(A_i)\right) \leq \mathrm{rank}(A_i) \cdot \exp\left(-\frac{d\alpha'}{12}\right) \leq \frac{1}{4m}$$

where the last inequality follows from the fact that $\mathrm{rank}(A_i) \leq r$. In the latter case, we have $\alpha' \in (0, 1)$, and a similar calculation shows that:

$$\Pr\left(A_i \bullet \hat{X} \geq \alpha \mathrm{tr}(A_i)\right) \leq \mathrm{rank}(A_i) \cdot \exp\left(-\frac{d\alpha'^2}{12}\right) \leq \frac{1}{4m}$$

for $i = 1, \ldots, m$. Hence, we conclude that:

$$\Pr\left(A_i \bullet \hat{X} \leq \alpha(m,r,d) \cdot \mathrm{tr}(A_i) \text{ for all } i = 1, \ldots, m\right) \geq 1 - \frac{1}{4} = \frac{3}{4} \qquad (2.18)$$

where $\alpha(m,r,d)$ is given by the Theorem 2.6.

Finally, upon combining (2.16) and (2.18), we conclude that:

$$Pr\left(\beta(m,d) \cdot \mathrm{tr}(A_i) \leq A_i \bullet \hat{X} \leq \alpha(m,r,d) \cdot \mathrm{tr}(A_i) \text{ for all } i = 1, \ldots, m\right)$$

is greater than $1 - \left(\frac{1}{4} + \frac{1}{2}\right) = \frac{1}{4}$, which completes the proof of Theorems 2.7 and 2.6.

As indicated by the exact low-rank Theorem 2.5, one can always reduce an n-dimensional SDP problem to a r-dimensional SDP problem where $r(r+1)/2 \leq m$. Thus, one can safely claim that $r \leq \sqrt{2m}$ in $\alpha(m,r,d)$ of Theorem 2.6.

If A_i is not positive semidefinite in system

$$A_i \bullet X = b_i \qquad \forall\ i = 1, \ldots, m,\ X \succeq \mathbf{0},$$

one cannot guarantee relative distortion bounds for $\hat{X}$. However, since

$$A_i = A_i^+ - A_i^-$$

where both A_i^+ and A_i^- are positive semidefinite. Thus, for any exact solution X, let

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

Then, we can have bounded relative distortions for $\hat{X}$ on b_i^+ and b_i^-, which lead to bounded absolute distortions on b_i.

2.5 Uniqueness of CLP Optimal Solution

In many applications, one may want to know if an optimization problem possesses a unique optimizer. Moreover, can such a uniqueness can be certified by an efficient algorithm. We give a positive answer for linear programming and semidefinite programming.

Theorem 2.11 *Consider LP and SDP.*

i) (Uniqueness Theorem for Linear Programming) An LP optimal solution $\mathbf{x}^*$ is unique if and only if the cardinality of supp($\mathbf{x}^*$) is maximal among all optimal solutions and the columns of $A_{supp(\mathbf{x}^*)}$ are linear independent, that is, the only solution for the null space, $A_{supp(\mathbf{x}^*)}\mathbf{w} = \mathbf{0}$, is $\mathbf{w} = \mathbf{0}$.

ii) (Uniqueness Theorem for Semidefinite Programming) An SDP optimal and complementary solution X^* is unique if and only if the rank of X^* is maximal among all optimal solutions and the only solution for the null space, $V^* A_i (V^*)^T \bullet W = 0$, $i = 1, ..., m$, is $W = \mathbf{0}$, where $X^* = (V^*)^T V^*$, $V^* \in \mathcal{R}^{r \times n}$ and r is the rank of X^*.

Proof.It is easy to see both conditions are necessary for LP case, since otherwise, one can find an optimal solution with a different support size. To see sufficiency, suppose there there is another optimal solution $\mathbf{y}^*$ such that $\mathbf{x}^* - \mathbf{y}^* \neq \mathbf{0}$. We must have $\text{supp}(\mathbf{y}^*) \subset \text{supp}(\mathbf{x}^*)$. Then we see

$$\mathbf{0} = A\mathbf{x}^* - A\mathbf{y}^* = A(\mathbf{x}^* - \mathbf{y}^*) = A_{\text{supp}(\mathbf{x}^*)}(\mathbf{x}^* - \mathbf{y}^*)_{\text{supp}(\mathbf{x}^*)}$$

which implies that columns of $A_{\text{supp}(\mathbf{x}^*)}$ are linearly dependent.

Note that for any optimal dual slack matrix S^*, we have $S^* \bullet (V^*)^T V^* = 0$ which implies that $S^*(V^*)^T = \mathbf{0}$. Consider any matrix $X(U) = (V^*)^T U V^*$ where $U \in \mathcal{S}^r_+$ and

$$b_i = A_i \bullet (V^*)^T U V^* = V^* A_i (V^*)^T \bullet U, \ i = 1, ..., m.$$

One can see that $X(U)$ remains an optimal SDP solution for any such $U \in \mathcal{S}^r_+$, since it makes $X(U)$ feasible and remain complementary to any optimal dual slack matrix. If one can find $W \neq \mathbf{0}$ such that

$$V^* A_i (V^*)^T \bullet W = 0, \quad i = 1, ..., m, \ \mathbf{0} \neq W \in \mathcal{S}^r.$$

Now consider $X(I + \alpha W) = (V^*)^T (I + \alpha \cdot W) V^*$, and one can choose $\alpha \neq 0$ such that $X(I + \alpha W) \succeq \mathbf{0}$ is another optimal solution. Thus, both conditions are necessary.

To see sufficiency, suppose there there is another optimal solution Y^* such that $X^* - Y^* \neq \mathbf{0}$. We must have $Y^* = (V^*)^T U V^*$ for some $U \in \mathcal{S}^r_+$, since X^* has the maximal rank. Then we see $I - U \neq bz$ but

$$V^* A_i (V^*)^T \bullet (I - U) = 0, \quad i = 1, ..., m,$$

which is a contradiction. ∎

Example 2.9 *Consider the LP and SDP examples presented earlier.*

1) The LP primal and dual example:

$$\begin{array}{rl} \text{minimize} & \begin{pmatrix} 2 & 1 & 1 \end{pmatrix} \mathbf{x} \\ \text{subject to} & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \mathbf{x} = 1, \\ & \mathbf{x} \geq \mathbf{0} \end{array} \qquad \begin{array}{rl} \text{maximize} & y \\ \text{subject to} & y \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \mathbf{s} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \\ & \mathbf{s} \geq \mathbf{0}, \end{array}$$

An optimal solution pair

$$\mathbf{x}^* = \begin{pmatrix} 0 \\ 0.5 \\ 0.5 \end{pmatrix} \quad \textit{and} \quad \mathbf{s}^* = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Clearly $\mathbf{x}^*$ *is a maximal cardinality optimal solution, since an dual optimal solution has cardinality* 1 *and* $n = 3$*. But the null space equation*

$$\mathbf{0} = A_{2,3}\mathbf{w} = (1 \quad 1)\mathbf{w} = w_1 + w_2$$

has a nonzero solution $\mathbf{w}$*, thus the linear program does not have a unique optimal solution.*

2) The SDP primal and dual example:

$$\begin{array}{ll} \text{minimize} & \begin{pmatrix} 2 & .5 \\ .5 & 1 \end{pmatrix} \bullet X \\ \text{subject to} & \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix} \bullet X = 1, \\ & X \succeq \mathbf{0}. \end{array} \qquad \begin{array}{ll} \text{maximize} & y \\ \text{subject to} & y \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix} + S = \begin{pmatrix} 2 & .5 \\ .5 & 1 \end{pmatrix}, \\ & S \succeq \mathbf{0}. \end{array}$$

An optimal solution pair

$$X^* = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad \textit{and} \quad S^* = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Clearly X^* *has rank* 1 *and it is maximal, since an dual optimal solution has rank* 1 *and* $n = 2$*. One can see* $V^* = (0 \quad 1)$ *is a factorization of* $X^* = (V^*)^T V^*$*. Then,* $V^* A_1 (V^*)^T = 1$ *so that the only solution to the null space equation* $V^* A_1 (V^*)^T \bullet W = \mathbf{0}$ *is* $W = \mathbf{0}$*. (Note that here* $W \in \mathcal{S}^1$ *so that it is a scalar.) Thus, the semidefinite program has a unique optimal solution.*

Later, we would show that polynomial-time interior-point algorithms are capable to compute a maximal cardinality or maximal rank optimal solution for (LP) or (SDP). At such a solution, we can check the null space condition in polynomial time. Thus, to certify the uniqueness of an linear program or semidefinite program can be done in polynomial time.

From the proof of Theorem 2.11, we can also derive a corollary

Corollary 2.12 *Consider LP and SDP.*

i) If all optimal solutions of an linear program have same number of nonzero entries, then the linear program has a unique optimal solution.

ii) If all optimal solutions of an semidefinite program have same rank, then the semidefinite program has a unique optimal solution.

2.6 Notes

Farkas' lemma for conic linear constraints are closely linked to convex analysis (i.e, Rockeafellar [266]) and the CLP duality theorems commented next, also see Barvinok [37].

The SDP duality theory was studied by Barvinok [37], Nesterov and Nemirovskii [243], Ramana [261], Ramana e al. [262], etc. The SDP example with a duality gap was constructed by R. Freund (private communication).

Complementarity and rank. The exact low-rank theorem described here is due to Pataki [254], also see Barvinok [36] and Alfakih and Wolkowicz [6]. The approximate low-rank theorem is due to So et al. [282] which can be seen as a generalization of the Johnson and Lindenstrauss theorem [177].

2.7 Exercises

2.1 *Find the optimal solution pair of Example 2.1.*

2.2 *Prove the second statement of Theorem 2.1.*

2.3 *Verify the weak duality theorem of the three CLP instances in Example 2.2.*

2.4 *This problem is related to the graph realization problem covered in class. If there is no anchor in the Euclidean-distance graph, then the problem is called anchor-free realization. It is clear that the realization cannot be unique since the configuration may be translated, rotated, or reflected, while the distances are being preserved. To remove the translation factor, one can add an objective function to minimize the norm of the solution in the problem formulation:*

$$\begin{array}{ll} \textit{minimize} & \sum_{j=1}^{n} \|\mathbf{x}_j\|^2 \\ \textit{s.t.} & \|\mathbf{x}_i - \mathbf{x}_j\|^2 = d_{ij}^2, \quad \forall\ (i,j) \in N_x, \\ & \mathbf{x}_i \in R^d. \end{array}$$

a) Show that what this minimization does is to translate the center of gravity of the point set to the origin, that is, if $\bar{\mathbf{x}}_j$, $j = 1, ..., n$, is the solution of the problem, then

$$\sum_{j=1}^{n} \bar{\mathbf{x}}_j = \mathbf{0}.$$

b) Write an SDP relazation to the minimization problem, and write the dual of the SDP relaxation.

2.5 *Let $S \succeq \mathbf{0}$, $X \succeq \mathbf{0}$ and $SX = \mathbf{0}$. Furthermore, let $X = V^T V$. Then $VS = SV^T = \mathbf{0}$.*

2.6 *Using the SDP rank reduction theorem to show that the optimality conditions for the global minimizer $\mathbf{y}^*$ of (BD) in Section 1.5.5:*

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

and

$$(Q + \mu^* I) \succeq \mathbf{0};$$

are necessary and sufficient.

2.7 *Given any matrix $A \in R^{n \times m}$, using the SDP rank reduction theorem to show that*

$$\begin{array}{ll} \textit{minimize} & \mathbf{x}^T A \mathbf{y} \\ \textit{s.t.} & \|\mathbf{x}\|^2 = 1, \\ & \|\mathbf{y}\|^2 = 1, \end{array}$$

is a hidden SDP problem. Note that the minimal value of the problem is the singular value of A.

2.8 *Consider the SDP problem*

$$\begin{array}{ll} \textit{minimize} & C \bullet X \\ \textit{s.t.} & A_i \bullet X = b_i, \quad i = 1, ..., m, \\ & Q_j \bullet X = 0, \quad j = 1, ..., q, \\ & X \succeq \mathbf{0}, \end{array}$$

where coefficient matrices Q_j, $j = 1, \ldots, q$, are positive semidefinite.

a) Suppose that there is an optimal solution X^* with zero duality gap, show that there must be an optimal solution matrix with its rank r satisfying $r(r+1)/2 \leq m$. (Note that the bound is independent of q.)

b) Using the above result to show that the quadratic problem

$$\begin{array}{ll} \textit{minimize} & \mathbf{x}^T Q \mathbf{x} + 2\mathbf{c}^T \mathbf{x} \\ \textit{s.t.} & A\mathbf{x} = \mathbf{0}, \\ & \|\mathbf{x}\|^2 = 1 \end{array}$$

is a hidden SDP problem, where given Q is an n-dimensional symmetric matrix and A is an $m \times n$ matrix with $m < n$.

2.9 *Now you have down loaded SEDUMI1.05, DSDP5.8, and/or CVX, and use them to solve the following SDP problems.*

a) The SDP problem

$$\begin{array}{ll} \textit{minimize} & C \bullet X \\ \textit{s.t.} & \mathbf{e}_i \mathbf{e}_i^T \bullet X = 1, \ i = 1, 2, 3, \\ & (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T \bullet X = 1, \ 1 \leq i < j \leq 3, \\ & X \succeq \mathbf{0} \in \mathcal{S}^3, \end{array}$$

and check the solution matrix rank when (i) $C = I$, (ii) $C = -I$, and (iii) $C = -e_3 e_3^T$.

b) The SDP problem

$$\begin{array}{ll} \textit{minimize} & C \bullet X \\ \textit{s.t.} & \mathbf{e}_1 \mathbf{e}_1^T \bullet X = 1, \\ & (\mathbf{e}_i - \mathbf{e}_{i+1})(\mathbf{e}_i - \mathbf{e}_{i+1})^T \bullet X = 1, \ i = 1, 2, \\ & X \succeq \mathbf{0} \in \mathcal{S}^3, \end{array}$$

and check the solution rank when (i) $C = I$, (ii) $C = -I$, and (iii) $C = -e_3 e_3^T$.

Here e_i is the vector of all zeros except 1 at the ith position.

Chapter 3

Interior-Point Algorithms

Although conic linear programs are much more general than classical linear programs, they are not much harder to solve. It has turned out that most interior-point methods for LP are extentable to solving CLP. As in LP, these algorithms possess polynomial-time worst-case complexity under certain computation models. They also perform well in practice. We will describe such extensions in this chapter.

3.1 Central Path and Path-Following

To develop efficient interior-point algorithms, the key is to find a suitable barrier or potential function, and the central-ray in $\mathcal{E}$ most interior to the cone. There is a general theory on selection of barrier functions for (CLP) or (CLD), depending on the convex cone involved. In this chapter, we consider self-dual cones ($K = K^*$) only and present their barrier functions and central rays.

Definition 3.1 *A differentiable function $B(X)$ is called barrier function for a closed convex cone K if for the sequence $\{X^k \in \overset{\circ}{K}\}$, $k = 1, \ldots,$*

$$X^k \to \partial K \quad \Rightarrow \quad B(X^k) \to \infty,$$

where ∂K represents the boundary of K. $X \bullet (-\nabla B(X))$ is called the barrier-coefficient of $B(X)$, denoted by ν; and a point in $\overset{\circ}{K}$ is called the central-ray point if it is a fixed point of

$$X = -\nabla B(X),$$

denoted by I^c.

No that for LP and SOCP, $X \in \mathcal{R}^n$ is a vector and we will often write it as $\mathbf{x}$. For SDP, $X \in \mathcal{S}^n$, that is, it is a n-dimensional symmetric matrix.

3.1.1 Logarithmic barrier function for convex cones

We present logarithmic barrier functions for several popular cones.

Example 3.1 *The following are logarithmic barrier functions, and their gradient vectors, Hessian matrices, central-ray points and barrier-coefficients for the following convex cones.*

- *The n-dimensional non-negative orthant $\mathcal{R}^n_+$:*

$$B(\mathbf{x}) = -\sum_{j=1}^{n} \ln(x_j),\ \nabla B(\mathbf{x}) = -\Delta(\mathbf{x})^{-1}\mathbf{e},\ \nabla^2 B(\mathbf{x}) = \Delta(\mathbf{x})^{-2} \in \mathcal{S}^n.$$

  The central-ray point of $B(\mathbf{x})$ is $\mathbf{e}$, the vector of all ones, and the barrier-coefficient is

$$\mathbf{x} \bullet (-\nabla B(\mathbf{x})) = \mathbf{x} \bullet \Delta(\mathbf{x})^{-1}\mathbf{e} = n$$

- *The n-dimensional semidefinite cone $\mathcal{S}^n_+$:*

$$B(X) = -\ln\det(X),\ \nabla B(X) = -X^{-1},$$

$$\nabla^2 B(X) = \{\partial^2 B(X)/\partial X_{ij}\partial X_{kl} = X^{-1}_{ik} X^{-1} jl\} = X^{-1} \otimes X^{-1} \in \mathcal{S}^{n^2},$$

  where $\otimes$ stands for matrix Kronecker product. The central-ray point of $B(X)$ is I, the identity matrix, and the barrier-coefficient is

$$X \bullet (-\nabla B(X)) = X \bullet X^{-1} = n$$

- *The n-dimensional second-order cone $\{\mathbf{x} :\ x_1 \geq \|\mathbf{x}_{-1}\|\}$:*

$$B(\mathbf{x}) = -\frac{1}{2}\ln(x_1^2 - \|\mathbf{x}_{-1}\|^2), \quad \nabla B(\mathbf{x}) = \frac{1}{\delta(\mathbf{x})^2}\begin{pmatrix} -x_1 \\ \mathbf{x}_{-1} \end{pmatrix},$$

$$\nabla^2 B(\mathbf{x}) = \frac{1}{\delta(\mathbf{x})^2}\begin{pmatrix} -1 & 0 \\ 0 & I \end{pmatrix} + \frac{2}{\delta(\mathbf{x})^4}\begin{pmatrix} x_1 \\ -\mathbf{x}_{-1} \end{pmatrix}\begin{pmatrix} x_1 \\ -\mathbf{x}_{-1} \end{pmatrix}^T,$$

  where $\delta(\mathbf{x}) = \sqrt{x_1^2 - \|\mathbf{x}_{-1}\|^2}$. The central-ray point of $B(\mathbf{x})$ is $\mathbf{e}_1$, the unit vector with 1 as its first element and 0 everywhere else, and the barrier-coefficient is $B(\mathbf{x})$ is

$$\mathbf{x} \bullet (-\nabla B(\mathbf{x})) = \mathbf{x} \bullet \frac{-1}{\delta(\mathbf{x})^2}\begin{pmatrix} -x_1 \\ \mathbf{x}_{-1} \end{pmatrix} = 1.$$

- *The mixed cone $K = K_1 \oplus K_2$, that is, $X = [X_1; X_2]$ where $X_1 \in K_1$ and $X_2 \in K_2$:*

$$B(X) = B_1(X_1) + B_2(X_2)$$

  where $B_1(\cdot)$ and $B_2(\cdot)$ are barrier functions for K_1 and K_2, respectively. The barrier-coefficient is $\nu_1 + \nu_2$, where ν_1 and ν_2 are barrier-coefficients of $B_1(\cdot)$ and $B_2(\cdot)$.

Lemma 3.1 *The following properties hold for the logarithmic barrier functions.*

i) They are strongly convex functions, that is, the Hessian matrix of each is positive definite in the interior of K.

ii) For any scaler $|d| < 1$,

$$-d \leq -\ln(1+d) \leq -d + \frac{d^2}{2(1-|d|)}.$$

iii) For any $X \in \overset{\circ}{K}$ and $D \in \mathcal{E}$, there is a constant γ such that when $\gamma \|D\|_{\nabla^2 B(X)} < 1$,

$$\nabla B(X) \bullet D \leq B(X+D) - B(X) \leq \nabla B(X) \bullet D + \frac{\gamma^2 \|D\|^2_{\nabla^2 B(X)}}{2(1-\gamma\|D\|_{\nabla^2 B(X)})};$$

and $\gamma = 1$ for LP and SDP and $\gamma = \sqrt{2}$ for SOCP.

Proof. To prove i), we can show that the Hessian matrix is positive definite for each of LP, SDP, and SOCP barrier functions. This fact is easy to see for LP and SDP, so that we give a detailed proof for SOCP as long as $\delta(\mathbf{x})^2 > 0$. Without loss of generality, let $x_1 = 1$. Then, for any $\mathbf{y} \in \mathcal{R}^n$,

$$\begin{aligned}
\mathbf{y}^T \nabla^2 B(\mathbf{x}) \mathbf{y} &= \frac{1}{\delta(\mathbf{x})^2} \mathbf{y}^T \begin{pmatrix} -1 & 0 \\ 0 & I \end{pmatrix} \mathbf{y} + \frac{2}{\delta(\mathbf{x})^4} \mathbf{y}^T \begin{pmatrix} 1 \\ -\mathbf{x}_{-1} \end{pmatrix} \begin{pmatrix} 1 \\ -\mathbf{x}_{-1} \end{pmatrix}^T \mathbf{y} \\
&= \frac{1}{\delta(\mathbf{x})^2} (\|\mathbf{y}_{-1}\|^2 - y_1^2) + \frac{2}{\delta(\mathbf{x})^4} (y_1 - \mathbf{x}_{-1}^T \mathbf{y}_{-1})^2.
\end{aligned}$$

Now consider

$$\begin{aligned}
& \delta(\mathbf{x})^2 (\|\mathbf{y}_{-1}\|^2 - y_1^2) + 2(y_1 - \mathbf{x}_{-1}^T \mathbf{y}_{-1})^2 \\
= \; & (1 - \|\mathbf{x}_{-1}\|^2)(\|\mathbf{y}_{-1}\|^2 - y_1^2) + 2(y_1 - \mathbf{x}_{-1}^T \mathbf{y}_{-1})^2 \\
= \; & (1 - \|\mathbf{x}_{-1}\|^2)\|\mathbf{y}_{-1}\|^2 + y_1^2(1 + \|\mathbf{x}_{-1}\|^2) - 4y_1(\mathbf{x}_{-1}^T \mathbf{y}_{-1}) + 2(\mathbf{x}_{-1}^T \mathbf{y}_{-1})^2.
\end{aligned}$$

The last term is minimized when $y_1 = 2(\mathbf{x}_{-1}^T \mathbf{y}_{-1})/(1 + \|\mathbf{x}_{-1}\|^2)$, so that

$$\begin{aligned}
& (1 - \|\mathbf{x}_{-1}\|^2)\|\mathbf{y}_{-1}\|^2 + y_1^2(1 + \|\mathbf{x}_{-1}\|^2) - 4y_1(\mathbf{x}_{-1}^T \mathbf{y}_{-1}) + 2(\mathbf{x}_{-1}^T \mathbf{y}_{-1})^2 \\
\geq \; & (1 - \|\mathbf{x}_{-1}\|^2) \left(\|\mathbf{y}_{-1}\|^2 - \frac{2(\mathbf{x}_{-1}^T \mathbf{y}_{-1})^2}{1 + \|\mathbf{x}_{-1}\|^2} \right) \\
\geq \; & (1 - \|\mathbf{x}_{-1}\|^2) \left(\|\mathbf{y}_{-1}\|^2 - \frac{2\|\mathbf{x}_{-1}\|^2 \|\mathbf{y}_{-1}\|^2}{1 + \|\mathbf{x}_{-1}\|^2} \right) \\
= \; & (1 - \|\mathbf{x}_{-1}\|^2)^2 \frac{\|\mathbf{y}_{-1}\|^2}{1 + \|\mathbf{x}_{-1}\|^2}.
\end{aligned}$$

This quantity if strictly positive if $\mathbf{y}_{-1} \neq \mathbf{0}$. On the other hand, if $\mathbf{y}_{-1} = \mathbf{0}$ and $y_1 \neq 0$, $\mathbf{y}^T \nabla^2 B(\mathbf{x}) \mathbf{y}$ is strictly positive as well.

To prove ii), note that $-\ln(1+d)$ is a convex function in d so that

$$-\ln(1+d) \geq (-\ln(1+d))'|_{d=0} \cdot d = \frac{-1}{1+d}|_{d=0} \cdot d = -d.$$

To prove the right site, we use the Taylor expansion

$$\begin{aligned}
-\ln(1+d) &= -d + \frac{d^2}{2} - \frac{d^3}{3} + \frac{d^4}{4} - \dots \\
&\leq -d + \frac{|d|^2}{2} + \frac{|d|^3}{3} + \frac{|d|^4}{4} + \dots \\
&\leq -d + \frac{|d|^2}{2} + \frac{|d|^3}{2} + \frac{|d|^4}{2} + \dots \\
&= -d + \frac{|d|^2}{2}(1 + |d| + |d|^2 + \dots) \\
&= -d + \frac{|d|^2}{2(1-|d|)}
\end{aligned}$$

We now prove iii), again, the left site is due to the convexity of $B(X)$. To prove the right site, we consider LP, SDP, and SOCP separately. For LP, first note that

$$\nabla B(\mathbf{x}) \bullet \mathbf{d} = -\sum_j (d_j/x_j) \quad \text{and} \quad \|\mathbf{d}\|^2_{\nabla^2 B(\mathbf{x})} = \sum_j |d_j/x_j|^2.$$

Then, using ii) we have

$$\begin{aligned}
B(\mathbf{x}+\mathbf{d}) - B(\mathbf{x}) &= \sum_j -\ln(1 + d_j/x_j) \\
&\leq \sum_j \left(-(d_j/x_j) + \frac{|d_j/x_j|^2}{2(1-|d_j/x_j|)} \right) \\
&\leq \sum_j \left(-(d_j/x_j) + \frac{|d_j/x_j|^2}{2(1-(\sum_j |d_j/x_j|^2)^{1/2})} \right) \\
&= \sum_j \left((-d_j/x_j) + \sum_j \frac{|d_j/x_j|^2}{2(1-(\sum_j |d_j/x_j|^2)^{1/2})} \right) \\
&= \sum_j \left((-d_j/x_j) + \frac{1}{2(1-(\sum_j |d_j/x_j|^2)^{1/2})} \sum_j |d_j/x_j|^2 \right) \\
&= \nabla B(\mathbf{x}) \bullet \mathbf{d} + \frac{\|\mathbf{d}\|^2_{\nabla^2 B(\mathbf{x})}}{2(1-\|\mathbf{d}\|_{\nabla^2 B(\mathbf{x})})}.
\end{aligned}$$

For SDP, first note that

$$\|D\|^2_{\nabla^2 B(X)} = \|X^{-1/2} D X^{-1/2}\|^2 = \sum_j (\lambda(X^{-1/2} D X^{-1/2})_j)^2$$

where $\lambda(X^{-1/2}DX^{-1/2})_j$ is the jth eigenvalue of symmetric matrix $(X^{-1/2}DX^{-1/2})$. Thus,

$$\begin{aligned} B(X+D)-B(X) &= -\ln\det(X+D)+\ln\det(X) \\ &= -\ln\det(X^{-1/2}(X+D)X^{-1/2}) \\ &= -\ln\det(I+X^{-1/2}DX^{-1/2}). \end{aligned}$$

Then, the proof simply follows the proof for the LP case if we treat the eigenvalues of $X^{-1/2}DX^{-1/2}$ as the elements of $\mathbf{d}$ in LP. For both LP and SDP, the constant γ in iii) is 1.

For SOCP, first note that

$$\begin{aligned} \|\mathbf{d}\|^2_{\nabla^2 B(\mathbf{x})} &= \mathbf{d}^T\nabla^2 B(\mathbf{x})\mathbf{d} \\ &= \tfrac{1}{\delta(\mathbf{x})^2}(\|\mathbf{d}_{-1}\|^2-d_1^2)+\tfrac{2}{\delta(\mathbf{x})^4}(x_1d_1-\mathbf{x}_{-1}^T\mathbf{d}_{-1})^2. \end{aligned} \tag{3.1}$$

Without loss of generality, again we assume $x_1=1$. Then,

$$\begin{aligned} & B(\mathbf{x}+\mathbf{d})-B(\mathbf{x}) \\ = & -\frac{1}{2}\ln((1+d_1)^2-\|\mathbf{x}_{-1}+\mathbf{d}_{-1}\|^2)+\frac{1}{2}\ln(\delta(\mathbf{x})^2) \\ = & \frac{-1}{2}\left(\ln\left(1-\|\mathbf{x}_{-1}\|^2+2(d_1-\mathbf{x}_{-1}^T\mathbf{d}_{-1})+d_1^2-\|\mathbf{d}_{-1}\|^2\right)-\ln(\delta(\mathbf{x})^2)\right) \\ = & \frac{-1}{2}\left(\ln\left(\delta(\mathbf{x})^2+2(d_1-\mathbf{x}_{-1}^T\mathbf{d}_{-1})+d_1^2-\|\mathbf{d}_{-1}\|^2\right)-\ln(\delta(\mathbf{x})^2)\right) \\ = & \frac{-1}{2}\ln\left(1+\frac{1}{\delta(\mathbf{x})^2}\left(2(d_1-\mathbf{x}_{-1}^T\mathbf{d}_{-1})+d_1^2-\|\mathbf{d}_{-1}\|^2\right)\right) \\ = & \frac{-1}{2}\ln\left(\left(1+\frac{1}{\delta(\mathbf{x})^2}(d_1-\mathbf{x}_{-1}^T\mathbf{d}_{-1})\right)^2\right. \\ & \qquad \left.-\frac{1}{\delta(\mathbf{x})^2}\left(\frac{(d_1-\mathbf{x}_{-1}^T\mathbf{d}_{-1})^2}{\delta(\mathbf{x})^2}+\|\mathbf{d}_{-1}\|^2-d_1^2\right)\right). \end{aligned}$$

Similar to the proof of i) for SOCP, the quantity $\frac{(d_1-\mathbf{x}_{-1}^T\mathbf{d}_{-1})^2}{\delta(\mathbf{x})^2}+\|\mathbf{d}_{-1}\|^2-d_1^2$ is non-negative so that we can define

$$\eta(\mathbf{x},\mathbf{d})^2=\frac{(d_1-\mathbf{x}_{-1}^T\mathbf{d}_{-1})^2}{\delta(\mathbf{x})^2}+\|\mathbf{d}_{-1}\|^2-d_1^2.$$

Thus,

$$\begin{aligned}
& B(\mathbf{x}+\mathbf{d}) - B(\mathbf{x}) \\
\leq\ & \frac{-1}{2}\ln\left((1+\frac{1}{\delta(\mathbf{x})^2}(d_1 - \mathbf{x}_{-1}^T\mathbf{d}_{-1}))^2 - \frac{\eta(\mathbf{x},\mathbf{d})^2}{\delta(\mathbf{x})^2}\right) \\
=\ & \frac{-1}{2}\left(\ln\left(1+\frac{1}{\delta(\mathbf{x})^2}(d_1 - \mathbf{x}_{-1}^T\mathbf{d}_{-1}) + \frac{\eta(\mathbf{x},\mathbf{d})}{\delta(\mathbf{x})}\right)\right. \\
& \qquad \left. + \ln\left(1+\frac{1}{\delta(\mathbf{x})^2}(d_1 - \mathbf{x}_{-1}^T\mathbf{d}_{-1}) - \frac{\eta(\mathbf{x},\mathbf{d})}{\delta(\mathbf{x})}\right)\right) \\
=\ & \frac{1}{2}\left(-\ln\left(1+\frac{1}{\delta(\mathbf{x})^2}(d_1 - \mathbf{x}_{-1}^T\mathbf{d}_{-1}) + \frac{\eta(\mathbf{x},\mathbf{d})}{\delta(\mathbf{x})}\right)\right. \\
& \qquad \left. - \ln\left(1+\frac{1}{\delta(\mathbf{x})^2}(d_1 - \mathbf{x}_{-1}^T\mathbf{d}_{-1}) - \frac{\eta(\mathbf{x},\mathbf{d})}{\delta(\mathbf{x})}\right)\right).
\end{aligned}$$

Let

$$d^+ = \frac{1}{\delta(\mathbf{x})^2}(d_1 - \mathbf{x}_{-1}^T\mathbf{d}_{-1}) + \frac{\eta(\mathbf{x},\mathbf{d})}{\delta(\mathbf{x})}$$

and

$$d^- = \frac{1}{\delta(\mathbf{x})^2}(d_1 - \mathbf{x}_{-1}^T\mathbf{d}_{-1}) - \frac{\eta(\mathbf{x},\mathbf{d})}{\delta(\mathbf{x})}.$$

Then, we have

$$|d^+| \leq \frac{1}{\delta(\mathbf{x})^2}|d_1 - \mathbf{x}_{-1}^T\mathbf{d}_{-1}| + \frac{\eta(\mathbf{x},\mathbf{d})}{\delta(\mathbf{x})}$$

so that

$$\begin{aligned}
|d^+|^2 &\leq 2\left(\frac{1}{\delta(\mathbf{x})^4}(d_1 - \mathbf{x}_{-1}^T\mathbf{d}_{-1})^2 + \frac{\eta(\mathbf{x},\mathbf{d})^2}{\delta(\mathbf{x})^2}\right) \\
&= 2\left(\frac{1}{\delta(\mathbf{x})^2}(\|\mathbf{d}_{-1}\|^2 - d_1^2) + \frac{2}{\delta(\mathbf{x})^4}(d_1 - \mathbf{x}_{-1}^T\mathbf{d}_{-1})^2\right) \\
&= 2\|\mathbf{d}\|^2_{\nabla^2 B(\mathbf{x})}
\end{aligned}$$

Similarly, $|d^-|^2 \leq 2\|\mathbf{d}\|^2_{\nabla^2 B(\mathbf{x})}$. Let $\|\mathbf{d}\|_{\nabla^2 B(\mathbf{x})} < 1/\sqrt{2}$ and apply ii). Then

$$\begin{aligned}
B(\mathbf{x}+\mathbf{d}) - B(\mathbf{x}) &\leq \frac{1}{2}(-d^+ - d^-) + \frac{(d^+)^2}{4(1-|d^+|)} + \frac{(d^-)^2}{4(1-|d^-|)} \\
&\leq \nabla B(\mathbf{x}) \bullet \mathbf{d} + \frac{2\|\mathbf{d}\|^2_{\nabla^2 B(\mathbf{x})}}{2(1-\sqrt{2}\|\mathbf{d}\|_{\nabla^2 B(\mathbf{x})})},
\end{aligned}$$

which completes the proof of iii) with $\gamma = \sqrt{2}$. ∎

3.1.2 The central path

Similar to LP, for a given parameter $\mu > 0$ we consider (CLP) with the barrier function added in the objective:

$$\begin{array}{lll} (CLPB) & \text{minimize} & C \bullet X + \mu B(X) \\ & \text{subjectto} & \mathcal{A}X = \mathbf{b}, \\ & & X \in K; \end{array} \tag{3.2}$$

or (CLD) with the barrier function added in the objective:

$$\begin{array}{lll} (CLDB) & \text{maximize} & \mathbf{b}^T\mathbf{y} - \mu B(S) \\ & \text{subjectto} & \mathcal{A}^T\mathbf{y} + S = C, \\ & & S \in K. \end{array} \tag{3.3}$$

Proposition 3.2 *Let both (CLP) and (CLD) have interior feasible solutions. Then, for any given $0 < \mu < \infty$, the optimizers of (CLPB) and (CLDB) exist and they are unique and in the interior of cone K, respectively.*

In fact, for given μ, the optimizers of (CLPB) have necessary and sufficient conditions:

$$\begin{array}{rcl} C + \mu\nabla B(X) - \mathcal{A}^T\mathbf{y} & = & \mathbf{0} \\ \mathcal{A}X & = & \mathbf{b} \end{array}$$

Let $S = C - \mathcal{A}^T\mathbf{y}$. Then the conditions become

$$\begin{array}{rcl} S + \mu\nabla B(X) & = & \mathbf{0} \\ \mathcal{A}X & = & \mathbf{b} \\ -\mathcal{A}^T\mathbf{y} - S & = & -C \end{array} \tag{3.4}$$

From the calculation of $\nabla B(X)$ and $S = -\mu\nabla B(X)$, we see $S \in \overset{\circ}{K}$.

Similarly, for given μ, the optimizers of (CLDB) have necessary and sufficient conditions:

$$\begin{array}{rcl} X + \mu\nabla B(S) & = & \mathbf{0} \\ \mathcal{A}X & = & \mathbf{b} \\ -\mathcal{A}^T\mathbf{y} - S & = & -C \end{array} \tag{3.5}$$

From the calculation of $\nabla B(S)$ and $X = -\mu\nabla B(S)$, we see $X \in \overset{\circ}{K}$.

A further analysis tells that conditions (3.4) and (3.5) are equivalent: X in either set of conditions is the optimizer of (CLPB) and $(\mathbf{y}, S)$ either set of conditions is the optimizer of (CLDB). Thus, we need only to look at one of the two. This is easy to see for LP and SDP, since they can be written respectively in a symmetric form:

$$\left.\begin{array}{rcl} \mathbf{x}. \cdot \mathbf{s} & = & \mu\mathbf{e} \\ A\mathbf{x} & = & \mathbf{b} \\ -A^T\mathbf{y} - \mathbf{s} & = & -\mathbf{c} \end{array}\right\} \quad \text{for LP, where } \mu = \frac{\mathbf{x}^T\mathbf{s}}{n}, \tag{3.6}$$

and

$$\left.\begin{array}{rcl} XS & = & \mu I \\ \mathcal{A}X & = & \mathbf{b} \\ -\mathcal{A}^T\mathbf{y} - S & = & -C \end{array}\right\} \quad \text{for SDP, where } \mu = \frac{X \bullet S}{n}. \tag{3.7}$$

For SOCP, let us look at each of the two conditions (3.4) and (3.5). From Example 3.1, we have

$$\begin{array}{rcl} \mathbf{s} + \mu \frac{1}{\delta(\mathbf{x})^2} \begin{pmatrix} -x_1 \\ \mathbf{x}_{-1} \end{pmatrix} & = & \mathbf{0} \\ A\mathbf{x} & = & \mathbf{b} \\ -A^T\mathbf{y} - \mathbf{s} & = & -\mathbf{c} \end{array} \tag{3.8}$$

and

$$\begin{array}{rcl} \mathbf{x} + \mu \frac{1}{\delta(\mathbf{s})^2} \begin{pmatrix} -s_1 \\ \mathbf{s}_{-1} \end{pmatrix} & = & \mathbf{0} \\ A\mathbf{x} & = & \mathbf{b} \\ -A^T\mathbf{y} - \mathbf{s} & = & -\mathbf{c} \end{array} \tag{3.9}$$

We like to show that one implies the other. Le us start from (3.8):

$$s_1 = \frac{\mu}{\delta(\mathbf{x})^2} x_1 \text{ and } \mathbf{s}_{-1} = -\frac{\mu}{\delta(\mathbf{x})^2}\mathbf{x}_{-1}.$$

Then,

$$s_1^2 - \|\mathbf{s}_{-1}\|^2 = \frac{\mu^2}{\delta(\mathbf{x})^4}(x_1^2 - \|\mathbf{x}_{-1}\|^2) = \frac{\mu^2}{\delta(\mathbf{x})^2} > 0,$$

which implies $\mathbf{s} \in \overset{\circ}{K}$. Moreover, we have

$$\mu = \delta(\mathbf{x})\delta(\mathbf{s}) = \mathbf{x}^T\mathbf{s}.$$

Thus, the first equation can be written as

$$\begin{pmatrix} s_1 \\ \mathbf{s}_{-1} \end{pmatrix} + \frac{\delta(\mathbf{s})}{\delta(\mathbf{x})} \begin{pmatrix} -x_1 \\ \mathbf{x}_{-1} \end{pmatrix} = \mathbf{0},$$

or

$$\frac{\delta(\mathbf{s})}{\delta(\mathbf{x})} \begin{pmatrix} s_1 \\ \mathbf{s}_{-1} \end{pmatrix} + \begin{pmatrix} -x_1 \\ \mathbf{x}_{-1} \end{pmatrix} = \mathbf{0},$$

or in symmetric form

$$\begin{array}{rcl} \delta(\mathbf{x}) \begin{pmatrix} -s_1 \\ \mathbf{s}_{-1} \end{pmatrix} + \delta(\mathbf{s}) \begin{pmatrix} x_1 \\ \mathbf{x}_{-1} \end{pmatrix} & = & \mathbf{0}, \\ A\mathbf{x} & = & \mathbf{b} \\ -A^T\mathbf{y} - \mathbf{s} & = & -\mathbf{c} \end{array} \tag{3.10}$$

which, from $\delta(\mathbf{x})\delta(\mathbf{s}) = \mu$, is exactly the first equation of 3.9.

The optimizers, denoted by $(\mathbf{x}(\mu), \mathbf{y}(\mu), \mathbf{s}(\mu))$ for (3.6) and (3.10), or $(X(\mu), \mathbf{y}(\mu), S(\mu))$ for (3.7), form the central path of (CLP) and (CLD) when mu varies in $(0, \infty)$. When $\mu \to 0$, the path tends to the optimal solution set.

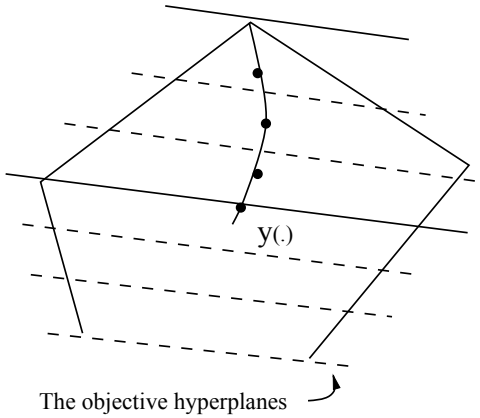


Figure 3.1: The central path of $\mathbf{y}(\mu)$ in a dual feasible region.

Theorem 3.3 *Let $(X(\mu), \mathbf{y}(\mu), S(\mu))$ be on the central path.*

i) *The central path point $(X(\mu), S(\mu))$ is bounded for $0 < \mu \leq \mu^0$ and any given $0 < \mu^0 < \infty$.*

ii) *For $0 < \mu' < \mu$,*

$$c^T X(\mu') < c^T X(\mu) \quad \textit{and} \quad \mathbf{b}^T \mathbf{y}(\mu') > \mathbf{b}^T \mathbf{y}(\mu).$$

iii) *$(X(\mu), S(\mu))$ converges to an interior optimal solution pair for (CLP) and (CLD), where an interior optimal solution is an optimal solution at which the number inequality constraints are strictly satisfied the most for LP and SOCP, or the rank of the optimal solution is maximal for SDP.*

Proof. Note that

$$(X(\mu^0) - X(\mu)) \bullet (S(\mu^0) - S(\mu)) = 0,$$

since $(X(\mu^0) - X(\mu))$ in the null space of $\mathcal{A}$ and $(S(\mu^0) - S(\mu))$ in the row space of $\mathcal{A}$. This can be rewritten as

$$X(\mu) \bullet S(\mu^0) + S(\mu) \bullet X(\mu^0) = X(\mu^0) \bullet S(\mu^0) + X(\mu) \bullet S(\mu) \leq 2X(\mu^0) \bullet S(\mu^0).$$

Thus,

$$X(\mu) \bullet S(\mu^0) \leq 2X(\mu^0) \bullet S(\mu^0) \quad \text{and} \quad S(\mu) \bullet X(\mu^0) \leq X(\mu^0) \bullet S(\mu^0),$$

together with the fact that $X(\mu^0)$ and $S(\mu^0)$ are fixed and both in the interior of the cone, $X(\mu)$ and $S(\mu)$ must be bounded, which proves i).

We leave the proof of ii) as an exercise.

We now prove iii). Since $X(\mu)$ and $S(\mu)$ are both bounded, they have at least one limit point which we denote by $X(0)$ and $S(0)$. Let X^* and $(\mathbf{y}^*, S^*)$ be an interior optimal solution for (CLP) and (CLD), respectively. Then,

$$(X(\mu) - X^*) \bullet (S(\mu) - S^*) = 0$$

implies

$$X(\mu) \bullet S^* + S(\mu) \bullet X^* = X(\mu) \bullet S(\mu),\ \forall\ 0 < \mu \leq \mu^0.$$

Using conditions (3.4) and (3.5), we have

$$-\nabla B(S(\mu)) \bullet S^* - \nabla B(X(\mu)) \bullet X^* = \frac{1}{\mu} X(\mu) \bullet S(\mu),\ \forall\ 0 < \mu \leq \mu^0.$$

Thus, for all $\mu \in (0, \mu^0]$

$$-\nabla B(S(\mu)) \bullet S^* \leq \frac{1}{\mu} X(\mu) \bullet S(\mu) \text{ and } -\nabla B(X(\mu)) \bullet X^* \leq \frac{1}{\mu} X(\mu) \bullet S(\mu). \quad (3.11)$$

In the case of linear programming, using the definition of $\nabla B(\cdot)$ and $X(\mu) \bullet S(\mu) = n\mu$ we have

$$\sum_j \left(\frac{x_j^*}{x(\mu)_j} \right) \leq n \quad \text{and} \quad \sum_j \left(\frac{s_j^*}{s(\mu)_j} \right) \leq n,\ \forall\ 0 < \mu \leq \mu^0.$$

Thus, we must have

$$n x(\mu)_j \geq x_j^* \geq 0 \quad \text{and} \quad n s(\mu)_j \geq s_j^* \geq 0,\ \forall\ j,\ \forall\ 0 < \mu \leq \mu^0.$$

This implies that any limit point of $\mathbf{x}(\mu)$

$$x(0)_j > 0,\ \forall\ j \in \{j :\ x_j^* > 0\},$$

and any limit point of $\mathbf{s}(\mu)$

$$s(0)_j > 0,\ \forall\ j \in \{j :\ s_j^* > 0\}.$$

In other words, they are also interior optimal solutions as $(\mathbf{x}^*, \mathbf{s}^*)$ are.

In the case of semidefinite programming, using the definition of $\nabla B(\cdot)$ and we have $X(\mu) \bullet S(\mu) = n\mu$ we have

$$X(\mu)^{-1} \bullet X^* \leq n \quad \text{and} \quad S(\mu)^{-1} \bullet \leq n,\ \forall\ 0 < \mu \leq \mu^0.$$

Or

$$n X(\mu) \succeq X^* \quad \text{and} \quad n S(\mu) \succeq S^*,\ \forall\ 0 < \mu \leq \mu^0.$$

This implies that any limit point of $X(\mu)$ has a rank at least as great as X^*, and any limit point of $S(\mu)$ has a rank at least as great as S^*.

In the case of second-order cone programming, using the definition of $\nabla B(\cdot)$ and $X(\mu) \bullet S(\mu) = 2\mu$ we have for all $0 < \mu \leq \mu^0$,

$$\frac{1}{\delta(\mathbf{x}(\mu))^2} \begin{pmatrix} x(\mu)_1 \\ -\mathbf{x}(\mu)_{-1} \end{pmatrix} \bullet \begin{pmatrix} x_1^* \\ \mathbf{x}_{-1}^* \end{pmatrix} \leq 2$$

and

$$\frac{1}{\delta(\mathbf{s}(\mu))^2} \begin{pmatrix} s(\mu)_1 \\ -\mathbf{s}(\mu)_{-1} \end{pmatrix} \bullet \begin{pmatrix} s_1^* \\ \mathbf{s}_{-1}^* \end{pmatrix} \leq 2.$$

Let $\mathbf{x}^*$ be in the interior of the second-order cone, that is, $x_1^* > \|\mathbf{x}_{-1}^*\|$. Then

$$\delta(\mathbf{x}(\mu))^2 \geq x(\mu)_1 x_1^* - \mathbf{x}(\mu)_{-1}^T \mathbf{x}_{-1}^*, \ \forall\ 0 < \mu \leq \mu^0.$$

Since $\|\mathbf{x}(\mu)_{-1}\| \leq x(\mu)_1$, for all $0 < \mu \leq \mu^0$,

$$\begin{aligned} \delta(\mathbf{x}(\mu))^2 &\geq x(\mu)_1 x_1^* - \|\mathbf{x}(\mu)_{-1}\| \|\mathbf{x}_{-1}^*\| \\ &\geq x(\mu)_1 x_1^* - x(\mu)_1 \|\mathbf{x}_{-1}^*\| \\ &= x(\mu)_1 (x_1^* - \|\mathbf{x}_{-1}^*\|) \\ &\geq \delta(\mathbf{x}(\mu))(x_1^* - \|\mathbf{x}_{-1}^*\|). \end{aligned}$$

But $\delta(\mathbf{x}(\mu)) > 0$ for all $0 < \mu \leq \mu^0$, we have

$$\delta(\mathbf{x}(\mu)) \geq x_1^* - \|\mathbf{x}_{-1}^*\| > 0, \ \forall\ 0 < \mu \leq \mu^0,$$

so that any limit of $\mathbf{x}(\mu)$ has $\delta(\mathbf{x}(0)) > 0$ and it is in the interior of the second-order cone. ∎

3.1.3 Path following algorithms

Suppose we have an approximate central path point $(X, \mathbf{y}, S)$ in a neighborhood of $(X(\mu), \mathbf{y}(\mu), S(\mu))$ for a given $\mu > 0$. Then we consider to compute a new approximate central-path point $(X^+, \mathbf{y}^+, S^+)$ corresponding to a chosen μ^+ where $0 < \mu^+ < \mu$. If one repeats this process, then a sequence of approximate central-path points $(X^k, \mathbf{y}^k, S^k)$, corresponding to $\mu^0 > \mu^1 > ... > \mu^k, ...$, would be generated, and it converges to the optimal solution set as $\mu^k \to 0$. Such an algorithm is called the central-path following algorithm.

If μ^+ is close to μ, we expect $(X(\mu^+), \mathbf{y}(\mu^+), S(\mu^+))$ is also close to $(X, \mathbf{y}, S)$, so that $(X, \mathbf{y}, S)$ would be a good initial point for computing $(X^+, \mathbf{y}^+, S^+)$ by numerical procedures such as Newton's method. Denote by $(D_X, \mathbf{d}_\mathbf{y}, D_S)$ the Newton direction at initial solution $(X, \mathbf{y}, S)$:

$$D_X = X^+ - X, \ \mathbf{d}_\mathbf{y} = \mathbf{y}^+ - \mathbf{y}, \ D_S = S^+ - S.$$

Since there are three different sets of conditions to describe the central-path condition, namely primal system (3.4), dual system (3.5) and primal-dual symmetric system, the Newton direction may computed from three different sets of equations.

First, $(D_X, \mathbf{d_y}, D_S)$ must meet the equality constraints

$$\text{(Feasibility)} \qquad \begin{array}{rcl} \mathcal{A}D_X & = & \mathbf{0} \\ -\mathcal{A}^T\mathbf{d_y} - D_S & = & \mathbf{0}. \end{array} \tag{3.12}$$

Then, the remaining equation based on the primal system:

$$\text{(Primal-Newton)} \qquad D_S + \mu^+\nabla^2 B(X)D_X = -(S + \mu^+\nabla B(X)); \tag{3.13}$$

the remaining equation based on the dual system:

$$\text{(Dual-Newton)} \qquad D_X + \mu^+\nabla^2 B(S)D_S = -(X + \mu^+\nabla B(S)); \tag{3.14}$$

and the remaining equation based on the primal-dual symmetric system is more involved

$$\text{(Primal-Dual-Newton)} \qquad D_S + \mu^+\nabla^2 B(Z)D_X = -(S + \mu^+\nabla B(X). \tag{3.15}$$

where $Z \in \mathcal{E}$ is chosen to satisfy

$$S = \nabla^2 B(Z)X. \tag{3.16}$$

We explain solution Z of (3.16) in more detail.

Example 3.2 *The following are solutions Z (LP), (SDP) and (SOCP).*

- *For LP cone*

$$\mathbf{z} = \mathbf{s}./\mathbf{x} \in \mathcal{R}^n, \quad \textit{that is} \quad \mathbf{z} = \begin{pmatrix} \frac{s_1}{x_1} \\ \vdots \\ \frac{s_n}{x_n} \end{pmatrix}.$$

- *For SDP cone*

$$Z = X^{.5}(X^{.5}SX^{.5})^{-.5}X^{.5} \in \mathcal{S}^n$$

- *For SOCP cone*

$$\mathbf{z} = \begin{pmatrix} \zeta x_1 + \eta s_1 \\ \zeta \mathbf{x}_{-1} - \eta \mathbf{s}_{-1} \end{pmatrix} \in \mathcal{R}^n$$

where

$$\zeta = \frac{2}{\sqrt{\delta(\mathbf{x})\delta(\mathbf{s}) + \mathbf{x}^T\mathbf{s}}} \quad \textit{and} \quad \eta = \zeta\frac{\delta(\mathbf{x})}{\delta(\mathbf{s})}.$$

One way to solve these systems is to form a normal equation: for example, pre-multiplying Dual-Newton (3.14) on both sides by $\mathcal{A}$, we have

$$\mathcal{A}D_X + \mu^+\mathcal{A}\nabla^2 B(S)D_S = -\mathcal{A}(X + \mu^+\nabla B(S)).$$

Then, using (3.12), it becomes

$$\mathcal{A}\nabla^2 B(S)\mathcal{A}^T\mathbf{d_y} = \frac{1}{\mu^+}\mathbf{b} + \mathcal{A}\nabla B(S), \tag{3.17}$$

where $\mathcal{A}\nabla^2 B(S)\mathcal{A}^T$ is called Gram matrix.

Example 3.3 *The detailed dual Gram matrices for LP, SDP and SOCP are:*

- *for LP, it is*
$$A\Delta(\mathbf{s})^{-2}A^T;$$
- *for SDP, it is*
$$\mathcal{A}S^{-2}\mathcal{A}^T := \begin{pmatrix} A_1S^{-1} \bullet S^{-1}A_1 & \cdots & A_1S^{-1} \bullet S^{-1}A_m \\ \vdots & \ddots & \vdots \\ A_mS^{-1} \bullet S^{-1}A_1 & \cdots & A_mS^{-1} \bullet S^{-1}A_m \end{pmatrix};$$
- *for SOCP, it is*
$$A\left(\frac{1}{\delta(\mathbf{s})^2}\begin{pmatrix} -1 & 0 \\ 0 & I \end{pmatrix} + \frac{2}{\delta(\mathbf{x})^4}\begin{pmatrix} s_1 \\ -\mathbf{s}_{-1} \end{pmatrix}\begin{pmatrix} s_1 \\ -\mathbf{s}_{-1} \end{pmatrix}^T\right)A^T.$$

Once have $\mathbf{d_y}$ calculated, we use (3.12) to calculate D_S and (3.14) to calculate D_X.

For Primal-Newton and Primal-Dual-Newton, we pre-multiply first $\nabla^2 B(X)^{-1}$ and $\nabla^2 B(Z)^{-1}$, respectively; and then pre-multiply $\mathcal{A}$ to obtain the normal equations. Once all Newton directions are calculated, the new approximate solution would be

$$X^+ = X + D_X,\ \mathbf{y}^+ = \mathbf{y} + \mathbf{d_y},\ S = S^0 + D_S,$$

and we may repeat the Newton step several times to find an approximate solution $(X^+, \mathbf{y}^+, S^+)$ that is in the neighborhood of the exact central-path point $(X(\mu^+), \mathbf{y}(\mu^+), S(\mu^+))$.

The following theorem further characterizes the path following algorithm.

Theorem 3.4 *There is a neighborhood $\mathcal{N}(\mu)$ of the central path such that for $(X, \mathbf{y}, S) \in \mathcal{N}(\mu)$, one can set $\mu^+ = (1 - \frac{O(1)}{\sqrt{\nu}})\mu$ and apply one Newton step to calculate $(X^+, \mathbf{y}^+, S^+) \in \mathcal{N}(\mu^+)$.*

3.2 Potential Reduction Algorithms

In practical computation, it is more efficient to generate iterative solutions in a large neighborhood as long as a merit function is monotonically decreasing, so that the greater the reduction of the function, the faster convergence of the iterative solutions to optimality. Such an algorithm is said function-driven. If the merit function is the objective function itself, a function-driven algorithm is likely to generate iterative solutions being prematurely too close to the boundary, and the convergence would be slow down in future iterations. A better driven function should balance the reduction of the objective function as well as a good position in the (interior) of the feasible region – we now present a potential function logarithmically combining the objective function and the barrier function.

3.2.1 Potential functions

Let X be any interior feasible solution of (CLP) and $(\mathbf{y}, S)$ be any interior feasible solution of (CLP). Then, a primal-dual potential function is defined as

$$\psi_\rho(X,S) := \rho \ln(X \bullet S) + B(X) + B(S), \tag{3.18}$$

for any positive integer ρ.

Proposition 3.5 *For any $X \in \overset{\circ}{K}$ and $S \in \overset{\circ}{K}^*$,*

i) if $K = \mathcal{R}^n_+$ or $K = \mathcal{S}^n_+$

$$\psi_n(X,S) := n \ln(X \bullet S) + B(X) + B(S) \geq n \ln(n);$$

ii) if $K = \mathcal{N}^n_2$

$$\psi_1(\mathbf{x},\mathbf{s}) := \ln(\mathbf{x} \bullet \mathbf{s}) + B(\mathbf{x}) + B(\mathbf{s}) \geq 0;$$

iii) if $K = K_1 \oplus K_2$ equipped with barrier-coefficients ν_1 and ν_2,

$$\psi_{\nu_1+\nu_2}(X,S) := (\nu_1+\nu_2) \ln(X \bullet S) + B(X) + B(S) \geq (\nu_1+\nu_2) \ln(\nu_1+\nu_2).$$

Proof. We leave i) as an exercise. To prove ii) of the second-order cone, we see first

$$\psi_1(\mathbf{x},\mathbf{s}) := \ln(\mathbf{x} \bullet \mathbf{s}) + B(\mathbf{x}) + B(\mathbf{s}) = \ln\left(\frac{\mathbf{x} \bullet \mathbf{s}}{\delta(\mathbf{x})\delta(\mathbf{s})}\right).$$

But

$$\mathbf{x} \bullet \mathbf{s} = x_1 s_1 + \mathbf{x}_{-1}^T \mathbf{s}_{-1} \geq x_1 s_1 - \|\mathbf{x}_{-1}\| \|\mathbf{s}_{-1}\| > 0.$$

Thus,

$$\begin{array}{rcl} (\mathbf{x} \bullet \mathbf{s})^2 & \geq & (x_1 s_1)^2 - 2 x_1 s_1 \|\mathbf{x}_{-1}\| \|\mathbf{s}_{-1}\| + \|\mathbf{x}_{-1}\|^2 \|\mathbf{s}_{-1}\|^2 \\ & \geq & (x_1 s_1)^2 - x_1^2 \|\mathbf{s}_{-1}\|^2 - s_1^2 \|\mathbf{x}_{-1}\|^2 + \|\mathbf{x}_{-1}\|^2 \|\mathbf{s}_{-1}\|^2 \\ & = & (x_1^2 - \|\mathbf{x}_{-1}\|^2)(s_1^2 - \|\mathbf{s}_{-1}\|^2) \\ & = & \delta(\mathbf{x})^2 \delta(\mathbf{s})^2, \end{array}$$

which proves ii).

To prove iii), we see $X = [X_1; X_2]$ and $S = [S_1; S_2]$, $B(X) = B_1(X_1) + B_2(X_2)$, $B(S) = B_1(S_1) + B_2(S_2)$,

$$X \bullet S = X_1 \bullet S_1 + X_1 \bullet S_1.$$

$$\begin{array}{rcl} \dfrac{X \bullet S}{\nu_1+\nu_2} & = & \dfrac{X_1 \bullet S_1 + X_1 \bullet S_1}{\nu_1+\nu_2} \\ & = & \dfrac{\nu_1}{\nu_1+\nu_2} \dfrac{X_1 \bullet S_1}{\nu_1} + \dfrac{\nu_2}{\nu_1+\nu_2} \dfrac{X_2 \bullet S_2}{\nu_2} \\ & \geq & \left(\dfrac{X_1 \bullet S_1}{\nu_1}\right)^{\nu_1/(\nu_1+\nu_2)} \cdot \left(\dfrac{X_2 \bullet S_2}{\nu_2}\right)^{\nu_2/(\nu_1+\nu_2)}. \end{array}$$

Thus,

$$\begin{aligned}\ln(X \bullet S) - \ln(\nu_1 + \nu_2) &\geq \frac{1}{\nu_1+\nu_2}(\nu_1 \ln(\frac{X_1 \bullet S_1}{\nu_1}) + \nu_2 \ln(\frac{X_1 \bullet S_1}{\nu_2})) \\ &\geq \frac{1}{\nu_1+\nu_2}(-B(X_1) - B(S_1) - B(X_2) - B(S_2)).\end{aligned}$$

Multiplying $(\nu_1 + \nu_2)$ on both sides proves iii). ∎

In summary, for any closed convex cone K with a barrier function $B(\cdot)$ of barrier-coefficient ν, we have

$$\psi_\nu(X, S) = \nu \ln(X \bullet S) - B(X) - B(S) \geq \nu \ln(\nu). \tag{3.19}$$

Thus, for $\rho > \nu$,

$$\begin{aligned}\psi_\rho(X, S) &= (\rho - \nu) \ln(X \bullet S) + \psi_\nu(X, S) \\ &\geq (\rho - \nu) \ln(X \bullet S) + \nu \ln(\nu) \geq (\rho - \nu) \ln(X \bullet S).\end{aligned}$$

Then, if there is a sequence of $(X^k \in \overset{\circ}{K}, S^k \in \overset{\circ}{K})$ such that $\psi_\rho(X^k, S^k) \to -\infty$, it must be true $\ln(X^k \bullet S^k) \to -\infty$ or $(X^k \bullet S^k) \to 0$. More precisely, we have

$$X^k \bullet S^k \leq \exp(\frac{\psi_\rho(X^k, S^k) - \nu \ln(\nu)}{\rho - \nu}). \tag{3.20}$$

Moreover, we have the following theorem:

Theorem 3.6 *Let $X \in \overset{\circ}{\mathcal{F}}_p$ and $(\mathbf{y}, S) \in \overset{\circ}{\mathcal{F}}_d$ or, in short, $(X, \mathbf{y}, S) \in \overset{\circ}{\mathcal{F}}$, and let the level set*

$$\Psi(\delta) := \{(X, \mathbf{y}, S) : \ \psi_\rho(X, S) \leq \delta\}.$$

Then,

i)

$$\Psi(\delta^1) \subset \Psi(\delta^2) \quad \textit{if} \quad \delta^1 \leq \delta^2.$$

ii) *For every δ, $\Psi(\delta)$ is bounded and its closure $\hat{\Psi}(\delta)$ has non-empty intersection with the solution set.*

Next we will show that a potential reduction algorithm generates sequences $\{X^k, \mathbf{y}^k, S^k\} \in \overset{\circ}{\mathcal{F}}$ such that

$$\psi_{\nu+\sqrt{\nu}}(X^{k+1}, \mathbf{y}^{k+1}, S^{k+1}) \leq \psi_{\nu+\sqrt{\nu}}(X^k, \mathbf{y}^k, S^k) - .05$$

for $k = 0, 1, 2, ...$ This indicates that the level sets shrink at least a constant rate independently of m or n.

3.2.2 Potential reduction algorithms

The primal-dual potential function of (3.18) can be split into a primal potential and a dual potential. Then, for any $X \in \overset{\circ}{\mathcal{F}}_p$ and $(\mathbf{y}, S) \in \overset{\circ}{\mathcal{F}}_d$, the primal potential function is defined as

$$\psi_\rho^p(X) := \rho \ln(S \bullet X) + B(X),$$

where S is dual feasible and fixed; the dual potential function is defined as

$$\psi_\rho^d(S) := \rho \ln(X \bullet S) + B(S),$$

where X is primal feasible and fixed. Here, we let $\rho \geq \nu$. They related to ψ in the following way:

$$\begin{array}{rcl} \psi_\rho(X,S) & = & \rho \ln(X \bullet S) + B(X) + B(S) \\ & = & \psi_\rho^p(X) + B(S) \\ & = & \psi_\rho^d(S) + B(X), \end{array}$$

where $\rho \geq \nu$.

For any interior feasible solution $(X, \mathbf{y}, S)$, regardless it is close or not to the central path, we compute a potential-reduction direction $(D_X, \mathbf{d_y}, D_S)$ as follows. Again, it first satisfies the feasibility equation (3.12), that is,

$$\begin{array}{rcl} \mathcal{A}D_X & = & \mathbf{0} \\ -\mathcal{A}^T\mathbf{d_y} - D_S & = & \mathbf{0}. \end{array}$$

Then, the remaining equation can drawn from three potential functions: the primal-based, the dual-based, and the primal-dual-based. The remaining condition based on the primal:

$$\text{(Primal)} \qquad D_S + \nabla^2 B(X) D_X = -\nabla \psi_\rho^p(X); \tag{3.21}$$

the remaining equation based on the dual system:

$$\text{(Dual)} \qquad D_X + \nabla^2 B(S) D_S = -\nabla \psi_\rho^d(S); \tag{3.22}$$

and the remaining equation based on the primal-dual symmetric system is more involved

$$\text{(Primal-Dual)} \qquad D_S + \nabla^2 B(Z) D_X = -\nabla_X \psi_\rho^p(X). \tag{3.23}$$

Once these directions are calculated, the new approximate solution would be

$$X^+ = X + \alpha D_X, \ \mathbf{y}^+ = \mathbf{y} + \beta \mathbf{d_y}, \ S = S + \beta D_S,$$

where step-sizes α and β are chosen to make them feasible and the primal-dual potential function reduced most.

Example 3.4 *Consider LP, SDP and SOCP.*

- *For LP, the primal, dual and primal-dual potential-reduction directions would be*

$$\textit{(LP-Primal)} \quad \mathbf{d_s} + \Delta(\mathbf{x})^{-2}\mathbf{d_x} = -\tfrac{\rho}{\mathbf{x}^T\mathbf{s}}\mathbf{s} + \Delta(\mathbf{x})^{-1}\mathbf{e};$$

$$\textit{(LP-Dual)} \quad \mathbf{d_x} + \Delta(\mathbf{s})^{-2}\mathbf{d_s} = -\tfrac{\rho}{\mathbf{x}^T\mathbf{s}}\mathbf{x} + \Delta(\mathbf{s})^{-1}\mathbf{e};$$

and

$$\textit{(LP-Primal-Dual)} \quad \mathbf{d_s} + \Delta(\mathbf{x})^{-1}\Delta(\mathbf{s})\mathbf{d_x} = -\tfrac{\rho}{\mathbf{x}^T\mathbf{s}}\mathbf{s} + \Delta(\mathbf{x})^{-1}\mathbf{e}$$

or after pre-multiplying $\Delta(\mathbf{x})^{-1}$ on both sides

$$\textit{(LP-Primal-Dual)} \quad \Delta(\mathbf{x})\mathbf{d_s} + \Delta(\mathbf{s})\mathbf{d_x} = -\tfrac{\rho}{\mathbf{x}^T\mathbf{s}}(\mathbf{x}. \cdot \mathbf{s}) + \mathbf{e}.$$

Recall that $\Delta(\cdot)$ is a diagonal matrix whose diagonal entries are elements of an input vector.

- *For SDP, the primal, dual and primal-dual potential-reduction directions would be*

$$\textit{(SDP-Primal)} \quad D_S + X^{-1}D_X X^{-1} = -\tfrac{\rho}{X \bullet S}S + X^{-1};$$

$$\textit{(SDP-Dual)} \quad D_X + S^{-1}D_S S^{-1} = -\tfrac{\rho}{X \bullet S}X + S^{-1};$$

and

$$\textit{(SDP-Primal-Dual)} \quad D_S + Z^{-1}D_X Z^{-1} = -\tfrac{\rho}{X \bullet S}S + X^{-1}.$$

where

$$Z = X^{.5}(X^{.5}SX^{.5})^{-.5}X^{.5}.$$

- *For SOCP, the primal, dual and primal-dual directions would be*

$$\textit{(SOCP-Primal)}$$

$$\mathbf{d_s} + \left(\tfrac{1}{\delta(\mathbf{x})^2}\begin{pmatrix} -1 & 0 \\ 0 & I \end{pmatrix} + \tfrac{2}{\delta(\mathbf{x})^4}\begin{pmatrix} x_1 \\ -\mathbf{x}_{-1} \end{pmatrix}\begin{pmatrix} x_1 \\ -\mathbf{x}_{-1} \end{pmatrix}^T\right)\mathbf{d_x} = -\tfrac{\rho}{\mathbf{x}^T\mathbf{s}}\mathbf{s} + \Delta(\mathbf{x})^{-1}\mathbf{e};$$

$$\textit{(SOCP-Dual)}$$

$$\mathbf{d_x} + \left(\tfrac{1}{\delta(\mathbf{s})^2}\begin{pmatrix} -1 & 0 \\ 0 & I \end{pmatrix} + \tfrac{2}{\delta(\mathbf{x})^4}\begin{pmatrix} s_1 \\ -\mathbf{s}_{-1} \end{pmatrix}\begin{pmatrix} s_1 \\ -\mathbf{s}_{-1} \end{pmatrix}^T\right)\mathbf{d_s} = -\tfrac{\rho}{\mathbf{x}^T\mathbf{s}}\mathbf{x} + \Delta(\mathbf{s})^{-1}\mathbf{e};$$

and

$$\textit{(SOCP-Primal-Dual)}$$

$$\mathbf{d_s} + \left(\tfrac{1}{\delta(\mathbf{z})^2}\begin{pmatrix} -1 & 0 \\ 0 & I \end{pmatrix} + \tfrac{2}{\delta(\mathbf{z})^4}\begin{pmatrix} z_1 \\ -\mathbf{z}_{-1} \end{pmatrix}\begin{pmatrix} z_1 \\ -\mathbf{z}_{-1} \end{pmatrix}^T\right)\mathbf{d_x} = -\tfrac{\rho}{\mathbf{x}^T\mathbf{s}}\mathbf{s} + \Delta(\mathbf{x})^{-1}\mathbf{e}$$

where $\mathbf{z}$ is given in (3.2).

The following theorem characterizes the efficiency of these potential-reduction algorithms.

Theorem 3.7 *For any* $(X, \mathbf{y}, S) \in \overset{\circ}{\mathcal{F}}$*, there are close-form formula to choose step-sizes* α *and* β *such that*

$$\psi_\rho(X^+, \mathbf{y}^+, S^+) \leq \psi_\rho(X, \mathbf{y}, S) - .05.$$

3.2.3 Analysis of the primal potential-reduction

Given a pair of $(X, \mathbf{y}, S) \in \overset{\circ}{\mathcal{F}}$, we fix $(\mathbf{y}, S)$ and consider only update X. For direction $D_X \in \mathcal{E}$, the primal potential function

$$\psi_\rho^p(X + D_X) := \rho \ln(S \bullet (X + D_X)) + B(X + D_X).$$

From the concavity of $\rho \ln()$

$$\rho \ln(S \bullet (X + D_X)) - \rho \ln(S \bullet X) \leq \frac{\rho}{S \bullet X} S \bullet D_X.$$

From Lemma 3.1,

$$B(X + D_X) - B(X) \leq \nabla B(X) \bullet D_X + \frac{\gamma^2 \|D_X\|^2_{\nabla^2 B(X)}}{2(1 - \gamma \|D_X\|_{\nabla^2 B(X)})}.$$

Combine the two, we have an overestimate function for the potential difference

$$\begin{array}{rl} & \psi_\rho^p(X + D_X) - \psi_\rho^p(X) \\ \leq & \frac{\rho}{S^k \bullet X^k} S \bullet D_X + \nabla B(X) \bullet D_X + \frac{\gamma^2 \|D_X\|^2_{\nabla^2 B(X)}}{2(1 - \gamma \|D_X\|_{\nabla^2 B(X)})} \\ = & \nabla \psi_\rho^p(X) \bullet D_X + \frac{\gamma^2 \|D_X\|^2_{\nabla^2 B(X)}}{2(1 - \gamma \|D_X\|_{\nabla^2 B(X)})} \end{array}$$

where $\gamma = 1$ for LP and SDP and $\gamma = \sqrt{2}$ for SOCP. This overestimate function is more or less a quadratic in D_X. Thus, the Newton step would compute the direction

$$\begin{array}{rl} \text{minimize} & \nabla \psi_\rho^p(X) \bullet D_X + \frac{1}{2} \|D_X\|^2_{\nabla^2 B(X)} \\ \text{subject to} & \mathcal{A} D_X = \mathbf{0}. \end{array}$$

The optimality condition of the Newton step is exactly (3.21) with $\mathbf{d_y}$ (and $D_S = -\mathcal{A}^T \mathbf{d_y}$) as the Lagrangian multiplier vector:

$$D_S + \nabla^2 B(X) D_X \quad = \quad -\nabla \psi_\rho^p(X).$$

Let the minimizer be $\bar{D}_X$. Then, since $D_S \bullet D_X = 0$,

$$\|\bar{D}_X\|^2_{\nabla^2 B(X)} = \bar{D}_X \bullet \nabla^2 B(X) \bar{D}_X = -\nabla \psi_\rho^p(X) \bullet \bar{D}_X.$$

If $\|\bar{D}_X\|^2_{\nabla^2 B(X)} \geq 1$, we make update

$$X^+ = X + \frac{\alpha}{\gamma} \frac{\bar{D}_X}{\|\bar{D}_X\|_{\nabla^2 B(X)}}, \tag{3.24}$$

where α is a constant that will be specified later. Clearly,

$$\gamma \|X^+ - X\|_{\nabla^2 B(X)} \leq \alpha$$

so that

$$\begin{aligned} \psi^p_\rho(X^+) - \psi^p_\rho(X) &\leq \nabla \psi^p_\rho(X) \bullet (X^+ - X) + \frac{\gamma^2 \|X^+ - X\|^2_{\nabla^2 B(X)}}{2(1 - \gamma \|X^+ - X\|_{\nabla^2 B(X)})} \\ &= -\frac{\alpha}{\gamma} \|\bar{D}_X\|_{\nabla^2 B(X)} + \frac{\alpha^2}{2(1-\alpha)} \\ &\leq -\frac{\alpha}{\gamma} + \frac{\alpha^2}{2(1-\alpha)}, \end{aligned}$$

where we recall that $\gamma = 1$ for LP and SDP, and $\gamma = \sqrt{2}$ for SOCP from Lemma 3.1. Thus, we can choose an α such that the $\psi^p_\rho(X^+)$ is reduced by a constant from $\psi^p_\rho(X)$. Moreover, from the relation beween $\psi^p_\rho(X)$ and $\psi_\rho(X, S)$, we let $S^+ = S$ so that $\psi_\rho(X^+, S^+)$ is reduced by the same constant from $\psi_\rho(X, S)$.

Now we consider the case $\|\bar{D}_X\|^2_{\nabla^2 B(X)} < 1$. Note that

$$\begin{aligned} D_X &= -\nabla^2 B(X)^{-1} \left(\nabla \psi^p_\rho(X) + D_S \right) \\ &= -\nabla^2 B(X)^{-1} \left(\tfrac{\rho}{S \bullet X} S + \nabla B(X) + D_S \right) \\ &= -\nabla^2 B(X)^{-1} \left(\tfrac{\rho}{S \bullet X} (S + \tfrac{S \bullet X}{\rho} D_S) + \nabla B(X) \right) \\ &= -\tfrac{\rho}{S \bullet X} \nabla^2 B(X)^{-1} \left(S^+ + \tfrac{S \bullet X}{\rho} \nabla B(X) \right), \end{aligned}$$

where $S^+ = S + \frac{S \bullet X}{\rho} D_S$. Then, $\|\bar{D}_X\|^2_{\nabla^2 B(X)} < 1$ implies

$$\|S^+ + \frac{S \bullet X}{\rho} \nabla B(X)\|_{\nabla^2 B(X)^{-1}} \leq \frac{S \bullet X}{\rho}.$$

Let $(x, y, s) \in \overset{\circ}{\mathcal{F}}$. Then consider the primal-dual potential function:

$$\psi_{n+\rho}(x, s) = (n + \rho) \ln(x^T s) - \sum_{j=1}^n \ln(x_j s_j),$$

where $\rho \geq 0$. Let $z = \mathbf{b}^T y$, then $s^T x = c^T x - z$ and we have

$$\psi_{n+\rho}(x, s) = \mathcal{P}_{n+\rho}(x, z) - \sum_{j=1}^n \ln s_j.$$

Recall that when $\rho = 0$, $\psi_{n+\rho}(x, s)$ is minimized along the central path. However, when $\rho > 0$, $\psi_{n+\rho}(x, s) \to -\infty$ means that x and s converge to the

optimal face, and the descent gets steeper as ρ increases. In this section we choose $\rho = \sqrt{n}$.

The process calculates steps for x and s, which guarantee a constant reduction in the primal-dual potential function. As the potential function decreases, both x and s are forced to an optimal solution pair.

Consider a pair of $(x^k, y^k, s^k) \in \overset{\circ}{\mathcal{F}}$. Fix $z^k = \mathbf{b}^T y^k$, then the gradient vector of the primal potential function at x^k is

$$\nabla \mathcal{P}_{n+\rho}(x^k, z^k) = \frac{(n+\rho)}{(s^k)^T x^k} c - (X^k)^{-1} e = \frac{(n+\rho)}{c^T x^k - z^k} c - (X^k)^{-1} e.$$

We directly solve the ball-constrained linear problem for direction d_x:

$$\begin{array}{cc} \text{minimize} & \nabla \mathcal{P}_{n+\rho}(x^k, z^k)^T d_x \\ \text{s.t.} & A d_x = 0,\ \|(X^k)^{-1} d_x\| \leq \alpha. \end{array}$$

Let the minimizer be d_x. Then

$$d_x = -\alpha \frac{X^k p^k}{\|p^k\|} \ ,$$

where

$$p^k = p(z^k) :== (I - X^k A^T (A(X^k)^2 A^T)^{-1} A X^k) X^k \nabla \mathcal{P}_{n+\rho}(x^k, z^k).$$

Update

$$x^{k+1} = x^k + d_x = x^k - \alpha \frac{X^k p^k}{\|p^k\|} \ , \tag{3.25}$$

and,

$$\mathcal{P}_{n+\rho}(x^{k+1}, z^k) - \mathcal{P}_{n+\rho}(x^k, z^k) \leq -\alpha \|p^k\| + \frac{\alpha^2}{2(1-\alpha)} \ .$$

Here, we have used the fact

$$\begin{aligned} & \mathcal{P}_{n+\rho}(x^{k+1}, z^k) - \mathcal{P}_{n+\rho}(x^k, z^k) \\ \leq & \ \frac{n+\rho}{c^T x^k - z^k} c^T d_x - e^T (X^k)^{-1} d_x + \frac{\|(X^k)^{-1} d_x\|^2}{2(1 - \|(X^k)^{-1} d_x\|_\infty)} \\ = & \ \nabla \mathcal{P}(x^k, z^k)^T d_x + \frac{\|(X^k)^{-1} d_x\|^2}{2(1 - \|(X^k)^{-1} d_x\|_\infty)} \\ = & \ -\alpha \|p^k\| + \frac{\alpha^2}{2(1-\alpha)} \end{aligned}$$

Thus, as long as $\|p^k\| \geq \eta > 0$, we may choose an appropriate α such that

$$\mathcal{P}_{n+\rho}(x^{k+1}, z^k) - \mathcal{P}_{n+\rho}(x^k, z^k) \leq -\delta$$

for some positive constant δ. By the relation between $\psi_{n+\rho}(x, s)$ and $\mathcal{P}_{n+\rho}(x, z)$, the primal-dual potential function is also reduced. That is,

$$\psi_{n+\rho}(x^{k+1}, s^k) - \psi_{n+\rho}(x^k, s^k) \leq -\delta.$$

However, even if $\|p^k\|$ is small, we will show that the primal-dual potential function can be reduced by a constant δ by increasing z^k and updating (y^k, s^k).

We focus on the expression of p^k, which can be rewritten as

$$\begin{aligned} p^k &= (I - X^k A^T (A(X^k)^2 A^T)^{-1} A X^k)(\frac{(n+\rho)}{c^T x^k - z^k} X^k c - e) \\ &= \frac{(n+\rho)}{c^T x^k - z^k} X^k s(z^k) - e, \end{aligned} \tag{3.26}$$

where

$$s(z^k) = \mathbf{c} - A^T \mathbf{y}(z^k) \tag{3.27}$$

and

$$\begin{aligned} y(z^k) &= y_2 - \tfrac{c^T x^k - z^k}{(n+\rho)} y_1, \\ y_1 &= (A(X^k)^2 A^T)^{-1} b, \\ y_2 &= (A(X^k)^2 A^T)^{-1} A (X^k)^2 c. \end{aligned} \tag{3.28}$$

Regarding $\|p^k\| = \|p(z^k)\|$, we have the following lemma:

Lemma 3.8 *Let*

$$\mu^k = \frac{(x^k)^T s^k}{n} = \frac{c^T x^k - z^k}{n} \quad \textit{and} \quad \mu = \frac{(x^k)^T s(z^k)}{n} \ .$$

If

$$\|p(z^k)\| < \min\left(\eta \sqrt{\frac{n}{n+\eta^2}}\ , 1-\eta\right), \tag{3.29}$$

then the following three inequalities hold:

$$s(z^k) > 0, \quad \|X^k s(z^k) - \mu e\| < \eta\mu, \quad \textit{and} \quad \mu < (1 - .5\eta/\sqrt{n})\mu^k. \tag{3.30}$$

Proof. The proof is by contradiction.

i) If the first inequality of (3.30) is not true, then $\exists\ j$ such that $s_j(z^k) \le 0$ and

$$\|p(z^k)\| \ge 1 - \frac{(n+\rho)}{n\mu^k} x_j s_j(z^k) \ge 1.$$

ii) If the second inequality of (3.30) does not hold, then

$$\begin{aligned} \|p(z^k)\|^2 &= \|\frac{(n+\rho)}{n\mu^k} X^k s(z^k) - \frac{(n+\rho)\mu}{n\mu^k} e + \frac{(n+\rho)\mu}{n\mu^k} e - e\|^2 \\ &= (\frac{(n+\rho)}{n\mu^k})^2 \|X^k s(z^k) - \mu e\|^2 + \|\frac{(n+\rho)\mu}{n\mu^k} e - e\|^2 \\ &\ge (\frac{(n+\rho)\mu}{n\mu^k})^2 \eta^2 + (\frac{(n+\rho)\mu}{n\mu^k} - 1)^2 n \\ &\ge \eta^2 \frac{n}{n+\eta^2}\ , \end{aligned} \tag{3.31}$$

where the last relation prevails since the quadratic term yields the minimum at

$$\frac{(n+\rho)\mu}{n\mu^k} = \frac{n}{n+\eta^2}\ .$$

iii) If the third inequality of (3.30) is violated, then

$$\frac{(n+\rho)\mu}{n\mu^k} \geq (1+\frac{1}{\sqrt{n}})(1-\frac{.5\eta}{\sqrt{n}}) \geq 1,$$

which, in view of (3.31), leads to

$$\begin{aligned} \|p(z^k)\|^2 &\geq (\frac{(n+\rho)\mu}{n\mu^k}-1)^2 n \\ &\geq ((1+\frac{1}{\sqrt{n}})(1-\frac{.5\eta}{\sqrt{n}})-1)^2 n \\ &\geq (1-\frac{\eta}{2}-\frac{\eta}{2\sqrt{n}})^2 \\ &\geq (1-\eta)^2. \end{aligned}$$

∎

The lemma says that, when $\|p(z^k)\|$ is small, then $(x^k, y(z^k), s(z^k))$ is in the neighborhood of the central path and $\mathbf{b}^T y(z^k) > z^k$. Thus, we can increase z^k to $\mathbf{b}^T y(z^k)$ to cut the dual level set $\Omega(z^k)$. We have the following potential reduction theorem to evaluate the progress.

Theorem 3.9 *Given $(x^k, y^k, s^k) \in \overset{\circ}{\mathcal{F}}$. Let $\rho = \sqrt{n}$, $z^k = \mathbf{b}^T y^k$, x^{k+1} be given by (3.25), and $y^{k+1} = y(z^k)$ in (3.28) and $s^{k+1} = s(z^k)$ in (3.27). Then, either*

$$\psi_{n+\rho}(x^{k+1}, s^k) \leq \psi_{n+\rho}(x^k, s^k) - \delta$$

or

$$\psi_{n+\rho}(x^k, s^{k+1}) \leq \psi_{n+\rho}(x^k, s^k) - \delta$$

where $\delta > 1/20$.

Proof. If (3.29) does not hold, i.e.,

$$\|p(z^k)\| \geq \min\left(\eta\sqrt{\frac{n}{n+\eta^2}}\,, 1-\eta\right),$$

then

$$\mathcal{P}_{n+\rho}(x^{k+1}, z^k) - \mathcal{P}_{n+\rho}(x^k, z^k) \leq -\alpha \min\left(\eta\sqrt{\frac{n}{n+\eta^2}}\,, 1-\eta\right) + \frac{\alpha^2}{2(1-\alpha)}\,,$$

hence from the relation between $\mathcal{P}_{n+\rho}$ and $\psi_{n+\rho}$,

$$\psi_{n+\rho}(x^{k+1}, s^k) - \psi_{n+\rho}(x^k, s^k) \leq -\alpha \min\left(\eta\sqrt{\frac{n}{n+\eta^2}}\,, 1-\eta\right) + \frac{\alpha^2}{2(1-\alpha)}\,.$$

Otherwise, from Lemma 3.8 the inequalities of (3.30) hold:

i) The first of (3.30) indicates that y^{k+1} and s^{k+1} are in $\overset{\circ}{\mathcal{F}}_d$.

ii) Using the second of (3.30) and applying Lemma **??** to vector $X^k s^{k+1}/\mu$, we have

$$\begin{aligned} & n\ln(x^k)^T s^{k+1} - \sum_{j=1}^{n} \ln(x_j^k s_j^{k+1}) \\ = \;& n\ln n - \sum_{j=1}^{n} \ln(x_j^k s_j^{k+1}/\mu) \\ \leq \;& n\ln n + \frac{\|X^k s^{k+1}/\mu - e\|^2}{2(1 - \|X^k s^{k+1}/\mu - e\|_\infty)} \\ \leq \;& n\ln n + \frac{\eta^2}{2(1-\eta)} \\ \leq \;& n\ln(x^k)^T s^k - \sum_{j=1}^{n} \ln(x_j^k s_j^k) + \frac{\eta^2}{2(1-\eta)} \; . \end{aligned}$$

iii) According to the third of (3.30), we have

$$\sqrt{n}(\ln(x^k)^T s^{k+1} - \ln(x^k)^T s^k) = \sqrt{n}\ln\frac{\mu}{\mu^k} \leq -\frac{\eta}{2} \; .$$

Adding the two inequalities in ii) and iii), we have

$$\psi_{n+\rho}(x^k, s^{k+1}) \leq \psi_{n+\rho}(x^k, s^k) - \frac{\eta}{2} + \frac{\eta^2}{2(1-\eta)} \; .$$

Thus, by choosing $\eta = .43$ and $\alpha = .3$ we have the desired result.

∎

Theorem 3.9 establishes an important fact: the *primal-dual* potential function can be reduced by a constant no matter where x^k and y^k are. In practice, one can perform the line search to minimize the primal-dual potential function. This results in the following primal-dual potential reduction algorithm.

Algorithm 3.1 *Given a central path point* $(x^0, y^0, s^0) \in \overset{\circ}{\mathcal{F}}$. *Let* $z^0 = \mathbf{b}^T y^0$. *Set* $k := 0$.

While $(s^k)^T x^k \geq \epsilon$ **do**

1. *Compute* y_1 *and* y_2 *from (3.28).*
2. *If there exists* z *such that* $s(z) > 0$, *compute*

$$\bar{z} = \arg\min_z \psi_{n+\rho}(x^k, s(z)),$$

and if $\psi_{n+\rho}(x^k, s(\bar{z})) < \psi_{n+\rho}(x^k, s^k)$ *then* $y^{k+1} = y(\bar{z})$, $s^{k+1} = s(\bar{z})$ *and* $z^{k+1} = \mathbf{b}^T y^{k+1}$; *otherwise,* $y^{k+1} = y^k$, $s^{k+1} = s^k$ *and* $z^{k+1} = z^k$.

3. Let $x^{k+1} = x^k - \bar{\alpha} X^k p(z^{k+1})$ *with*

$$\bar{\alpha} = \arg\min_{\alpha \geq 0} \psi_{n+\rho}(x^k - \alpha X^k p(z^{k+1}), s^{k+1}).$$

4. Let $k := k + 1$ *and return to Step 1.*

The performance of the algorithm results from the following corollary:

Corollary 3.10 *Let* $\rho = \sqrt{n}$*. Then, Algorithm 3.1 terminates in at most* $O(\sqrt{n} \ln(c^T x^0 - \mathbf{b}^T y^0)/\epsilon)$ *iterations with*

$$c^T x^k - \mathbf{b}^T y^k \leq \epsilon.$$

Proof. In $O(\sqrt{n} \ln((x^0)^T s^0/\epsilon))$ iterations

$$\begin{aligned} -\sqrt{n} \ln((x^0)^T s^0/\epsilon) &= \psi_{n+\rho}(x^k, s^k) - \psi_{n+\rho}(x^0, s^0) \\ &\geq \sqrt{n} \ln(x^k)^T s^k + n \ln n - \psi_{n+\rho}(x^0, s^0) \\ &= \sqrt{n} \ln((x^k)^T s^k/(x^0)^T s^0). \end{aligned}$$

Thus,

$$\sqrt{n} \ln(c^T x^k - \mathbf{b}^T y^k) = \sqrt{n} \ln(x^k)^T s^k \leq \sqrt{n} \ln \epsilon,$$

i.e.,

$$c^T x^k - \mathbf{b}^T y^k = (x^k)^T s^k \leq \epsilon.$$

∎

$$\begin{aligned} P^k &= \mathcal{P}_{\mathcal{A}'}(X^k)^{.5} \nabla \mathcal{P}(X^k, z^k)(X^k)^{.5} \\ &= (X^k)^{.5} \nabla \mathcal{P}(X^k, z^k)(X^k)^{.5} - \mathcal{A}'^T y^k \end{aligned}$$

or

$$P^k = \frac{n + \rho}{S^k \bullet X^k} (X^k)^{.5} (C - \mathcal{A}^T y^k)(X^k)^{.5} - I,$$

and

$$y^k = \frac{S^k \bullet X^k}{n + \rho} (\mathcal{A}' \mathcal{A}'^T)^{-1} \mathcal{A}' (X^k)^{.5} \nabla \mathcal{P}(X^k, z^k)(X^k)^{.5}.$$

Here, $\mathcal{P}_{\mathcal{A}'}$ is the projection operator onto the null space of $\mathcal{A}'$, and

$$\mathcal{A}' \mathcal{A}'^T := \begin{pmatrix} A'_1 \bullet A'_1 & A'_1 \bullet A'_2 & ... & A'_1 \bullet A'_m \\ A'_2 \bullet A'_1 & A'_2 \bullet A'_2 & ... & A'_2 \bullet A'_m \\ ... & ... & ... & ... \\ A'_m \bullet A'_1 & A'_m \bullet A'_2 & ... & A'_m \bullet A'_m \end{pmatrix} \in \mathcal{M}^m.$$

In view of Corollary **??** and

$$\begin{aligned}\nabla\mathcal{P}(X^k,z^k)\bullet(X^{k+1}-X^k) &= -\alpha\frac{\nabla\mathcal{P}(X^k,z^k)\bullet(X^k)^{.5}P^k(X^k)^{.5}}{\|P^k\|}\\ &= -\alpha\frac{(X^k)^{.5}\nabla\mathcal{P}(X^k,z^k)(X^k)^{.5}\bullet P^k}{\|P^k\|}\\ &= -\alpha\frac{\|P^k\|^2}{\|P^k\|}=-\alpha\|P^k\|,\end{aligned}$$

we have

$$\mathcal{P}(X^{k+1},z^k)-\mathcal{P}(X^k,z^k)\le-\alpha\|P^k\|+\frac{\alpha^2}{2(1-\alpha)}\ .$$

Thus, as long as $\|P^k\|\ge\beta>0$, we may choose an appropriate α such that

$$\mathcal{P}(X^{k+1},z^k)-\mathcal{P}(X^k,z^k)\le-\delta$$

for some positive constant δ.

Now, we focus on the expression of P^k, which can be rewritten as

$$P(z^k):=P^k=\frac{n+\rho}{S^k\bullet X^k}(X^k)^{.5}S(z^k)(X^k)^{.5}-I \tag{3.32}$$

with

$$S(z^k)=C-\mathcal{A}^Ty(z^k) \tag{3.33}$$

and

$$y(z^k):=y^k=y_2-\frac{S^k\bullet X^k}{n+\rho}y_1=y_2-\frac{C\bullet X^k-z^k}{n+\rho}y_1\ , \tag{3.34}$$

where y_1 and y_2 are given by

$$\begin{array}{rcl}y_1 &=& (\mathcal{A}'\mathcal{A}'^T)^{-1}\mathcal{A}'I=(\mathcal{A}'\mathcal{A}'^T)^{-1}b,\\ y_2 &=& (\mathcal{A}'\mathcal{A}'^T)^{-1}\mathcal{A}'(X^k)^{.5}C(X^k)^{.5}.\end{array} \tag{3.35}$$

Regarding $\|P^k\|=\|P(z^k)\|$, we have the following lemma resembling Lemma 3.8.

Lemma 3.11 *Let*

$$\mu^k=\frac{S^k\bullet X^k}{n}=\frac{C\bullet X^k-z^k}{n}\quad\textit{and}\quad\mu=\frac{S(z^k)\bullet X^k}{n}\ .$$

If

$$\|P(z^k)\|<\min\left(\beta\sqrt{\frac{n}{n+\beta^2}}\ ,1-\beta\right), \tag{3.36}$$

then the following three inequalities hold:

$$S(z^k)\succ 0,\quad\|(X^k)^{.5}S(z^k)(X^k)^{.5}-\mu I\|<\beta\mu,\quad\textit{and}\quad\mu<(1-.5\beta/\sqrt{n})\mu^k. \tag{3.37}$$

Proof. The proof is by contradiction. For example, if the first inequality of (3.37) is not true, then $(X^k)^{.5}S(z^k)(X^k)^{.5}$ has at least one eigenvalue less than or equal to zero, and

$$\|P(z^k)\| \geq 1.$$

The proof of the second and third inequalities are similar to that of Lemma 3.8. ∎

Based on this lemma, we have the following potential reduction theorem.

Theorem 3.12 *Given $X^k \in \overset{\circ}{\mathcal{F}}_p$ and $(y^k, S^k) \in \overset{\circ}{\mathcal{F}}_d$, let $\rho = \sqrt{n}$, $z^k = \mathbf{b}^T y^k$, X^{k+1} be given by (**??**), and $y^{k+1} = y(z^k)$ in (3.34) and $S^{k+1} = S(z^k)$ in (3.33). Then, either*

$$\psi(X^{k+1}, S^k) \leq \psi(X^k, S^k) - \delta$$

or

$$\psi(X^k, S^{k+1}) \leq \psi(X^k, S^k) - \delta,$$

where $\delta > 1/20$.

Proof. If (3.36) does not hold, i.e.,

$$\|P(z^k)\| \geq \min\left(\beta\sqrt{\frac{n}{n+\beta^2}}\ , 1-\beta\right),$$

then, since $\psi(X^{k+1}, S^k) - \psi(X^k, S^k) = \mathcal{P}(X^{k+1}, z^k) - \mathcal{P}(X^k, z^k)$,

$$\psi(X^{k+1}, S^k) - \psi(X^k, S^k) \leq -\alpha \min\left(\beta\sqrt{\frac{n}{n+\beta^2}}\ , 1-\beta\right) + \frac{\alpha^2}{2(1-\alpha)}\ .$$

Otherwise, from Lemma 3.11 the inequalities of (3.37) hold:

i) The first of (3.37) indicates that y^{k+1} and S^{k+1} are in $\overset{\circ}{\mathcal{F}}_d$.

ii) Using the second of (3.37) and applying Lemma **??** to matrix $(X^k)^{.5}S^{k+1}(X^k)^{.5}/\mu$, we have

$$\begin{aligned}
& n\ln S^{k+1} \bullet X^k - \ln\det S^{k+1}X^k \\
&= n\ln S^{k+1} \bullet X^k/\mu - \ln\det(X^k)^{.5}S^{k+1}(X^k)^{.5}/\mu \\
&= n\ln n - \ln\det(X^k)^{.5}S^{k+1}(X^k)^{.5}/\mu \\
&\leq n\ln n + \frac{\|(X^k)^{.5}S^{k+1}(X^k)^{.5}/\mu - I\|^2}{2(1-\|(X^k)^{.5}S^{k+1}(X^k)^{.5}/\mu - I\|_\infty)} \\
&\leq n\ln n + \frac{\beta^2}{2(1-\beta)} \\
&\leq n\ln S^k \bullet X^k - \ln\det S^k X^k + \frac{\beta^2}{2(1-\beta)}\ .
\end{aligned}$$

iii) According to the third of (3.37), we have

$$\sqrt{n}(\ln S^{k+1} \bullet X^k - \ln S^k \bullet X^k) = \sqrt{n} \ln \frac{\mu}{\mu^k} \leq -\frac{\beta}{2} .$$

Adding the two inequalities in ii) and iii), we have

$$\psi(X^k, S^{k+1}) \leq \psi(X^k, S^k) - \frac{\beta}{2} + \frac{\beta^2}{2(1-\beta)} .$$

Thus, by choosing $\beta = .43$ and $\alpha = .3$ we have the desired result.

∎

Theorem 3.12 establishes an important fact: the *primal-dual* potential function can be reduced by a constant no matter where X^k and y^k are. In practice, one can perform the line search to minimize the primal-dual potential function. This results in the following potential reduction algorithm.

Algorithm 3.2 *Given* $x^0 \in \overset{\circ}{\mathcal{F}}_p$ *and* $(y^0, s^0) \in \overset{\circ}{\mathcal{F}}_d$. *Let* $z^0 = \mathbf{b}^T y^0$. *Set* $k := 0$.
While $S^k \bullet X^k \geq \epsilon$ **do**

1. *Compute* y_1 *and* y_2 *from (3.35).*
2. *Set* $y^{k+1} = y(\bar{z})$, $S^{k+1} = S(\bar{z})$, $z^{k+1} = \mathbf{b}^T y^{k+1}$ *with*
$$\bar{z} = \arg\min_{z \geq z^k} \psi(X^k, S(z)).$$
If $\psi(X^k, S^{k+1}) > \psi(X^k, S^k)$ *then* $y^{k+1} = y^k$, $S^{k+1} = S^k$, $z^{k+1} = z^k$.
3. *Let* $X^{k+1} = X^k - \bar{\alpha}(X^k)^{.5} P(z^{k+1})(X^k)^{.5}$ *with*
$$\bar{\alpha} = \arg\min_{\alpha \geq 0} \psi(X^k - \alpha(X^k)^{.5} P(z^{k+1})(X^k)^{.5}, S^{k+1}).$$
4. *Let* $k := k + 1$ *and return to Step 1.*

The performance of the algorithm results from the following corollary:

Corollary 3.13 *Let* $\rho = \sqrt{n}$. *Then, Algorithm 3.2 terminates in at most* $O(\sqrt{n} \ln(C \bullet X^0 - \mathbf{b}^T y^0)/\epsilon)$ *iterations with*

$$C \bullet X^k - \mathbf{b}^T y^k \leq \epsilon.$$

Proof. In $O(\sqrt{n} \ln(S^0 \bullet X^0/\epsilon))$ iterations

$$\begin{aligned} -\sqrt{n} \ln(S^0 \bullet X^0/\epsilon) &= \psi(X^k, S^k) - \psi(X^0, S^0) \\ &\geq \sqrt{n} \ln S^k \bullet X^k + n \ln n - \psi(X^0, S^0) \\ &= \sqrt{n} \ln(S^k \bullet X^k / S^0 \bullet X^0). \end{aligned}$$

Thus,

$$\sqrt{n} \ln(C \bullet X^k - \mathbf{b}^T y^k) = \sqrt{n} \ln S^k \bullet X^k \leq \sqrt{n} \ln \epsilon,$$

i.e.,

$$C \bullet X^k - \mathbf{b}^T y^k = S^k \bullet X^k \leq \epsilon.$$

∎

3.3 Primal-Dual (Symmetric) Algorithm for LP and SDP

Another technique for solving linear programs is the symmetric primal-dual algorithm. Once we have a pair $(x, y, s) \in \overset{\circ}{\mathcal{F}}$ with $\mu = x^T s/n$, we can generate a new iterate x^+ and (y^+, s^+) by solving for d_x, d_y and d_s from the system of linear equations:

$$\begin{array}{rcl} Sd_x + Xd_s & = & \gamma\mu e - Xs, \\ Ad_x & = & 0, \\ -A^T d_y - d_s & = & 0. \end{array} \tag{3.38}$$

Let $d := (d_x, d_y, d_s)$. To show the dependence of d on the current pair (x, s) and the parameter γ, we write $d = d(x, s, \gamma)$. Note that $d_x^T d_s = -d_x^T A^T d_y = 0$ here.

The system (3.38) is the Newton step starting from (x, s) which helps to find the point on the central path with duality gap $\gamma n\mu$, see Section **??**. If $\gamma = 0$, it steps toward the optimal solution characterized by the system of equations (1.2); if $\gamma = 1$, it steps toward the central path point $(X(\mu), \mathbf{y}(\mu), S(\mu))$ characterized by the system of equations (**??**); if $0 < \gamma < 1$, it steps toward a central path point with a smaller complementarity gap. In the algorithm presented in this section, we choose $\gamma = n/(n + \rho) < 1$. Each iterate reduces the primal-dual potential function by at least a constant δ, as does the previous potential reduction algorithm.

To analyze this algorithm, we present the following lemma, whose proof is omitted.

Lemma 3.14 *Let the direction $d = (d_x, d_y, d_s)$ be generated by equation (3.38) with $\gamma = n/(n + \rho)$, and let*

$$\theta = \frac{\alpha\sqrt{\min(Xs)}}{\|(XS)^{-1/2}(\frac{x^T s}{(n+\rho)}e - Xs)\|} \ , \tag{3.39}$$

where α is a positive constant less than 1. Let

$$x^+ = x + \theta d_x, \quad y^+ = y + \theta d_y, \quad \text{and} \quad s^+ = s + \theta d_s.$$

Then, we have $(x^+, y^+, s^+) \in \overset{\circ}{\mathcal{F}}$ and

$$\psi_{n+\rho}(x^+, s^+) - \psi_{n+\rho}(x, s)$$

$$\leq -\alpha\sqrt{\min(Xs)}\|(XS)^{-1/2}(e - \frac{(n+\rho)}{x^T s}Xs)\| + \frac{\alpha^2}{2(1-\alpha)} \ .$$

Let $v = Xs$. Then, we can prove the following lemma (Exercise 3.5):

Lemma 3.15 *Let $v \in \mathcal{R}^n$ be a positive vector and $\rho \geq \sqrt{n}$. Then,*

$$\sqrt{\min(v)}\|V^{-1/2}(e - \frac{(n+\rho)}{e^T v}v)\| \geq \sqrt{3/4} \ .$$

Combining these two lemmas we have

$$\psi_{n+\rho}(x^+, s^+) - \psi_{n+\rho}(x, s)$$

$$\leq -\alpha\sqrt{3/4} + \frac{\alpha^2}{2(1-\alpha)} = -\delta$$

for a constant δ. This result will provide a competitive theoretical iteration bound, but a faster algorithm may be again implemented by conducting a line search along direction d to achieve the greatest reduction in the primal-dual potential function. This leads to

Algorithm 3.3 *Given* $(x^0, y^0, s^0) \in \overset{\circ}{\mathcal{F}}$. *Set* $\rho \geq \sqrt{n}$ *and* $k := 0$.
While $(s^k)^T x^k \geq \epsilon$ **do**

1. *Set* $(x, s) = (x^k, s^k)$ *and* $\gamma = n/(n+\rho)$ *and compute* (d_x, d_y, d_s) *from (3.38).*
2. *Let* $x^{k+1} = x^k + \bar{\alpha} d_x$, $y^{k+1} = y^k + \bar{\alpha} d_y$, *and* $s^{k+1} = s^k + \bar{\alpha} d_s$ *where*

$$\bar{\alpha} = \arg\min_{\alpha \geq 0} \psi_{n+\rho}(x^k + \alpha d_x, s^k + \alpha d_s).$$

3. *Let* $k := k + 1$ *and return to Step 1.*

Theorem 3.16 *Let* $\rho = O(\sqrt{n})$. *Then, Algorithm 3.3 terminates in at most* $O(\sqrt{n}\ln((x^0)^T s^0/\epsilon))$ *iterations with*

$$c^T x^k - \mathbf{b}^T y^k \leq \epsilon.$$

Once we have a pair $(X, y, S) \in \overset{\circ}{\mathcal{F}}$ with $\mu = S \bullet X/n$, we can apply the primal-dual Newton method to generate a new iterate X^+ and (y^+, S^+) as follows: Solve for d_X, d_y and d_S from the system of linear equations:

$$\begin{array}{rcl} D^{-1} d_X D^{-1} + d_S & = & R := \gamma\mu X^{-1} - S, \\ \mathcal{A} d_X & = & 0, \\ -\mathcal{A}^T d_y - d_S & = & 0, \end{array} \tag{3.40}$$

where

$$D = X^{.5}(X^{.5} S X^{.5})^{-.5} X^{.5}.$$

Note that $d_S \bullet d_X = 0$.

This system can be written as

$$\begin{array}{rcl} d_{X'} + d_{S'} & = & R', \\ \mathcal{A}' d_{X'} & = & 0, \\ -\mathcal{A}'^T d_y - d_{S'} & = & 0, \end{array} \tag{3.41}$$

where

$$d_{X'} = D^{-.5} d_X D^{-.5}, \quad d_{S'} = D^{.5} d_S D^{.5}, \quad R' = D^{.5}(\gamma\mu X^{-1} - S) D^{.5},$$

and

$$\mathcal{A}' = \begin{pmatrix} A_1' \\ A_2' \\ ... \\ A_m' \end{pmatrix} := \begin{pmatrix} D^{.5}A_1D^{.5} \\ D^{.5}A_2D^{.5} \\ ... \\ D^{.5}A_mD^{.5} \end{pmatrix}.$$

Again, we have $d_{S'} \bullet d_{X'} = 0$, and

$$d_y = (\mathcal{A}'\mathcal{A}'^T)^{-1}\mathcal{A}'R', \; d_{S'} = -\mathcal{A}'^T d_y, \text{ and } d_{X'} = R' - d_{S'}.$$

Then, assign

$$d_S = \mathcal{A}^T d_y \quad \text{and} \quad d_X = D(R - d_S)D.$$

Let

$$V^{1/2} = D^{-.5}XD^{-.5} = D^{.5}SD^{.5} \in \overset{\circ}{\mathcal{M}}{}^n_+ .$$

Then, we can verify that $S \bullet X = I \bullet V$. We now present the following lemma, whose proof is very similar to that for LP and 3.14 and will be omitted.

Lemma 3.17 *Let the direction d_X, d_y and d_S be generated by equation (3.40) with $\gamma = n/(n+\rho)$, and let*

$$\theta = \frac{\alpha}{\|V^{-1/2}\|_\infty \|\frac{I \bullet V}{n+\rho} V^{-1/2} - V^{1/2}\|} , \tag{3.42}$$

where α is a positive constant less than 1. Let

$$X^+ = X + \theta d_X, \quad y^+ = y + \theta d_y, \quad \textit{and} \quad S^+ = S + \theta d_S.$$

Then, we have $(X^+, y^+, S^+) \in \overset{\circ}{\mathcal{F}}$ and

$$\psi(X^+, S^+) - \psi(X, S)$$

$$\leq -\alpha \frac{\|V^{-1/2} - \frac{n+\rho}{I \bullet V} V^{1/2}\|}{\|V^{-1/2}\|_\infty} + \frac{\alpha^2}{2(1-\alpha)} .$$

Applying Lemma 3.15 to $v \in \mathcal{R}^n$ as the vector of the n eigenvalues of V, we can prove the following lemma:

Lemma 3.18 *Let $V \in \overset{\circ}{\mathcal{M}}{}^n_+$ and $\rho \geq \sqrt{n}$. Then,*

$$\frac{\|V^{-1/2} - \frac{n+\rho}{I \bullet V} V^{1/2}\|}{\|V^{-1/2}\|_\infty} \geq \sqrt{3/4}.$$

From these two lemmas we have

$$\psi(X^+, S^+) - \psi(X, S)$$

$$\leq -\alpha\sqrt{3/4} + \frac{\alpha^2}{2(1-\alpha)} = -\delta$$

for a constant δ. This leads to Algorithm 3.4.

Algorithm 3.4 *Given* $(X^0, y^0, S^0) \in \overset{\circ}{\mathcal{F}}$. *Set* $\rho = \sqrt{n}$ *and* $k := 0$.
While $S^k \bullet X^k \geq \epsilon$ **do**

1. *Set* $(X, S) = (X^k, S^k)$ *and* $\gamma = n/(n + \rho)$ *and compute* (d_X, d_y, d_S) *from (3.40).*
2. *Let* $X^{k+1} = X^k + \bar{\alpha} d_X$, $y^{k+1} = y^k + \bar{\alpha} d_y$, *and* $S^{k+1} = S^k + \bar{\alpha} d_S$, *where*

$$\bar{\alpha} = \arg\min_{\alpha \geq 0} \psi(X^k + \alpha d_X, S^k + \alpha d_S).$$

3. *Let* $k := k + 1$ *and return to Step 1.*

Theorem 3.19 *Let* $\rho = \sqrt{n}$. *Then, Algorithm 3.4 terminates in at most* $O(\sqrt{n} \ln(S^0 \bullet X^0/\epsilon))$ *iterations with*

$$C \bullet X^k - \mathbf{b}^T y^k \leq \epsilon.$$

3.4 Dual Algorithm for SDP

An open question is how to exploit the sparsity structure by polynomial interior-point algorithms so that they can also solve large-scale problems in practice. In this paper we try to respond to this question. We show that many large-scale semidefinite programs arisen from combinatorial and quadratic optimization have features which make the dual-scaling interior-point algorithm the most suitable choice:

1. The computational cost of each iteration in the dual algorithm is less that the cost the primal-dual iterations. Although primal-dual algorithms may possess superlinear convergence, the approximation problems under consideration require less accuracy than some other applications. Therefore, the superlinear convergence exhibited by primal-dual algorithms may not be utilized in our applications. The dual-scaling algorithm has been shown to perform equally well when only a lower precision answer is required.

2. In most combinatorial applications, we need only a lower bound for the optimal objective value of (SDP). Solving (SDD) alone would be sufficient to provide such a lower bound. Thus, we may not need to generate an X at all. Even if an optimal primal solution is necessary, our dual-scaling algorithm can generate an optimal X at the termination of the algorithm with little additional cost.

3. For large scale problems, S tends to be very sparse and structured since it is the linear combination of C and the A_i's. This sparsity allows considerable savings in both memory and computation time. The primal matrix, X, may be much less sparse and have a structure unknown beforehand. Consequently, primal and primal-dual algorithms may not fully exploit the sparseness and structure of the data.

These problems include the semidefinite relaxations of the graph-partition problem, the box-constrained quadratic optimization problem, the $0-1$ integer set covering problem, etc.

The dual-scaling algorithm, which is a modification of the dual-scaling linear programming algorithm, reduces the Tanabe-Todd-Ye primal-dual potential function

$$\Psi(X,S) = \rho \ln(X \bullet S) - \ln\det X - \ln\det S.$$

The first term decreases the duality gap, while the second and third terms keep X and S in the interior of the positive semidefinite matrix cone. When $\rho > n$, the infimum of the potential function occurs at an optimal solution. Also note that, using the arithmetic-geometric mean inequality, we have

$$n \ln(X \bullet S) - \ln\det X - \ln\det S \geq n \ln n.$$

Let operator $\mathcal{A}(X) : \mathcal{S}^n \rightarrow \Re^m$ be defined as

$$\mathcal{A}(X) = \begin{pmatrix} A_1 \bullet X \\ A_2 \bullet X \\ \vdots \\ A_m \bullet X \end{pmatrix}.$$

Since $\mathcal{A}(X)^T y = \sum_{i=1}^m y_i (A_i \bullet X) = (\sum_{i=1}^m y_i A_i) \bullet X$, the adjoint operator $\mathcal{A}^T : \Re^m \rightarrow \mathcal{S}^n$ is

$$\mathcal{A}^T(y) = \sum_{i=1}^m y_i A_i.$$

Let $\bar{z} = C \bullet X$ for some feasible X and consider the dual potential function

$$\psi(y,\bar{z}) = \rho \ln(\bar{z} - \mathbf{b}^T y) - \ln\det S.$$

Its gradient is

$$\nabla\psi(y,\bar{z}) = -\frac{\rho}{\bar{z} - \mathbf{b}^T y} b + \mathcal{A}(S^{-1}). \tag{3.43}$$

For any given y and $S = C - \mathcal{A}^T(y)$ such that $S \succ 0$ and

$$\|(S^k)^{-.5} \left(\mathcal{A}^T(y - y^k)\right) (S^k)^{-.5}\| < 1,$$

using the above lemma, the concavity of the first term in the potential function, and the fact that

$$(S^k)^{-.5} S (S^k)^{-.5} - I = (S^k)^{-.5}(S - S^k)(S^k)^{-.5} = (S^k)^{-.5} \left(\mathcal{A}^T(y - y^k)\right) (S^k)^{-.5},$$

we establish an overestimater for the potential reduction:

$$\begin{aligned}
&\psi(y,\bar{z}^k)-\psi(y^k,\bar{z}^k)\\
&\quad=\rho\ln(\bar{z}^k-\mathbf{b}^Ty)-\rho\ln(\bar{z}^k-\mathbf{b}^Ty^k)-\ln\det((S^k)^{-.5}S(S^k)^{-.5})\\
&\quad\le\rho\ln(\bar{z}^k-\mathbf{b}^Ty)-\rho\ln(\bar{z}^k-\mathbf{b}^Ty^k)+\mathrm{trace}((S^k)^{-.5}S(S^k)^{-.5}-I)\\
&\qquad+\frac{\|(S^k)^{-.5}\left(\mathcal{A}^T(y-y^k)\right)(S^k)^{-.5}\|}{2(1-\|(S^k)^{-.5}(\mathcal{A}^T(y-y^k))(S^k)^{-.5}\|_\infty)}\\
&\quad=\rho\ln(\bar{z}^k-\mathbf{b}^Ty)-\rho\ln(\bar{z}^k-\mathbf{b}^Ty^k)+\mathcal{A}((S^k)^{-1})^T(y-y^k)\\
&\qquad+\frac{\|(S^k)^{-.5}\left(\mathcal{A}^T(y-y^k)\right)(S^k)^{-.5}\|}{2(1-\|(S^k)^{-.5}(\mathcal{A}^T(y-y^k))(S^k)^{-.5}\|_\infty)}\\
&\quad\le\nabla\psi(y^k,\bar{z}^k)^T(y-y^k)+\frac{\|(S^k)^{-.5}\left(\mathcal{A}^T(y-y^k)\right)(S^k)^{-.5}\|}{2(1-\|(S^k)^{-.5}(\mathcal{A}^T(y-y^k))(S^k)^{-.5}\|_\infty)}.
\end{aligned}\tag{3.44}$$

Therefore, beginning with a strictly feasible dual point (y^k, S^k) and upper bound $\bar{z}^k$, each iteration solves this following problem.

$$\begin{array}{ll}\text{Minimize} & \nabla\psi^T(y^k,\bar{z}^k)(y-y^k)\\ \text{Subject to} & \|(S^k)^{-.5}\left(\mathcal{A}^T(y-y^k)\right)(S^k)^{-.5}\|\le\alpha,\end{array}\tag{3.45}$$

where α is a positive constant less than 1. For simplicity, in what follows we let

$$\Delta^k=\bar{z}^k-\mathbf{b}^Ty^k.$$

The first order Karusch-Kuhn-Tucker conditions state that the minimum point, y^{k+1}, of this convex problem satisfies

$$M^k(y^{k+1}-y^k)+\beta\nabla\psi(y^k,\bar{z}^k)=M^k(y^{k+1}-y^k)+\beta(-\frac{\rho}{\bar{z}^k-\mathbf{b}^Ty^k}b+\mathcal{A}((S^k)^{-1}))=0\tag{3.46}$$

for a positive value of β, where

$$M^k=\begin{pmatrix}A_1(S^k)^{-1}\bullet(S^k)^{-1}A_1 & \cdots & A_1(S^k)^{-1}\bullet(S^k)^{-1}A_m\\ \vdots & \ddots & \vdots\\ A_m(S^k)^{-1}\bullet(S^k)^{-1}A_1 & \cdots & A_m(S^k)^{-1}\bullet(S^k)^{-1}A_m\end{pmatrix}$$

and

$$\mathcal{A}((S^k)^{-1})=\begin{pmatrix}A_1\bullet(S^k)^{-1}\\ \vdots\\ A_m\bullet(S^k)^{-1}\end{pmatrix}.$$

The matrix M^k is a Gram matrix and is positive definite when $S^k\succ 0$ and the A_i's are linearly independent. In this paper, it will sometimes be referred to as M.

Using the ellipsoidal constraint, the minimal solution, y^{k+1}, of (3.45) is given by

$$y^{k+1}-y^k=\frac{\alpha}{\sqrt{\nabla\psi^T(y^k,\bar{z}^k)(M^k)^{-1}\nabla\psi(y^k,\bar{z}^k)}}d(\bar{z}^k)_y\tag{3.47}$$

where

$$d(\bar{z}^k)_y=-(M^k)^{-1}\nabla\psi(y^k,\bar{z}^k).\tag{3.48}$$

Unlike linear programming, semidefinite programs require a significant amount of the time to compute the system of equations used to to determine the step direction. For arbitrary symmetric matrices A_i, the AHO direction can be computed in $5nm^3 + n^2m^2 + O(\max\{m,n\}^3)$ operations, the HRVW/KSH/M direction uses $2nm^3 + n^2m^2 + O(\max\{m,n\}^3)$ operations, and the NT direction uses $nm^3 + n^2m^2/2 + O(\max\{m,n\}^3)$ operations. The complexity of computing the matrix is a full order of magnitude higher than any other step of the algorithm. Fujisawa, Kojima and Nakata explored another technique for computing primal-dual step directions that exploit the sparsity of the data matrices. However, it is our belief that only the dual-scaling algorithm can fully exploit the structure and sparsity of many problems, as explained below.

Generally, $M_{ij}^k = A_i(S^k)^{-1} \bullet (S^k)^{-1}A_j$. When $A_i = a_i a_i^T$, the Gram matrix can be rewritten in the form

$$M^k = \begin{pmatrix} (a_1^T(S^k)^{-1}a_1)^2 & \cdots & (a_1^T(S^k)^{-1}a_m)^2 \\ \vdots & \ddots & \vdots \\ (a_m^T(S^k)^{-1}a_1)^2 & \cdots & (a_m^T(S^k)^{-1}a_m)^2 \end{pmatrix} \tag{3.49}$$

and

$$\mathcal{A}((S^k)^{-1}) = \begin{pmatrix} a_1^T(S^k)^{-1}a_1 \\ \vdots \\ a_m^T(S^k)^{-1}a_m \end{pmatrix}.$$

This matrix can be computed very quickly without computing, or saving, $(S^k)^{-1}$. Instead, S^k can be factored, and then we can use

Algorithm M: To compute M^k and $\mathcal{A}((S^k)^{-1})$, factor $S^k = L^k(L^k)^T$ and do the following:

For $i = 1 : m$;
  Solve $L^k w_i = a_i$;
  $\mathcal{A}((S^k)^{-1})_i = w_i^T w_i$ and $M^k_{ii} = (\mathcal{A}((S^k)^{-1})_i)^2$;
  For $j = 1 : m-1$; $M^k_{ij} = (w_i^T w_j)^2$; end;
end.

Solving each of the m systems of equations uses $n^2 + O(n)$ floating point operations. Since there are $m(m+1)/2$ vector multiplications, Algorithm M, uses $nm^2 + n^2 m + O(nm)$ operations after factoring S^k. Note that these operations can be significantly reduced if S^k is structured and sparse. In applications like the maximum cut problem, the matrix S^k is indeed very sparse while its inverse is usually dense, so working with S^k is faster than working with its inverse. Using matrices of the form $A_i = a_i a_i^T$ also reduces the complexity of primal-dual algorithms by a factor of n, but even the quickest direction to compute takes about twice as long as our dual-scaling direction. Furthermore, they all need to handle dense X.

Algorithm M needs to store all vectors $w_1, ..., w_m$ and they are generally dense. To save storage and exploit the sparsity of $a_i, ..., a_m$, an alternative algorithm is

Algorithm M': To compute M^k and $\mathcal{A}((S^k)^{-1})$, factor $S^k = L^k(L^k)^T$ and do the following:

For $i = 1 : m$;
  Solve $S^k w_i = a_i$;
  $\mathcal{A}((S^k)^{-1})_i = w_i^T a_i$ and $M^k_{ii} = (\mathcal{A}((S^k)^{-1})_i)^2$;
  For $j = i+1 : m$; $M^k_{ij} = (w_i^T a_j)^2$; end;
end.

Algorithm M' does not need to store w_j but uses one more back-solver for w_i.

To find a feasible primal point X, we solve the least squares problem

$$\begin{array}{ll} \text{Minimize} & \|(S^k)^{.5} X (S^k)^{.5} - \frac{\Delta^k}{\rho} I\| \\ \text{Subject to} & \mathcal{A}(X) = b. \end{array} \tag{3.50}$$

This problem looks for a matrix $X(\bar{z}^k)$ near the central path. Larger values of ρ generally give a lower objective value, but provide a solution matrix that is not positive definite more frequently. The answer to (3.50) is a by-product of computing (3.48), given explicitly by

$$X(\bar{z}^k) = \frac{\Delta^k}{\rho}(S^k)^{-1}\left(\mathcal{A}^T(d(\bar{z}^k)_y) + S^k\right)(S^k)^{-1}. \tag{3.51}$$

Creating the primal matrix may be costly. However, the evaluation of the primal objective value $C \bullet X(\bar{z}^k)$ requires drastically less work.

$$\begin{aligned} C \bullet X(\bar{z}^k) &= \mathbf{b}^T y^k + X(\bar{z}^k) \bullet S^k \\ &= \mathbf{b}^T y^k + \text{trace}\left(\tfrac{\Delta^k}{\rho}(S^k)^{-1}\left(\mathcal{A}^T(d(\bar{z}^k)_y) + S^k\right)(S^k)^{-1}S^k\right) \\ &= \mathbf{b}^T y^k + \tfrac{\Delta^k}{\rho}\text{trace}\left((S^k)^{-1}\mathcal{A}^T(d(\bar{z}^k)_y) + I\right) \\ &= \mathbf{b}^T y^k + \tfrac{\Delta^k}{\rho}\left(d(\bar{z}^k)_y^T\mathcal{A}((S^k)^{-1}) + n\right) \end{aligned}$$

Since the vectors $\mathcal{A}((S^k)^{-1})$ and $d(\bar{z}^k)_y$ were previously found in calculating the dual step direction, the cost of computing a primal objective value is the cost of a vector dot product! The matrix $X(\bar{z}^k)$ never gets computed during the iterative process, saving time and memory. On the other hand, primal-dual methods require far more resources to compute the primal variables X.

Defining

$$P(\bar{z}^k) = \frac{\rho}{\Delta^k}(S^k)^{.5}X(\bar{z}^k)(S^k)^{.5} - I, \tag{3.52}$$

we have the following lemma:

Lemma 3.20 *Let* $\mu^k = \frac{\Delta^k}{n} = \frac{\bar{z}^k - \mathbf{b}^T y^k}{n}$, $\mu = \frac{X(\bar{z}^k)\bullet S^k}{n} = \frac{C\bullet X(\bar{z}^k) - \mathbf{b}^T y^k}{n}$, $\rho \geq n + \sqrt{n}$, *and* $\alpha < 1$. *If*

$$\|P(\bar{z}^k)\| < min(\alpha\sqrt{\frac{n}{n+\alpha^2}}, 1-\alpha), \tag{3.53}$$

then the following three inequalities hold:

1. $X(\bar{z}^k) \succ 0$;
2. $\|(S^k)^{.5}X(\bar{z}^k)(S^k)^{.5} - \mu I\| \leq \alpha\mu$;
3. $\mu \leq (1 - .5\alpha/\sqrt{n})\mu^k$.

Proof. The proofs are by contradiction. If the first inequality is false, then $(S^k)^{.5}X(\bar{z}^k)(S^k)^{.5}$ has at least one nonpositive eigenvalue, which by (3.52) implies that $\|P(\bar{z}^k)\| \geq 1$.

If the second does not hold, then

$$\begin{aligned} \|P(\bar{z}^k)\|^2 &= \|\tfrac{\rho}{n\mu^k}(S^k)^{.5}X(\bar{z}^k)(S^k)^{.5} - I\|^2 \\ &= \|\tfrac{\rho}{n\mu^k}(S^k)^{.5}X(\bar{z}^k)(S^k)^{.5} - \tfrac{\rho\mu}{n\mu^k}I + \tfrac{\rho\mu}{n\mu^k}I - I\|^2 \\ &= \|\tfrac{\rho}{n\mu^k}(S^k)^{.5}X(\bar{z}^k)(S^k)^{.5} - \tfrac{\rho\mu}{n\mu^k}I\|^2 + \|\tfrac{\rho\mu}{n\mu^k}I - I\|^2 \\ &> \left(\tfrac{\rho\mu}{n\mu^k}\right)^2\alpha^2 + \left(\tfrac{\rho\mu}{n\mu^k} - 1\right)^2 n \\ &\geq \alpha^2\left(\tfrac{n}{n+\alpha^2}\right) \end{aligned}$$

where the last inequality is true because the quadratic term has a minimum at $\frac{\rho\mu}{n\mu^k} = \frac{n}{n+\alpha^2}$.

If the third inequality does not hold, then

$$\frac{\rho\mu}{n\mu^k} > \left(1+\frac{1}{\sqrt{n}}\right)\left(1-\frac{.5\alpha}{\sqrt{n}}\right) \geq 1.$$

which leads to

$$\begin{aligned}
\|P(\bar{z}^k)\|^2 &\geq \left(\frac{\rho\mu}{n\mu^k}-1\right)^2 n \\
&\geq \left(\left(1+\frac{1}{\sqrt{n}}\right)\left(1-\frac{\alpha}{2\sqrt{n}}\right)-1\right)^2 n \\
&= \left(1-\frac{\alpha}{2}-\frac{\alpha}{2\sqrt{n}}\right)^2 \\
&\geq (1-\alpha)^2.
\end{aligned}$$

∎

Focusing on the expression $P(\bar{z}^k)$, it can be rewritten as

$$\begin{aligned}
P(\bar{z}^k) &= \tfrac{\rho}{\Delta^k}(S^k)^{.5}\left(\tfrac{\Delta^k}{\rho}(S^k)^{-1}\left(\mathcal{A}^T(d(\bar{z}^k)_y)+S^k\right)(S^k)^{-1}\right)(S^k)^{.5}-I \\
&= (S^k)^{-.5}\mathcal{A}^T\left(d(\bar{z}^k)_y\right)(S^k)^{-.5} \\
&= (S^k)^{-.5}\mathcal{A}^T\left(\tfrac{y^{k+1}-y^k}{\beta}\right)(S^k)^{-.5}
\end{aligned}$$

which by (3.45), makes

$$\nabla\psi^T(y^k,\bar{z}^k)d(\bar{z}^k)_y = -\|P(\bar{z}^k)\|^2 \tag{3.54}$$

and

$$\nabla\psi^T(y^k,\bar{z}^k)(y^{k+1}-y^k) = -\alpha\|P(\bar{z}^k)\|. \tag{3.55}$$

Updating the dual variables according to

$$y^{k+1} = y^k + \frac{\alpha}{\|P(\bar{z}^{k+1})\|}d(\bar{z})_y \qquad \text{and} \qquad S^{k+1} = C-\mathcal{A}^T(y^{k+1}), \tag{3.56}$$

assures the positive definiteness of S^{k+1} when $\alpha < 1$, which assures that they are feasible. Using (3.55) and (3.44), the reduction in the potential function satisfies the inequality

$$\psi(y^{k+1},\bar{z}^k)-\psi(y^k,\bar{z}^k) \leq -\alpha\|P(\bar{z}^k)\| + \frac{\alpha^2}{2(1-\alpha)}. \tag{3.57}$$

The theoretical algorithm can be stated as follows.

DUAL ALGORITHM. Given an upper bound $\bar{z}^0$ and a dual point (y^0, S^0) such that $S^0 = C-\mathcal{A}^T y^0 \succ 0$, set $k=0$, $\rho > n+\sqrt{n}$, $\alpha \in (0,1)$, and do the following:

while $\bar{z}^k - \mathbf{b}^T y^k \geq \epsilon$ **do**

begin

1. Compute $\mathcal{A}((S^k)^{-1})$ and the Gram matrix M^k (3.49) using Algorithm M or M'.
2. Solve (3.48) for the dual step direction $d(\bar{z}^k)_y$.
3. Calculate $\|P(\bar{z}^k)\|$ using (3.54).
4. **If** (3.53) is true, **then** $X^{k+1} = X(\bar{z}^k)$, $\bar{z}^{k+1} = C \bullet X^{k+1}$, and $(y^{k+1}, S^{k+1}) = (y^k, S^k)$;
   else $y^{k+1} = y^k + \frac{\alpha}{\|P(\bar{z}^k)\|} d(\bar{z}^{k+1})_y$, $S^{k+1} = C - \mathcal{A}^T(y^{k+1})$, $X^{k+1} = X^k$, and $\bar{z}^{k+1} = \bar{z}^k$.
   endif
5. $k := k + 1$.

end

We can derive the following potential reduction theorem based on the above lemma:

Theorem 3.21

$$\Psi(X^{k+1}, S^{k+1}) \leq \Psi(X^k, S^k) - \delta$$

where $\delta > 1/50$ *for a suitable* α.

Proof.

$$\Psi(X^{k+1}, S^{k+1}) - \Psi(X^k, S^k) = \left(\Psi(X^{k+1}, S^{k+1}) - \Psi(X^{k+1}, S^k)\right) + \left(\Psi(X^{k+1}, S^k) - \Psi(X^k, S^k)\right).$$

In each iteration, one of the differences is zero. If $\|P(\bar{z}^k)\|$ does not satisfy (3.53), the dual variables get updated and (3.57) shows sufficient improvement in the potential function when $\alpha = 0.4$.

On the other hand, if the primal matrix gets updated, then using Lemma **??** and the first two parts of Lemma 3.20,

$$\begin{aligned}
&n \ln \left(X^{k+1} \bullet S^k\right) - \ln \det \left(X^{k+1}\right) - \ln \det \left(S^k\right) \\
&= n \ln \left(X^{k+1} \bullet S^k\right) - \ln \det \left(X^{k+1} S^k\right) \\
&= n \ln \left(X^{k+1} \bullet S^k/\mu\right) - \ln \det \left(X^{k+1} S^k/\mu\right) \\
&= n \ln n - \ln \det \left((S^k)^{.5} X^{k+1} (S^k)^{.5}/\mu\right) \\
&\leq n \ln n + \frac{\|(S^k)^{.5} X^{k+1} (S^k)^{.5}/\mu - I\|}{2(1 - \|(S^k)^{.5} X^{k+1} (S^k)^{.5}/\mu - I\|_\infty)} \\
&\leq n \ln n + \frac{\alpha^2}{2(1-\alpha)} \\
&\leq n \ln \left(X^k \bullet S^k\right) - \ln \det \left(X^k\right) - \ln \det \left(S^k\right) + \frac{\alpha^2}{2(1-\alpha)}
\end{aligned}$$

Additionally, by the third part of Lemma 3.20

$$\sqrt{n} \left(\ln(X^{k+1} \bullet S^k) - \ln(X^k \bullet S^k)\right) = \sqrt{n} \ln \frac{\mu}{\mu^k} \leq -\frac{\alpha}{2}$$

Adding the two inequalities gives

$$\Psi(X^{k+1}, S^k) \leq \Psi(X^k, S^k) - \frac{\alpha}{2} + \frac{\alpha^2}{2(1-\alpha)}$$

By choosing $\alpha = 0.4$ again, we have the desired result. ∎

This theorem leads to

Corollary 3.22 *Let $\rho \geq n + \sqrt{n}$ and $\Psi(X^0, S^0) \leq (\rho - n)\ln(X^0 \bullet S^0)$. Then, the algorithm terminates in at most $O((\rho - n)\ln(X^0 \bullet S^0/\epsilon))$ iterations.*

Proof. In $O((\rho - n)\ln(X^0 \bullet S^0/\epsilon))$ iterations,

$$\Psi(X^k, S^k) \leq (\rho - n)\ln(\epsilon)).$$

Also,

$$(\rho - n)\ln(C \bullet X^k - \mathbf{b}^T y^k) = (\rho - n)\ln(X^k \bullet S^k) \leq \Psi(X^k, S^k) - n\ln n \leq \Psi(X^k, S^k).$$

Combining the two inequalities,

$$C \bullet X^k - \mathbf{b}^T y^k = X^k \bullet S^k < \epsilon.$$

∎

Again, from (3.51) we see that the algorithm can generate an X^k as a by-product. However, it is not needed in generating the iterate direction, and it is only explicitly used for proving convergence and complexity.

Theorem 3.23 *Each iteration of the dual algorithm uses $O(m^3 + nm^2 + n^2m + n^3)$ floating point iterations.*

Proof. Creating S, or $S + \mathcal{A}^T(d(\bar{z}^k))$, uses matrix additions and $O(mn^2)$ operations; factoring it uses $O(n^3)$ operations. Creating the Gram matrix uses $nm^2 + 2n^2m + O(nm)$ operations, and solving the system of equations uses $O(m^3)$ operations. Dot products for $\bar{z}^{k+1}$ and $\|P(\bar{z}^k)\|$, and the calculation of y^{k+1} use only $O(m)$ operations. These give the desired result. ∎

3.5 Initialization

A pair of SDP has *three alternatives*:

$$\begin{array}{lrl} \text{(Solvable)} & \mathcal{A}X - b &= 0 \\ & -\mathcal{A}^T y + C &\succeq 0, \\ & \mathbf{b}^T y - C \bullet X &= 0, \\ & y \text{ free}, \ X &\succeq 0 \end{array} \quad \text{or} \quad \begin{array}{lrl} \text{(Infeasible)} & \mathcal{A}X &= 0 \\ & -\mathcal{A}^T y &\succeq 0, \\ & \mathbf{b}^T y - C \bullet X &> 0, \\ & y \text{ free}, \ X &\succeq 0 \end{array}$$

or neither one has a finite solution. As we discussed earlier, the third case does not exists under some mild conditions

Now consider an integrated homogeneous system:

$$\begin{array}{lrl} (HSDP) & \mathcal{A}X - b\tau &= 0 \\ & -\mathcal{A}^T y + C\tau &= S \succeq 0, \\ & \mathbf{b}^T y - C \bullet X &= \kappa \geq 0, \\ & y \text{ free}, \ X \succeq 0, & \tau \geq 0, \end{array}$$

where the *three alternatives* can be classified as

$$\begin{array}{rl} \text{(Solvable)}: & (\tau > 0, \kappa = 0) \\ \text{(Infeasible)}: & (\tau = 0, \kappa > 0) \\ \text{(All others)}: & (\tau = \kappa = 0). \end{array}$$

Such a homogeneous system is self-dual – it dual can be written as:

$$\begin{array}{rll} (HSDD) & \mathcal{A}X' - b\tau' & = 0, \\ & \mathcal{A}^T y' - X\tau' & \succeq 0, \\ & -\mathbf{b}^T y' + C \bullet X' & \leq 0, \\ & y' \text{ free}, \ X' \succeq 0, & \tau' \geq 0, \end{array}$$

We have the following theorem:

Theorem 3.24 *System (HSDP) is feasible (e.g. all zeros) and any feasible solution* (y, X, τ, S, κ) *is* self-complementary*:* $X \bullet S + \tau\kappa = 0$ *or*

$$\begin{pmatrix} X & 0 \\ 0^T & \tau \end{pmatrix} \begin{pmatrix} S & 0 \\ 0^T & \kappa \end{pmatrix} = 0.$$

Furthermore, it has a max-rank *complementary feasible solution, that is,*

$$rank \begin{pmatrix} X & 0 \\ 0^T & \tau \end{pmatrix} + rank \begin{pmatrix} S & 0 \\ 0^T & \kappa \end{pmatrix}$$

is maximal.

Let's find such a feasible solution starting from $X^0 = I \succ 0$, $S^0 = I \succ 0$, and $y^0 = 0$. Then, we formulate

$$\begin{array}{rlccccl} (HSDP) & \min & & & & \theta & \\ & \text{s.t.} & & \mathcal{A}X & -b\tau & +\bar{b}\theta & = 0, \\ & & -\mathcal{A}^T y & & +C\tau & -\bar{C}\theta & \succeq 0, \\ & & \mathbf{b}^T y & -C \bullet X & & +\bar{z}\theta & \geq 0, \\ & & y \text{ free}, & X \succeq 0, & \tau \geq 0, & \theta \text{ free}, & \end{array}$$

where

$$\bar{b} = b - \mathcal{A} \bullet I, \quad \bar{C} = C - I, \quad \bar{z} = C \bullet I + 1.$$

Note that (HSDP) may just give us the *all-zero solution*. To eliminate such a trivial solution, we add one more constraint to the system:

$$\begin{array}{rlccccl} (HSDP) & \min & & & & (n+1)\theta & \\ & \text{s.t.} & & \mathcal{A}X & -b\tau & +\bar{b}\theta & = 0, \\ & & -\mathcal{A}^T y & & +C\tau & -\bar{C}\theta & = S \succeq 0, \\ & & \mathbf{b}^T y & -C \bullet X & & +\bar{z}\theta & = \kappa \geq 0, \\ & & -\bar{\mathbf{b}}^T y & +\bar{C} \bullet X & -\bar{z}\tau & & = -(n+1), \\ & & y \text{ free}, & X \succeq 0, & \tau \geq 0, & \theta \text{ free}. & \end{array}$$

Note that the constraints of (HSDP) form a *skew-symmetric system* and the objective coefficient vector is the negative of the right-hand-side vector, so that it remains a *self-dual* SDP program.

By combining the constraints, we can derive the last (equality) constraint as

$$I \bullet X + I \bullet S + \tau + \kappa - (n+1)\theta = (n+1),$$

which serves indeed as a *normalizing constraint* for (HSDP) to prevent the all-zero solution.

Theorem 3.25 *Consider problems (HSDP).*

i) *(HSDP) has a* strictly *feasible point*

$$y = 0, \quad X = I \succ 0, \quad \tau = 1, \quad \theta = 1, \quad S = I \succ 0, \quad \kappa = 1.$$

ii) *(HSDP) has an optimal solution and its optimal solution set is* bounded.

iii) *The optimal value of (HSDP) is zero, and*

$$(y, X, \tau, \theta, S, \kappa) \in \mathcal{F}_h \quad \textit{implies that} \quad (n+1)\theta = X \bullet S + \tau\kappa.$$

iv) *There is an optimal solution* $(y^*, X^*, \tau^*, \theta^* = 0, S^*, \kappa^*) \in \mathcal{F}_h$ *such that*

$$rank\begin{pmatrix} X^* & 0 \\ 0^T & \tau^* \end{pmatrix} + rank\begin{pmatrix} S^* & 0 \\ 0^T & \kappa^* \end{pmatrix}$$

is maximal.

Finally, we can derive

Theorem 3.26 *Let* $(y^*, X^*, \tau^*, \theta^* = 0, S^*, \kappa^*)$ *be a max-rank solution for (HSDP).*

i) *(SDP) and (SDD) are* (Solvable) *if and only if* $\tau^* > 0$. *In this case,* X^*/τ^* *is an optimal solution for (SDP) and* $(y^*/\tau^*, S^*/\tau^*)$ *is an optimal solution for (SDD).*

ii) *(SDP) or (SDD) is* (Infeasible) *if and only if* $\kappa^* > 0$. *In this case,* X^*/κ^* *or* y^*/κ^* *or both are certificates for proving* infeasibility*: if* $C \bullet X^* < 0$ *then (SDD) is infeasible; if* $-\mathbf{b}^T y^* < 0$ *then (SDP) is infeasible; and if both* $C \bullet X^* < 0$ *and* $-\mathbf{b}^T y^* < 0$ *then both (SDP) and (SDD) are infeasible.*

iii) *(SDP) and/or (SDD) are in the third alternative if and only if* $\tau^* = \kappa^* = 0$.

Thus, one can apply any (feasible) SDP algorithm to solve (HSDP). Since (HSDP) is self-dual, the algorithm does not need to solve the dual part, so that the size of the problem is almost as same as the size of a pair of SDP problems when a pair of feasible solutions is known.

3.6 Notes

The primal potential reduction algorithm for positive semi-definite programming is due to Alizadeh [9, 8], in which Ye has "suggested studying the primal-dual potential function for this problem" and "looking at symmetric preserving scalings of the form $X_0^{-1/2} X X_0^{-1/2}$," and to Nesterov and Nemirovskii [243], and the primal-dual algorithm described here is due to Nesterov and Todd [244, 245]. One can also develop a dual potential reduction algorithm.

Other primal-dual algorithms for positive semi-definite programming are in Alizadeh, Haeberly and Overton [10, 12], Boyd, Ghaoui, Feron and Balakrishnan [58], Helmberg, Rendl, Vanderbei and Wolkowicz [158], Jarre [172], de Klerk, Roos and Terlaky.[187], Kojima, Shindoh and Hara [192], Monteiro and Zhang [230], Nesterov, Todd and Ye [246], Potra and Sheng [259], Shida, Shindoh and Kojima [278], Sturm and Zhang [288], Tseng [308], Vandenberghe and Boyd [316, 317], and references therein. Efficient interior-point algorithms are also developed for optimization over the second-order cone; see Andersen and Christiansen [17], Lobo, Vandenberghe and Boyd [206], and Xue and Ye [329]. These algorithms have established the best approximation complexity results for some combinatorial problems.

Primal-dual adaptive path-following algorithms, the predictor-corrector algorithms and the wide-neighborhood algorithms can also be developed for solving (SDP).

The homogeneous and self-dual initialization model was developed by Ye, Todd and Mizuno for LP [340] and for SDP by Luo, Sturm and Zhang [212] and Nesterov, Todd and Ye [246], and it became the foundation algorithm implemented by Sturm [287] and Andersen and Andersen [14].

3.7 Exercises

3.1 *For SDP, prove a slightly stronger version of iii) in Lemma 3.1: If $D \in \mathcal{S}^n$ such that $0 \leq \|D\|_\infty < 1$ (that is, the largest absolute value of the eigenvalues of D is less than 1), then*

$$-I \bullet D \leq -\ln\det(I+D) \leq -I \bullet D + \frac{\|D\|^2}{2(1-\|D\|_\infty)} \ .$$

3.2 *Let $\mathbf{x}$ be in the interior of $\mathcal{N}_2^n$. Then for any $\mathbf{d} \in \mathcal{R}^n$,*

$$\frac{(d_1 - \mathbf{x}_{-1}^T \mathbf{d}_{-1})^2}{\delta(\mathbf{x})^2} + \|\mathbf{d}_{-1}\|^2 - d_1^2 \geq 0.$$

3.3 *Prove i) of Proposition 3.5.*

3.4 *Given $S \succ 0$, find the minimizer of the least-squares problem*

$$\begin{array}{ll} \textit{minimize} & \|S^{1/2} X S^{1/2} - I\| \\ \textit{s.t.} & \mathcal{A}X = 0. \end{array}$$

Given $X \succ 0$ find the minimizer of the least-squares problem

$$\begin{array}{cc} \textit{minimize} & \|X^{1/2}SX^{1/2} - I\| \\ \textit{s.t.} & S = C - \mathcal{A}^T y. \end{array}$$

3.5 *Let $v \in \mathcal{R}^n$ be a positive vector and $\rho \geq \sqrt{n}$. Prove*

$$\sqrt{\min(v)}\|V^{-1/2}(e - \frac{(n+\rho)}{e^T v}v)\| \geq \sqrt{3/4}\ .$$

3.6 *Prove the following convex quadratic inequality*

$$(Ay+b)^T(Ay+b) - c^T y - d \leq 0$$

is equivalent to a matrix inequality

$$\begin{pmatrix} I & Ay+b \\ (Ay+b)^T & c^T y + d \end{pmatrix} \succeq 0.$$

Using this relation to formulate a convex quadratic minimization problem with convex quadratic inequalities as an (SDD) problem.

3.7 *Prove Corollary* **??**.

3.8 *Prove Lemma 3.17.*

3.9 *Describe and analyze a dual potential algorithm for positive semi-definite programming in the standard form.*

3.10 *Prove Theorem 3.24.*

3.11 *Prove Theorem 3.25.*

3.12 *Prove Theorem 3.26.*

3.13 *Let (LP) and (LD) have interior. Prove the dual potential function $\mathcal{B}_{n+1}(y, s, z)$, where z is a upper bound of z^*, represents the volume of a coordinate-aligned ellipsoid whose intersection with the affine set $\{x : \ Ax = b\}$ contains the primal level set $\{x \in \mathcal{F}_p : \ c^T x \leq z\}$.*

3.14 *Let $X, S \in \mathcal{M}^n$ be both positive definite. Then prove*

$$\psi_n(X, S) = n\ln(X \bullet S) - \ln(\det(X) \cdot \det(S)) \geq n \ln n.$$

3.15 *Consider linear programming and the level set*

$$\Psi(\delta) := \{(X, y, S) \in \overset{\circ}{\mathcal{F}}: \ \psi_{n+\rho}(x, s) \leq \delta\}.$$

Prove that

$$\Psi(\delta^1) \subset \Psi(\delta^2) \quad \textit{if} \quad \delta^1 \leq \delta^2,$$

and for every δ $\Psi(\delta)$ is bounded and its closure $\hat{\Psi}(\delta)$ has non-empty intersection with the solution set.

3.16 *Prove (ii) of Theorem 3.3.*

3.17 *Prove Theorem* **??**.

3.18 *Prove Corollary* **??**. *Here we assume that* $X(\mu) \neq X(\mu')$ *and* $\mathbf{y}(\mu) \neq y(mu')$.

Chapter 4

SDP for Global Quadratic and Combinatorial Optimization

4.1 Approximation

A (randomized) algorithm for a maximization problem is called (randomized) r-approximation algorithm, where $0 < r \leq 1$, if it outputs a feasible solution with its (expected) value at least r times the optimum value for all instances of the problem.

More precisely, let $w^*(> 0)$ be the (global) maximum value of a given problem instance. Then, a r-approximate maximizer x satisfies

$$w(x) \geq r \cdot w^*.$$

or

$$\mathrm{E}[w(x)] \geq r \cdot w^*.$$

A (randomized) algorithm for a minimization problem is called (randomized) r-approximation algorithm, where $1 \leq r$, if it outputs a feasible solution with its (expected) value at most r times the optimum value for all instances of the problem.

More precisely, let $w^*(> 0)$ be the (global) minimal value of a given problem instance. Then, a r-approximate minimizer x satisfies

$$w(x) \leq r \cdot w^*.$$

or

$$\mathrm{E}[w(x)] \leq r \cdot w^*.$$

4.2 Ball-Constrained Quadratic Minimization

Consider a ball constrained quadratic minimization problem

$$\text{(BQP)} \qquad \begin{array}{rll} z^* := & \text{Minimize} & x^T Q x + 2q^T x \\ & \text{Subject to} & \|x\|^2 \leq 1. \end{array} \tag{4.1}$$

Here, the given matrix $Q \in \mathcal{M}^n$, the set of n-dimensional symmetric matrices; vector $q \in R^n$; and $\|.\|$ is the Euclidean norm.

4.2.1 Homogeneous Case: $q = 0$

Matrix-formulation: Let

$$X = xx^T.$$

$$\text{(BQP)} \qquad \begin{array}{rll} z^* = & \text{Minimize} & Q \bullet X \\ & \text{Subject to} & I \bullet X \leq 1, \\ & & X \succeq 0, \quad \text{Rank}(X) = 1. \end{array}$$

SDP-relaxation: Remove the rank-one constraint.

$$\text{(SDP)} \qquad \begin{array}{rll} z^{SDP} := & \text{Minimize} & Q \bullet X \\ & \text{Subject to} & I \bullet X \leq 1, \\ & & X \succeq 0. \end{array}$$

The dual of (SDP) can be written as:

$$\text{(DSDP)} \qquad \begin{array}{rll} z^{SDP} = & \text{Maximize} & y \\ & \text{Subject to} & yI + S = Q, \\ & & y \leq 0, \quad S \succeq 0. \end{array}$$

From the SDP duality theorem, this pair of SDP problems have an optimal solution pair and there is no duality gap. That is, X^* is a minimal matrix solution to SDP if and only if there exist a feasible dual variable $y^* \leq 0$ such that

$$S^* = Q - y^* I \succeq 0$$
$$y^*(1 - I \bullet X^*) = 0$$
$$S^* \bullet X^* = 0.$$

Obviously, $z^{SDP} \leq z^*$. On the other hand, from the SDP rank theorem, the SDP relaxation has an optimal solution X^* of rank one, which implies

Theorem 4.1 *The SDP relaxation is exact, meaning*

$$z^{SDP} = z^*.$$

Moreover, an optimal solution for (BQP) can be computed in strongly polynomial time from any optimal SDP solution of the relaxation.

4.2.2 Non-Homogeneous Case

Similarly, let

$$X = (1; x)(1; x)^T \in \mathcal{M}^{n+1},$$

$$Q' = \begin{pmatrix} 0 & q^T \\ q & Q \end{pmatrix},$$

$$I' = \begin{pmatrix} 0 & 0^T \\ 0 & I \end{pmatrix},$$

and

$$I_1 = \begin{pmatrix} 1 & 0^T \\ 0 & 0 \end{pmatrix}.$$

$$\text{(BQP)} \qquad \begin{array}{lll} z^* = & \text{Minimize} & Q' \bullet X \\ & \text{Subject to} & I' \bullet X \leq 1, \\ & & I_1 \bullet X = 1, \\ & & X \succeq 0, \quad \text{Rank}(X) = 1. \end{array}$$

SDP-relaxation: Remove the rank-one constraint.

$$\text{(SDP)} \qquad \begin{array}{lll} z^* = & \text{Minimize} & Q' \bullet X \\ & \text{Subject to} & I' \bullet X \leq 1, \\ & & I_1 \bullet X = 1, \\ & & X \succeq 0. \end{array}$$

The dual of (SDP) can be written as:

$$\text{(DSDP)} \qquad \begin{array}{lll} z^{SDP} = & \text{Maximize} & y_1 + y_2 \\ & \text{Subject to} & y_1 I' + y_2 I_1 + S = Q', \\ & & y_1 \leq 0, \quad S \succeq 0. \end{array}$$

Again, X^* is a minimal matrix solution to SDP if and only if there exist a feasible dual variables $(y_1^* \leq 0, y_2^*)$ such that

$$S^* = Q' - y_1^* I - y_2^* I_1 \succeq 0$$

$$y_1^*(1 - I' \bullet X^*) = 0$$

$$S^* \bullet X^* = 0.$$

Again, $z^{SDP} \leq z^*$, and we also have

Theorem 4.2 *The SDP relaxation is exact, meaning*

$$z^{SDP} = z^*.$$

Moreover, an optimal solution for (BQP) can be computed in strongly polynomial time from any optimal SDP solution of the relaxation.

Corollary 4.3 *Consider the quadratic minimization problem*

$$\begin{array}{lll} z^* := & \textit{Minimize} & x^T Q x + 2q^T x \\ \textit{(BQP)} & & \\ & \textit{Subject to} & \|x\|^2 = 1. \end{array} \tag{4.2}$$

Then, its SDP relaxation is exact. Moreover, an optimal solution for the problem can be computed in strongly polynomial time from any optimal SDP solution of the relaxation.

4.3 Quadratically Constrained Quadratic Problems (QCQP)

These quadratic minimization problems can be generalized to multiple quadratically constrained quadratic problems.

$$\begin{array}{lll} z^* := & \text{Minimize} & x^T Q x \\ \text{(QP-2)} & & \\ & \text{Subject to} & x^T A_1 x (\leq, =, \geq) 1, \\ & & \ldots, \\ & & x^T A_m x (\leq, =, \geq) 1, \end{array} \tag{4.3}$$

where m is a positive integer.

One application of the problem is the binary least squares problem:

$$\begin{array}{lll} z^* := & \text{Minimize} & \|Ax - c\|^2 \\ \text{(BLS)} & & \\ & \text{Subject to} & x_j^2 = 1, \ \forall i = 1, ..., n. \end{array}$$

The problem can be formulated as a homogeneous problem:

$$\begin{array}{lll} z^* := & \text{Minimize} & \|Ax - x_{n+1} c\|^2 \\ \text{(BLS)} & & \\ & \text{Subject to} & x_j^2 = 1, \ \forall i = 1, ..., n, n+1. \end{array}$$

The matrix-formulation of the QCQP problem would be

$$\text{(QP-2)} \qquad \begin{array}{lll} z^* = & \text{Minimize} & Q \bullet X \\ & \text{Subject to} & A_1 \bullet X (\leq, =, \geq) 1, \\ & & ..., \\ & & A_m \bullet X (\leq, =, \geq) 1, \\ & & X \succeq 0, \quad \text{Rank}(X) = 1. \end{array}$$

The SDP-relaxation is to remove the rank-one constraint:

$$\text{(SDP)} \qquad \begin{array}{lll} z^* = & \text{Minimize} & Q \bullet X \\ & \text{Subject to} & A_1 \bullet X (\leq, =, \geq) 1, \\ & & ..., \\ & & A_m \bullet X (\leq, =, \geq) 1, \\ & & X \succeq 0. \end{array}$$

We now consider several special cases of QCQP.

4.3.1 Multiple Ellipsoid-Constrained Quadratic Maximization

Consider

$$\text{(ECQP-m)} \quad z^* := \begin{array}{ll} \text{Maximize} & x^T Q x \\ \text{subject to} & x^T A_i x \leq 1, \; i = 1, ..., m. \end{array}$$

where Q and A_i, $i = 1, \ldots, m$, are positive semidefinite, and $\sum_i A_i$ is positive definite.

The SDP relaxation is

$$(SDP) \quad z^{SDP} := \begin{array}{ll} \text{Maximize} & Q \bullet X \\ \text{subject to} & A_i \bullet X \leq b_i, \; i = 1, ..., m \\ & X \succeq 0; \end{array}$$

and its dual is

$$(SDD) \quad z^{SDD} := \begin{array}{ll} \text{Minimize} & \sum_{i=1}^m b_i y_i \\ \text{subject to} & S = \sum_{i=1}^m y_i A_i - Q \\ & S \succeq 0, \; y_i \geq 0, \; i = 1, ..., m. \end{array}$$

Obviously, both (SDP) and (SDD) are feasible and each has an interior feasible region. Thus (SDP) and (SDD) have no duality gap, that is, $z^{SDP} = z^{SDD}$. Using the SDP rank theorem and the null-space reduction, we have

Theorem 4.4 *For solving (ECQP-m), one can find a feasible solution x such that*

$$x^T Q x \geq \frac{1}{\bar{r}} \cdot z^{SDP}$$

where $\bar{r}$ is the largest integer r such that $r(r+1) \leq 2m$.

On the other hand, we can use the randomized rank-reduction as follows. Let X^* be an optimal matrix solution of the SDP relaxation, that is,

$$\begin{array}{ll} A_i \bullet X^* & \leq b_i,\ \forall i = 1, \ldots, m \\ Q \bullet X^* & \geq z^{SDP} \geq z^* \end{array}$$

We like to find a rank-1 matrix solution $\hat{X} = xx^T$ such that

$$\begin{array}{ll} A_i \bullet \hat{X} & \leq \alpha \cdot b_i,\ \forall i = 1, \ldots, m \\ Q \bullet \hat{X} & \geq \beta \cdot z^* \end{array}$$

Recall that for any positive semidefinite matrix H, we can compute an random rank-1 $\hat{X}$ from X^* such that

$$\Pr\left(H \bullet \hat{X} \leq \beta \mathrm{Tr}(H)\right) \leq \exp\left(\frac{1}{2}(1 - \beta + \ln \beta)\right)$$

and

$$\Pr\left(H \bullet \hat{X} \geq \alpha \mathrm{Tr}(H)\right) \leq \exp\left(\frac{1}{2}(1 - \alpha + \ln \alpha)\right).$$

Let $\beta = \frac{1}{10}$ and $\alpha = 4\ln(4m)$. Then,

$$\begin{array}{ll} \Pr\left(A_i \bullet \hat{X} \geq \alpha \cdot b_i\right) & \leq \frac{1}{4m},\ i = 1, \ldots, m \\ \Pr\left(Q \bullet \hat{X} \leq \beta \cdot z^*\right) & \leq \frac{1}{2}. \end{array}$$

Therefore,

$$\Pr\left(Q \bullet \hat{X} \geq \beta \cdot z^* \text{ and } A_i \bullet \hat{X} \leq \alpha \cdot b_i\ \forall i = 1, \ldots, m\right) \geq 1 - \frac{1}{2} - \frac{1}{4} = \frac{1}{4}.$$

Or

$$\Pr\left(Q \bullet \hat{X}/\alpha \geq (\beta/\alpha) \cdot z^* \quad \text{and} \quad A_i \bullet \hat{X}/\alpha \leq b_i\ \forall i = 1, \ldots, m\right) \geq \frac{1}{4}.$$

Theorem 4.5 *With a positive probability, one can compute a* $\hat{X}/\alpha = (x/\sqrt{\alpha})(x/\sqrt{\alpha})^T = \hat{x}\hat{x}^T$ *where* $\hat{x}$ *is feasible for (ECQP-m) and, with high probability,*

$$\hat{x}^T Q \hat{x} \geq (\beta/\alpha) \cdot z^* = \frac{1}{40\ln(4m)} \cdot z^*.$$

4.3.2 Binary Quadratic Maximization

$$\begin{array}{llll} \text{(BQP)} & z^* := & \text{Maximize} & x^T Q x \\ & & \text{subject to} & (x_j)^2 = 1,\ j = 1, ..., n. \end{array}$$

where Q is assumed to be positive semidefinite. Note that even if the original Q is not psd, when can still formulate an equivalent problem by $Q := Q + |\lambda| \cdot I \succeq \mathbf{0}$ where λ is the minimal eigenvalue of Q.

The following is the pair of the SDP relaxation:

$$\begin{array}{rrll} z^{SDP} := & \text{Maimize} & Q \bullet X \\ & \text{s.t.} & I_j \bullet X = 1, \ j = 1, ..., n, \\ & & X \succeq \mathbf{0}. \end{array}$$

and

$$\begin{array}{rrll} z^{SDP} = & \text{Minimize} & e^T y \\ & \text{s.t.} & \text{Diag}(y) \succeq Q. \end{array}$$

Let X^* be an optimal matrix solution of the SDP relaxation, and let random vector

$$u \in N(0, X^*) \quad \text{and} \quad \hat{x} = \text{sign}(u)$$

where

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

Clearly, $\hat{x}$ is binary, and we have

$$\text{E}[\hat{x}^T Q \hat{x}] = Q \bullet \text{E}[\hat{x}\hat{x}^T] = Q \bullet \frac{2}{\pi} \arcsin[\bar{X}].$$

For any function of one variable $f(t)$ and $X \in \mathcal{R}^{n \times n}$, let $f[X] \in \mathcal{R}^{n \times n}$ be the matrix with the components $f(x_{ij})$. We have the next technical lemma whose proof is an exercise.

Lemma 4.6 *Let $X \succeq 0$ and $d(X) \leq 1$. Then* $\arcsin[X] \succeq X$.

Then, the following theorem holds.

Theorem 4.7 *For solving (BQP), we have an approximation ratio $\frac{2}{\pi}$, that is, one can find a random feasible solution $\hat{x}$ such that*

$$\text{E}[\hat{x}^T Q \hat{x}] \geq \frac{2}{\pi} z^{SDP} \geq \frac{2}{\pi} z^*.$$

4.3.3 Box Constrained Optimization

In this section we consider a more general QCQP:

$$\begin{array}{rrll} \underline{q}(Q) := & \text{minimize} & q(x) := x^T Q x \\ & \text{s.t.} & \sum_{j=1}^n a_{ij} x_j^2 = b_i, \ i = 1, \ldots, m, \\ & & -e \leq x \leq e, \end{array}$$

where symmetric matrix $Q \in \mathcal{M}^n$, $A = \{a_{ij}\} \in \mathcal{R}^{m \times n}$ and $b \in \mathcal{R}^m$ are given and $e \in \mathcal{R}^n$ is, again, the vector of all ones. We assume that the problem is feasible and denote by $\underline{x}(Q)$ its global minimizer.

The function $q(x)$ has a *global maximizer* over the bounded feasible set as well. Let $\bar{q} := -\underline{q}(-Q)$ and $\underline{q} := \underline{q}(Q)$ denote their maximal and minimal

objective values, respectively. We now present a "fast" algorithm to compute a 4/7-approximate minimizer: to compute a feasible $\hat{x}$ such that

$$\frac{q(\hat{x}) - \underline{q}}{\bar{q} - \underline{q}} \leq 4/7.$$

This new criterion is necessary since Q could be indefinite, and the global minimal value is 0.

The SDP relaxation would be:

$$\begin{array}{rlll} \underline{p}(Q) := & \text{minimize} & Q \bullet X & \\ & \text{s.t.} & A_i \bullet X = b_i, \ i = 1, \ldots, m, & \\ & & d(X) \leq e, \ X \succeq 0. & \end{array} \tag{4.4}$$

Here, $A_i = \text{diag}(a_i)$, $a_i = (a_{i1}, \ldots, a_{in})$, and unknown $X \in \mathcal{M}^n$ is a symmetric matrix. Furthermore, $d(X)$ is a vector containing the diagonal components of X. Note that $d(X) \leq e$ can be written as $I_j \bullet X \leq 1$, $j = 1, \ldots, n$, where I_j is the all-zero matrix except the jth diagonal component equal to 1.

The dual of the relaxation is

$$\begin{array}{rlll} \underline{p}(Q) = & \text{maximize} & e^T z + b^T y & \\ & \text{s.t.} & Q \succeq D(z) + \sum_{i=1}^m y_i A_i, \ z \leq 0, & \end{array} \tag{4.5}$$

where $D(z)$ is the diagonal matrix such that $d(D(z)) = z \in \mathcal{R}^n$. Note that the relaxation is feasible and its dual has an interior feasible point so that there is no duality gap between the primal and dual. Denote by $\underline{X}(Q)$ and $(\underline{y}(Q), \underline{z}(Q))$ an optimal solution pair for the primal (4.4) and dual (4.5). For simplicity, in what follows we let $\underline{x} = \underline{x}(Q)$ and $\underline{X} = \underline{X}(Q)$.

We have the following relations between the QP problem and its relaxation:

Proposition 4.8 *Let* $\underline{q} := \underline{q}(Q)$, $\bar{q} := -\underline{q}(-Q)$, $\underline{p} := \underline{p}(Q)$, $\bar{p} := -\underline{p}(-Q)$, $(\bar{y}, \bar{z}) = (-\underline{y}(-Q), -\underline{z}(-Q))$. *Then,*

i) $\bar{q}$ *is the maximal objective value of* $x^T Q x$ *in the feasible set of the QP problem;*

ii) $\bar{p} = e^T \bar{z} + b^T \bar{y}$ *and it is the maximal objective value of* $Q \bullet X$ *in the feasible set of the relaxation, and* $D(\bar{z}) + \sum_{i=1}^m \bar{y}_i A_i - Q \succeq 0$*;*

iii)

$$\underline{p} \leq \underline{q} \leq \bar{q} \leq \bar{p}.$$

Let matrix factorization $\underline{V} = (\underline{v}_1, \ldots, \underline{v}_n) \in \mathcal{R}^{n \times n}$, i.e., $\underline{v}_j$ is the jth column of $\underline{V}$, such that $\underline{X} = \underline{V}^T \underline{V}$. Then, we generate a random vector u uniformly distributed on the unit sphere in $\mathcal{R}^n$ and assign

$$\hat{x} = \underline{D} \sigma(\underline{V}^T u), \tag{4.6}$$

where

$$\underline{D} = \text{diag}(\|\underline{v}_1\|, \ldots, \|\underline{v}_n\|) = \text{diag}(\sqrt{\underline{x}_{11}}, \ldots, \sqrt{\underline{x}_{nn}}),$$

and, for any $x \in \mathcal{R}^n$, $\sigma(x) \in \mathcal{R}^n$ is the vector whose jth component is $\mathrm{sign}(x_j)$:

$$\mathrm{sign}(x_j) = \begin{cases} 1 & \text{if } x_j \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

It is easily see that $\hat{x}$ is a feasible solution for the original QP problem and we will show that its expected objective value, $\mathrm{E}_u q(\hat{x})$, satisfies

$$\frac{\mathrm{E}_u q(\hat{x}) - \underline{q}}{\bar{q} - \underline{q}} \leq \frac{\pi}{2} - 1 \leq \frac{4}{7} \ .$$

That is, $\hat{x}$ is a 4/7-approximate minimizer for the QP problem expectantly. One can generate u repeatedly and choose the best $\hat{x}$ in the process. Thus, we will almost surely generate a $\hat{x}$ that is a 4/7-approximate minimizer.

The proof is based on a lemma below.

Lemma 4.9 *Let u be uniformly distributed on the unit sphere in $\mathcal{R}^n$. Then,*

$$\begin{array}{rccl} \underline{q}(Q) = & \textit{minimize} & \mathrm{E}_u(\sigma(V^T u)^T DQD\sigma(V^T u)) \\ & \textit{s.t.} & A_i \bullet (V^T V) = b_i, \ i = 1, \ldots, m, \\ & & \|v_j\| \leq 1, \ j = 1, \ldots, n, \end{array}$$

where

$$D = \mathrm{diag}(\|v_1\|, \ldots, \|v_n\|).$$

Now we are ready to state the following theorem, where we use "infimum" to replace "minimum," since for simplicity we require X to be positive definite in our subsequent analysis.

Theorem 4.10

$$\begin{array}{rccl} \underline{q}(Q) = & \textit{infimum} & \frac{2}{\pi} Q \bullet (D \arcsin[D^{-1} X D^{-1}] D) \\ & \textit{s.t.} & A_i \bullet X = b_i, \ i = 1, \ldots, m, \\ & & d(X) \leq e, \ X \succ 0, \end{array}$$

where

$$D = \mathrm{diag}(\sqrt{x_{11}}, \ldots, \sqrt{x_{nn}}).$$

Theorem 4.10 and Lemma 4.6 lead to our main result:

Theorem 4.11 *We have*

i) $\bar{p} - \underline{q} \geq \frac{2}{\pi}(\bar{p} - \underline{p})$.

ii) $\bar{q} - \underline{p} \geq \frac{2}{\pi}(\bar{p} - \underline{p})$.

iii) $\bar{p} - \underline{p} \geq \bar{q} - \underline{q} \geq \frac{4-\pi}{\pi}(\bar{p} - \underline{p})$.

Proof. We prove (i). Recall $\bar{z} = -\underline{z}(-Q) \geq 0$, $\bar{y} = -\underline{y}(-Q)$, $\bar{p} = -\underline{p}(-Q) = e^T\bar{z} + b^T\bar{y}$, and $D(\bar{z}) + \sum_{i=1}^m \bar{y}_i A_i - Q \succeq 0$. Thus, for any $X \succ 0$, $d(X) \leq e$ and

$D = \mathrm{diag}(\sqrt{x_{11}}, \ldots, \sqrt{x_{nn}})$, from Theorem 4.10

$$
\begin{aligned}
\frac{\pi}{2}\underline{q} = \frac{\pi}{2}\underline{q}(Q) & \\
\leq\ & Q \bullet (D \arcsin[D^{-1}XD^{-1}]D) \\
=\ & \left(Q - D(\bar{z}) - \sum_{i=1}^{m} \bar{y}_i A_i + D(\bar{z}) + \sum_{i=1}^{m} \bar{y}_i A_i\right) \bullet \left(D \arcsin[D^{-1}XD^{-1}]D\right) \\
=\ & \left(Q - D(\bar{z}) - \sum_{i=1}^{m} \bar{y}_i A_i\right) \bullet \left(D \arcsin[D^{-1}XD^{-1}]D\right) \\
& + \left(D(\bar{z}) + \sum_{i=1}^{m} \bar{y}_i A_i\right) \bullet \left(D \arcsin[D^{-1}XD^{-1}]D\right) \\
\leq\ & \left(Q - D(\bar{z}) - \sum_{i=1}^{m} \bar{y}_i A_i\right) \bullet \left(DD^{-1}XD^{-1}D\right) \\
& + \left(D(\bar{z}) + \sum_{i=1}^{m} \bar{y}_i A_i\right) \bullet \left(D \arcsin[D^{-1}XD^{-1}]D\right) \\
& \text{(since } Q - D(\bar{z}) - \sum_{i=1}^{m} \bar{y}_i A_i \preceq 0 \text{ and } \arcsin[D^{-1}XD^{-1}] \succeq D^{-1}XD^{-1}.\text{)} \\
=\ & \left(Q - D(\bar{z}) - \sum_{i=1}^{m} \bar{y}_i A_i\right) \bullet X \\
& + \left(D(\bar{z}) + \sum_{i=1}^{m} \bar{y}_i A_i\right) \bullet \left(D \arcsin[D^{-1}XD^{-1}]D\right) \\
=\ & Q \bullet X - \left(D(\bar{z}) + \sum_{i=1}^{m} \bar{y}_i A_i\right) \bullet X \\
& + \left(D(\bar{z}) + \sum_{i=1}^{m} \bar{y}_i A_i\right) \bullet \left(D \arcsin[D^{-1}XD^{-1}]D\right) \\
=\ & Q \bullet X - \bar{z}^T d(X) - \sum_{i=1}^{m} \bar{y}_i a_i^T d(X) \\
& + \bar{z}^T d(D \arcsin[D^{-1}XD^{-1}]D) + \sum_{i=1}^{m} \bar{y}_i a_i^T d(D \arcsin[D^{-1}XD^{-1}]D) \\
=\ & Q \bullet X - \bar{z}^T d(X) - \bar{y}^T b + \bar{z}^T (\frac{\pi}{2} d(X)) + \bar{y}^T (\frac{\pi}{2} b) \\
& \text{(since } d(D \arcsin[D^{-1}XD^{-1}]D) = \frac{\pi}{2} d(X) \text{ and } a_i^T d(X) = b_i\text{)} \\
=\ & Q \bullet X + (\frac{\pi}{2} - 1)\bar{z}^T d(X) + (\frac{\pi}{2} - 1)\bar{y}^T b \\
\leq\ & Q \bullet X + (\frac{\pi}{2} - 1)(\bar{z}^T e + \bar{y}^T b) \\
& \text{(since } 0 \leq d(X) \leq e \text{ and } \bar{z} \geq 0\text{)} \\
=\ & Q \bullet X + (\frac{\pi}{2} - 1)\bar{p}.
\end{aligned}
$$

Let $X \succ 0$ converge to $\underline{X}$, then $Q \bullet X \to \underline{p}$ and we prove (i).

Replacing Q with $-Q$ proves (ii) in the theorem.

Adding the first two inequalities gives (iii) of the theorem. ∎

Similarly, the following corollary can be devised:

Corollary 4.12 *Let $X = V^T V \succ 0$, $d(X) \leq e$, $A_i \bullet X = b_i$ $(i = 1, \ldots, m)$, $D = \mathrm{diag}(\sqrt{x_{11}}, \ldots, \sqrt{x_{nn}})$, and $\hat{x} = D\sigma(V^T u)$ where u with $\|u\| = 1$ is a random vector uniformly distributed on the unit sphere. Moreover, let $X \succ 0 \to \underline{X}$. Then,*

$$\lim_{X \to \underline{X}} \mathrm{E}_u(q(\hat{x})) = \lim_{X \to \underline{X}} \frac{2}{\pi} Q \bullet (D \arcsin[D^{-1} X D^{-1}] D) \leq \frac{2}{\pi} \underline{p} + (1 - \frac{2}{\pi}) \bar{p}.$$

Finally, we have the following theorem:

Theorem 4.13 *Let $\hat{x}$ be randomly generated from $\underline{X}$. Then*

$$\frac{\mathrm{E}_u q(\hat{x}) - \underline{q}}{\bar{q} - \underline{q}} \leq \frac{\pi}{2} - 1 < 4/7.$$

Proof. Since

$$\bar{p} \geq \bar{q} \geq \frac{2}{\pi} \bar{p} + (1 - \frac{2}{\pi}) \underline{p} \geq (1 - \frac{2}{\pi}) \bar{p} + \frac{2}{\pi} \underline{p} \geq \underline{q} \geq \underline{p},$$

we have, from Corollary 4.12,

$$\begin{aligned}
\frac{\mathrm{E}_u q(\hat{x}) - \underline{q}}{\bar{q} - \underline{q}} &\leq \frac{\frac{2}{\pi}\underline{p} + (1 - \frac{2}{\pi})\bar{p} - \underline{q}}{\bar{q} - \underline{q}} \\
&\leq \frac{\frac{2}{\pi}\underline{p} + (1 - \frac{2}{\pi})\bar{p} - \underline{q}}{\frac{2}{\pi}\bar{p} + (1 - \frac{2}{\pi})\underline{p} - \underline{q}} \\
&\leq \frac{\frac{2}{\pi}\underline{p} + (1 - \frac{2}{\pi})\bar{p} - \underline{p}}{\frac{2}{\pi}\bar{p} + (1 - \frac{2}{\pi})\underline{p} - \underline{p}} \\
&= \frac{(1 - \frac{2}{\pi})(\bar{p} - \underline{p})}{\frac{2}{\pi}(\bar{p} - \underline{p})} \\
&= \frac{1 - \frac{2}{\pi}}{\frac{2}{\pi}} = \frac{\pi}{2} - 1.
\end{aligned}$$

∎

4.4 Max-Cut Problem

Consider the Max Cut problem on an undirected graph $G = (V, E)$ with non-negative weights w_{ij} for each edge in E (and $w_{ij} = 0$ if $(i, j) \notin E$), which is the problem of partitioning the nodes of V into two sets S and $V \setminus S$ so that

$$w(S) := \sum_{i \in S,\, j \in V \setminus S} w_{ij}$$

is maximized. A problem of this type arises from many network planning, circuit design, and scheduling applications.

This problem can be formulated by assigning each node a binary variable x_j:

$$\text{(MC)} \qquad \begin{array}{rll} z^* = & \text{Maximize} & w(x) := \dfrac{1}{4}\sum_{i,j} w_{ij}(1 - x_i x_j) \\ & \text{Subject to} & x_i^2 = 1, \quad i = 1, ..., n. \end{array}$$

The Coin-Toss Method: Let each node be selected to one side, or x_i be 1, independently with probability .5. Then, swap nodes from the majority side to the minority side using the greedy method. ∎

$$\mathrm{E}[w(x)] \geq 0.5 \cdot z^*.$$

4.4.1 SDP relaxation

$$\begin{array}{rll} z^{SDP} := & \text{minimize} & Q \bullet X \\ & \text{s.t.} & I_j \bullet X = 1, \ j = 1, ..., n, \\ & & X \succeq 0. \end{array} \tag{4.7}$$

The dual is

$$\begin{array}{rll} z^{SDP} = & \text{maximize} & e^T y \\ & \text{s.t.} & Q \succeq D(y). \end{array} \tag{4.8}$$

Let $V = (v_1, \ldots, v_n) \in \mathcal{R}^{n \times n}$, i.e., v_j is the jth column of V, such that $X^* = V^T V$.

Generate a random vector $u \in N(0, I)$:

$$\hat{x} = \mathrm{sign}(V^T u), \tag{4.9}$$

$$\mathrm{sign}(x_j) = \begin{cases} 1 & \text{if } x_j \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

4.4.2 Approximation analysis

Then, one can prove from Sheppard [276] (see Goemans and Williamson [127] and Bertsimas and Ye [52]):

$$\mathrm{E}[\hat{x}_i \hat{x}_j] = \frac{2}{\pi} \arcsin(\bar{X}_{ij}), \quad i, j = 1, 2, \ldots, n.$$

Lemma 4.14 *For $x \in [-1, 1)$*

$$\frac{1 - (2/\pi) \cdot \arcsin(x)}{1 - x} \geq .878.$$

Lemma 4.15 *Let $X \succeq 0$ and $d(X) \leq 1$. Then* $\arcsin[X] \succeq X$.

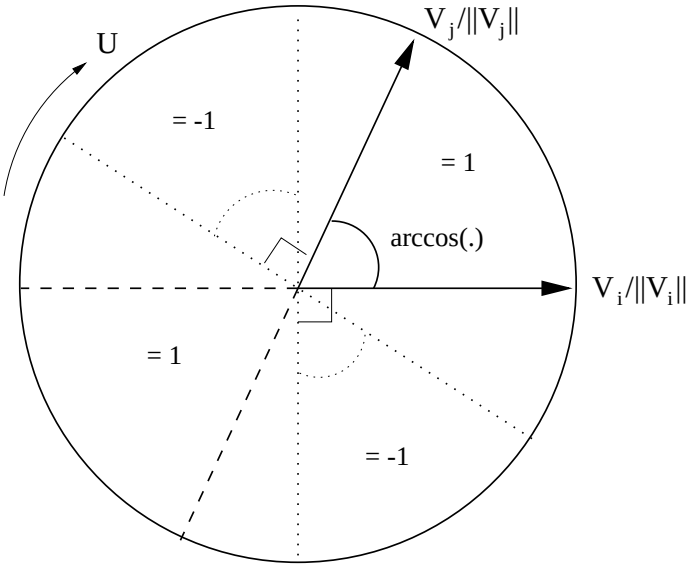


Figure 4.1: Illustration of the product $\sigma(\frac{v_i^T u}{\|v_i\|}) \cdot \sigma(\frac{v_j^T u}{\|v_j\|})$ on the 2-dimensional unit circle. As the unit vector u is uniformly generated along the circle, the product is either 1 or -1.

Theorem 4.16 *We have*

i) *If Q is a Laplacian matrix, then*

$$\mathrm{E}(\hat{x}^T Q \hat{x}) \geq .878 z^{SDP} \geq .878 z^*,$$

so that

$$z^* \geq .878 z^{SDP}.$$

ii) *If Q is positive semidefinite*

$$\mathrm{E}(\hat{x}^T Q \hat{x}) \geq \frac{2}{\pi} z^{SDP} \geq \frac{2}{\pi} z^*,$$

so that

$$z^* \geq \frac{2}{\pi} z^{SDP}.$$

4.5 Max-Bisection Problem

Consider the Max-Bisection problem on an undirected graph $G = (V, E)$ with non-negative weights w_{ij} for each edge in E (and $w_{ij} = 0$ if $(i, j) \notin E$), which is the problem of partitioning the even number of nodes in V into two sets S and $V \setminus S$ of equal cardinality so that

$$w(S) := \sum_{i \in S,\, j \in V \setminus S} w_{ij}$$

is maximized. This problem can be formulated by assigning each node a binary variable x_j:

$$\text{(MB)} \qquad \begin{array}{rll} w^* := & \text{Maximize} & \frac{1}{4} \sum_{i,j} w_{ij}(1 - x_i x_j) \\ & \text{subject to} & \sum_{j=1}^{n} x_j = 0 \quad \text{or} \quad e^T x = 0 \\ & & x_j^2 = 1, \ j = 1, \ldots, n, \end{array}$$

where $e \in \Re^n$ (n even) is the column vector of all ones, superscript ^{T} is the transpose operator. Note that x_j takes either 1 or -1, so that we can choose either $S = \{j : \ x_j = 1\}$ or $S = \{j : \ x_j = -1\}$. The constraint $e^T x = 0$ ensures that $|S| = |V \setminus S|$.

A problem of this type arises from many network planning, circuit design, and scheduling applications. In particular, the popular and widely used Min-Bisection problem sometimes can be solved by finding the Max-Bisection over the complementary graph of G.

The Coin-Toss Method: Let each node be selected to one side, or x_i be 1, independently with probability .5. Then, swap nodes from the majority side to the minority side using the greedy method. ∎

$$\mathrm{E}[w(x)] \geq 0.5 \cdot z^*.$$

4.5.1 SDP relaxation

$$\begin{aligned} \text{Maximize} \quad & \frac{1}{4}\sum_{i,j} w_{ij}(1 - X_{ij}) \\ \text{Subject to} \quad & X_{ii} = 1, \quad i = 1, ..., n, \\ & \textstyle\sum_{i,j} X_{ij} = 0, \\ & X \succeq 0. \end{aligned}$$

What complicates matters in Max-Bisection, comparing to Max-Cut, is that two objectives are actually sought—the objective value of $w(S)$ and the size of S. Therefore, in any (randomized) rounding method at the beginning, we need to balance the (expected) quality of $w(S)$ and the (expected) size of S. We want high $w(S)$; but, at the same time, zero or small difference between $|S|$ and $n/2$, since otherwise we have to either add or subtract nodes from S, resulting in a deterioration of $w(S)$ at the end. Our method is built upon a careful balance of the two, plus an improved proof technique.

4.5.2 The .651-method of Frieze and Jerrum

We first review the Frieze and Jerrum method, and then proceed with our improved method.

Let $\bar{X}$ be an optimal solution of Problem (SDP). Since $\bar{X}$ is positive semidefinite, the randomization method of Goemans and Williamson [127] essentially generates a random vector u from a multivariate normal distribution with 0 mean and covariance matrix $\bar{X}$, that is, using (4.9) to generate $S = \{i : \hat{x}_i = 1\}$ or $S = \{i : \hat{x}_i = -1\}$.

Then, one can prove Sheppard [276] (see Goemans and Williamson [127], Frieze and Jerrum [112], and Bertsimas and Ye [52]):

$$\mathrm{E}[\hat{x}_i\hat{x}_j] = \frac{2}{\pi}\arcsin(\bar{X}_{ij}), \quad i, j = 1, 2, \ldots, n, \tag{4.10}$$

and

$$1 - \frac{2}{\pi}\arcsin(\bar{X}_{ij}) \geq \alpha(1)(1 - \bar{X}_{ij}),$$

where

$$\alpha(1) := \min_{-1 \leq y < 1} \frac{1 - \frac{2}{\pi}\arcsin(y)}{1 - y} \geq .878567$$

which is a special case of Definition (4.22) in Section 5.

Thus,

$$\begin{aligned} \mathrm{E}[w(S)] &= \frac{1}{4}\sum_{i,j} w_{ij}\left(1-\frac{2}{\pi}\arcsin(\bar{X}_{ij})\right) \\ &\geq \frac{1}{4}\sum_{i,j} w_{ij}\cdot\alpha(1)(1-\bar{X}_{ij}) \\ &= \alpha(1)\cdot w^{SD} \\ &\geq \alpha(1)\cdot w^{*}; \end{aligned} \tag{4.11}$$

and

$$\begin{aligned} \mathrm{E}\left[\frac{n^2}{4}-\frac{(e^T\hat{x})^2}{4}\right] &= \frac{1}{4}\sum_{i,j}\left(1-\frac{2}{\pi}\arcsin(\bar{X}_{ij})\right) \\ &\geq \frac{1}{4}\sum_{i,j}\alpha(1)(1-\bar{X}_{ij}) \\ &= \alpha(1)\cdot\frac{n^2}{4}, \end{aligned} \tag{4.12}$$

since

$$\sum_{i,j}\bar{X}_{ij} = ee^T\bullet\bar{X} = 0.$$

However, $\hat{x}$ may not satisfy $e^T\hat{x}=0$, i.e., S may not be a bisection. Then, using a greedy method, Frieze and Jerrum have adjusted S by swapping nodes from the majority block into the minority block until they are equally sized. Note that inequality (4.12) assures that not too many nodes need to be swapped. Also, in selecting swapping nodes, Frieze and Jerrum make sure that the least weighted node gets swapped first. More precisely, (w.l.o.g) let $|S|\geq n/2$; and for each $i\in S$, let $\zeta(i)=\sum_{j\notin S}w_{ij}$ and $S=\{i_1,i_2,...,i_{|S|}\}$, where $\zeta(i_1)\geq\zeta(i_2)\geq ...\geq\zeta(i_{|S|})$. Then, assign $\tilde{S}=\{i_1,i_2,...,i_{n/2}\}$. Clearly, the construction of bisection $\tilde{S}$ guarantees that

$$w(\tilde{S})\geq\frac{n\cdot w(S)}{2|S|},\quad n/2\leq|S|\leq n. \tag{4.13}$$

In order to analyze the quality of bisection $\tilde{S}$, they define two random variables:

$$w := w(S) = \frac{1}{4}\sum_{i,j} w_{ij}(1-\hat{x}_i\hat{x}_j)$$

and

$$m := |S|(n-|S|) = \frac{n^2}{4}-\frac{(e^T\hat{x})^2}{4} = \frac{1}{4}\sum_{i,j}(1-\hat{x}_i\hat{x}_j)$$

(since $(e^T\hat{x})^2=(2|S|-n)^2$). Then, from (4.11) and (4.12),

$$\mathrm{E}[w]\geq\alpha(1)\cdot w^{*}\quad\text{and}\quad\mathrm{E}[m]\geq\alpha(1)\cdot m^{*},\quad\text{where}\quad m^{*}=\frac{n^2}{4}.$$

Thus, if a new random variable

$$z = \frac{w}{w^*} + \frac{m}{m^*}, \tag{4.14}$$

then

$$\mathrm{E}[z] \geq 2\alpha(1), \quad \text{and} \quad z \leq 3$$

since $w/w^* \leq 2|S|/n \leq 2$ and $m/m^* \leq 1$.

For simplicity, Frieze and Jerrum's analysis can be described as follows. They repeatedly generate samples $u \in N(0, \bar{X})$, create $\hat{x}$ or S using the randomization method (4.9), construct $\tilde{S}$ using the swapping procedure, and record the largest value of $w(\tilde{S})$ among these samples. Since $\mathrm{E}[z] \geq 2\alpha(1)$ and $z \leq 3$, they are almost sure to have one z to meet its expectation before too long, i.e., to have

$$z \geq 2\alpha(1). \tag{4.15}$$

Moreover, when (4.15) holds and suppose

$$w(S) = \lambda w^*,$$

which from (4.14) and (4.15) implies that

$$\frac{m}{m^*} \geq 2\alpha(1) - \lambda.$$

Suppose that $|S| = \delta n$; then the above inequality implies

$$\lambda \geq 2\alpha(1) - 4\delta(1-\delta). \tag{4.16}$$

Applying (4.13) and (4.16), one can see that

$$\begin{aligned} w(\tilde{S}) &\geq \frac{w(S)}{2\delta} \\ &= \frac{\lambda w^*}{2\delta} \\ &\geq \frac{(2\alpha(1) - 4\delta(1-\delta))w^*}{2\delta} \\ &\geq 2(\sqrt{2\alpha(1)} - 1)w^*. \end{aligned}$$

The last inequality follows from simple calculus that $\delta = \sqrt{2\alpha(1)}/2$ yields the minimal value for $(2\alpha(1) - 4\delta(1-\delta))/(2\delta)$ when $0 \leq \delta \leq 1$. Note that for $\alpha(1) \geq .878567$, $2(\sqrt{2\alpha(1)} - 1) > .651$. Since the largest $w(\tilde{S})$ in the process is at least as good as the one who meets $z \geq 2\alpha(1)$, they proceed to prove a .651-approximation algorithm for Max-Bisection.

4.5.3 A modified rounding and improved analyses

Our improved rounding method is to use a convex combination of $\bar{X}$ and a positive semidefinite matrix P as the covariance matrix to generate u and $\hat{x}$, i.e,

$$u \in N(0, \theta\bar{X} + (1-\theta)P),$$

$$\hat{x} = \text{sign}(u),$$
$$S = \{i : \ \hat{x}_i = 1\} \quad \text{or} \quad S = \{i : \ \hat{x}_i = -1\}$$

such that $|S| \geq n/2$, and then $\tilde{S}$ from the Frieze and Jerrum swapping procedure.

One choice of P is

$$P = \frac{n}{n-1}(I - \frac{1}{n}ee^T),$$

where I is the n-dimensional identity matrix. Note that $I - \frac{1}{n}ee^T$ is the projection matrix onto the null space of $e^T x = 0$, and P is a feasible solution for (SDP). The other choice is $P = I$, which was also proposed in Nesterov [242], and by Zwick [348] for approximating the Max-Cut problem when the graph is sparse.

Again, the overall performance of the rounding method is determined by two factors: the expected quality of $w(S)$ and how much S need to be downsized. The convex combination parameter θ used in $\theta\bar{X} + (1-\theta)P$ provides a balance between these two factors. Typically, the more use of $\bar{X}$ in the combination results in higher expected $w(S)$ but larger expected difference between $|S|$ and $n/2$; and the more use of P results in less expected $w(S)$ and more accurate $|S|$.

More precisely, our hope is that we could provide two new inequalities in replacing (4.11) and (4.12):

$$\text{E}[w(S)] = \frac{1}{4}\sum_{i,j} w_{ij}(1 - \text{E}[\hat{x}_i\hat{x}_j]) \geq \alpha \cdot w^* \tag{4.17}$$

and

$$\text{E}[|S|(n-|S|)] = \text{E}\left[\frac{n^2}{4} - \frac{(e^T\hat{x})^2}{4}\right] = \frac{1}{4}\sum_{i,j}(1 - \text{E}[\hat{x}_i\hat{x}_j]) \geq \beta \cdot \frac{n^2}{4}, \tag{4.18}$$

such that α would be slightly less than $\alpha(1)$ but β would be significantly greater than $\alpha(1)$. Thus, we could give a better overall bound than .651 for Max-Bisection.

Before proceed, we prove a technical result. Again, let two random variables

$$w := w(S) = \frac{1}{4}\sum_{i,j} w_{ij}(1 - \hat{x}_i\hat{x}_j) \quad \text{and} \quad m := |S|(n-|S|) = \frac{1}{4}\sum_{i,j}(1 - \hat{x}_i\hat{x}_j).$$

Then, from (4.17) and (4.18),

$$\text{E}[w] \geq \alpha \cdot w^* \quad \text{and} \quad \text{E}[m] \geq \beta \cdot m^*, \quad \text{where} \quad m^* = \frac{n^2}{4}.$$

Furthermore, for a given parameter $\gamma \geq 0$, let new random variable

$$z(\gamma) = \frac{w}{w^*} + \gamma\frac{m}{m^*}. \tag{4.19}$$

Then, we have

$$\text{E}[z(\gamma)] \geq \alpha + \gamma\beta \quad \text{and} \quad z \leq 2 + \gamma.$$

Note that Frieze and Jerrum used $\gamma = 1$ in their analyses.

Now we prove the following lemma:

Lemma 4.17 *Assume (4.17) and (4.18) hold. Then, for any given $\gamma \geq \alpha/(4-\beta)$, if random variable $z(\gamma)$ meets its expectation, i.e., $z(\gamma) \geq \alpha + \gamma\beta$, then*

$$w(\tilde{S}) \geq 2\left(\sqrt{\gamma(\alpha+\gamma\beta)} - \gamma\right) \cdot w^*.$$

In particular, if

$$\gamma = \frac{\alpha}{2\beta}\left(\frac{1}{\sqrt{1-\beta}} - 1\right)$$

(which is greater than $\alpha/(4-\beta)$ since $\beta > 0$), then

$$w(\tilde{S}) \geq \frac{\alpha}{1+\sqrt{1-\beta}} \cdot w^*.$$

Proof. Suppose

$$w(S) = \lambda w^* \quad \text{and} \quad |S| = \delta n,$$

which from (4.19) and $z(\gamma) \geq \alpha + \gamma\beta$ implies that

$$\lambda \geq \alpha + \gamma\beta - 4\gamma\delta(1-\delta).$$

Applying (4.13) we see that

$$\begin{aligned} w(\tilde{S}) &\geq \frac{w(S)}{2\delta} \\ &= \frac{\lambda w^*}{2\delta} \\ &\geq \frac{\alpha + \gamma\beta - 4\gamma\delta(1-\delta)}{2\delta} \cdot w^* \\ &\geq 2(\sqrt{\gamma(\alpha+\gamma\beta)} - \gamma) \cdot w^*. \end{aligned}$$

The last inequality follows from simple calculus that

$$\delta = \frac{\sqrt{\alpha+\gamma\beta}}{2\sqrt{\gamma}}$$

yields the minimal value for $(\alpha + \gamma\beta - 4\gamma\delta(1-\delta))/(2\delta)$ in the interval $[0,1]$, if $\gamma \geq \alpha/(4-\beta)$.

In particular, substitute

$$\gamma = \frac{\alpha}{2\beta}\left(\frac{1}{\sqrt{1-\beta}} - 1\right)$$

into the first inequality, we have the second desired result in the lemma. ∎

The motivation to select $\gamma = \frac{\alpha}{2\beta}(\frac{1}{\sqrt{1-\beta}} - 1)$ is that it yields the maximal value for $2(\sqrt{\gamma(\alpha+\gamma\beta)} - \gamma)$. In fact, when both $\alpha = \beta = \alpha(1) \geq .878567$ as in the case of Frieze and Jerrum,

$$\frac{\alpha}{1+\sqrt{1-\beta}} > 0.6515,$$

which is just slightly better than $2(\sqrt{2\alpha(1)}-1) \geq 0.6511$ proved by Frieze and Jerrum. So their choice $\gamma = 1$ is almost "optimal". We emphasize that γ is only used in the analysis of the quality bound, and is not used in the rounding method.

4.5.4 A simple .5-approximation

To see the impact of θ in the new rounding method, we analyze the other extreme case where $\theta = 0$ and

$$P = \frac{n}{n-1}(I - \frac{1}{n}ee^T).$$

That is, we generate $u \in N(0, P)$, then $\hat{x}$ and S. Now, we have

$$\mathrm{E}[w(S)] = \mathrm{E}\left[\frac{1}{4}\sum_{i,j} w_{ij}(1-\hat{x}_i\hat{x}_j)\right] = \frac{1}{4}\left(1 + \frac{2}{\pi}\arcsin(\frac{1}{n-1})\right)\sum_{i\neq j} w_{ij} \geq .5\cdot w^* \tag{4.20}$$

and

$$\mathrm{E}\left[\frac{n^2}{4} - \frac{(e^T\hat{x})^2}{4}\right] \geq \frac{n^2}{4} - \frac{n}{4} + \frac{n(n-1)}{4}\frac{2}{\pi}\arcsin(\frac{1}{n-1}) \geq (1-\frac{1}{n})\cdot\frac{n^2}{4}, \tag{4.21}$$

where from (4.10) we have used the facts that

$$\mathrm{E}[\hat{x}_i\hat{x}_j] = \frac{2}{\pi}\arcsin(\frac{-1}{n-1}), \quad i \neq j$$

and

$$\frac{1}{2}\sum_{i\neq j} w_{ij} = \sum_{i<j} w_{ij} \geq w^*.$$

In other words, we have in Lemma 4.17

$$\alpha = .5 \quad \text{and} \quad \beta = 1 - \frac{1}{n}.$$

Comparing (4.20) and (4.21) to (4.11) and (4.12), here the first inequality on $w(S)$ is worse, .5 vs .878567; but the second inequality is substantially better, $1 - \frac{1}{n}$ vs .878567; i.e., $\hat{x}$ here is a bisection with probability almost 1 when n is large. Using Lemma 4.17, we see that the method is a

$$\frac{\alpha}{1+\sqrt{1-\beta}} > \frac{.5}{1+\sqrt{1/n}}$$

approximation method. The same ratio can be established for $P = I$.

4.5.5 A .699-approximation

We now prove our main result. For simplicity, we will use $P = I$ in our rounding method. Therefore, we discuss using the convex combination of $\theta\bar{X} + (1-\theta)I$ as the covariance matrix to generate u, $\hat{x}$, S and $\tilde{S}$ for a given $0 \le \theta \le 1$, i.e.,

$$u \in N(0, \theta\bar{X} + (1-\theta)P),$$

$$\hat{x} = \text{sign}(u),$$

$$S = \{i : \ \hat{x}_i = 1\} \quad \text{or} \quad S = \{i : \ \hat{x}_i = -1\}$$

such that $|S| \ge n/2$, and then $\tilde{S}$ from the Frieze and Jerrum swapping procedure.

Define

$$\alpha(\theta) := \min_{-1 \le y < 1} \frac{1 - \frac{2}{\pi}\arcsin(\theta y)}{1-y}; \tag{4.22}$$

and

$$\beta(\theta) := (1 - \frac{1}{n})b(\theta) + c(\theta), \tag{4.23}$$

where

$$b(\theta) = 1 - \frac{2}{\pi}\arcsin(\theta) \quad \text{and} \quad c(\theta) = \min_{-1 \le y < 1} \frac{2}{\pi}\frac{\arcsin(\theta) - \arcsin(\theta y)}{1-y}.$$

Note that $\alpha(1) = \beta(1) \ge .878567$ as shown in Goemans and Williamson [127]; and $\alpha(0) = .5$ and $\beta(0) = 1 - \frac{1}{n}$. Similarly, one can also verify that

$$\alpha(.89) \ge .835578, \quad b(.89) \ge .301408 \quad \text{and} \quad c(.89) \ge .660695.$$

We now prove another technical lemma:

Lemma 4.18 *For any given $0 \le \theta \le 1$ in our rounding method, inequalities (4.17) and (4.18) hold for*

$$\alpha = \alpha(\theta) \quad \textit{and} \quad \beta = \beta(\theta).$$

Proof. Since $1 - \frac{2}{\pi}\arcsin(\theta) \ge 0$ for any $0 \le \theta \le 1$, from definition (4.22) we have

$$\begin{aligned} \mathrm{E}[w(S)] &= \frac{1}{4}\sum_{i,j} w_{ij}\left(1 - \frac{2}{\pi}\arcsin(\bar{X}_{ij})\right) \\ &\ge \frac{1}{4}\sum_{i,j} w_{ij} \cdot \alpha(\theta)(1 - \bar{X}_{ij}) \\ &= \alpha(\theta) \cdot w^{SD} \\ &\ge \alpha(\theta) \cdot w^*. \end{aligned}$$

Noting that

$$\sum_{i \ne j} \bar{X}_{ij} = -n,$$

from the definition (4.23) we have

$$\begin{aligned}
\mathrm{E}[|S|(n-|S|)] &= \mathrm{E}\left[\frac{n^2}{4} - \frac{(e^T\hat{x})^2}{4}\right] \\
&= \frac{1}{4}\sum_{i\neq j}\left(1 - \frac{2}{\pi}\arcsin(\theta\bar{X}_{ij})\right) \\
&= \frac{1}{4}\sum_{i\neq j}\left(1 - \frac{2}{\pi}\arcsin(\theta) + \frac{2}{\pi}\arcsin(\theta) - \frac{2}{\pi}\arcsin(\theta\bar{X}_{ij})\right) \\
&= \frac{1}{4}\sum_{i\neq j}\left(b(\theta) + \left(\frac{2}{\pi}\arcsin(\theta) - \frac{2}{\pi}\arcsin(\theta\bar{X}_{ij})\right)\right) \\
&\geq \frac{1}{4}\sum_{i\neq j}\big(b(\theta) + c(\theta)(1-\bar{X}_{ij})\big) \\
&= \frac{1}{4}\big((n^2-n)b(\theta) + (n^2-n)c(\theta) + nc(\theta)\big) \\
&= \left((1-\frac{1}{n})b(\theta) + c(\theta)\right)\cdot\frac{n^2}{4} \\
&= \beta(\theta)\cdot\frac{n^2}{4}.
\end{aligned}$$

∎

Lemmas 4.17 and 4.18 together imply that for any given θ between 0 and 1, our rounding method will generate a

$$w(\tilde{S}) \geq \frac{\alpha(\theta)}{1+\sqrt{1-\beta(\theta)}}\cdot w^*$$

as soon as (bounded) $z(\gamma)$ of (4.19) meets its expectation. Thus, we can set θ to a value θ^* in $[0,1]$ such that $\frac{\alpha(\theta)}{1+\sqrt{1-\beta(\theta)}}$ is maximized, that is, let

$$\theta^* = \arg\max_{\theta\in[0,1]}\frac{\alpha(\theta)}{1+\sqrt{1-\beta(\theta)}}$$

and

$$r^{MB} = \frac{\alpha(\theta^*)}{1+\sqrt{1-\beta(\theta^*)}}.$$

In particular, if $\theta = .89$ is selected in the new rounding method,

$$\alpha(.89) > .8355,$$

and for n sufficiently large ($\geq 10^4$)

$$\beta(.89) = (1-\frac{1}{n})b(.89) + c(.89) > .9620,$$

which imply

$$r^{MB} = \frac{\alpha(\theta^*)}{1+\sqrt{1-\beta(\theta^*)}} \geq \frac{\alpha(.89)}{1+\sqrt{1-\beta(.89)}} > .69920.$$

This bound yields the final result:

Theorem 4.19 *There is a polynomial-time approximation algorithm for Max-Bisection whose expected cut is at least r^{MB} times the maximal bisection cut, if the number of nodes in the graph is sufficiently large. In particular, if parameter $\theta = .89$ is used, our rounding method is a .699-approximation for Max-Bisection.*

The reader may ask why we have used two different formulations in defining $\alpha(\theta)$ of (4.22) and $\beta(\theta)$ of (4.23). The reason is that we have no control on the ratio, ρ, of the maximal bisection cut w^* over the total weight $\sum_{i<j} w_{ij}$, i.e.,

$$\rho := \frac{w^*}{\sum_{i<j} w_{ij}}.$$

Note that ρ ranges from 1/2 to 1. Indeed, using the second derivation in Lemma 4.18, we can also prove that in Lemma 4.17

$$\alpha \geq \frac{1}{2\rho} b(\theta) + c(\theta).$$

Thus, in the worse case $\rho = 1$, we can only establish

$$\alpha \geq \frac{1}{2} b(\theta) + c(\theta).$$

Then, for $\theta = .89$, we have $\alpha \geq .8113$, which is less than .8355 established by using the first derivation.

However, if $\rho \leq .8$, then we have $\alpha \geq .8490$. For Max-Bisection on these graphs, our method is a .710 approximation for setting $\theta = .89$. This bound can be further improved by setting a smaller θ. In general, the quality bound improves as ρ decreases. When ρ near 1/2, we have a close to 1 approximation if $\theta = 0$ is chosen, since $b(0) = 1$, $c(0) = 0$, $\alpha \geq \frac{1}{2\rho}$ and $\beta \geq 1 - \frac{1}{n}$.

In any case, we can run our rounding method 100 times for parameter $\theta =$ .00, .01, ..., .98, .99 and report the best rounding solution among the 100 tries. This will ensure us to produce a solution with a near best guarantee, but without the need to know ρ.

4.6 Notes

Semidefinite relaxations have recently appeared in relation to relaxations for 0-1 optimization problems. In [209], a "lifting" procedure is presented to obtain a problem in $\Re^{n^2}$; and then the problem is projected back to obtain tighter inequalities. See also [31]. Several of the operators that arise in our applications

are similar to those that appear in [209]. However, our motivation and approach is different. A discussion of several applications for semidefinite relaxation appears in [9]. Also see recent papers by Fujie and Kojima [115] and Polijak, Rendl and Wolkowicz [256], Zhao, Karisch, Rendl, and Wolkowicz [344], and Skutella [281].

During the past ten years, there have been several remarkable results on approximating specific quadratic problems using positive semi-definite programming. Goemans and Williamson [127] proved an approximation result for the Maxcut problem where $\epsilon \leq 1 - 0.878$. Nesterov [242] extended their result to approximating a boolean QP problem where $\epsilon \leq 4/7$. Most of the results are related to graph partition.

The proof of polynomial-time complexity for solving a single ball constrained quadratic problem was initially done using non-SDP techniques; see [321], [335] and Fu, Luo and Ye [114]. Results on QCQP with multiple quadartic constraints were developed by Ye [336], Nemirovski, Roos and Terlaky [240], Sturm and Zhang [289], Ye and Zhang [341], etc..

More details on graph partition. Given an undirected graph $G = (V, E)$ with $|V| = n$, non-negative weights w_{ij} on edges $(i, j) \in E$, and an integer k $(1 < k < n)$, the maximization graph partition (MAX-GP) problem is to determine a subset $S \subset V$ of k nodes such that an objective function $w(S)$ is maximized. Some examples of MAX-GP are: *Dense-k-Subgraph (DSP)*, where the total edge weights of the subgraph induced by S is maximized; *Max-Cut with size k (MC)*, where the total edge weights of the edges crossing between S and $V \setminus S$ is maximized; *Max-Not-Cut with size k (MNC)*, where the total edge weights of the non-crossing edges between S and $V \setminus S$ is maximized; *Max-Vertex-Cover with size k (MVC)*, where the total edge weights of the edges covered by S is maximized.

Since these MAX-GP problems are NP-hard (e.g., see [102] for DSP, [2] for MC, [153] for MNC, and [255] for MVC), one should not expect to find polynomial time algorithms for computing their optimal solutions. Therefore, we are interested in how close to optimality one can approach in polynomial time. A (randomized) polynomial time approximation algorithm for a maximization problem has a performance guarantee or worst case ratio $0 < r \leq 1$, if it outputs a feasible solution whose (expected) value is at least r times the maximal value for all instance of the problem. Such an algorithm is often called (randomized) r-approximation algorithm. A key step in designing a good approximation algorithm for such a maximization problem is to establish a good upper bound on the maximal objective value. Linear programming (LP) and semidefinite programming (SDP) have been frequently used to provide such upper bounds for many NP-hard problems.

There are several approximation algorithms for DSP. Kortsarz and Peleg [195] devised an approximation algorithm which has a performance guarantee $O(n^{-0.3885})$. Feige, Kortsarz and Peleg [103] improved it to $O(n^{-1/3+\epsilon})$, for some $\epsilon > 0$. The other approximation algorithms have performance guarantees which are the function of k/n, e.g., a greedy heuristic by Asahiro et.al. [24], and SDP relaxation based algorithms developed by Feige and Langberg [101]

and Feige and Seltser [102], and Srivastav and Wolf [285]. The previously best performance guarantee, k/n for general k and $k/n+\epsilon_k$ for $k \sim n/2$, was obtained by Feige and Langberg [101]. Moreover, for the case $k = n/2$, Ye and Zhang [153], using a new SDP relaxation, obtained an improved 0.586 performance guarantee from 0.517 of Feige and Langberg [101].

For approximating the MC problem with size k, both LP and SDP based approximation algorithms have a performance ratio $1/2$ for all k, see Ageev and Sviridenko [2] and Feige and Langberg [101]. For $k = n/2$, i.e., the Max-Bisection, Frieze and Jerrum [112] obtained a 0.651-approximation algorithm (the same bound was also obtained by Andersson [16] in his paper for max-p-section). Subsequently, this ratio has been improved to 0.699 by Ye [337]. Both of their approximation algorithms are based on SDP relaxations.

For approximating the MNC problem with size k, the LP-based approximation algorithm has a performance ratio $1 - \frac{2k(n-k)}{n(n-1)}$ for all k, and the SDP-based algorithm has a ratio $.5 + \epsilon_k$ for $k \sim n/2$, see Feige and Langberg [101]. Again, Ye and Zhang [153] obtained a 0.602-approximation algorithm for MNC when $k = n/2$, comparing to .541 of Feige and Langberg [101].

Han et al. [153] has presented an improved method to round an optimal solution of the SDP relaxation of the MAX-GP problem for general k. This rounding technique is related to the well-known rounding method introduced by Goemans and Williamson [127], Feige and Goemans [100] for MAX-DICUT and MAX 2-SAT, Zwick [347] for constraint satisfaction problems, and Nesterov[242] and Zwick [348] for MAX-CUT. This kind of randomized algorithm can be derandomized by the technique of Mahajan and Ramesh[216].

What complicates matters in the MAX-GP problem, comparing to the MAX-CUT problem, is that two objectives are sought—the objective value of $w(S)$ and the size of S. Therefore, in any (randomized) rounding method, we need to balance the (expected) quality of $w(S)$ and the (expected) size of S. One wants high $w(S)$; but, at the same time, zero or small difference between $|S|$ and k, since otherwise we have to either add or subtract nodes from S, resulting in a deterioration of $w(S)$ at the end. Our improved rounding method is built upon this balance need.

As consequences of the improved rounding method, they have yielded improved approximation performance ratios for DSP, MC, MNC and MVC, on a wide range of k. On approximating DSP, for example, our algorithm has guaranteed performance ratios .648 for $k = 3n/5$, 0.586 for $k = n/2$, 0.486 for $k = 2n/5$ and 0.278 for $k = n/4$. For MC and MNC, the performance guarantees are also much better than 0.5 for a wide range of k. On approximating MVC, our algorithm has guaranteed performance ratios .845 for $k = 3n/5$, 0.811 for $k = n/2$, and 0.733 for $k = 2n/5$.

4.7 Exercises

4.1 *Let X be a positive semidefinite matrix of rank r, A be a given symmetric matrix. Then, there is a decomposition of X*

$$X = \sum_{j=1}^{r} x_j x_j^T,$$

such that for all j,

$$x_j^T A x_j = A \bullet (x_j x_j^T) = A \bullet X/r.$$

4.2 *Given any matrix $A \in R^{n \times m}$, using the SDP rank reduction theorem to show that the matrix singular value problem*

$$\begin{array}{ll} \textit{minimize} & x^T A y \\ \textit{s.t.} & \|x\|^2 = 1, \\ & \|y\|^2 = 1, \end{array}$$

is an SDP problem.

4.3 *Consider the SDP problem*

$$\begin{array}{lll} \textit{minimize} & C \bullet X & \\ \textit{s.t.} & A_i \bullet X = b_i, & i = 1, ..., m, \\ & Q_j \bullet X = 0, & j = 1, ..., q, \\ & X \succeq 0, & \end{array}$$

where coefficient matrices Q_j, $j = 1, \ldots, q$, are positive semidefinite.

1. *Suppose that there is an optimal solution X^* with zero duality gap, show that there must be an optimal solution matrix with its rank r satisfying $r(r+1)/2 \leq m$. (Note that the bound is independent of q.)*

2. *Using the above result to show that the quadratic problem*

$$\begin{array}{ll} \textit{minimize} & x^T Q x + 2c^T x \\ \textit{s.t.} & Ax = b, \\ & \|x\|^2 = 1 \end{array}$$

is an SDP problem, where given Q is an n-dimensional symmetric matrix and A is an $m \times n$ matrix with $m < n$.

4.4 *Prove Lemma 4.6.*

4.5 *Using the exercise above, prove Theorem 4.7.*

4.6 *Prove Proposition 4.8.*

4.7 *Prove Lemma 4.9.*

4.8 *Using the exercise above, prove Theorem 4.10.*

4.9 *The $s-t$ Max-Cut Problem: This is a max-cut problem where we require that a pair of nodes s and t be separated by the cut.*

- *Construct the SDP relaxation for the problem.*
- *Show that the problem can be approximated by the factor* 0.878.
- *The result holds for more pairs need to be separated.*

4.10 *Triangle Inequality: In the Max-Cut or other problems, decision variable x_i is either* 1 *or* -1. *Thus, for any three variables we have*

$$|x_i + x_j + x_k| \geq 1$$

and

$$|x_i - x_j| + |x_j - x_k| \geq |x_k - x_i|$$

which call triangle-type inequalities. Show how to incoporate these inequalities in the SDP relaxation of the problem.

4.11 *In solving the standard SDP problem with m equality constraints, we have shown that there is an optimal SDP solution whose rank r satisfying*

$$\frac{r(r+1)}{2} \leq m,$$

and there is a strongly polynomial-time algorithm to find such a solution from any exactly optimal SDP solution. In reality, one may never generate an exactly optimal SDP solution but an ϵ-optimal solution X, meaning its minimal objective value is ϵ away from the optimal z^*:

$$C \bullet X \leq z^* + \epsilon.$$

Prove that, from X, one can find an ϵ-optimal SDP solution whose rank r satisfies

$$\frac{r(r+1)}{2} \leq m + 1$$

in strongly polynomial time. Show how the rank-reduction procedure works.

4.12 *Consider the following eigenvalue optimization problem*

$$\begin{array}{ll} \textit{minimize} & n\lambda_{max}(Q + Diag(\bar{y})) \\ \textit{subject to} & e^T\bar{y} = 0 \\ & \bar{y} \in R^m, \end{array} \tag{4.24}$$

where $\lambda_{max}(A)$ is the maximum eigenvalue of the matrix A. Show that this problem is equivalent to the dual of the SDP relaxation for Max-Cut.

4.13 *Quadratic 0-1 Programming: The following problem is referred as the Quadratic 0-1 programming*

$$\begin{array}{ll} \textit{maximize} & x^T B x \\ \textit{subject to} & x \in \{0,1\}^n. \end{array} \tag{4.25}$$

Show that the following matrix representation is equivalent to the above quadratic 0-1 programming problem (4.25):

$$\begin{array}{ll} \textit{maximize} & B \bullet X \\ \textit{subject to} & \bar{X} = \left[\begin{array}{cc} 1 & diag(X)^T \\ diag(X) & X \end{array} \right] \succeq 0, \\ & rank(\bar{X}) = 1, \end{array} \tag{4.26}$$

where $diag(A)$ is the vector of diagnoal elements in A.

Note 1. It suffices to show that both problems have the same feasible region.

Note 2. Problem (4.26) without the rank 1 constraint on X is an SDP relaxation of (4.25).

4.14 *The 2-Catalog Segmentation Problem ([328]): Given a ground set I of n items, a family $\{S_1, S_2, \cdots, S_m\}$ of subsets of I and an integer $1 \leq k \leq n$. The problem is to find two subsets $A_1, A_2 \subset I$ such that $|A_1| = |A_2| = k$ to maximize $\sum_{i=1}^{m} \max\{|S_i \cap A_1|, |S_i \cap A_2|\}$. Here I can be the list of goods; there are m customers where customer i is interested in the goods of S_i; A_1 and A_2 are the two catalogs one of which could be sent to each of the customers such that the (total) "satisfaction" is maximized. Find an SDP relaxation to the problem.*

4.15 *The Sparsest Cut Problem ([23]): Given a graph $G = (V, E)$. For any cut $(S, \bar{S})$ with $|S| \leq |V|/2$, the edge expansion of the cut is $|E(S, \bar{S})|/|S|$. The problem is to find the S such that $|E(S, \bar{S})|/|S|$ is the smallest. Find an SDP relaxation to the problem.*

Chapter 5

SDP for Geometry Computation

One of the most studied problems in distance geometry is the *Graph Realization* problem, in which one is given a graph $G = (V, E)$ and a set of non–negative weights $\{d_{ij} : (i, j) \in E\}$ on its edges, and the goal is to compute a realization of G in the Euclidean space R^d for a given dimension d, i.e. to place the vertices of G in R^d such that the Euclidean distance between every pair of adjacent vertices v_i, v_j equals to the prescribed weight d_{ij}. This problem and its variants arise from applications in various areas, such as molecular conformation, dimensionality reduction, Euclidean ball packing, and more recently, wireless sensor network localization. In this chapter, we discuss SDP formulations of such Euclidean distance geometry problems.

5.1 The basic SDP model

The basic mathematical model of Euclidean distance geometry optimization can be described as a quadratically constrained optimization problem. Let n unknown points $x_j \in R^d$, $j = 1, ..., n$. For each pair of (i, j), we are given Euclidean distance upper bound $\bar{d}_{ij}$ and lower bound $\underline{d}_{ij}$ between the two points x_i and x_j, and an objective function of $x_1, ..., x_n$. Moreover, we require that each point $x_j \in \Omega$ where Ω is a convex set. Then, the basic model can be formulated as:

$$\begin{array}{ll} \text{minimize} & f(x_1, ..., x_n) \\ \text{subject to} & (\underline{d}_{ij})^2 \leq \|x_i - x_j\|^2 \leq (\bar{d}_{ij})^2, \ \forall i < j \\ & x_j \in \Omega, \ \forall j, \end{array} \tag{5.1}$$

Note that the "bounding from below" constraint

$$\|(x_i - x_j)\|^2 \geq (\underline{d}_{ij})^2 \tag{5.2}$$

is not convex, so that the efficient convex optimization techniques cannot apply to solving this problem.

Let $X = [x_1\ x_2\ ...\ x_n]$ be the $d \times n$ matrix that needs to be determined. Then

$$\|x_i - x_j\|^2 = e_{ij}^T X^T X e_{ij},$$

where e_{ij} is the vector with 1 at the ith position, -1 at the jth position and zero everywhere else. Let $Y = X^T X$. Then problem (5.1) can be rewritten as:

$$\begin{array}{ll} \text{minimize} & f(X,Y) \\ \text{subject to} & (\underline{d}_{ij})^2 \le e_{ij}^T Y e_{ij} \le (\bar{d}_{ij})^2,\ \forall i<j, \\ & x_j \in \Omega, \forall j, \\ & Y = X^T X. \end{array} \tag{5.3}$$

Our approach is to relax problem (5.3) to a semidefinite program:

$$\begin{array}{ll} \text{minimize} & f(X,Y) \\ \text{subject to} & (\underline{d}_{ij})^2 \le e_{ij}^T Y e_{ij} \le (\bar{d}_{ij})^2,\ \forall i<j, \\ & x_j \in \Omega, \forall j, \\ & Y \succeq X^T X. \end{array} \tag{5.4}$$

The last matrix inequality is equivalent to (Boyd et al. [58])

$$Z := \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} \succeq 0.$$

Then, the problem can be written as a standard SDP problem:

$$\begin{array}{ll} \text{minimize} & f(Z) \\ \text{subject to} & Z(1:d,1:d) = I, \\ & (\underline{d}_{ij})^2 \le (0;e_{ij})^T Z (0;e_{ij}) \le (\bar{d}_{ij})^2,\ \forall i<j, \\ & Z(1:d,j) \in \Omega\ j = d+1,...,d+n, \\ & Z \succeq 0. \end{array} \tag{5.5}$$

If $f(Z)$ is a convex function of Z, then (5.5) becomes a convex optimization problem. In particular, if $f(Z)$ is a linear function of Z and Ω is a polyhedron, then (5.5) is a (linear) semidefinite program.

5.2 Wireless sensor network localization

There has been an increase in the use of semidefinite programming (SDP) for solving wide range of Euclidean distance geometry problems, such as data compression, metric-space embedding, ball packing, chain folding etc. One application of the SDP Euclidean distance geometry model lies in ad hoc wireless sensor networks which are constructed for monitoring environmental information(temperature, sound levels, light etc) across an entire physical space. Typical networks of this type consist of a large number of densely deployed sensor

nodes which gather local data and communicate with other nodes within a small range. The sensor data from these nodes are relevant only if we know what location they refer to. Therefore knowledge of of each sensor position becomes imperative. Generally, the use of a GPS system is a very expensive solution to this requirement.

Indeed, other techniques to estimate node positions are being developed that rely just on the measurements of distances between neighboring nodes [53, 67, 81, 118, 163, 165, 247, 270, 271, 272, 275]. The distance information could be based on criterion like time of arrival, angle of arrival and received signal strength. Depending on the accuracy of these measurements and processor, power and memory constraints at each of the nodes, there is some degree of error in the distance information. Furthermore, it is assumed that we already know the positions of a few anchor nodes. The problem of finding the positions of all the nodes given a few anchor nodes and partial distance information between the nodes is called the position estimation or localization problem.

In particular, the paper [53] describes an SDP relaxation based model for the position estimation problem in sensor networks. The optimization problem is set up so as to minimize the error in the approximate distances between the sensors. Observable traces are developed to measure the quality of the distance data. The basic idea behind the technique is to convert the non-convex quadratic distance constraints into linear constraints by introducing relaxations to remove the quadratic term in the formulation. The performance of this technique is highly satisfactory compared to other techniques. Very few anchor nodes are required to accurately estimate the position of all the unknown sensors in a network. Also the estimation errors are minimal even when the anchor nodes are not suitably placed within the network. More importantly, for each sensor the model generates numerical data to measure the reliability and accuracy of the positions computed from the model, which can be used to detect erroneous or outlier sensors.

5.2.1 An SDP relaxation model

For simplicity, let the sensor points be placed on a plane. Recall that we have m known anchor points $a_k \in R^2$, $k = 1, ..., m$, and n unknown sensor points $x_j \in R^2$, $j = 1, ..., n$. For every pair of two points, we have a Euclidean distance measure if the two are within a communication distance range R. Therefore, say for $(i, j) \in N_x$, we are given Euclidean distance data $\hat{d}_{ij}$ between unknown sensors i and j, and for $(k, j) \in N_a$ we know distance $\hat{d}_{kj}$ between anchor k and sensor j. Note that for the rest of pairs we have only a lower bound R for their pair-wise distances. Therefore, the localization problem can be formulated as

an error minimization problem with mixed equalities and inequalities:

$$\begin{array}{ll} \text{minimize} & \sum_{i,j\in N_x,\ i<j} |\alpha_{ij}| + \sum_{k,j\in N_a} |\alpha_{kj}| \\ \text{subject to} & \|x_i - x_j\|^2 = (\hat{d}_{ij})^2 + \alpha_{ij},\ \forall\ (i,j) \in N_x,\ i<j, \\ & \|a_k - x_j\|^2 = (\hat{d}_{kj})^2 + \alpha_{kj},\ \text{for } (k,j) \in N_a, \\ & \|x_i - x_j\|^2 \geq R^2,\ \text{for the rest } i<j, \\ & \|a_k - x_j\|^2 \geq R^2,\ \text{for the rest } k,j. \end{array}$$

Thus, we relax the problem to a semidefinite program:

$$\begin{array}{ll} \text{minimize} & \sum_{i,j\in N_x,\ i<j} |\alpha_{ij}| + \sum_{k,j\in N_a} |\alpha_{kj}| \\ \text{subject to} & (1;0;\mathbf{0})^T Z(1;0;\mathbf{0}) = 1 \\ & (0;1;\mathbf{0})^T Z(0;1;\mathbf{0}) = 1 \\ & (1;1;\mathbf{0})^T Z(1;1;\mathbf{0}) = 2 \\ & (\mathbf{0};e_{ij})^T Z(\mathbf{0};e_{ij}) = (\hat{d}_{ij})^2 + \alpha_{ij},\ \forall\ (i<j,j) \in N_x, \\ & (a_k;e_j)^T Z(a_k;e_j) = (\hat{d}_{kj})^2 + \alpha_{kj},\ \forall\ (k,j) \in N_a, \\ & (\mathbf{0};e_{ij})^T Z(\mathbf{0};e_{ij}) \geq R^2,\ \forall\ (i<j,j) \notin N_x, \\ & (a_k;e_j)^T Z(a_k;e_j) \geq R^2,\ \forall\ (k,j) \notin N_a, \\ & Z \succeq 0. \end{array} \tag{5.6}$$

The matrix of Z has $2n + n(n+1)/2$ unknown variables. Consider the case that among $\{k,i,j\}$, there are $2n + n(n+1)/2$ of the pairs where each of them has same distance upper and lower bounds, and $\alpha = 0$ for the minimal solution of (5.6). Then we have at least $2n + n(n+1)/2$ linear equalities among the constraints. Moreover, if these equalities are linearly independent, then Z has a unique solution. Therefore, we can show

Proposition 5.1 *If there are $2n + n(n+1)/2$ distance pairs each of which has an accurate distance measure and other distance bounds are feasible. Then, the minimal value of $\alpha = 0$ in (5.6). Moreover, if (5.6) has a unique minimal solution*

$$\bar{Z} = \begin{pmatrix} I & \bar{X} \\ \bar{X}^T & \bar{Y} \end{pmatrix},$$

then we must have $\bar{Y} = (\bar{X})^T \bar{X}$ and $\bar{X}$ equal true positions of the unknown sensors. That is, the SDP relaxation solves the original problem exactly.

Proof. Let X^* be the true locations of the n points, and

$$Z^* = \begin{pmatrix} I & X^* \\ (X^*)^T & (X^*)^T X^* \end{pmatrix}.$$

Then Z^* and $\alpha = 0$ is a feasible solution for (5.6).

On the other hand, since $\bar{Z}$ is the unique solution to satisfy the $2n+n(n+1)/2$ equalities, we must have $\bar{Z} = Z^*$ so that $\bar{Y} = (X^*)^T X^* = \bar{X}^T \bar{X}$.

We present a simple case to show what it means for the system has a unique solution. Consider $n = 1$ and $m = 3$. The accurate distance measures from

unknown b_1 to known a_1, a_2 and a_3 are d_{11}, d_{21} and d_{31}, respectively. Therefore, the three linear equations are

$$\begin{aligned} y - 2x^T a_1 &= (d_{11})^2 - \|a_1\|^2 \\ y - 2x^T a_2 &= (d_{21})^2 - \|a_2\|^2 \\ y - 2x^T a_3 &= (d_{31})^2 - \|a_3\|^2 \end{aligned}$$

This system has a unique solution if it has a solution and the matrix

$$\begin{pmatrix} 1 & 1 & 1 \\ a_1 & a_2 & a_3 \end{pmatrix}$$

is nonsingular. This essentially means that the three points a_1, a_2 and a_3 are not on the same line, and then $\bar{x} = b_1$ can be uniquely determined. Here, the SDP method reduces to the so-called triangular method. Proposition 5.1 and the example show that the SDP relaxation method has the advantage of the triangular method in solving the original problem.

5.2.2 Probabilistic or error analyses

The case discussed in Proposition 5.1 is deterministic. Alternatively, each x_j can be viewed a random point $\tilde{x}_j$ since the distance measures contain random errors. Then the solution to the SDP problem provides the first and second moment information on $\tilde{x}_j$, $j = 1, ..., n$. Such an interpretation appears to be first stated in Bertsimas and Ye [52].

Generally, we have

$$\mathrm{E}[\tilde{x}_j] \sim \bar{x}_j, \quad j = 1, ..., n$$

and

$$\mathrm{E}[\tilde{x}_i^T \tilde{x}_j] \sim \bar{Y}_{ij}, \quad i, j = 1, ..., n.$$

where

$$\bar{Z} = \begin{pmatrix} I & \bar{X} \\ \bar{X}^T & \bar{Y} \end{pmatrix}$$

is the optimal solution of the SDP problem. Thus,

$$\bar{Y} - \bar{X}^T \bar{X}$$

represents the co-variance matrix of $\tilde{x}_j$, $j = 1, ..., n$.

These quantities also constitute error management and analyses of the original problem data. For example,

$$\mathrm{tr}(\bar{Y} - \bar{X}^T \bar{X}) = \sum_{j=1}^{n} (\bar{Y}_{jj} - \|\bar{x}_j\|^2),$$

the total trace of the co-variance matrix, measures the quality of distance sample data d_{ij} and d_{kj}. In particular, individual trace

$$\bar{Y}_{jj} - \|\bar{x}_j\|^2, \tag{5.7}$$

which is also the variance of $\|\tilde{x}_j\|$, helps us to detect possible distance measure errors, and outlier or defect sensors. These errors often occur in real applications either due to the lack of data information or noisy measurement, and are often difficult to detect since the true location of sensors is unknown.

We again use the same simple case to illustrate our theory. Consider $n = 1$ and $m = 3$. The inexact distance measures from unknown b_1 to known a_1, a_2 and a_3 are $d_{11} + \epsilon$, $d_{21} + \epsilon$ and $d_{31} + \epsilon$, respectively, where ϵ is a random error with zero mean. Therefore, the three linear equations are

$$\begin{aligned} \bar{y} - 2\bar{x}^T a_1 + \|a_1\|^2 &= (d_{11})^2 + 2\epsilon d_{11} + \epsilon^2 \\ \bar{y} - 2\bar{x}^T a_2 + \|a_2\|^2 &= (d_{21})^2 + 2\epsilon d_{21} + \epsilon^2 \\ \bar{y} - 2\bar{x}^T a_3 + \|a_3\|^2 &= (d_{31})^2 + 2\epsilon d_{31} + \epsilon^2. \end{aligned}$$

Taking expect values on both sides, we have

$$\begin{aligned} \mathrm{E}[\bar{y}] - 2\mathrm{E}[\bar{x}]^T a_1 + \|a_1\|^2 &= (d_{11})^2 + \mathrm{E}[\epsilon^2] \\ \mathrm{E}[\bar{y}] - 2\mathrm{E}[\bar{x}]^T a_2 + \|a_2\|^2 &= (d_{21})^2 + \mathrm{E}[\epsilon^2] \\ \mathrm{E}[\bar{y}] - 2\mathrm{E}[\bar{x}]^T a_3 + \|a_3\|^2 &= (d_{31})^2 + \mathrm{E}[\epsilon^2] \end{aligned}$$

or

$$\begin{aligned} \mathrm{E}[\bar{y}] - \mathrm{E}[\bar{x}]^T \mathrm{E}[\bar{x}] + \|\mathrm{E}[\bar{x}] - a_1\|^2 &= (d_{11})^2 + \mathrm{E}[\epsilon^2] \\ \mathrm{E}[\bar{y}] - \mathrm{E}[\bar{x}]^T \mathrm{E}[\bar{x}] + \|\mathrm{E}[\bar{x}] - a_2\|^2 &= (d_{21})^2 + \mathrm{E}[\epsilon^2] \\ \mathrm{E}[\bar{y}] - \mathrm{E}[\bar{x}]^T \mathrm{E}[\bar{x}] + \|\mathrm{E}[\bar{x}] - a_3\|^2 &= (d_{31})^2 + \mathrm{E}[\epsilon^2]. \end{aligned}$$

The solution to the linear equation is

$$\mathrm{E}[\bar{x}] = b_1,$$

and

$$\mathrm{E}[\bar{y}] - \mathrm{E}[\bar{x}]^T \mathrm{E}[\bar{x}] = \mathrm{E}[\epsilon^2]$$

or

$$\mathrm{E}[\bar{y}] = \mathrm{E}[\epsilon^2] + \|b_1\|^2.$$

That is, $\bar{x}$ is a point estimate of b_1. Moreover, from

$$\mathrm{E}[\bar{y}] - \mathrm{E}[\|\bar{x}\|^2] + \mathrm{E}[\|\bar{x}\|^2] - \|b_1\|^2 = \mathrm{E}[\bar{y}] - \|b_1\|^2 = \mathrm{E}[\epsilon^2],$$

we have

$$\mathrm{E}[\bar{y} - \|\bar{x}\|^2] \le \mathrm{E}[\epsilon^2] \quad \text{and} \quad \mathrm{E}[\|\bar{x}\|^2] - \|b_1\|^2 \le \mathrm{E}[\epsilon^2],$$

so that the quantity of $y - \|\bar{x}\|^2$ is a lower bound estimate for the error variance and the variance of $\bar{x}$ is also bounded by the error variance. This quantity gives an interval estimation of b_1.

More generally, we have

Proposition 5.2 *Let the noisy measurements*

$$\hat{d}_{ij} = d_{ij} + \epsilon_i + \epsilon_j, \ \forall \ i \neq j$$

and

$$\hat{d}_{kj} = d_{kj} + \epsilon_j, \ \forall k, j$$

where d_{ij} are the true distances and ϵ_j are independent random errors with zero mean. Moreover, let the minimal $\alpha = 0$ in (5.6) and the anchor points are linear independent. Then, we have

$$\mathrm{E}[\bar{x}_j] = b_j \quad \textit{and} \quad \mathrm{E}[\bar{Y}_{jj}] = \|b_j\|^2 + \mathrm{E}[\epsilon_j^2] \ \forall \ j$$

and

$$\mathrm{E}[\bar{Y}_{ij}] = (b_i)^T b_j \ \forall \ i \neq j,$$

where b_j is the true position of x_j, $j = 1, ..., n$, and

$$\bar{Z} = \begin{pmatrix} I & \bar{X} \\ \bar{X}^T & \bar{Y} \end{pmatrix}$$

is the minimizer of (5.6).

Proof. Since $\alpha = 0$, we have, for all i, j, k,

$$\begin{aligned} \bar{Y}_{ii} - 2\bar{Y}_{ij} + \bar{Y}_{jj} &= (d_{ij} + \epsilon_i + \epsilon_j)^2 \\ \bar{Y}_{jj} - 2\bar{x}_j^T a_k + \|a_k\|^2 &= (d_{kj} + \epsilon_j)^2. \end{aligned}$$

Taking expect values on both sides, we have

$$\begin{aligned} \mathrm{E}[\bar{Y}_{ii}] - 2\mathrm{E}[\bar{Y}_{ij}] + \mathrm{E}[\bar{Y}_{jj}] &= (d_{ij})^2 + \mathrm{E}[\epsilon_i^2] + \mathrm{E}[\epsilon_j^2] \\ \mathrm{E}[\bar{Y}_{jj}] - \mathrm{E}[\bar{x}_j]^T \mathrm{E}[\bar{x}_j] + \|\mathrm{E}[\bar{x}_j] - a_k\|^2 &= (d_{kj})^2 + \mathrm{E}[\epsilon_j^2]. \end{aligned}$$

or

$$\begin{aligned} \mathrm{E}[\bar{Y}_{ii}] - 2\mathrm{E}[\bar{Y}_{ij}] + \mathrm{E}[\bar{Y}_{jj}] - \|\mathrm{E}[\bar{x}_i] - \mathrm{E}[\bar{x}_j]\|^2 + \|\mathrm{E}[\bar{x}_j] - \mathrm{E}[\bar{x}_i]\|^2 &= (d_{ij})^2 + \mathrm{E}[\epsilon_i^2] + \mathrm{E}[\epsilon_j^2] \\ \mathrm{E}[\bar{Y}_{jj}] - \mathrm{E}[\bar{x}_j]^T \mathrm{E}[\bar{x}_j] + \|\mathrm{E}[\bar{x}_j] - a_k\|^2 &= (d_{kj})^2 + \mathrm{E}[\epsilon_j^2]. \end{aligned}$$

Thus,

$$\mathrm{E}[\bar{x}_j] = b_j \quad \text{and} \quad \mathrm{E}[\bar{Y}_{jj}] = \|b_j\|^2 + \mathrm{E}[\epsilon_j^2] \ \forall \ j$$

and

$$\mathrm{E}[\bar{Y}_{ij}] = (b_i)^T b_j \ \forall \ i \neq j$$

is the solution satisfying these equations.

5.3 General SDP theory on graph realization

In the previous section, we show that if all accurate distances are given, then the graph can be localized by solving an SDP problem in polynomial time. In real applications, only partial distances(or edges) are known to the model. Can these graph be localized or realized in polynomial time? In this section, we given a positive answer to a family of *uniquely localizable* graphs. Informally, a graph is uniquely localizable in dimension d if (i) it has a unique realization in R^d, and (ii) it does not have any nontrivial realization whose affine span is R^h, where $h > d$. Specifically, we present an SDP model that guarantees to find the unique realization in polynomial time when the input graph is uniquely localizable. The proof employs SDP duality theory and properties of interior–point algorithms for SDP. To the best of our knowledge, this is the first time such a theoretical guarantee is proven for a general localization algorithm. Moreover, in view of the hardness result of Aspnes et. al. [25], The result is close to be the best possible in terms of identifying the largest family of efficiently realizable graphs. We also introduce the concept of strong localizability. Informally, a graph is strongly localizable if it is uniquely localizable and remains so under slight perturbations. We show that the SDP model will identify all the strongly localizable subgraphs in the input graph.

5.3.1 Preliminaries

For the simplification of our analyses, we ignore the lower bound constraints and study the *Graph Localization* problem that is defined as follows: find a realization of $x_1, \ldots, x_n \in R^d$ such that

$$\begin{aligned} \|a_k - x_j\|^2 &= \bar{d}_{kj}^2 \quad \forall\ (k,j) \in N_a \\ \|x_i - x_j\|^2 &= d_{ij}^2 \quad \forall\ (i,j) \in N_x \end{aligned} \tag{5.8}$$

Recall that m *anchor* points $a_1, \ldots, a_m \in R^d$ whose locations are known, and n *sensor* points $x_1, \ldots, x_n \in R^d$ whose locations we wish to determine. Furthermore, we are given partial accurate Euclidean distance values $\bar{d}_{kj}$ between a_k and x_j for $(k,j) \in N_a$, and d_{ij} between x_i and x_j for some $(i < j, j) \in N_x$.

Again, we can write the relaxed problem as a standard SDP problem, namely, find a symmetric matrix $Z \in R^{(d+n)\times(d+n)}$ to:

$$\begin{aligned} \text{maximize} \quad & 0 \\ \text{subject to} \quad & Z_{1:d,1:d} = I_d \\ & (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^T \bullet Z = d_{ij}^2 \quad \forall\ (i,j) \in N_x \\ & (a_k; e_j)(a_k; e_j)^T \bullet Z = \bar{d}_{kj}^2 \quad \forall\ (k,j) \in N_a \\ & Z \succeq 0 \end{aligned} \tag{5.9}$$

where $Z_{1:d,1:d}$ is the $d \times d$ principal submatrix of Z. Note that this formulation forces any possible feasible solution matrix to have rank at least d.

The dual of the SDP relaxation is given by:

$$\begin{aligned} \text{minimize} \quad & I_d \bullet V + \sum_{(i,j)\in N_x} y_{ij} d_{ij}^2 + \sum_{(k,j)\in N_a} w_{kj} \bar{d}_{kj}^2 \\ \text{subject to} \quad & \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{(i,j)\in N_x} y_{ij} (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^T \\ & + \sum_{(k,j)\in N_a} w_{kj} (a_k; e_j)(a_k; e_j)^T \succeq 0 \end{aligned} \tag{5.10}$$

Note that the dual is always feasible, as $V = 0$, $y_{ij} = 0$ for all $(i,j) \in N_x$ and $w_{kj} = 0$ for all $(k,j) \in N_a$ is a feasible solution.

5.3.2 Analysis of the SDP relaxation

We now investigate when will the SDP (5.9) have an exact relaxation, i.e. when will the solution matrix Z have rank d. Suppose that problem (5.9) is feasible. This occurs when, for instance, $\bar{d}_{kj}$ and d_{ij} represent exact distance values for the positions $\bar{X} = [\bar{x}_1\ \bar{x}_2\ \ldots\ \bar{x}_n]$. Then, the matrix $\bar{Z} = (I_d; \bar{X})^T (I_d; \bar{X})$ is a feasible solution for (5.9). Now, since the primal is feasible, the minimal value of the dual must be 0, i.e. there is no duality gap between the primal and dual.

Let U be the $(d+n)$–dimensional dual slack matrix, i.e.:

$$\begin{aligned} U \quad = \quad & \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{(i,j)\in N_x} y_{ij} (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^T \\ & + \sum_{(k,j)\in N_a} w_{kj} (a_k; e_j)(a_k; e_j)^T \end{aligned}$$

Then, from the duality theorem for SDP (see, e.g., [10]), we have:

Theorem 5.3 *Let $\bar{Z}$ be a feasible solution for (5.9) and $\bar{U}$ be an optimal slack matrix of (5.10). Then,*

1. *complementarity condition holds: $\bar{Z} \bullet \bar{U} = 0$ or $\bar{Z}\bar{U} = \mathbf{0}$;*
2. *rank$(\bar{Z})$ + rank$(\bar{U}) \leq d + n$;*
3. *rank$(\bar{Z}) \geq d$ and rank$(\bar{U}) \leq n$.*

An immediate result from the theorem is the following:

Corollary 5.4 *If an optimal dual slack matrix has rank n, then every solution of (5.9) has rank d. That is, problems (5.8) and (5.9) are equivalent and (5.8) can be solved as an SDP in polynomial time.*

Another technical result is the following:

Proposition 5.5 *If every sensor point is connected, directly or indirectly, to an anchor point in (5.8), then any solution to (5.9) must be bounded, that is, Y_{jj} is bounded for all $j = 1, \ldots, n$.*

Proof. If sensor point x_j is connected to an anchor point a_k, then we have:

$$\|x_j\|^2 - 2a_k^T x_j + \|a_k\|^2 \leq Y_{jj} - 2a_k^T x_j + \|a_k\|^2 = \bar{d}_{kj}^2$$

so that from the triangle inequality $\|x_j\|$ in (5.9) is bounded. Hence, we have:

$$Y_{jj} \leq \bar{d}_{kj}^2 + 2\|a_k\|\|x_j\| - \|a_k\|^2$$

Furthermore, if x_i is connected to x_j and Y_{jj} is bounded, we have:

$$Y_{ii} - 2\sqrt{Y_{ii}Y_{jj}} + Y_{jj} \leq Y_{ii} - 2Y_{ij} + Y_{jj} = d_{ij}^2$$

so that from the triangle inequality Y_{ii} must be also bounded.

In general, a primal (dual) max–rank solution is a solution that has the highest rank among all solutions for primal (5.9) (dual (5.10)). It is known [149, 132, 212] that various path–following interior–point algorithms compute the max–rank solutions for both the primal and dual in polynomial time. This motivates the following definition.

Definition 5.1 *Problem (5.8) is uniquely localizable if there is a unique localization $\bar{X} \in R^{d\times n}$ and there is no $x_j \in R^h$, $j = 1, \ldots, n$, where $h > d$, such that:*

$$\begin{array}{ll} \|(a_k; \mathbf{0}) - x_j\|^2 = \bar{d}_{kj}^2 & \forall\ (k,j) \in N_a \\ \|x_i - x_j\|^2 = d_{ij}^2 & \forall\ (i,j) \in N_x \\ x_j \neq (\bar{x}_j; \mathbf{0}) & \text{for some } j \in \{1, \ldots, n\} \end{array}$$

The latter says that the problem cannot have a non–trivial localization in some higher dimensional space R^h (i.e. a localization different from the one obtained by setting $x_j = (\bar{x}_j; \mathbf{0})$ for $j = 1, \ldots, n$), where anchor points are augmented to $(a_k; \mathbf{0}) \in R^h$, for $k = 1, \ldots, m$.

We now develop the following theorem:

Theorem 5.6 *Suppose that the network is connected. Then, the following statements are equivalent:*

1. *Problem (5.8) is uniquely localizable.*
2. *The max–rank solution matrix of (5.9) has rank d.*
3. *The solution matrix of (5.9) satisfies $Y = X^T X$.*

Proof. The equivalence between 2. and 3. is straightforward.

Now, since any rank d solution of (5.9) is a solution to (5.8), from 2. to 1. we need to prove that if the max–rank solution matrix of (5.9) has rank d then it is unique. Suppose not, i.e., (5.9) has two rank–d feasible solutions:

$$Z_1 = \begin{pmatrix} I_d & X_1 \\ X_1^T & X_1^T X_1 \end{pmatrix} \text{ and } Z_2 = \begin{pmatrix} I_d & X_2 \\ X_2^T & X_2^T X_2 \end{pmatrix}$$

Then, the matrix $Z = \alpha Z_1 + \beta Z_2$, where $\alpha + \beta = 1$ and $\alpha, \beta > 0$ is a feasible solution and its rank must be d, since all feasible solution of (5.9) has rank at least d but the max–rank is assumed to be d. Therefore, we have:

$$\begin{aligned} Z &= \begin{pmatrix} I_d & \alpha X_1 + \beta X_2 \\ \alpha X_1^T + \beta X_2^T & \alpha X_1^T X_1 + \beta X_2^T X_2 \end{pmatrix} \\ &= \begin{pmatrix} I_d & B \\ B^T & B^T B \end{pmatrix} \end{aligned}$$

where $B = \alpha X_1 + \beta X_2$. It follows that $(X_1 - X_2)^T (X_1 - X_2) = \mathbf{0}$, or $\|X_1 - X_2\| = 0$, i.e. $Z_1 = Z_2$, which is a contradiction.

Next, we prove the direction from 1. to 2., that is, the rank of a max–rank solution of (5.9) is d. Suppose that there is a feasible solution Z of (5.9) whose rank is greater than d. Then, we must have $Y \succeq X^T X$ and $Y \neq X^T X$. Thus, we have the decomposition $Y - X^T X = (X')^T X'$, where $X' = [x'_1, \ \ldots, \ x'_n] \in R^{r \times n}$ and r is the rank of $Y - X^T X$. Now, consider the point:

$$\tilde{x}_j = \begin{pmatrix} x_j \\ x'_j \end{pmatrix} \in R^{d+r} \qquad \text{for } j = 1, \ldots, n$$

Then, we have:

$$\|\tilde{x}_j\|^2 = Y_{jj}, \ (\tilde{x}_i)^T \tilde{x}_j = Y_{ij} \qquad \forall \ i, j;$$

and there exist at least one $\tilde{x}_j$ such that $\|\tilde{x}_j\| \neq \|x_j\|$ or $x'_j \neq 0$. Moreover, since the network is connected, we conclude from Proposition 5.5 that Y_{ii} and Y_{ij} are bounded for all i, j. Hence, we have:

$$\begin{aligned} \|(a_k; \mathbf{0}) - \tilde{x}_j\|^2 &= \bar{d}_{kj}^2 \qquad \forall \ (k, j) \in N_a \\ \|\tilde{x}_i - \tilde{x}_j\|^2 &= d_{ij}^2 \qquad \forall \ (i, j) \in N_x \end{aligned}$$

In other words, $\tilde{x}_j$ is a localization of problem (5.8) in R^{d+r}, which is a contradiction.

Theorem 5.6 establishes, for the first time, that as long as problem (5.8) is uniquely localizable, then the realization can be computed in polynomial time by solving the SDP relaxation. Conversely, if the relaxation solution computed by an interior–point algorithm (which generates max–rank feasible solutions) has rank d (and hence $Y = X^T X$), then X is the unique realization of problem (5.8). Moreover, as the recent result of Aspnes et. al. [25] shows, the results of Theorem 5.6 are close to be the best possible, since the problem of computing a realization of the sensors on the plane is NP–complete in general, even when the instance has a unique solution on the plane.

5.3.3 Strongly Localizable Problem

Although unique localizability is an useful notion in determining the solvability of the Sensor Network Localization problem, it is not stable under perturbation. As we shall see in Section 5.3.4, there exist networks which are uniquely localizable, but may no longer be so after small perturbation of the sensor points. This motivates us to define another notion called strong localizability.

Definition 5.2 *We say problem (5.8) is strongly localizable if the dual of its SDP relaxation (5.10) has an optimal dual slack matrix with rank n.*

Note that if a network is strongly localizable, then it is uniquely localizable from Theorems 5.3 and 5.6, since the rank of all feasible solution of the primal is d.

We show how we can construct a rank–n optimal dual slack matrix. First, note that if U is an optimal dual slack matrix of rank n, then it can be written in the form $U = (-X^T; I_n)^T W(-X^T; I_n)$ for some positive definite matrix W of rank n. Now, consider the dual matrix U. It has the form:

$$U = \begin{pmatrix} U_{11} & U_{12} \\ U_{12}^T & U_{22} \end{pmatrix}$$

where U_{22} is an $n \times n$ matrix. Moreover, it can be decomposed as $U_{22} = A + D$, where $A_{ij} = y_{ij}$ if $i \neq j$, $A_{ii} = -\sum_j A_{ij}$; and D is a diagonal matrix where $D_{ii} = -\sum_{(k,i)\in N_a} w_{ki}$. (If there is no $(k,i) \in N_a$, then $D_{ii} = 0$.) Note that if we impose the constraints $y_{ij} \leq 0$ and $w_{ki} \leq 0$, then both A and D are positive semidefinite. Moreover, we have the following:

Proposition 5.7 *Suppose that the network is connected. Furthermore, suppose that $y_{ij} < 0$ for all $(i,j) \in N_x$, and that $w_{ki} < 0$ for some $(k,i) \in N_a$, with $N_a \neq \emptyset$. Then, U_{22} is positive definite, i.e. it has rank n.*

Proof. Since A and D are positive semidefinite, we have $x^T U_{22} x \geq 0$ for all $x \in R^n$. We now show that there is no $x \in R^n \backslash \{0\}$ such that $x^T Ax = x^T Dx = 0$. Suppose to the contrary that we have such an x. Then, since D is diagonal, we have $x^T Dx = \sum_{i=1}^n D_{ii} x_i^2 = 0$. In particular, for $D_{ii} > 0$, we have $x_i = 0$. Now, note that:

$$x^T Ax = \sum_{i=1}^n \sum_{j=1}^n x_i x_j A_{ij} = -\sum_{i<j} (x_i - x_j)^2 A_{ij}$$

Thus, $x^T Ax = 0$ implies that $x_i = x_j$ for all i, j. Since $N_a \neq \emptyset$, there exists an i such that $D_{ii} > 0$, whence $x_i = 0$. It follows that $x = 0$. Proposition 5.7 gives us a recipe for putting U into the desired form. First, we set U_{22} to be a positive definite matrix. Then, we need to set $U_{12} = -\bar{X} U_{22}$, where $\bar{X}$ is the matrix containing the true locations of the sensors. We now investigate when this is possible. Note that the above condition is simply a system of linear equations. Let A_i be the set of sensors connected to anchor i, and let E be the number of sensor–sensor edges. Then, the above system has $E + \sum_i |A_i|$ variables. The number of equations is $E + 3m$, where m is the number of sensors that are connected to some anchors. Hence, a sufficient condition for solvability is that the system of equations are linearly independent, and that $\sum_i |A_i| \geq 3m$. In particular, this shows that the trilateration graphs defined in [91] are strongly localizable.

We now develop the next theorem.

Theorem 5.8 *If a problem (graph) contains a subproblem (subgraph) that is strongly localizable, then the submatrix solution corresponding to the subproblem in the SDP solution has rank d. That is, the SDP relaxation computes a solution that localizes all possibly localizable unknown points.*

Proof. Let the subproblem have n_s unknown points and they are indexed as $1, \ldots, n_s$. Since it is strongly localizable, an optimal dual slack matrix U_s of the SDP relaxation for the subproblem has rank n_s. Then in the dual problem of the SDP relaxation for the whole problem, we set V and those $w.$ associated with the subproblem to the optimal slack matrix U_s and set all other $w.$ equal 0. Then, the slack matrix:

$$U = \begin{pmatrix} U_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \succeq \mathbf{0}$$

must be optimal for the dual of the (whole–problem) SDP relaxation, and it is complementary to any primal feasible solution of the (whole–problem) SDP relaxation:

$$Z = \begin{pmatrix} Z_s & * \\ * & * \end{pmatrix} \succeq \mathbf{0} \quad \text{where} \quad Z_s = \begin{pmatrix} I_d & X_s \\ X_s^T & Y_s \end{pmatrix}$$

However, we have $0 = Z \bullet U = Z_s \bullet U_s$ and $U_s, Z_s \succeq 0$. The rank of U_s is n_s implies that the rank of Z_s is exactly d, i.e. $Y_s = (X_s)^T X_s$, so X_s is the unique realization of the subproblem.

5.3.4 A Comparison of Notions

In this section, we will show that the notions of unique localizability, strong localizability and rigidity in R^2 are all distinct.

5.3.5 Unique Localizability $\not\Rightarrow$ Strong Localizability

We have already remarked earlier that a strongly localizable graph is necessarily uniquely localizable. However, as we shall see, the converse is not true.

Let G_1 be the network shown in Figure 5.1(a). The key feature of G_1 is that the sensor x_2 lies on the line joining anchors a_1 and a_3. It is not hard to check that this network is uniquely localizable. Now, suppose to the contrary that G_1 is strongly localizable. Then, the dual slack matrix U admits the decomposition $U = (-\bar{X}^T, I)^T W (-\bar{X}^T, I)$. It is easy to verify that:

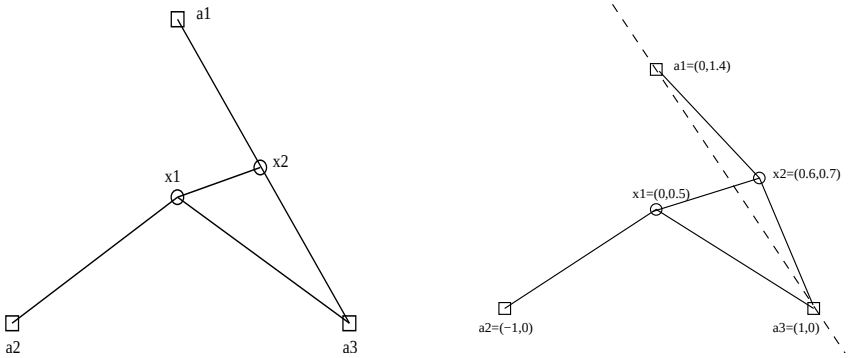
$$\begin{aligned} U_{12} &= (\bar{y}_{21} a_2 + \bar{y}_{31} a_3, \bar{y}_{12} a_1 + \bar{y}_{32} a_3) \\ U_{22} &= \begin{pmatrix} -(\bar{y}_{21} + \bar{y}_{31}) - y_{12} & y_{12} \\ y_{12} & -(\bar{y}_{12} + \bar{y}_{32}) - y_{12} \end{pmatrix} \end{aligned}$$

and the form of U requires that $U_{12} = -\bar{X} U_{22}$. This is equivalent to the following system of equations:

$$(\bar{x}_1 - a_2)\bar{y}_{21} + (\bar{x}_1 - a_3)\bar{y}_{31} = (\bar{x}_1 - \bar{x}_2) y_{12} \tag{5.11}$$

$$(\bar{x}_2 - a_1)\bar{y}_{12} + (\bar{x}_2 - a_3)\bar{y}_{32} = -(\bar{x}_1 - \bar{x}_2) y_{12} \tag{5.12}$$

Since $\bar{x}_2$ lies on the affine space spanned by a_1 and a_3, equation (5.12) implies that $y_{12} = 0$. However, equation (5.11) would then imply that $\bar{x}_1$ lies on the affine space spanned by a_2 and a_3, which is a contradiction. Thus, we conclude that G_1 is not strongly localizable.



(a) A uniquely localizable, but not strongly localizable network

(b) A rigid network that is not uniquely localizable

Figure 5.1: A comparison of graph notions

5.3.6 Rigid in $R^2 \not\Rightarrow$ Unique Localizability

By definition, a uniquely localizable network is rigid in R^2. However, the converse is not true. To see this, let G_2 be the network shown in Figure 5.1(b).

Note that G_2 can be viewed as a perturbed version of G_1. It is easy to verify that G_2 is rigid. Thus, by Theorem 5.6, it can fail to be uniquely localizable only if it has a realization in some higher dimension. Indeed, the above network has an 3–dimensional realization. The idea for constructing such a realization is as follows. Let us first remove the edge (x_1, x_2). Then, reflect the subgraph induced by a_1, x_2, a_2 across the dotted line. Now, consider two spheres, one centered at a_2 and the other centered at a_3, both having radius $\sqrt{5}/2$. The intersection of these spheres is a circle, and we can move x_1 along this circle until the distance between x_1 and x_2 equals to the prespecified value. Then, we can put the edge (x_1, x_2) back and obtain an 3–dimensional realization of the network.

More precisely, for the above realization, the reflected version of x_2 has coordinates $x_2' = \left(\frac{173}{370}, \frac{112}{185}, 0\right)$. Now, let $x_1' = \left(0, \frac{23}{64}, \frac{\sqrt{495}}{64}\right)$. It is straightforward to verify that:

$$\|x_1 - a_2\|^2 = \|x_1' - a_2\|^2 = \tfrac{5}{4}$$
$$\|x_1 - a_3\|^2 = \|x_1' - a_3\|^2 = \tfrac{5}{4}$$
$$\|x_1 - x_2\|^2 = \|x_1' - x_2'\|^2 = \tfrac{2}{5}$$

Hence, we conclude that G_2 is not uniquely localizable.

It would be nice to have a characterization on those graphs which are rigid in the plane but have higher dimensional realizations. However, finding such a characterization remains a challenging task, as such characterization would necessarily be non–combinatorial, and would depend heavily on the geometry of the network. For instance, the networks shown in Figure 5.2, while having the same combinatorial property as the one shown in Figure 5.1(b), are uniquely localizable (in fact, they are both strongly localizable):

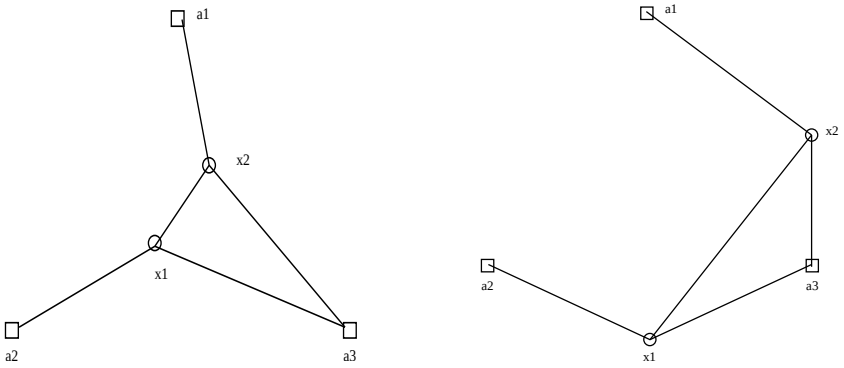


Figure 5.2: Strongly localizable networks

5.3.7 Preliminary computational and simulation results

The computational results presented here were generated using the interior-point algorithm SDP solvers SeDuMi of Sturm and DSDP2.0 of Benson and Ye. The performance of this technique seems highly satisfactory compared to other techniques. Very few anchor nodes are required to accurately estimate the position of all the unknown nodes in a network. Also the estimation errors are minimal even when the anchor nodes are not suitably placed within the network.

Simulations were performed on a network of 50 sensors or nodes randomly generated in a square region of $[-.5\ .5] \times [-.5\ .5]$. The distances between the nodes was then calculated. If the distance between two notes was less than a given *radiorange* between $[0, 1]$, a random error was added to it

$$\hat{d}_{ij} = d_{ij} \cdot (1 + randn(1) * noisyfactor),$$

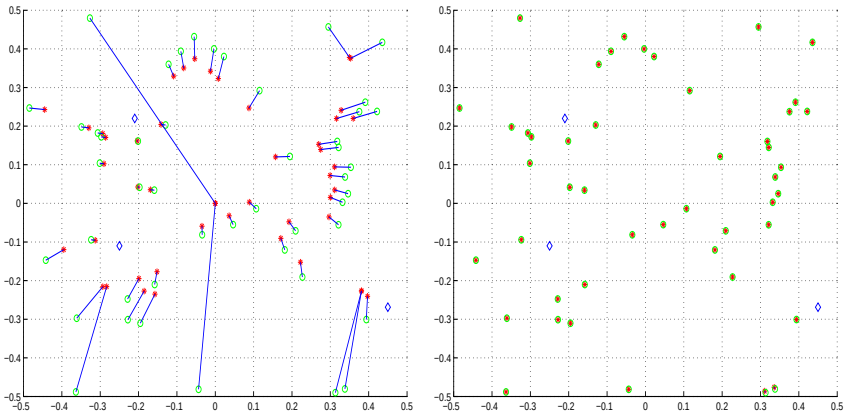
where $noisyfactor$ was a given number between $[0, 1]$, and then both upper and lower bound constraints were applied for that distance in the SDP model. If the distance was beyond the given $radio - range$, only the lower bound constraint,

$\geq 1.001 * radiorange^2$, was applied. The average estimation error is defined by

$$\frac{1}{n} \cdot \sum_{j=1}^{n} \|\bar{x}_j - a_j\|,$$

where $\bar{x}_j$ comes from the SDP solution and a_j is the true position of the jth node. As discussed earlier, we called the trace of $\bar{Y} - \bar{X}^T\bar{X}$ the total trace. Connectivity indicates how many of the nodes, on average, are within the radio range of a sensor.

Also the original and the estimated sensors were plotted. The blue diamond nodes refer to the positions of the anchor nodes, green circle nodes to the true locations of the unknown sensors and red star nodes to their estimated positions from $\bar{X}$. The discrepancies in the positions can be estimated by the offsets between the true and the estimated points as indicated by the solid lines, see Figure 5.3. Even with noisy data measurement, the position estimations for the sensors are fairly accurate. Figure 5.4 shows the correlation between individual offset errors (not observable) and individual traces (observable) of corresponding sensors when the radio range is set at 0.25.

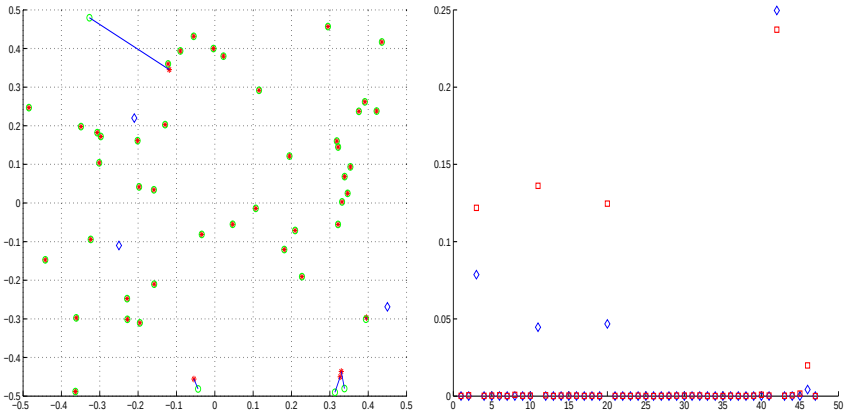


(a) error:0.023, connectivity:7.8, total trace:0.16, radio range=0.20.

(b) error:0.002, connectivity:10.5, total trace:0.04, radio range=0.30.

Figure 5.3: Position estimations with 3 anchors, noisy factor=0, and various radio ranges.

Several key questions have to be answered for the future research: What is the minimal radio range such that the network is localizable? How does the sensor geographical distribution change the minimal radio range? What is the sensitivity of the distance data? How to filter the data noisy? etc.



(a) number of anchors=3, radio range=0.25.

(b) number of anchors=3, radio range=0.30.

Figure 5.4: Diamond: the offset distance between estimated and true positions, Box: the square root of trace $\bar{Y}_{jj} - \|\bar{x}_j\|^2$.

5.4 Other distance geometry problems

We list few other variances of the distance geometry or graph localization problem.

5.4.1 Metric distance embedding

The basic mathematical model of the metric distance embedding problem can be described as a quadratically constrained optimization problem. We are given metric p-norm distances d_{ij} for all $i, j = 1, ..., n$. The problem is to find n unknown points $x_j \in R^n$, $j = 1, ..., n$ such that

$$\|x_i - x_j\|^2 = (d_{ij})^2 \ \forall i \neq j$$

or the error is minimized.

Thus, an optimization problem can be formulated as:

$$\begin{array}{ll} \text{minimize} & \alpha \\ \text{subject to} & (d_{ij})^2 \leq \|x_i - x_j\|^2 \leq \alpha \cdot (d_{ij})^2, \ \forall i < j \\ & x_j \in R^n, j = 1, ..., n. \end{array} \tag{5.13}$$

Its SDP relaxation becomes

$$\begin{array}{lll} \alpha_p^* := & \text{minimize} & \alpha \\ & \text{subject to} & (d_{ij})^2 \leq (e_{ij} e_{ij}^T) \bullet Y \leq \alpha \cdot (d_{ij})^2, \ \forall i < j \\ & & Y \succeq 0. \end{array} \tag{5.14}$$

Again

$$e_{ij} = (0, ..., 1, ..., -1, ...)^T.$$

The SDP dual is

$$\begin{array}{rll} \alpha_p^* := & \text{Maximize} & \sum_{i<j} u_{ij}(d_{ij})^2 \\ & \text{Subject to} & 0 \succeq \sum_{i,j}(u_{ij} - v_{ij}) e_{ij} e_{ij}^T, \\ & & \sum_{i,j} v_{ij}(d_{ij})^2 = 1 \\ & & u_{ij},\ v_{ij} \geq 0. \end{array}$$

Result and questions:

- $\alpha_2^* = 1$, and $\alpha_p^* \leq O(\log n)$ for all p.
- Reduce Y^*'s rank to $O(\log n)$ for $\alpha_2^* = 1 + \epsilon$.
- rank $\leq O(\log n)$ for $\alpha_p^* \leq O(\log n)$ for all p.
- Better bound for α_1^*?

5.4.2 Molecular confirmation

The basic mathematical model of the molecular confirmation problem can be described as a quadratically constrained optimization problem as well. For a pair of $(i, j) \in \mathcal{N}$, we are given Euclidean distance upper bound $\bar{d}_{ij}$ and lower bound $\underline{d}_{ij}$ between points i and j, the problem is to find Y such that

$$(\underline{d}_{ij})^2 \leq (e_{ij} e_{ij}^T) \bullet Y \leq (\bar{d}_{ij})^2,\ \forall i < j \in \mathcal{N}.$$

Furthermore, it is desired to find other missing d_{ij} such that the Euclidean distance matrix can be completed.

The progem can be formulated as an SDP:

$$\begin{array}{ll} \text{minimize} & I \bullet Y \\ \text{subject to} & (\underline{d}_{ij})^2 \leq (e_{ij} e_{ij}^T) \bullet Y \leq (\bar{d}_{ij})^2,\ \forall i < j \in \mathcal{N} \\ & Y \succeq 0. \end{array} \tag{5.15}$$

5.4.3 Euclidean ball parking

The Euclidean ball packing problem is an old mathematical geometry problem with plenty modern applications in Bio-X and chemical structures. It is a special case of the Euclidean distance geometry problem, say $d = 3$, $x_j \in R^3$ is the unknown position of the center of ball j, $\bar{d}_{ij} = \infty$ and $\underline{d}_{ij} = r_i + r_j$, where r_i and r_j are the radii of balls i and j, respectively; and Ω represents a convex container where the balls should be packed.

Say that we want to pack n balls in a box with width and length equal $2R$ and like to minimize the height of the box, we can formulate it as

$$\begin{array}{ll} \text{minimize} & \alpha \\ \text{subject to} & \|x_i - x_j\|^2 \geq (r_i + r_j)^2, \ \forall i \neq j, \\ & -R + r_j \leq x_j(1) \leq R - r_j, \ \forall j, \\ & -R + r_j \leq x_j(2) \leq R - r_j, \ \forall j, \\ & r_j \leq x_j(3) \leq \alpha - r_j, \ \forall j, \end{array} \tag{5.16}$$

The first constraint tells that the distance between any two balls should be greater than or equal to the sum of the two balls' radii, the second says that the first coordinate of x_j is within the width constraint and the third is for the length requirement, and the fourth indicates that the ball center has to be above r_j from the bottom and its height is below $\alpha - r_j$ where α is to be minimized.

5.4.4 Data dimensionality reduction

Given P, a data point set of $p_1, ..., p_n \in R^d$, a fundamental question is how to embed P into Q of $q_1, ..., q_n \in R^k$, where $k \ll d$, such that q_js keep all essential information of P, such as the norms, distances and angles between p_js. In other words, find a $d-k$-dimension subspace such that the projections of p_js onto the subspace has a minimal "information loss." We believe that this problem can be posted as an Euclidean distance geometry problem

$$\begin{array}{lll} L_k := & \text{minimize} & \alpha \\ & \text{subject to} & \sum_{i=1}^{n-k}((p_l)^T x_i)^2 \leq \alpha, \ \forall l, \\ & & \sum_{i=1}^{n-k}((p_l - p_m)^T x_i)^2 \leq \alpha, \ \forall l < m, \\ & & \|x_i\|^2 = 1, \ i = 1, ..., n-k, \quad (x_h)^T x_i = 0, \ \forall h < i. \end{array} \tag{5.17}$$

Here, $x_1, ..., x_{n-k}$ represent the $n-k$ orthogonal bases of a $n-k$-dimension subspace. Thus, $(p_l^T x_i; ...; p_l^T x_{n-k})$ represents the projection of p_l onto the $n-k$ subspace. After the problem being solved and the subspace is found, the projections of p_ls onto the complement k-dimension subspace must have all remaining information of p_ls. In fact, the model without pair-wise distance constraints is called the minimum radius problem, which is a fundamental computational geometry and statistics problem.

Assume that $x_1, x_2, \cdots, x_{n-k} \in R^d$ are the optimal solution of (5.17). Then one can easily verify that the matrix $Y = x_1 x_1^T + x_2 x_2^T + \cdots + x_{n-k} x_{n-k}^T$ is a feasible solution for the following SDP model:

$$\begin{array}{lll} \alpha_k^* := & \text{minimize} & \alpha \\ & \text{subject to} & p_l^T Y p_l \leq \alpha, \ \forall l, \\ & & (p_l - p_m)^T Y (p_l - p_m) \leq \alpha, \ \forall l < m, \\ & & \text{tr}(Y) = n - k, \quad I - Y \succeq 0, \quad Y \succeq 0. \end{array} \tag{5.18}$$

It follows that $\alpha_k^* \leq R_k(P)^2$. Then, we can design a rounding procedure to round $\bar{Y}$, a solution to (5.18), into orthogonal $n-k$ bases and prove their quality.

Theorem 5.9 *We can computed in polynomial time, a $(n-k)$-subspace such that, with probability at least $1-\frac{2}{n}$, its projected value of (5.17) is at most $12\log(n)$ of the minimal one.*

5.5 A special case: The k-radius of P

The outer-radius, $R_k(P)$, is the minimum of the Euclidean radius of P projected onto a k-dimensional subspace. Computing the minimal radius is a fundamental problem in computational convexity with applications in global optimization, data mining, statistics and clustering, and has received considerable attention in the computational geometry literature [62, 155].

The square of $R_k(P)$ can be defined by the optimal value of the following Euclidean distance minimization:

$$\begin{array}{rlll} R_k(P)^2 := & \text{minimize} & \alpha & \\ & \text{subject to} & \sum_{i=1}^{k}((p_l)^T x_i)^2 \le \alpha, \ \forall p_l \in P, & \\ & & \|x_i\|^2 = 1, \ i=1,...,k, \quad (x_i)^T x_j = 0, \ \forall i \ne j \le k. & \end{array} \tag{5.19}$$

This leads to the following SDP relaxation:

$$\begin{array}{rlll} \alpha_k^* := & \text{minimize} & \alpha & \\ & \text{subject to} & \text{tr}(p_l p_l^T X) \ \le \alpha, \ \forall p_l \in P, & \\ & & \text{tr}(X) = k, \quad I - X \succeq 0, \quad X \succeq 0. & \end{array} \tag{5.20}$$

It follows that $\alpha_k^* \le R_k(P)^2$.

Lemma 5.10 *There exists an integer $r \ge k$ such that we can compute, in polynomial time, r nonnegative reals $\lambda_1, \lambda_2, \cdots, \lambda_r$ and r orthogonal unit vectors $v_1, v_2, \cdots, v_r$ such that*

(i). $\sum_{i=1}^{r} \lambda_i = k$.

(ii). $\max_{1 \le i \le r} \lambda_i \le 1$.

(iii). $\sum_{i=1}^{r} \lambda_i \langle p, v_i \rangle^2 \le R_k(P)^2$, *for any $p \in P$.*

Proof. We solve the semidefinte program (5.20), and let X^* be an optimal solution of (5.20). We claim that the rank of X^*, say r, is at least k. This follows from the fact that $\text{tr}(X^*) = k$ and $I - X^* \succeq 0$. In other words, $\text{tr}(X^*) = k$ implies that the sum of the eigenvalues of X^* is equal to k, and $I - X^* \succeq 0$ implies that the all eigenvalues are less than or equal to 1. Therefore, X^* has at least k non-zero eigenvalues, which implies that the rank of X^* is at least k. Let $\lambda_1, \lambda_2, \cdots, \lambda_r$ be the r nonnegative eigenvalues and $v_1, v_2, \cdots, v_r$ be the corresponding eigenvectors. Then we have $\sum_{i=1}^{r} \lambda_i = k$ and $\max_{1 \le i \le r} \lambda_i \le 1$. Furthermore, for any $p \in P$,

$$\sum_{i=1}^{r} \lambda_i \langle p, v_i \rangle^2 = \text{tr}(pp^T \sum_{i=1}^{r} \lambda_i v_i v_i^T) = \text{tr}(pp^T X^*) \le \alpha_k^* \le R_k(P)^2.$$

∎

5.5.1 Deterministic First Rounding

In this section, we prove a lemma concerning how to deterministically group the eigenvalues and their eigenvectors. The proof of the lemma is elementary but it plays an important role for proving our main result.

Lemma 5.11 *The index set* $\{1, 2, \cdots, r\}$ *can be partitioned into* k *sets* $I_1, I_2, \cdots, I_k$ *such that*

(i). $\cup_{i=1}^{k} I_i = \{1, 2, \cdots, r\}$, *and for any* $i \neq j$, $I_i \cap I_j = \emptyset$.

(ii). *For any* $i : 1 \leq i \leq k$, $\sum_{j \in I_i} \lambda_j \geq \frac{1}{2}$.

Proof. Recall that $\sum_{j=1}^{r} \lambda_j = k$ and $0 \leq \lambda_j \leq 1$ for all j. Without loss of generality, we can assume that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r$. Our partitioning algorithm is the same as the Longest-Processing-Time heuristic algorithm for parallel machine scheduling problem. The algorithm works as follows:

STEP 1. For $i = 1, 2, \cdots, k$, set $I_i = \emptyset$ and let $L_i = 0$. Let $I = \{1, 2, \cdots, r\}$.

STPE 2. While $I \neq \emptyset$

choose j from I with the smallest index;

choose set i with the smallest value L_i;

Let $I_i := I_i \cup \{j\}$, $L_i := L_i + \lambda_j$ and $I := I - \{j\}$.

It is clear that when the algorithm stops, the sets $I_1, I_2, \cdots, I_k$ satisfy condition (i). Now we prove condition (2) by contradiction. Assume that there exists some t such that $\sum_{j \in I_t} \lambda_j < \frac{1}{2}$.

We now claim that, for all i, $\sum_{j \in I_i} \lambda_j \leq 1$. Otherwise, suppose $\sum_{j \in I_{t'}} \lambda_j > 1$ for some t'. Note that $\lambda_j \leq 1$ for every j and thus there are at least two eigenvalues are assigned to $I_{t'}$. Denote the last eigenvalue by $\lambda_{s'}$. It follows that $\sum_{j \in I_{t'}} \lambda_j - \lambda_{s'} = \sum_{j \in I_{t'} \setminus \{s'\}} \lambda_j \leq \sum_{j \in I_t} \lambda_j$ since, otherwise, we would have not assigned $\lambda_{s'}$ to $I_{t'}$ in the algorithm. However, since $\sum_{j \in I_t} \lambda_j < \frac{1}{2}$, we must have $\sum_{j \in I_{t'}} \lambda_j - \lambda_{s'} = \sum_{j \in I_{t'} \setminus \{s'\}} \lambda_j < \frac{1}{2}$. Thus, $\lambda_{s'} > \sum_{j \in I_{t'}} \lambda_j - \frac{1}{2} > \frac{1}{2}$. This is impossible since $\lambda_{s'}$ is the last eigenvalue assigned to $I_{t'}$, which implies $\lambda_{s'} \leq \lambda_j$ for every $j \in I_{t'}$, and we have already proved that there must exist an l such that $s' \neq l \in I_{t'}$ and $\lambda_l \leq \sum_{j \in I_{t'} \setminus \{s'\}} \lambda_j < \frac{1}{2}$. Therefore, $\sum_{j \in I_i} \lambda_j \leq 1$ for all i, and in particular $\sum_{j \in I_t} \lambda_j < \frac{1}{2}$. It follows that $\sum_{i=1}^{k} \sum_{j \in I_i} \lambda_j < k$. However, we know that, by condition (i), $\sum_{i=1}^{k} \sum_{j \in I_i} \lambda_j = \sum_{j=1}^{r} \lambda_j = k$. This results a contradiction. Therefore, such t does not exists and we have proved condition (ii). ∎

Notice that the running time of the partitioning algorithm is bounded by $O(r \cdot k)$. An alternative way of partitioning the eigenvalues is the following: First, put the eigenvalues that are greater than or equal to $1/2$ into distinct subsets. If the number of such eigenvalues, say l, is not less than k, then we are done. Otherwise, arbitrarily put the remaining eigenvalues into $k - l$ subsets such that the sum of eigenvalues in each subset is greater than or equal to $1/2$. This method is suggested by an anonymous referee.

5.5.2 Randomized Second Rounding

Assume now that we have found $I_1, I_2, \cdots, I_k$. Then our next randomized rounding procedure works as follows.

STEP 1. Generate a r dimensional random vector ϕ such that each entry of ϕ takes value, independently, -1 or 1 with probability $\frac{1}{2}$ each way.

STEP 2. For $i = 1, 2, \cdots, k$, let

$$x_i = \frac{\sum_{j \in I_i} \phi_j \sqrt{\lambda_j} \cdot v_j}{\sqrt{\sum_{j \in I_i} \lambda_j}}.$$

The following Lemmas show that $x_1, x_2, \cdots, x_k$ form a feasible solution for the original problem. In other words, they are k orthogonal unit vectors.

Lemma 5.12 *For $i = 1, 2, \cdots, k$, $\|x_i\| = 1$.*

Proof. Recall that $\langle v_l, v_j \rangle = 0$ for any $l \neq j$ and $\|v_j\| = 1$. By definition,

$$\begin{aligned}
& \|x_i\|^2 = \langle x_i, x_i \rangle \\
= & \left\langle \frac{\sum_{j \in I_i} \phi_j \sqrt{\lambda_j} v_j}{\sqrt{\sum_{j \in I_i} \lambda_j}}, \frac{\sum_{j \in I_i} \phi_j \sqrt{\lambda_j} v_j}{\sqrt{\sum_{j \in I_i} \lambda_j}} \right\rangle \\
= & \frac{1}{\sum_{j \in I_i} \lambda_j} \left\langle \sum_{j \in I_i} \phi_j \sqrt{\lambda_j} v_j, \sum_{j \in I_i} \phi_j \sqrt{\lambda_j} v_j \right\rangle \\
= & \frac{1}{\sum_{j \in I_i} \lambda_j} \sum_{j \in I_i} \langle \phi_j \sqrt{\lambda_j} v_j, \phi_j \sqrt{\lambda_j} v_j \rangle \\
= & \frac{1}{\sum_{j \in I_i} \lambda_j} \sum_{j \in I_i} \|\phi_j \sqrt{\lambda_j} v_j\|^2 \\
= & \frac{1}{\sum_{j \in I_i} \lambda_j} \sum_{j \in I_i} (\phi_j)^2 \lambda_j \|v_j\|^2 \\
= & \frac{1}{\sum_{j \in I_i} \lambda_j} \sum_{j \in I_i} \lambda_j \\
= & 1
\end{aligned}$$

∎

Lemma 5.13 *If $s \neq t$ then $\langle x_s, x_t \rangle = 0$.*

Proof. The proof is similar as that of Lemma 5.12.

$$\begin{aligned} & \langle x_s, x_t \rangle \\ = & \left\langle \frac{\sum_{j\in I_s} \phi_j \sqrt{\lambda_j} v_j}{\sqrt{\sum_{j\in I_s} \lambda_j}}, \frac{\sum_{j\in I_t} \phi_j \sqrt{\lambda_j} v_j}{\sqrt{\sum_{j\in I_t} \lambda_j}} \right\rangle \\ = & \frac{1}{\sqrt{\sum_{j\in I_s} \lambda_j \cdot \sum_{j\in I_t} \lambda_j}} \left\langle \sum_{j\in I_s} \phi_j \sqrt{\lambda_j} v_j, \sum_{j\in I_t} \phi_j \sqrt{\lambda_j} v_j \right\rangle \\ = & 0. \end{aligned}$$

The last equality holds since for any $j \in I_s$ and $l \in I_t$, $\langle v_j, v_l \rangle = 0$. ∎

Now we establish a bound on the performance of our algorithm. First, let us introduce Bernstein's Theorem (see, e.g., [240]), which is a form of the Chernoff Bound.

Lemma 5.14 *Let ϕ be a random vector whose entries are independent and either 1 or -1 with probability .5 each way. Then, for any vector e and $\beta > 0$,*

$$\text{prob}\{\langle \phi, e \rangle^2 > \beta \|e\|^2\} < 2 \cdot \exp(-\frac{\beta}{2}).$$

Let $C_{ip} = \sum_{j\in I_i} \lambda_j \langle p, v_j \rangle^2$. Then we have

Lemma 5.15 *For each $i = 1, 2, \cdots, k$ and each $p \in P$, we have*

$$\text{prob}\{\langle p, x_i \rangle^2 > 12 \log(n) \cdot C_{ip}\} < \frac{2}{n^3}.$$

Proof. Given i and p, define a $|I_i|$ dimensional vector e such that its entries are $\sqrt{\lambda_j}\langle p, v_j \rangle$, $j \in I_i$, respectively. Furthermore, we define the vector $\phi|_{I_i}$ whose entries are those of ϕ with indices in I_i. First notice that

$$\|e\|^2 = \sum_{j\in I_i} (\sqrt{\lambda_j}\langle p, v_j \rangle)^2 = \sum_{j\in I_i} \lambda_j \cdot \langle p, v_j \rangle^2 = C_{ip}.$$

On the other hand,

$$\begin{aligned}
& \langle p, x_i\rangle^2 \\
= & \left\langle p, \frac{\sum_{j\in I_i}\sqrt{\lambda_j}v_j\phi_j}{\sqrt{\sum_{j\in I_i}\lambda_j}}\right\rangle^2 \\
= & \frac{1}{\sum_{j\in I_i}\lambda_j}\left\langle p, \sum_{j\in I_i}\sqrt{\lambda_j}v_j\phi_j\right\rangle^2 \\
\leq & 2\left\langle p, \sum_{j\in I_i}\sqrt{\lambda_j}v_j\phi_j\right\rangle^2 \qquad (\text{since}\sum_{j\in I_i}\lambda_j \geq \frac{1}{2}) \\
= & 2\left(\sum_{j\in I_i}\sqrt{\lambda_j}\phi_j\langle p, v_j\rangle\right)^2 \\
= & 2\left\langle \phi|_{I_i}, e\right\rangle^2
\end{aligned}$$

Thus

$$\text{prob}\{\langle p, x_i\rangle^2 > 12\log(n)C_{ip}\} \leq \text{prob}\{\langle \phi|_{I_i}, e\rangle^2 > 6\log(n)\|e\|^2\}.$$

Therefore, the conclusion of the lemma follows by using Lemma 5.14 and by letting $\beta = 6\log(n)$. ∎

Theorem 5.16 *We can computed in polynomial time, a $(d-k)$-flat such that, with probability at least $1-\frac{2}{n}$, the distance between any point $p \in P$ and F is at most $\sqrt{12\log(n)}\cdot R_k(P)$.*

Proof. For given $i = 1, 2, \cdots, k$ and $p \in P$, consider the event

$$B_{ip} = \{\phi | \langle p, x_i\rangle^2 > 12\log(n)\cdot C_{ip}\}$$

and $B = \cup_{i,p} B_{ip}$. The probability that the event B happens is bounded by

$$\sum_{i,p}\text{prob}\{\langle p, x_i\rangle^2 > 12\log(n)\cdot C_{ip}\} < \frac{2kn}{n^3} \leq \frac{2}{n}.$$

If B does not happen, then for any i and p,

$$\langle p, x_i\rangle^2 \leq 12\log(n)\cdot C_{ip}.$$

Therefore, for each $p \in P$,

$$\sum_{i=1}^{k}\langle p, x_i\rangle^2 \leq 12\log(n)\sum_{i=1}^{k}C_{ip} \leq 12\log(n)\cdot R_k(P)^2.$$

The last inequality follows from Lemma 5.10. This completes the proof by taking F as the flat which is orthogonal to the vectors $x_1, x_2, \cdots, x_k$. ∎

5.6 Distributed SDP computing

The SDP problems presented in this chapter can be solved in a distributed fashion which has not been studied before. Here we describe an iterative distributed SDP computation scheme to for sensor network localization. We first partition the anchors into many clusters according to their physical positions, and assign some sensors into these clusters if a sensor has a direct connection to one of the anchors. We then solve semidefinite programs *independently* at each cluster, and fix those sensors' positions which have high accuracy measures according the SDP computation. These positioned sensors become "ghost anchors" and are used to decide the remaining un-positioned sensors. The distributed scheme then repeats. A round of the distributed computation method is straightforward and intuitive:

1. Partition the anchors into a number of clusters according to their geographical positions. In our implementation, we partition the entire sensor area into a number of equal-sized squares and those anchors in a same square form a regional cluster.

2. Each (unpositioned) sensor sees if it has a direct connection to an anchor (within the communication range to an anchor). If it does, it becomes an unknown sensor point in the cluster to which the anchor belongs. Note that a sensor may be assigned into multiple clusters and some sensors are not assigned into any cluster.

3. For each cluster of anchors and unknown sensors, formulate the error minimization problem for that cluster, and solve the resulting SDP model if the number of anchors is more than 2. Typically, each cluster has less than 100 sensors and the model can be solved efficiently.

4. After solving each SDP model, check the individual trace (5.7) for each unknown sensor in the model. If it is below a predetermined small tolerance, label the sensor as *positioned* and its estimation $\bar{x}_j$ becomes an " anchor". If a sensor is assigned in multiple clusters, we choose the $\bar{x}_j$ that has the smallest individual trace. This is done so as to choose the best estimation of the particular sensor from the estimations provided by solving the different clusters.

5. Consider positioned sensors as anchors and return to Step 1 to start the next round of estimation.

Note that the solution of the SDP problem in each cluster can be carried out at the cluster level so that the computation is highly distributive. The only information that needs to be passed among the neighboring clusters is which of the unknown sensors become positioned after a round of SDP solutions.

In solving the SDP model for each cluster, even if the number of sensors is below 100, the total number of constraints could be in the range of thousands.

However, many of those "bounding away" constraints, i.e., the constraints between two remote points, are inactive or redundant at the optimal solution. Therefore, we adapt an iterative active constraint generation method. First, we solve the problem including only partial equality constraints and completely ignoring the bounding-away inequality constraints to obtain a solution. Secondly we verify the equality and inequality constraints and add those violated at the current solution into the model, and then resolve it with a "warm-start" solution. We can repeat this process until all of the constraints are satisfied. Typically, only about $O(n+m)$ constraints are active at the final solution so that the total number of constraints in the model can be controlled at $O(n+m)$.

5.6.1 Preliminary computational and simulation results

Simulations were performed on networks of $2,000$ to $4,000$ sensor points which are randomly generated in a square region of $[-.5\ .5] \times [-.5\ .5]$. We generally select the first $5-10\%$ of the points as anchors, that is, anchors are also uniformly distributed in the same random manner. The tolerance for labeling a sensor as *positioned* is set at $0.01 \cdot (1 + noisyfactor) \cdot radiorange$. One simulation solves a network localization with $4,000$ sensors, where the entire sensor region is partitioned into 10×10 equal-sized squares, that is, 100 clusters, and the radio range is set at .035. The total solution time for the five round computation on the single Pentium 1.2 GHz and 500 MB PC, excluding computing $\hat{d}_{ij}$, is about four minutes using DSDP2.0. The final solution is ploted in Figure 3.

It is usually the outlying sensors at the boundary or the sensors which do not have many anchors within the radio range that are not estimated in the initial stages of the method. Gradually, as the number of well estimated sensors or 'ghost' anchors grows, more and more of these points are positioned. We have also estimated the same $4,000$ network using noisy data. It is noted that the erroneous points are isolated within particular regions. This clearly indicates that the clustering approach prevents the propagation of errors to other clusters. We feel that our distributed computation techlogy is promising for solving very very large scale distance geometry related problems.

5.7 Notes

The distance geometry problem and its variants arise from applications in various areas, such as molecular conformation, dimensionality reduction, Euclidean ball packing, and more recently, wireless sensor network localization [5, 53, 81, 161, 271, 275]. In the sensor networks setting, the vertices of G correspond to sensors, the edges of G correspond to communication links, and the weights correspond to distances. Furthermore, the vertices are partitioned into two sets – one is the *anchors*, whose exact positions are known (via GPS, for example); and the other is the *sensors*, whose positions are unknown. The goal is to determine the positions of all the sensors. We shall refer to this problem as the *Sensor Network Localization* problem. Note that we can view the Sensor

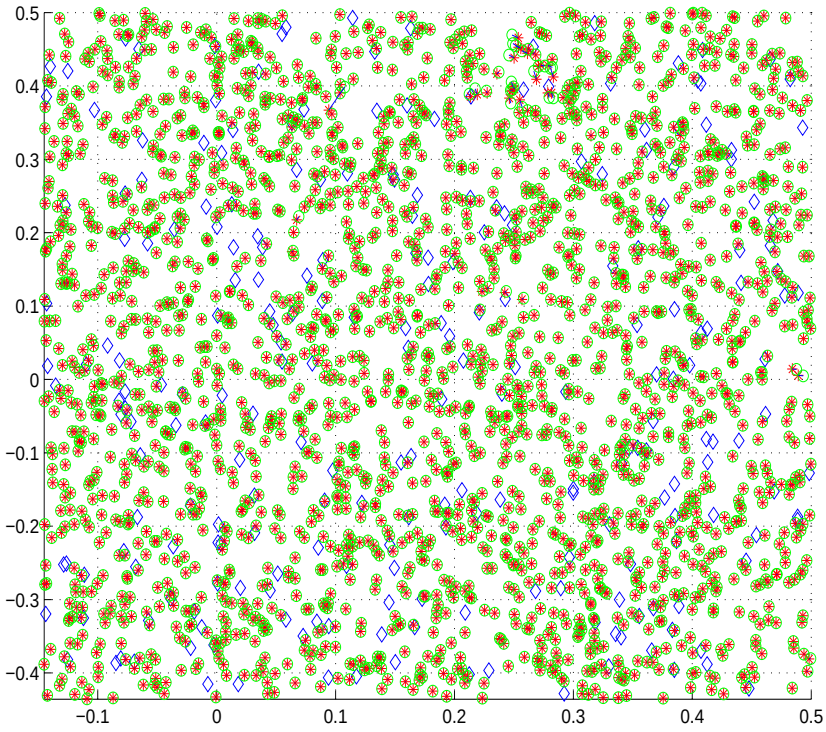


Figure 5.5: Third and final round position estimations in the $4,000$ sensor network, noisy-factor=0, radio-range=0.045, and the number of clusters=100.

Network Localization problem as a variant of the Graph Realization problem in which a subset of the vertices are constrained to be in certain positions.

In many sensor networks applications, sensors collect data that are location–dependent. Thus, another related question is whether the given instance has a unique realization in the required dimension (say, in R^2). Indeed, most of the previous works on the Sensor Network Localization problem fall into two categories – one deals with computing a realization of a given instance [53, 81, 91, 161, 270, 271, 272, 275], and the other deals with determining whether a given instance has a unique realization in R^d using graph rigidity [91, 139]. It is interesting to note that from an algorithmic viewpoint, the two problems above have very different characteristics. Under certain non–degeneracy assumptions, the question of whether a given instance has a unique realization on the plane can be decided efficiently [169], while the problem of computing a realization on the plane is NP–complete in general, even if the given instance has a unique realization on the plane [25]. Thus, it is not surprising that all the aforementioned heuristics for computing a realization of a given instance do not guarantee to find it in the required dimension. On another front, there has been attempts to characterize families of graphs that admit polynomial time algorithms for computing a realization in the required dimension. For instance, Eren et. al. [91] have shown that the family of *trilateration graphs* has such property. (A graph is a trilateration graph in dimension d if there exists an ordering of the vertices $1, \ldots, d+1, d+2, \ldots, n$ such that (i) the first $d+1$ vertices form a complete graph, and (ii) each vertex $j > d+1$ has at least $d+1$ edges to vertices earlier in the sequence.) However, the question of whether there exist larger families of graphs with such property is open.

We should mention here that various researchers have used SDP to study the distance geometry problem (or its variants) before. For instance, Barvinok [35, 36] has studied this problem in the context of quadratic maps and used SDP theory to analyze the possible dimensions of the realization. Alfakih and Wolkowicz [6, 7] have related this problem to the *Euclidean Distance Matrix Completion* problem and obtained an SDP formulation for the former. Moreover, Alfakih has obtained a characterization of rigid graphs in [3] using Euclidean distance matrices and has studied some of the computational aspects of such characterization in [4] using SDP. However, these papers mostly addressed the question of realizability of the input graph, and the analyses of their SDP models only guarantee that they will find a realization whose dimension lies within a certain range. Thus, these models are not quite suitable for our application. In contrast, our analysis takes advantage of the presence of anchors and gives a condition which guarantees that our SDP model will find a realization in the required dimension. We remark that SDP has also been used to compute and analyze distance geometry problems where the realization is allowed to have a certain amount of distortion in the distances [37, 205]. Again, these methods can only guarantee to find a realization that lies in a range of dimensions. Thus, it would be interesting to extend our method to compute low–distortion realizations in a given dimension. For some related work in this direction, see, e.g., [28].

The SDP models were first used by [65, 66] for the ad hoc wireless sensor network localization, and the results described here were based on their work.

The result on the radii of point sets is due to [342], which improves the ratio of [320] by a factor of $O(\sqrt{\log d})$ that could be as large as $O(\sqrt{\log n})$. This ratio also matches the previously best known ratio for approximating the special case $R_1(P)$ of [240], the width of point set P.

5.8 Exercises

5.1 *Find the SDP relaxation to the following minimization problem*

$$\begin{array}{ll} \textit{minimize} & \sum_{i,j\in N_x,\ i<j}(\alpha_{ij})^2 + \sum_{k,j\in N_a}(\alpha_{kj})^2 \\ \textit{subject to} & \|x_i - x_j\|^2 = (\hat{d}_{ij})^2 + \alpha_{ij},\ \forall\ (i,j) \in N_x,\ i<j, \\ & \|a_k - x_j\|^2 = (\hat{d}_{kj})^2 + \alpha_{kj},\ \textit{for}\ (k,j) \in N_a, \\ & \|x_i - x_j\|^2 \geq R^2,\ \textit{for the rest}\ i<j, \\ & \|a_k - x_j\|^2 \geq R^2,\ \textit{for the rest}\ k,j; \end{array}$$

and interpret the objective function.

5.2 *Find the SDP relaxation to the following minimization problem*

$$\begin{array}{ll} \textit{minimize} & \sum_{i,j\in N_x,\ i<j}(\alpha_{ij})^2 + \sum_{k,j\in N_a}(\alpha_{kj})^2 \\ \textit{subject to} & \|x_i - x_j\| = \hat{d}_{ij} + \alpha_{ij},\ \forall\ (i,j) \in N_x,\ i<j, \\ & \|a_k - x_j\| = \hat{d}_{kj} + \alpha_{kj},\ \textit{for}\ (k,j) \in N_a, \\ & \|x_i - x_j\|^2 \geq R^2,\ \textit{for the rest}\ i<j, \\ & \|a_k - x_j\|^2 \geq R^2,\ \textit{for the rest}\ k,j; \end{array}$$

and interpret the objective function.

5.3 *The Un-Folding Problem: with the same notions, consider the problem*

$$\begin{array}{lll} \textit{maximize} & \sum_i \|x_i\|^2 & \\ \textit{subject to} & \|a_k - x_j\|^2 = \bar{d}_{kj}^2 & \forall\ (k,j) \in N_a \\ & \|x_i - x_j\|^2 = d_{ij}^2 & \forall\ (i,j) \in N_x \end{array} \tag{5.21}$$

Find the SDP relaxation and its dual, and explain what they are.

5.4 *Given one sensor and three anchors in R^2, and let the three distances from the sensor to three anchors are given. Show that the graph is stronly localizable if and only if the sensor's position lies in the liner space of the three anchors' positions.*

5.5 *Prove that the graphs depicted in Figures 5.2(a) and 5.2(b) are strongly localizable.*

5.6 *Suppose for the angle of two edges joint at a point is known. How to incoporate this information in the SDP relaxation of the distance geometry problem?*

Chapter 6

SDP for Robust Optimization

6.1 Robust Optimization

Consider an optimization problem

$$\text{(OPT)} \qquad \begin{array}{ll} \text{Minimize} & f(x,\xi) \\ \text{Subject to} & F(x,\xi) \in K \subset R^m. \end{array} \tag{6.1}$$

where ξ is the data of the problem and $x \in R^n$ is the decision vector, and K is a convex cone.

For deterministic optimization, we assume ξ is known and fixed. In reality, ξ may not be certain.

- Knowledge of ξ belonging to a given uncertain set U.
- The constraints must be satisfied for every ξ in the uncertain set U.
- Optimal solution must give the best guaranteed value of $\sup_{\xi \in U} f(x,\xi)$.

This leads to the so-called robust counterpart:

$$\text{(ROPT)} \qquad \begin{array}{lll} \text{Minimize} & \sup_{\xi \in U} f(x,\xi) & \\ \text{Subject to} & F(x,\xi) \in K \quad \text{for all} & \xi \in U. \end{array} \tag{6.2}$$

Other methods dealing with data uncertainty include:

6.1.1 Stochastic Method

$$\text{(EOPT)} \qquad \begin{array}{ll} \text{Minimize} & \mathrm{E}_\xi[f(x,\xi)] \\ \text{Subject to} & \mathrm{E}_\xi[F(x,\xi)] \in K. \end{array}$$

6.1.2 Sampling Method

$$\text{(SOPT)} \qquad \begin{array}{ll} \text{Minimize} & z \\ \text{Subject to} & F(x,\xi^k) \in K \\ & f(x,\xi^k) \le z \\ & \quad \text{for large sxamples} \quad \xi^k \in U. \end{array}$$

This problem may be a semin-infinite programming problem.

6.2 Robust Quadratic Optimization

$$\text{(EQP)} \qquad \begin{array}{ll} \text{Minimize} & q^T x \\ \text{Subject to} & \|Ax\|^2 \le 1. \end{array} \tag{6.3}$$

Here, vector $q \in R^n$ and $A \in R^{m\times n}$; and $\|.\|$ is the Euclidean norm.

6.2.1 Ellipsoid uncertainty

Let A be uncertain and

$$A = A^0 + \sum_{j=1}^{k} u_j A^j |\ u^T u \le 1.$$

$$\text{(REQP)} \qquad \begin{array}{ll} \text{Minimize} & q^T x \\ \text{Subject to} & \|(A^0 + \sum_{j=1}^k u_j A^j)x\| \le 1 \quad \forall u^T u \le 1. \end{array}$$

The problem can be rewritten as

$$\text{(REQP)} \qquad \begin{array}{ll} \text{Minimize} & q^T x \\ \text{Subject to} & \|(A^0 + \sum_{j=1}^k u_j A^j)x\|^2 \le 1 \quad \forall u^T u \le 1. \end{array}$$

Let

$$F(x) = (A^0 x,\ A^1 x,\ \cdots A^k x).$$

Then

$$\|(A^0 + \sum_{j=1}^{k} u_j A^j)x\|^2 = \begin{pmatrix} 1 \\ u \end{pmatrix}^T F(x)^T F(x) \begin{pmatrix} 1 \\ u \end{pmatrix}$$

The problem becomes

$$\text{(REQP)} \quad \begin{array}{ll} \text{Minimize} & q^T x \\ \text{Subject to} & \begin{pmatrix} 1 \\ u \end{pmatrix}^T \left(\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - F(x)^T F(x) \right) \begin{pmatrix} 1 \\ u \end{pmatrix} \geq 0 \\ & \forall \quad \begin{pmatrix} 1 \\ u \end{pmatrix}^T \begin{pmatrix} 1 & 0 \\ 0 & -I \end{pmatrix} \begin{pmatrix} 1 \\ u \end{pmatrix} \geq 0. \end{array}$$

6.2.2 S-Lemma

Lemma 6.1 *Let P and Q be two symmetric matrices such that there exists u^0 satisfying $(u^0)^T P u^0 > 0$. Then the implication that*

$$u^T P u \geq 0 \Rightarrow u^T Q u \geq 0$$

holds true if and only if there exists $\lambda \geq 0$ such that

$$Q \succeq \lambda P.$$

From the lemma, we immediately see

$$\begin{pmatrix} 1 \\ u \end{pmatrix}^T \left(\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - F(x)^T F(x) \right) \begin{pmatrix} 1 \\ u \end{pmatrix} \geq 0$$

$$\forall \begin{pmatrix} 1 \\ u \end{pmatrix}^T \begin{pmatrix} 1 & 0 \\ 0 & -I \end{pmatrix} \begin{pmatrix} 1 \\ u \end{pmatrix} \geq 0$$

if and only if there is a $\lambda \geq 0$ such that

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - F(x)^T F(x) \succeq \lambda \begin{pmatrix} 1 & 0 \\ 0 & -I \end{pmatrix}.$$

This is equivalent to

$$\begin{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \lambda \begin{pmatrix} -1 & 0 \\ 0 & I \end{pmatrix} & F(x)^T \\ F(x) & I \end{pmatrix} \succeq 0$$

which is a SDP constraint since $F(x)$ is linear in x. Here we have used another technical lemma

Lemma 6.2 *Let P a symmetric matrix and A be a rectangle matrix. Then*

$$P - A^T A \succeq 0$$

if and only if

$$\begin{pmatrix} P & A^T \\ A & I \end{pmatrix} \succeq 0.$$

6.2.3 SDP for Robust QP

Consider λ and x as variables, REQP finally becomes a SDP problem:

$$\text{(REQP)} \quad \begin{array}{ll} \text{Minimize} & q^T x \\ \text{Subject to} & \left(\begin{array}{cc} \left(\begin{array}{cc} 1-\lambda & 0 \\ 0 & \lambda I \end{array} \right) & F(x)^T \\ F(x) & I \end{array} \right) \succeq 0. \end{array}$$

6.3 General Robust Quadratic Optimization

$$\text{(EQP)} \quad \begin{array}{ll} \text{Minimize} & q^T x \\ \text{Subject to} & -x^T A^T A x + 2b^T x + \gamma \geq 0. \end{array} \tag{6.4}$$

Here, vector $q, b \in R^n$ and and $A \in R^{m \times n}$.

6.3.1 Ellipsoid uncertainty

Let A be uncertain and

$$(A, b, \gamma) = (A^0, b^0, \gamma^0) + \sum_{j=1}^{k} u_j (A^j, b^j, \gamma^j) |\ u^T u \leq 1.$$

$$\text{(REQP)} \quad \begin{array}{ll} \text{Minimize} & q^T x \\ \text{Subject to} & -x^T A^T A x + 2b^T x + \gamma \geq 0 \quad \forall u^T u \leq 1. \end{array}$$

Let again

$$F(x) = (A^0 x,\ A^1 x,\ \cdots A^k x).$$

Then

$$-x^T A^T A x + 2b^T x + \gamma =$$

$$\left(\begin{array}{c} 1 \\ u \end{array} \right)^T \left(\left(\begin{array}{cccc} \gamma^0 + 2x^T b^0 & \gamma^1/2 + x^T b^1 & \cdots & \gamma^k/2 + x^T b^k \\ \gamma^1/2 + x^T b^1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \gamma^k/2 + x^T b^k & 0 & \cdots & 0 \end{array} \right) - F(x)^T F(x) \right) \left(\begin{array}{c} 1 \\ u \end{array} \right).$$

6.3.2 SDP formulation

From the S-lemma, we immediately see if and only if there is $\lambda \geq 0$ such that

$$\begin{pmatrix} \gamma^0 + 2x^T b^0 & \gamma^1/2 + x^T b^1 & \cdots & \gamma^k/2 + x^T b^k \\ \gamma^1/2 + x^T b^1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \gamma^k/2 + x^T b^k & 0 & \cdots & 0 \end{pmatrix} - F(x)^T F(x)) \succeq \lambda \begin{pmatrix} 1 & 0 \\ 0 & -I \end{pmatrix};$$

which is is equivalent to

$$\begin{pmatrix} \begin{pmatrix} \gamma^0 + 2x^T b^0 & \gamma^1/2 + x^T b^1 & \cdots & \gamma^k/2 + x^T b^k \\ \gamma^1/2 + x^T b^1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \gamma^k/2 + x^T b^k & 0 & \cdots & 0 \end{pmatrix} + \lambda \begin{pmatrix} -1 & 0 \\ 0 & I \end{pmatrix} & F(x)^T \\ F(x) & I \end{pmatrix} \succeq 0.$$

which is a SDP constraint since the matrix is linear in x.

Consider λ and x as variables, REQP finally becomes a SDP problem:

$$\text{(REQP)} \quad \begin{array}{ll} \text{Minimize} & q^T x \\ \text{Subject to} & \begin{pmatrix} \begin{pmatrix} \gamma^0 + 2x^T b^0 - \lambda & \gamma^1/2 + x^T b^1 & \cdots & \gamma^k/2 + x^T b^k \\ \gamma^1/2 + x^T b^1 & \lambda & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \gamma^k/2 + x^T b^k & 0 & \cdots & \lambda \end{pmatrix} & F(x)^T \\ F(x) & I \end{pmatrix} \succeq 0. \end{array}$$

6.4 More Robust Quadratic Optimization

$$\text{(EQP)} \quad \begin{array}{ll} \text{Minimize} & q^T x \\ \text{Subject to} & -x^T A_i^T A_i x + 2b_i^T x + \gamma_i \geq 0 \quad i = 1, ..., t. \end{array} \tag{6.5}$$

Here, vector $q, b_i \in R^n$ and and $A_i \in R^{m \times n}$.

6.4.1 Ellipsoid uncertainty

Let A be uncertain and

$$(A_i, b_i, \gamma_i) = (A_i^0, b_i^0, \gamma_i^0) + \sum_{j=1}^{k} u_j (A_i^j, b_i^j, \gamma_i^j) |\ u^T u \leq 1.$$

$$\text{(REQP)} \quad \begin{array}{ll} \text{Minimize} & q^T x \\ \text{Subject to} & -x^T A_i^T A_i x + 2b_i^T x + \gamma_i \geq 0,\ i = 1, ..., t, \quad \forall u^T u \leq 1. \end{array}$$

6.4.2 SDP formulation

Let again

$$F_i(x) = (A_i^0 x,\ A_i^1 x,\ \cdots A_i^k x).$$

Consider $\lambda_i,\ i = 1, ..., t$, and x as variables, REQP finally becomes a SDP problem:

Min $\quad q^T x$

s.t. $$\begin{pmatrix} \begin{pmatrix} \gamma_i^0 + 2x^T b_i^0 - \lambda_i & \gamma_i^1/2 + x^T b_i^1 & \cdots & \gamma_i^k/2 + x^T b_i^k \\ \gamma_i^1/2 + x^T b_i^1 & \lambda_i & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \gamma_i^k/2 + x^T b_i^k & 0 & \cdots & \lambda_i \end{pmatrix} & F_i(x)^T \\ F_i(x) & I \end{pmatrix} \succeq 0,$$

$i = 1, ..., t.$

6.5 Tool for Robust Optimization

Recall:

$$\text{(ROPT)} \qquad \begin{array}{ll} \text{Minimize} & \sup_{\xi \in U} f(x, \xi) \\ \text{Subject to} & F(x, \xi) \leq 0 \quad \text{for all} \quad \xi \in U. \end{array} \tag{6.6}$$

Express

$$\xi = \xi(u) \quad \text{where} \quad u \in \hat{U}.$$

Then,

$$\sup_{\xi \in U} f(x, \xi) = \sup_{u \in \hat{U}} f(x, \xi(u))$$

In convex cases, we represent

$$\sup_{u \in \hat{U}} f(x, \xi(u))$$

by its conic dual problem and let the dual objective function be

$$\phi(\lambda^0, x)$$

where λ^0 is the dual variables. The dual objective function is an upper bound on $f(x, \xi(u))$ so we could represent the problem by

$$\text{(ROPT)} \qquad \begin{array}{ll} \text{Minimize} & \phi(\lambda^0, x) \\ \text{Subject to} & F(x, \xi) \leq 0 \quad \text{for all} \quad \xi \in U \\ & \text{dual constraints on } \lambda^0. \end{array} \tag{6.7}$$

We handle the constraints in the exact same manner, and the problem becomes

$$\text{(ROPT)} \qquad \begin{array}{ll} \text{Minimize} & \phi(\lambda^0, x) \\ \text{Subject to} & \Phi(\lambda, x) \in C \\ & \text{dual constraints on } \lambda^0, \lambda, \end{array} \tag{6.8}$$

where $\Phi(\lambda, x)$ is, component-wise, the dual objective function of

$$\sup_{u \in \hat{U}} F(x, \xi(u)),$$

and C is a suitable cone.

6.5.1 Examples

$$ax + b \leq 0$$

where

$$(a, b) = (a^0, b^0) + \sum_{i=1}^{k} u_i (a^i, b^i) | \quad u \in \hat{U}.$$

Case of $\|u\| \leq 1$:

$$\max_{u : \|u\| \leq 1} \quad a^0 x + b^0 + \sum_{i=1}^{k} u_i (a^i x + b^i)$$

The dual is

$$\begin{array}{ll} \text{Minimize} & \lambda \\ \text{Subject to} & \lambda - (a^0 x + b^0) \geq \sqrt{\sum_{i=1}^{k} (a^i x + b^i)^2}. \end{array}$$

Thus, the robust constraint becomes

$$0 \geq \lambda \geq (a^0 x + b^0) + \sqrt{\sum_{i=1}^{k} (a^i x + b^i)^2}.$$

Case of $\|u\|_\infty \leq 1$:

$$\max_{u : \|u\|_\infty \leq 1} \quad a^0 x + b^0 + \sum_{i=1}^{k} u_i (a^i x + b^i)$$

The dual is

$$\begin{array}{ll} \text{Minimize} & \lambda \\ \text{Subject to} & \lambda - (a^0 x + b^0) \geq \sum_{i=1}^{k} |(a^i x + b^i)|. \end{array}$$

Thus, the robust constraint becomes

$$0 \geq \lambda \geq (a^0 x + b^0) + \sum_{i=1}^{k} |(a^i x + b^i)|.$$

Case of $\|u\|_1 \leq 1$:

$$\max_{u:\|u\|_1 \leq 1} \quad a^0 x + b^0 + \sum_{i=1}^{k} u_i (a^i x + b^i)$$

The dual is

$$\begin{array}{ll} \text{Minimize} & \lambda \\ \text{Subject to} & \lambda - (a^0 x + b^0) \geq \max_i\{|(a^i x + b^i)|\}. \end{array}$$

Thus, the robust constraint becomes

$$0 \geq \lambda \geq (a^0 x + b^0) + \max_i\{|(a^i x + b^i)|\}.$$

6.6 Applications

6.6.1 Robust Linear Least Squares

$$\begin{array}{ll} \text{Minimize} & \|Ay - c\|^2 \\ & y \in Y. \end{array}$$

If the data (A, c) are from the set

$$(A, c) = (A^0, c^0) + \sum_{i=1}^{k} u_i (A^i, c^i) \mid \|u\| \leq 1,$$

the robust counter part can be drawn from the earlier disscusion, since the original problem is a convex QP problem.

6.6.2 Heat dissipation problem

Consider an electric circuit represented as a connected oriented graph Γ with N arcs, each of which possesses a given conductance s_{ij}. Assuming that the arcs of the circuit are charged with external voltages y and internal voltages v. These voltages induce currents in the circuit; as a result, the circuit dissipates heat. According to Kirchoff's laws, the heat is given by

$$H = \min_{v \in R^m} (Qv + Py)^T S(Qv + Py),$$

where Q and P are some matrices given by the topology of the circuit, m is the number of nodes of the circuit, and S is $N \times N$ diagonal matrix with s_{ij} as its diagonal entries.

The problem is to choose y such that

$$\min_{y \in Y} \left[\min_{v \in R^m} (Qv + Py)^T S(Qv + Py) \right],$$

where Y is a convex set.

Now if

$$S = S(u) = S^0 + \sum_{i=1}^{k} u_i S^i | \ \|u\| \leq 1,$$

we have a robust optimization counterpart:

$$\min_{y \in Y} \left\{ \max_{u \in \hat{U}} \left[\min_{v \in R^m} (Qv + Py)^T S(u)(Qv + Py) \right] \right\}.$$

Since $(Qu+Py)^T S(u)(Qu+Py)$ is convex in (y, v) and linear in u, this problem is equivalent to

$$\min_{y \in Y, v \in R^m} \left[\max_{u \in \hat{U}} (Qv + Py)^T S(u)(Qv + Py) \right].$$

6.6.3 Portfolio optimization

$$\begin{array}{ll} \text{Minimize} & x^T V x \\ \text{Subject to} & \mu^T x \geq \alpha \\ & e^T x = 1. \end{array}$$

Now V and μ may be uncertain, for example,

$$(V, \mu) = (V^0, \mu^0) + \sum_{i=1}^{k} u_i (V^i, \mu^i) | \ \|u\| \leq 1.$$

The robust counterpart of the portfolio problem is

$$\begin{array}{ll} \text{Minimize} & \max_{V \in \mathcal{V}} [x^T V x] \\ \text{Subject to} & \min_{\mu \in S} [\mu^T x] \geq \alpha \\ & e^T x = 1 \end{array}$$

6.7 Exercises

6.1 *Prove the S-lemma.*

6.2 *In the linear least squares problem, assume that*

$$(A, c) = (A^0, c^0) + \sum_{i=1}^{k} u_i (A^i, c^i) | \ \|u\| \leq 1.$$

Construct the robust counterparts of the problem for three norms on u:

$$\|u\|_2 \leq 1, \quad \|u\|_1 \leq 1, \quad \textit{and} \quad \|u\|_\infty \leq 1.$$

6.3 *In the Heat dissipation problem, assume that*

$$S = S(u) = S^0 + \sum_{i=1}^{k} u_i S^i | \ \|u\| \leq 1,$$

and $s^i \succeq 0$ for all $i = 0, ..., k$. Construct the robust counterparts of the problem for three norms on u:

$$\|u\|_2 \leq 1, \quad \|u\|_1 \leq 1, \quad \textit{and} \quad \|u\|_\infty \leq 1.$$

6.4 *In the portafolio optimization problem, assume that*

$$(V, \mu) = (V^0, \mu^0) + \sum_{i=1}^{k} u_i (V^i, \mu^i) | \ \|u\| \leq 1,$$

and $V^i \succeq 0$ for all $i = 0, ..., k$. Construct the robust counterparts of the problem for three norms on u:

$$\|u\|_2 \leq 1, \quad \|u\|_1 \leq 1, \quad \textit{and} \quad \|u\|_\infty \leq 1.$$

Chapter 7

SDP for Quantum Computation

7.1 Quantum Computation

Quantum Computation uses Quantum Mechanics to enhance computation power. First concrete result was developed for Integer Factorization by Shor in 1994; and then for Quantum search algorithm and space reduction by Grover in 1996. The field is closely linked to Quantum Information Theory and Statistics, Quantum Channels and Optimization.

To describe quantum mechanics, Hilbert sapces and linear operators are generally used. In this lecture, we restrict ourselves to finite-dimensional real vector spaces. In classical quantum physics, a *quantum state*, or a *state*, is a column vector v in the N-dimensional vector space R^N with norm $||v||_2 = \sqrt{v \cdot v} = 1$.

Nowadays, another formalism is widely accepted, in which a state is a postive semidefinite symmetric matrix V with trace 1 that belongs to $R^{N \times N}$. A matrix of such property is called a *density matrix*. For example, a density matrix V with rank one can be represented as $V = vv^T$, and $Tr(V) = 1$. A density matrix of rank higher than 1 can be written as

$$V = \sum_{i=1}^{N} \lambda_i v_i v_i^T,$$

where $\lambda_i \geq 0$ for all i and $\sum_i \lambda_i = 1$.

A density matrix of rank 1 is called a *pure sate*, and a density matrix of rank higher than 1 is called a *mixed state*. The state V is interpreted as a probalistic mixture of pure states $v_i v_i^T$ with mixing probabilities λ_i.

7.2 Completely Positive Map

The state of a system may change as time goes on.

Given $\{A_1, A_2, \cdots, A_k\}$, where $A_i \in R^{M\times N}$, such that

$$\sum_{i=1}^{k} A_i A_i^T = I,$$

trace-preserving complete positive (CP) map of V is

$$T(V) = \sum_{i=1}^{k} A_i V A_i^T.$$

Note that $T(V) \in R^{M\times M}$, so CP is a linear mapping that conveys quantum states in $R^{N\times N}$ to the states in $R^{M\times M}$. In quantum computing, the pure state is in R^{2^n} or $N = 2^n$, called n-qubit. When $n = 1$, a unit orthogonal $(1; 0)$ and $(0; 1)$ represent the classical bits.

A *measurement* is described as a finite set of symmetric positive semidefinite matrices $\{M_1, M_2, \cdots, M_k\}$ whose sum is an identity matrix, i.e.,

$$M_i \succeq 0,\ \forall i = 1, \cdots, k,\ \text{and}\ \sum_{i=1}^{k} M_i = I,$$

called Positive Operator-Valued Measures (POVM).

Given the system in state V and a measurement $\{M_1, M_2, \cdots, M_k\}$ performed on the system, we obtain a random varible X with

$$\mathbf{P}r(X = l) = V \bullet M_l, \quad l = 1, \cdots, k.$$

A fundamental problem is quantum detection theory is to find an optimal measurement for given mixed states.

Suppose we are given m states $V_1, V_2, \cdots, V_m$ with corresponding prior probabilities $\xi_1, \xi_2, \cdots, \xi_m$. For a measurement $M = \{M_1, \ldots, M_k\}$, we have

$$P(j|i) = Tr(V_i M_j).$$

Let $c_{ij}(\geq 0)$ be the cost (penlaty) of taking a true state i to be j. Then the average cost is

$$C(M) = \sum_{i,j} \xi_i P(j|i) c_{ij} = \sum_j Tr\left[\left(\sum_i \xi_i V_i c_{ij}\right) M_j\right].$$

Let $W_j = \sum_i \xi_i V_i c_{ij},\ j = 1, \ldots, k$. The problem of minimizing the average cost can be formulated as the following SDP:

$$\begin{array}{ll} \text{Minimize} & \sum_{j=1}^{k} Tr(W_j M_j) \\ \text{Subject to} & \sum_{j=1}^{k} M_j = I, \\ & M_j \succeq 0,\ j = 1, \ldots, k. \end{array}$$

What is the dual?

$$\begin{aligned} &\text{Maximize} && \textstyle\sum_{j=1}^{k} Tr(U) \\ &\text{Subject to} && W_j \succeq U,\ j = 1, \ldots, k. \end{aligned}$$

Thus, the optimality Condition becomes

$$W_i \succeq \sum_{j=1}^{k} M_j W_j = \sum_{j=1}^{k} W_j M_j, \quad i = 1, ..., k.$$

7.3 Channel Capacity Problem

A classical discrete memoryless channel consists of a set of input symbols $A = \{a_1, a_2, \ldots, a_m\}$, a set of output symbols $B = \{b_1, b_2, \ldots, b_n\}$ and a set of conditional probabilities $V_{ij} = P(b_j|a_i)$ with $\sum_{j=1}^{n} V_{ij} = 1$, for $i = 1, \ldots, m$.

Therefore, $V = \{V_{ij}\}$ is a stochastic matrix, where V_{ij} is the probability that b_j is received when a_i is sent through the channel. The capacity $C(V)$ of a channel V, defined as the maximum of rates (speeds) at which information can be transmitted reliably, is given by following theorem due to Shannon.

The classical channel coding theorem is modeled as:

$$\begin{aligned} C(V) \quad &\text{Maximize} && I(p, V) \\ &\text{Subject to} && \textstyle\sum_{i=1}^{m} p_i = 1, \text{ with } p = \{p_i\} \\ & && p_i \geq 0,\ i = 1, \ldots, m, \end{aligned}$$

where

$$I(p, V) = \sum_i p_i \sum_j V_{ij} \log \frac{V_{ij}}{\sum_k p_k V_{kj}}$$

The Kullback -Leibler divergence, defined as follows

$$D(q||r) = \sum_j q_j \log \frac{q_j}{r_j},$$

is a measure of *distance* between two probability distributions $q = (q_j)$ and $r = (r_j)$, and plays an important role in information theory.

Then, the function $I(p, V)$ can be written in terms of the above Kullback-Leibler divergence

$$I(p, V) = \sum_i p_i D(V_i.||pV),$$

where $r = pV$ is a distribution defined by $r_j = \sum_i p_i V_{ij}$. With a channgel V fiexed, $I(p, V)$ is concave in $p = (p_i)$. Hence, the computatution of $C(V)$ is a concave maximization problem with linear constraints.

A quantum memoryless channel consists of a set S_1 of input states (or density matrices) in $R^{m\times m}$, S_2 of output states (or density matrices) in $R^{n\times n}$, and a CP (map) $\Gamma : S_1 \to S_2$. The capacity $C(\Gamma)$ of a quantum channel Γ, defined to be the maximum of rates (speeds) at which reliable transmission of classical information through the channel Γ is possible.

The Quantum channel coding theorem due to A. S. Holevo is modeled by

$$\begin{array}{lll} C(V) & \text{Maximize} & I(\{\lambda_i\},\{X_i\},\Gamma) \\ & \text{Subject to} & X_i \in S_1, \quad i = 1,\ldots,d \\ & & \sum_{i=1}^{d} \lambda_i = 1 \\ & & \lambda_i \geq 0,\ i = 1,\ldots,d, \end{array}$$

where $d = n^2$, and

$$I(\{\lambda_i\},\{X_i\},\Gamma) = \sum_i \lambda_i Tr(\Gamma(X_i)) \left[\log Tr(\Gamma(X_i)) - \log Tr(\Gamma(\bar{X}))\right],$$

with $\bar{X} = \sum_i \lambda_i X_i$.

Observe that $I(\{\lambda_i\},\{X_i\},\Gamma)$ can be written in terms of the quantum counterpart of the Kullback-Leibler divergence (often called *relative entropy*)

$$D(X||Y) = Tr(X)[\log Tr(X) - \log Tr(Y)].$$

Once the ensemble $X = \{X_i\}$ is fixed, $I(\{\lambda_i\},\{X_i\},\Gamma)$ is a concave function in $\lambda = \{\lambda_i\}$. This is again a convex programming problem.

But, in general, $I(\{\lambda_i\},\{X_i\},\Gamma)$ is neither convex nor concave in (λ, X).

7.4 Quantum Interactive Proof System

Interation Between Prover and Verifier.

Consider two CP mappings: given $\{A_1, A_2, \cdots, A_k\}$ and $\{B_1, B_2, \cdots, B_l\}$, where $A_i, B_i \in R^{M\times N}$, such that

$$\sum_{i=1}^{k} A_i A_i^T = I \quad \text{and} \quad \sum_{i=1}^{l} B_i B_i^T = I,$$

Define

$$T_1(V) = \sum_{i=1}^{k} A_i V A_i^T$$

and

$$T_2(V) = \sum_{i=1}^{l} B_i V B_i^T.$$

The question is, are there two states V_1 and V_2 such that

$$T_1(V_1) = T_2(V_2);$$

or for all V_1 and V_2

$$\|T_1(V_1) - T_2(V_2)\| \geq \epsilon.$$

An SDP Formulation for this problem is

$$\begin{array}{ll} \text{Minimize} & t \\ \\ \text{Subject to} & I \bullet V_1 = I \bullet V_2 = 1, \\ \\ & V_1, V_2 \succeq 0 \\ & T_1(V_1) - T_2(V_2) + t \cdot I \succeq 0 \\ & t \cdot I - T_1(V_1) + T_2(V_2) \succeq 0. \end{array}$$

7.5 Notes

7.6 Exercises

Chapter 8

Computational Issues

It is common to have a gap between a theoretical algorithm and its practical implementation: the theoretical algorithm makes sure that it works for all instances and never fails, while the practical implementation emphasizes average performance and uses many ad-hoc "tricks." 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 solver, the high-order predictor-corrector method, the homogeneous and self-dual method, and the optimal basis finder. Our goal is to provide additional theoretical justification for using these techniques and to explain their practical effectiveness.

8.1 Presolver

One way to eliminate data error and to improve solution efficiency in solving linear programs is called the "presolver"—a preliminary process to check the data set (A, b, c) in order to detect inconsistency and to remove redundancy. This process could reduce the size of the problem as well, because users of many LP problems likely introduce superfluous variables and redundant constraints for simplicity and convenience in model formulation.

In general, detecting all inconsistency and redundancy in an LP problem is computationally intractable. Therefore all presolvers use an arsenal of simple inspection techniques. These techniques are applied repeatedly until the problem cannot be reduced any further. Below, we briefly present the most common inspection procedures.

- Remove empty or all-zero rows and columns.
- Eliminate a fixed variable, the variable has a fixed value, from the problem by substitution.
- Remove duplicate constraints. Two constraints are said to be duplicate if they are identical up to a scalar multiplier. One of the duplicate constraints is removed from the problem.

  Remove duplicate columns. Two columns are said to be duplicate if they are identical up to a scalar multiplier. (They make duplicate constraints in the dual.)
- Remove linearly dependent constraints. The presence of linearly dependent rows in A may lead to serious numerical problems in an interior-point methods, since it implies a rank deficiency in the Newton equation system.
- Remove a singleton row (only one non-zero coefficient in the row) by construction of a simple variable bound. For example, if the ith constraint is in the form $a_{i1}x_1 \leq b_i$, we can convert it to $x_1 \leq b_i/a_{i1}$ or $x_1 \geq b_i/a_{i1}$, depending on the sign of a_{i1},
- Remove a free and singleton column (only one non-zero coefficient in the column and the associated variable is free). For example, let free variable x_1 appears only in the ith constraint. Then, x_1 and the ith constraint can be eliminated, while the optimal value of x_1 can be recovered from the ith constraint by substitution of the optimal solution of the remaining LP problem.

  A nonnegative but unbounded variable, say, $0 \leq x_1 < +\infty$ in singleton column 1, can be used to generate a bound on dual variables y_i. Namely,

  $$a_{i1}y_i \leq c_1.$$

  This inequality can be used, depending on the sign of a_{i1}, to produce a lower or upper bound on y_i.
- Determine lower and upper limits for every constraint and detect infeasibility. For example, consider the ith (inequality) constraint

  $$\sum_j a_{ij}x_j \leq b_i,$$

  and let each variable x_j lie on $[0, u_j]$. Then compute

  $$\underline{b}_i = \sum_{\{j:\ a_{ij}<0\}} a_{ij}u_j \leq 0 \quad \text{and} \quad \bar{b}_i = \sum_{\{j:\ a_{ij}>0\}} a_{ij}u_j \geq 0. \tag{8.1}$$

  Thus, we must have

  $$\underline{b}_i \leq \sum_j a_{ij}x_j \leq \bar{b}_i. \tag{8.2}$$

If $\overline{b}_i \leq b_i$, then the ith constraint is *redundant* and can be removed. If $\underline{b}_i > b_I$, then the problem is infeasible. If $\underline{b}_i = b_i$, the i constraint becomes equality and will force all involved variables take values at their appropriate bounds.

The same technique can be applied to each of the dual constraints.

- Add implicit bound to a free primal variable. For example, suppose the ith constraint is
$$\sum_j a_{ij} x_j = b_i,$$
where $a_{i1} > 0$, x_1 is a free variable, and all other variables x_j lie on $[0, u_j]$. Then
$$x_1 \leq \left(b_i - \sum_{\{j \neq 1:\ a_{ij} < 0\}} a_{ij} u_j \right) / a_{i1}$$
and
$$x_1 \geq \left(b_i - \sum_{\{j \neq 1:\ a_{ij} > 0\}} a_{ij} u_j \right) / a_{i1}.$$
The same technique can be applied to a dual free variable.

- Improve the sparsity of A, i.e., reduce the non-zero elements in A. We could look for a nonsingular matrix $M \in \mathcal{R}^{m \times m}$ such that the matrix MA is as sparse as possible. Primal constraints can in such case be replaced with equivalent
$$MAx = Mb, \tag{8.3}$$
which may be more suitable for an interior-point solver. Exact solution of this *sparsity problem* is an NP-complete problem but efficient heuristics usually produce satisfactory non-zero reduction in A.

The application of the above presolver techniques often results in impressive size-reduction of an initial LP formulation. Thus, it is our hope that the reduced problem obtained after the presolver can be solved faster. Once a solution is found, it could be used to recover a complete primal and dual solution to the original LP problem. This phase is called the *postsolver*.

8.2 Linear System Solver

The major work in a single iteration of all interior-point algorithms is to solve a set of linear equations, such as (3.38). It can be reduced to the so-called KKT system:
$$\begin{pmatrix} D^{-2} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} d_x \\ -d_y \end{pmatrix} = \begin{pmatrix} \bar{c} \\ \bar{b} \end{pmatrix}, \tag{8.4}$$

The diagonal matrix D varies in different interior-point methods. Most general purpose codes use *direct* methods to solve the KKT system. Two competitive direct methods are: the *normal equation* approach and the *augmented system* approach. The former works with a smaller positive definite matrix, and the latter requires factorization of a symmetric indefinite matrix. They all use variants of the symmetric triangular $L\Lambda L^T$ decomposition, where L is a lower triangular matrix and Λ is a block diagonal matrix with blocks of dimension 1 or 2. (The QR decomposition of A uses an orthogonal transformation and guarantees high accuracy, but it cannot be used in practice due to its costly operations.)

8.2.1 Solving normal equation

The normal equation approach further reduces (8.4) to the normal equation:

$$(AD^2A^T)d_y = \bar{b} - AD^2\bar{c}. \tag{8.5}$$

An advantage of this approach is that it works with a positive definite matrix AD^2A^T if A has full row rank. Thus the Choleski decomposition of this matrix exists for any D and numerical stability is assured in the pivoting process. Moreover, the sparsity pattern in the decomposition is independent of the value of D and hence it is invariant in all iterations. Consequently, once a good sparsity preserving pivoting order is chosen, it can be used throughout the entire iterative process. This argument has been used to justify the application of the normal equations approach in very first interior-point method implementation.

The success of the Choleski factorization depends on a pivoting order for preserving sparsity in the Choleski factor L. Its goal is to find a permutation matrix P such that the factor of $PAD^2A^TP^T$ is as sparse as possible. In practice, heuristics are used to find such a permutation or ordering. (Finding an optimal permutation is an NP-complete problem.) After an ordering is found, the data structure and indirect index addressing of L are setup. This is referred to as the symbolic phase because no numerical computation is involved.

Two heuristic orderings, *minimum degree* and the *minimum local fill-in*, are particularly useful in implementing interior-point algorithms. They are both "local" or myopic, i.e. they select a pivot only from a set of currently best candidates.

Minimum degree ordering

Assume that, in the kth step of the Gaussian elimination, the ith column of the Schur complement contains d_i non-zero entries and its diagonal element becomes a pivot. The kth step of the elimination requires thus

$$l_i = (1/2)(d_i - 1)^2, \tag{8.6}$$

floating-point operations or *flops* to be executed.

In what follows, keep in your mind the fact that the decomposed matrix AD^2A^T is positive definite so the pivot choice can be limited to the diagonal elements. In fact, only this choice preserves symmetry.

Note that if the ith diagonal element becomes a pivot, l_i evaluates flops and gives an overestimate of the fill-ins, the new non-zeros created in the Schur complement, which can result from the current elimination step. Thus, the "best" pivot at step k, in terms of the number of flops required to complete the kth elimination step, is the one that minimizes d_i among all diagonal elements in the Schur complement. Interpreting the elimination process as the corresponding incidence graph elimination, one can see that this strategy chooses a node (diagonal element) in the graph which has the minimum degree (d_i). This is how the strategy is named. This ordering procedure can be implemented efficiently both in terms of time speed and storage requirement.

There is also an *approximate* minimum degree ordering available. The method is faster while generates the same quality ordering.

Minimum local fill-in ordering

In general, l_i of (8.6) considerably overestimates the number of fill-ins in the kth step of the Gaussian elimination, because it does not take into account the fact that in many positions of the predicted fill-ins, non-zero entries already exist. It is possible that another pivot candidate, although may not minimize d_i, would produce least fill-ins in the remaining Schur complement. The minimum local fill-in ordering chooses such a pivot. Generally, the minimum local fill-in procedure produces an ordering resulting in a sparser factorization but at a higher cost, because it chooses the pivot that produces the minimum number of fill-ins among all remaining pivot candidates.

Pros and cons

Solving the normal equation is proved to be a reliable approach to solutions of most practical linear programs. However, it suffers two drawbacks. First, the normal equation behaves badly whenever a primal linear program contains free variables. In order to transform such a problem to the standard form, a free variable is usually replaced with the difference of two nonnegative variables: $x = x^+ - x^-$. Interior-point algorithms typically generate iterates in which both x^+ and x^- converge to ∞, although their difference is kept relatively close to the optimal value of x. This results in a serious ill-condition in the normal matrix and a loss of accuracy in solving (8.5). A remedy used in many implementations is to prevent excessive growth of x^+ and x^- by enforcing bounds on x^+ and x^-.

Second, a more serious drawback of the normal equation approach is that it looses sparsity from the presence of dense columns in A. The reason is that a single dense column in A with p non-zero elements creates a complete dense submatrix of size $p \times p$ in AD^2A^T after a symmetric row and column permutation. Special care has to be taken in this case.

Assume that

$$A = (A_S,\ A_D), \tag{8.7}$$

where $A_S \in \mathcal{R}^{m\times n-k}$ is sparse and $A_D \in \mathcal{R}^{m\times k}$ is dense. Then, we need to treat A_D separately. The most popular way in solving the normal equation employs the *Schur complement* mechanism. It is based on separating the normal matrix

$$AD^2A^T = A_SD_S^2A_S^T\ +\ A_DD_D^2A_D^T, \tag{8.8}$$

into the presumably sparse part $A_SD_S^2A_S^T$ and the significantly denser symmetric rank-k matrix $A_DD_D^2A_D^T$. A Choleski decomposition is computed for the sparse part and the dense rank-k matrix is then updated by the Sherman-Morrison-Woodbury formula (see Exercise 1.1).

This method is not guaranteed to work correctly because the sparse part may be rank deficient, since A_S may not have full row rank. Whenever this happens, the Choleski decomposition of $A_SD_S^2A_S^T$ does not exist and the Sherman-Morrison-Woodbury update is not well defined. Therefore in a practical implementation diagonal elements are selectively added to $A_SD_S^2A_S^T$ to make the decomposition exist. We observe that the rank deficiency of $A_SD_S^2A_S^T$ cannot exceed k, the number of dense columns. This method usually works in satisfaction for a small number of dense columns.

This is how we do it. If unacceptably small pivots are encountered during the Choleski decomposition of $A_SD_S^2A_S^T$, we add a "regularizing" diagonal term to each of them. Consequently, instead of computing the decomposition of $A_SD_S^2A_S^T$, we compute the decomposition of another matrix $A_SD_S^2A_S^T+\sigma EE^T$, where positive number σ is a regularizing term and E is a matrix built from unit columns where each non-zero appears in the row corresponding to regularized pivots, that is,

$$L\Lambda L^T = A_SD_S^2A_S^T + \sigma EE^T. \tag{8.9}$$

L is used as a stable "working basis" in the Sherman-Morrison-Woodbury update of the Schur complement to compute

$$(AD^2A^T)^{-1} = (L\Lambda L^T\ +\ (A_DD_D^2A_D^T\ -\ \sigma EE^T))^{-1}.$$

In many cases, choosing $\sigma = 1$ seems sufficient.

Summing up, it is possible to overcome the dense column difficulty arisen in the normal equation approach. But there remains a question to decide which columns should be treated as dense ones. A naive selection rule, which is based on counting the number of non-zero elements in a column, does not necessarily identify all the "troubling" columns—the columns make the decomposition dense. This motivated researchers to directly solve the augmented system of the Newton equation (8.4), which allows more freedom in selecting pivots.

8.2.2 Solving augmented system

The augmented system approach is a well understood technique to solve a least-squares problem. It applies a factorization to a symmetric indefinite matrix

$$L\Lambda L^T = \begin{pmatrix} D^{-2} & A^T \\ A & 0 \end{pmatrix}, \tag{8.10}$$

where Λ is an indefinite block diagonal matrix where each block is either 1×1 or 2×2.

In contrast to solving the normal equation in which the sparsity ordering and the numerical factorization are separated, the factorization of (8.10) is computed dynamically. In other words, the choice of a pivot is concerned with both sparsity and stability of the triangular factor L. Thus, the factorization of the augmented system is at least as stable as that of the normal equation. Moreover, due to greater freedom in the choice of a pivot order, the augmented system factorization may produce a significantly sparser factor than that of the normal equation. Indeed the latter is actually a special case of (8.10) in which the first n pivots are chosen solely from D^2, regardless their stability and sparsity outcome.

The stable property of solving the augmented system has motivated many researchers to incorporate this approach into their implementation. There are other advantages for this approach, such as easy handling of free variables and dense columns, and its effortless extension to solving convex quadratic programming problems.

However, efficiency of the augmented system approach depends highly on keeping a consistent pivot ordering. One should avoid reordering pivots on every iteration and try to use the current pivot order in subsequent iterations as much as possible. The order is only updated occasionally when the KKT system has changed considerably.

One specific pivoting rule is again detecting "dense" columns in A and pivoting early those diagonal elements of D^{-2} which are not associated with the dense columns. One can set a density threshold to partition A into the sparse and dense parts as in (8.7).

A fixed threshold value approach works well only in a case when dense columns are easily identifiable, i.e., when the number of non-zero in each of them exceeds significantly the average number of entries in sparse columns. Whenever more complicated sparsity structure appears in A, a more sophisticated heuristic is needed.

Instead of the simple column partition (8.7), one may consider more complicated sparsity structure and the following partition of A:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}. \tag{8.11}$$

Here A_{11} is supposed sparse and is assumed to create a sparse normal matrix $A_{11}A_{11}^T$, A_{12} is a small set of "troubling" columns (either dense columns

or columns associated with free variables), and $(A_{21}\ A_{22})$ represents a set of "troubling" rows.

Once the partition (8.11) is determined, (8.4) becomes

$$\begin{pmatrix} D_1^{-2} & & A_{11}^T & A_{21}^T \\ & D_2^{-2} & A_{12}^T & A_{22}^T \\ A_{11} & A_{12} & & \\ A_{21} & A_{22} & & \end{pmatrix} \begin{pmatrix} d_{x1} \\ d_{x2} \\ d_{y1} \\ d_{y2} \end{pmatrix} = \begin{pmatrix} \bar{c}_1 \\ \bar{c}_2 \\ \bar{b}_1 \\ \bar{b}_2 \end{pmatrix}.$$

The structure of this system shows immediately which block, such as D_1^{-2}, can be inexpensively pivoted out, and which block, such as D_2^{-2}, should be pivoted lately.

The elimination of D_1^{-2} causes very limited fill-ins and reduces the KKT system to

$$\begin{pmatrix} D_2^{-2} & A_{12}^T & A_{22}^T \\ A_{12} & -A_{11}D_1^2A_{11}^T & -A_{11}D_1^2A_{21}^T \\ A_{22} & -A_{21}D_1^2A_{11}^T & -A_{21}D_1^2A_{21}^T \end{pmatrix}. \tag{8.12}$$

The elimination of D_2^{-2} should be delayed after all attractive pivot candidates from $A_{11}D_1^2A_{11}^T$ and $A_{21}D_1^2A_{21}^T$ blocks are exploited.

8.2.3 Numerical phase

So far we have extensively discussed the symbolic phase—the pivoting rule and pivoting order. Now we turn our attention to the numerical phase of a sparse symmetric system solver. This is a well developed area both in theory and in computational practice. Here we demonstrate several implementation techniques of the numerical factorization phase in the normal equation approach. These methods could be applied to the general symmetric decomposition of the augmented system as well.

Let $M = AD^2A^T$ and consider its Choleski factorization $L\Lambda L^T = M$, where L is a lower triangular matrix and Λ is a diagonal matrix. The basic formulae for computing the column j of L, denoted by $L_{\cdot j}$, and the pivot Λ_{jj} are:

$$\begin{array}{rcll} \Lambda_{11} & = & M_{11}, & \\ L_{\cdot 1} & = & \frac{1}{\Lambda_{jj}} M_{\cdot 1}, & \\ \Lambda_{jj} & = & M_{jj} - \sum_{k=1}^{j-1} L_{jk}^2 & j \geq 2, \\ L_{\cdot j} & = & \frac{1}{\Lambda_{jj}} \left(M_j - \sum_{k=1}^{j-1} (\Lambda_{kk} L_{jk}) L_k \right) & j \geq 2. \end{array} \tag{8.13}$$

Several methods have been developed to compute the factorization. They all exploit sparsity of the matrix but use different storage techniques in computations. These calculations can be organized either by rows or by columns. During the *row*-Choleski factorization the rows of the Choleski factor L are computed one by one.

The commonly used factorization is the *column*-Choleski factorization in which the columns of L are computed one by one as in (8.13). Its efficient

implementations can be found, for example, in the Yale Sparse Matrix Package and Waterloo SPARSPAK. This method is also called *left-looking* factorization, because the computation of column $L_{.j}$ follows the left-to-right order. Its implementation uses dynamic linked lists to look at all "left" columns when computing the current pivot and column, and a double precision work array to accumulate the column modifications and to resolve the non-zero matching between different columns.

Another commonly used approach is the *submatrix*-Choleski factorization, also referred to as the *right-looking* factorization. In this approach, once a column $L_{.j}$ has been computed, we immediately update its contributions to all subsequent columns, i.e. to all columns on its right side using (8.13). In this method the matching of non-zero during the process is not trivial, but several solutions have been found. Interest in this method has been increased in the past few years because of its ability to better exploit high performance architecture and memory hierarchy.

We now present several numerical "tricks" that work very well in interior-point methods. These techniques all based on using matrix-vector operations in a "dense" mode (assuming matrices and vectors are complete dense) to reduce the overhead computation and book-keeping map in a sparse mode using indirect index addressing and sophisticated memory referencing.

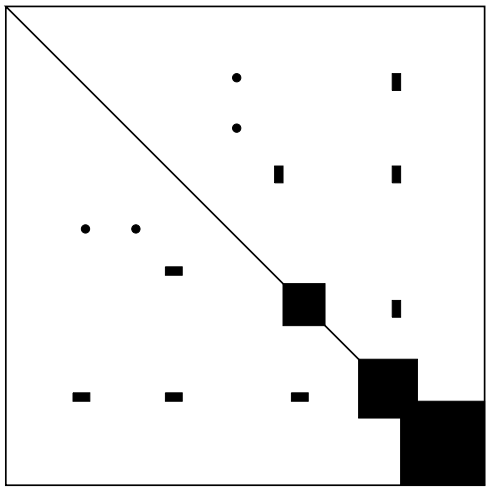


Figure 8.1: Illustration of dense sub-factors in a Choleski factorization.

Dense window

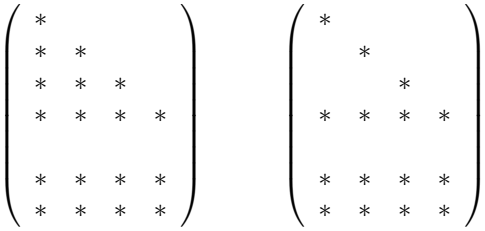
The most straightforward improvement of the factorization is exploitation of a *dense window*. In practice, some triangular sub-factors become completely

dense near the end of the Choleski factorization; see Figure 8.1. Therefore, we can treat these blocks complete dense and use dense matrix factorization, even though there may still be some zeros in this block. This is called a dense window. In doing so we avoid the overhead of sparse computation, such as indirect index addressing and memory referencing. It might also be beneficial to treat some almost-dense columns complete dense and to include them in a dense window.

Supernode

It is often observed that several columns in L tend to have the same sparsity pattern below the diagonal. Such a block of columns is called a *supernode* and it can be treated as a dense submatrix. The supernode name comes from the elimination graph representation of the Choleski decomposition, because these nodes (columns) more-or-less share the same set of adjacent nodes and they can be grouped as a single "super" node (share the same addressing), as illustrated below:

Supernode Type 1 Supernode Type 2



Both types of supernodes can be exploited in a similar manner within the numerical factorization. Similar to the dense window technique, the use of supernodes increases the portion of matrix-vector operations in the dense mode, and thereby saves on indirect addressing and memory referencing. Specifically, the following operations take advantage of supernodes:

- When column j is a member of a supernode, the operation of computing $L_{\cdot j}$ and other columns of the supernode are done in the dense mode.
- When column j is not a member of a supernode but it has a summation term from a set of columns that belong to a supernode, a temporary work array is used to accumulate the sum from the whole supernode in the dense mode before the term is added to $L_{\cdot j}$.

Sometime it is even beneficial to treat some zeros as non-zeros in L to create supernodes. The introduction of zeros does not necessarily lead to an increase in the memory allocation. This is due to the fact that only indexes of the last column in a supernode are booked, so there is a saving in index addressing.

Cache memory

Computers has a memory hierarchy consisting a slow and large main memory and a fast and small cache memory. Computation will be more efficient if

memory references are stored in the cache memory so they can be fetched faster. Thus, it is advisable to set an upper bound on the number of non-zeros in each supernode to such that they can be all stored in the cache memory during the dense mode matrix-vector operations. One should leave 5 percent of the cache memory for overhead.

Of course, such partition of large supernodes leads to more overhead computation. An advise is to also impose a lower bound on the size of supernodes since the extra work in constructing the work array may not pay off if the size of the supernode is too small.

Block Choleski factorization

Another possibility is to partition L into smaller, presumably dense blocks. For example, try to divide L into block diagonal dense submatrices. This technique is very effective in some cases, because a typical Choleski factor contains many such blocks, the largest of which is usually the dense window located at the bottom of L. Consider the following matrix:

$$AD^2A^T = \begin{pmatrix} M_{11} & M_{21}^T \\ M_{21} & M_{22} \end{pmatrix},$$

with an additional simplifying assumption that the blocks L_{11} and L_{22} of L are dense matrices. The Choleski factorization of this matrix can be computed in the following steps:

1. Factorize $L_{11}\Lambda_{11}L_{11}^T = M_{11}$.
2. Compute $L_{21} = M_{21}(L_{11}^{-1})^T$.
3. Compute $\hat{M}_{22} = M_{22} - L_{21}\Lambda_{11}L_{21}^T$.
4. Factorize $L_{22}\Lambda_{22}L_{22}^T = \hat{M}_{22}$.

The advantage of this procedure is that steps 1 and 4 can be performed in the dense mode.

Loop unrolling

Dense mode computation can be further specialized to exploit a loop unrolling technique. Let a be the target column, b the source column, and α the multiplier kept in a single register. Then the steps performed by a computer to execute the transformation $a \leftarrow a + \alpha \cdot b$ can be written as follows:

1. Read $a(i)$ from the memory.
2. Read $b(i)$ from the memory.
3. Compute $a(i) + \alpha \cdot b(i)$.
4. Store the result in the memory.

Consequently, three memory references in steps 1, 2, and 4 are associated with only one arithmetic multiplication in step 3.

During a typical inner loop of the factorization, several multiple columns are added to a single column; see (8.13). This opens a possibility to unroll the loop over the multiple column transformation. Let a be the target column, b, c, d, e, f and g the source columns, and $\alpha(1), \ldots, \alpha(6)$ their multipliers kept in a single register. A loop rolling technique to compute

$$a \leftarrow a + \alpha(1)b + \alpha(2)c + \alpha(3)d + \alpha(4)e + \alpha(5)f + \alpha(6)g$$

is to execute the above procedure 6 times and uses total 18 memory references and 6 multiplications. However, a 6-step loop unrolling technique consists of first reading a, b, c, d, e, f, g, then performing 6 multiplications, and finally storing new a. This execution needs only eight memory references. Hence, 10 memory references have been saved compared with the loop rolling execution. The loop unrolling technique generally makes considerable time savings on many different computer architectures.

8.2.4 Iterative method

An alternative to solve the KKT system (8.4) or the normal equation (8.5) is the *iterative* method, e.g., the conjugate gradient method. This method automatically exploits the sparse structure of the system because it neither uses nor stores any inverse matrix. Its effectiveness highly depends on the selection of an appropriate and simple preconditioner. In solving general linear programs the iterative method seems not competitive with the direct method, but it becomes highly successful in solving special LP problems such as network-flow problems.

Consider the network-flow problem, where A matrix is the node-arc incidence matrix for a network with $m + 1$ nodes and n arcs (For ease of notation, an arbitrary row in A is assumed to have been deleted so that A has full row rank m). Let $A = (A_B, A_N)$ where A_B is a basis of A. Then,

$$AD^2A^T = \left(A_B D_B^2 A_B^T + A_N D_N^2 A_N^T\right).$$

If the diagonal components of D_B are all greater than or equal to the diagonal components of D_N, then we expect that $A_B D_B^2 A_B^T$ becomes a dominate block and it is a good estimation of AD^2A^T. The following theorem indicates "why", whose proof is derived from Exercise 8.1.

Theorem 8.1 *Choose π so that A_π contains a basis of A and $D_{ii} \geq D_{jj}$ for all $i \in \pi$ and $j \notin \pi$. Then,*

$$(2m^3 + 1)I \succeq (A_\pi D_\pi^2 A_\pi^T)^{-.5}(AD^2A^T)(A_\pi D_\pi^2 A_\pi^T)^{-.5} \succeq I.$$

Thus, $A_B D_B^2 A_B^T$ is a reasonable preconditioner for AD^2A^T, where B is a basis of π. Note that A_B can be reordered as a triangular matrix so that the Choleski factor of $A_B D_B^2 A_B^T$ is A_B itself after a permutation. Furthermore, B can be found by the maximum-spanning tree algorithm where D_{jj} is the weight of arc j. This algorithm is very cost-effective.

8.3 High-Order Method

If a direct approach is used to solve the KKT system, in each iteration a matrix factorization (8.4) or (8.5) is computed and followed by several backsolve steps. The factorization phase, $O(n^3)$ operations for a dense matrix, consumes the major amount of work, and the backsolve phase, at most $O(n^2)$ operations, is usually significantly easier in theory as well as in practice. An obvious idea, known from different applications of the Newton method, is to reuse the factorization in several subsequent iterations or, equivalently, to repeat several backsolves to generate a better next iterate. We call such an approach a *high-order method*. The goal is to reduce the total number of interior point iterations and therefore the total number of factorizations as well.

8.3.1 High-order predictor-corrector method

The second-order predictor-corrector strategy has two components: one is an adaptive choice of the barrier parameter γ and the other is the computation of a sort of second-order approximation to the central path. For simplicity we illustrate the algorithm with a feasible starting point.

The first step of the predictor-corrector strategy is to compute the predictor direction of the predictor-corrector algorithm in Section **??**. Recall that the predictor direction solves the Newton equation system (3.38) for $\gamma = 0$ and is denoted with $d^p := d(x^k, s^k, 0)$. It is easy to show that if a step of size θ is taken along this direction, then the complementarity gap is reduced by the factor $(1 - \theta)$. Therefore, the larger step can be made, the more progress can be achieved. On the other hand, if the step-size in this direction is small, then the current point is probably too close to the boundary. In this case the barrier parameter should not be reduced too much in order to move a way from boundary like the corrector step.

Thus, it is reasonable to use this possible complementarity gap reduction in the predictor step to adjust the new barrier parameter γ. After the predictor direction is computed, the maximum step-sizes θ_p and θ_d along this direction in the primal and dual spaces are determined to preserve nonnegativity of $(x(\theta_p), s(\theta_d))$. The possible new complementarity gap

$$n\mu^+ := (x + \theta_p d_x^p)^T (s + \theta_d d_s^p).$$

Then, the barrier parameter is chosen using the heuristic

$$\gamma := \Big(\frac{\mu^+}{\mu^k}\Big)^2 \min\{\frac{\mu^+}{\mu^k}, \eta\} \tag{8.14}$$

for a constant $\eta \in (0, 1)$. We could come back to compute the actual direction $d(x^k, s^k, \gamma)$ from (3.38) where γ is given above. But we like to do more, which is the second component of the second-order predictor-corrector method.

Note that we ideally want to compute a direction such that the next iterate is perfectly centered for $\gamma\mu^k$, i.e.,

$$(X^k + D_x)(s^k + d_s) = \gamma\mu^k e.$$

The above system can be rewritten as

$$S^k d_x + X^k d_s = -X^k s^k + \gamma \mu^k e - D_x d_s. \tag{8.15}$$

Observe that in the "first-order" direction $d = d(x^k, s^k, \gamma)$ in equation (3.38), we have ignored the second order term $D_x d_s$ on the right-hand side and it becomes the residual error. This needs to be corrected: Instead of setting the second order term equal to zero, we would approximate $D_X d_s$ on the right-hand side using the available predictor direction $D_x^p d_s^p$. The actual direction d is then computed from system (8.15) with parameter γ chosen through (8.14). (Again, the matrix of the system is already factorized and it is "free" now.) We finally choose the next iterate

$$(x^{k+1}, s^{k+1}) = (x(\bar{\theta}_p), s(\bar{\theta}_d)) \in \mathcal{N}_\infty^-(\eta)$$

for η close to 1.

We should note here that the second-order predictor-corrector method basically tries to approximate the second-order Taylor expansion of the central path. A single iteration of the method needs two solves of the same large but sparse linear system for two different right hand sides. The benefit of the method is, we obtain a good estimate for the barrier parameter γ and a second-order approximation to the central path. Indeed computational practice shows that the additional solve cost of this method is more than offset by a reduction in the total number of iterations (factorizations).

Why not use even higher-order Taylor expansions? Indeed, in solving many large scale linear programs where the factorization is extremely expensive and the need to save on the number of factorizations becomes more important, a high-order predictor-corrector method is beneficial. We will explain the method in the next section.

8.3.2 Analysis of a high-order method

Now we would like to provide some theoretical justification for using the techniques involved in the high-order method.

One theoretical support of the method is already seen in Section **??**, where we showed that $A(X^{k+1})^2 A^T$ only differs slightly from $A(X^k)^2 A^T$, and it is sufficient to inverse a matrix AD^2A^T to generate next iterate where D is still close to X^k. This justifies that the normal matrix could be used repeatedly.

Another support relates to the neighborhoods used in the high-order method. Among all existing path-following (infeasible or feasible) LP algorithms, the theoretical iteration complexity of small-neighborhood ($\mathcal{N}_2$) algorithms is $O(\sqrt{n}L)$, and the complexity of wide-neighborhood ($\mathcal{N}_\infty$ or $\mathcal{N}_\infty^-$) algorithms is at least $O(nL)$. In contrast, wide-neighborhood algorithms outperform small-neighborhood ones by a big margin in practice. It seems that smaller neighborhoods generally restrict all iterates moved by a short step and they might be too conservative for solving real LP problems.

To support using the wide-neighborhood and high-order Taylor expansions, we present a r-order Taylor expansion primal-dual path-following algorithm that is based on $\mathcal{N}_\infty^-(\beta)$ where β is any fixed constant in $(0,1)$. We show that its iteration complexity is $O(n^{\frac{r+1}{2r}}L)$ where $r \in [1,n]$. Again, each iteration uses $O(n^3)$ arithmetic operations. Note that if we let $r = O(n)$, then this iteration bound is asymptotical $O(\sqrt{n}L)$ as n increases.

Algorithm 8.1 *Given $(x^0,s^0) \in \mathcal{N}_\infty^-(\eta)$ with $\eta \in (0,1)$, and integer $r \geq 1$ and $\gamma \in (0,1)$. Set $k := 0$.*

While $(x^k)^T s^k > \epsilon$ **do***:*

1. *First-order step: Solve for the first order direction $d^{(1)} := d(x^k,s^k,\gamma)$ from (3.38).*

2. *High-order steps: For $j = 2,3,\ldots,r$, solve for the jth order direction from*

$$\begin{array}{rcrcl} & A\ d_x^{(j)} & & = & 0, \\ -A^T\ d_y^{(j)} & & -d_s^{(j)} & = & 0, \end{array} \tag{8.16}$$

and

$$X^k d_s^{(j)} + S^k d_x^{(j)} \quad = \quad -\textstyle\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}. \tag{8.17}$$

3. *Compute the largest θ^k so that*

$$\begin{array}{rcl} x(\theta) & = & x^k + \sum_{j=1}^r (\theta)^j d_x^{(j)}, \\ y(\theta) & = & y^k + \sum_{j=1}^r (\theta)^j d_y^{(j)}, \\ s(\theta) & = & s^k + \sum_{j=1}^r (\theta)^j d_s^{(j)}, \end{array}$$

lies in $\mathcal{N}_\infty^-(\eta)$ for $\theta \in [0,\theta^k]$. Let

$$(y^{k+1},x^{k+1},s^{k+1}) := (y(\theta^k),x(\theta^k),s(\theta^k)).$$

4. *Let $k := k+1$ and return to Step 1.*

Note that for $r = 1$, Algorithm 8.1 is identical to the wide-neighborhood algorithm in Section **??**. For $r = 2$, it is close to the second-order predictor-corrector strategy described earlier; see Exercise 8.2.

In general, the step-size selection involves locating roots for each of $n+1$ polynomials with degree $2r$, which with specified error is in the complexity class NC and can be solved efficiently in theory. In practice, we need to locate only an approximate step-size. (Even for the case $r = 1$, one will never obtain the exact α^k since it is generally irrational.)

We will have a lower bound for θ^k:

$$\theta^k \geq \frac{const}{(n+1)^{\frac{r+1}{2r}}} \ ,$$

where

$$const = \frac{1-\eta}{4(1-\gamma)}\sqrt[r]{\eta\gamma}\,.$$

Thus, we need only to compute an approximate step-size, $\bar{\theta}$, such that

$$(x(\bar{\theta}), s(\bar{\theta})) \in \mathcal{N}_\infty^-(\eta),$$

and

$$\theta^k - \bar{\theta} \leq \frac{.001const}{n+1} \leq 0.001\theta^k,$$

that is, $\bar{\theta}$ will be at least a fraction, .999, of the exact step-size θ^k, and it approaches above θ^k as $n \to \infty$.

We may compute such an approximate step-size using the bisection method. We know that the step-size must be in

$$\left[\frac{const}{(n+1)^{\frac{r+1}{2r}}}, \frac{1}{1-\gamma}\right].$$

Obviously, the total number of operations of this process is of order $nr(\log n)$. Even when $r = n$, the cost, $n^2(\log n)$, is well below n^3.

We now present the main complexity result.

Theorem 8.2 *Given any initial point in $\mathcal{N}_\infty^-(\eta)$ for any constant $\eta \in (0,1)$, Algorithm 8.1, with any constant $0 < \gamma < 1$, will terminate in $O(n^{\frac{r+1}{2r}}\log((x^0)^T s^0/\epsilon))$ iterations, where $r \in [1,n]$, and each iteration uses $O(n^3)$ arithmetic operations.*

As we can see that if $r = n$ and n increases, the iteration complexity of the algorithm tends to $O(\sqrt{n}\log((x^0)^T s^0/\epsilon))$ asymptotically. Furthermore, a popular choice for γ in practice is not a constant but $\gamma = O(1/n)$. Interestingly, the asymptotical iteration complexity of the algorithm for such a choice of γ is still $O(\sqrt{n}\log((x^0)^T s^0/\epsilon))$. More precisely, we have

A number of implications and points can be drawn from the main result:

- The high-order Taylor expansion method, where iterative points move along a high-order polynomial curve, has been used in practice and partially analyzed in theory. The main result indicates that the use of this method also significantly improves the worst-case iteration complexity. The result provides a further theoretical base for using this approach.

- The order r of Taylor expansion has a diminishing role in improving the worst-case complexity result. Thus, we probably expect only the first few order steps really make a difference in algorithm performance. This seems what is observed in practice.

- The result also provides a justification for using the wider neighborhood $\mathcal{N}_\infty^-$, coupled with a high-order method. The theoretical complexity based on wider neighborhoods is not much worse than that based on smaller neighborhoods. We hope this is a significant step to bridge the gap between theory and practice.

- The result also indicates how insensitive of the value of γ, the centering weight, is in high-order power-series methods. Virtually, γ can be set to any positive number if iterative points move along a high-order polynomial curve. This implies that the method has a sufficient self-centering function even γ is close to 0. Note that, when $\gamma = 0$, the algorithm becomes the pure Newton method for the LP optimality condition.

8.4 Homogeneous and Self-Dual Method

In Section **??** we described a homogeneous and self-dual method to solve (LP) and (LD) simultaneously. From the implementation point of view, each iteration of the method solves the linear system (**??**) and (**??**).

It can be shown that (Exercise 8.4)

$$d_\theta = \gamma - 1.$$

Then eliminating d_s and d_κ, we face the KKT system of linear equations:

$$\begin{pmatrix} X^k S^k & -X^k A^T & X^k c\tau^k \\ AX^k & 0 & -\tau^k b \\ -\tau^k c^T X^k & \tau^k b^T & \tau^k \kappa^k \end{pmatrix} \begin{pmatrix} (X^k)^{-1} d_x \\ d_y \\ (\tau^k)^{-1} d_\tau \end{pmatrix}$$

$$= \begin{pmatrix} \gamma\mu^k e - X^k s^k \\ 0 \\ \gamma\mu^k - \tau^k \kappa^k \\ 0 \end{pmatrix} + (1-\gamma) \begin{pmatrix} -X^k \bar{c} \\ \bar{b} \\ \tau^k \bar{z} \end{pmatrix}.$$

Thus, the dimension of the system is increased only by 1 over the case when strictly feasible points for both (LP) and (LD) are known and used for starting primal-dual interior-point algorithms. (It seems that the benefit of knowing a starting interior point is not great.)

All implementation techniques discussed earlier for feasible-starting interior point algorithms can be used in the homogeneous and self-dual method. For example, If the second-order predictor-corrector scheme is used, it means that we have 3 solves instead of 2 for each factorization. Again, the additional solve cost is still more than offset by a reduction in the total number of iterations (factorizations), and all favorable features discussed in Section **??** of the method are retained.

It is also possible to take different step-sizes to update x and s. In doing so special attention should be paid to update τ since it couples both the primal and dual.

8.5 Optimal-Basis Identifier

Contrary to the simplex algorithm an interior-point algorithm never generates the exact optimal solution during its iterative process; instead it generates an

infinite sequence converging towards an optimal solution. Thus, the algorithm discussed produces an approximate optimal basic solution only if the optimal solution is unique (which is very rare in practice). In fact, in the case that either multiple primal or dual solutions exist, the sequence converges to the analytic center of the optimal face as discussed before. Therefore, an important problem is to generate an optimal basic solution from of interior-point algorithms, which is desirable in solving many practical problems.

It can be shown that if a pair of exact primal and dual solutions is known, then an optimal basic solution can be produced in strongly polynomial time using a simplified a pivoting or simplex method. We now discuss a algorithm which combines the termination scheme in Section **??** and the pivoting method to produce an optimal basic solution.

Consider solving (LP). It is well-known that any optimal solution (x^*, y^*, z^*) must satisfy the complementarity slackness condition $x_j^* z_j^* = 0$ for each j. Moreover, it is known from Theorem 1.17 that there exists a strictly complementary solution that satisfies $x_j^* + z_j^* > 0$ for each j, and the complementarity partition (P^*, Z^*) is unique. The pair (P^*, Z^*), where $Z = \{1, \ldots, n\} \setminus P$ for any index set P, determines an optimal partition.

Recall that (B, N) denote a partition of the variables into basic and non-basic variables. (B, N) is an optimal basis, if B is non-singular and

$$x_B = A_B^{-1} b \geq 0; \quad x_N = 0$$

and

$$y = A_B^{-T} c_B; \quad s_B = c_B - A_B y = 0; \quad s_N = c_N - A_N^T y \geq 0.$$

8.5.1 A pivoting algorithm

Given a complementary solution pair, a pivoting algorithm can construct an optimal basis in less than n pivoting steps. Below we shall discuss the algorithm and its implementation. For convenience we assume that a set of artificial variables has been added to the problem (LP). Let $V = \{n+1, \ldots, n+m\}$ denote the set of artificial variables; naturally, we must have $x_V = 0$ in any optimal solution. Furthermore, we assume that a strictly complementary solution is known. Hence, we assume that:

- We know the complementarity partition (P^*, Z^*) and $V \subseteq Z^*$.
- We know an optimal primal solution x^* such that $Ax^* = b, \quad x_{Z^*}^* = 0$ and $x_{P^*}^* \geq 0$.
- We know an optimal dual solution (y^*, s^*) such that $A^T y^* + s^* = c$, $s_{Z^* \setminus V}^* \geq 0$ and $s_{P^*}^* = 0$.

The algorithm consists of a primal and a dual phase. We start with a description of the primal phase.

Let (B, N) be any partition of the variables of the problem (LP) into basic and non-basic parts. Let

$$x_B := A_B^{-1}(b - Nx_N^*) = x_B^* \geq 0.$$

Here solution x_B^* is called a *super-basic* solution since some of non-basic variables x_N^* may not be zero, and variables of x_N^* that are not zero are called super-non-basic variables. For each of super-non-basic variables, the primal phase is to either move it to zero or pivot it into basis B using the simplex (pivoting) step. The resulting basis will be primal optimal, because it is feasible and it is still complementary with respect to the dual optimal solution (y^*, s^*). Each moving or pivoting step reduces the number of super-non-basic variables at least by one. Since the number of super-non-basic variables cannot exceed $|P^*|$, the primal phase terminates after at most $|P^*|$ steps.

Now we will formally state the primal phase.

Algorithm 8.2

1. *Choose a basis* B *and let* $x = x^*$.
2. *While*$(\exists\ j \in P^* \setminus B : x_j \neq 0)$
3. *Use a primal ratio test to move variable* x_j *to zero if we can keep* $A_B^{-1}(b - Nx_N) \geq 0$, *or pivot it into the basis.*
4. *Update* x, *or* (B, N) *and* x.
5. *end while*
6. B *is a primal optimal basis.*

It is always possible to choose an initial basis B in Step 1. One possible choice is $B = V$, the set of artificial variables. Algorithm 8.2 can be viewed as a simplified version of the primal simplex method, because there is no pricing step in selecting an incoming variable and those incoming candidates are predetermined from $x_{P^*}^*$.

The dual phase of the algorithm is similar to the primal phase because, in this case, a super-basic dual solution is known, which means that some of the reduced costs of s_B^* might not be zero. Similarly to the primal phase, those non-zero reduced costs in s_B^* can either be moved to zero or the corresponding primal variable has to be pivoted out of basis B. The dual phase can be stated as follows:

Algorithm 8.3

1. *Choose a basis* B *and let* $y = y^*,\ s = c - A^T y$.
2. *While*$(\exists\ j \in Z^* \cap B : s_j \neq 0)$
3. *Use the dual ratio test to move variable* s_j *to zero if we can keep* $c_N + N^T B^{-T}(s_B - c_B) \geq 0$, *or pivot it out of the basis.*
4. *Update* (y, s) *or* (B, N) *and* (y, s).
5. *end while*
6. B *is a dual optimal basis.*

If the initial basis B of the dual phase is primal optimal, i.e, $x^*_B := B^{-1}b \geq 0$ and $x^*_{Z^*} = 0$, then it remains primal optimal throughout all steps of Algorithm 8.3 because $x^*_N = 0$ and all pivots are primal degenerate. Once Algorithm 8.3 terminates, the final basis is both primal and dual feasible and hence optimal. Algorithm 8.3 can be viewed as a simplified version of the dual simplex method, because there is no pricing step in selecting an outgoing variable and those outgoing candidates are predetermined from $s^*_{Z^*}$. Furthermore, the number of moves or pivots in the dual phase cannot exceed $|Z^*|$.

In summary, Algorithms 8.2 and 8.3 generate an optimal basis after at most n moving or pivoting steps. In practice, the total number of steps is dependent on the level of primal and dual degeneracy of the problem.

8.5.2 Theoretical and computational issues

The algorithm presented in the previous subsection assumes that an exact optimal solution is known. This assumption is never met in practice, because the primal-dual algorithm only generates a sequence of solutions converging towards an optimal solution. Furthermore, due to the finite precision of computations, the solution returned by an interior-point algorithm is neither exactly feasible nor exactly complementary.

Let (x^k, y^k, z^k) be the iterate generated by an algorithm on iteration k and (P^k, Z^k) be a guess of the complementarity partition generated on iteration k. Now define the following perturbed problem:

$$\text{minimize} \quad (c^k)^T x \quad \text{s.t.} \quad Ax = b^k; \quad x \geq 0, \tag{8.18}$$

where

$$b^k = P^k x^k_{P^k}; \quad c^k_{P^k} = (P^k)^T y^k \quad \text{and} \quad c^k_{Z^k} = (\bar{P}^k)^T y^k + z^k_{Z^k}.$$

Assume that variables in (8.18) are reordered such that $x = (x_{P^k}, x_{Z^k})$ then the vector $(x, y, s) = ((x^k_{P^k}, 0), y^k, (0, z^k_{Z^k}))$ is a strictly complementary solution to (8.18). Moreover, if x^k converges towards an optimal primal solution and P^k converges towards P^*, then b^k converges towards b and, similarly, c^k converges towards c. Therefore the two problems (LP) and (8.18) will eventually become close and share some same optimal bases according to Exercises 8.3 and the following theorem.

Theorem 8.3 *Let B be an optimal basis for $LP(A, b^k, c^k)$. Then, there is $0 < \bar{t} < \infty$ such that B must be also an optimal basis for the original $LP(A, b, c)$ when $(x^k)^T s^k \leq 2^{-\bar{t}}$. Furthermore, if $LP(A, b, c)$ has rational data, then $\bar{t} \leq O(L)$.*

This advocates for an application of the above basis identification procedure to the perturbed problem (8.18), since an optimal complementary solution to problem (8.18) is known, and it will be an optimal basis for (LP) when problem (8.18 is near (LP).

An important practical issue is how to select P^k that equals to the complementarity partition P^*. A trivial one is

$$P^k = \{j : x_j^k \geq z_j^k\}. \tag{8.19}$$

A more practically effective choice is

$$P^k = \{j : |d_{x_j}{}^d|/x_j^k \leq |d_{s_j}{}^d|/s_j^k\}, \tag{8.20}$$

where (d_x^d, d_s^d) is the primal-dual predictor direction. These quantities are scaling invariant. It uses the relative variable change to indicate the optimal partition. This indicator is justified by the theory of Section **??**, where they converges to 1 for $j \in P^*$ and to 0 otherwise.

Another question is the choice of the right time to start the pivoting procedure. According to Theorem 8.3 the generated basis can only be expected to be the correct optimal basis of (LP) if the interior point solution is almost optimal and P^k is a good guess for P^*. A reasonable and practical criterion is the moment when fast (quadratic) convergence of the primal-dual gap μ^k to zero occurs, which is also consistent to the theory of Section **??**.

8.6 Notes

The use of a presolver is an old but effective idea, see for example, Brearley et al. [**?**]; its role was acknowledged in many simplex algorithm optimizers. The simplex method for LP works with sparse submatrices of A (bases) [**?**] while any interior-point algorithm needs an inversion of a considerably denser matrix AA^T. Consequently, the potential savings resulting from an initial problem reduction may be larger in interior-point implementations. This is the reason why the presolve analysis has recently enjoyed great attention [**?**, **?**, **?**, **?**, **?**, **?**, 215, **?**]. An additional important motivation is that large-scale LP problems are solved routinely nowadays and the amount of redundancy is increasing with the size of the problem.

When discussing the disadvantages of the normal equations approach in Section 8.2.1, we have mentioned the negative consequences of splitting free variables. Sometimes it is possible to generate a finite explicit bound on a free variable [**?**] and avoid the need of splitting it. Subramanian et al. [**?**] and Andersen [**?**] report that in some cases the computational saving from removing the linearly dependent constraints are significant.

Exact solution of the *sparsity problem* of Section 8.1 is an NP–complete problem ([**?**]) but efficient heuristics [**?**, **?**, **?**] usually produce satisfactory non-zero reductions in A. The algorithm of [**?**], for example, looks for such a row of A that has a sparsity pattern being the subset of the sparsity pattern of other rows and uses it to pivot out non-zero elements from other rows. Also, the postsolver analysis has been discussed extensively in [**?**].

Most general purpose interior-point codes use the *direct* method [**?**] to solve the KKT system. Two competitive direct methods are: the *normal equation*

approach [?, ?] and the *augmented system* approach. The former works with a smaller positive definite matrix, and the latter requires factorization of a symmetric indefinite matrix.

The normal equation approach was used among very first "professional" interior-point implementations [?, ?, ?]. The success of their application of the Choleski factorization relies on the quality of a pivoting order for preserving sparsity [?, ?]. To find an optimal order or permutation is an NP-complete problem [331]). Two effective heuristics described in this book, the *minimum degree* and the *minimum local fill-in* order rules, are due to Duff [?] and George and Liu [?, ?]. In the minimum-degree order l_i is actually the Markowitz merit function applied to a symmetric matrix [?]. For details, the reader is referred to an excellent summary in [?]. Another efficient technique to determine the pivot order has been proposed in Mészáros [?]. The remedy to the rank deficiency arising in the Schur complement mechanism is due to Andersen [?]. His approach employs an old technique due to Stewart [?].

The augmented system approach is an old and well understood technique to solve a least squares problem [?, ?, ?, ?]. It consists in the application of the Bunch-Parlett [?] factorization to the symmetric indefinite matrix. Mehrotra's augmented system implementation [?, ?], for example, is based on the Bunch-Parlett factorization [?] and on the use of the generalized Markowitz [?] count of type (8.6) for 2×2 pivots. Maros and Mészáros [?] give a detailed analysis of this issue as well. The stable condition of the augmented system approach motivated many researchers to incorporate it into their LP codes; see [?, ?, ?, ?, ?]. Other advantages include easy handling of free variables and dense columns, and a straightforward extension to solving convex quadratic programming problems [?, ?, ?].

In the numerical factorization, George and Liu [?] demonstrate how the Choleski factorization can be organized either by rows or by columns. Several enhancements can be found in [?, ?] and [?, ?]. The Yale Sparse Matrix Package is due to [?] and the Waterloo SPARSPAK Package is due to [?].

Lustig et al. [?] explored the supernode in their implementation. The effect of the supernodal method is highly hardware-dependent and several results can be found in the literature: the efficiency of the supernodal decomposition on the shared-memory multiprocessors is discussed by Esmond and Peyton [?], the exploitation of the cache memory on high-performance workstations is studied by Rothberg and Gupta [?] in the framework of the right-looking factorization, while the case of the left-looking factorization is investigated by Mészáros [?].

The iterative becomes highly successful in solving special LP problems, such as network-flow problems; see [?, ?, ?]. Theorem 8.1 is proved by Kaliski [?].

The first high-order method was incorporated into a dual affine-scaling method of AT&T's Korbx system [?]. An efficient high-order method was proposed by Mehrotra; his second-order predictor-corrector strategy [221] was incorporated in almost all primal-dual interior-point implementations. As shown in Mehrotra [?], the improvement from using orders higher than 2 seems very limited. Recently, Gondzio [?] has proposed a new way to exploit higher order information in a primal-dual algorithm and shown considerable improvement in solving

large-scale problems. His approach applies multiple centrality corrections and combines their use with a choice of reasonable, well-centered targets that are supposed to be easier to reach than perfectly centered (but usually unreachable) analytic centers. The idea to use targets that are not analytic centers comes from Jansen, Roos, Terlaky and Vial [**?**]. They define a sequence of traceable weighted analytic centers, called *targets*, that go from an arbitrary interior point to a point close to the central path. The algorithm follows these targets and continuously (although very slowly) improves the centrality of subsequent iterates. The targets are defined in the space of the complementarity products.

Another high-order approach, due to Domich et al. [**?**] uses three independent directions and solves an auxiliary linear program in a three dimensional subspace to find a search direction. The method of Sonnevend et al. [**?**] uses subspaces spanned by directions generated by higher order derivatives of the feasible central path, or earlier computed points of it as a predictor step. This is later followed by one (or more) centering steps to take the next iterate sufficiently close to the central path. Hung and Ye [**?**] has studied theoretically higher order predictor-corrector techniques, incorporated them in the homogeneous self-dual algorithm, and proved Theorem 8.2.

The fact of Exercise 8.4 was proved by Xu et al. [**?**], who also first implemented the homogeneous and self-dual algorithm and presented favorable computational results in solving both feasible and infeasible LP problems. Extensive implementation results were recently given by Andersen and Andersen [14]. They even discussed how the solution resulted from the homogeneous and self-dual model can be used in diagnosing the cause of infeasibility.

Recovering an optimal basis from a near-optimal solution is necessary in solving integer programming problems. We would also like to note that there are LP applications in which an optimal interior-point solution is preferable; see, e.g., Christiansen and Kortanek [**?**] and Greenberg [**?**].

Bixby and Lustig solve the basis-recovering problem using a Big-M version of Megiddo's procedure [**?**]. Their procedure drives both complementarity and feasibility to zero. Andersen and Ye [**?**] propose an alternative solution to this problem, which is the perturbed problem construction described in this book. For a discussion of linear algebra issues related to implementing the simplex or pivoting algorithm we refer the reader to the papers [**?**, **?**].

There are some open implementation issues. In many practical applications of linear programming, a sequence of closely related problems is solved, such as in branch and bound algorithms for integer programming an in column generation (cutting planes) methods. Obviously when two closely related problems are solved the previous optimal solution should be and could be used to solve the new problem faster. In the context of the simplex algorithm this aim is achieved by starting from the previous optimal basic solution, which is called the "warm-start." In the context of interior-point methods, an effective warm start procedure is difficult to find. Some hope comes from a particular application demonstrated in [130].

8.7 Exercises

8.1 *Let A be the node-arc incidence matrix with $m+1$ nodes and n arcs (For ease of notation, an arbitrary row in A is assumed to have been deleted so that A has full row rank m), and let D be an $n \times n$ positive diagonal matrix. Choose π so that A_π contains a basis of A and $D_{ii} \geq \beta \geq D_{jj}$ for all $i \in \pi$ and $j \notin \pi$. Then,*

$$\underline{\lambda}(A_\pi D_\pi^2 A_\pi^T) \geq \beta m^{-2}$$

and

$$\bar{\lambda}(AD^2A^T - A_\pi D_\pi^2 A_\pi^T) \leq 2\beta m.$$

8.2 *In Algorithm 8.1, let*

$$\begin{array}{rcccl} \triangle x & = & x^{k+1} - x^k & = & \sum_{j=1}^r (\theta^k)^j d_x^{(j)}, \\ \triangle y & = & y^{k+1} - y^k & = & \sum_{j=1}^r (\theta^k)^j d_y^{(j)}, \\ \triangle s & = & s^{k+1} - s^k & = & \sum_{j=1}^r (\theta^k)^j d_s^{(j)}. \end{array}$$

Show that $(\triangle x, \triangle s, \triangle y)$ satisfy

1. $$X^k \triangle s + S^k \triangle x = \theta^k(\gamma\mu^k e - X^k s^k) - \sum_{j=2}^{r} (\theta^k)^j \left(\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)} \right).$$

2. $$(x^k)^T \triangle s + (s^k)^T \triangle x = \theta^k(\gamma - 1)(x^k)^T s^k,$$
$$\triangle x^T \triangle s = 0.$$

3. $$\mu^{k+1} = [1 - \theta^k(1-\gamma)]\mu^k.$$

8.3 *Let B be an optimal basis for $LP(A, b^k, c^k)$ of Section 8.5.2. There there is a positive number $\zeta(A, b, c) > 0$ such that when $\left\|b^k - b\right\| < \zeta(A, b, c)$ and $\left\|c^k - c\right\| < \zeta(A, b, c)$, B is also an optimal basis for $LP(A, b, c)$, i.e., both solutions $\bar{x}_B$ of $B\bar{x}_B = b$ and $\bar{s} = c - A^T\bar{y}$, where $B^T\bar{y} = c_B$, are nonnegative.*

8.4 *In solving system of equations (**??**) and (**??**) of Section **??**, shown that*

$$d_\theta = \gamma - 1.$$

Bibliography

[1] M. Abdi, H. El Nahas, A. Jard, and E. Moulines. Semidefinite positive relaxation of the maximum-likelihood criterion applied to multiuser detection in a CDMA context. *IEEE Signal Processing Letters*, 9(6):165–167, June 2002.

[2] A. A. Ageev and M. I. Sviridenko, "An Approximation Algorithm for Hypergraph Max k-Cut with Given Sizes of Parts," *Proceedings of ESA'2000*, Lecture Notes in Computer Science 1879, (2000) 32-41.

[3] A. Y. Alfakih. Graph Rigidity via Euclidean Distance Matrices. *Linear Algebra and Its Applications* 310:149–165, 2000.

[4] A. Y. Alfakih. On Rigidity and Realizability of Weighted Graphs. *Linear Algebra and Its Applications* 325:57–70, 2001.

[5] A. Y. Alfakih, A. Khandani, H. Wolkowicz. Solving Euclidean Distance Matrix Completion Problems Via Semidefinite Programming. *Comput. Opt. and Appl.* 12:13–30, 1999.

[6] A. Y. Alfakih, H. Wolkowicz. On the Embeddability of Weighted Graphs in Euclidean Spaces. *Research Report CORR 98–12, University of Waterloo, Dept. of Combinatorics and Optimization*, 1998.

[7] A. Y. Alfakih, H. Wolkowicz. Euclidean Distance Matrices and the Molecular Conformation Problem. *Research Report CORR 2002–17, University of Waterloo, Dept. of Combinatorics and Optimization*, 2002.

[8] F. Alizadeh. *Combinatorial optimization with interior point methods and semi–definite matrices*. Ph.D. Thesis, University of Minnesota, Minneapolis, MN, 1991.

[9] F. Alizadeh. Optimization over the positive semi-definite cone: Interior–point methods and combinatorial applications. In P. M. Pardalos, editor, *Advances in Optimization and Parallel Computing*, pages 1–25. North Holland, Amsterdam, The Netherlands, 1992.

[10] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optimization*, 5:13–51, 1995.

[11] F. Alizadeh, J. P. A. Haeberly, and M. L. Overton, "Primal-dual interior point methods for semidefinite programming," Technical Report 659, Computer Science Department, Courant Institute of Mathematical Sciences, New York University, 1994.

[12] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton. Complementarity and nondegeneracy in semidefinite programming. *Math. Programming*, 77:111–128, 1997.

[13] B. Alkire and L. Vandenberghe. Convex optimization problems involving finite autocorrelation sequences. *Mathematical Programming*. To appear.

[14] E. D. Andersen and K. D. Andersen. The APOS linear programming solver: an implementation of the homogeneous algorithm. Working Paper, Department of Management, Odense University, Denmark, 1997.

[15] E. D. Andersen and Y. Ye. On a homogeneous algorithm for the monotone complementarity problem. Working Paper, Department of Management Sciences, The University of Iowa, IA, 1995. To appear in *Math. Programming*.

[16] G. Andersson and L. Engebrestsen, "Better Approximation Algorithms for Set Splitting and Not-All-Equal SAT," *Information Processing Letters* 65, (1998) 305-311.

[17] K. D. Andersen and E. Christiansen. A newton barrier method for minimizing a sum of Euclidean norms subject to linear equality constraints. Preprint 95-07, Department of Mathematics and Computer Science, Odense University, Denmark, 1995.

[18] M.F. Anjos and H. Wolkowicz. Strengthened semidefinite relaxations via a second lifting for the Max-Cut problem. *Discrete Appl. Math.*, 119(1-2):79–106, 2002. Foundations of heuristics in combinatorial optimization.

[19] K. M. Anstreicher. A standard form variant and safeguarded linesearch for the modified Karmarkar algorithm. *Math. Programming*, 47:337–351, 1990.

[20] K. M. Anstreicher and M. Fampa, "A long-step path following algorithm for semidefinite programming problems," Working Paper, Department of Management Science, The University of Iowa, Iowa City, 1996.

[21] K. M. Anstreicher, D. den Hertog, C. Roos, and T. Terlaky. A long step barrier method for convex quadratic programming. *Algorithmica*, 10:365–382, 1993.

[22] K. M. Anstreicher. On long step path following and SUMT for linear and quadratic programming. *SIAM J. Optimization*, 6:33-46, 1996.

[23] S. Arora, S. Rao and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. *The 35th STOC'04* 2004.

[24] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama, "Greedily finding a dense subgraph," in *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory (SWAT)*, Lecture Notes in Computers Science, 1097, 136-148, Springer-Verlag, 1996.

[25] James Aspnes, David Goldenberg, Yang Richard Yang. On the Computational Complexity of Sensor Network Localization. *ALGOSENSORS 2004*, in *LNCS* 3121:32–44, 2004.

[26] M. Badoiu, S. Har-Peled and P. Indyk, "Approximate Clustering via Core-sets," In *Proc. ACM Symp. Theory of Computing*, 2002.

[27] A. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete Comput. Geom.*, 13(2):189–202, 1995.

[28] M. B**u**adoiu. Approximation Algorithm for Embedding Metrics into a Two–Dimensional Space. *Proc. 14th SODA* 434-443, 2003.

[29] M. B**u**adoiu, E. D. Demaine, M. T. Hajiaghayi, P. Indyk. Low–Dimensional Embedding with Extra Information. *Proc. 20th SoCG* 320–329, 2004.

[30] O. Bahn, O. Du Merle, J. L. Goffin, and J. P. Vial. A cutting plane method from analytic centers for stochastic programming. *Math. Programming*, 69:45–74, 1995.

[31] Balas, E., Ceria, S. and Cornuejols, G., A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58:295–324, 1993.

[32] R. Banavar and A. Kalele. A mixed norm performance measure for the design of multirate filterbanks. *IEEE Transactions on Signal Processing*, 49(2):354–359, Feb. 2001.

[33] I. Bárány and Füredi. Computing the volume is difficult. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, page 442-447, ACM, New York, 1986.

[34] G. Barequet and S. Har-Peled, "Efficiently Approximating the Minimum-Volume Bounding Box of a Point Set in Three Dimensions," *J. Algorithms*, 38:91-109, 2001.

[35] A. Barvinok. Problems of Distance Geometry and Convex Properties of Quadratic Maps. *Disc. Comp. Geom.* 13:189–202, 1995.

[36] A. Barvinok. A Remark on the Rank of Positive Semidefinite Matrices Subject to Affine Constraints. *Disc. Comp. Geom.* 25(1):23–31, 2001.

[37] A. Barvinok. A Course in Convexity. *AMS*, 2002.

[38] D. A. Bayer and J. C. Lagarias. The nonlinear geometry of linear programming, Part I: Affine and projective scaling trajectories. *Transactions of the American Mathematical Society*, 314(2):499–526, 1989.

[39] D. A. Bayer and J. C. Lagarias. The nonlinear geometry of linear programming, Part II: Legendre transform coordinates. *Transactions of the American Mathematical Society*, 314(2):527–581, 1989.

[40] D. A. Bayer and J. C. Lagarias. Karmarkar's linear programming algorithm and Newton's method. *Math. Programming*, 50:291–330, 1991.

[41] M. Bellare and P. Rogaway. The complexity of approximating a nonlinear program. *Math. Programming*, 69:429-442, 1995.

[42] S. J. Benson and Y. Ye. DSDP User Guide. http://www-unix.mcs.anl.gov/∼ benson, November 2000.

[43] S. J. Benson, Y. Ye, and X. Zhang. Mixed linear and semidefinite programming for combinatorial optimization. *Optimization Methods and Software*, 10:515–544, 1999.

[44] S. J. Benson, Y. Ye, and X. Zhang. Solving large scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal of Optimization*, 10:443–461, 2000.

[45] A. Ben-Tal, M. Kocvara, A. Nemirovski, and J. Zowe. Free material optimization via semidefinite programming: the multiload case with contact conditions. *SIAM Review*, 42(4):695–715, 2000.

[46] A. Ben-Tal and A. Nemirovski. Robust truss topology design via semidefinite programming. *SIAM J. Optim.*, 7(4):991–1016, 1997.

[47] A. Ben-Tal and A. Nemirovski. Structural design via semidefinite programming. In *Handbook on Semidefinite Programming*, pages 443–467. Kluwer, Boston, 2000.

[48] A. Ben–Tal and A. S. Nemirovskii. Interior point polynomial time method for truss topology design. *SIAM J. Optimization*, 4:596–612, 1994.

[49] D. P. Bertsekas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, 1997.

[50] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995.

[51] D. Bertsimas and J. Nino-Mora. Optimization of multiclass queuing networks with changeover times via the achievable region approach: part ii, the multi-station case. *Mathematics of Operations Research*, 24(2), May 1999.

[52] D. Bertsimas and Y. Ye. Semidefinite relaxations, multivariate normal distributions, and order statistics. *Handbook of Combinatorial Optimization (Vol. 3)*, D.-Z. Du and P.M. Pardalos (Eds.) pp. 1-19, (1998 Kluwer Academic Publishers).

[53] P. Biswas, Y. Ye. Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization. *Proc. 3rd IPSN* 46–54, 2004.

[54] R. G. Bland, D. Goldfarb and M. J. Todd. The ellipsoidal method: a survey. *Operations Research*, 29:1039–1091, 1981.

[55] L. Blum, M. Shub and S. Smale. On a theory of computations over the real numbers: NP-completeness, recursive functions and universal machines. *Proc. 29th Symp. Foundations of Computer Science*, 387–397, 1988.

[56] H.L. Bodlaender, P. Gritzmann, V. Klee and J. Van Leeuwen, "The Computational Complexity of Norm Maximization", *Combinatorica*, 10: 203-225, 1990.

[57] K. H. Borgwardt. *The Simplex Method: A Probabilistic Analysis.* Springer, Berlin, 1987.

[58] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Science.* SIAM Publications. SIAM, Philadelphia, 1994.

[59] B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Software and Methods*, 11:613–623, 1999.

[60] B. Borchers. SDPLIB 1.0 : A collection of semidefinite programming test problems. Technical report, Faculty of Mathematics, Institute of Mining and Technology, New Mexico Tech, Socorro, NM, USA, July 1998.

[61] B. Borchers. Presentation at the Seventh DIMACS Implementation Challenge Piscataway, NJ, November 2, 2000.

[62] A. Brieden, P. Gritzmann, R. Kannan, V. Klee, L. Lovasz and M. Simonovits, "Deterministic and Randomized Polynomial-time Approximation of Radii," To appear in *Mathematika*.

[63] A. Brieden, P. Gritzmann, R. Kannan, V. Klee, L. Lovasz and M. Simonovits, "Approximation of Diameters: Randomization Doesn't Help," In *Proc. IEEE Symp. Foundations of Comp. Sci.*, 244-251, 1998.

[64] A. Brieden, "Geometric Optimization Problems Likely Not Contained in APX," *Discrete Comput. Geom.,* 28:201-209, 2002.

[65] P. Biswas and Y. Ye. Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization. Management Science and Engineering, Stanford, CA 94305, September 2003.

[66] P. Biswas and Y. Ye. A distributed method for solving semideinite programs arising from Ad Hoc Wireless Sensor Network Localization. Management Science and Engineering, Stanford, CA 94305, October 2003.

[67] N. Bulusu, J. Heidemann, D. Estrin. GPS-less low cost outdoor localization for very small devices. *TR 00-729, Computer Science, University of Southern California*, April, 2000.

[68] S. Burer and R. D. C. Monteiro. An efficient algorithm for solving the MAXCUT SDP relaxation. Manuscript, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA USA, December 1998.

[69] S. Burer, R. D. C. Monteiro, and Y. Zhang. Solving semidefinite programs via nonlinear programming. part I: Transformations and derivatives. Technical Report TR99-17. Department of Computational and Applied Mathematics, Rice University, TX, September 1999.

[70] S. Burer, R. D. C. Monteiro, and Y. Zhang. Solving semidefinite programs via nonlinear programming. part II: Interior point methods for a subclass of SDPs. Technical Report TR99-23. Department of Computational and Applied Mathematics, Rice University, TX, October 1999.

[71] G. Calafiore and M. Indri. Robust calibration and control of robotic manipulators. In *American Control Conference*, pages 2003–2007, 2000.

[72] S. A. Cook. The complexity of theorem-proving procedures. *Proc. 3rd ACM Symposium on the Theory of Computing*, 151–158, 1971.

[73] R. W. Cottle and G. B. Dantzig. Complementary pivot theory in mathematical programming. *Linear Algebra and its Applications*, 1:103-125, 1968.

[74] R. Cottle, J. S. Pang, and R. E. Stone. *The Linear Complementarity Problem*, chapter 5.9 : Interior–point methods, pages 461–475. Academic Press, Boston, 1992.

[75] G. B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, New Jersey, 1963.

[76] L. Danzer, B. Grünbaum and V. Klee. Helly's theorem and its relatives. in V. Klee, editor, *Convexity*, page 101–180, AMS, Providence, RI, 1963.

[77] T. Davidson, Z. Luo, and K. Wong. Design of orthogonal pulse shapes for communications via semidefinite programming. *IEEE Transactions on Communications*, 48(5):1433–1445, May 2000.

[78] J. E. Dennis and R. E. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice–Hall, Englewood Cliffs, New Jersey, 1983.

[79] C. de Soua, R. Palhares, and P. Peres. Robust H_∞ filter design for uncertain linear systems with multiple time-varying state delays. *IEEE Transactions on Signal Processing*, 49(3):569–575, March 2001.

[80] I. I. Dikin. On the convergence of an iterative process. *Upravlyaemye Sistemi*, 12:54–60, 1974. (In Russian).

[81] L. Doherty, L. E. Ghaoui, S. J. Pister. Convex Position Estimation in Wireless Sensor Networks. *Proc. 20th INFOCOM* Volume 3:1655–1663, 2001.

[82] A. Doherty, P. Parrilo, and F. Spedalieri. Distinguishing separable and entangled states. *Physical Review Letters*, 88(18), 2002.

[83] Y. Doids, V. Guruswami, and S. Khanna. The 2-catalog segmentation problem. In *Proceedings of SODA*, pages 378–380, 1999.

[84] C. Du, L. Xie, and Y. Soh. H_∞ filtering of 2-D discrete systems. *IEEE Transactions on Signal Processing*, 48(6):1760–1768, June 2000.

[85] H. Du, L. Xie, and Y. Soh. H_∞ reduced-order approximation of 2-D digital filters. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(6):688–698, June 2001.

[86] G. E. Dullerud and F. Paganini. *A Course in Robust Control Theory*, volume 36 of *Texts in Applied Mathematics*. Springer-Verlag, 2000.

[87] B. Dumitrescu, I. Tabus, and P Stoica. On the parameterization of positive real sequences and MA parameter estimation. *IEEE Transactions on Signal Processing*, 49(11):2630–2639, November 2001.

[88] M. E. Dyer and A. M. Frieze. The complexity of computing the volume of a polyhedron. *SIAM J. Comput.*, 17:967–974, 1988.

[89] M. E. Dyer, A. M. Frieze and R. Kannan. A random polynomial time algorithm for estimating volumes of convex bodies. *J. Assoc. Comp. Mach.*, 38:1–17, 1991.

[90] J. Edmonds. Systems of distinct representatives and linear algebra. *J. Res. Nat. Bur. Standards*, 71B, 1967, pages 241–245.

[91] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Moore, B. D. O. Anderson, P. N. Belhumeur. Rigidity, Computation, and Randomization in Network Localization. *Proc. 23rd INFOCOM*, 2004.

[92] G. Elekes. A geometric inequality and the complexity of computing volume. *Discr. Comput. Geom.*, 1:289–292, 1986.

[93] J. Elzinga and T. G. Moore. A central cutting plane algorithm for the convex programming problem. *Math. Programming*, 8:134–145, 1975.

[94] U. Faigle, W. Kern, and M. Streng, "Note on the Computaional Complexity of j-Radii of Polytopes in R^n," *Mathematical Programming*, 73:1-5, 1996.

[95] J. Falkner, F. Rendl, and H. Wolkowicz, "A computational study of graph partitioning," *Mathematical Programming*, 66(2):211-240, 1994.

[96] S. C. Fang and S. Puthenpura. *Linear Optimization and Extensions*. Prentice–Hall, Englewood Cliffs, NJ, 1994.

[97] L. Faybusovich. Jordan algebras, Symmetric cones and Interior-point methods. Research Report, University of Notre Dame, Notre Dame, 1995.

[98] L. Faybusovich. Semidefinite programming: a path-following algorithm for a linear-quadratic functional. *SIAM J. Optimization*, 6:1007–1024, 1996.

[99] U. Feige and M. X. Goemans. Approximating the value of two prover proof systems, with applications to max 2sat and max dicut. In *Proceedings of the 3nd Israel Symposium on Theory and Computing Systems*, pages 182–189, 1995.

[100] U. Feige and M. X. Goemans, "Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT," in *Proceedings of the Third Israel Symposium on Theory of Computing and Systems*, (1995) 182-189.

[101] U. Feige and M. Langberg, "Approximation algorithms for maximization problems arising in graph partitioning," to appear in *J. Algorithms*.

[102] U. Feige and M. Seltser, "On the densest k-subgraph problem," Technical report, Department of Applied Mathematics and Computer Science, The Weizmann Institute, Rehovot, September 1997.

[103] U. Feige, G. Kortsarz and D. Peleg, "The dense k-subgraph problem," to appear in *Algorithmica*.

[104] U. Feige and M. Langberg. Approximation algorithms for maximization problems arising in graph partitioning. *Journal of Algorithms*, 41:174–211, 2001.

[105] U. Feige and M. Langberg. The rpr^2 rounding technique for semidefinte programs. In *ICALP*, Lecture Notes in Computer Science. Springer, Berlin, 2001.

[106] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming : Sequential Unconstrained Minimization Techniques*. John Wiley & Sons, New York, 1968. Reprint : Volume 4 of *SIAM Classics in Applied Mathematics*, SIAM Publications, Philadelphia, PA, 1990.

[107] R. Freund and F. Jarre. An interior–point method for fractional programs with convex constraints. *Math. Programming*, 67:407–440, 1994.

[108] R. M. Freund. Dual gauge programs, with applications to quadratic programming and the minimum-norm problem. *Math. Programming*, 38:47–67, 1987.

[109] R. M. Freund. Polynomial–time algorithms for linear programming based only on primal scaling and projected gradients of a potential function. *Math. Programming*, 51:203–222, 1991.

[110] E. Fridman and U. Shaked. A new H_∞ filter design for linear time delay systems. *IEEE Transactions on Signal Processing*, 49(11):2839–2843, July 2001.

[111] K. R. Frisch. The logarithmic potential method for convex programming. Unpublished manuscript, Institute of Economics, University of Oslo, Oslo, Norway, 1955.

[112] A. Frieze and M. Jerrum. Improved approximation algorithms for max k-cut and max bisection. *Algorithmica*, 18:67–81, 1997.

[113] M. Fu, C. de Souza, and Z. Luo. Finite-horizon robust Kalman filter design. *IEEE Transactions on Signal Processing*, 49(9):2103–2112, Sept. 2001.

[114] M. Fu, Z.-Q. Luo and Y. Ye. Approximation algorithms for quadratic programming. manuscript, Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, CANADA L8S 4K1, 1996.

[115] T. Fujie and M. Kojima, "Semidefinite programming relaxation for nonconvex quadratic programs," *Journal of Global Optimization* 10:367-380, (1997).

[116] K. Fujisawa, M. Fukuda, M. Kojima, and K. Nakata. Numerical Evaluation of SDPA (SemiDefinite Programming Algorithm). In *High Performance Optimization*. Kluwer Academic Press, 1999, pp. 267–301.

[117] K. Fujisawa, M. Kojima and K. Nakata, "Exploiting Sparsity in Primal-Dual Interior Point Methods for Semidefinite Programming" Research Report B-324, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152, September 1997.

[118] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. An empirical study of epidemic algorithms in large scale multihop wireless networks. *UCLA/CSD-TR-02-0013, Computer Science, UCLA*, 2002.

[119] D. Gale. *The Theory of Linear Economic Models*. McGraw-Hill, New York, 1960.

[120] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, 1979.

[121] D. M. Gay. A variant of Karmarkar's linear programming algorithm for problems in standard form. *Math. Programming*, 37:81–90, 1987. Errata in *Math. Programming*, 40:111, 1988.

[122] J. Geromel. Optimal linear filtering under parameter uncertainty. *IEEE Transactions on Signal Processing*, 47(1):168–175, Jan. 1999.

[123] J. Geromel and M. De Oliveira. H_2/H_∞ robust filtering for convex bounded uncertain systems. *IEEE Transactions on Automatic Control*, 46(1):100–107, Jan. 2001.

[124] L. E. Gibbons, D. W. Hearn and P. M. Pardalos. A continuous based heuristic for the maximum clique problem. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 26:103-124, 1996.

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

[126] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright. On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. *Math. Programming*, 36:183–209, 1986.

[127] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of Assoc. Comput. Mach.*, 42:1115–1145, 1995.

[128] M. X. Goemans and D. P. Williamson, " Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming," in *33rd STOC*, pp. 443-452, 2001.

[129] J. L. Goffin, Z. Q. Luo, and Y. Ye. Complexity analysis of an interior-point cutting plane method for convex feasibility problem. *SIAM J. Optimization*, 6(3):638–652, 1996.

[130] J. L. Goffin and J. P. Vial. Cutting planes and column generation techniques with the projective algorithm. *J. Optim. Theory Appl.*, 65:409–429, 1990.

[131] D. Goldfarb and G. Iyengar. Robust portfolio selection problems. *Mathematics of Operations Research*. Submitted.

[132] D. Goldfarb, K. Scheinberg. Interior Point Trajectories in Semidefinite Programming. *SIAM J. Opt.* 8(4):871–886, 1998.

[133] D. Goldfarb and M. J. Todd. Linear Programming. In G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, editors, *Optimization*, volume 1 of *Handbooks in Operations Research and Management Science*, pages 141–170. North Holland, Amsterdam, The Netherlands, 1989.

[134] A. J. Goldman and A. W. Tucker. Polyhedral convex cones. In H. W. Kuhn and A. W. Tucker, Editors, *Linear Inequalities and Related Systems*, page 19–40, Princeton University Press, Princeton, NJ, 1956.

[135] G. H. Golub and C. F. Van Loan. *Matrix Computations*, 2nd Edition. Johns Hopkins University Press, Baltimore, MD, 1989.

[136] C. C. Gonzaga. Polynomial affine algorithms for linear programming. *Math. Programming*, 49:7–21, 1990.

[137] C. C. Gonzaga. An algorithm for solving linear programming problems in $O(n^3L)$ operations. In N. Megiddo, editor, *Progress in Mathematical Programming : Interior Point and Related Methods*, Springer Verlag, New York, 1989, pages 1–28.

[138] C. C. Gonzaga. Path following methods for linear programming. *SIAM Review*, 34(2):167–227, 1992.

[139] J. Graver, B. Servatius, H. Servatius. Combinatorial Rigidity. *AMS*, 1993.

[140] P. Gritzmann and V. Klee, "Inner and Outer j-Radii of Convex Bodies in Finite-Dimensional Normed Spaces," *Discrete Comput. Geom.*, 7:255-280, 1992.

[141] P. Gritzmann and V. Klee, "Computational Complexity of Inner and Outer j-Radii of Polytopes in Finite-Dimenional Normed Spaces," *Math. Program.*, 59:162-213, 1993.

[142] P. Gritzmann and V. Klee, "On the Complexity of Some basic Problems in Computational Convexity: I. Containment Problems," *Discrete Math.*, 136:129-174, 1994.

[143] P. Gritzmann and V. Klee. Mathematical programming and convex geometry. In P. Gruber and J. Wills, editors, *Hand Book of Convex Geometry*, North Holland, Amsterdam, 1993, page 627–674.

[144] M. Grötschel, L. Lovász and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, Berlin, 1988.

[145] B. Grünbaum. *Convex Polytopes*. John Wiley & Sons, New York, NY, 1967.

[146] O. Güler. Barrier functions in interior-point methods. *Mathematics of Operations Research*, 21(4):860–885, 1996.

[147] O. Güler. Existence of interior points and interior paths in nonlinear monotone complementarity problems. *Mathematics of Operations Research*, 18(1):128–147, 1993.

[148] O. Güler and L. Tuncel. Characterizations of the barrier parameter of homogeneous convex cones. Research Report CORR 95-14, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 1995. To appear in *Math. Programming*.

[149] O. Güler and Y. Ye. Convergence behavior of interior point algorithms. *Math. Programming*, 60:215–228, 1993.

[150] D. Guo, L. Rasmussen, S. Sun, and T. Lim. A matrix-algebraic approach to linear parallel interference cancellation in CDMA. *IEEE Transactions on Communications*, 48(1):152–161, Jan. 2000.

[151] E. Halperin and U. Zwick, "Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs," Proceedings of *7th IPCO* (1999), 202-217.

[152] E. Halperin and U. Zwick. A unified framework for obtaining improved approximation algorithms for maximum graph bisection problems. In *IPCO*, Lecture Notes in Computer Science. Springer, Berlin, 2001.

[153] Q. Han, Y. Ye, and J. Zhang. An improved rounding method and semidefinite programming relaxation for graph partition. *Math. Programming*, 92:509–535, 2002.

[154] S. Har-Peled and K.R. Varadarajan, "Approximate Shape Fitting via Linearization," In *Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci.*, 66-73, 2001.

[155] S. Har-Peled and K. Varadarajan, "Projective Clustering in High Dimensions Using Core-sets," In *Proc. ACM Symp. Comput. Geom.*, 2002.

[156] S. Har-Peled and K. Varadarajan, "High-Dimensional Shap Fitting in Linear Time," In *Proc. ACM Symp. Comput. Geom.*, 2003.

[157] J. Hartmanis and R. E. Stearns, On the computational complexity of algorithms. *Trans. A.M.S*, 117:285–306, 1965.

[158] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal of Optimization*, 10:673–696, 2000.

[159] C. Helmberg, F. Rendl, R. J. Vanderbei and H. Wolkowicz. An interior point method for semidefinite programming. *SIAM J. Optimization*, 6:342–361, 1996.

[160] B. Hendrickson. Conditions for Unique Graph Realizations. *SIAM J. Comput.* 21(1):65–84, 1992.

[161] B. Hendrickson. The Molecule Problem: Exploiting Structure in Global Optimization. *SIAM J. Opt.* 5(4):835–857, 1995.

[162] D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming, Algorithms and Complexity.* Ph.D. Thesis, Faculty of Mathematics and Informatics, TU Delft, NL–2628 BL Delft, The Netherlands, 1992.

[163] J. Hightower and G. Boriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8) (2001) 57-66.

[164] R. Horst, P. M. Pardalos, and N. V. Thoai. *Introduction to Global Optimization.* Kluwer Academic Publishers, Boston, 1995.

[165] A. Howard, M. J. Mataric, and G. S. Sukhatme. Relaxation on a mesh: a formalism for generalized localization. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS01)* (2001) 1055-1060.

[166] L. Huaizhong and M. Fu. A linear matrix inequality approach to robust H_∞ filtering. *IEEE Transactions on Signal Processing*, 45(9):2338–2350, Sept. 1997.

[167] P. Huard. Resolution of mathematical programming with nonlinear constraints by the method of centers. In J. Abadie, editor, *Nonlinear Programming*, pages 207–219. North Holland, Amsterdam, The Netherlands, 1967.

[168] P. Huard and B. T. Liêu. La méthode des centres dans un espace topologique. *Numeriche Methematik*, 8:56-67, 1966.

[169] B. Jackson, T. Jordán. Connected Rigidity Matroids and Unique Realizations of Graphs. Preprint, 2003.

[170] R. Jagannathan and S. Schaible. Duality in generalized fractional programming via Farkas' lemma. *Journal of Optimization Theory and Applications*, 41:417–424, 1983.

[171] B. Jansen. *Interior Point Techniques in Optimization.* Ph.D. Thesis, Faculty of Mathematics and Informatics, TU Delft, NL–2628 BL Delft, The Netherlands, 1995.

[172] F. Jarre. An interior–point method for minimizing the largest eigenvalue of a linear combination of matrices. *SIAM J. Control and Optimization*, 31:1360–1377, 1993.

[173] F. Jarre. Interior–point methods for convex programming. *Applied Mathematics & Optimization*, 26:287–311, 1992.

[174] F. Jarre, M. Kocvara, and J. Zowe. Optimal truss design by interior point methods. *SIAM J. Optim.*, 8(4):1084–1107, 1998.

[175] F. John. Extremum problems with inequalities as subsidiary conditions. In *Sduties and Essays, presented to R. Courant on his 60th Birthday, January 8, 1948*, Interscience, New York, 1948, page 187-204.

[176] N. Johnson and S. Kotz, *Distributions in Statistics: Continuous Multivariate Distributions*, John Wiley & Sons, 1972.

[177] W.B. Johnson and J. Lindenstrauss, "Extensions of Lipshitz Mapping into Hilbert Space," *Comtemporary Mathematics*, 26:189-206, 1984.

[178] M. J Kaiser, T. L. Morin and T. B. Trafalis. Centers and invariant points of convex bodies. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, Amer. Math. Soc. and Ass. Comput. Mach., Providence, RI, 1991.

[179] D. Karger, R. Motwani and M. Sudan, "Approximate graph coloring by semidefinite programming," in *35th FOCS* (1994), 2-13.

[180] S. E. Karisch, F. Rendl, and J. Clausen, "Solving graph bisection problems with semidefinite programming," Technical Report DIKU-TR-97/9, Department of Computer Science, University of Copenhagen, July 1, 1997.

[181] N. K. Karmarkar. A new polynomial–time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[182] R. M. Karp. Reducibility among combinatorial problems. R. E. Miller and J. W. Thatcher, eds., *Complexity of Computer Computations*, Plenum Press, New York, 1972, pages 85–103.

[183] L. G. Khachiyan. A polynomial algorithm for linear programming. *Doklady Akad. Nauk USSR*, 244:1093–1096, 1979. Translated in *Soviet Math. Doklady*, 20:191–194, 1979.

[184] L. G. Khachiyan and M. J. Todd. On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Math. Programming*, 61:137–160, 1993.

[185] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Segmentation problems. In *Proceedings of the 30th Symposium on Theory of Computation*, pages 473–482, 1998.

[186] V. Klee and G. J. Minty. How good is the simplex method. In O. Shisha, editor, *Inequalities III*, Academic Press, New York, NY, 1972.

[187] E. de Klerk, C. Roos and T. Terlaky. Initialization in semidefinite programming via a self–dual skew–symmetric embedding. Report 96–10, Faculty of Technical Mathematics and Computer Science, Delft University of Technology, Delft, 1996. To appear in *Operations Research Letters*.

[188] M. Kocvara, J. Zowe, and A. Nemirovski. Cascading−an approach to robust material optimization. *Computers and Structures*, 76:431–442, 2000.

[189] M. Kojima, N. Megiddo, T. Noma and A. Yoshise. *A unified approach to interior point algorithms for linear complementarity problems*. Lecture Notes in Computer Science 538, Springer Verlag, New York, 1991.

[190] M. Kojima, S. Mizuno, and A. Yoshise. A polynomial–time algorithm for a class of linear complementarity problems. *Math. Programming*, 44:1–26, 1989.

[191] M. Kojima, S. Mizuno, and A. Yoshise. An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems. *Math. Programming*, 50:331–342, 1991.

[192] M. Kojima, S. Shindoh, and S. Hara Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices. *SIAM J. Optimization*, 7:86–125, 1997.

[193] M. Kojima, N. Megiddo, and Y. Ye. An interior point potential reduction algorithm for the linear complementarity problem. *Math. Programming*, 54:267–279, 1992.

[194] K. O. Kortanek, F. Potra, and Y. Ye. On some efficient interior point methods for nonlinear convex programming. *Linear Algebra and Its Applications*, 152:169–189, 1991.

[195] G. Kortsarz and D. Peleg, "On choosing a dense subgraph," in *34th FOCS*, (1993) 692-701.

[196] E. Kranich. Interior–point methods bibliography. *SIAG/OPT Views–and–News, A Forum for the SIAM Activity Group on Optimization*, 1:11, 1992.

[197] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In J. Neyman, editor, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, University of California Press, Berkeley and Los Angeles, CA, 1961.

[198] J. C. Lagarias. The nonlinear geometry of linear programming, Part III : Projective Legendre transform coordinates and Hilbert geometry. *Transactions of the American Mathematical Society*, 320:193–225, 1990.

[199] J. C. Lagarias. A collinear scaling interpretation of Karmarkar's linear programming algorithm. *SIAM J. Optimization*, 3:630–636, 1993.

[200] J. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optimization*, 11:796–817, 2001.

[201] J. Lasserre. Semidefinite programming vs. lp relaxation for polynomial programming. *Mathematics of Operations Research*, 27(2):347–360, May 2002.

[202] M. Laurent. Matrix Completion Problems. *The Encyclopedia of Optimization* 3:221–229, 2001.

[203] A. Levin. On an algorithm for the minimization of convex functions. *Soviet Math. Doklady*, 6:286-290, 1965.

[204] C. Lin and R. Saigal, "On solving large scale semidefinite programming problems - a case study of quadratic assignment problem," Technical Report, Dept. of Ind. and Oper. Eng., University of Michigan, Ann Arbor, MI, 1997.

[205] N. Linial, E. London, Yu. Rabinovich. The Geometry of Graphs and Some of Its Algorithmic Applications. *Combinatorica* 15(2):215–245, 1995.

[206] M. S. Lobo, L. Vandenberghe, and S. Boyd. Second-order cone programming. Manuscript, Information Systems Laboratory, Dept. of Electrical Engineering, Stanford University, Stanford, CA, 1997.

[207] L. Lovasz. *An Algorithmic Theory of Numbers, Graphs and Convexity*, volume 50 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1986.

[208] L. Lovász, "On the Shannon Capacity of a graph," *IEEE Transactions of Information Theory* IT-25(1):1-7, Jan. 1979.

[209] L. Lovász and A. Shrijver. Cones of matrices and setfunctions, and $0-1$ optimization. *SIAM J. Optimization*, 1:166-190, 1990.

[210] W. Lu. A unified approach for the design of 2-D digital filters via semidefinite programming. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 49(6):814–826, June 2002.

[211] D. G. Luenberger. *Linear and Nonlinear Programming*, 2nd Edition. Addison–Wesley, Menlo Park, CA, 1984.

[212] Z. Q. Luo, J. Sturm and S. Zhang. Superlinear convergence of a symmetric primal-dual path following algorithm for semidefinite programming. *SIAM J. Optimization* 8(1):59-81, 1998.

[213] Z. Q. Luo, J. Sturm and S. Zhang. Duality and self-duality for conic convex programming. Report 9620/A, Econometric Institute, Erasmus University Rotterdam, 1996.

[214] I. J. Lustig, R. E. Marsten, and D. F. Shanno. Computational experience with a globally convergent primal–dual predictor–corrector algorithm for linear programming. *Math. Programming*, 66:123–135, 1994.

[215] I. J. Lustig, R. E. Marsten, and D. F. Shanno. Interior point methods : Computational state of the art. *ORSA Journal on Computing*, 6:1–14, 1994.

[216] S. Mahajan and H. Ramesh, "Derandomizing semidefinite programming based approximation algorithms," in *36th FOCS*, pp. 162-169, 1995.

[217] M. Mahmoud and A. Boujarwah. Robust H_∞ filtering for a class of linear parameter-varying systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(9):1131–1138, Sept. 2001.

[218] L. McLinden. The analogue of Moreau's proximation theorem, with applications to the nonlinear complementarity problem. *Pacific Journal of Mathematics*, 88:101–161, 1980.

[219] N. Megiddo. Pathways to the optimal set in linear programming. In N. Megiddo, editor, *Progress in Mathematical Programming : Interior Point and Related Methods*, pages 131–158. Springer Verlag, New York, 1989. Identical version in : *Proceedings of the 6th Mathematical Programming Symposium of Japan, Nagoya, Japan*, 1986, pages 1–35.

[220] S. Mehrotra. Quadratic convergence in a primal–dual method. *Mathematics of Operations Research*, 18:741-751, 1993.

[221] S. Mehrotra. On the implementation of a primal–dual interior point method. *SIAM J. Optimization*, 2(4):575–601, 1992.

[222] G. Millerioux and J. Daafouz. Global chaos synchronization and robust filtering in noisy context. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(10):1170–1176, Oct. 2001.

[223] Hans D. Mittelmann. Independent Benchmark Results http://plato.la.asu.edu/, November 2000.

[224] S. Mizuno, M. J. Todd, and Y. Ye. On adaptive step primal–dual interior–point algorithms for linear programming. *Mathematics of Operations Research*, 18:964–981, 1993.

[225] S. Mizuno, M. J. Todd, and Y. Ye. A surface of analytic centers and primal–dual infeasible–interior–point algorithms for linear programming. *Mathematics of Operations Research*, 20:135–162, 1995.

[226] R. D. C. Monteiro and I. Adler. Interior path following primal–dual algorithms : Part I : Linear programming. *Math. Programming*, 44:27–41, 1989.

[227] R. D. C. Monteiro and I. Adler. Interior path following primal–dual algorithms : Part II : Convex quadratic programming. *Math. Programming*, 44:43–66, 1989.

[228] R. D. C. Monteiro and J-S. Pang. On two interior-point mappings for nonlinear semidefinite complementarity problems. Working Paper, John Hopkins University, 1996.

[229] R. D. C. Monteiro and P. R. Zanjácomo, "Implementation of primal-dual methods for semidefinite programming based on Monteiro and Tsuchiya directions and their variants," Technical Report, School of Ind. and Systems Engineering, Georgia Institute of Technology, Atlanta, 1997.

[230] R. D. C. Monteiro and Y. Zhang. A unified analysis for a class of path-following primal-dual interior-point algorithms for semidefinite programming. School of ISyE, Georgia Tech, GA 30332, 1995.

[231] J. J. Moré. The Levenberg-Marquardt algorithm: implementation and theory. In G. A. Watson, editor: *Numerical Analysis* Springer-Verlag, New York, 1977.

[232] J. Moré, Z. Wu. Global Continuation for Distance Geometry Problems. *SIAM J. Opt.* 7:814–836, 1997.

[233] K. G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann, Verlag, Berlin, 1988.

[234] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Math. Programming*, 39:117-129, 1987.

[235] M. Muramatsu. Affine scaling algorithm fails for semidefinite programming. Research report No.16, Department of Mechanical Engineering, Sophia University, Japan, July 1996.

[236] M. Muramatsu and T. Tsuchiya. An affine scaling method with an infeasible starting point: convergence analysis under nondegeneracy assumption. *Annals of Operations Research*, 62:325–356, 1996.

[237] M. Muramatsu and R. Vanderbei, "Primal-dual affine scaling algorithms fail for semidefinite programming," Technical Report, SOR, Princeton University, Princeton, NJ, 1997.

[238] S. G. Nash and A. Sofer. A barrier method for large–scale constrained optimization. *ORSA Journal on Computing*, 5:40–53, 1993.

[239] D. J. Newman. Location of the maximum on unimodal surfaces. *J. Assoc. Comput. Math.*, 12:395-398, 1965.

[240] A. Nemirovski, C. Roos and T. Terlaky, "On Maximization of Quadratic Forms Over Intersection of Ellipsoids with Common Center," *Math. Program.*, 86:463-473, 1999.

[241] A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley and Sons, Chichester, 1983. Translated by E. R. Dawson from *Slozhnost' Zadach i Effektivnost' Metodov Optimizatsii*, 1979, Glavnaya redaktsiya fiziko-matematicheskoi literatury, Izdatelstva "Nauka".

[242] Y. E. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software* 9 (1998) 141-160.

[243] Yu. E. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms*. SIAM Publications. SIAM, Philadelphia, 1993.

[244] Yu. E. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research*, 22(1):1–42, 1997.

[245] Yu. E. Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. Technical Report No. 1125, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1995, to appear in *SIAM J. Optimization*.

[246] Yu. Nesterov, M. J. Todd and Y. Ye. Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems. Technical Report No. 1156, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1996.

[247] D. Niculescu and B. Nath. Ad-hoc positioning system. In *IEEE GlobeCom*, Nov. 2001.

[248] R. Palhares, C. de Souza, and P. Dias Peres. Robust H_∞ filtering for uncertain discrete-time state-delayed systems. *IEEE Transactions on Signal Processing*, 48(8):1696–1703, Aug. 2001.

[249] R. Palhares and P. Peres. LMI approach to the mixed H_2/H_∞ filtering design for discrete-time uncertain systems. *IEEE Transactions on Aerospace and Electronic Systems*, 37(1):292–296, Jan. 2001.

[250] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice–Hall, Englewood Cliffs, NJ, 1982.

[251] P. M. Pardalos and J. B. Rosen. *Constrained Global Optimization: Algorithms and Applications*. Springer-Verlag, Lecture Notes in Computer Sciences 268, 1987.

[252] J. Park, H. Cho, and D. Park. Design of GBSB neural associative memories using semidefinite programming. *IEEE Transactions on Neural Networks*, 10(4):946–950, July 1999

[253] P. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*. To appear.

[254] G. Pataki. On the Rank of Extreme Matrices in Semidefinite Programs and the Multiplicity of Optimal Eigenvalues. *Mathematics of Operations Research*, 23(2):339-358, 1998.

[255] E. Petrank, "The hardness of approximation: gap location," *Computational Complexity*, 4 (1994), 133-157.

[256] S. Polijak, F. Rendl and H. Wolkowicz. A recipe for semidefinite relaxation for 0-1 quadratic programming. *Journal of Global Optimization*, 7:51-73, 1995.

[257] R. Polyak. Modified barrier functions (theory and methods). *Math. Programming*, 54:177–222, 1992.

[258] F. A. Potra and Y. Ye. Interior point methods for nonlinear complementarity problems. *Journal of Optimization Theory and Application*, 88:617–642, 1996.

[259] F. Potra and R. Sheng. Homogeneous interior–point algorithms for semidefinite programming. Reports On Computational Mathematics, No. 82/1995, Department of Mathematics, The University of Iowa, 1995.

[260] E. Rains. A semidefinite program for distillable entanglement. *IEEE Transactions on Information Theory*, 47(7):2921–2933, November 2001.

[261] M. Ramana. An exact duality theory for semidefinite programming and its complexity implications. *Math. Programming*, 77:129–162, 1997.

[262] M. Ramana, L. Tuncel and H. Wolkowicz. Strong duality for semidefinite programming. CORR 95-12, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 1995. To appear in *SIAM J. Optimization*.

[263] L. Rasmussen, T. Lim, and A. Johansson. A matrix-algebraic approach to successive interference cancellation in CDMA. *IEEE Transactions on Communications*, 48(1):145–151, Jan. 2000.

[264] F. Rendl and H. Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Math. Programming*, 77:273–300, 1997.

[265] J. Renegar. A polynomial–time algorithm, based on Newton's method, for linear programming. *Math. Programming*, 40:59–93, 1988.

[266] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

[267] C. Roos, T. Terlaky and J.-Ph. Vial. *Theory and Algorithms for Linear Optimization: An Interior Point Approach.* John Wiley & Sons, Chichester, 1997.

[268] M. Saadat and T. Chen. Design of nonuniform multirate filter banks by semidefinite programming. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(11):1311–1314, November 2000.

[269] R. Saigal and C. J. Lin. An infeasible start predictor corrector method for semi-definite linear programming. Tech. Report, Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, Dec 1995.

[270] C. Savarese, J. Rabay, K. Langendoen. Robust Positioning Algorithms for Distributed Ad–Hoc Wireless Sensor Networks. *USENIX Annual Technical Conference*, 2002.

[271] A. Savvides, C.–C. Han, M. B. Srivastava. Dynamic Fine–Grained Localization in Ad–Hoc Networks of Sensors. *Proc. 7th MOBICOM*, 166–179, 2001.

[272] A. Savvides, H. Park, M. B. Srivastava. The Bits and Flops of the n–hop Multilateration Primitive for Node Localization Problems. *Proc. 1st WSNA*, 112–121, 2002.

[273] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, New York, 1986.

[274] U. Shaked, L. Xie, and Y. Soh. New approaches to robust minimum variance filter design. *IEEE Transactions on Signal Processing*, 49(11):2620–2629, November 2001.

[275] Y. Shang, W. Ruml, Y. Zhang, M. P. J. Fromherz. Localization from Mere Connectivity. *Proc. 4th MOBIHOC*, 201–212, 2003.

[276] W. F. Sheppard, "On the calculation of the double integral expressing normal correlation," *Transactions of the Cambridge Philosophical Society* 19 (1900) 23-66.

[277] H. D. Sherali and W. P. Adams, "Computational advances using the reformulation-linearization technique (rlt) to solve discrete and continuous nonconvex problems. *Optima*, 49:1-6, 1996.

[278] Masayuki Shida, Susumu Shindoh and Masakazu Kojima. Existence of search directions in interior-point algorithms for the SDP and the monotone SDLCP. Research Report B-310, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152 Japan, 1996. To appear in *SIAM J. Optimization.*

[279] N. Z. Shor, "Quadratic optimization problems," *Soviet Journal of Computer and Systems Sciences*, 25:1-11, 1987. Originally published in Tekhnicheskaya Kibernetika, No. 1, 1987, pp. 128-139.

[280] M. Skutella, "Semidefinite relaxations for parallel machine scheduling," in *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, pp. 472-481, 1998.

[281] M. Skutella, "Convex quadratic and semidefinite programming relaxations in scheduling," *Journal of the ACM*, 48(2), 206-242, 2001.

[282] So, AM., Ye, Y., and Zhang, J., "A unified theorem on SDP rank reduction," *Mathematics of Operations Research.* **33**, 2008, pages 910-920.

[283] G. Sonnevend. An "analytic center" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In A. Prekopa, J. Szelezsan, and B. Strazicky, editors, *System Modelling and Optimization : Proceedings of the 12th IFIP–Conference held in Budapest, Hungary, September 1985*, volume 84 of *Lecture Notes in Control and Information Sciences*, pages 866–876. Springer Verlag, Berlin, Germany, 1986.

[284] D. C. Sorenson. Newton's method with a model trust region modification. *SIAM J. Numer. Anal.*, 19:409-426, 1982.

[285] A. Srivastav and K. Wolf, "Finding dense subgraphs with semidefinite programming," in *Approximation Algorithms for Combinatorial Optimization*, K. Jansen and J. Rolim (Eds.) (1998), 181-191.

[286] P. Stoica, T. McKelvey, and J. Mari. Ma estimation in polynomial time. *IEEE Transactions on Signal Processing*, 48(7):1999–2012, July 2000.

[287] J. F. Sturm. SeDumi, Let SeDuMi seduce you. http://fewcal.kub.nl/sturm/software/sedumi.html, October 2001.

[288] J. F. Sturm and S. Zhang. Symmetric primal-dual path following algorithms for semidefinite programming. Report 9554/A, Econometric Institute, Erasmus University, Rotterdam, 1995

[289] J.F. Sturm and S. Zhang. On cones of nonnegative quadratic functions. SEEM2001-01, Chinese University of Hong Kong, March 2001.

[290] K. Tanabe. Complementarity–enforced centered Newton method for mathematical programming. In K. Tone, editor, *New Methods for Linear Programming*, pages 118–144, The Institute of Statistical Mathematics, 4–6–7 Minami Azabu, Minatoku, Tokyo 106, Japan, 1987.

[291] K. Tanabe. Centered Newton method for mathematical programming. In M. Iri and K. Yajima, editors, *System Modeling and Optimization*, Proceedings of the 13th IFIP Conference, pages 197–206, Springer-Verlag, Tokyo, Japan, 1988.

[292] R. A. Tapia. Current research in numerical optimization. *SIAM News*, 20:10–11, March 1987.

[293] S. P. Tarasov, L. G. Khachiyan, and I. I. Érlikh. The method of inscribed ellipsoids. *Soviet Mathematics Doklady*, 37:226–230, 1988.

[294] H. Tan and L. K. Rasmussen. The application of semidefinite programming for detection in CDMA. *IEEE Journal on Selected Areas in Communications*, 19(8):1442–1449, August 2001.

[295] Z. Tan, Y. Soh, and L. Xie. Envelope-constrained H_∞ filter design: an LMI optimization approach. *IEEE Transactions on Signal Processing*, 48(10):2960–2963, Oct. 2000.

[296] Z. Tan, Y. Soh, and L. Xie. Envelope-constrained H_∞ FIR filter design. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(1):79–82, Jan. 2000.

[297] T. Terlaky, editor. *Interior-Point Methods in Mathematical Programming* Kluwer Academic Publisher, Boston, 1996.

[298] M. J. Todd, Improved bounds and containing ellipsoids in Karmarkar's linear programming algorithm. *Mathematics of Operations Research*, 13:650-659, 1988.

[299] M. J. Todd. Potential-reduction methods in mathematical programming. *Math. Programming*, 76:3–45, 1996.

[300] M. J. Todd, "A study of search directions in primal-dual interior-point methods for semidefinite programming," *Optimization Methods and Software* 11&12 (1999) 1-46.

[301] M. J. Todd and B. P. Burrell. An extension of Karmarkar's algorithm for linear programming using dual variables. *Algorithmica*, 1(4):409–424, 1986.

[302] M. J. Todd and Y. Ye. A centered projective algorithm for linear programming. *Mathematics of Operations Research*, 15:508–529, 1990.

[303] K. C. Toh. A note on the calculation of steplengths in interior point methods for semidefinite programming. To appear *Computational Optimization and Applications*, 2001.

[304] K. C. Toh, M. J. Todd, and R. H. Tütüncü. SDPT3 – A MATLAB software package for semidefinite programming, version 1.3 *Optimization Software and Methods*, 11:545-581, 1999.

[305] K. C. Toh and M. Kojima. Solving some large scale semidefinite programs via the conjugate residual method. Working paper, August 2000.

[306] Y. L. Tong, *The Multivariate Normal Distribution*, Springer-Verlag, New York, 1990.

[307] J. F. Traub, G. W. Wasilkowski and H. Wozniakowski. *Information-Based Complexity.* Academic Press, San Diego, CA, 1988.

[308] P. Tseng. Search directions and convergence analysis of some infeasible path-following methods for the monotone semi-definite LCP. Report, Department of Mathematics, University of Washington, Seattle, Washington 98195, 1996.

[309] P. Tseng. Approximating QP. Report, Department of Mathematics, University of Washington, Seattle, Washington, 2003.

[310] C. Tseng and B. Chen. H_∞ fuzzy estimation for a class of nonlinear discrete-time dynamic systems. *IEEE Transactions on Signal Processing*, 49(11):2605–2619, Nov. 2001.

[311] H. Tuan, P. Apkarian, and T. Nguyen. Robust and reduced-order filtering: new LMI-based characterizations and methods. *IEEE Transactions on Signal Processing*, 49(12):2975–2984, Dec. 2001.

[312] H. Tuan, P. Apkarian, T. Nguyen, and T. Narikiyo. Robust mixed H_2/H_∞ filtering of 2-D systems. *IEEE Transactions on Signal Processing*, 50(7):1759–1771, July 2002.

[313] A. W. Tucker. Dual systems of homogeneous linear relations, in H. W. Kuhn and A. W. Tucker, Editors, *Linear Inequalities and Related Systems*, page 3–, Princeton University Press, Princeton, NJ, 1956.

[314] P. M. Vaidya. A new algorithm for minimizing a convex function over convex sets. *Math. Programming*, 73:291–341, 1996.

[315] P. M. Vaidya and D. S. Atkinson. A technique for bounding the number of iterations in path following algorithms. In P. Pardalos, editor, *Complexity in Numerical Optimization* page 462–489, World Scientific, New Jersey, 1993.

[316] L. Vandenberghe and S. Boyd. A primal–dual potential reduction method for problems involving matrix inequalities. *Math. Programming*, 69:205–236, 1995.

[317] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[318] L. Vandenberghe, S. Boyd, and S.P. Wu. Determinant maximization with linear matrix inequality constraints. Technical Report, Information Systems Laboratory, Stanford University, CA, 1996.

[319] R. J. Vanderbei. LOQO: An Interior Point Code for Quadratic Programming. Program in Statistics & Operations Research, Princeton University, NJ, 1995.

[320] K. Varadarajan, S. Venkatesh and J. Zhang, "On Approximating the Radii of Point Sets in High Dimensions," In *Proc. 43rd Annu. IEEE Sympos. Found. Comput. Sci.*, 2002.

[321] S. A. Vavasis. *Nonlinear Optimization: Complexity Issues*. Oxford Science, New York, NY, 1991.

[322] F. Wang and V. Balakrishnan. Robust Kalman filters for linear time-varying systems with stochastic parametric uncertainties. *IEEE Transactions on Signal Processing*, 50(4):803–813, April 2002.

[323] S. Wang, L. Xie, and C. Zhang. H_2 optimal inverse of periodic FIR digital filters. *IEEE Transactions on Signal Processing*, 48(9):2696–2700, Sept. 2000.

[324] T. Wang, R. D. C. Monteiro, and J.-S. Pang. An interior point potential reduction method for constrained equations. *Math. Programming*, 74(2):159–196, 1996.

[325] H. Wolkowicz and Q. Zhao, "Semidefinite programming for the graph partitioning problem," manuscript, 1996, to appear *Discrete Applied Math.*.

[326] M. H. Wright. A brief history of linear programming. *SIAM News*, 18(3):4, November 1985.

[327] S. J. Wright. *Primal-Dual Interior-Point Methods* SIAM, Philadelphia, 1996.

[328] C. Xu, Y. Ye and J. Zhang. Approximate the 2-Catalog Segmentation Problem Using Semidefinite Programming Relaxations. *Optimization Methods and Software*, 18:(6):705 - 719, 2003

[329] G. Xue and Y. Ye. Efficient algorithms for minimizing a sum of Euclidean norms with applications. Working Paper, Department of Management Sciences, The University of Iowa, 1995. To appear in *SIAM J. Optimization*.

[330] F. Yang and Y. Hung. Robust mixed H_2/H_∞ filtering with regional pole assignment for uncertain discrete-time systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 49(8):1236–1241, Aug. 2002.

[331] M. Yannakakis. Computing the minimum fill–in is NP–complete. *SIAM J. Algebraic Discrete Methods*, pages 77–79, 1981.

[332] Y. Ye. *Interior Point Algorithms : Theory and Analysis*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, New York, 1997.

[333] Y. Ye. Karmarkar's algorithm and the ellipsoid method. *Operations Research Letters*, 6:177–182, 1987.

[334] Y. Ye. An $O(n^3L)$ potential reduction algorithm for linear programming. *Math. Programming*, 50:239–258, 1991.

[335] Y. Ye. Combining binary search and Newton's method to compute real roots for a class of real functions. *Journal of Complexity*, 10:271–280, 1994.

[336] Y. Ye. Approximating quadratic programming with bound and quadratic constraints. *Mathematical Programming 84* (1999) 219-226.

[337] Y. Ye. A .699 approximation algorithm for Max-Bisection. *Mathematical Programming* 90:1 (2001) 101-111

[338] Y. Ye and M. Kojima. Recovering optimal dual solutions in Karmarkar's polynomial algorithm for linear programming. *Math. Programming*, 39:305–317, 1987.

[339] Y. Ye, O. Güler, R. A. Tapia, and Y. Zhang. A quadratically convergent $O(\sqrt{n}L)$–iteration algorithm for linear programming. *Math. Programming*, 59:151–162, 1993.

[340] Y. Ye, M. J. Todd and S. Mizuno. An $O(\sqrt{n}L)$–iteration homogeneous and self–dual linear programming algorithm. *Mathematics of Operations Research*, 19:53–67, 1994.

[341] Y. Ye and S. Zhang. New results on quadratic minimization. *SIAM J. Optim.*, 14(1):245–267, 2003.

[342] Y. Ye and J. Zhang. An improved algorithm for approximating the radii of point sets. APPROX 2003, August 2003.

[343] Q. Zhao, "Semidefinite Programming for Assignment and Partitioning Problems" PhD thesis, University of Waterloo, 1996.

[344] Q. Zhao, S. Karisch, F. Rendl, and H. Wolkowicz, "Semidefinite programming relaxations for the quadratic assignment problems," *J. Combinatorial Optimization*, 2.1, 1998, CORR 95-27.

[345] H. Zhou, L. Xie, and C. Zhang. A direct approach to H_2 optimal deconvolution of periodic digital channels. *IEEE Transactions on Signal Processing*, 50(7):1685–1698, July 2002.

[346] J. Zowe, M. Kocvara, and M. Bendsoe. Free material optimization via mathematical programming. *Mathematical Programming*, 9:445–466, 1997.

[347] U. Zwick, "Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint," In *9th SODA* (1998) 201-210.

[348] U. Zwick. Outward rotationvs: a tool for rounding solutions of semidefinite programming relaxations, with applications to max cut and other problems. In *Proceedings of the 31th Symposium on Theory of Computation*, pages 679–687, 1999.

Index