Optimization Over Symmetric Cones

by

Madhu V. Nayakkankuppam

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science
Graduate School of Arts and Science
New York University
September 1999

Dissertation Advisor: Professor Michael L. Overton
Department of Computer Science
New York University

Courant Institute of Mathematical Sciences, New York. Last revised: April 3, 2000.

For Amma, Appa, Doonda, and in fond memory of Patti,
who all took so little and gave so much

Acknowledgment

My first and most grateful thanks go to my close collaborators **Michael Overton** (advisor), **Jean–Pierre Haeberly**, **François Oustry** and **Bas Braams**, without whose help this thesis could not have been finished (or even started).

Any attempt to adequately thank Michael Overton and Jean–Pierre Haeberly, who taught me all I know about semidefinite programming and interior–point methods, can only be an exercise in futility. So much of the research described in this thesis has been a direct result of their ideas, insights, experience and efforts. Michael pointed me to the Kantorovič theory, contributed valuable numerical insights towards algorithm and code development, urged me to work on the quantum chemistry project, and is responsible for my participation in several rewarding collaborations. Jean-Pierre came up with many of the ideas we used in our algorithms for SDP, and did a great deal of the code development with cheerful enthusiasm. He was very generous to me, and was always a reliable source of sound mathematical judgement and constructive criticisms.

François Oustry taught me all about bundle methods. During his postdoc year in New York (which couldn't have been more timely), we had many exciting discussions in the "talking rooms" of the Courant library on topics within and oustide of this thesis. He generously shared, and continues to share, many of his valuable insights and original ideas with me with a true spirit of camaraderie.

Bas Braams is responsible for starting the quantum chemistry project, which he spearheads with great enthusiasm. I would like to thank him for his many patient tutorials on the subject, and for keeping the excitement going by formulating (on an almost weekly basis) new numerical approaches to tackle the *representability problem*.

I would also like to thank Shidong Jiang for his help in the early stages of the quantum chemistry project, for his tutorials explaining the basics, and for use of his code.

Special thanks to Olof Widlund for his interest in my case, and for his support and help on numerous occassions.

Thanks are due to Bas Braams, Jean–Pierre Haeberly, Davi Geiger, Michael Overton, Olof Widlund for serving on my thesis committee, and for providing useful feedback on the thesis. I would especially like to thank Bas for the improvements he suggested to the chapter on electronic structure calculations; Jean–Pierre for many valuable comments (including some quite subtle ones) on the whole thesis; François Oustry for comments on the chapter on bundle methods; and most of all, Michael who meticulously read several earlier versions and helped *vastly* improve the quality of the exposition.

I also gratefully acknowledge helpful early conversations with Farid Alizadeh on semidefinite programming, and more recently, with Franz Rendl and Christoph Helmberg on their spectral bundle method for semidefinite programming.

Thanks to Alan Siegel, who first suggested that I work with Michael Overton, and our (ruthlessly efficient) graduate coordinator Anina Karmen, who knew me before anyone else in the department did, and has helped me in *so many* ways since then.

I gratefully acknowledge financial support in the form of teaching assistantships (two semesters and one summer) from the department, and research assistantships (eight semesters and two summers) from Michael Overton's research grants from the National Science Foundation (CCR–9731777) and the Department of Energy (DE–FG02–98ER25352), which made it possible for me to spend five wonderful years in New York, and work with such a terrific group of people.

On a more personal note, I would like to thank Michael Overton for being a great friend, an excellent guide, and fortunately for me, far too pragmatic to be a philosopher. For his constant support, his encouragement in trying times, his ample supply of generosity and genuine concern, and most of all, his valuable friendship, I owe him a big debt of gratitude.

Finally, to the members of my family who continue to silently endure this long separation ten thousand miles away, I owe an apology.

Madhu Nayakkankuppam

September 1999, New York

Abstract

We consider the problem of optimizing a linear function over the intersection of an affine space and a special class of closed, convex cones, namely the symmetric cones over the reals. This problem subsumes linear programming, convex quadratically constrained quadratic programming, and semidefinite programming as special cases. First, we derive some perturbation results for this problem class. Then, we discuss two solution methods: an *interior–point* method capable of delivering highly accurate solutions to problems of modest size, and a first order *bundle method* which provides solutions of low accuracy, but can handle much larger problems. Finally, we describe an application of semidefinite programming in electronic structure calculations, and give some numerical results on sample problems.

Contents

List of Figures

List of Tables

List of Symbols and Notations

Generally, we use blackboard symbols for fields and vector spaces. Ordinary capital letters are used for sets and linear operators. Functions are denoted by small letters. Calligraphic letters are used for cones and special classes of functions. For each entry in this index, the page reference given is for the defining instance.

Chapter 1

Conic Optimization Problems

This chapter is expository in nature, and its purpose is to formulate a certain conic optimization problem, and to develop background material and related notation that will be needed in the rest of the thesis. The results of this chapter are available in the literature on conic programming and interior–point methods, although not always in the form stated here; we give appropriate references. The development of most of the basic concepts in this chapter is based on Rockafellar and Wets [96], whereas the section on Jordan algebras is based on Faraut and Korányi [31]. At times, results cited from these two sources have been simplified to better suit the current context.

1.1 Problem Formulation

Given a real, d–dimensional Euclidean space $\mathbb{E}$ with inner product $\langle \cdot, \cdot \rangle$, a surjective linear operator $A : \mathbb{E} \to \mathbb{R}^m$, a closed, convex cone $\mathcal{K} \subseteq \mathbb{E}$, and two fixed elements $c \in \mathbb{E}$ and $b \in \mathbb{R}^m$, we consider the following *conic optimization problem* (COP):

$$\inf_{x \in \mathbb{E}} \quad \langle c, x \rangle \quad \text{s.t.} \quad Ax = b\,; \quad x \in \mathcal{K}, \tag{1.1}$$

This problem, in fact, subsumes many familiar types of convex programs.

- When $\mathbb{E} = \mathbb{S}^k$, the space of real, symmetric $k \times k$ matrices equipped with the trace inner product $\langle x, z \rangle = \mathrm{tr}(xz)$, and $\mathcal{K} = \mathbb{S}^k_+$, the cone of positive semidefinite matrices, we obtain *semidefinite programming* (SDP).

- When $\mathbb{E} = \mathbb{Q}^k \triangleq \mathbb{R} \times \mathbb{R}^k$, with the inner product $\langle x, z \rangle = 2(x_0 z_0 + x_1 z_1 + \ldots + x_k z_k)$, and $\mathcal{K} = \mathbb{Q}^k_+ \triangleq \{(x_0, \overline{x}) \in \mathbb{E} \mid x_0 \geq \|\overline{x}\|_2\}$ (the *Lorentz* or *quadratic cone*), we recover all forms of *convex quadratically constrained quadratic programming* (QCQP). (Here $\|\overline{x}\|_2$ is the usual 2–norm $\sqrt{x_1^2 + \ldots + x_k^2}$.)

- When $\mathbb{E} = \mathbb{R}^k$, the space of k–dimensional vectors with the standard inner product $\langle x, z \rangle = x_1 z_1 + \ldots + x_k z_k$, and $\mathcal{K} = \mathbb{R}^k_+$ is the cone of vectors with nonnegative entries (the *nonnegative orthant*), we get *linear programming* (LP).

We denote by $\mathrm{GL}(\mathbb{E})$, the group of general linear transformations on $\mathbb{E}$, and by $\mathrm{Aut}(\mathcal{K})$, the automorphism group of $\mathcal{K}$, *i.e.*

$$\mathrm{Aut}(\mathcal{K}) = \{T \in \mathrm{GL}(\mathbb{E}) \mid T\mathcal{K} = \mathcal{K}\}\,.$$

Let $\mathrm{int}\,\mathcal{K}$ denote the interior of $\mathcal{K}$. We say that $\mathcal{K}$ is *homogeneous* if

$$\forall\, u, v \in \mathrm{int}\,\mathcal{K}, \;\; \exists\, T \in \mathrm{Aut}(\mathcal{K}) \;\; \text{s.t.} \;\; Tu = v, \tag{1.2}$$

and *self–dual* if $\mathcal{K}$ coincides with its dual cone $\mathcal{K}^*$, *i.e.*

$$\mathcal{K} = \mathcal{K}^* \triangleq \{z \in \mathbb{E} \,:\, \langle z, x \rangle \geq 0 \;\; \forall x \in \mathbb{E}\}\,.$$

For any x and z in a self–dual cone $\mathcal{K}$, $\langle x, z \rangle \geq 0$, with the inequality being strict if at least one of x or z lies in $\operatorname{int}\mathcal{K}$, and the other is nonzero.

Among the three cones mentioned above, only the nonnegative orthant is polyhedral, but they are all homogeneous and self–dual (see, for instance, [31, Ch.I]). A cone that is both homogeneous and self–dual is said to be *symmetric*, and we will confine our attention to such cones over the reals. More details are given in Section 1.5.

The dual program of (1.1) is

$$\sup_{y \in \mathbb{R}^m} \quad \langle b, y \rangle \quad \text{s.t.} \quad A^* y + z = c; \quad z \in \mathcal{K}, \tag{1.3}$$

where A^* denotes the adjoint of A.

In the sequel, $\mathbb{E}$ (respectively $\mathcal{K}$) refers only to the three Euclidean spaces (respectively cones) mentioned above, and their direct sums. Specifically, given two positive integer vectors $N = [N_1, \ldots, N_s]$ and $n = [n_1, \ldots, n_q]$ of "block sizes", and a positive integer n_0, we deal with the Euclidean space

$$\begin{aligned}
&\mathbb{E} = \mathbb{S}^N \oplus \mathbb{Q}^n \oplus \mathbb{R}^{n_0}, \quad \text{with} \\
&\mathbb{S}^N \stackrel{\Delta}{=} \oplus_{i=1}^{s} \mathbb{S}^{N_i}, \\
&\mathbb{Q}^n \stackrel{\Delta}{=} \oplus_{i=1}^{q} \mathbb{Q}^{n_i},
\end{aligned}$$

and its associated cone

$$\begin{aligned}
&\mathcal{K} = \mathbb{S}^N_+ \oplus \mathbb{Q}^n_+ \oplus \mathbb{R}^{n_0}_+, \quad \text{with} \\
&\mathbb{S}^N_+ \stackrel{\Delta}{=} \oplus_{i=1}^{s} \mathbb{S}^{N_i}_+, \\
&\mathbb{Q}^n_+ \stackrel{\Delta}{=} \oplus_{i=1}^{q} \mathbb{Q}^{n_i}_+.
\end{aligned}$$

Any $x \in \mathbb{E}$ is then written as $x = (x_S, x_Q, x_L)$ with $x_S \in \mathbb{S}^N$ being a block diagonal matrix, $x_Q \in \mathbb{Q}^n$ being a block vector, and the vector $x_L \in \mathbb{R}^{n_0}$, an ordinary vector. We will also write $x_S(j)$ (respectively $x_Q(j)$) to denote the j^{th} "block" of x_S (respectively x_Q). We say that the COP has a semidefinite (SD), a quadratic (QC) or a linear (LP) component depending on which of $\mathbb{S}^N$, $\mathbb{Q}^n$ and $\mathbb{R}^{n_0}$ are present in $\mathbb{E}$. At times, we will specialize $\mathbb{E}$ to be one of $\mathbb{S}^N$, $\mathbb{Q}^n$ or $\mathbb{R}^{n_0}$. In this case, we will write $x(j)$, dropping the subscript (S, Q, or L) which will be evident from context.

We set $d_S \stackrel{\Delta}{=} \dim \mathbb{S}^N$, $d_Q \stackrel{\Delta}{=} \dim \mathbb{Q}^n$, and $d \stackrel{\Delta}{=} \dim \mathbb{E} = d_S + d_Q + n_0$. To facilitate the use of matrix notation, we identify $\mathbb{E}$ with $\mathbb{R}^d$ (equipped with the standard inner product) as follows. We employ the isometry $\mathbf{vec} : \mathbb{S}^N \to \mathbb{R}^{d_S}$ to identify $\mathbb{S}^N$ with $\mathbb{R}^{d_S}$.[1] Using this isometry, any $x = (x_S, x_Q, x_L) \in \mathbb{E}$ may be mapped onto $[\mathbf{vec}(x_S), \sqrt{2}x_Q, x_L] \in \mathbb{R}^d$. The operator A may be viewed as an $m \times d$ matrix. We will use the same symbol for both $x \in \mathbb{E}$ and its equivalent $x \in \mathbb{R}^d$, as the latter usage will be a vector enclosed within square brackets, and will usually appear in conjunction with matrix operations.

We conclude this section with some basic notational conventions employed in the thesis. Generally, blackboard symbols (such as $\mathbb{R}$) are used for fields and vector spaces, capital letters are usually for sets and linear operators. Elements of a vector space and functions are denoted by small letters. Matrices may be denoted by either small or capital letters. Components of matrices and vectors are indexed by subscripts, and are enclosed within square brackets. We do not use the standard visual representation of row vectors (entries on the same line) and column vectors (entries on successive lines); all vectors are column vectors, unless explicitly superscripted with a $*$. Small Greek letters are generally used for scalars and functions, whereas capital Greek letters denote sets. Calligraphic letters denote cones and special function classes, *e.g.* $\mathcal{C}^p$ for the class of p times continuously differentiable functions. For a $\mathcal{C}^p$ function h between two vector spaces, the notation $D^i_{x_{k_1} \ldots x_{k_i}} h(x)$ stands for the i^{th} order partial derivative of h with respect to the variables $x_{k_1}, \ldots, x_{k_i}$ (The superscript is omitted when $i = 1$, and the subscript is omitted when $[x_{k_1}, \ldots, x_{k_i}] = x$).

Although notation is introduced as needed, or recalled with an appropriate reference, a list of symbols and notation is available on page (xi) to serve as a quick reference guide.

[1] For instance, we can choose **vec** to be the map that stacks the columns of the lower triangular part of a matrix in $\mathbb{S}^N$ blockwise, multiplying the off–diagonal elements by $\sqrt{2}$.

1.2 Optimality Conditions

We begin by introducing the definition of the normal cone to a convex set. This is a central object in this chapter, and much of the development to follow will be based on it.

Definition 1.2.1 (Normal Cone To Convex Set)
Let $H \subseteq \mathbb{E}$ be a closed, convex set. The *normal cone* to H at $x \in F$ is defined to be

$$\mathcal{N}_H(x) = \{v \in \mathbb{E} \,:\, \langle\, v, x' - x \,\rangle \leq 0 \quad \forall x' \in H\}\,. \tag{1.4}$$

The following theorem illustrates the fundamental role of the variational geometry of feasible sets in the derivation of optimality conditions.

Theorem 1.2.1 (Basic Optimality Condition [96, Theorem 6.12])
Let $h : \mathbb{E} \to \mathbb{R}$ be a convex $\mathcal{C}^1$ function, and let $H \subseteq \mathbb{E}$ be a convex set. Then $x \in F$ is a solution to

$$\inf_{x \in H} h(x) \tag{1.5}$$

if and only if

$$0 \;\in\; Dh(x) + \mathcal{N}_H(x). \tag{1.6}$$

The feasible region in a COP is $H = H_1 \cap H_2$, where $H_1 = \{x \,:\, Ax = b\}$ and $H_2 \stackrel{\Delta}{=} \mathcal{K}$ is a closed, convex cone.

Whether a solution exists or not depends on the nature of the separation between H_1 and H_2. In general, several cases arise [70]. Let B denote the open unit ball in $\mathbb{E}$.

(i) Strongly infeasible: $(H_1 + \epsilon B) \cap H_2 = \emptyset$ for some $\epsilon > 0$. The problem is infeasible and remains so under infinitesimal perturbations.

(ii) Weakly infeasible: $H_1 \cap H_2 = \emptyset$, but $(H_1 + \epsilon B) \cap H_2 \neq \emptyset \; \forall \epsilon > 0$. An arbitrarily small perturbation to H_1 or H_2 could make the problem strongly infeasible or strongly feasible (see (iv)).

(iii) Weakly feasible: $H_1 \cap H_2 \neq \emptyset$, but $H_1 \cap \operatorname{int} H_2 = \emptyset$. An arbitrarily small perturbation in H_1 or H_2 could render the problem strongly infeasible or strongly feasible (see (iv)).

(iv) Strongly feasible: $H_1 \cap \operatorname{int} H_2 \neq \emptyset$. This property is preserved under sufficiently small perturbations to H_1 and H_2.

Depending on which of these four cases the primal and the dual problems fall in, various possibilities arise with respect to the finiteness of optimal objective values, the attainment of solutions, and the existence of a *duality gap*, *i.e.* a difference in the optimal primal and the dual objective values. See [70] for a detailed description of all these cases. The situation in (i) is uninteresting, and those in (ii), (iii) are, in some sense, ill–posed. The following standard constraint qualification precludes the difficulties in (i) – (iii) by putting us in the *strongly feasible* case.

Assumption 1.2.1 (Slater Condition)
Primal: There exists a strictly feasible point for (1.1), *i.e.* $\exists\, x \in \operatorname{int} \mathcal{K}$ s.t. $Ax = b$.
Dual: There exists a strictly feasible point for (1.3), *i.e.* $\exists\, (y, z) \in \mathbb{R}^m \times \operatorname{int} \mathcal{K}$ s.t. $A^*y + z = c$.

If both the primal and the dual problems satisfy the Slater condition, then it can be shown that the optimal primal and the dual objective values are finite and attained, and that the primal and dual solution sets are compact [79, Sec.4.2]. Hence, in fact, the "inf" in (1.1) and the "sup" in (1.3) can be replaced by "min" and "max" respectively.

The Slater condition enables the computation of the normal cone to the intersection $H = H_1 \cap H_2$ in terms of the normal cones to H_1 and to H_2. We say that two convex sets H_1 and H_2 cannot be *separated* if there exists no $(a, \alpha) \in \mathbb{E} \times \mathbb{R}$ such that $H_1 \subseteq \{x \in \mathbb{E} \,:\, \langle\, a, x \,\rangle \leq \alpha\}$ and $H_2 \subseteq \{x \in \mathbb{E} \,:\, \langle\, a, x \,\rangle \geq \alpha\}$. Clearly, when the Slater condition is satisfied (*i.e.* $H_1 \cap \operatorname{int} H_2 \neq \emptyset$), H_1 and H_2 cannot be separated.

THEOREM 1.2.2 (NORMAL CONE TO INTERSECTIONS [96, P.228])
Let $H = H_1 \cap H_2$, where H_1 and H_2 are nonempty, closed, convex subsets of $\mathbb{E}$ which cannot be separated. At any $x \in H$, we have

$$\mathcal{N}_H(x) = \mathcal{N}_{H_1}(x) + \mathcal{N}_{H_2}(x).$$

Let us now compute explicit expressions for the normal cones to the feasible sets that arise in COP, *i.e.* affine spaces and the three cones mentioned in Section 1.1.

PROPOSITION 1.2.1 (NORMAL CONE TO AFFINE SPACES)
Let $H = \{x \in \mathbb{E} : Ax = b\}$ where $A : \mathbb{E} \to \mathbb{R}^m$ is a linear operator. At any $x \in H$, we have $\mathcal{N}_H(x) = \{A^*y : y \in \mathbb{R}^m\}$.

Proof. Immediate. ■

DEFINITION 1.2.2 (COMPLEMENTARITY)
We say that $(x, z) \in \mathcal{K} \times \mathcal{K}$ are *complementary*, or satisfy *complementarity*, if $\langle x, z \rangle = 0$.

Using basic linear algebra, we can provide an alternative characterization of complementarity. In the following lemma, we introduce the Diag$(\cdot)$ operator which maps its arguments to a block diagonal matrix. (For example, if $v \in \mathbb{R}^k$, then Diag(v) denotes the $k \times k$ diagonal matrix with the entries of v on the diagonal. If v_1 and v_2 are matrices, then Diag(v_1, v_2) denotes the block diagonal matrix with v_1 and v_2 as its first and its second diagonal blocks respectively.)

LEMMA 1.2.1 (REFORMULATION OF COMPLEMENTARITY)
Let $(x, z) \in \mathcal{K} \times \mathcal{K}$.

(i) $\mathcal{K} = \mathbb{S}^N_+$**:** $\langle x, z \rangle = 0 \iff (xz + zx)/2 = 0$.

(ii) $\mathcal{K} = \mathbb{Q}^n_+$**:** $\langle x, z \rangle = 0 \iff \mathrm{Arw}(x)z = 0$, *where*

$$\mathrm{Arw}(x) = \mathrm{Diag}(\mathrm{arw}(x(1)), \ldots, \mathrm{arw}(x(q))),$$

and for $x(j) = [x_0(j), \ldots, x_{n_j}(j)]$,

$$\mathrm{arw}(x(j)) \triangleq \begin{bmatrix} x_0(j) & x_1(j) & x_2(j) & \cdots & x_{n_j}(j) \\ x_1(j) & x_0(j) & 0 & \cdots & 0 \\ x_2(j) & 0 & x_0(j) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n_j}(j) & 0 & 0 & 0 & x_0(j) \end{bmatrix}. \tag{1.7}$$

(iii) $\mathcal{K} = \mathbb{R}^{n_0}_+$**:** $\langle x, z \rangle = 0 \iff \mathrm{Diag}(x)z = 0$.

Proof.

(i) Since $0 = \mathrm{tr}(xz) = \mathrm{tr}(x^{\frac{1}{2}} z x^{\frac{1}{2}}) = \left\langle x^{\frac{1}{2}} z^{\frac{1}{2}}, x^{\frac{1}{2}} z^{\frac{1}{2}} \right\rangle$, we have $x^{\frac{1}{2}} z^{\frac{1}{2}} = 0$, which implies that xz (hence also $(xz + zx)/2$) is zero. The reverse implication follows trivially by taking the trace of both sides of the equation $(xz + zx)/2 = 0$.

(ii) It suffices to prove the statement for an arbitrary block j. Let $x(j) = [x_0, \bar{x}]$ and $z(j) = [z_0, \bar{z}]$. Since $x_0 \geq \|\bar{x}\|_2$ and $z_0 \geq \|\bar{z}\|_2$, we have

$$\|\bar{x}\|_2 \|\bar{z}\|_2 \leq x_0 z_0 = -\langle \bar{x}, \bar{z} \rangle \leq \|\bar{x}\|_2 \|\bar{z}\|_2, \tag{1.8}$$

where the last inequality is Cauchy–Schwarz. Hence,

$$x_0 z_0 = \|\bar{x}\|_2 \|\bar{z}\|_2 = -\langle \bar{x}, \bar{z} \rangle. \tag{1.9}$$

From the first equality in (1.9), if $x_0 > \|\bar{x}\|$ (respectively $z_0 > \|\bar{z}\|$), then $z_0 = \|\bar{z}\| = 0$ (respectively $x_0 = \|\bar{x}\| = 0$), and the result is true. The only remaining case is $x_0 = \|\bar{x}\|_2 > 0$ and $z_0 = \|\bar{z}\|_2 > 0$. The second equality in (1.9) shows that the Cauchy–Schwarz inequality in (1.8) holds with equality, so that

$$\bar{z} = \frac{\langle\, \bar{x}, \bar{z}\,\rangle}{\|\bar{x}\|_2^2}\bar{x} = -\left(\frac{z_0}{x_0}\right)\bar{x},$$

which is exactly the result. The reverse implication is immediate.

(iii) Both implications are obvious.

■

The "arrow matrix" notation arw$(\cdot)$ in Lemma 1.2.1 (ii) is due to [1]. The symmetrized complementarity condition in (1.17) for the semidefinite cone plays a key role in [7], and is derived from the equivalent form $x_S z_S = 0$ (discussed later in Remark 1.2.1) which appeared earlier in [2, 3, 79]. See also an early paper of Bellman and Fan [9], which derives the duality theory for SDP. The complementarity condition (1.18) for the quadratic cone is given in [1, 79].

It is now easy to derive explicit expressions for the normal cones to our three cones of interest. For the sake of simplicity, we state Proposition 1.2.2 for the j^{th} block, denoted $\mathcal{K}_j$. The normal cone to a direct sum is then just the direct sum of the individual normal cones.

Proposition 1.2.2 (Normal Cones To Symmetric Cones)
(i) $\mathcal{K}_j = \mathbb{S}_+^{N_j}$**:** *Let $x(j) \in \mathcal{K}_j$ have spectral decomposition*

$$x(j) = p\,\mathrm{Diag}([\lambda_1, \ldots, \lambda_{N_j}])\,p^*,$$

where without loss of generality, we may assume that $\lambda_1 \geq \ldots \geq \lambda_r > 0$ and $\lambda_{r+1} = \ldots = \lambda_{N_j} = 0$. Then

$$\mathcal{N}_{\mathcal{K}_j}(x(j)) = \left\{ v \in -\mathcal{K}_j \;:\; \frac{1}{2}(x(j)v + vx(j)) = 0 \right\}. \tag{1.10}$$

Equivalently,

$$\mathcal{N}_{\mathcal{K}_j}(x(j)) = \left\{ v \in -\mathcal{K}_j \;:\; v = p \begin{bmatrix} 0 & 0 \\ 0 & w \end{bmatrix} p^*,\; w \in \mathbb{S}^{N_j - r} \right\}. \tag{1.11}$$

(ii) $\mathcal{K}_j = \mathbb{Q}_+^{n_j}$**:** *Let $x(j) = [x_0(j), \bar{x}(j)] \in \mathcal{K}_j$.*

$$\mathcal{N}_{\mathcal{K}_j}(x(i)) = \{ v \in -\mathcal{K}_j \;:\; \mathrm{arw}(x(j))v = 0 \}. \tag{1.12}$$

Equivalently,

$$\mathcal{N}_{\mathcal{K}_j}(x(j)) = \begin{cases} \{0\} & \text{if} \quad x_0(j) > \|\bar{x}(j)\|_2 \\ -\mathcal{K}_j & \text{if} \quad x_0(j) = \|\bar{x}(j)\|_2 = 0 \\ \{ v \in -\mathcal{K}_j \;:\; v = \alpha[-x_0(j), \bar{x}(j)], \alpha \geq 0 \} & \text{if} \quad x_0(j) = \|\bar{x}(j)\|_2 > 0. \end{cases} \tag{1.13}$$

(iii) $\mathcal{K}_j = \mathbb{R}_+^{n_0}$**:** *Let $x(j) \in \mathcal{K}_j$.*

$$\mathcal{N}_{\mathcal{K}_j}(x(j)) = \{ v \in -\mathcal{K}_j \;:\; \mathrm{Diag}(x(j))v = 0 \}.$$

Proof. It is easy to verify one inclusion, namely that the vectors $v \in -\mathcal{K}_j$ in all three cases above satisy the normal cone inequality (1.4). To verify the other inclusion, assume that v satisfies (1.4). Substituting $x' = 2x$ and $x' = 0$ in (1.4) yields

$$\langle\, v, x(j) \,\rangle = 0, \tag{1.14}$$

hence in fact, $\langle\, v, x' \,\rangle \leq 0 \;\; \forall x' \in \mathcal{K}_j$. This implies that $v \in -\mathcal{K}_j$. Now apply Lemma 1.2.1 with $z(j) = -v$ to get the result.

Let us consider the equivalence in (i). One inclusion, namely that the normal cone of (1.11) is contained in the normal cone of (1.10), is easy to verify. To show the reverse inclusion, let $x(j) \in K_j$ have rank r and take a v satisfying (1.10). Since such a v must satisfy the normal cone inequality (1.4), we have, from (1.14) and Lemma 1.2.1, that $x(j)v = 0$. This means that $x(j)$ and v commute, and hence can be diagonalized by the same orthonormal matrix (see, for instance, [52, Theorem 2.5.5]), say p. Using this fact in $x(j)v = 0$, we obtain the desired inclusion. The equivalence in (ii) is straighforward. ■

Remark 1.2.1 (Semidefinite Cone Complementarity)
From the above proof of the equivalence in (i) in Proposition 1.2.2, it is clear that, for $(x, z) \in \mathcal{K} \times \mathcal{K}$, the complementarity condition $(xz + zx)/2 = 0$ involving a symmetrized matrix product is equivalent to the condition $xz = 0$, which involves a nonsymmetric matrix product.

From the normal cone expressions in Proposition 1.2.2 and the alternative characterization of complementarity in Lemma 1.2.1, it is clear that complementarity (as in Definition 1.2.2) is actually a property involving the normal cone, as the following equivalent definition suggests.

Definition 1.2.3 (Complementarity Via The Normal Cone)
We say that $(x, z) \in \mathcal{K} \times \mathcal{K}$ are *complementary*, or satisfy *complementarity*, if $-z \in \mathcal{N}_{\mathcal{K}}(x)$.

We can now write out explicit optimality conditions for COP.

Theorem 1.2.3 (Optimality Condition For COP)
Suppose the primal and the dual COP satisfy the Slater condition. Then x is primal optimal and (y, z) dual optimal if and only if

$$\begin{aligned}
&Ax = b && \textit{(primal equality constraints)} && (1.15)\\
&A^*y + z = c && \textit{(dual equality constraints)} && (1.16)\\
&(x_S z_S + z_S x_S)/2 = 0 && \textit{(SD complementarity)} && (1.17)\\
&\mathrm{Arw}(x_Q) z_Q = 0 && \textit{(QC complementarity)} && (1.18)\\
&\mathrm{Diag}(x_L) z_L = 0 && \textit{(LP complementarity)} && (1.19)
\end{aligned}$$

and

$$x_S, z_S \in \mathbb{S}^n_+; \quad x_Q, z_Q \in \mathbb{Q}^n_+; \quad x_L, z_L \in \mathbb{R}^n_+ \quad \textit{(cone constraints)}. \tag{1.20}$$

Proof. From Theorem 1.2.1 (Basic Optimality Condition) and Theorem 1.2.2 (Normal Cones to Intersections), we know that a primal feasible x is optimal if and only if there exists $(y, v) \in \mathbb{R}^m \times \mathcal{N}_{\mathcal{K}}(x)$ satisfying $A^*y + v + c = 0$. Setting $z = -v$, and using the normal cone expressions in Proposition 1.2.2, we get the optimality conditions claimed above.

Similarly, by retracing the steps above, a dual feasible (y, z) is optimal if and only if there exists $v' \in \mathcal{N}_{\mathcal{K}}(z)$ satisfying $-Av' = b$. Setting $x = -v'$, we recover the same optimality conditions again. ■

The following corollary about the duality gap is easily proved. (See also [79]).

Corollary 1.2.1 (Weak And Strong Duality)
(i) Weak duality *For any COP, if x and (y, z) are primal and dual feasible respectively, then $\langle\, c, x \,\rangle \geq \langle\, b, y \,\rangle$.*

(ii) Strong duality *Let COP satisy the Slater condition (Assumption 1.2.1). If x and (y, z) are optimal for the primal and the dual respectively, then $\langle c, x \rangle = \langle b, y \rangle$.*

Proof.

(i) Since x and (y, z) satisfy the primal (1.15) and the dual (1.16) equality constraints respectively, we have

$$\langle x, z \rangle = \langle x, c - A^* y \rangle = \langle c, x \rangle - \langle Ax, y \rangle = \langle c, x \rangle - \langle b, y \rangle, \tag{1.21}$$

i.e. the duality gap $\langle c, x \rangle - \langle b, y \rangle$ equals $\langle x, z \rangle$. Since $(x, z) \in \mathcal{K} \times \mathcal{K}$, we have $\langle x, z \rangle \geq 0$, resulting in the *weak duality* theorem: $\langle c, x \rangle \geq \langle b, y \rangle$, *i.e.* the duality gap $\langle c, x \rangle - \langle b, y \rangle$ is nonnegative.

(ii) If x and (y, z) are optimal, then the conditions (1.17) – (1.19) coupled with Lemma 1.2.1 imply that $\langle x, z \rangle = 0$ in (1.21). Hence the *strong duality* theorem: $\langle c, x \rangle = \langle b, y \rangle$, *i.e.* the duality gap must be zero.

■

1.3 Strict Complementarity

For a set C, let $\operatorname{ri} C$ denote its relative interior.

Definition 1.3.1 (Strict Complementarity)
A pair $(x, z) \in \mathcal{K} \times \mathcal{K}$ is said to be *strictly complementary*, or satisfy *strict complementarity*, if $-z \in \operatorname{ri} \mathcal{N}_{\mathcal{K}}(x)$.

Of course, strict complementarity implies complementarity, but not conversely.

We characterize complementarity and strict complementarity in terms of certain "eigenvalues" in the following proposition. Its proof follows from the definitions of complementarity (Definition 1.2.3), strict complementarity (Definition 1.3.1), and the explicit normal cone expressions in Proposition 1.2.2.

Proposition 1.3.1 (Complementarity Via Eigenvalues)
Let $(x, z) \in \mathcal{K} \times \mathcal{K}$.

(i) $\mathcal{K} = \mathbb{S}^N_+$**:** *For each $j = 1, \ldots, s$, let $\lambda_1(j) \geq \ldots \geq \lambda_{N_j}(j) \geq 0$ and $0 \leq \omega_1(j) \leq \ldots \leq \omega_{N_j}(j)$ be the eigenvalues of the matrices $x(j)$ and $z(j)$ arranged in nonincreasing and nondecreasing orders respectively. Then, x and z are complementary if and only if for all blocks j, each product $\lambda_i(j)\omega_i(j)$ $(i = 1, \ldots, N_j)$ vanishes, and are strictly complementary if and only if for all blocks j, exactly one term in each product $\lambda_i(j)\omega_i(j)$ $(i = 1, \ldots, N_j)$ vanishes.*

(ii) $\mathcal{K} = \mathbb{Q}^n_+$**:** *For each $j = 1, \ldots, q$, let $x(j) = [x_0(j), \bar{x}(j)]$ and $z(j) = [z_0(j), \bar{z}(j)]$. Let*

$$\begin{aligned} \lambda_1(j) &= x_0(j) + \|\bar{x}(j)\|_2; \qquad \lambda_2(j) = x_0(j) - \|\bar{x}(j)\|_2 \\ \omega_1(j) &= z_0(j) - \|\bar{z}(j)\|_2; \qquad \omega_2(j) = z_0(j) + \|\bar{z}(j)\|_2 \end{aligned}$$

Then x and z are complementary if and only if for all blocks j,

$$\lambda_i(j)\omega_i(j) = 0 \quad (i = 1, 2),$$

and are strictly complementary if and only if for all blocks j, exactly one term in each of the products above vanishes.

(iii) $\mathcal{K} = \mathbb{R}^{n_0}_+$**:** *Let $\lambda_1 \geq \ldots \geq \lambda_{n_0} \geq 0$ and $0 \leq \omega_1 \leq \ldots \leq \omega_{n_0}$ be rearrangements of the vectors x and z in nonincreasing and nondecreasing orders respectively. Then, x and z are complementary if and only if each product $\lambda_i\omega_i$ vanishes, and are strictly complementary if and only if exactly one term in each product $\lambda_i\omega_i$ vanishes.*

1.4 Nondegeneracy

In this section, we introduce another constraint qualification called *nondegeneracy*. Again, it is a notion involving the normal cone. For a convex set C, let $\operatorname{lin} C$ denote the smallest subspace containing C.

Definition 1.4.1 (Primal Nondegeneracy)
A feasible x for the primal COP (1.1) is said to be *primal nondegenerate* if

$$\operatorname{lin} \mathcal{N}_{\mathcal{K}}(x) \cap \operatorname{ran} A^* = \{0\}\,. \tag{1.22}$$

Definition 1.4.2 (Dual Nondegeneracy)
A feasible (y, z) for the dual COP (1.3) is said to be *dual nondegenerate* if

$$\operatorname{lin} \mathcal{N}_{\mathcal{K}}(z) \cap \ker A = \{0\}\,. \tag{1.23}$$

There is a geometric interpretation to the primal and the dual nondegeneracy conditions. Taking orthogonal complements on both sides of (1.22) and (1.23), it can be verified that primal (dual) nondegeneracy at a given feasible point x ((y, z)) amounts to the transversal intersection of the affine space $\{x : Ax = b\}$ ($\{z = c - A^*y : y \in \mathbb{R}^m\}$) and the boundary of the cone $\mathcal{K}$ at the chosen primal (dual) feasible point. For more details on nondegeneracy and strict complementarity, see [6, 89, 100] for SDP and [8, 44] for convex QCQP.

The following lemma about the uniqueness of solutions is established in [6] for SDP and in [8] for convex QCQP.

Lemma 1.4.1 (Uniqueness Of Solutions)
Let x (respectively (y, z)) be a primal (respectively dual) nondegenerate solution. Then the dual (respectively primal) solution (y, z) (respectively x) is unique.

1.5 Connections With Jordan Algebras

The close connections between the cones arising in COP and the theory of Jordan algebras were pointed out by L. Faybusovich [32, 33] and O. Güler [42, 43]. We give a brief introduction to the basic definitions and results, leading to the definition of an eigenvalue map on $\mathbb{E}$.

Definition 1.5.1 (Jordan Algebras)
Jordan algebra A vector space $\mathbb{E}$ with a commutative, bilinear product $\circ : \mathbb{E} \times \mathbb{E} \to \mathbb{E}$ is a *Jordan algebra* if for all $x \in \mathbb{E}$, the "multiplication–by–x" linear operator $L(x) : \mathbb{E} \to \mathbb{E} : y \mapsto x \circ y$ commutes with the operator $L(x^2)$, *i.e.*

$$x \circ (x^2 \circ y) = x^2 \circ (x \circ y) \quad \forall x, y \in \mathbb{E}.$$

Here, x^2 denotes $x \circ x$. Jordan algebras, though not necessarily associative, are power associative, *i.e.* $x^p \circ x^q = x^{p+q} \ \ \forall x \in \mathbb{E}$.

Identity A Jordan algebra $(\mathbb{E}, \circ)$ has an *identity element* $e \in \mathbb{E}$ if $e \circ x = x \circ e = x \ \forall x \in \mathbb{E}$. In the sequel, we only consider Jordan algebras with identity.

Rank For $x \in \mathbb{E}$, we define

$$\operatorname{rank}(x) = \min\{k > 0 \mid (e, x, x^2, \ldots, x^k) \text{ are linearly dependent}\},$$

and $\operatorname{rank}(\mathbb{E}) = \max_{x \in \mathbb{E}} \operatorname{rank}(x)$. Of course,[2] $\operatorname{rank}(x) \le \operatorname{rank}(\mathbb{E}) \le \dim \mathbb{E}$.

[2] The Jordan notion of rank is not to be confused with the linear algebraic notion of rank of a matrix which is the dimension of its range space. For example, $\operatorname{rank}(I) = 1$ in the Jordan sense.

Euclidean Jordan algebra A Jordan algebra is *Euclidean* if it admits a positive definite, symmetric, bilinear form which is also associative, *i.e.* there exists an inner product $\langle \cdot, \cdot \rangle$ on $\mathbb{E}$ such that $\forall x, u, v \in \mathbb{E}$,

$$\begin{array}{rcl} \langle x, x \rangle & > & 0 \ \ \forall x \neq 0 \\ \langle u, v \rangle & = & \langle v, u \rangle \\ \langle u, x \circ v \rangle & = & \langle x \circ u, v \rangle . \end{array} \tag{1.24}$$

Idempotents and Jordan frame For a Jordan algebra $\mathbb{E}$, an element $c \in \mathbb{E}$ is called an *idempotent* if $c^2 = c$. Two idempotents $c, d \in \mathbb{E}$ are orthogonal if $c \circ d = 0$. An idempotent is *primitive* if it is nonzero, and cannot be expressed as the sum of two (necessary orthogonal) nonzero idempotents. An orthogonal collection of idempotents $\{c_i\}$ is *complete* if $\sum_i c_i = e$, the identity element of the algebra. A complete, orthogonal collection of primitive idempotents is called a *Jordan frame*. If $\mathbb{E}$ has rank r, then every Jordan frame in $\mathbb{E}$ has exactly r elements. If $x \in \mathbb{E}$ satisfies $x \circ c_i = 0$ for all c_i in a Jordan frame, then $x = 0$.

Much of the fundamental work in the theory of Jordan algebras is due to Köcher [58], but the following nontrivial results about Euclidean Jordan algebras are conveniently compiled in Faraut and Korányi [31, Ch.I–V].

Theorem 1.5.1 (Results on Euclidean Jordan Algebras)
Let $(\mathbb{E}, \circ)$ be a Euclidean Jordan algebra.

Cone of squares *The set of squares $\mathcal{K} = \left\{x^2 \ : \ x \in \mathbb{E}\right\}$ forms a symmetric cone (called the* cone of squares*) in $\mathbb{E}$. Conversely, every symmetric cone is the cone of squares of some Euclidean Jordan algebra. (Recall that a cone is symmetric if it is both self–dual and homogeneous.)*

Classification *Let $\mathbb{C}$, $\mathbb{H}$ and $\mathbb{O}$ denote the complex numbers, the quaternions and the octonions respectively. Every Euclidean Jordan algebra $\mathbb{E}$ is isomorphic to*

- *the space of $k \times k$ symmetric matrices over $\mathbb{R}$, or Hermitian matrices over $\mathbb{C}$ or $\mathbb{H}$, with the product $x \circ z = (xz + zx)/2$, or*
- *the space $\mathbb{R} \times \mathbb{R}^k$ with the product $x \circ z = [x_0, \bar{x}] \circ [z_0, \bar{z}] = [\langle x, z \rangle, x_0\bar{z} + z_0\bar{x}]$, or*
- *the space of 3×3 matrices over $\mathbb{O}$ with the product $x \circ z = (xz + zx)/2$,*

or a direct sum thereof. Consequently, every symmetric cone is isomorphic to

- *$k \times k$ symmetric positive semidefinite matrices over $\mathbb{R}$, or Hermitian positive semidefinite matrices over $\mathbb{C}$ or $\mathbb{H}$, or*
- *vectors $x = [x_0, \bar{x}] \in \mathbb{R} \times \mathbb{R}^k$ satisfying $x_0 \geq \|\bar{x}\|_2$, or*
- *3×3 Hermitian positive semidefinite matrices over $\mathbb{O}$*

or a direct sum thereof.

Spectral theorem: *Let $\mathrm{rank}(\mathbb{E}) = r$. For every $x \in \mathbb{E}$, there exist (not necessarily distinct) real numbers $\lambda_1(x), \ldots, \lambda_r(x)$ (called* eigenvalues*) uniquely determined by x, and a (not necessarily unique) Jordan frame $\{c_1(x), \ldots, c_r(x)\}$ such that*

$$x = \sum_{i=1}^{r} \lambda_i(x) c_i(x).$$

The interior of the cone of squares is the set of all elements with positive eigenvalues.

Trace, Determinant and Inverse *For $x \in \mathbb{E}$, define*

$$\mathrm{tr}(x) = \sum_{i=1}^{r} \lambda_i(x)$$

$$\det(x) = \prod_{i=1}^{r} \lambda_i(x). \tag{1.25}$$

If $\det(x) \neq 0$*, then* x *is invertible, and its inverse is given by*

$$x^{-1} = \sum_{i=1}^{r} \lambda_i(x)^{-1} c_i(x).$$

The factor of $\frac{1}{2}$ in the Jordan product on the symmetric matrices ensures that $x \circ x = xx$, and hence allows the convenience of treating x^2 (a Jordan product) as an ordinary matrix product. We point out that for $\mathbb{S}^N$, the identity element is just the identity matrix I. For $\mathbb{Q}^n$, the identity, denoted e_Q, is the vector whose j^{th} block is $[1, 0, \ldots, 0] \in \mathbb{R} \times \mathbb{R}^{n_j}$. For $\mathbb{R}^{n_0}$, the identity, denoted e_L, is the vector of all ones.

The following proposition gives an explicit description of eigenvalues and Jordan frames for the cases of interest to us. In each case, it can be easily verified that the eigenvalues and Jordan frames constitute a valid spectral decomposition.

Proposition 1.5.1 (Eigenvalues And Jordan Frames)
Let $x \in \mathbb{E}$*, a Euclidean Jordan algebra.*

$\mathbb{E} = \mathbb{S}^N$ *For each block* $j = 1, \ldots, s$*, the eigenvalues of* $x(j)$ *given by the usual eigenvalues* $\lambda_i(x(j))$ $(i = 1, \ldots, N_j)$ *of the matrix* $x(j)$*, and the idempotents* $c_i(x(j)) = p_i(x(j))p_i(x(j))^*$ $(j = 1, \ldots, N_j)$*, where the* $p_i(x(j))$ *are corresponding eigenvectors, constitute a Jordan frame. The rank of* $\mathbb{S}^{N_j}$ *is* N_j*.*

$\mathbb{E} = \mathbb{Q}^n$ *For each block* $j = 1, \ldots, q$*, let* $x(j) = [x_0(j), \bar{x}(j)]$*, and*

$$u(x(j)) = \begin{cases} \bar{x}(j) / \|\bar{x}(j)\|_2, & \text{if } \|\bar{x}(j)\|_2 > 0 \\ \text{any vector in } \mathbb{R}^{n_j} \text{ satisfying } \|u(x(j))\|_2 = 1, & \text{otherwise.} \end{cases} \tag{1.26}$$

The eigenvalues of $x(j)$ *are* $\lambda_1(x(j)) = x_0(j) + \|\bar{x}(j)\|_2$ *and* $\lambda_2(x(j)) = x_0 - \|\bar{x}(j)\|_2$*. The idempotents* $c_1(x(j)) = \frac{1}{2}[1, u(j)]$ *and* $c_2(x(j)) = \frac{1}{2}[1, -u(j)]$ *constitute a Jordan frame. The rank of* $\mathbb{Q}^{n_j}$ *is* 2*.*

$\mathbb{E} = \mathbb{R}^{n_0}$ *The eigenvalues of* x *are its entries. The idempotents* $c_i(x) = e_i$ $(i = 1, \ldots, n)$ *(the unit vectors in* $\mathbb{R}^{n_0}$*) constitute a Jordan frame. The rank of* $\mathbb{R}^{n_0}$ *is* n_0*.*

The rank of a direct sum of Euclidean Jordan algebras is just the sum of the individual ranks.

Definition 1.5.2 (Eigenvalue Map)
Let $\mathbb{E}$ have rank r. For $x = (x_S, x_Q, x_L) \in \mathbb{E}$, we define the *eigenvalue map*

$$\lambda : \mathbb{E} \to \mathbb{R}^r : x \mapsto \lambda(x) = [\lambda(x_S), \lambda(x_Q), \lambda(x_L)],$$

where, in accordance with Proposition 1.5.1, the eigenvalues of x_S, x_Q and x_L are arranged in nonincreasing order within each block. We denote by $\lambda_i(x)$, the i^{th} component of $\lambda(x)$. We define

$$\begin{aligned} \lambda_{\max}(x) &= \max_i \lambda_i(x), \\ \lambda_{\min}(x) &= \min_i \lambda_i(x), \quad \text{and} \\ \lambda^+_{\min}(x) &= \min_{\lambda_i(x) > 0} \lambda_i(x). \end{aligned} \tag{1.27}$$

We define another eigenvalue map $\omega(\cdot)$ on $\mathbb{E}$ that orders eigenvalues in nondecreasing order within each block, *i.e.* $\omega(x) = -\lambda(-x)$ $\forall x \in \mathbb{E}$.

In particular, by Lemma 1.2.1, the pair $(x, z) \in \mathcal{K} \times \mathcal{K}$ is complementary if and only if $x \circ z = 0$. Also, from Proposition 1.3.1, (x, z) are complementary if and only if $\langle \lambda(x), \omega(z) \rangle = 0$, and are strictly complementary if and only if for each i, either $\lambda_i(x) = 0$ or $\omega_i(z) = 0$, but are not both zero.

1.6 Interior–Point Algorithms

In this section, we discuss interior–point alogrithms, which started with the landmark paper of Karmarkar [55] for LP. We only give a very brief and incomplete overview of the taxonomy and the basic ideas underlying these algorithms; details of convergence and complexity properties are well beyond the scope of the current discussion. For a comprehensive treatment of modern interior–point theory for COP, we refer the reader to the monograph of Nesterov and Nemirovskii [79], the forthcoming monograph of Renegar [91], and the articles of Nesterov and Todd [80, 81]. The book by Wright [111] deals with both the algorithmic issues and the computational details of primal–dual interior–point algorithms, albeit almost exclusively for LP. We have not cited the original references in the literature on which the development of modern interior–point theory is based; most of them are available in the combined bibliographies of the works cited above, and in that of [60].

Interior–point theory centers around the notion of a *nondegenerate, strongly ν–self–concordant, logarithmically homogeneous* barrier function for an open convex set $Q \subseteq \mathbb{E}$, *i.e.* a $\mathcal{C}^3$ convex function $\vartheta(x) : Q \to \mathbb{R}$ which satisfies for all $x \in Q$, $v \in \mathbb{E}$, and $t > 0$,

$$
\begin{aligned}
&| D^3\vartheta(x)[v,v,v] | \leq \frac{2}{\sqrt{\nu}} \left(D^2\vartheta(x)[v,v]\right)^{3/2} && (\nu\text{–self–concordance})\\
&x_i \to \partial Q \Rightarrow \vartheta(x_i) \to +\infty && (\text{strong self–concordance})\\
&v \neq 0 \Rightarrow D^2\vartheta(x)[v,v] > 0 && (\text{nondegenerate self–concordance})\\
&\vartheta(tx) = \vartheta(x) - \nu \ln t && (\text{logarithmic homogeneity}).
\end{aligned}
$$

For example, for the symmetric cones we consider, a traditional and well studied barrier function satisfying all the above conditions is the *log barrier*

$$\vartheta : \operatorname{int}\mathcal{K} \to \mathbb{R} : x \to -\ln\det(x) \quad (\text{see (1.25)}),$$

with self–concordance parameter $\nu = \sum_{j=1}^{s} N_j$, $2q$ and n_0 for $\mathcal{K} = \mathbb{S}^N$, $\mathbb{Q}^n$ and $\mathbb{R}^{n_0}$ respectively. Let us assume that the primal (1.1) and the dual (1.3) problems satisfy the Slater condition (Assumption 1.2.1), and that a strictly feasible starting point is known.

An interior–point algorithm may be a *potential reduction* or a *path following* algorithm. In a potential reduction algorithm, a suitable potential function ϕ_ρ, involving a suitable parameter ρ, is first chosen:

$$
\begin{aligned}
\phi_\rho(x;\zeta) &= \rho \ln\left(\langle\, c, x\,\rangle - \zeta\right) - \ln\det(x) && (\text{primal only})\\
\phi_\rho(x,y,z) &= \rho \ln\langle\, x, z\,\rangle - \ln\det(x) - \ln\det(z) && (\text{primal–dual}),
\end{aligned}
$$

where the parameter ζ is a lower bound on the optimal value of the program. (The dual potential function may be defined analogously.) From an interior point, we apply one step of Newton's method to approximately minimize the potential function ϕ_ρ, subject to the appropriate linear equality constraints ($Ax = b$ for primal, $A^*y + z = c$ for dual, or both of these for primal–dual). A suitable step taken along this direction ensures that the new iterate lies in the interior of the cone. Points where the potential is $-\infty$ correspond to solutions, and the bulk of the analysis is then devoted to establishing that the potential function can be decreased by a constant amount in each iteration. See [105] for details.

In this thesis, we will be interested in path following algorithms. For any $\mu > 0$, the function $(1/\mu)\langle\, c, x\,\rangle - \ln\det(x)$ is strongly convex on the primal feasible region $\{x : Ax = b, x \in \mathcal{K}\}$, and hence has a unique minimizer

$$x_\mu \triangleq \arg\min_x \left\{ \frac{1}{\mu}\langle\, c, x\,\rangle - \ln\det(x) \;:\; Ax = b, x \in \mathcal{K} \right\}. \tag{1.28}$$

Similarly, for the dual problem, we may associate with any $\mu > 0$, the unique minimizer

$$(y_\mu, z_\mu) \triangleq \arg\min_{(y,z)} \left\{ \frac{-1}{\mu}\langle\, b, y\,\rangle - \ln\det(z) \;:\; A^*y + z = c, z \in \mathcal{K} \right\}. \tag{1.29}$$

The trajectories $\{x_\mu : \mu > 0\}$, $\{(y_\mu, z_\mu) : \mu > 0\}$, and $\{(x_\mu, y_\mu, z_\mu) : \mu > 0\}$ are called the *primal central path*, the *dual central path*, and the *primal–dual central path* respectively, and limit points, as $\mu \downarrow 0$, of these

trajectories, correspond to a primal, a dual, or a primal–dual solution respectively. It is instructive to see the connection between the primal and the primal–dual central paths. If x_μ is on the primal central path, then the optimality condition for (1.28) — setting the gradient (using the Jordan inner product of (1.24)) of the Lagrangian to zero — yields

$$c - \mu x^{-1} - A^* y = 0,$$

where x^{-1} is the inverse of x defined in Theorem 1.5.1. From the dual equality constraint in (1.3), we recognize that $z = \mu x^{-1}$, and upon taking Jordan product with x, we get

$$x \circ z = \mu e,$$

where e is the identity element of $\mathbb{E}$. It is easy to verify that we arrive at the same parameterization of the primal–dual central path by starting with the optimality conditions for a point (y_μ, z_μ) on the dual central path.

A path following interior–point algorithm computes an approximate solution by following one of these central paths, and is called a *primal*, a *dual*, or a *primal–dual* algorithm depending on which central path it tries to follow. As before, the algorithm starts at an interior point, and then attempts to stay sufficiently close to the central path. At each iteration, the value of μ is decreased, and a new search direction is generated by one step of Newton's method. A suitable step along this direction is then calculated to ensure that the new iterate will remain in the interior of the cone. Indeed, as $\mu \downarrow 0$, the iterates converge to a solution of the program. Here, the bulk of the analysis lies in showing that the iterates can be restricted to some neighborhood of the central path while still achieving a sufficient decrease in μ.

Depending on the definition of the neighborhood used in each iteration, a path following method may be a *short step* (iterates confined to a tight neighborhood around the central path), or a *long step* (iterates lying in a wide neighborhood around the central path). Further, these algorithms may be combined with a *predictor–corrector* strategy whereby the search direction computed may be altered with one or more Newton steps, called *correctors*, that improve proximity to the central path. One type of a predictor–corrector algorithm is explained in Chapter 3.

When a strictly feasible starting point is not known, there is a class of "infeasible" algorithms, which start from interior points not satisfying the equality constraints in (1.1) and (1.3), but still enjoy the nice convergence and complexity properties of "feasible" algorithms by generating iterates that lie in an *infeasible neighborhood* of the central path. Points within the infeasible neighborhood satisfy the the property that the infeasibility (violation of primal and/or dual equality constraints) is bounded above by a constant times μ, so that decreasing μ automatically produces a corresponding decrease in the infeasibility. Yet another approach to handle the lack of a strictly feasible starting point is the *homogeneous self–dual embedding* of [112]. This transforms the original problem into a new COP with extra variables, but for which a strictly feasible starting point is readily available. See [78] for details.

Nesterov and Todd [80, 81] introduce a class of cones called *self–scaled cones*, namely cones $\mathcal{K}$ (say, in vector spaces $\mathbb{V}$) that admit a ν–self–concordant barrier function ϑ, which along with its conjugate

$$\vartheta^*(z) = \max_{x \in \mathcal{K}} - \langle\, z, x \,\rangle - \vartheta(x),$$

satisfies for all $v, x \in \operatorname{int} \mathcal{K}$,

$$\begin{aligned} &D^2\vartheta(v)x \in \operatorname{int} \mathcal{K}^*, \quad \text{and} \\ &\vartheta^*(D^2\vartheta(v)x) = \vartheta(x) - 2\vartheta(v) - \nu. \end{aligned}$$

They establish that such cones are isomorphic to their duals (in the sense that there exists an isomorphism $\pi : \mathbb{V} \to \mathbb{V}^*$ with $\pi\mathcal{K} = \mathcal{K}^*$), and that for any $x \in \operatorname{int} \mathcal{K}$ and $z \in \operatorname{int} \mathcal{K}^*$, there exists a unique "scaling point" $w \in \operatorname{int} \mathcal{K}$ such that $z = D^2\vartheta(w)x$ (cf. (1.2)), hence implying that the self–scaled cones are, in fact, exactly the symmetric cones.

We conclude this section with some remarks on the generality of conic optimization and interior–point theory. The success of interior–point methods lies in their theoretical properties (polynomial iteration complexity, and under modest assumptions, asymptotic superlinear convergence), as well as their practical success. Nesterov and Nemirovskii [79, p.50] have demonstrated the existence of a universal barrier — a self–concordant barrier for any convex set — so that the interior–point theory applies, at least in principle, to any convex program. Moreover, they have also shown [79, p.103] that any convex domain can be expressed as the intersection of an affine space and a convex (not necessarily symmetric) cone in a (possibly) higher dimension, hence implying that optimizing over the intersection of an affine space and a convex cone is the most general model for convex programming.

1.7 Applications

The fact that COP can be solved efficiently has led to a growing number of applications in recent years, especially for SDP and convex QCQP. See [68, 79, 109] and the references therein.

Several applications arise as instances of eigenvalue optimization. Here, the goal is to minimize the sum of the k largest eigenvalues of a symmetric matrix, which is a nonsmooth but convex function. Alizadeh [2] pointed out that these problems can be expressed as SDP's. A wide variety of problems in control theory involve *linear matrix inequalities*, *i.e.* semidefiniteness constraints on affine, (symmetric) matrix–valued functions. A large number of applications in control are listed in [14] and its sequel [27]. Another rich area of applications is in the design of relaxations for NP–hard combinatorial optimization problems (see, for instance, [69] for an early application of SDP relaxations in this area). For $c \in \mathbb{S}^k$, the following equivalence between *Boolean quadratic programming* (left) and an SDP with a rank constraint (right)

$$\min_{v \in \{-1,1\}^k} \langle\, v, cv \,\rangle \qquad \longleftrightarrow \qquad \begin{aligned} \max_{x \in \mathbb{S}^k} & \langle\, c, x \,\rangle \\ \text{s.t. } & \mathrm{Diag}(x) = e_L \\ & \mathrm{rank}(x) = 1 \\ & x \in \mathbb{S}^k_+ \end{aligned} \tag{1.30}$$

is easily verified by setting $x = vv^*$, and noting that the set of binary k–vectors v on $\{-1, 1\}$ is exactly the set of $k \times k$ rank–one, symmetric, positive semidefinite matrices x, with diagonal entries equal to 1. In (1.30), $\mathrm{rank}(x)$ denotes the rank of x in the linear algebraic sense, *i.e.* the dimension of the range space of x. Omitting the (nonconvex) rank–one constraint results in an SDP relaxation of the original Boolean quadratic program. The analysis of this relaxation by Goemans and Williamson [38] for the *maximum cut* problem (which resulted in a 0.878 approximation guarantee) was responsible for great interest in SDP in the combinatorial optimization community. Various other applications of SDP are cataloged in [109].

The *robust least squares* problem is to find $x \in \mathbb{R}^k$ minimizing $\|(A + \Delta A)x - (b + \Delta b)\|$,where only bounds on the uncertainties $\Delta A \in \mathbb{R}^{p \times k}$ and $\Delta b \in \mathbb{R}^p$ are known. This problem is formulated in [26] as a convex QCQP. Minimizing a sum of 2–norms and approximation in the complex ℓ_1–norm are also examples of convex QCQP. Another important application is in the optimal design of mechanical structures, especially truss topologies, where the total elastic energy stored in the structure is minimized subject to volume constraints on the interconnecting links [10–12]. Many other applications, *e.g.* contact problems in friction, digital filter design, portfolio optimization, are discussed in [68].

In Chapter 5, we describe a new application of SDP arising from quantum chemistry calculations.

1.8 Concluding Remarks

In principle, it suffices to consider only semidefinite programming, as the nonnegative orthant and the quadratic cone can be embedded in a semidefinite cone of appropriate dimension. For example, $\mathbb{R}^{n_0}_+$ is just the direct sum of n_0 trivial 1×1 semidefinite cones. Similarly, vectors $x(j) = [x_0(j), \bar{x}(j)] \in \mathbb{Q}^{n_j}_+$ are exactly those for which the corresponding symmetric matrix $\mathrm{arw}(x(j))$ is positive semidefinite. However, there is little to be gained from such an embedding. In fact, it is computationally advantageous to retain three distinct classes of cones.

The close connections between COP, interior–point algorithms and Euclidean Jordan algebras are that

- the Euclidean spaces $\mathbb{E}$ we have been considering thus far turn into Euclidean Jordan algebras when equipped with a suitable bilinear product,
- the cones in these spaces that we have dealt with are the well known symmetric cones studied in the theory of Jordan algebras,
- the usual log barrier function used in potential reduction interior–point algorithms is related to a well studied object in the theory of Jordan algebras, namely the *characteristic function* (see [31, Ch.I] for the definition) of the underlying symmetric cone, and
- the reformulated complementarity conditions (1.17) – (1.19) that appear in every interior–point iteration of a path following algorithm using the so called "XZ+ZX" search direction [7] (also known as the AHO search direction; see Chapter 3) correspond exactly to the Jordan product on $\mathbb{E}$.

Whereas earlier papers developed algorithms and complexity analyses for COP for each type of cone separately (*i.e.* LP, convex QCQP and SDP), the Jordan algebraic approach of [32, 33] (see also the related works [42, 43]) and the approach of Nesterov and Todd [80, 81] via self–scaled cones, provide an elegant and unified framework for COP. The recent work [97] uses associative algebras to show that the polynomial complexity results of [74] can be extended verbatim to all the symmetric cones, except the 3×3 Hermitian matrices over $\mathbb{O}$, which is not an associative algebra [31, p.86].

Chapter 2

Perturbation Theory

In this chapter, we derive perturbation results for conic optimization problems. This chapter extends earlier results of [77], which was joint work with M. L. Overton.

2.1 Introduction

Our aim in this chapter is to study the conditioning of conic optimization problems (COP) with respect to small perturbations, *i.e.* to quantify the change in the solution of a COP induced by a sufficiently small perturbation in the problem data. This section develops the notation and the background needed for the rest of the chapter.

First, recall the definitions of $\mathbb{E} = \mathbb{S}^N \times \mathbb{Q}^n \times \mathbb{R}^{n_0}$, the inner product $\langle \cdot, \cdot \rangle$ and the Jordan product $\circ$ on $\mathbb{E}$, the cone $\mathcal{K} \subseteq \mathbb{E}$, the primal (1.1) and the dual (1.3) COP, the Diag$(\cdot)$ and the Arw$(\cdot)$ notations of Chapter 1.

The Euclidean structure on $\mathbb{E}$ induces the *Frobenius* norm $\|x\| \stackrel{\Delta}{=} \sqrt{\langle x, x \rangle}$ on $x \in \mathbb{E}$. When a different norm is needed, we use an appropriate subscript, *e.g.* $\|\cdot\|_2$ for the 2–norm of a vector. Of course, using the eigenvalue map in Definition 1.5.2, we immediately recognize that

- $\|x_S\| \stackrel{\Delta}{=} \sqrt{\mathrm{tr}(x_S^2)} = \|\lambda(x_S)\|_2$,
- for $x_Q = [x_0(1), \bar{x}(1), \ldots, x_0(q), \bar{x}(q)]$,

$$\|x_Q\| \stackrel{\Delta}{=} \sqrt{2\left\{(x_0(1)^2 + \|\bar{x}(1)\|_2^2) + \ldots + (x_0(q)^2 + \|\bar{x}(q)\|_2^2)\right\}} = \sqrt{2}\,\|x_Q\|_2 = \|\lambda(x_Q)\|_2\,, \text{ and}$$

- $\|x_L\| \stackrel{\Delta}{=} \|x_L\|_2 = \|\lambda(x_L)\|_2$

The space $\mathbb{E} \times \mathbb{R}^m$, whose elements are denoted by pairs *e.g.* (x, y), is equipped with the norm $\|(x, y)\| \stackrel{\Delta}{=} \sqrt{\|x\|^2 + \|y\|^2}$. For an element $u = (x, y, z)$ in the primal–dual space $\mathbb{E} \times \mathbb{R}^m \times \mathbb{E}$, the norm is defined as

$$\|u\| = \|(x, y, z)\| = \sqrt{\|x\|^2 + \|y\|^2 + \|z\|^2}. \tag{2.1}$$

The norm on $y \in \mathbb{R}^m$ is arbitrary.

We use the symbol $|||\cdot|||$ to denote the induced operator norm.

We denote by $B(u, \rho)$, the open ball of radius ρ centered at u. Among functions that map $\mathbb{E} \times \mathbb{R}^m \times \mathbb{E}$ to itself, $\mathrm{Lip}_\gamma(B(u, \rho))$ denotes those that are Lipschitz continuous in $B(u, \rho)$, γ being the Lipschitz constant in the norm chosen in (2.1). We say that a function is *uniformly* Lipschitz continuous if it is Lipschitz continuous at every point in its domain with the same Lipschitz constant.

We will assume that the primal and the dual COP satisfy the Slater condition (see Assumption 1.2.1). Consequently, the primal and dual solutions exist, and the primal and the dual objective values are both

finite and equal (see Corollary 1.2.1). Thus, a triple (x_0, y_0, z_0) solves (1.1) and (1.3) if and only if x_0 is primal feasible, (y_0, z_0) is dual feasible and the complementarity condition $x_0 \circ z_0 = 0$ is satisfied, *i.e.* (x_0, y_0, z_0) is optimal if and only if it is a root of

$$f : \mathbb{E} \to \mathbb{E} \times \mathbb{R}^m \times \mathbb{E} : (x, y, z) \mapsto \begin{bmatrix} A^*y + z - c \\ Ax - b \\ x \circ z \end{bmatrix} \tag{2.2}$$

and $(x_0, z_0) \in \mathcal{K} \times \mathcal{K}$. We remind the reader that, as stated in Section 1.1, the square brackets in (2.2) indicate that the first and the third rows are to be interpreted as vectors, rather than as elements in $\mathbb{E}$.

We will also assume that the following condition is satisfied.

Assumption 2.1.1 (Nondegeneracy And Strict Complementarity)
There exist a primal nondegenerate solution x and a dual nondegenerate solution (y, z) such that (x, z) are strictly complementary.

By Lemma 1.4.1, this assumption guarantees that such a solution is unique.

We denote by $\circledast$ the symmetrized Kronecker product introduced in [7], *i.e.* given $v, w \in \mathbb{S}^N$,

$$v \circledast w : \mathbb{S}^N \to \mathbb{S}^N : a \mapsto \frac{1}{2}(wav + vaw).$$

The operator $v \circledast w$ may be represented as as a symmetric matrix of order d_S. The Jacobian of f can then be verified to be

$$J(x, y, z) = \begin{bmatrix} 0 & A^* & I \\ A & 0 & 0 \\ E & 0 & F \end{bmatrix}, \tag{2.3}$$

where $E = \mathrm{Diag}(z_S \circledast I, \mathrm{Arw}(z_Q), z_L)$, $F = \mathrm{Diag}(x_S \circledast I, \mathrm{Arw}(x_Q), x_L)$, and I is the identity matrix of appropriate dimension. Finally, we use the compact notation (A, b, c) to denote the COP's in (1.1) and (1.3).

The organization of the rest of the chapter is as follows. In Section 2.2, we state some preliminary results needed for the perturbation analysis in Section 2.3. In Section 2.4, we see how these bounds specialize to the case of linear programming, where a straighforward approach based on linear algebra is possible. We present an alternative approach for infinitesimal perturbations, based on the implicit function theorem, in Section 2.5, and conclude in Section 2.6 with some remarks on the assumptions made and pointers to related work in the literature.

2.2 Preliminary Results

We begin by stating some preliminary lemmas that will be used in the perturbation analysis. The following well known result is one that we will use repeatedly.

Lemma 2.2.1 (Banach Lemma [101, p. 118])
Let T be a nonsingular operator mapping $\mathbb{E} \times \mathbb{R}^m \times \mathbb{E}$ to itself, and let $\widetilde{T} = T + \Delta T$ be a perturbation of T. If $|||\, T^{-1}\Delta T \,||| < 1$, then $\widetilde{T}$ is nonsingular, and

$$|||\, \widetilde{T}^{-1} \,||| \leq \frac{|||\, T^{-1} \,|||}{1 - |||\, T^{-1}\Delta T \,|||}.$$

In what follows, $|\cdot|$ denotes the absolute value (componentwise, if the argument is a vector). An inequality between a vector and a scalar is to be interpreted componentwise. Also recall the eigenvalue map of Definition 1.5.2. We refer to the eigenvalues of $x \in \mathbb{E}$ in the Jordan sense (Definition 1.5.2), and also eigenvalues of linear operators on $\mathbb{E}$ (defined in the usual manner). For example, $x \in \mathbb{Q}^n$ has $2q$ eigenvalues in the Jordan sense, whereas the linear operator $\mathrm{Arw}(x) : \mathbb{Q}^n \to \mathbb{Q}^n$ has $q + \sum_{j=1}^{q} n_j$ eigenvalues, which are just the ordinary eigenvalues of the matrix $\mathrm{Arw}(x)$. We point this out to avoid potential confusion between these two usages of the word "eigenvalues". For example, $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue of its argument, which can be an element in $\mathbb{E}$ (see Definition 1.5.2) or a linear operator from $\mathbb{E}$ to $\mathbb{E}$.

LEMMA 2.2.2 (EIGENVALUES OF KRONECKER PRODUCTS)
Let $x, v \in \mathbb{S}^N$ be commuting matrices, and for each block $j = 1, \ldots, s$, let $\alpha_1, \ldots, \alpha_{N_j}$ and $\beta_1, \ldots, \beta_{N_j}$ denote the eigenvalues of $x(j)$ and $v(j)$ respectively. For each $j = 1, \ldots, s$, the $N_j(N_j+1)/2$ eigenvalues of $x(j) \circledast v(j)$ are given by

$$\frac{1}{2}\left(\alpha_k \beta_l + \beta_k \alpha_l\right), \quad 1 \le k \le l \le N_j.$$

In particular, when $v = I$, we have $|||\, x \circledast I \,||| = \max_i \mid \lambda_i(x) \mid \le \|x\|$.

Proof. See [7]. ■

LEMMA 2.2.3 (EIGENVALUES OF ARROW MATRICES)
Let $x \in \mathbb{Q}^n$, and for each block $j = 1, \ldots, q$, let $x(j) = [x_0(j), \bar{x}(j)] \in \mathbb{R} \times \mathbb{R}^{n_j}$. For each $j = 1, \ldots, q$, the $n_j + 1$ eigenvalues of $\operatorname{arw}(x(j))$ are

$$x_0(j) + \|\overline{x}(j)\|_2, x_0(j), \ldots, x_0(j), x_0(j) - \|\overline{x}(j)\|_2.$$

In particular, $|||\, \operatorname{Arw}(x) \,||| \le \|x\|$.

Proof. Since $\operatorname{arw}(x(j)) = x_0(j)I + e_Q(j)\bar{x}(j)^* + \bar{x}(j)e_Q(j)^*$, its eigenvalues are those of the matrix $e_Q(j)\bar{x}(j)^* + \bar{x}(j)e_Q(j)^*$ shifted by $x_0(j)$. The latter matrix has rank at most 2, and it is easy to verify that its two possibly nonzero eigenvalues are $\pm \|\bar{x}(j)\|_2$, with corresponding eigenvectors given by (1.26). The claimed inequality follows quickly: for $x(j), z \in \mathbb{R} \times \mathbb{R}^{n_j}$,

$$\begin{aligned} |||\, \operatorname{arw}(x(j)) \,||| &= \max_{\|z\|=1} \|\operatorname{arw}(x(j))z\| = \max_{\|z\|_2 = 1/\sqrt{2}} \sqrt{2}\, \|\operatorname{arw}(x(j))z\|_2 = \max_i \mid \lambda_i(\operatorname{arw}(x(j))) \mid \\ &= \mid x_0(j) + \|\bar{x}(j)\|_2 \mid \le \|x(j)\|, \end{aligned}$$

where the last inequality can be verified by squaring both sides. If $j_0 = \arg\max_j |||\, \operatorname{arw}(x(j)) \,|||$, then

$$|||\, \operatorname{Arw}(x) \,||| = \max_j |||\, \operatorname{arw}(x(j)) \,||| = |||\, \operatorname{arw}(x(j_0)) \,||| \le \|x(j_0)\| \le \|x\|,$$

thus concluding the proof. ■

LEMMA 2.2.4 (EIGENVALUE PERTURBATIONS IN $\mathbb{S}^N$ [82, P. 58])
Let $x, x + v \in \mathbb{S}^N$. For any block $j = 1, \ldots, s$,

$$\mid \lambda(x(j) + v(j)) - \lambda(x(j)) \mid \le \|v(j)\|.$$

LEMMA 2.2.5 (EIGENVALUE PERTURBATIONS IN $\mathbb{Q}^n$)
Let $x, x + v \in \mathbb{Q}^n$. For each block $j = 1, \ldots, q$,

$$\mid \lambda(x(j) + v(j)) - \lambda(x(j)) \mid \le \|v(j)\|.$$

Proof. Apply Lemma 2.2.4 to $\operatorname{Arw}(x)$ and $\operatorname{Arw}(x + v)$. ■

The next two lemmas deal with properties of the Jacobian (see (2.3)) associated with a COP. We will say that a solution $u_0 = (x_0, y_0, z_0)$ to a COP satisfies

- nondegeneracy, if x_0 is primal nondegenerate and (y_0, z_0) is dual nondegenerate, and
- strict complementarity, if (x_0, z_0) satisfy strict complementarity.

LEMMA 2.2.6 (NONSINGULARITY OF JACOBIAN)
Let $u_0 = (x_0, y_0, z_0)$ be a primal–dual solution to COP. Then, the Jacobian of f at the solution, $J(u_0)$, is nonsingular, if and only if u_0 satisfies nondegeneracy and strict complementarity.

See [44] for the proof. The "if" implication is also proved in [32], and extends the original result for SDP [7]. See also [8].

LEMMA 2.2.7 (LIPSCHITZ CONTINUITY OF JACOBIAN)
Let (A, b, c) define any COP, not necessarily satisfying the Assumption 2.1.1. Then, the Jacobian $J(u)$ associated with it is uniformly Lipschitz continuous, with 1 being a global Lipschitz constant.

Proof. Let $u_1 = (x_1, y_1, z_1)$, $v = (v_1, v_2, v_3)$, and $u_2 = (x_2, y_2, z_2)$ all be elements of $\mathbb{E} \times \mathbb{R}^m \times \mathbb{E}$. We have

$$\begin{aligned}
|||\ J(u_2) - J(u_1)\ ||| &= \max_{\|v\|=1} |||\ (E_2 - E_1)v_1 + (F_2 - F_1)v_3\ ||| \quad \text{(see (2.3))} \\
&\leq \max_{\|v_1\|=1} |||\ (E_2 - E_1)v_1\ ||| + \max_{\|v_3\|=1} |||\ (F_2 - F_1)v_3\ ||| \\
&= |||\ E_2 - E_1\ ||| + |||\ F_2 - F_1\ ||| \\
&\leq \|z_2 - z_1\| + \|x_2 - x_1\| \quad \text{(by Lemmas 2.2.2 and 2.2.3)} \\
&\leq \|u_2 - u_1\|\,,
\end{aligned}$$

hence the result. ■

We state a finite dimensional version of the Kantorovič theorem which is central to our perturbation analysis.

THEOREM 2.2.1 (KANTOROVIČ THEOREM [54, CH.XVIII])
Let $\rho_0 > 0$, $u_0 \in \mathbb{R}^p$, $h : \mathbb{R}^p \to \mathbb{R}^p$, and $h \in \mathcal{C}^1(B(u_0, \rho_0))$. Assume for any norm on $\mathbb{R}^p$ and the corresponding induced operator norm that the Jacobian $Dh(\cdot) \in \mathrm{Lip}_\gamma(B(u_0, \rho_0))$ with $Dh(u_0)$ nonsingular, and let

$$\beta = |||\ Dh(u_0)^{-1}\ |||, \quad \eta = |||\ Dh(u_0)^{-1} h(u_0)\ |||, \quad \alpha = \beta\gamma\eta, \quad \rho = \frac{1 - \sqrt{1 - 2\alpha}}{\beta\gamma}.$$

If (a) $\alpha \leq \frac{1}{2}$, and (b) $\rho \leq \rho_0$, then
(i) h has a unique zero, say $\tilde{u}_0$, in $\mathrm{cl}\, B(u_0, \rho)$, and
(ii) Newton's method with unit steps, started at u_0, converges to this unique zero $\tilde{u}_0$.

The following corollary is immediate.

COROLLARY 2.2.1 (NONSINGULARITY OF PERTURBED JACOBIAN)
Let the conditions of Theorem 2.2.1 be satisfied. If $\alpha < 1/2$, then $Dh(\tilde{u}_0)$ is nonsingular.

Proof. Since the conditions of Theorem 2.2.1 are satisfied, G must have a zero, say $\tilde{u}_0$, such that

$$\begin{aligned}
\|\tilde{u}_0 - u_0\| &\leq \rho = \frac{1 - \sqrt{1 - 2\alpha}}{\beta\gamma} && (2.4) \\
&\leq \frac{2\alpha}{\beta\gamma} \quad \text{when } 0 \leq \alpha \leq 1/2 && (2.5) \\
&< \frac{1}{\beta\gamma} \quad \text{when } \alpha < 1/2
\end{aligned}$$

so that

$$|||\ Dh(\tilde{u}_0) - Dh(u_0)\ ||| \leq \gamma\, \|\tilde{u}_0 - u_0\| < \frac{1}{\beta} = \frac{1}{|||\ Dh(u_0)^{-1}\ |||}.$$

The Banach Lemma (Lemma 2.2.1) now implies that $Dh(\tilde{u}_0)$ is nonsingular. ■

2.3 Perturbation Analysis

The two classical, qualitative notions of stability for a general mathematical programming problem are stability with respect to the optimal value, and stability with respect to the solution set [66]. Our analysis quantifies the latter for a COP satisfying the assumption, by explicitly bounding the change in the solution for a sufficiently small perturbation in the problem data. Consider a perturbation of the problem data A, b, and c in (1.1). In what follows,

$$\widetilde{A} = A + \Delta A, \quad \tilde{b} = b + \Delta b, \quad \text{and} \quad \widetilde{c} = c + \Delta c \tag{2.6}$$

all denote perturbations in the original problem (1.1). Correspondingly, (2.2) for the perturbed system becomes

$$\widetilde{f}(u) \stackrel{\Delta}{=} \widetilde{f}(x,y,z) \stackrel{\Delta}{=} \begin{bmatrix} \widetilde{A}^* y + z - \widetilde{c} \\ \widetilde{A}x - \tilde{b} \\ x \circ z \end{bmatrix} = 0. \tag{2.7}$$

and the Jacobian of $\widetilde{f}$ (see (2.3)) becomes

$$\widetilde{J}(u) \stackrel{\Delta}{=} \widetilde{J}(x,y,z) = \begin{bmatrix} 0 & \widetilde{A}^* & I \\ \widetilde{A} & 0 & 0 \\ E & 0 & F \end{bmatrix}. \tag{2.8}$$

We denote the solution to the original problem by $u_0 = (x_0, y_0, z_0)$ and the solution to the perturbed problem by $\tilde{u}_0 = (\tilde{x}_0, \tilde{y}_0, \tilde{z}_0)$. For a COP (A, b, c) satisfying Assumption 2.1.1 and whose solution is $u_0 = (x_0, y_0, z_0)$, and for its perturbation given in (2.6), we define the following quantities which will be used in the next theorem and the corollary following it. Using the projection $\pi : \mathbb{E} \times \mathbb{R}^m \times \mathbb{E} \to \mathbb{E} \times \mathbb{R}^m \times \mathbb{E} : (x,y,z) \mapsto (x,y,0)$, we define the linear operators $K = J(u_0)^{-1}\pi$ (see (2.3)) and $L = \pi J(u_0)$. Then, we set

$$\begin{aligned} \beta_0 &\stackrel{\Delta}{=} |\!|\!| J(u_0)^{-1} |\!|\!| \\ \beta_1 &\stackrel{\Delta}{=} |\!|\!| K |\!|\!|, \text{ and} \\ \delta_0 &\stackrel{\Delta}{=} \min\left\{\lambda^+_{\min}(x_0), \lambda^+_{\min}(z_0)\right\} \quad \text{(see Definition 1.5.2).} \end{aligned}$$

We now state the main result.

Theorem 2.3.1 (COP Perturbation Theorem)
Let u_0 be the primal–dual solution to the COP (A, b, c) satisfying the Assumptions, and let $[\widetilde{A}, \tilde{b}, \widetilde{c}] = (A + \Delta A, b + \Delta b, c + \Delta c)$. Let

$$\epsilon_0 \stackrel{\Delta}{=} |\!|\!| \Delta A |\!|\!| \, \|(x_0, y_0)\| + \|(\Delta c, \Delta b)\|.$$

If

$$|\!|\!| \Delta A |\!|\!| \leq \frac{1}{2\beta_1}, \quad \text{and} \tag{2.9}$$

$$\epsilon_0 < \min\left(\frac{\sigma - 1}{2\sigma^2\beta_0\beta_1}, \frac{\delta_0}{2\sigma\beta_1}\right) \quad \text{for some } 1 < \sigma \leq 2, \tag{2.10}$$

then
(i) the COP defined by $[\widetilde{A}, \tilde{b}, \widetilde{c}]$ has a solution, say $\tilde{u}_0$, which satisfies

$$\|\tilde{u}_0 - u_0\| \leq \frac{\sigma\beta_1\epsilon_0}{1 - \beta_1 |\!|\!| \Delta A |\!|\!|}, \tag{2.11}$$

(ii) the solution to $[\widetilde{A}, \tilde{b}, \widetilde{c}]$ is unique.
(iii) Newton's method with unit steps applied to $\widetilde{f}$, started at u_0, converges to $\tilde{u}_0$ quadratically.

Proof. To prove (i), we proceed in two steps. First, we use the Kantorovič theorem to show that $\widetilde{f}$ has a root $\tilde{u}_0$ that satisfies the bound in (2.11). Second, we show that $(\tilde{x}_0, \tilde{z}_0)$ lies in $\mathcal{K} \times \mathcal{K}$, and hence is a solution to the COP.

To use the Kantorovič theorem in the first step, we note the nonsingularity of the Jacobian $J(u_0)$ and the Lipschitz continuity of $J(\cdot)$ with Lipschitz constant $\gamma = 1$ (Lemma 2.2.6 and Lemma 2.2.7). Defining

$$\Delta J \triangleq \widetilde{J}(u_0) - J(u_0) = \begin{bmatrix} 0 & \Delta A^* & 0 \\ \Delta A & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.12}$$

we have $|||\ \Delta J\ ||| = |||\ \Delta A\ |||$, and therefore

$$|||\ J(u_0)^{-1}\Delta J\ ||| \leq \beta_1 |||\ \Delta A\ ||| \leq \frac{1}{2} \quad \text{(from (2.9))} \tag{2.13}$$

so that by the Banach Lemma (Lemma 2.2.1), $\widetilde{J}(u_0)$ is nonsingular with

$$\beta = |||\ \widetilde{J}(u_0)^{-1}\ ||| \leq 2\beta_0. \tag{2.14}$$

Let

$$\eta = \left\| \widetilde{J}(u_0)^{-1} \widetilde{f}(u_0) \right\| \quad \text{and} \quad \alpha = \beta\eta. \tag{2.15}$$

We need only verify assumption (a) of the Kantorovič theorem (Theorem 2.2.1), *i.e.* that $\alpha \leq \frac{1}{2}$; assumption (b) then follows trivially from the fact that the Lipschitz constant is global. We have

$$\widetilde{f}(u_0) = \begin{bmatrix} (A + \Delta A)x_0 - (b + \Delta b) \\ (A + \Delta A)^* y_0 + z_0 - (c + \Delta c) \\ x \circ z \end{bmatrix}$$

$$= \begin{bmatrix} (\Delta A)x_0 - \Delta b \\ (\Delta A)^* y_0 - \Delta c \\ 0 \end{bmatrix}, \tag{2.16}$$

so that

$$\left\| J(u_0)^{-1} \widetilde{f}(u_0) \right\| \leq \beta_1 \left\{ |||\ \Delta A\ |||\ \|\, (x_0, y_0)\, \| + \|(\Delta b, \Delta c)\| \right\} = \beta_1 \epsilon_0. \tag{2.17}$$

Therefore, we obtain the estimate

$$\begin{aligned} \eta &= \left\| \widetilde{J}(u_0)^{-1} \widetilde{f}(u_0) \right\| \\ &= \left\| \left(I + J(u_0)^{-1}\Delta J \right)^{-1} J(u_0)^{-1} \widetilde{f}(u_0) \right\| \\ &\leq \frac{\beta_1 \epsilon_0}{1 - |||\ J(u_0)^{-1}\Delta J\ |||} \quad \text{(from (2.17) and Lemma 2.2.1)} \qquad (2.18) \\ &\leq 2\beta_1\epsilon_0 \quad \text{(from (2.13))} \qquad (2.19) \end{aligned}$$

and from (2.14), (2.19) and (2.10), we conclude that

$$\alpha = \beta\eta \ \leq \ 4\beta_0\beta_1\epsilon_0 \ < \ \frac{2(\sigma - 1)}{\sigma^2} \ \leq \ \frac{1}{2}. \tag{2.20}$$

Since $\alpha < \frac{1}{2}$, the hypotheses of the Kantorovič theorem hold, whence we can conclude that $\widetilde{f}$ has a unique zero $\tilde{u}_0$, with

$$\|u_0 - \tilde{u}_0\| \leq \frac{1 - \sqrt{1 - 2\alpha}}{\beta}. \tag{2.21}$$

We have

$$\begin{aligned} & \sigma^2\alpha - 2\sigma + 2 < 0 \quad \text{(from (2.20))} \\ \Rightarrow \quad & \sigma^2\alpha^2 - 2\sigma\alpha + 2\alpha \leq 0 \quad \text{(since } \alpha = \beta\eta \geq 0\text{)} \\ \Rightarrow \quad & 1 - \sigma\alpha \leq \sqrt{1-2\alpha}, \end{aligned}$$

or equivalently,

$$\frac{1-\sqrt{1-2\alpha}}{\beta} \leq \frac{\sigma\alpha}{\beta} = \sigma\eta,$$

so that, using (2.21),

$$\|u_0 - \tilde{u}_0\| \leq \sigma\eta. \tag{2.22}$$

Combining this with (2.18) and (2.13) yields (2.11).

To show that this root is actually a solution to the COP, we need to establish that $\tilde{x}_0, \tilde{z}_0 \in \mathcal{K}$. To this end, note that

$$\begin{aligned} \|\tilde{u}_0 - u_0\| \quad & \leq \quad 2\sigma\beta_1\epsilon_0 \quad \text{(from (2.22) and (2.19))} \\ & < \quad \delta_0 \quad \text{(from (2.10))} \end{aligned}$$

so that

$$\|\tilde{x}_0 - x_0\| < \delta_0 \quad \text{and} \quad \|\tilde{z}_0 - z_0\| < \delta_0. \tag{2.23}$$

The following argument shows that $\tilde{x}_0 \in \mathcal{K}$. For each i,

$$\lambda_i(x_0) > 0 \Rightarrow \lambda_i(\tilde{x}_0) > 0 \quad \text{(from (2.23) and Lemmas 2.2.4, 2.2.5)},$$

and

$$\begin{aligned} \lambda_i(x_0) = 0 & \Rightarrow \omega_i(z_0) > 0 \quad \text{(strict complementarity of } x_0 \text{ and } z_0\text{)} \\ & \Rightarrow \omega_i(\tilde{z}_0) > 0 \quad \text{(from (2.23) and Lemmas 2.2.4, 2.2.5)} \\ & \Rightarrow \lambda_i(\tilde{x}_0) = 0 \quad \text{(complementarity of } \tilde{x}_0 \text{ and } \tilde{z}_0\text{)}. \end{aligned}$$

A similar argument shows that $\tilde{z}_0 \in \mathcal{K}$. Thus, $\tilde{u}_0 = (\tilde{x}_0, \tilde{y}_0, \tilde{z}_0)$ is indeed a solution to the perturbed COP. This concludes the proof of (i) in the theorem.

The proof of (ii) in the theorem is an immediate consequence of Corollary 2.2.1: since $\widetilde{J}(\tilde{u}_0)$ is nonsingular, Lemma 2.2.6 implies that the solution $\tilde{u}_0$ to the perturbed problem $[\widetilde{A}, \tilde{b}, \tilde{c}]$ satisfies strict complementarity and nondegeneracy. The latter property then guarantees, by Lemma 1.4.1, that $\tilde{u}_0$ is the unique solution.

The proof of (iii) is a consequence of the second conclusion of Theorem 2.2.1, combined with the nonsingularity of $\widetilde{J}(\tilde{u}_0)$. ■

The following corollary establishes a bound on the *relative* error in the solution of a perturbed COP, and thus introduces the notion of a *condition number* for COP.

Corollary 2.3.1 (Conditioning of COP)
Let the conditions of the COP Perturbation Theorem (Theorem 2.3.1) hold, and let $\Delta u_0 = \tilde{u}_0 - u_0$. *Then,*

$$\frac{\|\Delta u_0\|}{\|u_0\|} \leq \frac{\sigma \,|||\, K \,|||\, |||\, L \,|||}{1 - |||\, K \,|||\, |||\, \Delta A \,|||} \left(\frac{|||\, \Delta A \,|||\, \|(x_0, y_0)\|}{\|(c, b)\|} + \frac{\|(\Delta c, \Delta b)\|}{\|(c, b)\|} \right). \tag{2.24}$$

Proof. Observe that u_0 satisfies $Lu_0 = (c, b, 0)$, so that

$$|||\ L\ |||\, \|u_0\| \geq \|(c, b)\|\,.$$

The result follows by combining this inequality with (2.11). ■

Thus, $|||\ K\ |||\, |||\ L\ |||$ may be viewed as a condition number. In the special case $\Delta A = 0$, we have $\beta = \beta_0$, the inequality in (2.10) can be relaxed to

$$\epsilon_0 < \min\left(\frac{2(\sigma-1)}{\sigma^2\beta_0\beta_1}, \frac{\delta_0}{\sigma\beta_1}\right) \quad (1 < \sigma \leq 2)$$

and (2.24) reduces to

$$\frac{\|\Delta u_0\|}{\|u_0\|} \leq \sigma\, |||\ K\ |||\, |||\ L\ |||\, \left(\frac{\|(\Delta c, \Delta b)\|}{\|(c, b)\|}\right).$$

2.4 Linear Programs

In the case $\mathbb{E} = \mathbb{R}^n$, we recover linear programming (LP), and the cone under consideration is $\mathbb{R}^n_+$, the polyhedral cone of componentwise nonnegative vectors. In this case, a simpler perturbation analysis is possible via linear algebra. It is instructive to see how the perturbation results of the previous section specialize to LP.

Note that the nondegeneracy and the strict complementarity assumption (Assumption 2.1.1) simplifies considerably in this case. Nondegeneracy implies that the primal solution, rearranged as $x_0 = (x_0^1, x_0^2)$, has exactly m strictly positive components (denoted by x_0^1), and, that if we rearrange the columns of A as $[A_1\ \ A_2]$ with A_1 and A_2 corresponding to x_0^1 and x_0^2 respectively, then A_1 (the "basis matrix") is nonsingular. Writing $z_0 = (z_0^1, z_0^2)$ accordingly, we have $z_0^1 = 0$, and by strict complementarity, $z_0^2 > 0$. Therefore,

$$J(u_0) = \begin{bmatrix} A_1 & A_2 & 0 & 0 & 0 \\ 0 & 0 & A_1^* & I & 0 \\ 0 & 0 & A_2^* & 0 & I \\ 0 & 0 & 0 & \mathrm{Diag}(x_0^1) & 0 \\ 0 & \mathrm{Diag}(z_0^2) & 0 & 0 & 0 \end{bmatrix}.$$

Thus, Theorem 2.3.1 holds with $\beta_1 = |||\ K\ |||$, with

$$K = \begin{bmatrix} A_1^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & (A_1^{-1})^* & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -A_2(A_1^{-1})^* & I & 0 & 0 \end{bmatrix}.$$

However, it is possible to use a simple linear algebra argument to obtain a perturbation bound. Rearranging c as (c^1, c^2) and defining

$$R = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_1^* & 0 \\ 0 & A_2^* & I \end{bmatrix}, \tag{2.25}$$

the solution to the LP is given by $R\ (x_0^1, y_0, z_0^2) = (b, c^1, c^2)$, and $x_0^2 = 0, z_0^1 = 0$. Since this holds for any sufficiently small perturbation (so that the basis does not change), the standard perturbation result for square, nonsingular linear systems [99, p. 26] gives

$$\begin{aligned} \|\Delta u_0\| &= \left\|(\Delta x_0^1, \Delta y_0, \Delta z_0^2)\right\| \\ &\leq \frac{|||\ R^{-1}\ |||}{1 - |||\ R^{-1}\ |||\, |||\ \Delta R\ |||}\left(|||\ \Delta R\ |||\ \ \|(x_0, y_0)\| + \|(\Delta c, \Delta b)\|\right). \end{aligned} \tag{2.26}$$

Here, ΔR is the matrix obtained by replacing A_1, A_2 and I in (2.25) by ΔA_1, ΔA_2 and 0 respectively. Since $||\ \Delta R\ || = ||\ \Delta A\ ||$ and $||\ R^{-1}\ || = ||\ K\ ||$, the bound obtained in Theorem 2.3.1 via the Kantorovič theory specializes, except for the factor of σ, to the one in (2.26) obtained by linear algebra.

2.5 Asymptotic Error Bounds

An alternative approach is to introduce a perturbation parameter t (assumed to be a scalar, for simplicity), and study the solution $u(t) = (x(t), y(t), z(t))$ of the parametrized COP $[A(t), b(t), c(t)]$, where $A(\cdot)$, $b(\cdot)$ and $c(\cdot)$ are assumed to be at least $\mathcal{C}^1$ in some neighborhood around t_0, and $t = t_0$ corresponds to the original problem. We may now regard f defined in (2.2) to be $f(t, x, y, z)$, and replace A, b, c in the right hand side of (2.2) by the functions $A(t), b(t), c(t)$ respectively. In view of Lemma 2.2.6, the implicit function theorem states that $Du(t)$, the derivative of the solution with respect to t, is well defined and continuous in some neighborhood $(t_0 - \epsilon_0, t_0 + \epsilon_0)$ around t_0, with

$$Du(t_0) = -J(u_0)^{-1} D_t f(t_0, x_0, y_0, z_0). \tag{2.27}$$

Indeed, we can conclude that $\forall\, \delta > 0, \exists\, \epsilon(\delta) > 0$ such that $\forall t \in (t_0 - \epsilon(\delta), t_0 + \epsilon(\delta))$,

$$\| u(t) - u(t_0) \| \;\leq\; (|||\, Du(t_0)\, ||| + \delta) \mid t - t_0 \mid .$$

Thus, $|||\, Du(t_0)\, |||$ can be considered to be an *asymptotic error bound*.[1] However, the implicit function theorem does not provide a way to estimate $\epsilon(\delta)$. On the other hand, the Kantorovič approach uses $\Delta A = A(t) - A(t_0)$, $\Delta b = b(t) - b(t_0)$ and $\Delta c = c(t) - c(t_0)$ to provide explicit bounds both on $\|\tilde{u}_0 - u_0\|$ (see (2.11)) and on the permissible perturbations (see (2.9) and (2.10)), without any assumptions on the functions $A(\cdot), b(\cdot)$, and $c(\cdot)$. In the limiting case $t \longrightarrow t_0$, we have $\epsilon_0 \longrightarrow 0$, so that we may let $\sigma \longrightarrow 1$ in (2.10). Then, from (2.15) and (2.16), the quotient $\eta / \mid t - t_0 \mid \longrightarrow |||\, Du(t_0)\, |||$ given in (2.27). Hence, the Kantorovič bound in (2.22) divided by $\mid t - t_0 \mid$ approaches $|||\, Du(t_0)\, |||$.

2.6 Concluding Remarks

We conclude by making a few remarks about the assumptions made. The Slater condition (the existence of strictly feasible points for the primal and the dual) guarantees that the problem remains well–posed under small perturbations. A problem violating the Slater condition is ill–posed in the sense that it could become infeasible under an arbitrarily small perturbation. Indeed, in the absence of further qualifications on the data and the nature of the perturbation, the solution multifunction may not even be outer semicontinuous [96, Def. 5.4], if the Slater condition is violated.

Our assumption of nondegeneracy and strict complementarity (Assumption 2.1.1) is stronger than the Slater condition, but guarantees a unique solution to the COP, and is crucial for the application of the Kantorovič theory. Again, in the absence of further qualifications on the data and the nature of the perturbation, the solution multifunction may not be inner semicontinuous [96, Def. 5.4], if the nondegeneracy condition is violated. Nevertheless, the nondegeneracy and strict complementarity assumption is generically satisfied [6, 90].

For linear programming, in the special case of perturbations to b alone (*i.e.* $\Delta A = 0$, $\Delta c = 0$) and under the assumption that the perturbed problem has a nonempty solution set, Mangasarian and Shiau [71] bound the distance between the solution sets of the original and the perturbed problems in terms of the perturbation in b. Robinson [94] uses Hoffman's lemma for linear inequalities to bound the distance between the solution set of a linear program and a fixed point in the solution space. Renegar [92, 93] introduces the notion of the distance to ill–posedness, and derives error bounds for a general class of mathematical programs in the setting of reflexive Banach spaces. However, a feature common to all these results, including the one presented in this chapter, is that they require some form of knowledge of the solution (or the active set at the solution) of the original program. In this sense, computing the condition number of a COP involves at least as much work as solving the program itself.

[1] This notion of an asymptotic error bound was suggested by one of the anonymous referees of [77].

Chapter 3

Interior–Point Methods

Several interior–point codes for COP, and particularly SDP, have been actively developed over the last few years (see, for instance, [36, 103, 107]). In this chapter, we describe primal–dual path following methods for COP. We address several computational issues relating to linear algebra, data sparsity, safeguards and higher order corrections, and present numerical results on benchmark problems using *SeQuL*, a COP solver, which evolved from its predecessor SDPpack [4, 5], but incorporates many improvements. This chapter is based on joint work with J–P. Haeberly and M. L. Overton [45, 46]. The code *SeQuL* was almost entirely written by J–P. Haeberly.

3.1 Introduction

Recall the basic definitions of $\mathbb{E}$, $\mathcal{K} \subseteq \mathbb{E}$, the primal (1.1) and the dual (1.3) COP, and the notations of Section 1.1. Assuming the Slater condition (Assumption 1.2.1), the optimality conditions (see Theorem 1.2.3 and (2.2)) reduce to solving the smooth system of nonlinear equations

$$f(x,y,z) \triangleq \begin{bmatrix} A^*y + z - c \\ Ax - b \\ \frac{1}{2}\mathbf{vec}(x_S z_S + z_S x_S) \\ \mathrm{Arw}(x_Q) z_Q \\ \mathrm{Diag}(x_L) z_L \end{bmatrix} = 0 \tag{3.1}$$

along with the cone constraints in (1.20). The isometry **vec** as well as the Arw$(\cdot)$ and the Diag$(\cdot)$ notations in (3.1) were defined in Chapter 1.

As we will see in the next section, solving the COP will reduce to applying Newton's method to a relaxed version of the optimality conditions in (3.1) to generate a *search direction* $(\Delta x, \Delta y, \Delta z)$, along which a new iterate will be chosen. However, it turns out that there are many equivalent versions of the semidefinite complementarity condition in (3.1), which result in different search directions. The form we use, $\frac{1}{2}(x_S z_S + z_S x_S) = 0$, results in the AHO [7] (also known as the XZ+ZX search direction), whereas using $x_S z_S = 0$ (see Remark 1.2.1) results in the KSH/HRVW/M [49, 59, 75] (also known as the XZ direction) of the SDP literature. Yet another direction is the NT direction [108], which is obtained by using the condition $w^{-1/2}(x_S z_S)w^{1/2} = 0$, with $w = x_S^{1/2}(x_S^{1/2} z_S x_S^{1/2})^{-1/2} x_S^{1/2}$. More generally, using the Monteiro–Zhang symmetrization operator [76]

$$H_p(x_s z_S) = \frac{1}{2}\left(p(x_S z_S)p^{-1} + (p^{-1})^*(x_S z_S)^* p^*\right)$$

several families of search directions can be unified. The nonsingular matrix p is chosen so that $H_p(x_S z_S) = \mu I \iff x_S z_S = \mu I$. Hence the central path (see Section 1.6) remains unaltered if $x_S z_S = \mu I$ is replaced by $H_p(x_S z_S) = \mu I$. Choosing $p = I$ ($p = x_S^{1/2}$, or $p = w^{-1/2}$ with w as defined above) gives rise to the AHO (KSH/HRVW/M, or NT) direction. See [106] for more details on various search directions in SDP.

The sequel is organized as follows. We begin by describing a primal–dual path following interior–point algorithm, based on the AHO direction, in Section 3.2, and some computational issues pertaining to

this algorithm in Section 3.3. In Section 3.4 and Section 3.5, we introduce extensions of the "higher order corrections" (HOC) and "multiple centrality corrections" (MCC) schemes, introduced for linear programming (LP) by Mehrotra [72] and Gondzio [40] respectively, to COP, and describe the use of these methods in the primal–dual path–following algorithm of Section 3.2. We present numerical results demonstrating the relative merits of the higher order schemes in Section 3.6, and conclude in Section 3.7 with a brief summary.

3.2 Primal–Dual Path Following Methods

Recall that e_Q and e_L (see page 10) are the identity elements of $\mathbb{Q}^n$ and $\mathbb{R}^{n_0}$ respectively, *i.e.* for each block $j = 1, \ldots, q$, $e_Q(j) = [1, 0, \ldots, 0] \in \mathbb{R} \times \mathbb{R}^{n_j}$, and $e_L = [1, \ldots, 1]$, and that I denotes the identity element of $\mathbb{S}^N$.

For $\mu > 0$ consider the modified system

$$f_\mu(x, y, z) \stackrel{\Delta}{=} \begin{bmatrix} A^*y + z - c \\ Ax - b \\ g_\mu(x, z) \end{bmatrix} = 0, \tag{3.2}$$

where

$$g_\mu(x, z) = \begin{bmatrix} \frac{1}{2}\mathbf{vec}(x_S z_S + z_S x_S - 2\mu I) \\ \mathrm{Arw}(x_Q) z_Q - \mu e_Q \\ \mathrm{Diag}(x_L) z_L - \mu e_L \end{bmatrix}.$$

For any given $\mu > 0$, there exists a unique solution $(x_\mu, y_\mu, z_\mu) \in \mathbb{S}^N_{++} \times \mathbb{Q}^n_{++} \times \mathbb{R}^{n_0}_{++}$ to the system (3.2) [79]. Recall from Section 1.6 that the trajectory $\{(x_\mu, y_\mu, z_\mu) \mid \mu > 0\}$ is called the central path. The parameter μ plays the role of a homotopy parameter. Any limit point of the central path as $\mu \downarrow 0$ is a solution to the COP.

A typical primal–dual path–following algorithm for COP approximately solves a sequence of systems $f_{\mu^k}(x, y, z) = 0$ where $\mu^k \downarrow 0$. For each $\mu = \mu^k$, an approximate solution to (3.2) is computed by applying one step of Newton's method. Thus one computes a search direction $(\Delta x, \Delta y, \Delta z)$ as the solution of the linear system

$$\begin{bmatrix} 0 & A^* & I \\ A & 0 & 0 \\ E & 0 & F \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = -f_\mu(x, y, z) \tag{3.3}$$

where we recognize the matrix on the left hand side to be the Jacobian (see (2.3)) of f in (3.2). The current point is then updated as

$$\begin{aligned} x &\leftarrow x + \alpha \Delta x, \\ y &\leftarrow y + \beta \Delta y, \\ z &\leftarrow z + \beta \Delta z, \end{aligned}$$

for suitable values of the step lengths α and β, chosen to ensure that the new iterates x and z remain inside the cone.

The algorithm is described as follows. Let $\Delta x = (\Delta x_S, \Delta x_Q, \Delta x_L)$, and $\Delta z = (\Delta z_S, \Delta z_Q, \Delta z_L)$. For an iterate (x, y, z) strictly satisfying the cone constraints, we perform the following steps, which constitute one basic interior–point iteration:

Algorithm 3.2.1 (Basic Interior–Point Iteration)

Step 1 (Predictor Direction) Compute a predictor search direction $(\Delta x^{\mathrm{pred}}, \Delta y^{\mathrm{pred}}, \Delta z^{\mathrm{pred}})$ by solving (3.3) with μ set to 0.

Step 2 (Step Length Calculation) Compute positive primal and dual step lengths $\widehat{\alpha}^{\mathrm{pred}}$ and $\widehat{\beta}^{\mathrm{pred}}$ such that

$$x + \widehat{\alpha}^{\mathrm{pred}} \Delta x^{\mathrm{pred}} \quad \text{and} \quad z + \widehat{\beta}^{\mathrm{pred}} \Delta z^{\mathrm{pred}}$$

lie on the boundary of the cone $\mathbb{S}^N_+ \times \mathbb{Q}^n_+ \times \mathbb{R}^{n_0}_+$.

Step 3 (μ Calculation) Let

$$x' \triangleq x + \alpha^{\text{pred}} \Delta x^{\text{pred}} \quad \text{and} \quad z' \triangleq z + \beta^{\text{pred}} \Delta z^{\text{pred}}$$

where $\alpha^{\text{pred}} = \min(\widehat{\alpha}^{\text{pred}}, 1)$ and $\beta^{\text{pred}} = \min(\widehat{\beta}^{\text{pred}}, 1)$ and set

$$\mu = \left(\frac{\langle\, x', z' \,\rangle}{\langle\, x, z \,\rangle} \right)^3 \frac{\langle\, x, z \,\rangle}{\nu}, \tag{3.4}$$

where

$$\nu = \sum_{j=1}^{s} N_j + \sum_{j=1}^{q} n_j + n_0.$$

Step 4 (Corrector Direction) Recompute the (relaxed) complementarity conditions in the right hand side of (3.3) as

$$-g_\mu(x,z) - \begin{bmatrix} \frac{1}{2}\mathbf{vec}(\Delta x_S^{\text{pred}} \Delta z_S^{\text{pred}} + \Delta z_S^{\text{pred}} \Delta x_S^{\text{pred}}) \\ \text{Arw}(\Delta x_Q^{\text{pred}})\, \Delta z_Q^{\text{pred}} \\ \text{Diag}(\Delta x_L^{\text{pred}})\, \Delta z_L^{\text{pred}} \end{bmatrix}$$

and solve (3.3) again for a new search direction $(\Delta x, \Delta y, \Delta z)$.

Step 5 (Step Length Calculation) Recompute positive primal and dual step lengths $\widehat{\alpha}$ and $\widehat{\beta}$ such that

$$x + \widehat{\alpha} \Delta x \quad \text{and} \quad z + \widehat{\beta} \Delta z$$

lie on the boundary of the cone $\mathbb{S}_+^N \times \mathbb{Q}_+^n \times \mathbb{R}_+^{n_0}$.

Step 6 (Update) Update the iterates as

$$x \leftarrow x + \alpha \Delta x; \quad y \leftarrow y + \beta \Delta y; \quad z \leftarrow z + \beta \Delta z$$

where $\alpha = \min(\tau \widehat{\alpha}, 1)$ and $\beta = \min(\tau \widehat{\beta}, 1)$, with $\tau < 1$, *e.g.* $\tau = 0.999$.

The performance of the algorithm depends upon the choice of μ in Step 3 of the basic iteration. The rule in (3.4) is Mehrotra's predictor–corrector method [73], originally proposed for LP, which significantly reduces the number of interior–point iterations required to compute a solution of given accuracy. In developing the algorithm and the code, we have followed the practice in LP codes, where the implementations deviate (often significantly) from the theoretical versions of the algorithms, which impose careful restrictions on updating μ to ensure that all the iterates lie within a certain neighborhood of the central path. The algorithm as described above does not possess convergence or complexity guarantees. See the recent theses [24, 102] which discuss several variants of primal–dual interior–point algorithms for SDP and analyze their theoretical properties.

3.3 Computational Issues

In this section, we discuss computational issues that are important to the performance of the algorithm. First, we describe how the Newton system may be reduced to a smaller *Schur complement* matrix. Then, we discuss how sparsity in the data is handled. Finally, we elaborate on a safeguard used for setting the value of μ in Step 3 of the basic interior–point iteration.

3.3.1 Schur Complement

The linear system (3.3) may be reduced, by a standard block Gauss elimination procedure [7], to a more compact system (involving only the variable Δy) whose coefficient matrix on the left-hand side is the m by m Schur complement matrix

$$M \triangleq A\,E^{-1}F\,A^*. \tag{3.5}$$

Let us represent the $m \times d$ matrix A by the triples $((A_S)_i, (A_Q)_i, (A_L)_i) \in \mathbb{S}^N \times \mathbb{Q}^n \times \mathbb{R}^{n_0}$ $(i = 1, \ldots, m)$, where the i^{th} row of A is $[\mathbf{vec}((A_S)_i), \sqrt{2}(A_Q)_i, (A_L)_i]$ (see Section 1.1). We may then partition the columns of this matrix as $A = [A_S \;\; A_Q \;\; A_L]$, where A_S, A_Q, and A_L are $m \times d_S$, $m \times d_Q$ and $m \times n_0$ matrices respectively. With this notation, it is easy to verify that the Schur complement matrix M can be computed by summing up the contributions from the semidefinite, the quadratic and the linear parts of the problem, *i.e.*

$$M = M_S + M_Q + M_L, \tag{3.6}$$

$$M_S \triangleq A_S(I \circledast z_S)^{-1}(I \circledast x_S)A_S^*, \tag{3.7}$$

$$M_Q \triangleq A_Q\mathrm{Arw}(z_Q)^{-1}\mathrm{Arw}(x_Q)A_Q^*, \tag{3.8}$$

$$M_L \triangleq A_L\mathrm{Diag}(z_L)^{-1}\mathrm{Diag}(x_L)A_L^*. \tag{3.9}$$

Forming and factoring M is the most expensive part of the algorithm. Since M_S and M_Q are generally not symmetric[1] for the AHO search direction, the code uses the LU factorization provided by LAPACK to factor M. This factorization is then used to first obtain Δy, and upon back substitution, Δx and Δz.

To form contribution M_Q from the quadratic part, observe that multiplying a block vector by $\mathrm{Arw}(x_Q)$ requires only a blockwise scalar multiplication and two rank one updates, and is thus an $O(d_Q)$ operation. Similarly, the cost of solving a system of equations whose coefficient matrix is $\mathrm{Arw}(x_Q)$ is $O(d_Q)$, using the Sherman–Morrison–Woodbury formula [39].

3.3.2 Step Length Calculation

Once a search direction $(\Delta x, \Delta y, \Delta z)$ is calculated, we need to determine the distance from the current iterate (x, y, z) to $\partial\mathcal{K}$ along the direction $(\Delta x, \Delta y, \Delta z)$, *i.e.* we need to determine

$$\widehat{\alpha} = \sup\{t > 0 \;:\; x + t\Delta x \in \mathcal{K}\}, \;\text{ and}$$
$$\widehat{\beta} = \sup\{t > 0 \;:\; z + t\Delta z \in \mathcal{K}\}.$$

For a semidefinite block $x_S(j)$, this amounts to solving an eigenvalue problem. If $\Delta x_S(j)$ is not positive semidefinite, we may compute $\alpha_S(j)$, the largest value of t for which $x_S(j) + t\Delta x_S(j) \in \mathbb{S}_+^{N_j}$, as

$$\alpha_S(j) = \frac{1}{\lambda_{\max}(-x_S(j)^{-1}\Delta x_S(j))}.$$

If $\Delta x_S(j)$ is positive semidefinite, then $\alpha_S(j) = +\infty$.

For the j^{th} quadratic block, let $x_Q(j) = [x_0, \bar{x}]$ and $\Delta x_Q(j) = [\Delta x_0, \Delta\overline{x}]$. The inequality $x_Q(j) + \alpha_Q(j)\Delta x_Q(j) \in \mathbb{Q}_+^{n_j}$ is equivalent to $u_1(j)\alpha^2 + u_2(j)\alpha + u_3(j) \geq 0$, where

$$u_1(j) = \Delta x_0^2 - \|\Delta\overline{x}\|_2^2,$$
$$u_2(j) = 2\left(x_0\Delta x_0 - \langle\,\overline{x}, \Delta\overline{x}\,\rangle\right), \;\text{ and}$$
$$u_3(j) = x_0^2 - \|\overline{x}\|_2^2 > 0 \quad (\text{since } x_Q(j) \in \operatorname{int}\mathbb{Q}_+^{n_j}).$$

Let the roots of $u_1(j)\alpha^2 + u_2(j)\alpha + u_3(j) = 0$ be α_1 and α_2. Then $\alpha_Q(j)$ is computed as follows:

$$u_1(j) > 0: \quad \text{real positive roots} \qquad \Rightarrow \quad \alpha_Q(j) = \min\{\alpha_1, \alpha_2\}$$

[1] The KSH/HRVW/M and NT directions result in a symmetric Schur complement matrix.

$$\begin{array}{lll} & \text{real negative roots} & \Rightarrow \quad \alpha_Q(j) = +\infty \\ & \text{complex roots} & \Rightarrow \quad \alpha_Q(j) = +\infty \\ u_1(j) < 0: & \text{exactly one negative root} & \Rightarrow \quad \alpha_Q(j) = \max\{\alpha_1, \alpha_2\} \\ u_1(j) = 0: & \text{negative root} & \Rightarrow \quad \alpha_Q(j) = +\infty \\ & \text{positive root} & \Rightarrow \quad \alpha_Q(j) = -c/b. \end{array}$$

To see that the cases excluded above can never occur, it suffices to observe that when $u_1(j) \neq 0$, we have $\alpha_1\alpha_2 = u_3(j)/u_1(j)$ with $u_3(j) > 0$.

For the LP component, the step length calculation is the usual ratio test. Letting $x_L = [x_1, \ldots, x_{n_0}]$ and $\Delta x_L = [\Delta x_1, \ldots, \Delta x_{n_0}]$,

$$\alpha_L = \max_{1 \leq i \leq n_0} \frac{-x_i}{\Delta x_i}.$$

If $\Delta x_L \geq 0$, then $\alpha_L = +\infty$.

The primal step length $\widehat{\alpha}$ to $\partial\mathcal{K}$ is then

$$\widehat{\alpha} = \min\left\{\min_j \alpha_S(j), \quad \min_j \alpha_Q(j), \quad \alpha_L\right\}.$$

The dual step length $\widehat{\beta}$ is calculated similarly.

3.3.3 Sparsity

Many problems arising in applications have very sparse constraint matrices. This is the case, for example, in problems of truss topology design and clock mesh design [110]. In order to obtain the best performance of the algorithm, it is imperative to exploit sparsity in the formation and the factorization of the Schur complement matrix M in (3.5). The code takes advantage of sparsity in the formation of M_S, the semidefinite component of M (see (3.7)).

The code stores sparse blocks in the constraint matrices $(A_S)_i \quad (i = 1, \ldots, m)$ in the Harwell–Boeing format [25], and efficiently implements multiplication of a dense matrix with a sparse matrix and computation of the inner product of a dense matrix with a sparse matrix. If D is a dense matrix of size p_1 by p_2 and S is sparse of size p_2 by p_3, then the dense product $B = DS$ is computed as

$$B_j = \sum_{k=1}^{p_2} S(k,j) D_k,$$

where $B_j, D_k \in \mathbb{R}^{p_3}$ denote the j^{th} and k^{th} columns of B and D respectively. These are the only sparse operations that are needed to compute M_S. To see this, note that the (k,l) entry of M_S is obtained as

$$M_S(k,l) = \sum_{i=1}^{s} (\mathbf{vec}((A_S)_k(i)))^* \left(I \circledast z_S(i)\right)^{-1} \left(I \circledast x_S(i)\right) \mathbf{vec}((A_S)_l(i))$$

where the sum is over the s semidefinite blocks, and $(A_S)_k(i)$ denotes the i^{th} block of the k^{th} constraint matrix $(A_S)_k$. Each term in this sum is computed as follows. First, $h_l(i)$ is computed as the solution to the Lyapunov equation

$$h_l(i) z_S(i) + z_S(i) h_l(i) = x_S(i)(A_S)_l(i) + (A_S)_l(i) x_S(i). \tag{3.10}$$

Then, $M_S(k,l)$ is computed as

$$M_S(k,l) = \sum_{i=1}^{s} \left\langle\, (A_S)_k(i), h_l(i) \,\right\rangle. \tag{3.11}$$

Thus, we need to multiply sparse matrices with dense matrices to compute the right–hand sides in (3.10) and we need to compute inner product of sparse matrices with dense matrices to evaluate the summands in (3.11). Observe that even if the matrices $(A_S)_k \quad (k = 1, \ldots, m)$, are very sparse, x_S is generally dense.

See [35] for a more involved strategy to exploit sparsity.

3.3.4 Selecting μ

Let us rewrite equation (3.4) in Step 3 of the basic interior–point iteration as

$$\mu = \sigma \frac{\langle x, z \rangle}{\nu}$$

where σ is chosen by Mehrotra's cubic rule, namely

$$\sigma = \left(\frac{\langle x', z' \rangle}{\langle x, z \rangle} \right)^3 . \tag{3.12}$$

This choice of σ works well in practice but may be too aggressive on some of the more difficult COP's, causing a failure of the algorithm due to short steps, *i.e.* the step lengths computed in Step 6 of the basic interior–point iteration are too small to result in any significant progress. On such problems, it is preferable to adopt a more conservative strategy. For example, the algorithm can monitor the size of the step lengths $\widehat{\alpha}^{\text{pred}}$ and $\widehat{\beta}^{\text{pred}}$ computed in Step 2 of the basic interior–point iteration. If one of $\widehat{\alpha}^{\text{pred}}$ and $\widehat{\beta}^{\text{pred}}$ is small then σ is chosen closer to 1, increasing the centering component in $(\Delta x, \Delta y, \Delta z)$.

Here is an example of such a strategy as we implemented it. Modify the choice of σ in Step 5 of the basic interior–point iteration as follows:

$$\begin{array}{llllll}
\text{if} & \widehat{\alpha}^{\text{pred}} < 0.1 & \text{or} & \widehat{\beta}^{\text{pred}} < 0.1 & \text{then} & \sigma = 0.5 \\
\text{elseif} & \widehat{\alpha}^{\text{pred}} < 0.2 & \text{or} & \widehat{\beta}^{\text{pred}} < 0.2 & \text{then} & \sigma = 0.2 \\
\text{elseif} & \widehat{\alpha}^{\text{pred}} < 0.5 & \text{or} & \widehat{\beta}^{\text{pred}} < 0.5 & \text{then} & \sigma = 0.1 \\
\text{else} & \sigma = \left(\dfrac{\langle x', z' \rangle}{\langle x, z \rangle} \right)^3 . & & & &
\end{array}$$

In the next two sections we describe two modifications to the basic interior–point iteration. They are both extensions of well known methods for LP.

3.4 Higher Order Corrections

Implementations of Mehrotra higher order corrections (HOC) scheme for LP have been described in [16, 72, 111]. The performance of the method has been tested by Mehrotra [72] and Carpenter, Lustig, Mulvey and Shanno [16] on some collections of Netlib test problems [23]. Mehrotra reports that the use of higher order corrections resulted on the average in a reduction of 25% to 35% in the number of iterations, and significant savings in the *cpu* time on several problems. Carpenter, Lustig, Mulvey and Shanno report that the number of outer iterations of the algorithm can often be reduced with the use of higher order corrections but that the total number of back solves required is usually increased. The consensus seems to be that the method is not practical unless the cost of a back solve is negligible compared to the cost of factoring the Schur complement matrix.

The situation is more promising for SDP, and hence for COP, because the ratio of the cost of forming and factoring the Schur complement matrix defined in (3.5) to that of performing a back solve to compute a direction $(\Delta x, \Delta y, \Delta z)$ and the associated step lengths α and β is usually very much greater than it is in LP. Indeed, recalling the definitions of d, d_S, d_Q and d (see Section 1.1), we see that in the case of dense blocks, the cost of forming the Schur complement matrix M is

$$O\left(m \sum_{i=1}^{s} N_i^3 + m^2 d_s + m d_q + m^2 q + m^2 n_0 \right), \tag{3.13}$$

and the cost of factoring M is $O(m^3)$. On the other hand, the cost of computing the directions $(\Delta x, \Delta y, \Delta z)$ once the matrix M has been factored is

$$O\left(m^2 + \sum_{i=1}^{s} N_i^3 + m d \right) \tag{3.14}$$

and the cost of computing the steps to the cone boundaries is

$$O\left(\sum_{i=1}^{s} N_i^3 + d_q + n_0\right). \tag{3.15}$$

(See Appendix A for the derivation of the operation counts in (3.13), (3.14) and (3.15).)

The higher order corrections scheme (HOC) extends readily to COP. We assume that the directions $(\Delta x^{\rm pred}, \Delta y^{\rm pred}, \Delta z^{\rm pred})$ and the corresponding step lengths $\widehat{\alpha}^{\rm pred}$, $\widehat{\beta}^{\rm pred}$, $\alpha^{\rm pred}$ and $\beta^{\rm pred}$ have been initialized as in Steps 1 – 3 of the basic interior–point iteration, and we replace Steps 4 and 5 in the basic iteration with the following steps:

Algorithm 3.4.1 (Higher Order Corrections)
HOC Step 0 Initialize a loop counter to 1

HOC Step 1 Compute a new direction $(\Delta x, \Delta y, \Delta z)$ as in Step 4 and corresponding step lengths $\widehat{\alpha}$ and $\widehat{\beta}$ as in Step 5.

HOC Step 2 If $\widehat{\alpha} \geq \alpha^{\rm pred}$ or if the loop counter is 1 then accept the primal direction and set

$$\Delta x^{\rm pred} \leftarrow \Delta x \quad \text{and} \quad \widehat{\alpha}^{\rm pred} \leftarrow \widehat{\alpha}.$$

If $\widehat{\beta} \geq \beta^{\rm pred}$ or if the loop counter is 1 then accept the dual direction and set

$$\Delta z^{\rm pred} \leftarrow \Delta z, \quad \Delta y^{\rm pred} \leftarrow \Delta y \quad \text{and} \quad \widehat{\beta}^{\rm pred} \leftarrow \widehat{\beta}.$$

HOC Step 3 If both the primal and the dual direction have been accepted then proceed to HOC Step 4. Otherwise, terminate the HOC iteration: restore

$$\Delta x \leftarrow \Delta x^{\rm pred} \quad \text{and} \quad \widehat{\alpha} \leftarrow \widehat{\alpha}^{\rm pred}$$

if the primal direction was rejected; restore

$$\Delta z \leftarrow \Delta z^{\rm pred}, \quad \Delta y \leftarrow \Delta y^{\rm pred} \quad \text{and} \quad \widehat{\beta} \leftarrow \widehat{\beta}^{\rm pred}$$

if the dual direction was rejected; proceed to Step 6 of the basic iteration.

HOC Step 4 Increment the loop counter. If the iteration count exceeds a predefined bound then terminate the HOC iteration and proceed to Step 6 of the basic iteration. Otherwise go to HOC Step 1.

3.5 Multiple Centrality Corrections

The multiple centrality corrections (MCC) scheme for LP was introduced by Gondzio [40]. It is currently implemented in several LP solvers, including HOPDM [41] and PCx [22]. Its extension to COP can be summarized as follows. Let $(\Delta x^{\rm pred}, \Delta y^{\rm pred}, \Delta z^{\rm pred})$ denote a predictor direction from the current iterate (x, y, z) and let $\widehat{\alpha}^{\rm pred}$ and $\widehat{\beta}^{\rm pred}$ denote the corresponding primal and dual step lengths to the boundary of the cone $\mathbb{S}_+^N \times \mathbb{Q}_+^n \times \mathbb{R}_+^{n_0}$. Note that these quantities are typically initialized with the direction and step lengths computed in Steps 4 and 5 of the basic iteration. However, they can be initialized with the values obtained at the end of a HOC loop allowing the MCC scheme to be combined with HOC. Let μ be defined as in Step 3 of the basic interior–point iteration. We consider step lengths of the form $\overline{\alpha} = \min(1, \widehat{\alpha}^{\rm pred} + \delta)$ and $\overline{\beta} = \min(1, \widehat{\beta}^{\rm pred} + \delta)$ for some parameter $\delta > 0$, so $\overline{\alpha} > \widehat{\alpha}^{\rm pred}$ when $\widehat{\alpha}^{\rm pred} < 1$ and $\overline{\beta} > \widehat{\beta}^{\rm pred}$ when $\widehat{\beta}^{\rm pred} < 1$, and we consider the new point

$$(\overline{x}, \overline{y}, \overline{z}) = (x + \overline{\alpha}\Delta x^{\rm pred}, y + \overline{\beta}\Delta y^{\rm pred}, z + \overline{\beta}\Delta z^{\rm pred}).$$

Next we compute a corrector direction $(\Delta x^{\rm corr}, \Delta y^{\rm corr}, \Delta z^{\rm corr})$ aimed at improving the centrality of $(\overline{x}, \overline{y}, \overline{z})$. Let ρ denote the vector of eigenvalues of the matrix $\frac{1}{2}(\overline{x}_S\overline{z}_S + \overline{z}_S\overline{x}_S)$ and let p denote the corresponding orthogonal matrix of eigenvectors, so that

$$\frac{1}{2}(\overline{x}_S\overline{z}_S + \overline{z}_S\overline{x}_S) = p\,\mathrm{Diag}(\rho)\,p^*.$$

Let r_S (r_Q, r_L) denote the vector obtained by projecting the entries of ρ ($\mathrm{Arw}(\overline{x}_Q)\overline{z}_Q$, $\mathrm{Diag}(\overline{x}_L)\overline{z}_L$) onto a suitable neighborhood of $\mu[1, \ldots, 1]$ (μe_Q, μe_L), and let

$$R_S = p\,\mathrm{Diag}(r_S)\,p^*.$$

One may then consider a corrector direction $(\Delta x^{\rm corr}, \Delta y^{\rm corr}, \Delta z^{\rm corr})$ defined by the solution to the system (3.3) with the right hand side given by

$$\begin{bmatrix} 0 \\ 0 \\ \mathbf{vec}(R_S) - \frac{1}{2}(\overline{x}_S\overline{z}_S + \overline{z}_S\overline{x}_S) \\ r_Q - \mathrm{Arw}(\overline{x}_Q)\overline{z}_Q \\ r_L - \mathrm{Diag}(\overline{x}_L)\overline{z}_L \end{bmatrix}.$$

We may then combine the predictor and the corrector directions to get

$$(\Delta x, \Delta y, \Delta z) = (\Delta x^{\rm pred}, \Delta y^{\rm pred}, \Delta z^{\rm pred}) + (\Delta x^{\rm corr}, \Delta y^{\rm corr}, \Delta z^{\rm corr}).$$

The motivation for the correction is to obtain an increase in the step lengths when replacing the direction $(\Delta x^{\rm pred}, \Delta y^{\rm pred}, \Delta z^{\rm pred})$ by $(\Delta x, \Delta y, \Delta z)$, and so to obtain a larger reduction in complementarity.

Note that we attempt to adjust the eigenvalues of $\frac{1}{2}(\overline{x}_S\overline{z}_S + \overline{z}_S\overline{x}_S)$ rather than those of $\overline{x}_S\overline{z}_S$. The advantage of using the former matrix is that it is symmetric, but neither matrix is necessarily positive definite. Both have the same trace and therefore the same eigenvalue average. Furthermore, we have the following inequalities:

$$\lambda_{\min}(\frac{1}{2}(\overline{x}_S\overline{z}_S + \overline{z}_S\overline{x}_S)) \le \lambda_{\min}(\overline{x}_S\overline{z}_S)$$
$$\lambda_{\max}(\frac{1}{2}(\overline{x}_S\overline{z}_S + \overline{z}_S\overline{x}_S)) \ge \lambda_{\max}(\overline{x}_S\overline{z}_S).$$

where $\lambda_{\min}(B)$ and $\lambda_{\max}(B)$ denote the smallest and largest eigenvalues of the matrix B (see [53, Ex.20,p.187]).

The correction can be repeated, taking the direction $(\Delta x, \Delta y, \Delta z)$ just computed as the predictor direction for a new correction. This iteration stops when the new direction fails to yield sufficient progress or the iteration count exceeds a predefined bound. Here progress is deemed sufficient if the step lengths for the new direction are not less than those of the previous direction and either the new primal step length or the new dual step length exceeds the corresponding step length for the previous direction by a preset margin.

We now describe the actual implementation of MCC that we used. Following Gondzio's code for LP (private communication), we define the function $\phi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ by

$$\phi(\xi, \mu) = \begin{cases} \mu - \xi & \text{if } \xi < \kappa_{\min}\mu \\ \max(\kappa_{\max}\mu - \xi, -\kappa_{\max}\mu) & \text{if } \xi > \kappa_{\max}\mu \\ 0 & \text{otherwise} \end{cases}$$

where $\kappa_{\min}$, $\kappa_{\max}$ are parameters chosen to satisfy

$$0 < \kappa_{\min} \le 1 \le \kappa_{\max}.$$

We need two more parameters, γ and δ, satisfying $0 < \gamma < 1$ and $\delta > 0$. We assume that the step lengths $\widehat{\alpha}$, $\widehat{\beta}$ and the direction $(\Delta x, \Delta y, \Delta z)$ have been computed as in Steps 4 and 5 of the basic interior–point iteration or by HOC. We insert the following steps in between Steps 5 and 6 of the basic interior–point iteration:

Algorithm 3.5.1 (Multiple Centrality Corrections)
MCC Step 1 Set

$$\Delta x^{\text{pred}} \leftarrow \Delta x, \;\; \Delta y^{\text{pred}} \leftarrow \Delta y, \;\; \Delta z^{\text{pred}} \leftarrow \Delta z, \;\; \widehat{\alpha}^{\text{pred}} \leftarrow \widehat{\alpha}, \;\; \widehat{\beta}^{\text{pred}} \leftarrow \widehat{\beta}.$$

MCC Step 2 Compute the modified step lengths as

$$\overline{\alpha} = \min(1, \widehat{\alpha}^{\text{pred}} + \delta) \quad \text{and} \quad \overline{\beta} = \min(1, \widehat{\beta}^{\text{pred}} + \delta)$$

and compute the point $(\overline{x}, \overline{y}, \overline{z})$ as

$$(\overline{x}, \overline{y}, \overline{z}) = (x + \overline{\alpha}\Delta x^{\text{pred}}, y + \overline{\beta}\Delta y^{\text{pred}}, z + \overline{\beta}\Delta z^{\text{pred}}).$$

MCC Step 3 Compute v_S, v_Q and v_L as follows:

1. Let ρ denote the vector of eigenvalues of the matrix $\frac{1}{2}(\overline{x}_S\overline{z}_S + \overline{z}_S\overline{x}_S)$, let p denote the corresponding orthogonal matrix of eigenvectors, and let u_S denote the vector obtained from ρ by setting

$$(u_S)_j = \phi(\rho_j; \mu), \quad 1 \le j \le \sum_{i=1}^{s} N_i.$$

and let

$$v_S = \mathbf{vec}(p\,\text{Diag}(u_S)\,p^*).$$

2. Let v_Q denote the block vector obtained from $w_Q = \text{Arw}(\overline{x}_Q)\overline{z}_Q$ as follows. Let $(w_0, w_1, \ldots, w_{n_i})$ (respectively $(v_0, v_1, \ldots, v_{n_i})$) denote the i^{th} block of w_Q (respectively v_Q) and set

$$v_0 = \phi(w_0, \mu).$$

For $1 \le j \le n_i$, set

$$v_j = \begin{cases} -\epsilon - w_j & \text{if } w_j < -\epsilon \\ \epsilon - w_j & \text{if } w_j > \epsilon \\ 0 & \text{otherwise} \end{cases}$$

where $\epsilon = \min((1 - \kappa_{\min})\mu, v_0/n_i)$.

3. Let v_L denote the vector obtained from $w_L = \text{Diag}(\overline{x}_L)\overline{z}_L$ by setting

$$(v_L)_j = \phi((w_L)_j; \mu), \quad 1 \le j \le n_0.$$

MCC Step 4 Compute a corrector direction $(\Delta x^{\text{corr}}, \Delta y^{\text{corr}}, \Delta z^{\text{corr}})$ as the solution of the system of equations (3.3) with right–hand side

$$[0 \quad 0 \quad v_S \quad v_Q \quad v_L].$$

MCC Step 5 Compute the step lengths $\widetilde{\alpha}$ and $\widetilde{\beta}$ to the boundary of the cones for the direction

$$(\Delta x^{\text{pred}}, \Delta y^{\text{pred}}, \Delta z^{\text{pred}}) + (\Delta x^{\text{corr}}, \Delta y^{\text{corr}}, \Delta z^{\text{corr}}).$$

Let $\alpha^{\text{pred}} = \min(\widehat{\alpha}^{\text{pred}}, 1)$ and $\beta^{\text{pred}} = \min(\widehat{\beta}^{\text{pred}}, 1)$. If $\widetilde{\alpha} \ge \alpha^{\text{pred}}$ then accept the new primal direction and set $\Delta x \leftarrow \Delta x^{\text{pred}} + \Delta x^{\text{corr}}$ and $\widehat{\alpha} \leftarrow \widetilde{\alpha}$. If $\widetilde{\beta} \ge \beta^{\text{pred}}$ then accept the new dual direction and set $\Delta y \leftarrow \Delta y^{\text{pred}} + \Delta y^{\text{corr}}$, $\Delta z \leftarrow \Delta z^{\text{pred}} + \Delta z^{\text{corr}}$ and $\widehat{\beta} \leftarrow \widetilde{\beta}$.

MCC Step 6 If $\widetilde{\alpha} < \alpha^{\text{pred}} + \gamma\delta$ and $\widetilde{\beta} < \beta^{\text{pred}} + \gamma\delta$ then terminate the MCC iteration, because the last correction has failed to achieve sufficient progress. Also terminate the MCC iteration if either the new primal or the new dual direction was rejected or if $\min(\widehat{\alpha}, \widehat{\beta})$ is greater than a predefined threshold value, *e.g.* 0.99. Finally, if the iteration count exceeds a predefined bound then terminate the MCC iteration; otherwise, go back to MCC Step 1.

Several variants of the implementation are possible. For example, one can terminate the MCC iterations only when both the new primal and dual directions have been rejected; one can also modify the criteria that determine whether sufficient progress has been achieved (see MCC Step 6). We have found that the implementation described above performed best on our set of test problems.

3.6 Numerical Results

We now present numerical results for the proposed schemes on a variety of test problems described in Table 3.1. The truss topology design problems were contributed by A. Nemirovskii. The clock mesh design problems are circuit design problems dealing with the propagation of a clock signal in integrated circuits; see [110] for details. All the runs were performed on an SGI workstation with an R10000 processor, an R10010 floating point unit and 128 MB of memory.

All the algorithms use the infeasible initial point

$$y = 0,\ (x_S, x_Q, x_L) = (z_S, z_Q, z_L) = \eta(I, e_Q, e_L)$$

where η is a positive scale factor. The choice of η can greatly affect the robustness; generally a larger value of η makes the algorithm more reliable on harder problems but also require more iterations on easier ones. For the runs reported here we used $\eta = 1000$. The choice of step length parameter τ in Step 6 of the basic iteration affects the results in similar ways: a value close to one gives faster convergence on most problems but may occasionally lead to failure. For the runs reported here, we used $\tau = 0.999$. We also set $\kappa_{\min} = 0.1$, $\kappa_{\max} = 10$, $\delta = 0.1$, and $\gamma = 0.1$ (see Section 3.5). The threshold value in MCC Step 6 was set to 0.99.

In Figure 3.1, we show the performance of the higher order corrections (HOC), the multiple centrality corrections (MCC) and the combination of HOC, MCC and SAFE (the safeguard discussed in Section 3.3.4) on the eight problems in Table 3.1. If the safeguard is activated, *i.e.* σ is not set according to Mehrotra's cubic formula, then no HOC or MCC type of corrections are attempted for the current iteration. We discuss the effect of this option in Section 3.6. The maximum number of corrections per basic interior–point iteration was set to six for both HOC and MCC. These results are shown together with those for the basic algorithm (BASIC). The top two plots in Figure 3.1 show the final primal infeasibility (the norm of the left-hand side of (1.15)) and the final complementarity (the sum of the left hand sides of (1.17), (1.18) and (1.19)) on log scales (base 10). Looking at these plots, we see that all problems were satisfactorily solved by all schemes, except that MCC did not solve problem 1 (*truss5* in Table 3.1) and BASIC, MCC, and MCC+HOC+SAFE all failed to solve problem 8 (*sqlp4* in Table 3.1)[2]. In all cases (except the failures just noted), the dual infeasibility norm was reduced below 10^{-12}, so this is not shown. The bottom two plots in Figure 3.1 show the iteration count and the relative *cpu* time, omitting the data points for the failures just noted, so that the other results may be compared more conveniently. The data shown in the latter of these two plots is normalized so that the time for HOC is 1. In other words, it shows the *cpu* time taken by a particular scheme divided by the *cpu* time taken by the HOC scheme.

We conclude that the HOC scheme is superior: it is the only scheme to solve all the problems; it yields the smallest number of iterations on all the problems except problem 7 where MCC wins by one iteration; and it achieves the best *cpu* time on all the problems except problems 4 and 7 where MCC is faster by less than 10%.

In Figure 3.2, we show the variation in the number of interior–point iterations and the *cpu* time needed to solve *truss8* (see Table 3.1) as a function of the maximum number of corrections allowed per basic interior–point iteration, both for HOC and MCC. We see that, for this particular test problem, the optimal strategy for HOC is to set the maximum number of corrections to six and that even when allowing up to twenty corrections per iteration HOC still performs better than the basic code. On the other hand, MCC performs best when allowing no more than one correction per iteration and the benefits of the scheme rapidly disappear as the maximum number of corrections allowed increases. The best performance of MCC is just slightly better than HOC's best performance.

Finally, in Figure 3.3, we compare the savings in iteration count and *cpu* time as a function of problem size for HOC and MCC. We consider randomly generated dense SDP's and QCQP's whose sizes are given in Table 3.2 and Table 3.3 respectively. For each size, averages were computed over five random instances. The maximum number of corrections per basic interior–point iteration was set to six for both HOC and MCC. We see that, in the case of SDP, all of HOC, MCC, and MCC+HOC+SAFE achieve substantial reductions in the number of iterations in all the problems and substantial reductions in the *cpu* time for problems of size greater than 50. In the case of QCQP, MCC+HOC+SAFE performs poorly, but the performance of HOC and MCC is comparable to the case of SDP. Overall, HOC is clearly superior, although it should be noted that MCC outperforms HOC on small problems.

[2]The BASIC scheme still does not solve problem 8 when the initial scale factor η is raised to 10000, but it does so when τ is set to 0.95 instead of 0.999.

#	Name	Type	N	n	n_0	m	Description
1	*truss5*	SD	$N_1 = 10$ $\vdots$ $N_{33} = 10$ $N_{34} = 1$	[ ]	0	208	Problems from truss topology design
2	*truss8*	SD	$N_1 = 19$ $\vdots$ $N_{33} = 19$ $N_{34} = 1$	[ ]	0	496	(sparse)
3	*clk1*	SD, LP	$N_1 = 81$	[ ]	288	144	Clock mesh design in
4	*clk2*	SD, LP	$N_1 = 169$	[ ]	624	312	VLSI circuits (sparse)
5	*sqlp1*	SD	$N_1 = 50$ $N_2 = 50$ $N_3 = 50$	[ ]	0	300	Randomly generated problems (dense)
6	*sqlp2*	QC	[ ]	$n_1 = 200$ $n_2 = 200$ $n_3 = 200$	0	300	
7	*sqlp3*	LP	[ ]	[ ]	1000	300	
8	*sqlp4*	SD QC LP	$N_1 = 50$ $N_2 = 50$ $N_3 = 50$	$n_1 = 200$ $n_2 = 200$ $n_3 = 200$	1000	300	

Table 3.1: The set of test problems. The columns marked N, n and n_0 correspond to the block structure of the semidefinite (SD), quadratic (QC) and linear (LP) parts of the problem, and m is the number of primal equality constraints.

3.7 Concluding Remarks

We presented a primal–dual path following interior–point algorithm based on Mehrotra's predictor–corrector method for COP, and discussed several of computational issues such as linear algebra and sparsity. We presented extensions of Mehrotra's higher order corrections (HOC) scheme and Gondzio's multiple centrality corrections (MCC) scheme to COP.

Experience reported for LP has been that, although both these higher order methods usually reduce the number of iterations required, MCC typically yields a reduction in *cpu* time while HOC does not. Our experience with these methods in the context of COP has been somewhat different. These methods are attractive for COP because the relative cost of performing a single correction, compared to the cost of one basic interior–point iteration, is much less for COP than for LP. While the savings obtained with MCC in COP are comparable to those reported for LP, our experience is that the benefits derived from Mehrotra's HOC method are much greater for COP than have been reported for LP. Finally, we have presented benchmarks showing that the use of the higher order schemes substantially improves the iteration count, the solution time and the robustness of the algorithm.

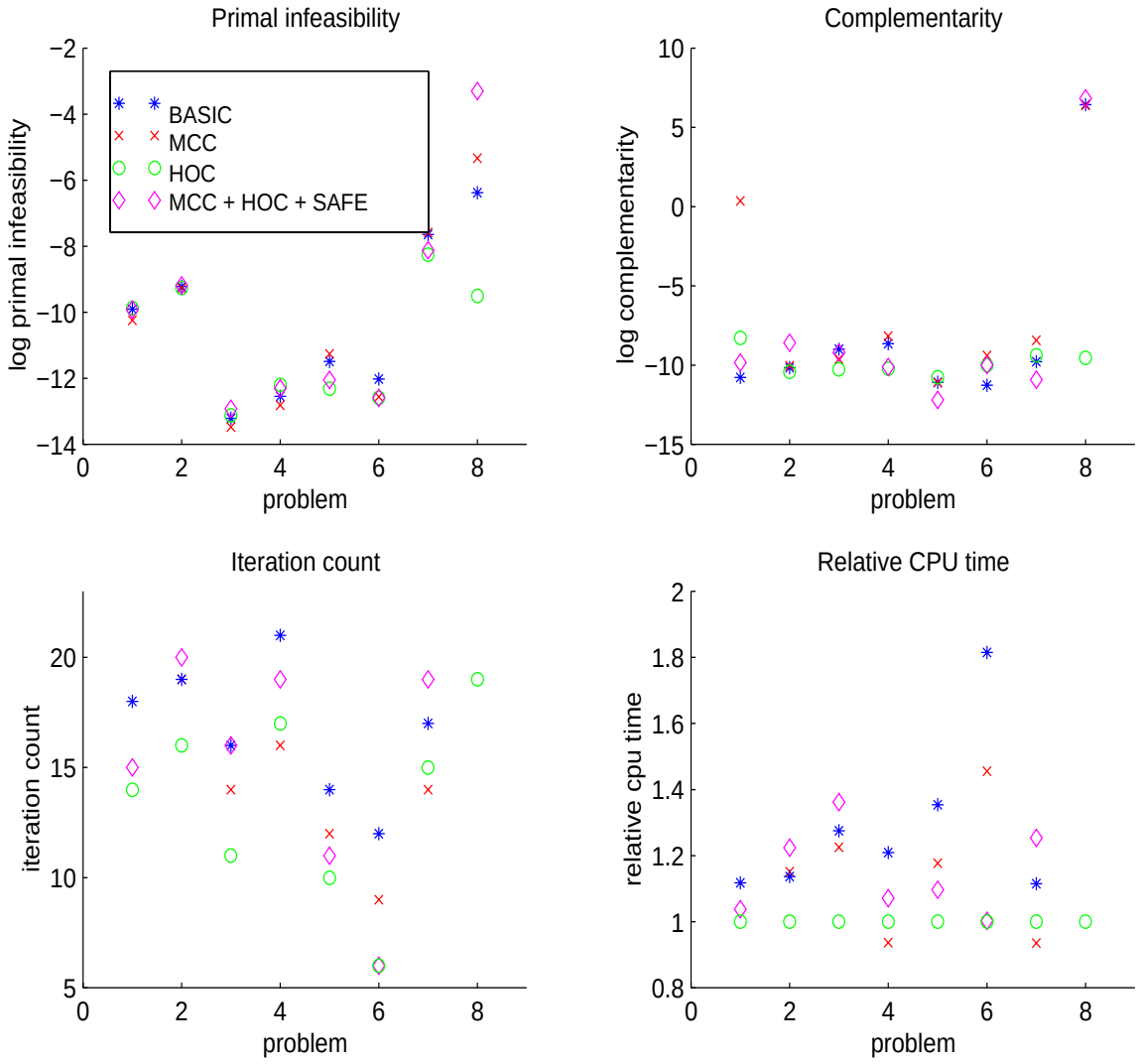

Figure 3.1: Performance of HOC, MCC and MCC+HOC+SAFE in comparison with BASIC on test set in Table 3.1.

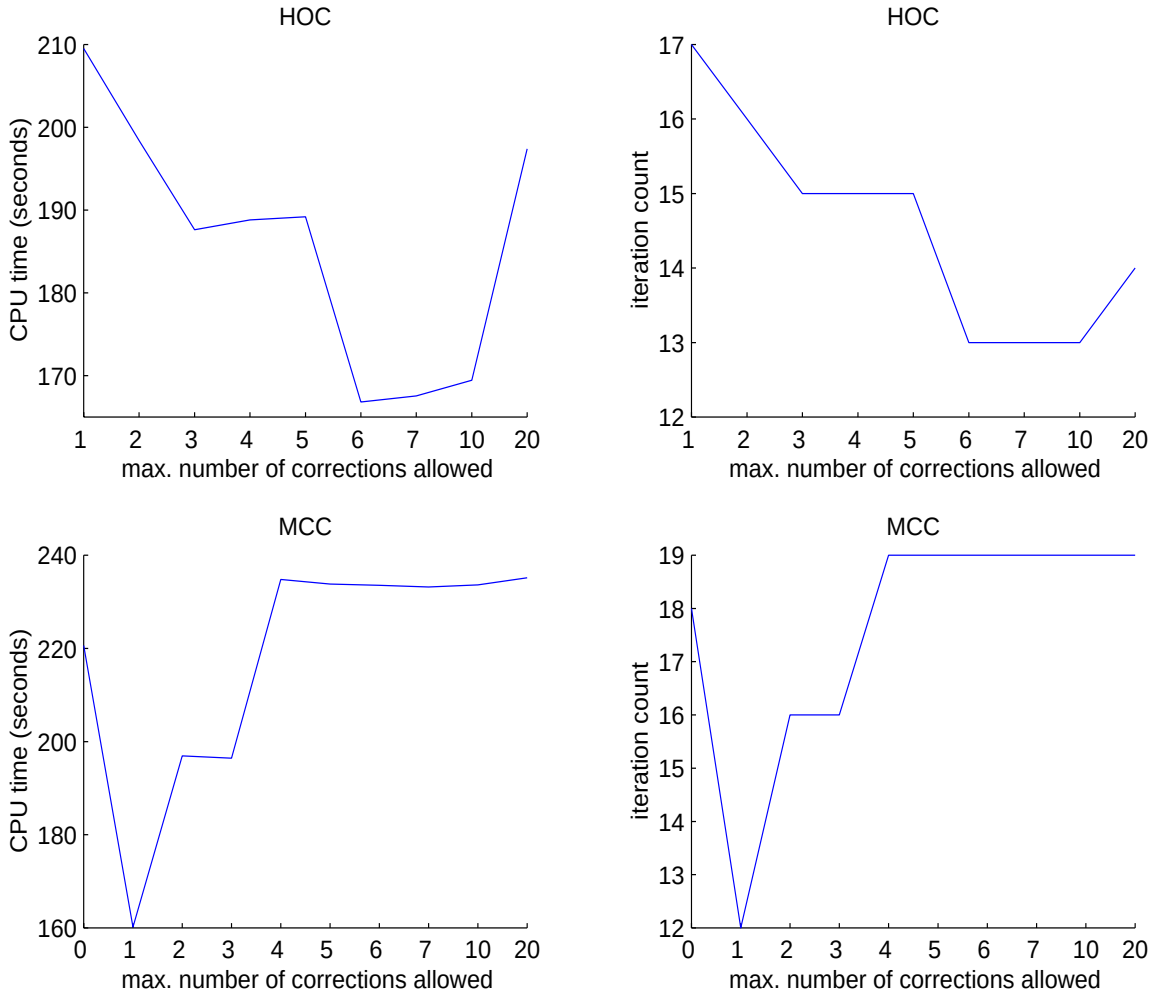

Figure 3.2: Performance of HOC and MCC as a function of the maximum number of corrections allowed per interior–point iteration. The problem selected was *truss8* shown in Table 3.1.

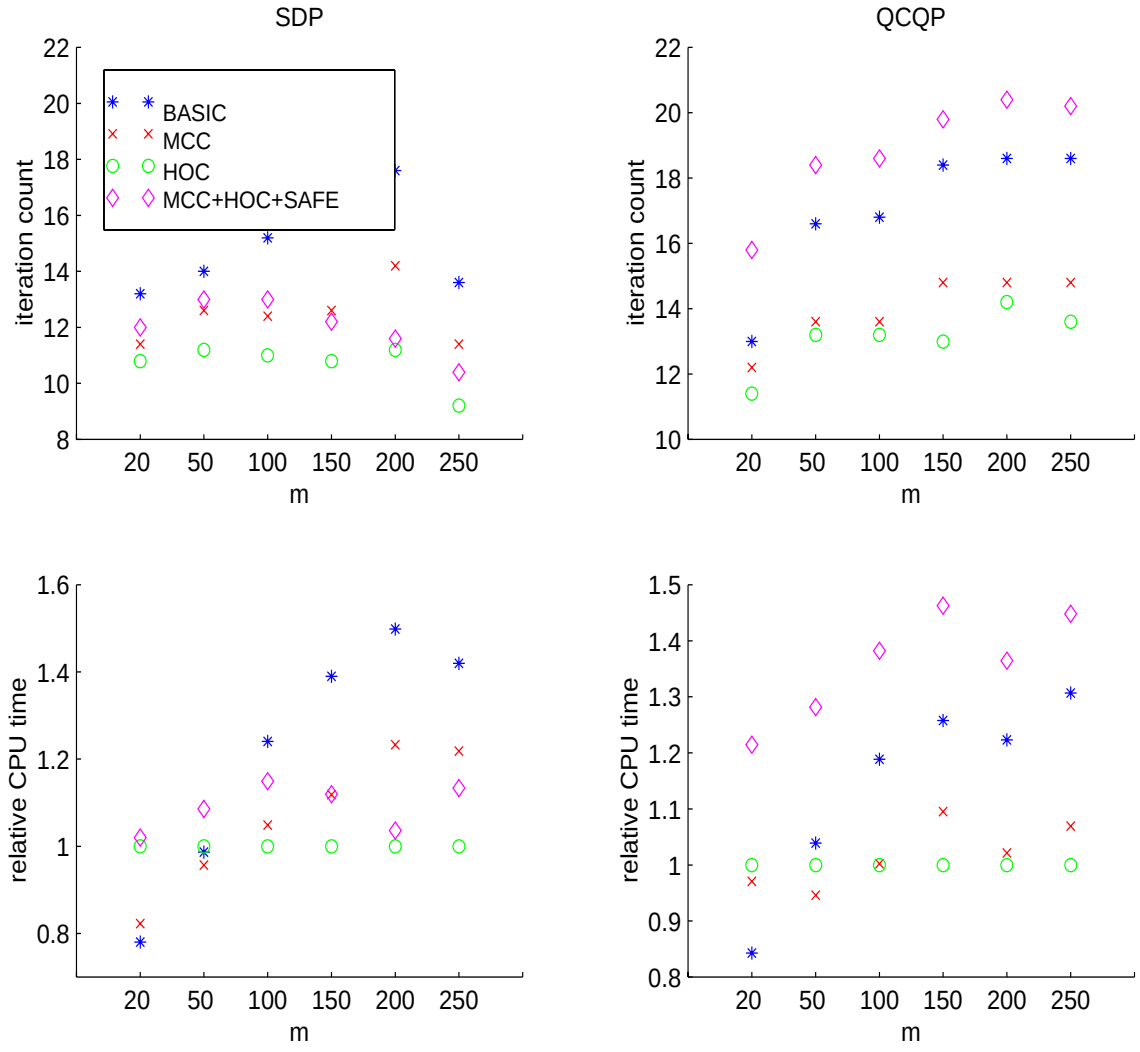

Figure 3.3: Performance of HOC and MCC as function of problem size for randomly generated, dense problems shown in Table 3.2 and Table 3.3.

#	N	m
1	$N_1 = 20$	20
2	$N_1 = 50$	50
3	$N_1 = 100$	100
4	$N_1 = 150$	150
5	$N_1 = 200$	200
6	$N_1 = 250$	250

Table 3.2: Randomly generated dense SDP's used to study performance of HOC and MCC as a function of problem size.

#	n	m
1	$n_1 = \ldots = n_{60} = 3$	20
2	$n_1 = \ldots = n_{150} = 3$	50
3	$n_1 = \ldots = n_{300} = 3$	100
4	$n_1 = \ldots = n_{450} = 3$	150
5	$n_1 = \ldots = n_{600} = 3$	200
6	$n_1 = \ldots = n_{750} = 3$	250

Table 3.3: Randomly generated dense QCQP's used to study performance of HOC and MCC as a function of problem size.

Chapter 4

Bundle Methods

This chapter is of a survey nature, and is based on Helmberg and Rendl [48], Lemaréchal and Oustry [64], and Helmberg and Oustry [47]. We describe our implementation of a first order bundle method for a special class of semidefinite programs which can be formulated as unconstrained eigenvalue optimization problems. Although this transformation technique applies to the general COP (with quadratic and linear components also) as described in Section 1.1, the present implementation, and hence also this discussion, is confined to semidefinite programming. This implementation is a part of on–going joint work with F. Oustry and M. L. Overton on bundle methods for eigenvalue optimization. The author is grateful to F. Oustry whose valuable insights have directly contributed to the implementation described here.

4.1 Introduction

By restricting $\mathbb{E} = \mathbb{S}^N$ and $\mathcal{K} = \mathbb{S}^N_+$ in the general formulation of COP in (1.1) and (1.3), we obtain a primal–dual pair of SDP. We assume that the linear operator A is given by

$$A : \mathbb{E} \to \mathbb{R}^m : x \mapsto [\langle a_1, x \rangle, \ldots, \langle a_m, x \rangle], \tag{4.1}$$

where $a_1, \ldots, a_m \in \mathbb{S}^N$, and use the symbol $\succeq$ to denote membership in the cone of positive semidefinite matrices, *i.e.* for any symmetric matrix v, $v \succeq 0$ means that v is positive semidefinite. We will assume that the Slater condition (Assumption 1.2.1) is satisfied.

Let $\overline{N} = N_1 + \ldots + N_s$. We *redefine* the eigenvalue map $\lambda(\cdot)$ of Definition 1.5.2 as follows:

$$\lambda : \mathbb{S}^N \to \mathbb{R}^{\overline{N}} : z \mapsto [\lambda_1(z), \ldots, \lambda_{\overline{N}}(z)]$$

where $\lambda_1(z) \geq \ldots \geq \lambda_{\overline{N}}(z)$. (This differs from Definition 1.5.2 as the sorting is done for the whole matrix z rather than within each block.) For $1 \leq k \leq \overline{N}$, we denote by $E_k(z)$, a subspace containing the eigenspace of $\lambda_1(z), \ldots, \lambda_{k-1}(z)$, and contained in the eigenspace of $\lambda_1(z), \ldots, \lambda_{k-1}(z)$. Note that $E_k(z)$ is not uniquely defined if $\lambda_k(z) = \lambda_{k+1}(z)$. We use

$$\lambda_{\max}(z) \stackrel{\Delta}{=} \max_{1 \leq i \leq \overline{N}} \lambda_i(z)$$

to denote the maximum eigenvalue map, and $E_{\max}(z)$ to denote the eigenspace corresponding to $\lambda_{\max}(z)$. Observe that although $\lambda_{\max}(z) = \lambda_1(z)$, $E_{\max}(z) \neq E_1(z)$, unless the largest eigenvalue is simple. Of course, $E_{\max}(z) \supseteq E_1(z)$ always.

The class of SDP's we are interested in are those that satisfy the following assumption.

Assumption 4.1.1 (Constant Trace Condition)
All primal feasible x satisfy $\mathrm{tr}(x) = a$, where $a > 0$ is a known constant.

SDP's arising from combinatorial optimization typically satisfy this assumption (see (1.30)), as do those in the quantum mechanics application described in Chapter 5.

Helmberg and Rendl [48] observed that an SDP satisfying the constant trace condition is equivalent to an unconstrained eigenvalue optimization problem, and hence is amenable to nonsmooth algorithms such as the *bundle method*. To see this, let us add the constant trace condition as a redundant constraint to get the equivalent primal SDP

$$\inf_{x\in\mathbb{E}} \quad \langle\, c, x \,\rangle \quad \text{s.t.} \quad Ax = b\,; \quad \mathrm{tr}(x) = a\,; \quad x \succeq 0, \tag{4.2}$$

and its dual

$$\sup_{(y_0,y)\in\mathbb{R}\times\mathbb{R}^m} \quad \langle\, b, y \,\rangle + a y_0 \quad \text{s.t.} \quad A^*y + y_0 I + z' = c; \quad z' \succeq 0. \tag{4.3}$$

Since $\mathrm{tr}(x) = a > 0$, the optimal primal solution $x \neq 0$, which by complementarity of eigenvalues (see Proposition 1.3.1), implies that $\lambda_{\min}(z') = 0$. Rewriting this as $\lambda_{\max}(-z') = 0$, we get from the dual equality constraint that

$$0 = \lambda_{\max}(A^*y - c + y_0 I) = \lambda_{\max}(A^*y - c) + y_0,$$

hence $y_0 = -\lambda_{\max}(A^*y - c)$. Substituting for y_0 in the dual objective, and rewriting the sup as an inf, we get

$$\inf_{y\in\mathbb{R}^m} h(y) \tag{4.4}$$

$$h(y) \stackrel{\Delta}{=} a\lambda_{\max}(A^*y - c) - \langle\, b, y \,\rangle \tag{4.5}$$

which is an unconstrained eigenvalue optimization problem, where the objective function $h(y)$ is nonsmooth but convex. Hence $y \in \mathbb{R}^m$ is a solution to (4.4) if and only if

$$0 \in \partial h(y)$$

where $\partial h(y)$ is the subdifferential of f at y (see Section 4.2).

In the sequel, we assume without any loss of generality that $a = 1$.

Eigenvalue optimization problems such as (4.4) have been extensively studied in the literature [21, 85, 86, 88]; see [67] for a tutorial survey. Details on subgradient bundle methods can be found in [50, 56, 57, 98], which develop the original pioneering work of Lemaréchal [61, 62] (see also [63]). Bundle methods specifically tailored for eigenvalue optimization are discussed in [47, 48, 64, 83, 84].

The reason to consider a bundle method is that, being a first order method, it can quickly provide solutions of low accuracy to SDP's which are so large that there is no hope of solving them with a primal–dual interior–point algorithm. The latter class of algorithms typically involve factorizing a matrix whose order m is the number of dual variables (see Section 3.3.1 and Appendix A). For large m, even storing this matrix may not be possible. On the other hand, each iteration of a bundle method is cheap, both with respect to computational time and memory requirement. However, we note that some large SDP's arising from sparse graph problems have recently been solved with a pure *dual* (as opposed to primal–dual) potential reduction interior–point algorithm [13].

4.2 Proximal Bundle Methods

In this section, we describe the proximal bundle method [57] for a finite–valued convex function $h : \mathbb{R}^m \to \mathbb{R}$. For such a function, we begin with the definition and the continuity properties of its ϵ–subdifferential, a fundamental object in bundle algorithms. Subsequently, we specialize the definitions to the function h in (4.5).

Definition 4.2.1 (ϵ–subdifferential [50, Vol.II, p.92])
Given $\epsilon \geq 0$, an element $v \in \mathbb{R}^m$ is said to be an ϵ–*subgradient* of the convex function h at $x \in \mathbb{R}^m$ if

$$f(x') \geq f(x) + \langle\, v, x' - x \,\rangle - \epsilon \quad \forall x' \in \mathbb{R}^m.$$

The ϵ–*subdifferential* is the multifunction $\partial_\epsilon h : \mathbb{R}^m \rightrightarrows \mathbb{R}^m$ that maps $x \in \mathbb{R}^m$ to the set of ϵ–subgradients of h at x. When $\epsilon = 0$, the ϵ–subdifferential is simply called the *subdifferential*, and the subscript is dropped.[1]

[1] Not to be confused with the earlier usage $\partial\mathcal{K}$, the boundary of the cone $\mathcal{K}$.

While the subdifferential ∂h is generally only outer semicontinuous [50, Vol.I, p.283], the ϵ–subdifferential is continuous for all $\epsilon > 0$ [50, Cor.XI.4.1.5]. Furthermore, a subgradient at one point in the space can be used to approximate an ϵ–subgradient at another point. This property makes the ϵ–subdifferential a very useful object in bundle algorithms (see [50, XI.4]).

LEMMA 4.2.1 (TRANSPORT OF SUBGRADIENTS [15])
Let $h : \mathbb{R}^k \to \mathbb{R}$ be a closed, convex function, $y \in \mathbb{R}^k$ and $\epsilon \geq 0$. For any $\eta > 0$ and $v \in \partial_\epsilon h(y)$, there exist $y_\eta \in B(y, \eta)$ and $v_\eta \in \partial h(y_\eta)$ such that $\|v_\eta - v\| \leq \epsilon/\eta$.

Let $W = \left\{ w \ : \ w \in \mathbb{S}^N_+, \ \mathrm{tr}(w) = 1 \right\}$. For any $z \in \mathbb{S}^N$, Rayleigh's formula yields

$$\lambda_{\max}(z) = \max_{\|p\|_2 = 1} \langle\, p, zp \,\rangle = \max_{w = pp^*} \langle\, z, w \,\rangle = \max_{w \in \mathbb{S}^N_+;\, \mathrm{tr}(w) = 1} \langle\, z, w \,\rangle = \max_{w \in W} \langle\, z, w \,\rangle\,,$$

which clearly shows $\lambda_{\max}(\cdot)$ to be the support function [50] of the compact, convex set W. Using this, the following explicit expression can be derived for $\partial_\epsilon \lambda_{\max}(z)$ [34, 51, 86, 87]:

$$\partial_\epsilon \lambda_{\max}(z) = \{ w \in W \ : \ \langle\, w, z \,\rangle \geq \lambda_{\max}(z) - \epsilon \} \quad (\epsilon \geq 0). \tag{4.6}$$

When $\epsilon = 0$, (4.6) reduces to

$$\partial \lambda_{\max}(z) = \mathrm{conv}\ \{ pp^* \ : \ \|p\|_2 = 1, \ \ p \text{ is an eigenvector of } z \text{ corresponding to } \lambda_{\max}(z) \}\,.$$

Equivalently, we can write

$$\partial \lambda_{\max}(z) = \{ pvp^* \ : \ \mathrm{tr}(v) = 1 \}\,, \tag{4.7}$$

where the columns of p are an orthonormal basis for $E_{\max}(z)$. Setting $h(y) = a\lambda_{\max}(A^*y - c) - \langle\, b, y \,\rangle$ and using a chain rule (see, for instance, [17, 50]), we get

$$\partial_\epsilon h(y) = \{ Aw - b \in \mathbb{R}^m \ : \ w \in \partial_\epsilon \lambda_{\max}(A^*y - c) \} \quad (\epsilon \geq 0).$$

Therefore, for any inner approximation $\widehat{W} \subseteq W$, the function

$$\hat{\phi}(y) \triangleq \left(\max_{w \in \widehat{W}} \langle\, A^*y - c, w \,\rangle \right) - \langle\, b, y \,\rangle \tag{4.8}$$

minorizes $\lambda_{\max}(A^*y - c) - \langle\, b, y \,\rangle$.

We warn the reader of an *abus de language* in the sequel. We use the term "subgradient" (dropping the prefix ϵ) both for a matrix $w \in W$ and the vector Aw, with the tacit understanding that it is actually $Aw - b$ that is an ϵ–subgradient of h. Similarly, we will use the term "eigenvectors" (dropping the prefix ϵ) for the columns of the matrix p in (4.7), although p might contain "ϵ–eigenvectors" rather than eigenvectors of $A^*y - c$, *i.e.* if k is the multiplicity of $\lambda_{\max}(z)$, then p would satisfy, for some $\epsilon > 0$, $\langle\, pp^*, z \,\rangle \geq k\lambda_{\max}(z) - \epsilon$, rather than $\langle\, pp^*, z \,\rangle = k\lambda_{\max}(z)$. The reader must bear in mind that the ϵ prefix is implicit.

Now, the basic idea behind the proximal bundle method is to minimize at each iteration k, a model of the form

$$\phi^k(y) \triangleq \left(\max_{w \in W^k} \langle\, A^*y - c, w \,\rangle \right) - \langle\, b, y \,\rangle + \frac{\rho^k}{2} \left\| y^k - y \right\|_2^2. \tag{4.9}$$

The first two terms in the right hand side of (4.9) is a subgradient model minorizing h, and is based on the chosen inner approximation $W^k \subseteq W$. The third term in the right hand side of (4.9) penalizes (with a penalty parameter $\rho^k > 0$), the distance between the current iterate y^k and the new minimizer. Hence the subproblem at each iteration assumes the form

$$\min_{y \in \mathbb{R}^m} \left(\max_{w \in W^k} \langle\, A^*y - c, w \,\rangle \right) - \langle\, b, y \,\rangle + \frac{\rho^k}{2} \left\| y^k - y \right\|_2^2. \tag{4.10}$$

Let $(\overline{y}, \overline{w})$ be the minimizer of (4.10). If $\overline{y}$ produces "sufficient decrease" in the objective function in (4.4), we perform a *serious step* by setting $y^{k+1} = \overline{y}$. Otherwise, we perform a *null step* by setting $y^{k+1} = y^k$. In either case, a new approximation W^{k+1} to W is chosen to contain $\overline{w}$ and at least one new subgradient from $\partial h(\overline{y})$, resulting in a new model $\phi^{k+1}(y)$. In general, W^{k+1} would contain subgradients computed at earlier iterates $y^i \quad (i = 0, \ldots, k)$, which by Lemma 4.2.1, may be viewed as ϵ–subgradients at the current point y^{k+1}, for a suitable value of ϵ. We then proceed to the next iteration.

The algorithm may be stated as follows. The original algorithm and its proof of global convergence are due to Kiwiel [57]. Its specialization to the $\lambda_{\max}(\cdot)$ function, using a nontraditional "semidefinite model" for W^{k+1} (explained in Section 4.3) is due to Helmberg and Rendl [48]. A unifying view of various bundle methods for eigenvalue optimization is developed in Lemaréchal and Oustry [64], and in Helmberg and Oustry [47].

Algorithm 4.2.1 (Proximal Bundle Method)
Input The following are given:

$y^0 \in \mathbb{R}^m$**:** Initial guess.

$\delta > 0$**:** Termination threshold.

$\gamma \in (0, \frac{1}{2})$**:** Sufficient decrease parameter to determines which iterations are *serious steps* and which are *null steps*.

$\rho^0 > \rho^{\min} > 0$**:** Penalty for the proximal term.

Step 1 (Initialization) Set $k = 0$, compute $g^0 \in \partial h(y^0)$.

Step 2 (Model Minimization) Solve the subproblem (4.10) to obtain the solution $(\overline{y}, \overline{w})$.

Step 3 (Termination) If $h(y^k) - \hat{\phi}^k(\overline{y}) \leq \delta(\mid h(y^k) \mid +1)$, then return y^k as an approximate solution and terminate.

Step 4 (Step Quality) If

$$h(\overline{y}) \leq h(y^k) - \gamma \left(h(y^k) - \hat{\phi}^k(\overline{y}) \right), \tag{4.11}$$

set $y^{k+1} = \overline{y}$ (serious step). Otherwise, set $y^{k+1} = y^k$ (null step).

Step 5 (Update) Compute at least one new subgradient $w \in \partial \lambda_{\max}(A^*\overline{y} - c)$. Choose $W^{k+1} \subseteq W$ such that $\{\overline{w}, w\} \subseteq W^{k+1}$. If (4.11) was satisfied, choose $\rho^{k+1} \in [\rho^{\min}, \rho^k]$. Otherwise, choose $\rho^{k+1} \in [\rho^k, 10\rho^k]$. Set $k = k + 1$, and go back to Step 2 (model minimization).

Ensuring that $\{\overline{w}, w\} \subseteq W^{k+1}$ in Step 5 guarantees that after a null step, the objective value in (4.10) will increase in the next iteration, *i.e.* the model will approximate the function better. The sequence of objective values $h(y^k) \quad (k = 0, 1, \ldots)$ monotonically decreases to $\inf_{y \in \mathbb{R}^m} h(y)$. The sequence of iterates $y^k \quad (k = 0, 1, \ldots)$ converges to a minimizer of h, if one exists, and is unbounded otherwise [56, 57].

4.3 The COP Subproblem

We now give details about the model minimization step (Step 2), and the resulting subproblem.

Let $(\overline{y}, \overline{w})$ be the solution to (4.10). Interchanging the min and the max [95, Corollary 37.3.2] in (4.10), we get

$$\max_{w \in W^k} \min_{y \in \mathbb{R}^m} \langle A^*y - c, w \rangle - \langle b, y \rangle + \frac{\rho^k}{2} \left\| y^k - y \right\|_2^2, \tag{4.12}$$

where the inner minimization is unconstrained, and hence whose minimizer $\overline{y}$ must satisfy $0 = Aw - b + \rho^k(y^k - \overline{y})$, or equivalently,

$$\overline{y} = y^k + \frac{1}{\rho^k} (Aw - b). \tag{4.13}$$

Substituting this expression for $\overline{y}$ back into the objective of (4.12), we get

$$\min_{w \in W^{k+1}} \left\langle c - A^* y^k, w \right\rangle + \frac{1}{2\rho^k} \left\langle Aw - b, Aw - b \right\rangle. \tag{4.14}$$

This is the subproblem to be solved at each iteration.

Let $w_i \in W \;\; (i = 1, \ldots, l)$ be subgradients accumulated from earlier iterations, $z = A^* y^k - c$, and p, an orthonormal basis for $E_{\overline{r}}(z) \;\; (\overline{r} \geq 1)$. Let $r_j \;\; (j = 1, \ldots, s)$ be the number of the $\overline{r}$ largest eigenvalues $\lambda_1(z), \ldots, \lambda_{\overline{r}}(z)$ that occur within the j^{th} block of z, and let $r = [r_1, \ldots, r_s]$. We choose, for $\alpha = [\alpha_1, \ldots, \alpha_l]$, the set

$$W^{k+1} \triangleq \left\{ \sum_{i=1}^{l} \alpha_i w_i + p\, v\, p^* \;\; : \;\; \sum_{i=1}^{l} \alpha_i + \operatorname{tr}(v) = 1, \;\; \alpha \in \mathbb{R}^l_+, \;\; v \in \mathbb{S}^r_+ \right\},$$

which is contained in W, to define the subgradient model at the $(k+1)^{\text{th}}$ iteration. Although it suffices to take $\overline{r} = 1$, the use of multiple eigenvectors ($\overline{r} > 1$) improves the model by including the "semidefinite" component $p\, v\, p^*$. This is due to [48], and is different from traditional bundle methods which typically use a polyhedral representation for W^{k+1}.

Further, let us write $\alpha = [\alpha_1, \ldots, \alpha_l]$. Now (4.14) reduces to

$$\begin{array}{ll} \min\limits_{(\alpha, v) \in \mathbb{R}^l \times \mathbb{S}^r} & \beta \\ \quad \text{s.t.} & \left\langle c - A^* y^k, w \right\rangle + \dfrac{1}{2\rho^k} \left\langle Aw - b, Aw - b \right\rangle - \beta \leq 0 \\ & w = \displaystyle\sum_{i=1}^{l} \alpha_i w_i + p\, v\, p^* \\ & \displaystyle\sum_{i=1}^{l} \alpha_i + \operatorname{tr}(v) = 1 \\ & \alpha \geq 0; \;\; v \succeq 0. \end{array}$$

Eliminating w, we can write this problem as

$$\begin{array}{ll} \min\limits_{(\alpha, v) \in \mathbb{R}^l \times \mathbb{S}^r} & \beta \\ \quad \text{s.t.} & \dfrac{1}{2} \left\langle [\alpha, \mathbf{vec}(v)], Q\, [\alpha, \mathbf{vec}(v)] \right\rangle + \left\langle u, [\alpha, \mathbf{vec}(v)] \right\rangle - \beta \leq 0 \\ & \displaystyle\sum_{i=1}^{l} \alpha_i + \operatorname{tr}(v) = 1 \\ & \alpha \geq 0; \;\; v \succeq 0, \end{array} \tag{4.15}$$

where Q is a positive semidefinite matrix of order $l + \overline{r}(\overline{r}+1)/2$, and u is a vector of length $l + \overline{r}(\overline{r}+1)/2$. The exact expressions for Q and u will be given in Section 4.4.1. Upon further rewriting, (4.15) can be cast in the standard form of a COP over a direct sum of semidefinite cones, quadratic cones and the nonnegative orthant (see Chapter 1).

4.4 Computational Issues

We discuss computational issues relating to the solution of the COP subproblem, updates of the bundle, and the exploitation of sparsity. Our implementation is partly in Matlab (bundle management, eigenvalue routines) and partly in C (the operators $x \mapsto Ax$ and $y \mapsto c - A^* y$, computation of the data for the subproblem *etc.*).

4.4.1 Solving The Subproblem

We use *SeQuL* (see Chapter 3) to solve the subproblem in (4.15). For convenience, let us drop the superscript k on y, ρ and p. Using the **vec** isometry of Section 3.2, the matrix Q and the vector $u = [u_1, \ldots, u_l, \tilde{u}]$ in the convex quadratic constraint in (4.15) can be written as

$$u_i = \left\langle\, y - \frac{1}{\rho} b, Aw_i \,\right\rangle - \langle\, c, w_i \,\rangle \quad (i = 1, \ldots, l), \tag{4.16}$$

$$\tilde{u} = \mathbf{vec}(p^*(A^*(y - \frac{1}{\rho} b) - c)p)], \text{ and} \tag{4.17}$$

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix}, \text{ with} \tag{4.18}$$

$$Q_{11} = \frac{1}{\rho} G^* G, \tag{4.19}$$

$$Q_{12} = \frac{1}{\rho} \begin{bmatrix} (\mathbf{vec}(p^*\,(A^* g_i)\,p))^* \\ \vdots \\ (\mathbf{vec}(p^*\,(A^* g_l)\,p))^* \end{bmatrix}, \text{ and} \tag{4.20}$$

$$Q_{22} = \frac{1}{\rho} \sum_{i=1}^{m} \mathbf{vec}(p^*\, a_i\, p)\, (\mathbf{vec}(p^*\, a_i\, p))^*. \tag{4.21}$$

The convex quadratic constraint in (4.15) can be written in the standard quadratic cone format using the Cholesky factorization of Q (see [79, p.221]). Note that this factorization is cheap because the dimension $l + \overline{r}(\overline{r}+1)/2$ of Q depends only on the size of the model parameters l and $\overline{r}$, which are generally small constants compared to m and d_S. In fact, even solving the COP is relatively inexpensive compared to the cost of one bundle iteration. The dominant cost of each iteration lies in computing the new eigenvectors to be stored in p. However, when m becomes comparable to d_S (the dimension of $\mathbb{S}^N$), forming Q can be the most expensive part of each bundle iteration (cf. Section 3.3.1 and Appendix A). This is the case with the application from electronic structure calculations described in Chapter 5. See Appendix B for estimates on operation counts.

4.4.2 Managing the bundle

From (4.17) – (4.21), it is clear that there is no need to explicitly store the matrices w_i. It suffices to store the vectors $g_i = Aw_i$ and the scalars $t_i = \langle\, c, w_i \,\rangle \quad (i = 1, \ldots, l)$. The bundle then consists of the matrix $m \times l$ matrix $G = [g_1 \ \ldots \ g_l]$, the vector $t = [t_1 \ \ldots \ t_l]$ (together called the *polyhedral component*), and the matrix of eigenvectors p (called the *semidefinite component*), whose j^{th} block ($j = 1, \ldots, s$) is of size $N_j \times r_j$. Note that r_j would be zero if none of the $\overline{r}$ largest eigenvalues occurred in the j^{th}. Since a significant amount of computational time may be spent in conjugations by p (see Section 4.4.3), it is important to retain this block structure in p.

At a given iteration, suppose the "bundle is full", *i.e.* the number of columns in G and/or the number of eigenvectors in p have reached their allowable limit of, say, l and $\overline{r}$ respectively. In order to make space for the current model minimizer and/or the new eigenvectors of $A^*\overline{y} - c$, some columns in G and/or p can be "lumped" with a strategy called *aggregation*. This concept was introduced by Kiwiel [56], who has shown that aggregation can be performed without impeding global convergence. We describe two possible strategies. For vectors (matrices), the notation $\cdot|_J$ denotes the restriction of the vector (matrix) to those components (columns) in the index set J.

Polyhedral aggregation Let $g_i = Aw_i$, $t_i = \langle\, c, w_i \,\rangle \quad (i = 1, \ldots, l)$ be the subgradient information used in the current model. Assume that $(\overline{\alpha}, \overline{v})$ is the solution to the subproblem (4.15), and let the corresponding subgradient information be $\overline{g} = \sum_{i=1}^{l} \overline{\alpha}_i g_i + A(p\overline{v}p^*)$, $\overline{t} = \langle\, \overline{\alpha}, t \,\rangle + \langle\, c, p\overline{v}p^* \,\rangle$. Choosing a threshold $\eta \in (0, 1)$, partition the index set $J = \{1, \ldots, l\}$ as

$$P_{\text{zero}} = \{i \in J \ : \ \alpha_i = 0\}$$
$$P_{\text{small}} = \{i \in J \ : \ 0 < \alpha_i \leq \eta\}$$

$$P_{\text{large}} = \{i \in J \ : \ \alpha_i > \eta\}\,,$$

and perform the aggregation as follows:

$$\begin{aligned}
g_{\text{agg}} &= \frac{\sum_{i \in P_{\text{small}}} \overline{\alpha}_i g_i}{\sum_{i \in P_{\text{small}}} \overline{\alpha}_i} \\
t_{\text{agg}} &= \frac{\sum_{i \in P_{\text{small}}} \overline{\alpha}_i t_i}{\sum_{i \in P_{\text{small}}} \alpha_i} \\
G &\leftarrow [G|_{P_{\text{large}}} \ \ \overline{g} \ \ g_{\text{agg}}] \\
t &\leftarrow [t|_{P_{\text{large}}} \ \ \overline{t} \ \ t_{\text{agg}}].
\end{aligned}$$

The vectors stored in p may now be discarded, and be replaced by eigenvectors corresponding to a few of the largest eigenvalues of $A^*\overline{y} - c$.

Polyhedral + Semidefinite aggregation Except for the fact that we use multiple polyhedral pieces $g_i \ \ (i = 1, \ldots, l)$, this strategy is similar to the one described in [48].

Let $\overline{v} = q\,\mathrm{Diag}(\omega)\,q^*$ be a spectral decomposition of $\overline{v}$. Choose $\eta \in (0, 1)$ and partition the index sets $J = \{1, \ldots, l\}$ and $K = \{1, \ldots, \overline{r}\}$ as

$$\begin{aligned}
P^{\text{poly}}_{\text{zero}} &= \{i \in J \ : \ \alpha_i = 0\} & P^{\text{sd}}_{\text{zero}} &= \{i \in K \ : \ \omega_i = 0\} \\
P^{\text{poly}}_{\text{small}} &= \{i \in J \ : \ 0 < \alpha_i \leq \eta\} & P^{\text{sd}}_{\text{small}} &= \{i \in K \ : \ 0 < \omega_i \leq \eta\} \\
P^{\text{poly}}_{\text{large}} &= \{i \in J \ : \ \alpha_i > \eta\} & P^{\text{sd}}_{\text{large}} &= \{i \in K \ : \ \omega_i > \eta\}\,,
\end{aligned}$$

and perform the aggregation as follows:

$$\begin{aligned}
g_{\text{agg}} &= \frac{\sum_{i \in P^{\text{poly}}_{\text{small}}} \overline{\alpha}_i g_i + A(p\,q|_{P^{\text{sd}}_{\text{small}}}\,\mathrm{Diag}(\omega|_{P^{\text{sd}}_{\text{small}}})\,(q|_{P^{\text{sd}}_{\text{small}}})^*\,p^*)}{\sum_{i \in P^{\text{poly}}_{\text{small}}} \overline{\alpha}_i + \sum_{i \in P^{\text{sd}}_{\text{small}}} \omega_i} \\
t_{\text{agg}} &= \frac{\sum_{i \in P^{\text{poly}}_{\text{small}}} \overline{\alpha}_i t_i + \left\langle c,\ p\,q|_{P^{\text{sd}}_{\text{small}}}\,\mathrm{Diag}(\omega|_{P^{\text{sd}}_{\text{small}}})\,(q|_{P^{\text{sd}}_{\text{small}}})^*\,p^* \right\rangle}{\sum_{i \in P^{\text{poly}}_{\text{small}}} \overline{\alpha}_i + \sum_{i \in P^{\text{sd}}_{\text{small}}} \omega_i} \\
G &\leftarrow [G|_{P^{\text{poly}}_{\text{large}}} \ \ \overline{g} \ \ g_{\text{agg}}] \\
t &\leftarrow [t|_{P^{\text{poly}}_{\text{large}}} \ \ \overline{t} \ \ t_{\text{agg}}] \\
p &\leftarrow p\,q|_{P^{\text{sd}}_{\text{large}}}
\end{aligned}$$

This reduces the number of columns in p, and makes room for some of the eigenvectors corresponding to a few of the largest eigenvalues of $A^*\overline{y} - c$. As the new eigenvectors added will generally not be orthogonal to the span of the existing eigenvectors in p, it is necessary to orthogonalize the columns of p.

Note that pure polyhedral aggregation separates the old subgradients from previous iterations ("global information") in G from the subgradients at the current point ("local information") in p. On the contrary, when semidefinite aggregation is included, p (upon updating) becomes $p\,q|_{P^{\text{sd}}_{\text{large}}}$, and hence contains some global information too. Since global convergence of the algorithm relies only on retaining an aggregate subgradient, and adding at least one new subgradient at each iteration, the aggregation strategy and the choice of η are just additional flexibilities in the algorithm. Our implementation requires the upper bounds $l_{\max} \geq 1$ and $r_{\max} \geq 1$ on l and $\overline{r}$ respectively, and the number of new eigenvectors to be added to p at each iteration, to be specified. Although the memory consumption of the algorithm varies from one iteration to another, the numbers $l_{\max}$ and $r_{\max}$ dictate a uniform upper bound on the dominant storage requirement of the algorithm.

4.4.3 Sparsity

Most large scale applications, *e.g.* SDP relaxations of combinatorial optimization problems involving sparse graphs, result in SDP's with sparse data matrices $c, a_i \ \ (i = 1, \ldots, m)$, and it is crucial to effectively exploit this sparsity.

Let $p(j)$ denote the j^{th} block of p. If $a_i(j)$ contains only $o(N_j + N_j^2/r_j)$ nonzeros, then much effort expended in computing Q_{12} (see (4.20)) and Q_{22} (see (4.21)) can be saved in the orthogonal conjugations $p(j)^* \, a_i(j) \, p(j)$ by computing them as

$$p(j)^* \, a_i(j) \, p(j) = \sum_{k,l} a_i(j)(k,l) \, p_k \, p_l^* \tag{4.22}$$

where $a_i(j)(k,l)$ is the (k,l) entry of $a_i(j)$, p_k and p_l are the k^{th} and l^{th} columns of $p(j)^*$ respectively, and the summation ranges over the nonzero entries of $a_i(j)$.

The COP subproblem itself has sparse semidefinite and linear components, but the quadratic component is generally dense (see (4.19) – (4.21)). However, it is not crucial to exploit sparsity at the level of the subproblem, as solving it is relatively inexpensive compared to the cost of one full bundle iteration (see Appendix B for details).

If the $c, a_i \ \ (i = 1, \ldots, m)$ are have sparsity patterns such that the matrix–vector multiplication with the matrix $A^*y^k - c$ is cheap (as is the case in many applications), then a Lanczos routine is used to determine a few of the the largest eigenvalues of $A^*y^k - c$ and the corresponding eigenvectors. For small matrices (say, smaller than 400×400), it is faster to compute a full spectral decomposition of $A^*y^k - c$ with the QR algorithm. Our implementation switches between the Matlab routines `eigs` (Lanczos) and `eig` (QR) depending on the size of the matrix.

4.5 Numerical Results

In this section, we illustrate the behavior of the algorithm on a moderate sized and randomly generated, sparse problem with $N = [30 \ \ 900]$ and $m = 1000$. (Some larger problems arising from electronic structure calculations are solved in Section 5.4.) The algorithm was executed on a SUN Ultra Sparc I with a 143 MHz CPU and 64 MB of memory.

We start with a randomly generated point as an initial guess. We set $\gamma = 0.2$, $\eta = 0.1$, $l_{\max} = r_{\max} = 20$. In each iteration, we add 5 new eigenvectors to p. We terminate if, for $\epsilon = 10^{-3}$, we find an ϵ–subgradient of norm less than 10^{-3}, or if 50 iterations are performed, whichever occurs earlier.

Table 4.1 tabulates the serious steps in a typical run of the algorithm. We use polyhedral aggregation when necessary. Despite $l_{\max}$ and $r_{\max}$ being 20, the algorithm did not use more than $l = 5$ polyhedral pieces and $\overline{r} = 10$ eigenvectors in p in any iteration. The algorithm can make quick progress initially and get into a vicinity of the solution, but after this stage, progress is very slow. About 93% of the time is spent in eigenvalue/eigenvector calculation, and about 6% of the time is spent in forming the matrix Q for the subproblem. Hence the cost of solving the subproblem is negligible.

Iter	$\|\overline{g}\|$	ϵ	obj	cpu
0	—	—	7.578e+00	0
1	1.9e+00	1.0e+00	4.304e+00	24
2	1.5e+00	9.2e–01	3.341e+00	1:04
3	3.5e–01	1.3e+00	1.802e+00	1:54
4	2.5e–01	1.0e–01	1.551e+00	2:42
6	2.2e–01	2.4e–01	1.222e+00	4:07
8	1.7e–01	1.0e–01	1.072e+00	5:30
10	1.1e–01	7.0e–02	1.029e+00	6:37
12	6.4e–02	4.5e–02	9.959e–01	7:57
14	3.9e–02	1.3e–02	9.900e–01	9:19
17	2.6e–02	3.2e–03	9.888e–01	11:32
19	2.2e–02	6.3e–04	9.884e–01	13:03
20	1.6e–02	9.4e–04	9.873e–01	13:47
22	1.3e–02	7.7e–04	9.866e–01	15:28

Continued from previous page ...				
Iter	$\|\overline{g}\|$	ϵ	obj	cpu
24	1.2e–02	5.7e–04	9.860e–01	17:07
26	1.0e–02	4.5e–04	9.855e–01	18:46
28	9.6e–03	3.1e–04	9.851e–01	20:21
29	9.0e–03	2.3e–04	9.847e–01	21:12
30	8.8e–03	5.8e–04	9.839e–01	21:58
32	8.0e–03	5.2e–04	9.837e–01	23:42
34	8.2e–03	4.1e–04	9.834e–01	25:24
35	8.0e–03	3.3e–04	9.832e–01	26:09
36	7.7e–03	6.4e–04	9.825e–01	26:54
37	7.7e–03	3.5e–04	9.823e–01	27:48
38	7.7e–03	6.5e–04	9.816e–01	28:42
39	7.4e–03	4.2e–04	9.811e–01	29:36
40	7.4e–03	4.2e–04	9.806e–01	30:33
41	7.1e–03	4.6e–04	9.801e–01	31:36
42	7.1e–03	4.5e–04	9.798e–01	32:30
43	7.0e–03	4.9e–04	9.792e–01	33:36
45	6.6e–03	3.8e–04	9.790e–01	35:33
48	6.6e–03	1.7e–04	9.789e–01	38:52
50	6.8e–03	7.0e–05	9.789e–01	41:10

Table 4.1: Sequence of serious steps in a typical run of the bundle method.

In Table 4.2, we compare using a single polyhedral piece versus multiple polyhedral pieces in the approximation of the set W. Columns two through four correspond to $l_{\max} = 1$, whereas columns five through seven correspond to $l_{\max} = 20$. We tabulate the first and the last iterations and every serious step in between for $l_{\max} = 1$. The first column gives the iteration numbers at which the serious steps occurred. The last column is the ratio, expressed as a percentage, of the *cpu* time taken with $l_{\max} = 20$ (to execute the number of iterations stated in the first column) to the total time (50 iterations) taken with $l_{\max} = 1$. The numbers in boldface at iteration 30 and 50 show that using $l_{\max} = 20$ achieves the same decrease in objective function in 59% of the time taken by the algorithm with $l_{\max} = 1$.

	$l_{\max} = 1$			$l_{\max} = 20$			
Iter	$\|\overline{g}\|$	ϵ	obj	$\|\overline{g}\|$	ϵ	obj	% cpu
0	—	—	7.578e+00	—	—	7.578e+00	0
1	1.9e+00	1.0e+00	7.578e+00	1.9e+00	1.0e+00	4.304e+00	1
2	1.3e+00	1.0e+00	4.304e+00	1.3e+00	1.0e+00	2.842e+00	2
3	3.1e–01	9.0e–01	2.842e+00	3.1e–01	9.0e–01	1.885e+00	4
4	2.5e–01	2.3e–01	1.885e+00	2.5e–01	2.3e–01	1.473e+00	6
6	2.0e–01	2.0e–01	1.155e+00	2.0e–01	1.8e–01	1.173e+00	9
11	2.6e–01	1.2e–02	1.117e+00	8.5e–02	5.6e–02	1.028e+00	18
12	1.8e–01	3.0e–02	1.058e+00	6.6e–02	2.5e–02	1.007e+00	21
14	1.8e–01	4.7e–02	1.051e+00	4.1e–02	2.0e–02	9.991e–01	25
15	1.1e–01	2.6e–02	1.037e+00	3.6e–02	1.7e–02	9.991e–01	26
18	8.7e–02	3.6e–02	1.015e+00	3.1e–02	9.3e–03	9.946e–01	33
21	6.6e–02	2.1e–02	1.000e+00	2.7e–02	3.0e–03	9.879e–01	39
22	4.8e–02	1.0e–02	9.931e–01	1.9e–02	1.1e–03	9.879e–01	42
24	3.3e–02	4.5e–04	9.910e–01	1.7e–02	1.1e–03	9.878e–01	46
27	2.5e–02	7.4e–04	9.905e–01	1.6e–02	6.8e–04	9.869e–01	53
30	2.6e–02	2.4e–04	9.900e–01	**1.1e–02**	**8.4e–04**	**9.862e–01**	**59**
31	2.3e–02	1.3e–04	9.895e–01	1.0e–02	1.4e–03	9.862e–01	62
32	2.7e–02	3.0e–05	9.893e–01	1.0e–02	1.4e–03	9.854e–01	64
33	2.4e–02	4.6e–04	9.891e–01	1.0e–02	9.6e–04	9.854e–01	66
37	2.2e–02	1.3e–03	9.880e–01	1.4e–02	5.0e–04	9.844e–01	76
39	1.8e–02	3.6e–04	9.874e–01	8.8e–03	8.1e–04	9.843e–01	81

Continued from previous page ...							
Iter	$\|\overline{g}\|$	ϵ	obj	$\|\overline{g}\|$	ϵ	obj	% cpu
42	1.7e–02	7.8e–05	9.872e–01	9.9e–03	3.2e–04	9.838e–01	88
43	1.3e–02	7.1e–05	9.871e–01	1.2e–02	3.0e–04	9.838e–01	91
46	1.6e–02	1.7e–04	9.866e–01	7.3e–03	8.7e–04	9.837e–01	98
48	1.0e–02	9.3e–05	9.864e–01	7.4e–03	5.6e–04	9.832e–01	103
50	**5.2e–02**	**7.4e–04**	**9.864e–01**	7.4e–03	7.9e–04	9.824e–01	108

Table 4.2: Comparison of single ($l_{\max} = 1$) versus multiple ($l_{\max} > 1$) polyhedral pieces in the approximation of W.

In Table 4.3, we compare combined polyhedral and semidefinite aggregation (columns two through four), and plain polyhedral aggregation (columns five through seven). We show the first and the last iterations, and every serious step in between for the algorithm with combined polyhedral and semidefinite aggregation. The meaning of the first and last columns is similar to that in the previous table. The iterations 28 and 50 in boldface indicate that polyhedral aggregation takes about 40% of the time taken by the algorithm using polyhedral and semidefinite aggregation to reduce the objective function to the same extent.

	Polyhedral + semidefinite			Polyhedral			
Iter	$\|\overline{g}\|$	ϵ	obj	$\|\overline{g}\|$	ϵ	obj	% cpu
0	—	—	7.578e+00	—	—	7.578e+00	0
1	1.9e+00	1.0e+00	4.304e+00	1.9e+00	1.0e+00	4.304e+00	1
3	1.0e+00	1.3e+00	2.676e+00	3.5e–01	1.3e+00	1.802e+00	4
4	5.6e–01	7.8e–01	1.935e+00	2.5e–01	1.0e–01	1.551e+00	5
5	2.7e–01	1.2e–01	1.703e+00	2.2e–01	2.4e–01	1.551e+00	6
6	2.3e–01	1.4e–01	1.486e+00	2.2e–01	2.4e–01	1.222e+00	8
9	1.8e–01	2.7e–01	1.242e+00	1.3e–01	5.8e–02	1.072e+00	11
11	1.5e–01	1.1e–01	1.144e+00	7.6e–02	4.1e–02	1.029e+00	14
12	1.5e–01	6.9e–02	1.109e+00	6.4e–02	4.5e–02	9.959e–01	15
17	9.9e–02	9.6e–02	1.033e+00	2.6e–02	3.2e–03	9.888e–01	22
18	9.4e–02	2.1e–02	1.018e+00	2.3e–02	5.5e–04	9.888e–01	24
20	6.1e–02	1.7e–02	9.995e–01	1.6e–02	9.4e–04	9.873e–01	27
23	5.3e–02	7.7e–04	9.972e–01	1.2e–02	5.0e–04	9.866e–01	32
24	5.0e–02	3.2e–04	9.962e–01	1.2e–02	5.7e–04	9.860e–01	33
26	3.9e–02	2.0e–03	9.928e–01	1.0e–02	4.5e–04	9.855e–01	36
28	3.4e–02	3.6e–04	9.915e–01	**9.6e–03**	**3.1e–04**	**9.851e–01**	**40**
30	2.8e–02	2.0e–04	9.906e–01	8.8e–03	5.8e–04	9.839e–01	43
32	2.5e–02	1.9e–04	9.899e–01	8.0e–03	5.2e–04	9.837e–01	46
33	2.3e–02	1.2e–04	9.890e–01	8.6e–03	4.2e–04	9.837e–01	48
35	2.0e–02	5.2e–04	9.881e–01	8.0e–03	3.3e–04	9.832e–01	50
38	1.7e–02	1.0e–04	9.878e–01	7.7e–03	6.5e–04	9.816e–01	56
39	1.4e–02	1.6e–04	9.873e–01	7.4e–03	4.2e–04	9.811e–01	58
41	1.2e–02	6.6e–05	9.869e–01	7.1e–03	4.6e–04	9.801e–01	61
43	1.1e–02	8.1e–05	9.866e–01	7.0e–03	4.9e–04	9.792e–01	65
45	1.1e–02	1.1e–04	9.863e–01	6.6e–03	3.8e–04	9.790e–01	69
47	1.0e–02	7.8e–05	9.860e–01	7.0e–03	2.1e–04	9.790e–01	73
49	9.9e–03	7.1e–05	9.858e–01	7.1e–03	6.2e–05	9.789e–01	78
50	**9.9e–03**	**6.1e–05**	**9.853e–01**	6.8e–03	7.0e–05	9.789e–01	80

Table 4.3: Comparison of polyhedral versus polyhedral + semidefinite aggregation.

4.6 Concluding Remarks

There is much room for improvement in the present implementation. In particular, eigenvalue and eigenvector computation can be terminated prematurely when an estimate for the largest eigenvalue (provided by the

Lanczos routine) is sufficiently large to guarantee a null step. This can result in significant savings, as observed in [48]. The matrix Q does not have to be recomputed from scratch in each bundle iteration. In any iteration in which aggregation is not performed, p only changes slightly (new columns are added), and hence the matrix Q (see (4.18) – (4.21)) may be updated efficiently. Finally, a line search could be employed to choose ρ^k (F. Rendl, private communication). If the time taken for the solution of the subproblem is negligible compared to a full bundle iteration (as is often the case), a line search in ρ^k, which would require the subproblem to be solved a small number of times (say, about two or three times) within each bundle iteration, could improve the overall efficiency of the algorithm.

The bundle method described in this chapter is a first order method, whereas interior–point methods (see Section 1.6) are essentially second order methods. While the latter enjoy the desirable properties of global convergence with polynomial complexity and, with mild assumptions, rapid local convergence, the only guarantee of first order bundle methods is global convergence. The asymptotic local convergence can be sublinear in the worst case. (See [50, Example IX.2.2.3] for an example where the norm of the subgradient decreases like $1/\sqrt{k}$, where k is the iteration index.) Oustry [84] applies the $\mathcal{U}$–Lagrangian theory of [65] to the $\lambda_{\max}(\cdot)$ function to derive an asymptotically quadratically convergent second order bundle method in [83]. The bundling process adds global convergence to the second order local method of Overton and Womersley [88] (see also [100]). However, computing exact second order information (the so called $\mathcal{U}$–Hessian of the $\lambda_{\max}(\cdot)$ function) is prohibitively expensive in the large scale case. The use of (possibly approximate) second order information in a practical algorithm is still an open issue for large scale problems.

Chapter 5

Electronic Structure Calculations

In this chapter, we describe a new application of semidefinite programming in the area of electronic structure calculations. We give some numerical results for these problems using both the interior–point method and the bundle method described earlier. This is a part of on–going joint work with B. Braams, S. Jiang, J. Percus, F. Oustry and M. L. Overton. The author is grateful to B. Braams for many tutorials explaining the details of this application, and to S. Jiang for use of his *configuration interaction* code.

5.1 Introduction

We deal with the k–fold exterior product $\wedge^k \mathbb{L}^2(\mathbb{R}^3 \times \{\pm\frac{1}{2}\})$ (denoted $\mathbb{L}^2$ for brevity), the Hilbert space of (Lebesgue) measurable, complex–valued, square–integrable, completely antisymmetric functions of k spatial coordinates $r_i \in \mathbb{R}^3$ and k *spin* coordinates $s_i \in \{\pm\frac{1}{2}\}$ $(i = 1, \ldots, k)$, with the inner product

$$\langle f, g \rangle = \sum_{s_1,\ldots,s_k} \int g^*(r_1, \ldots, r_k; s_1, \ldots, s_k)\, f(r_1, \ldots, r_k; s_1, \ldots, s_k) d^3r_1 \ldots d^3r_k,$$

where g^* is the complex conjugate of g, d^3r_i is the usual Lebesgue measure on $\mathbb{R}^3$, and the integration is over the whole space. The norm on this space is $\|f\| = \sqrt{\langle f, f \rangle}$. As usual, we let $\hbar = h/(2\pi)$, where h is Planck's constant. We use Δ_i to denote the Laplacian with respect to the i^{th} spatial coordinate. As in Chapter 4, we will use the symbol $\succeq$ to denote membership in the positive semidefinite cone, *i.e.* for a symmetric matrix v, $v \succeq 0$ means that v is positive semidefinite. The symbol ${}^lC_k$ $(l \geq k \geq 0)$ denotes the binomial coefficient $l!/(k!(l-k)!)$.

Consider a system of k identical particles each of mass m, and whose positions we denote by $r_i \in \mathbb{R}^3$ $(i = 1, \ldots, k)$, moving in an external potential $V(\cdot)$. The *nonrelativistic Hamiltonian* for such a system is the operator $H : \mathbb{L}^2 \to \mathbb{L}^2$, whose action on $\psi \in \mathbb{L}^2$ is given by

$$H\psi = H_1\psi + H_2\psi \tag{5.1}$$

$$H_1\psi = \sum_{i=1}^{k} \frac{-\hbar^2}{2m}\Delta_i\psi + V(r_i)\psi \tag{5.2}$$

$$H_2\psi = \sum_{1 \leq i < j \leq k} U(r_i, r_j)\psi \tag{5.3}$$

The first and the second terms in the summand of (5.2) are respectively the kinetic energy and the potential energy, whereas (5.3) represents two–particle Coulomb interactions. H_1 and H_2 are respectively called the *one–body* and the *two–body* terms of the Hamiltonian.[1] For example, if we are dealing with an atomic system with atomic number z, with the nucleus located at the origin, then

$$V(r_i) \propto \frac{-ze^2}{\|r_i\|_2}$$

[1] The Hamiltonian H, and its one–body and two–body parts H_1 and H_2 respectively, are not to be confused with the earlier usage for feasible sets on page 3.

$$U(r_i, r_j) \propto \frac{e^2}{\|r_i - r_j\|_2},$$

where e is the electronic charge.

Underlying such a system is the notion of a quantum state, *i.e.* a *wave function* $\psi \in \mathbb{L}^2$, which satisfies the (time independent, many–body) Schrödinger equation

$$H\psi = \lambda\psi. \tag{5.4}$$

The eigenvalues λ of H correspond to possible energy levels of the system, and the associated eigenfunctions ψ are the so called *pure states*. An eigenfunction ψ, despite the freedom in its magnitude and phase, corresponds to a single quantum state. We will always normalize ψ to unit norm. The physical interpretation of the normalized wave function is that it is a *probability amplitude*, and that $|\ \psi(r_1, \ldots, r_k)\ |^2$ is the *positional probability density*. For a system of electrons, the wave function is antisymmetric, *i.e.* for any permutation π of the electrons,

$$\psi(r_1, \ldots, r_k) = \mathrm{sgn}(\pi)\, \psi(\pi(r_1, \ldots, r_k))$$

where $\mathrm{sgn}(\pi)$, the sign of the permutation, is $+1$ if the permutation is even, and -1 if the permutation is odd. We assume throughout that we are dealing with a system of electrons, and hence with an antisymmetric wave function. These facts may be found in a textbook on quantum chemistry, for instance [104].

5.2 Reduced Density Operators

This section discusses the *representability* of *reduced density operators*, and is based on Coleman [18–20], Garrod and Percus [37] and Erdahl [28]. See also the surveys by Erdahl [29, 30] for the key role played by convexity in the representability problem.

Our interest lies in a fundamental quantity in quantum chemistry, namely the smallest eigenvalue of H,

$$\lambda_{\min}(H) = \min_{\|\psi\|=1} \langle\, \psi, H\psi \,\rangle, \tag{5.5}$$

which is called the *ground state energy* of the system.

In order to see how this may be calculated, let us move to a finite dimensional setting by introducing a finite basis for the k–particle wave function $\psi(r_1, \ldots, r_k)$. Such a finite basis may be obtained by first selecting a finite basis $\{\phi_i(r)\}$ $(i = 1, \ldots, l)$ for the *one–particle* wave function $\psi(r)$, and then computing the k–particle basis functions via the *Slater determinants*:

$$\phi_{i_1 \ldots i_k}(r_1, \ldots, r_k) = \frac{1}{\sqrt{k!}} \begin{vmatrix} \phi_{i_1}(r_1) & \ldots & \phi_{i_1}(r_k) \\ \vdots & \ldots & \vdots \\ \phi_{i_k}(r_1) & \ldots & \phi_{i_k}(r_k) \end{vmatrix} \quad (1 \le i_1, \ldots, i_k \le l).$$

Note that interchange of two particles results in the interchange of two columns in the determinant, and hence a change in the sign of the basis function $\phi_{i_1 \ldots i_k}$. This automatically builds in antisymmetry into the wave function ψ. From these antisymmetry considerations, it suffices to have ${}^lC_k$ k–particle basis functions to represent ψ. We will not be concerned with the selection of the one–particle basis functions; we assume that they are given, and that an appropriate basis for the k–particle wave function can be generated.

We associate any $\psi(r_1, \ldots, r_k) \in \mathbb{L}^2$ with its vector of coordinates in this basis, with each coordinate $\psi(i_1, \ldots, i_k) \in \wedge^k \mathbb{C}^l$. We denote the latter ${}^lC_k$–dimensional coordinate space by $\mathbb{F}_k$. Viewing H as a Hermitian operator on $\mathbb{F}_k$, the minimization in (5.5) can be written as

$$\lambda_{\min}(H) = \min_{P \in \mathcal{P};\ \mathrm{tr}(P)=1} \langle\, H, P \,\rangle, \tag{5.6}$$

where $\mathcal{P}$ is the cone of Hermitian, positive semidefinite operators on $\mathbb{F}_k$. However, the dimension of the space of linear operators on $\mathbb{F}_k$ is $({}^lC_k)^2$. Computing the smallest eigenvalue in this framework, called a *full configuration interaction* (CI) calculation, is prohibitively expensive even for modest values of l and k.

All the information contained in the wave function is not needed for the computation of the ground state energy. It is possible to reformulate the problem in terms of more economical objects, namely the *reduced density operators*, which contain just enough information for our purpose, and which greatly reduce the dimensionality of the problem.

Let us define the following one–particle and two–particle reduced density operators

$$\Gamma_1(i,i') = k \sum_{i_2,\dots,i_k} \psi(i,i_2,\dots,i_k)\psi^*(i',i_2,\dots,i_k) \qquad (1 \le i,i' \le l) \tag{5.7}$$

$$\Gamma_2(i,j;i',j') = \frac{1}{2}k(k-1) \sum_{i_3,\dots,i_k} \psi(i,j,i_3,\dots,i_k)\psi^*(i',j',i_3,\dots,i_k) \quad (1 \le i,i',j,j' \le l). \tag{5.8}$$

We may associate the one–particle and the two–particle terms of the Hamiltonian with their matrices of coordinates

$$H_1(i,j) = \int \phi_i^*(r) H_1(r,r') \phi_j(r) d^3r d^3r' \qquad (1 \le i,j \le l)$$

$$H_2(i,j;i',j') = \int \phi_i^*(r_1)\phi_j^*(r_2) H_2(r_1,r_2;r_1'r_2') \phi_{i'}(r_1')\phi_{j'}(r_2')\, d^3r_1 d^3r_2 d^3r_1' d^3r_2' \quad (1 \le i,j,i',j' \le l)$$

in the chosen basis, and rewrite (5.5) as

$$\lambda_{\min}(H) = \min_{(\Gamma_1,\Gamma_2)\in D_1\times D_2} \langle\, H_1, \Gamma_1 \,\rangle + \langle\, H_2, \Gamma_2 \,\rangle \tag{5.9}$$

where the inner product is now the trace inner product between matrices, and $D_1 \times D_2$ is the set of all pairs (Γ_1, Γ_2) derivable, as in (5.7) and (5.8), from some antisymmetric wave function $\psi \in \mathbb{L}^2$ of unit norm. However, note that Γ_1 and Γ_2 are not independent variables. In fact, from (5.7) and (5.8),

$$\Gamma_1(i,i') = \frac{2}{k-1} \sum_j \Gamma_2(i,j;i',j),$$

so that Γ_1 is just a scaled partial trace of Γ_2. We denote this relation as

$$\Gamma_1 = \ell_0(\Gamma_2),$$

where ℓ_0 is a linear map.

These reduced density operators suffice to compute any quantity of the system that depends only on one-body and two-body interactions. This class of quantities includes the Hamiltonian.

Although D_1 and D_2 are convex cones [28], the fundamental problem with the formulation (5.9) is that, for $k \ge 3$, there is no explicit description available for D_2, the set of "representable" two–body reduced density operators, *i.e.* there is no known characterization that allows us to verify if a given $\Gamma_2 \in D_2$ is representable, via (5.8), from some antisymmetric $\psi \in \mathbb{L}^2$. (The structure of representable Γ_1 has been completely characterized by Coleman [18].) This fundamental problem in quantum mechanics is called the *representability problem*. It is well known that D_2 is far from being polyhedral.

A *representability condition* for D_2 is a symmetric matrix V such that $\langle\, \Gamma_2, V \,\rangle \ge 0 \;\; \forall \Gamma_2 \in D_2$, *i.e.* a *necessary* condition for Γ_2 to be a valid two–body operator. Observe that the set of all representability conditions for D_2 is itself a cone, and coincides with D_2^*, the dual cone of D_2.

In the next section, we will describe three representability conditions in the literature, all of which lead to semidefinite constraints.

5.3 Semidefinite Relaxations

In this section, we describe three representability conditions (called the P–, the Q–, and the G–conditions), which were discovered by Garrod and Percus [37].

As described in Section 5.2, let $\mathbb{F}_k$ denote the ${}^lC_k$–dimensional coordinate space of the k–particle basis functions. Then, the 2^l–dimensional space $\mathbb{F} = \oplus_{i=1}^k \mathbb{F}_k$ is called *Fock space*, and is spanned by basis elements

denoted ϕ_I, where I is an arbitrary subset of $\{1,\ldots,l\}$. For any $i \in \{1,\ldots,k\}$ and $I \in 2^{\{1,\ldots,l\}}$, define the "parity" function

$$p(i,I) = |\,\{j \in I \;:\; j < i\}\,|,$$

where $|\cdot|$ denotes cardinality. Then, we may define on Fock space, the *annihilation operator*[2]

$$a_i \;:\; \mathbb{F} \to \mathbb{F} \quad :\; \phi_I \to a_i\phi_I = \begin{cases} (-1)^{p(i,I)}\phi_{I\setminus\{i\}} & \text{if } i \in I \\ 0 & \text{otherwise,} \end{cases}$$

whose adjoint is the *creation operator*

$$a_i^\dagger \;:\; \mathbb{F} \to \mathbb{F} \quad :\; \phi_I \mapsto a_i^\dagger\phi_I = \begin{cases} (-1)^{p(i,I)}\phi_{I\cup\{i\}} & \text{if } i \notin I \\ 0 & \text{otherwise.} \end{cases}$$

Then, for any polynomial Q with complex coefficients in the creation and the annihilation operators, the operator Q^*Q is positive semidefinite, hence yielding the representability condition that for all ψ,

$$\langle\, \psi, Q^*Q\psi \,\rangle \geq 0. \tag{5.10}$$

Different choices of the polynomial Q lead to different representability conditions in (5.10). However, Q^*Q is expressible in terms of Γ_1 and Γ_2 only when Q is of degree 1 or 2. Each of the following specific forms for Q, when substituted into (5.10), results in a representability condition, whose name we indicate within parentheses.

$$Q = \sum_i c_i a_i \quad \Rightarrow \quad Q^*Q = \Gamma_1 \quad \text{(p–condition)} \tag{5.11}$$

$$Q = \sum_i c_i a_i^\dagger \quad \Rightarrow \quad Q^*Q = I - \Gamma_1 \quad \text{(q–condition)} \tag{5.12}$$

$$Q = \sum_{i,j} c_{ij} a_i a_j \quad \Rightarrow \quad Q^*Q = \Gamma_2 \quad \text{(P–condition)} \tag{5.13}$$

$$Q = \sum_{i,j} c_{ij} a_i^\dagger a_j^\dagger \quad \Rightarrow \quad Q^*Q = \ell_1(\Gamma_1,\Gamma_2) \quad \text{(Q–condition)} \tag{5.14}$$

$$Q = c_0 + \sum_{i,j} c_{ij} a_i a_j^\dagger \quad \Rightarrow \quad Q^*Q = \ell_2(\Gamma_1,\Gamma_2) \quad \text{(G–condition).} \tag{5.15}$$

Here $c_0, c_i, c_{ij} \in \mathbb{C}$, and ℓ_1 and ℓ_2 are affine functions of the matrix variables Γ_1 and Γ_2. The exact forms of ℓ_1 and ℓ_2 are somewhat complicated, and are detailed in [37].

Thus we obtain the semidefinite relaxation

$$\begin{aligned} \min \quad & \langle\, H_1, \Gamma_1 \,\rangle + \langle\, H_2, \Gamma_2 \,\rangle \\ & \Gamma_1 = \ell_0(\Gamma_2); \quad \operatorname{tr}(\Gamma_1) = k; \quad \operatorname{tr}(\Gamma_2) = k(k-1)/2 \\ & \ell_1(\Gamma_1,\Gamma_2) \succeq 0; \quad \ell_2(\Gamma_1,\Gamma_2) \succeq 0 \\ & \Gamma_1 \succeq 0; \quad \Gamma_2 \succeq 0; \quad I - \Gamma_1 \succeq 0, \end{aligned} \tag{5.16}$$

which results in a lower bound on the ground state energy $\lambda_{\min}(H)$.

It turns out that for $k \leq 2$, the representability conditions resulting from (5.11) – (5.15) are also sufficient. In this case, (5.16) yields the exact answer.

This size of this relaxation is independent of the number of particles k in the system, which enters only as a parameter in the data. When cast in standard form, this SDP has only $O(l^4)$ dual variables, and a primal matrix variable of order $O(l^2)$, and hence solving the SDP is more economical than a full CI calculation.

The specific forms of ℓ_1 and ℓ_2 are such that the constant trace condition on Γ_1 and Γ_2 also imply that $\ell_1(\Gamma_1,\Gamma_2)$ and $\ell_2(\Gamma_1,\Gamma_2)$ have constant trace. Thus, this SDP is amenable to the transformation of Section 4.1, and is expressible as an unconstrained eigenvalue optimization problem. In the next section, we will describe the results of applying both the interior–point method of Chapter 3 and the bundle method of Chapter 4 to solve the SDP relaxation in (5.16).

[2]Not to be confused with the earlier usage in (4.1).

5.4 Numerical Results

In this section, we present numerical results on some instances of (5.16). In all cases, both the one–body and two–body terms in the Hamiltonian were randomly generated. All the instances were solved on a Sun Ultra Sparc I with a 143 MHz CPU and 64 MB of memory.

In Table 5.1, we solve some small problems with the interior–point code *SeQuL* (see Chapter 3), and with a CI code written by S. Jiang, to compute $\lambda_{\min}(H)$. Thus, the answer computed by the CI code is to be considered as the true ground state energy, whereas the optimal value of the SDP in (5.16) is an approximation. The SDP relaxation is actually exact in the case $k \leq 2$, hence we start with $k = 3$. Also, there is a symmetry between the cases when the particle number is k and $l - k$, hence it suffices to consdider only half the possible range for k, *i.e.* $3 \leq k \leq \lceil l/2 \rceil$. The first two columns in Table 5.1 show the number of one–particle basis functions and particle number respectively. The third and the fourth columns indicate the block structure vector N and the number of primal equality constraints m. (Note that these depend only the number of one–particle basis functions l, and not on the particle number k.) The next two columns represent the minimum and the average *relative deviation* of the optimal value of the SDP relaxation from the true ground state energy (obtained via a full CI calculation), averaged over 5 randomly generated Hamiltonians. By relative deviation, we mean the ratio (value(SDP) − value(CI))/value(CI). The negative signs in the table denote that the optimal value of the SDP is a *lower bound*. The last column is the average CPU time, in hours (hh), minutes (mm) and seconds (ss), taken by the interior–point code to solve the SDP to high accuracy (given by *SeQuL*'s default tolerances). The quality of the relaxation tends to deteriorate both with increase in the particle number k and the number l of one–particle basis functions.

In Table 5.2, we solve some larger instances, again with randomly generated Hamiltonians, but with the bundle method described in Chapter 4, by reformulating the SDP in (5.16) as an unconstrained eigenvalue optimization problem. The first four columns are as before. The fifth and the sixth columns give the norm of a computed ϵ–subgradient and the corresponding value of ϵ. The columns λ and Q give the percentage of time spent in the eigenvalue/eigenvector computatation and in forming the matrix Q in (4.18). Note that, in contrast with the observation of Section 4.5, forming Q now dominates the computational cost. The last column gives the CPU time taken in hours (hh), minutes (mm) and seconds (ss).

The problem instances studied here are very small by the standards of quantum chemistry. However, our on–going work is concentrated at present on investigating the strength of various representability conditions (including the P–, the Q–, and the G–conditions), and on numerical exploration of the boundaries of the representable region. For these studies, already the small model systems explored here are of much interest.

l	k	N	m	min	ave	hh:mm:ss
6	3	[6 15 15 36]	808	–6.72e–02	–2.78e–02	3:07
7	3	[7 21 21 49]	1485	–1.38e–01	–9.09e–02	33:47
	4			–1.07e–01	–6.35e–02	1:16:40
8	3	[8 28 28 64]	2523	–1.79e–01	–1.37e–01	4:38:20
	4			–2.91e–01	–2.32e–01	9:22:04

Table 5.1: Comparison of ground state energies computed via reduced density operators and full configuration interaction, averaged over 5 randomly generated Hamiltonians.

l	k	N	m	Ser/Tot	$\|\overline{g}\|$	ϵ	λ	Q	hh:mm:ss
6	3	[6 15 15 36]	808	23/50	1.4e–01	5.4e–03	27	71	5:37
7	3	[7 21 21 49]	1485	23/50	1.2e–01	5.8e–04	19	79	16:09
	4			25/50	1.7e–01	6.9e–04	19	79	16:57
8	3	[8 28 28 64]	2523	25/50	1.4e–01	8.5e–02	16	84	48:42
	4			23/50	1.9e–01	7.6e–02	16	84	50:02
9	3	[9 36 36 81]	4033	24/50	1.6e–01	1.0e–02	13	87	1:44:09
12	3	[12 66 66 144]	12,370	25/50	3.2e–01	1.3e–03	10	90	13:10:02
15	3	[15 105 105 225]	31,111	14/50	2.9e–01	2.5e–02	6	94	55:42:35

Table 5.2: Ground state energy calculations via reduced density operators, solved with a first order bundle method.

Appendix A

Costs Per Interior–Point Iteration

This appendix is extracted from [45]. Here we derive the estimates for the costs of forming the Schur complement matrix M in (3.5), solving the linear system once M has been factored (3.14), and computing the step lengths to the boundary of the cone for a given direction (3.15). We assume that all data is dense, and that the basic costs of matrix multiplication and eigenvalue computation are $O(n^3)$ for n by n matrices.

We begin by estimating the cost of forming the matrices M_S, M_Q and M_L, defined in (3.7), (3.8) and (3.9), making up the Schur complement matrix (3.5). The cost of forming M_L is clearly $O(m^2 n_0)$. From (3.8) we get that

$$M_Q = A_Q \mathrm{Arw}(z_Q)^{-1} \mathrm{Arw}(x_Q) A_Q^* = \sum_{i=1}^{q} M_Q(i)$$

where

$$M_Q(i) = A_Q(i) \mathrm{arw}(z_Q(i))^{-1} \mathrm{arw}(x_Q(i)) A_Q(i)^*,$$

$\mathrm{arw}(x_Q(i))$ is the i^{th} diagonal block of $\mathrm{Arw}(x_Q)$, and $A_Q(i)$ is the m by $n_i + 1$ matrix whose rows are the transpose of the i^{th} blocks of the vectors $(A_Q)_k$, $1 \le k \le m$. Let $[x_0, x_1, \ldots, x_{n_i}]$ and $[z_0, z_1, \ldots, z_{n_i}]$ denote the i^{th} blocks of x_Q and z_Q respectively, and let $\widetilde{x} = [0, x_1, \ldots, x_{n_i}] \in \mathbb{R}^{n_i+1}$, $\widetilde{z} = [0, z_1, \ldots, z_{n_i}] \in \mathbb{R}^{n_i+1}$, and $e_0 = [1, 0, \ldots, 0] \in \mathbb{R}^{n_i+1}$. Finally, write $B = A_Q(i)$ and let $\zeta = z_0^2 - \widetilde{z}^T \widetilde{z}$. Then we have

$$\begin{aligned} M_Q(i) = {} & \frac{x_0}{z_0} BB^* + \frac{x_0}{\zeta z_0} (B\widetilde{z})(B\widetilde{z})^* + \frac{1}{z_0} \left[(Be_0)(B\widetilde{x})^* + (B\widetilde{x})(Be_0)^* \right] \\ & + \frac{1}{\zeta} \left[\left(\frac{x_0}{z_0} \widetilde{z}^* \widetilde{z} - \widetilde{z}^* \widetilde{x} \right) (Be_0)(Be_0)^* - x_0 \Big((Be_0)(B\widetilde{z})^* + (B\widetilde{z})(Be_0)^* \Big) \right] \\ & + \frac{1}{\zeta} \left[\frac{\widetilde{z}^* \widetilde{z}}{z_0} (Be_0)(B\widetilde{x})^* - (B\widetilde{z})(B\widetilde{x})^* + \frac{\widetilde{z}^* \widetilde{x}}{z_0} (B\widetilde{z})(Be_0)^* \right]. \end{aligned}$$

Observe that the matrix BB^* needs to be computed only once (for each i) at the start of the algorithm, and we therefore ignore the cost of computing it. Thus the cost of computing $M_Q(i)$ is $O(mn_i + m^2)$ and the cost of computing M_Q is $O(md_q + m^2 q)$.

To estimate the cost of computing M_S, observe that computing the right–hand side of (3.10) and computing the solution to the Lyapunov equation (3.10) require $O(N_i^3)$ operations and there are m such equations to solve. Computing the i^{th} summand in (3.11) requires $O(N_i^2)$ operations so that the cost of computing the (k, l) entry of M_S is $O(d_S)$. Thus the total cost of computing M_S is $O(m \sum_{i=1}^{s} N_i^3 + m^2 d_S)$.

Next we consider the cost of solving the linear system (3.3) once the factorization of the Schur complement matrix M has been obtained. Computing the right–hand side of the Schur complement system (see [7]) requires $O(\sum_{i=1}^{s} N_i^3 + d_Q + n_0)$ operations, while the cost of the backsolve to obtain Δy is $O(m^2)$. The cost of computing Δz is $O(m(d_S + d_Q + n_0)) = O(md)$ and the cost of computing Δx from Δz is $O(\sum_{i=1}^{s} N_i^3 + d_Q + n_0)$. Thus we obtain the estimate (3.14).

Finally, computing the step lengths to the boundary of the semidefinite cone in the directions Δx_S and Δz_S requires matrix–matrix multiplications and an eigenvalue decomposition and so is an $O(\sum_{i=1}^{s} N_i^3)$ operation. The cost of computing the step lengths to the boundary of the quadratic cone (respectively positive orthant) in the directions Δx_Q, Δz_Q (respectively Δx_L, Δz_L) is $O(d_Q)$ (respectively $O(n_0)$). Summing up the contributions, we obtain the estimate (3.15).

Appendix B

Costs Per Bundle Iteration

Recall that A is given by the m matrices $a_i \in \mathbb{S}^N$, and p, the current bundle of eigenvectors, is a block matrix whose j^{th} block is an $N_j \times r_j$ orthonormal matrix. We set

$$R = \sum_{j=1}^{s} r_j(r_j+1)/2, \text{ and}$$
$$d_S = \sum_{j=1}^{s} N_j(N_j+1)/2.$$

Although l and R are parameters independent of the size of the original problem m and d_S, we retain them in the operation counts in order to have some idea of their influence on the constant factors. Let the i^{th} column of G^* be h_i $(1 \le i \le m)$. We assume that the cost of multiplying a $p_1 \times p_2$ matrix and a $p_2 \times p_3$ matrix is $O(p_1p_2p_3)$.

Forming Q_{11} costs $O(m(l+R)^2)$. Rewriting (4.20) as a sum of m outer products,

$$Q_{12} = \frac{1}{\rho}\sum_{i=1}^{m} h_i(\mathbf{vec}(p^*a_ip))^*,$$

we need to compute the m orthogonal conjugations p^*a_ip $(i = 1, \ldots, m)$ in order to compute Q_{12} and Q_{22}. Let c_{ij} be the cost of computing the j^{th} block of the i^{th} conjugation. This conjugation may be computed as a direct matrix product, incurring a cost of $O(N_jr_j^2 + r_jN_j^2)$, or as in (4.22), incurring a cost of $f_{ij}O(r_j^2)$, where f_{ij} is the number of nonzeros in $a_i(j)$. Thus, it is cheaper to use (4.22) if $f_{ij} = o(N_j + N_j^2/r_j)$. The m orthogonal conjugations cost $\sum_{i,j} c_{ij}$. Computing each summand (an outer product) in Q_{12} and Q_{22} will additionally cost $O(lR)$ and $O(R^2)$ operations respectively, so the total cost of forming Q is

$$\sum_{i=1}^{m}\sum_{j=1}^{s} c_{ij} + m\,O(lR + R^2 + (l+R)^2),$$

and the cost of factorizing Q (see Section 4.4.1) is $O((l+R)^3)$.

When is expressed in standard form, the COP has $l + R + 1$ dual variables, and the primal variables in its SD, QC and LP components have the structure given in Table B.1.

Variable type	Block structure	Data sparsity
SD	$[r_1, \ldots, r_s]$	Yes
QC	$[l+R+3]$	No
LP	$l+2$	Yes

Table B.1: Structure of subproblem in every bundle iteration.

The costs involved in solving the subproblem have been detailed in Appendix A. The dominant cost in every interior–point iteration is the formation and factorization of the Schur complement matrix (see Section 3.3.1), which is of order $l+R+1$. Since l and r are typically small (say, $l = 50$, $r = 25$), compared to the parameters of the original problem ($N = [N_1, \ldots, N_s]$ and m), the cost of solving the subproblem is generally small compared that of forming Q, and of computing a few of the largest eigenvalues and eigenvectors of $A^*\overline{y} - c$ in Step 5 of Algorithm 4.2.1.

Note that when m gets large, *e.g.* if $m = O(d_S)$, then the cost of forming Q can exceed the cost of computing eigenvalues and eigenvectors.

Bibliography

[1] I. Adler and F. Alizadeh. Primal–dual interior point algorithms for convex quadratically constrained and semidefinite optimization problems. Technical Report RRR 46–95, RUTCOR, Rutgers University, Piscataway (New Jersey), December 1995.

[2] F. Alizadeh. *Combinatorial optimziation with interior–point methods and semidefinite matrices*. PhD thesis, University of Minnesota, 1991.

[3] F. Alizadeh. Interior–point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.

[4] F. Alizadeh, J–P. Haeberly, M. V. Nayakkankuppam, and M. L. Overton. SDPpack User Guide (Version 0.8 Beta). Technical Report 734, Courant Institute of Mathematical Sciences, New York (New York), March 1997. Also available from the SDPpack web page: http://www.cs.nyu.edu/faculty/overton/sdppack.

[5] F. Alizadeh, J–P. Haeberly, M. V. Nayakkankuppam, M. L. Overton, and S. Schmieta. SDPpack User Guide (Version 0.9 Beta). Technical Report 737, Courant Institute of Mathematical Sciences, New York (New York), June 1997. Also available from the SDPpack web page: http://www.cs.nyu.edu/faculty/overton/sdppack.

[6] F. Alizadeh, J–P. Haeberly, and M.L. Overton. Complementarity and nondegeneracy in semidefinite programming. *Mathematical Programming (Series B)*, 77(2):29–43, May 1997.

[7] F. Alizadeh, J–P. Haeberly, and M.L. Overton. Primal–dual interior–point algorithms for semidefinite programming: stability, convergence and numerical results. *SIAM Journal on Optimization*, 8(3):743–768, 1998.

[8] F. Alizadeh and S. Schmieta. Optimization with semidefinite, quadratic and linear constraints. Technical Report RRR 23–97, RUTCOR, Rutgers University, Piscataway (New Jersey), November 1997.

[9] R. Bellman and K. Fan. On systems of linear inequalities in Hermitian matrix variables. In V. L. Klee, editor, *Convexity*, volume 7 of *Proceedings of Symposia in Pure Mathematics*, pages 1–11. American Mathematical Society, 1963.

[10] A. Ben–Tal and M. P. Bendsøe. A new method for optimal truss topology design. *SIAM Journal on Optimization*, 3:322–358, 1993.

[11] A. Ben–Tal and A. Nemirovskii. Interior–point polynomial–time method for truss topology design. Technical report, Faculty of Industrial Engineering and Management, Technion Institute of Technology, Haifa (Israel), 1992.

[12] M. Bendsøe, A. Ben–Tal, and J. Zowe. Optimization methods for truss geometry and topology design. *Structural Optimization*, 7:141–159, 1994.

[13] S. Benson, Y. Ye, and X. Zhang. Solving large–scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, To appear.

[14] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia, PA, 1994.

[15] A. Brøndsted and R. T. Rockafellar. On the subdifferentiability of convex functions. *Proceedings of the American Mathematical Society*, pages 605–611, 1965.

[16] T. J. Carpenter, I. J. Lustig, J. M. Mulvey, and D. F. Shanno. Higher–order predictor–corrector interior–point methods with application to quadratic objectives. *SIAM Journal on Optimization*, 3:696–725, 1993.

[17] F. H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley & Sons, 1983.

[18] A. J. Coleman. Structure of Fermion density matrices. *Reviews of Modern Physics*, 35(3), July 1963.

[19] A. J. Coleman. Necessary conditions for the n–representability of reduced density matrices. *Journal of Mathematical Physics*, 13(2):214–222, 1972.

[20] A. J. Coleman. Reduced density operators and the n–particle problem. *International Journal of Quantum Chemistry*, 13:67–82, 1978.

[21] J. Cullum, W. E. Hoffman, and P. Wolfe. The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices. *Mathematical Programming Study*, 3:35–65, 1975.

[22] J. Czyzyk, S. Mehrotra, M. Wagner, and S. J. Wright. PCx User Guide Version 1.1. Technical Report OTC 96/01, Optimization Technology Center, November 1997.

[23] Netlib LP problem set. URL: www.netlib.org/lp/data/readme.

[24] E. de Klerk. *Interior–point methods for semidefinite programming*. PhD thesis, Technische Universiteit Delft, 1997. ISBN 90-9011055-0.

[25] I. S. Duff, R. G. Grimes, and J. G. Lewis. User's guide for the Harwell–Boeing sparse matrix collection (release 1). Technical Report RAL–92–086, rutherford Appleton Laboratory, December 1992.

[26] L. El Ghaoui and H. Lebret. Robust solutions to least–squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

[27] L. El Ghaoui and S. Niculescu, editors. *Recent Advances in Linear Matrix Inequality Methods in Control*. SIAM, Philadelphia (Pennsylvania), 1999.

[28] R. Erdahl. The convex structure of the set of n–representable reduced 2–matrices. *Journal of Mathematical Physics*, 13(10):1608–1621, 1972.

[29] R. Erdahl. Representability. *International Journal On Quantum Chemistry*, 13:697–718, 1978.

[30] R. Erdahl. Representability conditions. In R. Erdahl and Jr. V. H. Smith, editors, *Density Matrices And Density Functionals*, pages 51–75. D. Reidel Publishing Company, 1987.

[31] J. Faraut and A. Korányi. *Analysis on symmetric cones*. Oxford Mathematical Monographs. Oxford University Press, 1994.

[32] L. Faybusovich. Linear systems in Jordan algebras and primal–dual interior–point algorithms. Technical report, Department of Mathematics, University of Notre Dame, August 1996. To appear in Journal of Computational and Applied Mathematics.

[33] L. Faybusovich. Euclidean Jordan algebras and interior–point algorithms. *Positivity*, I:331–357, 1997.

[34] R. Fletcher. Semidefinite matrix constraints in optimization. *SIAM Journal on Control and Optimization*, 23:493–523, 1985.

[35] K. Fujisawa, M. Kojima, and K. Nakata. Exploiting sparsity in primal–dual interior–point methods for semidefinite programming. *Mathematical Programming (Series B)*, pages 235–254, 1997.

[36] K. Fujisawa, M. Kokima, and K. Nakata. SDPA User's Guide. Technical Report B–308, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh–Okayama, Meguro, Tokyo 152 (Japan), 1995. Revised 1996.

[37] C. Garrod and J. K. Percus. Reduction of the N–particle variational problem. *Journal of Mathematical Physics*, 5(12):1756–1776, December 1964.

[38] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[39] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore (Maryland), second edition, 1989.

[40] J. Gondzio. Multiple centrality corrections in a primal–dual method for linear programming. *Computational Optimization and Applications*, 6:137–156, 1996.

[41] J. Gondzio and M. Makowski. HOPDM modular solver for LP problems, User's guide to version 2.12. Technical Report WP–95–50, International Institute for Applied Systems Analysis, Laxenburg (Austria), June 1995.

[42] O. Güler. Hyperbolic polynomials and interior point methods for convex programming. *Mathematics of Operations Research*, 22:350–377, 1997.

[43] O. Güler and L. Tunçel. Characterization of the barrier parameter in homogeneous convex cones. *Mathematical Programming (Series A)*, 81:55–76, 1998.

[44] J–P. Haeberly. Remarks on nondegeneracy in mixed semidefinite–quadratic programming. Unpublished note, URL: www.fordham.edu/haeberly/papers/sqldegen.ps.gz, September 1998.

[45] J–P. Haeberly, M. V. Nayakkankuppam, and M. L. Overton. Extending Mehrotra and Gondzio higher order methods to mixed semidefinite–quadratic–linear programming. *Optimization Methods and Software*, To appear.

[46] J–P. Haeberly, M. V. Nayakkankuppam, and M. L. Overton. Mixed semidefinite–quadratic–linear programming. In L. El Ghaoui and S. Niculescu, editors, *Recent Advances in Linear Matrix Inequality Methods for Control*. SIAM, Philadelphia (Pennsylvania), To appear.

[47] C. Helmberg and F. Oustry. Bundle methods to minimize the maximum eigenvalue function. In L. Vandenberghe R. Saigal and H. Wolkowicz, editors, *Hanbook on Semidefinite Programming: Theory, Algorithms and Applications*. Kluwer Academic Publisher, 1999. To appear.

[48] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, To appear.

[49] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz. An interior–point method for semidefinite programming. *SIAM Journal on Optimization*, 6:342–361, 1996.

[50] J–B. Hiriart–Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms*, volume I & II. Springer–Verlag, 1993.

[51] J–B. Hiriart–Urruty and D. Ye. Sensitivity analysis of all eigenvalues of a symmetric matrix. *Numerische Mathematik*, 70:45–72, 1995.

[52] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge (U.K.), 1990.

[53] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge (U.K.), 1994.

[54] L. V. Kantorovič and G. P. Akilov. *Functional Analysis*. Pergamon Press, New York, second edition, 1982.

[55] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[56] K. C. Kiwiel. An aggregate subgradient method for nonsmooth convex minimization. *Mathematical Programming*, 1983.

[57] K. C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, 46:105–122, 1990.

[58] M. Köcher. Jordan algebras and their applications. Lecture notes (University of Minnesota), 1962.

[59] M. Kojima, S. Shindoh, and S. Hara. Interior–point methods for the monotone complementarity problem in symmetric spaces. *SIAM Journal on Optimization*, 7:86–125, 1997.

[60] E. Kranich. Interior–point methods for mathematical programming: a bibliography. URL: www.netlib.org/bib/ipmbib.

[61] C. Lemaréchal. Nonsmooth optimization and descent methods. Technical Report RR 78–4, International Institute of Applied Systems Analysis, Laxenburg (Austria), 1977.

[62] C. Lemaréchal. Nonlinear programming and nonsmooth optimization – a unification. Technical Report 332, Institut de Recherche d'Informatique et d'Automatique, Le Chesnay (France), 1978.

[63] C. Lemaréchal. Constructing bundle methods for convex optimization. In J.–B. Hiriart–Urruty, editor, *Fermat Days 85: Mathematics for Optimization*, volume 129 of *Mathematics Studies*, pages 201–240. North Holland, 1986.

[64] C. Lemaréchal and F. Oustry. Nonsmooth algorithms to solve semidefinite programs. In L. El Ghaoui and S. Niculescu, editors, *Recent Advances in Linear Matrix Inequality Methods in Control*. SIAM, To appear.

[65] C. Lemaréchal, F. Oustry, and C. Sagastizábal. The $\mathcal{U}$–lagrangian of a convex function. *Transactions of the American Mathematical Society*, To appear.

[66] E. S. Levitin. *Perturbation Theory in Mathematical Programming and its Applications*. Wiley–Interscience series in discrete mathematics and optimization. Wiley & Sons, New York, 1994.

[67] A. S. Lewis and M. L. Overton. Eigenvalue optimization. *Acta Numerica*, pages 149–190, 1996.

[68] M. Lobo, L. Vandenberghe, and S. Boyd. Applications of second order cone programming. *Linear Algebra and its Applications*, 284:193–228, November 1998. Special issue on linear algebra in controls, signals and image processing.

[69] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25:355–381, 1979.

[70] Z. .Q. Luo, J. F. Sturm, and S. Zhang. Duality and self–duality in conic convex programming. Technical Report 9620, Erasmus Institute, Rotterdam (The Netherlands), May 1996.

[71] O. L. Mangasarian and T. H. Shiau. Lipschitz continuity of solutions to linear inequalities, programs and complementarity problems. *SIAM Journal on Control and Optimization*, 25(3):583–595, May 1987.

[72] S. Mehrotra. Higher order methods and their performance. Technical Report 90–16R1, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston (Illinois), July 1991.

[73] S. Mehrotra. On the implementation of a priml-dual interior point method. *SIAM Journal on Optimization*, 2:575–601, 1992.

[74] R. D. C. Monteiro. Polynomial convergence of primal–dual algorithms for semidefinite programming based on Monteiro and Zhang family of directions. Technical report, Department of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta (Georgia), 1996. In submission.

[75] R. D. C. Monteiro. Primal–dual path–following algorithms for semidefinite programming. *SIAM Journal on Optimization*, 7:663–678, 1997.

[76] R. D. C. Monteiro and Y. Zhang. A unified analysis for a class of path–following primal–dual interior–point algorithms for semidefinite programming. *Mathematical Programming*, 81:281–299, 1998.

[77] M. V. Nayakkankuppam and M. L. Overton. Conditioning of semidefinite programs. *Mathematical Programming*, To appear.

[78] Y. Nesterov, M. Todd, and Y. Ye. Infeasible–start primal–dual methods infeasibility detectors for nonlinear programming. *Mathematical Programming*, 84:227–267, 1996.

[79] Y. E. Nesterov and A. Nemirovskii. *Interior–point polynomial algorithms in convex programming*, volume 13. SIAM, Philadelphia (Pennsylvania), 1994.

[80] Y. E. Nesterov and M. J. Todd. Self–scaled barriers and interior–point algorithms for convex programming. *Mathematics of Operations Research*, 22:1–42, 1997.

[81] Y. E. Nesterov and M. J. Todd. Primal–dual interior–point methods for self–scaled cones. *SIAM Journal on Optimization*, 8:324–364, 1998.

[82] J. Ortega. *Numerical Analysis: A Second Course*. Computer Science and Applied Mathematics. Academic Press, New York, 1972.

[83] F. Oustry. A second order bundle method to minimize the maximum eigenvalue function. *Mathematical Programming*, Submitted (1998).

[84] F. Oustry. The $\mathcal{U}$–Lagrangian of the maximum eigenvalue function. *SIAM Journal on Optimization*, To appear.

[85] M. L. Overton. On minimizing the maximum eigenvalue of a symmetric matrix. *SIAM Journal on Matrix Analysis and Applications*, 9(2), 1988.

[86] M. L. Overton. Large–scale optimization of eigenvalues. *SIAM Journal on Optimization*, 2(1):88–120, 1992.

[87] M. L. Overton and R. S. Womersley. Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of a symmetric matrix. *Mathematical Programming*, 62:321–357, 1993.

[88] M. L. Overton and R. S. Womersley. Second derivatives for optimizing eigenvalues of symmetric matrices. *SIAM Journal on Matrix Analysis and Applications*, 16(3):697–718, 1995.

[89] G. Pataki. Cone LP's and semidefinite programs: Geometry and Simplex–type method. In *Proceedings of the Fifth Conference on Integer Programming and Combinatorial Optimization (IPCO)*, number 84 in Lecture Notes in Computer Science, Vancouver (Canada), 1996. Springer–Verlag.

[90] G. Pataki and L. Tunçel. On the generic properties of convex optimization problems in conic form. Technical Report CORR 97–16, Department of Combinatorics and Optimization, University of Waterloo, Ontario (Canada), 1997.

[91] J. Renegar. A Mathematical View of Interior–Point Methods in Convex Optimization. manuscript.

[92] J. Renegar. Some perturbation theory for linear programming. *Mathematical Programming*, 65:73–91, 1994.

[93] J. Renegar. Condition numbers, barrier method and the conjugate gradient method. *SIAM Journal on Optimization*, 6(4):879–912, 1996.

[94] S. M. Robinson. Bounds for error in the solution set of a perturbed linear program. *Linear Algebra and its Applications*, 6:69–81, 1973.

[95] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton (New Jersey), 1970.

[96] R. T. Rockafellar and R. J–B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer–Verlag, New York, 1998.

[97] S. Schmieta and F. Alizadeh. Associative algebras, symmetric cones and polynomial time interior–point algorithms. Technical Report RRR 17–98, RUTCOR, Rutgers University, June 1998.

[98] H. Schramm and J. Zowe. A version of the bundle iea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, 2(1):121–152, 1992.

[99] H. R. Schwarz. *Numerical Analysis: A Comprehensive Introduction*. John Wiley & Sons, New York, 1989.

[100] A. Shapiro and M. K. H. Fan. On eigenvalue optimization. *SIAM Journal on Optimization*, 3:552–568, 1995.

[101] G.W. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[102] J. F. Sturm. *Primal–dual interior–point approach to semidefinite programming*. PhD thesis, Erasmus Institute, Rotterdam (The Netherlands), 1997. Tinbergen Institute Research Series, Vol. 156, Thesis Publishers, Amsterdam (The Netherlands) 1997.

[103] J. F. Sturm. *Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones*. Department of Quantitative Economics, Maastricht University, Maastricht (The Netherlands), 1998.

[104] A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry*. McGraw–Hill Publishing Company, first edition (revised) edition, 1989.

[105] M. J. Todd. Potential reduction methods in mathematical programming. *Mathematical Programming*, 76:3–45, 1996.

[106] M. J. Todd. A study of search directions in semidefinite programming. Technical Report TR1205, Department of Operations Research and Industrial Engineering, Cornell University, 1997.

[107] M. J. Todd, K. C. Toh, and R. Tütüncü. SDPT3 – A Matlab software package for semidefinite programming. Technical Report 1177, Department of Operations Research and Industrial Engineering, Cornell University, December 1996. Code available from: http://www.math.nus.edu.sg/ mattohkc/.

[108] M. J. Todd, K. C. Toh, and R. H. Tütüncü. On the Nesterov–Todd direction in semidefinite programming. *SIAM Journal on Optimization*, 8:769–796, 1998.

[109] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[110] L. Vandenberghe, S. Boyd, and A. El Gamal. Optimal wire and transistor sizing for circuits with non-tree topology. In *Proceedings of the IEEE International Conference on Computer–Aided Design*, San Jose (California), November 1997.

[111] S. J. Wright. *Primal–dual interior–point methods*. SIAM, Philadelphia (Pennsylvania), 1997.

[112] 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.

Index of Authors

Index of Statements

Index of Topics