A second-order cone cutting surface method: complexity and application *

Mohammad R. Oskoorouchi
College of Business Administration,
California State University San Marcos,
San Marcos, CA 92096.
moskooro@csusm.edu

John E. Mitchell†
Department of Mathematical Sciences,
Rensselaer Polytechnic Institute,
Troy, NY 12180.
mitchj@rpi.edu

May 11, 2005

Abstract

We present an analytic center cutting surface algorithm that uses mixed linear and multiple second-order cone cuts. Theoretical issues and applications of this technique are discussed. From the theoretical viewpoint, we derive two complexity results. We show that an approximate analytic center can be recovered after simultaneously adding p second-order cone cuts in $O(p\log(p+1))$ Newton steps, and that the overall algorithm is polynomial. In the implementation part, we apply the algorithm to the eigenvalue optimization problem by relaxing the semidefinite inequalities into multiple second-order cone inequalities. Computational results on randomly generated problems are reported.

*This work has been completed with the support of the partial research grant from the College of of Business Administration, California State University San Marcos, and the University Professional Development Grant.

†This material is based upon work supported by the National Science Foundation under Grant No. 0317323. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Keywords: Second-order cone, semidefinite inequality, cutting plane techniques, eigenvalue optimization.

1 Introduction

The analytic center cutting plane method (ACCPM) is an efficient technique for nondifferentiable optimization problems. The method was first introduced by Sonnevend [22] in 1988. Theoretical issues of this method have been studied in the literature for several settings. The main difference in all methods is the geometry of cuts. In polyhedral cases, single linear, multiple linear, and quadratic cuts have been studied. The theoretical complexity of the method has been reported in several papers. See for instance, Ye [25], Atkinson and Vaidya [2], and Goffin, Luo and Ye [7] for single linear cuts, Ye [26] and Goffin and Vial [8] for multiple linear case, and Luo and Sun [12], Lüthi and Büeler [13], and Sharifi Mokhtarian and Goffin [21] for the case of quadratic cuts.

Recently, there has been a growing interest in ACCPM incorporated with nonpolyhedral models, which we would like to call, analytic center cutting surface method (ACCSM). Complexity results for ACCSM have been studied in Sun, Toh, and Zhao [23], Toh, Zhao, Sun [24], Oskoorouchi and Goffin [17], and Chua, Toh, and Zhao [4] for semidefinite cuts and in Oskoorouchi and Goffin [18] and Basescu [3] for second-order cone cuts.

These methods have been implemented in practice for various applications. Goffin, Gondzio, Sarkissian, and Vial [6] implement ACCPM for solving nonlinear multicommodity flow problems, Mitchell [14] employs this technique for integer programming problems and uses a predictor-corrector interior point method to solve the relaxations, and Elhedhli and Goffin [5] integrate ACCPM with branch-and-price algorithm and implement it for the bin-packing problem and the capacitated facility location problem with single sourcing.

In the nonpolyhedral models, Oskoorouchi and Goffin [19] implement an ACCSM with semidefinite cuts to solve eigenvalue optimization problem and Krishnan and Mitchell [9] use a cut-and-price approach based on a polyhedral approximation of ACCSM to solve the maxcut problem to optimality.

In this paper, we first explore the theoretical issues of integrating mixed linear cuts (LC) and multiple second-order cone cuts (SOCC) with ACCSM. We then apply this technique to eigenvalue optimization. In the theoretical part, we derive two complexity results. First, we show that an approximate analytic center can be achieved after simultaneously adding $p(\geq 1)$ SOCCs in $O(p \log(p+1))$ Newton steps. Then, we obtain the complexity of the overall algorithm by deriving a bound on the total number of linear and second-order cone cuts, and show that ACCSM with mixed linear and multiple second-order cone cuts is a fully polynomial algorithm.

In the implementation part, we consider minimizing the maximum eigenvalue of an affine combination of symmetric matrices. We show that this problem is equivalent to a convex feasibility problem with semidefinite inequalities. In [19], this problem is treated by an ACCSM with semidefinite cuts. The difficulty in this approach is the need to compute the *Gram* matrix at each iteration. Computing this expensive matrix increases the computation time of the algorithm. In order to achieve better computational results and at the same time benefit from the nice properties of ACCSM, we relax the positive semidefinite inequality into multiple second-order cone inequalities. Although this relaxation enlarges the working set and may result in the need for more cuts to find the optimal solution, it saves a lot of computation within each iteration. We report numerical results and compare them with those reported in [19].

2 Preliminaries

Throughout this paper we extensively use some well-known characteristics of second-order cone programming. To keep the paper self-contained, we briefly review the most important properties of the second-order cone. Proofs of statements given in this section and more comprehensive analysis can be found in Alizadeh and Goldfarb [1].

First we introduce our notation: We use uppercase letters for matrices, lowercase letters for vectors and Greek letters for scalars. The space of $n-$dimensional symmetric matrices is denoted by $\mathcal{M}^n$, positive semidefinite matrices by $\mathcal{M}^n_+$, $n-$dimensional real vectors by $\Re^n$, and nonnegative real vectors by $\Re^n_+$. We also use $A \succeq 0$ to indicate that $A \in \mathcal{M}^n_+$. We use $\mathbf{1}$ for an all one vector, and $\mathbf{1}_i$ for a vector with 1 in the ith position and zero elsewhere. The largest eigenvalue of a matrix $A \in \mathcal{M}^n$ is denoted λ_1 and the second largest by λ_2.

For $x, s \in \Re^n$, we use the following notations: "xs" is a component-wise product of x_i and s_i, that is $(xs)_i = x_i s_i$; x^{-1} is the component-wise inverse of x, and $\prod x = \prod_{i=1}^n x_i$. We use ";" for joining two vectors in a column, i.e., $(x; s)$ is a vector in $\Re^{2n}$ made up of vectors x and s joined in a column.

For two matrices A and B, (A, B) makes a matrix by joining them in rows, and $A \oplus B$ makes a matrix by joining A and B in the diagonal

$$A \oplus B = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix};$$

$\|A\|_F$ is the Frobenius norm of A defined via $\|A\|_F = tr(A^T A)$, where "tr" adds the diagonal elements of a symmetric matrix; and finally $A \bullet B$ is the inner product of A and B defined via $A \bullet B = tr(A^T B)$.

The *second-order cone* is defined as follows:

$$\mathcal{S}_n = \{x \in \Re^n : x = (\xi; \bar{x}), \|\bar{x}\| \leq \xi\},$$

where $\|.\|$ is the standard Euclidean norm, ξ is a scalar, $\bar{x} \in \Re^{n-1}$, and n is the dimension of $\mathcal{S}_n$. We use $x \succeq_{\mathcal{S}_n} y$ to indicate that $x - y \in \mathcal{S}_n$. When $n = 1$, $\mathcal{S}_n = \Re_+$.

Associated with the second-order cone, one can define a special case of Euclidean Jordan Algebra. Let $x = (\xi; \bar{x}) \in \Re^n$ and $s = (\sigma; \bar{s}) \in \Re^n$. Define $x \circ s = (x^T s; \bar{u})$, where $\bar{u} \in \Re^{n-1}$, with $\bar{u}_i = \xi s_i + \sigma x_i$.

One can verify that binary operator "$\circ$" is distributive and commutative, but not associative. The unique identity vector of this algebra is represented by $e = (1; 0)$. Clearly, $x \circ e = e \circ x = x$.

Conventionally, we represent $x \circ x$ by x^2. One can verify that every $x \in \mathcal{S}_n$ has a unique square root in $\mathcal{S}_n$.

Spectral decomposition and eigenvalues of x can be defined analogously to the cone of symmetric matrices. For $x = (\xi, \bar{x})$, one has

$$\lambda_1 = \xi + \|\bar{x}\| \text{ and } \lambda_2 = \xi - \|\bar{x}\|.$$

If λ_1 and λ_2 are both nonzero, then x is invertible, with x^{-1} satisfying $x \circ x^{-1} = e$. If λ_1 and λ_2 are both nonnegative, then $x \in \mathcal{S}_n$, and if they are both positive, then $x \in \mathcal{S}_n^\circ := \{x \in \Re^n : x = (\xi; \bar{x}), \|\bar{x}\| < \xi\}$

Using the eigenvalues of x, the following algebraic matrix functions can be defined:

$$\mathbf{tr}(x) := \lambda_1 + \lambda_2 = 2\xi$$

$$\det(x) := \lambda_1 \lambda_2 = \xi^2 - \|\bar{x}\|^2$$

$$\|x\|_F := \sqrt{\lambda_1^2 + \lambda_2^2} = \sqrt{2}\|x\|$$

$\|x\|_2 := \max\{|\lambda_1|, |\lambda_2|\} = |\xi| + \|\bar{x}\|$

Now let $x = (\xi, \bar{x}) \in \Re^n$ with $n \geq 2$ and define

$$Q_x := \begin{pmatrix} \|x\|^2 & 2\xi\bar{x}^T \\ 2\xi\bar{x} & \det(x)I + 2\bar{x}\bar{x}^T \end{pmatrix}.$$

Q_x is a quadratic operator that maps any vector $s \in \Re^n$ to a vector composed of quadratic terms of x. For $n = 1$ with $x = \xi$, we can define the scalar $Q_x = \xi^2$.

In this paper we are dealing with the vectors of the form $\mathbf{x} = (x_1; ...; x_k)$, where x_i is in the second-order cone $\mathcal{S}_{n_i}$. The primal algebra is therefore $\mathcal{S}^k = \mathcal{S}_{n_1} \times ... \times \mathcal{S}_{n_k}$. When there is no ambiguity, we drop the superscript k from $\mathcal{S}^k$. One can extend the above algebraic functions to these block forms. Let $\mathbf{s} = (s_1; ...; s_k)$ and $\mathbf{e} = (e_1; \ldots; e_k)$. Then

$\mathbf{x} \circ \mathbf{s} := (x_1 \circ s_1; ...; x_k \circ s_k)$.

$Q_{\mathbf{x}} := Q_{x_1} \oplus ... \oplus Q_{x_k}$

$\mathbf{tr}(\mathbf{x}) := 2\mathbf{e}^T\mathbf{x} = \sum_{i=1}^k \mathbf{tr}(x_i)$

$\det(\mathbf{x}) := \prod_{i=1}^k \det(x_i)$

$\|\mathbf{x}\|_F^2 := \sum_{i=1}^k \|x_i\|_F^2$

$\|\mathbf{x}\|_2 := \max_i \|x_i\|_2$

$\mathbf{x}^{-1} := (x_1^{-1}; \ldots; x_k^{-1})$

The following lemma presents some important properties of the quadratic operator $Q_{\mathbf{x}}$ in the block form.

Lemma 1 *Let $\mathbf{x} = (x_1; ...; x_k)$ and $\mathbf{s} = (s_1; ...; s_k)$, where $x_i, s_i \in \mathcal{S}_{n_i}$, for $i = 1, ..., k$, and $\mathbf{x}$ is nonsingular. Then*

1. $Q_{\mathbf{x}}\mathbf{x}^{-1} = \mathbf{x}$ *and thus* $Q_{\mathbf{x}}^{-1}\mathbf{x} = \mathbf{x}^{-1}$
2. $Q_{\mathbf{s}}\mathbf{e} = \mathbf{s}^2$
3. $Q_{\mathbf{x}^{-1}} = Q_{\mathbf{x}}^{-1}$
4. $Q_{\mathbf{x}^{-1/2}}\mathbf{x} = \mathbf{e}$
5. $\nabla_{\mathbf{x}}(\log\det(\mathbf{x})) = 2\mathbf{x}^{-1}$ *and* $\nabla^2_{\mathbf{x}}(\log\det(\mathbf{x})) = -2Q_{\mathbf{x}}^{-1}$

6. $Q_{Q_{\mathbf{s}}\mathbf{x}} = Q_{\mathbf{s}} Q_{\mathbf{x}} Q_{\mathbf{s}}$

7. $\det(Q_{\mathbf{x}}\mathbf{s}) = \det^2(\mathbf{x}) \det(\mathbf{s}) = \det(\mathbf{x}^2) \det(\mathbf{s})$

8. $Q_{\mathbf{x}}(\mathcal{S}) = \mathcal{S}$ *and* $Q_{\mathbf{x}}(\mathcal{S}^\circ) = \mathcal{S}^\circ$.

Alizadeh and Goldfarb [1] prove the same lemma for the vectors with a single block. As they note, the extension to k blocks is trivial.

The next lemma generalizes the inequality proved in [18] to the block format:

Lemma 2 *Let* $\mathbf{x} = (x_1; ...; x_k) \in \mathcal{S}$, *where* $x_i \in \mathcal{S}_{n_i}$. *If* $\|\mathbf{x} - \mathbf{e}\|_2 < 1$, *then*

$$\log\det(\mathbf{x}) \geq \mathbf{tr}(\mathbf{x} - \mathbf{e}) - \frac{\|\mathbf{x} - \mathbf{e}\|_F^2}{2(1 - \|\mathbf{x} - \mathbf{e}\|_2)}. \tag{1}$$

Moreover, if $\|\mathbf{x}\|_F \leq 1$, *then*

$$\log\det(\mathbf{x} + \mathbf{e}) \geq \mathbf{tr}(\mathbf{x}) + \|\mathbf{x}\|_F + \log(1 - \|\mathbf{x}\|_F). \tag{2}$$

Proof. Since $\log\det(\mathbf{x}) = \sum \log\det(x_i)$, and (see [18])

$$\log\det(x_i) \geq \mathbf{tr}(x_i - e_i) - \frac{\|x_i - e_i\|_F^2}{2(1 - \|x_i - e_i\|_2)}$$

then

$$\begin{aligned} \log\det \mathbf{x} &\geq \sum_{i=1}^k \mathbf{tr}(x_i - e_i) - \sum_{i=1}^k \frac{\|x_i - e_i\|_F^2}{2(1 - \|x_i - e_i\|_2)} \\ &\geq \mathbf{tr}(\mathbf{x} - \mathbf{e}) - \frac{\sum_{i=1}^k \|x_i - e_i\|_F^2}{2(1 - \max_i \|x_i - e_i\|_2)}. \end{aligned}$$

The first inequality follows from the definition of the Frobenius norm and 2-norm. Now let $\lambda \in \Re^{2k}$ be a vector made up of the eigenvalues of x_i, i.e.,

$$\lambda^T = (\lambda_1(x_1), \lambda_2(x_1), \ldots, \lambda_1(x_k), \lambda_2(x_k)).$$

Observe that

$$\|\mathbf{x}\|_F^2 = \sum_{i=1}^k \|x_i\|_F^2 = \sum_{i=1}^k \left(\lambda_1^2(x_i) + \lambda_2^2(x_i)\right) = \|\lambda\|^2.$$

Now since $\|\mathbf{x}\|_F \leq 1$, then $\|\lambda\| \leq 1$, and therefore

$$\sum_{i=1}^{k}(\log(1+\lambda_1(x_i)) + \log(1+\lambda_2(x_i))) \geq \mathbf{1}^T\lambda + \|\lambda\| + \log(1-\|\lambda\|).$$

The right hand side of the above inequality is clearly equal to that of (2). On the other hand

$$\sum_{i=1}^{k} \log\,(1+\lambda_1(x_i))(1+\lambda_2(x_i)) \quad = \quad \log \prod_{i=1}^{k} \det(e_i + x_i),$$

which is equal to the left hand side of (2). ■

3 Second-order cone cutting surface method

In this section we present an analytic center cutting surface technique that uses multiple second-order cone and single linear cuts.

Let $A_i^T y \preceq_{\mathcal{S}_{q_i}} c_i$, for $i = 1, \ldots, n^{soc}$ be n^{soc} second-order cone inequalities and $A^T y \leq c$ be n^{lc} linear inequalities. Define

$$\mathcal{D} = \{y \in \Re^m : \mathbf{A}^T y \preceq_{\mathcal{S}} \mathbf{c}, \text{ and } A^T y \leq c\},$$

where $\mathbf{A} = (A_1, A_2, \ldots, A_{n^{soc}})$, $\mathbf{c} = (c_1; c_2; \ldots; c_{n^{soc}})$, $A \in \Re^{m \times n^{lc}}$ and $\mathcal{S} = \mathcal{S}_{q_1} \times \ldots \times \mathcal{S}_{q_{n^{soc}}}$.

Suppose that $\mathcal{D}$ is a compact convex set that contains a full dimensional ball with ε radius. We are interested in finding a point in this ball. Let us call $\mathcal{D}$, the *dual set of localization*.

In the algorithm that we describe here, a query point is obtained by computing an approximate analytic center of the set of localization. For the moment, we assume that there exists an oracle that determines either the query point is in the ε-ball, or returns a cut that cuts off the current query point and contains the ε-ball. The cut is either a linear cut (LC) or a set of multiple second-order cone cuts (SOCC). We describe the details of this oracle in Section 5, where we discuss the implementations of the algorithm.

Let us first discuss a computational algorithm for the analytic center of $\mathcal{D}$. Let

$$\phi(\mathbf{s}, s) = \frac{1}{2}\log\det\mathbf{s} + \log\prod s,$$

where

$$\begin{aligned} \mathbf{s} &:= \mathbf{c} - \mathbf{A}^T y \succeq_{\mathcal{S}} 0 \\ s &:= c - A^T y \geq 0. \end{aligned} \tag{3}$$

It is easily verified that ϕ is a strictly concave function on $\mathcal{D}$. Therefore, the maximizer of this function over $\mathcal{D}$ exists and is unique. This maximizer is called the *analytic center* of $\mathcal{D}$. From the KKT optimality conditions y is the analytic center of $\mathcal{D}$ if and only if, there exists $\mathbf{x} = \mathbf{s}^{-1} \succeq_{\mathcal{S}} 0$ and $x = s^{-1} \geq 0$ such that

$$\mathbf{A}\mathbf{x} + Ax = 0, \tag{4}$$

where $\mathbf{s}$ and s satisfy (3).

Corresponding to the optimality condition (4), one can derive the primal set of localization and its associated barrier function. Let

$$\mathcal{P} = \{\mathbf{x} \in \mathcal{S}^{n^{soc}}, x \in \Re_+^{n^{lc}} : \mathbf{A}\mathbf{x} + Ax = 0\},$$

then

$$\psi(\mathbf{x}, x) = -\mathbf{c}^T\mathbf{x} + \frac{1}{2}\log\det\mathbf{x} - c^T x + \log\prod x$$

is strictly concave on $\mathcal{P}$. The Cartesian product of $\mathcal{P}$ and $\mathcal{D}$ gives the primal-dual set of localization. The corresponding barrier function is defined via

$$\Phi(\mathbf{x}, x, \mathbf{s}, s) = \psi(\mathbf{x}, x) + \phi(\mathbf{s}, s).$$

The unique maximizer of ψ over $\mathcal{P}$ and that of Φ over $\mathcal{P} \times \mathcal{D}$ coincide with the analytic center derived for $\mathcal{D}$. Therefore when there is no ambiguity, we refer to this point just as the analytic center.

Definition 3 *An approximate analytic center is a point that satisfies the dual feasibility (3), the primal feasibility (4) and*

$$\eta(\mathbf{x}, x, \mathbf{s}, s) \leq \eta < 1,$$

where $\eta^2(\mathbf{x}, x, \mathbf{s}, s) = \|Q_{\mathbf{x}^{1/2}}\mathbf{s} - \mathbf{e}\|_F^2 + \|xs - \mathbf{1}\|^2$.

An approximate analytic center can be computed using the primal, dual or primal-dual barrier functions. In this paper we use the primal directions to compute the analytic center. The reason is that in practice, adding the cuts returned by the oracle to the primal set of localization can be handled more efficiently than that of the dual or primal-dual sets. Notice that calculating the analytic center of one set yields the center of the other. Therefore, one can switch between the primal, dual and primal-dual sets as needed.

Let us derive the primal direction now. Let a strictly feasible point of $\mathcal{P}$ be given. Since ψ is strictly concave on $\mathcal{P}$, one can efficiently implement Newton's method to maximize $\psi(\mathbf{x}, x)$ over $\mathcal{P}$. Doing so, yields

$$d_{\mathbf{x}} = \mathbf{x} - Q_{\mathbf{x}}\mathbf{s} \tag{5}$$
$$d_x = x - X^2 s, \tag{6}$$

where X is a diagonal matrix made up of vector x, and $\mathbf{s}$ and s satisfy (3), with

$$y = G^{-1} g, \tag{7}$$

where

$$G = \mathbf{A} Q_{\mathbf{x}} \mathbf{A}^T + A X^2 A^T$$
$$g = \mathbf{A} Q_{\mathbf{x}} \mathbf{c} + A X^2 c.$$

Starting from a strictly feasible point, the above direction is implemented at each iteration. One can prove that, this increases the barrier function ψ at least by a constant amount at each step. The rate of convergence becomes quadratic as the iteration gets closer to the analytic center.

Let us present the framework of the analytic center cutting surface algorithm:

Algorithm 1 (ACCSM) *Let $(\mathbf{x}^0, x^0)$, a strictly feasible point of $\mathcal{P}$ be given*

Step 1. *Compute $(\bar{\mathbf{x}}, \bar{x})$, an approximate analytic center of $\mathcal{P}$ using the directions $d_{\mathbf{x}}$ and d_x given in (5) and (6). Compute $\bar{y}$, an approximate center of $\mathcal{D}$ from (7).*

Step 2. *Call the oracle. If $\bar{y}$ is in the ε-ball, stop.*

Step 3. *If the oracle returns a single linear cut $b^T y \leq d$, update $\mathcal{P}$ via*

$$\mathcal{P}^+ = \{\mathbf{x} \succeq_{\mathcal{S}^{n^{soc}}} 0, x \geq 0, \zeta \geq 0 : \mathbf{A}\mathbf{x} + Ax + b\zeta = 0\}.$$

Otherwise go to Step 4.

Step 4. *If the oracle returns multiple second-order cone cuts $\mathbf{B}^T y \preceq_{\mathcal{S}} \mathbf{d}$, update $\mathcal{P}$ via*

$$\mathcal{P}^+ = \{\mathbf{x} \succeq_{\mathcal{S}^{n^{soc}}} 0, \mathbf{z} \succeq_{\mathcal{S}^p} 0, x \geq 0 : \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} + Ax = 0\}.$$

Step 5. *Find a strictly feasible point of $\mathcal{P}^+$ and return to Step 1.*

In the remainder of this section, we elaborate Steps 3-5 in greater detail. Step 2 will be discussed in Section 5.

After adding a cut to the set of localization, whether an LC or a set of SOCCs, the analytic center of the updated set of localization should be recovered. As mentioned before, Newton's method is employed to obtain an approximate center from a strictly feasible point. However, after adding a cut, the only available information is the previous approximate center, which may not be strictly feasible. Therefore, we need an efficient procedure to obtain an initial point for Newton's algorithm. The procedure that we describe here not only gives a strictly feasible point in $\mathcal{P}^+$, but also gives a *warm start* for the Newton directions. In Section 4.1, we show that starting from such a point requires $O(p\log(p+1))$ Newton steps to recover the analytic center after adding p SOCCs. This procedure was initially proposed by Mitchell and Todd [16] in the linear case when a single cut is added.

For the sake of simplicity, we combine the two types of cuts and treat a linear cut as a second-order cone cut of size 1. All algebraic functions defined for the second-order cone can be simplified to be used for the linear inequalities. For example for the linear cut $s := c - a^T y \geq 0$, we define $\det(s) = s^2$. With this definition the potential function term from the SOCC and LC become identical. Therefore "$\phi(\mathbf{s}) = \frac{1}{2}\log\det\mathbf{s}$" works for both cases regardless of the size of n. Notice that with the above definition $\mathbf{tr}(s) = 2s$, and Q_s is simply the scalar s^2. All other definitions as well as Lemma 1 and Lemma 2 hold.

3.1 The updating direction

A strictly feasible point of $\mathcal{P}^+$ updated in Step 3 or 4, can be obtain from the following optimization problem:

$$\begin{array}{ll} \max & \frac{1}{2}\log\det\mathbf{z} \\ s.t. & \\ & \mathbf{A}d_{\mathbf{x}} + \mathbf{B}\mathbf{z} = 0 \\ & \|Q_{\mathbf{w}^{-1/2}} d_{\mathbf{x}}\|_F \leq 1, \end{array}$$

where standard choices for $\mathbf{w}$ include $\mathbf{w} = \mathbf{x}$, $\mathbf{w} = \mathbf{s}^{-1}$, and $\mathbf{w} = \mathbf{x}^{1/2}\mathbf{s}^{-1/2}$. Using the KKT optimality conditions, the updating direction reads

$$d_{\mathbf{x}} = -Q_{\mathbf{w}}\mathbf{A}^T H^{-1}\mathbf{B}\mathbf{z}$$

where

$$H = AQ_{\mathbf{w}}A^T \tag{8}$$

and

$$\mathbf{z}^{-1} = p\mathbf{B}^T H^{-1}\mathbf{B}\mathbf{z}. \tag{9}$$

The updating directions depend on $\mathbf{z}$. Let

$$\varphi(\mathbf{z}) = -\frac{p}{2}\mathbf{z}^T V\mathbf{z} + \frac{1}{2}\log\det\mathbf{z}, \tag{10}$$

where $V = \mathbf{B}^T H^{-1}\mathbf{B}$. Observe that (9) is indeed the optimality condition of

$$\max\{\varphi(\mathbf{z}) : \mathbf{z} \in \mathcal{S}^p\},$$

and since φ is a strictly concave function, Newton's method is most suitable for this problem. Therefore using the quadratic approximation of $\varphi(\mathbf{z}+d_{\mathbf{z}})$, one can derive

$$d_{\mathbf{z}} = (pV + Q_{\mathbf{z}}^{-1})^{-1}(\mathbf{z}^{-1} - pV\mathbf{z})$$

or

$$d_{\mathbf{z}} = Q_{\mathbf{z}^{1/2}}(pQ_{\mathbf{z}^{1/2}}VQ_{\mathbf{z}^{1/2}} + I)^{-1}(\mathbf{e} - pQ_{\mathbf{z}^{1/2}}V\mathbf{z})$$

An advantage of using second-order cone cuts over semidefinite cuts is in the computation of the updating directions. When the set of localization is updated by adding a semidefinite cut, exact Newton iterations cannot be used to obtain a feasible direction (see [19]). This is because, an explicit expression of the gradient of the counterpart of φ in terms of $d_{\mathbf{z}}$, cannot be achieved in the semidefinite case. Oskoorouchi and Goffin [19] overcome this difficulty by using a Pseudo-Newton direction. Although, their technique works fine in practice, however, they lose the nice quadratic convergence property of Newton's method. As shown above, when a semidefinite cut is relaxed into second-order cone cuts, the precise Newton direction $d_{\mathbf{z}}$ can be employed to obtain an updating direction $d_{\mathbf{x}}$ and therefore enhancing the performance of the algorithm in practice.

Observe that the updating directions after adding a single linear cut can be simplified via

$$\begin{aligned} d_{\mathbf{x}} &= -\zeta Q_{\mathbf{w}}\mathbf{A}^T H^{-1} b \\ \zeta &= (b^T H^{-1} b)^{-1/2}. \end{aligned}$$

In the next section we discuss the convergence analysis and complexity of our algorithm.

4 Convergence analysis and complexity

In this section we present two complexity results. First we establish a bound on the number of Newton steps to recover centrality after adding multiple SOCCs, and then we discuss the convergence and complexity of the overall algorithm.

4.1 Complexity of recovering the center

Let $\mathcal{P}$ and $\mathcal{D}$ be the current primal and dual localization sets respectively, and $\bar{\mathbf{x}}$ and $\bar{y}$ be their approximate analytic centers. Let $\mathcal{P}^+$ be the updated primal set as in Step 4 of Algorithm 1. In order to derive the theoretical complexity, we need to make an assumption on the cuts in the dual space.

Assumption 1 *The updated dual set of localization is*

$$\mathcal{D}^+ = \{y \in \mathcal{D} : \mathbf{B}^T y \preceq_{\mathcal{S}^p} \mathbf{B}^T \bar{y}\}.$$

That is, the cuts pass through the center.

Note that while Assumption 1 appears to be necessary in the complexity analysis, it does not interfere with our algorithm in practice. This is because in practice we use the primal space to recover the centrality. As we observed in Section 3.1, in the primal space the location of the cuts does not matter; and the updating direction can always be efficiently obtained using the primal setting.

We need a dual direction. Similar to the primal case, one can obtain a dual updating direction by solving the optimization problem

$$\begin{array}{ll} \max & \frac{1}{2}\log\det(-\mathbf{B}^T d_y) \\ s.t. & \\ & \|Q_{\mathbf{w}^{1/2}}\mathbf{A}^T d_y\|_F \le 1 \end{array}$$

where $d_y = y - \bar{y}$, and the same choices are available for $\mathbf{w}$ as before. From the KKT optimality conditions the optimal d_y reads

$$d_y = -\frac{1}{p} H^{-1}\mathbf{B}\mathbf{t}^{-1}$$

and

$$\mathbf{t} = \frac{1}{p}\mathbf{B}^T H^{-1}\mathbf{B}\mathbf{t}^{-1}, \tag{11}$$

with

$$d_{\mathbf{s}} = -\mathbf{A}^T d_y$$

Note that in view of (11)

$$\mathbf{z} \circ \mathbf{t} = \frac{1}{p}\mathbf{e}. \tag{12}$$

We now fix $\mathbf{w} = \mathbf{s}^{-1}$, so $H = \mathbf{A}Q_{\mathbf{s}^{-1}}\mathbf{A}^T = \mathbf{A}Q_s^{-1}\mathbf{A}^T$. We have the following lemma.

Lemma 4 *Let* $\mathbf{x}^+ = (\mathbf{x} + \alpha d_{\mathbf{x}}; \alpha\mathbf{z})$ *and* $\mathbf{s}^+ = (\mathbf{s} + \alpha d_{\mathbf{s}}; \alpha\mathbf{t})$, *for* $\alpha < 1 - \eta$. *Then*

$$\Phi(\mathbf{x}^+, \mathbf{s}^+) \quad \geq \quad \Phi(\mathbf{x}, \mathbf{s}) + (\alpha + \log(1 - \frac{\alpha}{1-\eta})) + 2p\log\alpha - p\log p.$$

Proof. A similar lemma for the case of single SOCC is proved in [18]. The extension to multiple SOCCs can be done following the same line of proof. We only sketch the proof here.

Let us first derive a bound on the dual barrier function ϕ. Observe that

$$\phi(\mathbf{s}^+) \quad = \quad \phi(\mathbf{s}) + \frac{1}{2}\log\det\alpha\mathbf{t} + \frac{1}{2}\log\det(\mathbf{e} + \alpha Q_{\mathbf{s}^{-1/2}}d_{\mathbf{s}}).$$

Now since $\|Q_{\mathbf{s}^{-1/2}}d_{\mathbf{s}}\|_F \leq \frac{1}{1-\eta}$, in view of (2)

$$\begin{aligned}&\phi(\mathbf{s}^+) \geq \phi(\mathbf{s})\\ &\quad + \frac{1}{2}\log\det\alpha\mathbf{t} + \frac{\alpha}{2}\mathbf{tr}(Q_{\mathbf{s}^{-1/2}}d_{\mathbf{s}}) + \frac{\alpha}{2(1-\eta)} + \frac{1}{2}\log(1 - \frac{\alpha}{1-\eta}).\end{aligned}$$

On the other hand since $\mathbf{A}\mathbf{x} = 0$, then $\frac{1}{2}\mathbf{tr}(Q_{\mathbf{x}^{1/2}}d_{\mathbf{s}}) = 0$, and

$$\begin{aligned}\frac{1}{2}\mathbf{tr}(Q_{\mathbf{s}^{-1/2}}d_{\mathbf{s}}) \quad &= \quad d_{\mathbf{s}}^T(Q_{\mathbf{s}^{-1/2}} - Q_{\mathbf{x}^{1/2}})\mathbf{e}\\ &= \quad d_{\mathbf{s}}^T Q_{\mathbf{s}^{-1/2}}(\mathbf{e} - Q_{\mathbf{s}^{1/2}}\mathbf{x}).\end{aligned}$$

Now since $\mathbf{x}$ is an η-approximate analytic center, then

$$\frac{1}{2}\mathbf{tr}(Q_{\mathbf{s}^{-1/2}}d_{\mathbf{s}}) \geq \frac{-\eta}{2(1-\eta)}.$$

Therefore

$$\phi(\mathbf{s}^+) \geq \phi(\mathbf{s}) + \frac{1}{2}\left(\alpha + \log(1 - \frac{\alpha}{1-\eta})\right) + \frac{1}{2}\log\det\alpha\mathbf{t}. \tag{13}$$

Similarly, one can obtain a bound on the primal barrier function.

$$\psi(\mathbf{x}^+) \geq \psi(\mathbf{x}) + \frac{1}{2}\left(\alpha + \log(1 - \frac{\alpha}{1-\eta})\right) + \frac{1}{2}\log\det\alpha\mathbf{z}. \tag{14}$$

Now adding up (13) and (14), in view of (12) and the following observation

$$\log\det Q_{\mathbf{z}^{1/2}}(\frac{1}{p}\mathbf{z}^{-1}) = \log(\frac{1}{p})^{2p} + \log\det(Q_{\mathbf{z}^{1/2}}\mathbf{z}^{-1}) = -2p\log p$$

we prove the lemma. ■

The next theorem establishes a bound on the gap between the next analytic center and the updated point $(\mathbf{x}^+, \mathbf{s}^+)$.

Theorem 5 *Let $\mathcal{P}^+ \times \mathcal{D}^+$ be the updated primal-dual set of localization and $(\mathbf{x}^+, \mathbf{s}^+)$ be a strictly feasible point defined in Lemma 4. Let $(\mathbf{x}^a, \mathbf{s}^a)$ be the analytic center of $\mathcal{P}^+ \times \mathcal{D}^+$. Then*

$$\Phi(\mathbf{x}^a, \mathbf{s}^a) - \Phi(\mathbf{x}^+, \mathbf{s}^+) \le p\log p - \vartheta(p, \alpha, \eta)$$

where

$$\vartheta(p, \alpha, \eta) = p + 2p\log\alpha + (\alpha + \log(1 - \frac{\alpha}{1-\eta})) - \frac{\eta^2}{4(1-\eta)}$$

Proof. Since

$$\Phi(\mathbf{x}, \mathbf{s}) = -\mathbf{x}^T\mathbf{s} + \frac{1}{2}\log\det(Q_{\mathbf{x}^{1/2}}\mathbf{s})$$

and $(\mathbf{x}, \mathbf{s})$ is an approximate analytic center, in view of (1)

$$\begin{aligned} \Phi(\mathbf{x}, \mathbf{s}) &\ge -\mathbf{x}^T\mathbf{s} + \frac{1}{2}\mathbf{tr}(Q_{\mathbf{x}^{1/2}}\mathbf{s} - \mathbf{e}) - \frac{\eta^2}{4(1-\eta)} \\ &= -n^k - \frac{\eta^2}{4(1-\eta)}. \end{aligned} \tag{15}$$

Now in view of Lemma 4

$$\Phi(\mathbf{x}^+, \mathbf{s}^+) \ge$$
$$-n^k - \frac{\eta^2}{4(1-\eta)} + (\alpha + \log(1 - \frac{\alpha}{1-\eta})) + 2p\log\alpha - p\log p.$$

The lemma now follows from the above inequality and noting that

$$\Phi(\mathbf{x}^a, \mathbf{s}^a) = -n^k - p.$$

■

Theorem 5 proves that after adding $p(\geq 1)$ SOCCs to the set of localization simultaneously, the gap between the primal-dual barrier function at $(\mathbf{x}^+, \mathbf{s}^+)$ and at the new analytic center is bounded by $O(p \log(p+1))$. On the other hand Newton's method is known to increase the primal-dual potential function at least by a constant amount at each iteration. Thus at most $O(p \log(p+1))$ Newton steps are needed to recover centrality after adding p SOCCs.

4.2 Convergence

In this section we derive a bound on the total number of cuts needed to obtain a point in the ε-ball. We establish upper and lower bounds on the dual barrier function after k iterations, and then show that these bounds must cross. The algorithm must terminate before the bounds cross. Let us first establish a bound on the dual barrier function at the analytic center of $\mathcal{D}^+$.

Let ψ^*, ϕ^*, Φ^* be the the optimal values of primal, dual and primal-dual barrier functions respectively. That is $\psi^* = \psi(\mathbf{x}^a)$, $\phi^* = \phi(\mathbf{s}^a)$, and $\Phi^* = \Phi(\mathbf{x}^a, \mathbf{s}^a)$, and let ψ^+, ϕ^+, Φ^+ be the updated barrier functions for $\mathcal{P}^+$, $\mathcal{D}^+$ and $\mathcal{P}^+ \times \mathcal{D}^+$ respectively. From (14)

$$(\psi^+)^* \geq \psi(\mathbf{x}) + \frac{1}{2} \log \det \mathbf{z} + \vartheta_1(p, \alpha, \eta),$$

where

$$\vartheta_1(p, \alpha, \eta) = \frac{1}{2}\left(\alpha + \log(1 - \frac{\alpha}{1-\eta})\right) + p \log \alpha.$$

Since $(\psi^+)^* + (\phi^+)^* = -n - p$, one has

$$(\phi^+)^* \leq -n - p - \psi(\mathbf{x}) - \frac{1}{2} \log \det \mathbf{z} - \vartheta_1(p, \alpha, \eta).$$

On the other hand in view of (15)

$$\psi(\mathbf{x}) \geq \psi^* - \frac{\eta^2}{4(1-\eta)}.$$

Therefore

$$(\phi^+)^* \leq \phi^* - \frac{1}{2} \log \det \mathbf{z} + \frac{\eta^2}{4(1-\eta)} - p - \vartheta_1(p, \alpha, \eta). \tag{16}$$

We now obtain a lower bound on $\frac{1}{2}\log\det \mathbf{z}$. Notice that from (12) $\mathbf{z}^T(\mathbf{B}^T H^{-1}\mathbf{B})\mathbf{z} = \frac{1}{p}\mathbf{z}^T\mathbf{z}^{-1} = 1$. Consequently, since z maximizes $\varphi(z)$, for any $\mathbf{z}^0 \in \mathcal{S}^p$ such that

$$(\mathbf{z}^0)^T(\mathbf{B}^T H^{-1}\mathbf{B})\mathbf{z}^0 = 1, \tag{17}$$

one has

$$\log\det \mathbf{z} \geq \log\det \mathbf{z}^0. \tag{18}$$

Now define

$$\mathbf{z}^0 = \frac{\bar{\mathbf{z}}}{\sqrt{\bar{\mathbf{z}}^T\mathbf{B}^T H^{-1}\mathbf{B}\bar{\mathbf{z}}}} \tag{19}$$

where $\bar{\mathbf{z}} = (\bar{z}_1;\ldots;\bar{z}_p)$ is defined such that $\bar{z}_i = \gamma_i^{-1}e_i$, where

$$\gamma_i = \sqrt{(b_1^i)^T H^{-1} b_1^i}, \tag{20}$$

and b_1^i is the first column of matrix B_i. With this definition observe that

$$\begin{aligned}
\bar{\mathbf{z}}^T\mathbf{B}^T H^{-1}\mathbf{B}\bar{\mathbf{z}} &= \sum_{i=1}^{p}\sum_{j=1}^{p} \bar{z}_i^T B_i^T H^{-1} B_j \bar{z}_j \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} \gamma_i\gamma_j e_i^T B_i^T H^{-1} B_j e_j \\
&\leq p^2
\end{aligned}$$

Now on one hand, $\mathbf{z}^0$ satisfies (17), and on the other hand

$$\log\det \mathbf{z}^0 \geq \log\left(\frac{1}{p}\right)^{2p} + \log\det \bar{\mathbf{z}}$$

Therefore (18) reads

$$\log\det \mathbf{z} \geq -2p\log p + \log\det \bar{\mathbf{z}}$$

and since $\log\det \bar{\mathbf{z}} = \sum \log\det \bar{z}_i = \sum \log \gamma_i^{-2}$, then

$$\log\det \mathbf{z} \geq -2p\log p - 2\sum \log \gamma_i. \tag{21}$$

Inequalities (16) and (21) together yield the following inequality:

$$(\phi^+)^* \leq \phi^* + p\log p + \sum \log \gamma_i + \frac{\eta^2}{4(1-\eta)} - p - \vartheta_1(p,\alpha,\eta).$$

With the arbitrary values $\eta = 0.15$ and $\alpha = 0.60$, we proved the following lemma:

Lemma 6 *If the oracle returns p blocks of SOCC, where $p \geq 1$ and the dual set of localization $\mathcal{D}$ is updated by adding these cuts simultaneously, then the optimal value of the updated dual barrier function has the following upper bound:*

$$(\phi^+)^* \leq \phi^* + \sum_{i=1}^{p} \log \gamma_i + p \log p.$$

Next, we present a lemma to establish an upper bound on the optimal value of the dual barrier function at the k-th iteration. In order to keep this bound simple, we make a scaling assumption.

Assumption 2 *The initial dual set of localization $\mathcal{D}^0$ is the unit ball.*

It is important to note that Assumption 2 is simply a scaling assumption and it is made to keep constants away from the bound.

Lemma 7 *Let n_k be the total number of cuts up to the iteration k. Let γ_i, for $i = 1, \ldots, n_k$ be defined as in (20). Then*

$$(\phi^k)^* \leq \sum_{i=1}^{n_k} \log \gamma_i + n_k \log p_{max},$$

where $p_{max} := \max\{p_i, i = 1, \ldots, n_k\}$.

Proof. Let k be the current iteration. From Lemma 6

$$(\phi^k)^* \leq (\phi^{k-1})^* + \sum_{i=1}^{n_k} \log \gamma_i + p_k \log p_k,$$

where p_k is the number of SOCCs added in the k-th iteration. Since $p_{max} \geq p_k$ for all k, applying this inequality recursively, one has

$$(\phi^k)^* \leq (\phi^0)^* + \sum_{i=1}^{n_k} \log \gamma_i + n_k \log p_{max}.$$

The lemma follows from Assumption 2. ∎

We now define a condition number on a second-order cone cut.

Definition 8 *Let $B^T y \preceq_{\mathcal{S}} d$ be a second-order cone cut and $u \in \Re^m$. Define*

$$\mu = \max_{\|u\| \leq 1} \det(B^T u),$$

This condition number was first defined for semidefinite cuts in [17], and then modified for the second-order cone cuts in [18]. We make the following assumption on the multiple second-order cone cuts added at each iteration:

Assumption 3 *Let $(A_i^j)^T y \preceq c_i^j$, for $i = 1, \ldots, p_j$ be multiple SOCCs added at iteration $j = 1, \ldots, k$ and let*

$$\mu_i^j = \max_{\|u\| \leq 1} \det((A_i^j)^T u) \tag{22}$$

be the condition numbers. Then

$$\mu^j := \min_{i=1,\ldots,p_j} \mu_i^j > 0, \quad \text{for all} \quad j = 1, \ldots, k.$$

We now find a lower bound on the optimal value of the dual barrier function at the kth iteration.

Lemma 9

$$(\phi^k)^* \geq n_k \log(\varepsilon \sqrt{\mu_{min}}),$$

where $\mu_{min} = \min_{j=1,\ldots,k} \mu^j$ and ε is the radius of the ε-ball.

Proof. Let y^c be the center of the ε-ball. For each $i = 1, \ldots, p_j$ and each $j = 1, \ldots, k$, let u_i^j be the vector that achieves the maximum μ_i^j in (22). Then since the dual set of localization $\mathcal{D}$ contains the ε-ball, one has $c_i^j - (A_i^j)^T y \succeq \varepsilon (A_i^j)^T u_i^j$, and so

$$\begin{aligned}
\det(\mathbf{c} - \mathbf{A}^T y^c) &= \prod_{j=1}^{k} \prod_{i=1}^{p_j} \det(c_i^j - (A_i^j)^T y) \\
&\geq \prod_{j=1}^{k} \prod_{i=1}^{p_j} \det(\varepsilon (A_i^j)^T u_i^j) \\
&\geq \varepsilon^{2n_k} \prod_{j=1}^{k} (\mu^j)^{p_j} \\
&\geq \varepsilon^{2n_k} \mu_{min}^{n_k}.
\end{aligned}$$

The proof follows. ■

Combining Lemmas 7 and 9 gives the following inequality:

$$\sum_{i=1}^{n_k} \log \gamma_i \geq n_k \log \frac{\varepsilon \sqrt{\mu_{min}}}{p_{max}}$$

or

$$\frac{1}{n_k}\sum_{i=1}^{n_k}\log\gamma_i^2 \geq \log\left(\frac{\varepsilon\sqrt{\mu_{min}}}{p_{max}}\right)^2. \tag{23}$$

On the other hand, since $\prod \gamma_i^2 \leq \left(\frac{\sum \gamma_i^2}{n_k}\right)^{n_k}$ then

$$\frac{1}{n_k}\sum_{i=1}^{n_k}\log\gamma_i^2 \leq \log\frac{\sum_{i=1}^{n_k}\gamma_i^2}{n_k}. \tag{24}$$

Inequalities (23) and (24) yield

$$n_k\left(\frac{\varepsilon\sqrt{\mu_{min}}}{p_{max}}\right)^2 \leq \sum_{i=1}^{n_k}\gamma_i^2. \tag{25}$$

It remains to bound the right hand side of (25). Let us first make another scaling assumption.

Assumption 4 *Let $A_i^T y \preceq_{\mathcal{S}} A_i^T \bar{y}$, for $i = 1, \ldots, n^{soc}$, and $a_i^T y \leq a_i^T \bar{y}$, for $i = 1, \ldots, n^{lc}$, be second-order cone cuts and linear cuts added to the set of localization. One can assume that*

$$\max_i \{\|A_i\|_F, \|a_j\|\} \leq 1.$$

Assumption 4 is another scaling assumption and does not reduce generality. Notice that if $\|A_i\|_F \leq 1$, then the Euclidean norm of all columns of A_i is less than or equal 1.

Lemma 10 *Let*

$$\mathcal{H}_k = I + \frac{1}{16}\sum_{i=1}^{n_k}(b_1^i)(b_1^i)^T,$$

where b_1^i's are the first columns of matrices B_i of the second-order cone cuts, Then

$$(b_1^i)^T \mathcal{H}_k^{-1} b_1^i \geq \gamma_i^2.$$

Proof. See Lemma 15 of [18]. ■

The next lemma establishes an upper bound on the right hand side of Inequality (25).

Lemma 11

$$\sum_{i=1}^{n_k}\gamma_i^2 \leq 2m(p_{max} + 16)\log(1 + \frac{n_k}{4m})$$

Proof. Let
$$\mathcal{H}_k = \mathcal{H}_{k-1} + \frac{1}{16}\sum_{i=1}^{p_k} b_1^i (b_1^i)^T,$$
with $\mathcal{H}_0 = I$. Let $p_k \geq 2$ (the case of $p_k = 1$ yields a tighter bound, see [18]). One has

$$\det(\mathcal{H}_k) = (1 + \frac{\bar{\gamma}^2}{16}) \det \left(\mathcal{H}_{k-1} + \frac{1}{16}\sum_{i=2}^{p_k} b_1^i (b_1^i)^T \right) \tag{26}$$

where $\bar{\gamma}^2 = (b_1^1)^T \left(\mathcal{H}_{k-1} + \frac{1}{16}\sum_{i=2}^{p_k} b_1^i (b_1^i)^T \right)^{-1} b_1^1$. Now let

$$\mathcal{J} = I + \frac{1}{16}\sum_{i=2}^{p_k} \mathcal{H}_{k-1}^{-1/2} b_1^i (b_1^i)^T \mathcal{H}_{k-1}^{-1/2}.$$

We prove that
$$\mathcal{J}^{-1} \succeq \frac{16}{p_{max} + 16} I. \tag{27}$$

It suffices to show that $x^T \mathcal{J} x \leq \frac{p_{max}+16}{16}$, for all $x \in \Re^m$ with $\|x\| = 1$. This can be seen from the following chain of inequalities and Assumption 4.

$$\begin{aligned} x^T \mathcal{J} x &= \|x\| + \frac{1}{16}\sum_{i=2}^{p_k} (x^T \mathcal{H}_{k-1}^{-1/2} b_1^i)^2 \\ &\leq 1 + \frac{1}{16}\sum_{i=2}^{p_k} (b_1^i)^T \mathcal{H}_{k-1}^{-1} b_1^i \\ &\leq 1 + \frac{1}{16}\sum_{i=2}^{p_k} \|b_1^i\|^2 \end{aligned}$$

Therefore

$$\begin{aligned} \bar{\gamma}^2 &= (b_1^1)^T \mathcal{H}_{k-1}^{-1/2} \mathcal{J}^{-1} \mathcal{H}_{k-1}^{-1/2} b_1^1 \\ &\geq \frac{16}{p_{max} + 16} (b_1^1)^T \mathcal{H}_{k-1}^{-1} b_1^1 \\ &\geq \frac{16\gamma_1^2}{p_{max} + 16} \end{aligned}$$

Therefore (26) reads

$$\det(\mathcal{H}_k) \geq (1 + \frac{\gamma_1^2}{p_{max} + 16}) \det \left(\mathcal{H}_{k-1} + \frac{1}{16}\sum_{i=2}^{p_k} b_1^i (b_1^i)^T \right)$$

Repeating this inequality for $i = 2, \ldots, p_k$, and taking "log" from both sides one has

$$\log\det \mathcal{H}_k \geq \sum_{i=1}^{p_k} \log\left(1 + \frac{\gamma_i^2}{p_{max} + 16}\right) + \log\det \mathcal{H}_{k-1}.$$

On the other hand since $\gamma_i \leq 1$ and $p_{max} \geq 2$,one has

$$\log\left(1 + \frac{\gamma_i^2}{p_{max} + 16}\right) \geq \frac{\gamma_i^2}{2(p_{max} + 16)}.$$

Consequently

$$\log\det \mathcal{H}_k \geq \frac{1}{2}\sum_{i=1}^{p_k} \frac{\gamma_i^2}{p_{max} + 16} + \log\det \mathcal{H}_{k-1}.$$

Notice that, a tighter inequality can be derived when $p_k = 1$.

By repeating the same procedure for all second-order cone cuts, one has

$$\log\det \mathcal{H}_k \geq \frac{1}{2(p_{max} + 16)}\sum_{i=1}^{n_k} \gamma_i^2 + \log\det \mathcal{H}_0$$

Now since $\log\det \mathcal{H}_k \leq m \log(\frac{\mathbf{tr}\mathcal{H}_k}{m})$ and

$$\mathbf{tr}\mathcal{H}_k \leq m + \frac{n_k}{16},$$

therefore

$$\frac{1}{2(p_{max} + 16)}\sum_{i=1}^{n_k} \gamma_i^2 \leq m \log\left(1 + \frac{n_k}{16m}\right).$$

The lemma follows immediately. ■

Combining Lemma 11 and inequality (25), yields our main result.

Theorem 12 *The analytic center cutting surface algorithm (Algorithm 1) finds a point in the ε-ball when the total number of linear and second-order cone cuts reaches to the bound*

$$O\left(\frac{mp_{max}^3}{\varepsilon^2 \mu_{min}}\right)$$

Theorem 12 shows that Algorithm 1 is polynomial with respect to m and p.

5 Application to the maximum eigenvalue function

Consider the following eigenvalue optimization problem

$$\begin{array}{ll} \min & \lambda_1(F_0 + \sum_{i=1}^{m} y_i F_i) \\ s.t. & \\ & \qquad \|y\| \leq \beta, \end{array} \tag{28}$$

where the F_i's are linearly independent symmetric matrices. We implement Algorithm 1 to solve this problem.

Let us first briefly study the Problem (28) and review some important properties of the maximum eigenvalue function. The proofs of statements given in this section and more comprehensive analysis can be found in Overton [20] and Lewis and Overton [11].

It is well-known that that the maximum eigenvalue of a symmetric matrix can be cast as a semidefinite programming problem. That is, if $A \in \mathcal{M}^n$, then

$$\lambda_1(A) = \max\{A \bullet V : \mathbf{tr}V = 1, V \in \mathcal{M}_+^n\},$$

and therefore λ_1 is a convex function of A. With this definition, the objective function of Problem (28) reads

$$h(y) = \max\{F(y) \bullet V : \mathbf{tr}V = 1, V \in \mathcal{M}_+^n\},$$

where $F(y) = F_0 + \sum_{i=1}^{m} y_i F_i$. Although $F(y)$ is a differentiable matrix function, he eigenvalues of $F(y)$ are not differentiable at points where they have multiplicity greater than one. In minimizing the maximum eigenvalue of an affine combination of symmetric matrices, it is often the case that the minimum occurs where $h(y)$ is nondifferentiable. In such cases, one can work with the subdifferential set rather than the gradients. The subdifferential of function $h(y)$ using the Clarke generalized gradient of $\lambda_1(F(y))$ and a chain rule can be derived as

$$\partial h(y) := \{v \in \Re^{\hat{p}} : v_i = (Q^T F_i Q) \bullet V, trV = 1, V \in \mathcal{M}_+^n\},$$

where $Q \in \Re^{n \times \hat{p}}$ is a matrix whose orthonormal columns are the eigenvectors corresponding to the maximum eigenvalue with multiplicity $\hat{p}$. Observe that if the maximum eigenvalue is unique ($\hat{p} = 1$), then the subdifferential set will reduce to a unique vector, which is the gradient of λ_1. In other words, function $h(y)$ is differentiable, if the multiplicity of λ_1 is one.

Let δ be a large positive number such that the optimal objective value h^* satisfies $-\delta < h^* < \delta$. Define

$$\mathcal{F} = \{y = (\tilde{y}, \tau) \in \Re^{m+1} : \|\tilde{y}\| \leq \beta, |\tau| \leq \delta\}.$$

The set $\mathcal{F}$ contains the feasible region of Problem (28) and the optimal objective value h^*, and of course, is bounded and convex. Now let $y^0 = (\tilde{y}^0, \tau^0) \in \mathcal{F}$ be an initial query point. Let us evaluate the objective function h at $\tilde{y}^0$. There are two cases:

1. h is differentiable at $\tilde{y}^0$ ($\hat{p} = 1$).

   In this case matrix Q reduces to a column vector q. The set $\mathcal{F}$ can be replaced by

   $$\mathcal{D} := \{y \in \mathcal{F} : b^T y \leq d, \;\; \tau \leq \min(\delta, h(\tilde{y}^0))\}$$

   where b is a vector in $\Re^{m+1}$, with $b_i = q^T F_i q$, for $i = 1, ..., m$, $b_{m+1} = -1$ and $d = -q^T F_0 q$.

2. h is not differentiable at y^0 ($\hat{p} > 1$). In this case

   $$\mathcal{B}^T y \preceq D, \tag{29}$$

   where $\mathcal{B}^T y = \sum_{i=1}^{m+1} y_i B_i$, $B_i = Q^T F_i Q$, for $i = 1, ..., m$, $B_{m+1} = -I$, and $D = -Q^T F_0 Q$ is a semidefinite inequality that contains the optimal solution of Problem (28). Let

   $$\mathcal{D} = \{y \in \mathcal{F} : \mathcal{B}^T y \preceq D, \;\; \tau \leq \min(\delta, h(\tilde{y}^0))\}.$$

Clearly, in both cases $\mathcal{D}$ contains the optimal solution of Problem (28). Note that Inequality $\tau \leq \min(\delta, h(\tilde{y}^0))$ gives the best upper bound on the optimal objective value in both cases.

We relax this $\hat{p}$-dimensional semidefinite inequality by second-order cone inequalities. Let us first show that how a semidefinite inequality is relaxed into multiple second-order cone inequalities.

Observe that if $A \in \mathcal{M}_+^2$, then positive semidefiniteness of A can be represented as a second-order cone inequality [10]. That is

$$\begin{pmatrix} a & c \\ c & b \end{pmatrix} \succeq 0, \quad \text{if and only if} \quad a + b \geq \left\| \begin{pmatrix} a - b \\ 2c \end{pmatrix} \right\|.$$

The norm inequality is equivalent to

$$\begin{pmatrix} a+b \\ a-b \\ 2c \end{pmatrix} \in \mathcal{S}_3. \tag{30}$$

On the other hand, we know that every 2×2 principle submatrix of a positive semidefinite matrix must be positive semidefinite. Therefore $A \in \mathcal{M}^n_+$ can be relaxed into $\frac{n(n-1)}{2}$ second-order cone inequalities.

Now consider the semidefinite inequality $\sum_{k=1}^m y_k B^k \preceq D$, where $B_k, D \in \mathcal{M}^{\hat{p}}$. Consider the 2×2 principle submatrix in locations i and j, for $i<j$. One has

$$\begin{pmatrix} D_{ii} & D_{ij} \\ D_{ij} & D_{jj} \end{pmatrix} - \sum_{k=1}^m y_k \begin{pmatrix} B^k_{ii} & B^k_{ij} \\ B^k_{ij} & B^k_{jj} \end{pmatrix} \succeq 0$$

or

$$\begin{pmatrix} D_{ii} - \sum y_k B^k_{ii} & D_{ij} - \sum y_k B^k_{ij} \\ D_{ij} - \sum y_k B^k_{ij} & D_{jj} - \sum y_k B^k_{jj} \end{pmatrix} \succeq 0.$$

In view of (30), the above inequality is equivalent to

$$\begin{pmatrix} D_{ii} + D_{jj} - \sum y_k (B^k_{ii} + B^k_{jj}) \\ D_{ii} - D_{jj} - \sum y_k (B^k_{ii} - B^k_{jj}) \\ 2D_{ij} - 2\sum y_k B^k_{ij} \end{pmatrix} \in \mathcal{S}_3. \tag{31}$$

Now let $d^{ij}_{soc} = (D_{ii}+D_{jj}; D_{ii}-D_{jj}; 2D_{ij})$ and $B^{ij}_{soc} \in \Re^{m\times 3}$ be a matrix whose kth row is defined via $(B^k_{ii}+B^k_{jj}, B^k_{ii}-B^k_{jj}, 2B^k_{ij})$. Then (31) reads

$$d^{ij}_{soc} - (B^{ij}_{soc})^T y \in \mathcal{S}_3,$$

for all $i<j$. Therefore the semidefinite inequality (29) can be relaxed into $\frac{\hat{p}(\hat{p}-1)}{2}$ second-order cone inequalities, and the set of localization be enlarged via

$$\mathcal{D} = \{y \in \mathcal{F} : \mathbf{B}^T y \preceq_{\mathcal{S}} \mathbf{d},\ \ and\ \ \tau \le \min(\delta, h(\tilde{y}^0))\}$$

where $\mathbf{B} = (B_1\ \ B_2\ \ \ldots\ \ B_p)$, $\mathbf{d}_{soc} = (d_1; d_2; \ldots; d_p)$, and $\mathcal{S} = \mathcal{S}_3 \times \mathcal{S}_3 \times \ldots \times \mathcal{S}_3$ composed of p blocks, where $p = \frac{\hat{p}(\hat{p}-1)}{2}$ and B_k's and d_k's are as defined above.

Notice that one can generate far more SOCCs from a single SDC than just those coming from pairs of eigenvectors from a particular eigenbasis. This can be done by multiplying an appropriate $\hat{p}\times 2$ matrix U by the semidefinite cut to give $U^T(D - \mathcal{B}^T y)U$ and requiring this 2×2 matrix to be positive semidefinite. Different U's would give different combinations of eigenvectors. Unfortunately, it is not clear which are the useful U's. The procedure described above can be regarded as putting one 1 in each column of U.

We implement Algorithm 1 with the set of localization $\mathcal{D}$. Depending on which one of the above cases occurs, a single linear cut or a set of multiple second-order cone cuts will be added to $\mathcal{D}$ and the upper bound is updated. Therefore at the kth iteration the set of localization has the following structure:

$$\mathcal{D}^k = \{y \in \Re^{m+1} : (\mathbf{A}^k)^T y \preceq_{\mathcal{S}} \mathbf{c}^k, (A^k)^T y \leq c^k, \tau \leq \theta^k\},$$

where $\mathbf{A}^k$ contains $n^{soc} = 2 + \sum_{i=1}^k p_i$ blocks of SOCCs, $A^k \in \Re^{(m+1)\times n^{lc}}$ contains n^{lc} linear cuts, and $\theta^k = \min(\theta^{k-1}, h(\tilde{y}^{k-1}))$, the best upper bound.

The set $\mathcal{D}^k$ is a compact convex set that is described by linear and second-order cone inequalities. The upper bound cut $\tau \leq \theta$ is a linear cut and could be incorporated into the linear inequalities, however, we prefer to study this cut separately. This is because adding the subdifferential cuts (LC and SOCC) causes the analytic center to get close to the upper bound cut. As the algorithm proceeds, the distance between the analytic center and the upper bound cut vanishes. In order to avoid this phenomenon, we use a weighted analytic center as the query point, and the weight, which is equal to the total number of cuts $n^{soc} + n^{lc}$, is placed on the upper bound cut.

6 Computational experience and conclusions

In this section we illustrate some preliminary computational results of Algorithm 1 when implemented on Problem (28). The data for the test problems are randomly generated from normal distributions with different means and standard deviations. We use the following stopping criterion to terminate the algorithm:

$$\frac{\theta^k - \tau}{\theta^k} < 5.0 \times 10^{-3}.$$

This criterion utilizes the analytic property of weighted analytic centers. That is since the weight on the upper bound cut is equal to the dimension of the current subgradient cut, τ approaches the upper bound only when the localization set $\mathcal{D}^k$ is small enough.

As mentioned earlier, each call to the oracle returns either a linear cut or a p-dimensional semidefinite cut, where the SDC can be relaxed into

Table 1: $m = 100$, $n = 50$, $d = 6\%$

ratio	calls	LC	SDC	SOCC	dim(cut)	h^*	CPU (sec)
	22	0	22	-	198	9.3903	675
.500	49	0	-	49	1980	9.3790	899
	63	0	-	63	530	9.3940	269
	261	227	34	-	318	9.3804	2845
.005	249	205	-	44	347	9.4032	122
	241	194	-	47	330	9.3862	114

multiple SOCCs. We illustrate the performance of our algorithm with the full relaxation ($\frac{p(p-1)}{2}$ SOCCs) and with a partial relaxation (p SOCCs), and compare the results with SDC. The partial relaxation is chosen by randomly selecting a subset of the SOCCs. We can't sort the SOCCs based on violation, because all of the potential SOCCs given by pairs of eigenvectors have the same violation.

The tables are structured such that each table represents numerical results of implementing ACCSM on a randomly generated problem of size m, dimension n and density d. In determining the multiplicity of the maximum eigenvalue, we look at a ratio between the two maximum eigenvalues λ_1 and λ_2, namely $\frac{\lambda_1 - \lambda_2}{\lambda_1}$. If this ratio is close to zero, the semidefinite cut is more powerful than a linear cut; if the ratio is close to one, the cheaper linear cut is almost as powerful as the semidefinite cut. A threshold for this ratio is represented in the first column. When this threshold is small (typically 0.005 or less), the oracle returns more LC than SDC. We use various thresholds to show the performance of ACCSM when dealing with different types of cuts.

Each segment of the table has three rows. The first row represent the result of the algorithm when the SDC is used. The second and third rows show the data when the SDC is relaxed into $\frac{p(p-1)}{2}$ and p SOCCs respectively. For each problem set we look at the number of calls to the oracle, the number of LCs, SDCs and blocks of SOCCs returned by the oracle. The column under "dim(cut)" represent the sum of the number of linear cuts and the dimensions of the SDC or SOCC. We report the upper bound $h(y^*)$ of the optimal value h^* and the cpu time in seconds in the last two columns.

We use a MATLAB code and run our algorithm on a 1.60 GHz PC computer with 384 MB of RAM. We note that our intention here is only to

Table 2: $m = 100$, $n = 100$, $d = 6\%$

ratio	calls	LC	SDC	SOCC	dim(cut)	h^*	CPU (sec)
	92	17	75	-	297	16.110	2442
.050	118	14	-	104	960	16.117	457
	116	13	-	103	387	16.127	265
	248	201	47	-	330	16.110	1583
.005	241	144	-	57	385	16.116	220
	242	190	-	52	302	16.113	206

Table 3: $m = 100$, $n = 100$, $d = 99\%$

ratio	calls	LC	SDC	SOCC	dim(cut)	h^*	CPU (sec)
	34	26	8	-	72	69.660	278
.500	38	27	-	11	150	69.626	56
	32	22	-	10	45	69.938	17
	32	32	0	-	32	69.578	8
.005	31	30	-	1	32	69.643	9
	31	30	-	1	32	69.643	9

Table 4: $m = 100$, $n = 300$, $d = 6\%$

ratio	calls	LC	SDC	SOCC	dim(cut)	h^*	CPU (sec)
	41	0	41	-	410	30.594	5054
.500	97	0	-	97	4322	30.582	4809
	119	0	-	119	1064	30.596	1906
	230	158	72	-	363	30.586	8339
.005	251	172	-	79	415	30.590	1335
	245	153	-	92	323	30.584	1292

Table 5: $m = 100$, $n = 300$, $d = 99\%$

ratio	calls	LC	SDC	SOCC	dim(cut)	h^*	CPU (sec)
	160	1	159	-	1522	39.879	32160
.500	29	8	-	21	165	40.361	127
	27	5	-	22	64	40.411	74
	250	250	0	-	250	39.960	409
.005	21	20	-	1	22	40.412	34
	21	20	-	1	22	40.412	34

explore the computational advantage of SOCCs over SDCs and not to show the ability of ACCSM. This is why we test only moderate size problems rather than large problems. For the latter a computer code in a lower level language such as C^{++} would be more appropriate.

A common observation is that when "ratio" is large, using the original SDC requires less calls to the oracle than using the SOCC relaxation. On the other hand, although using the second-order cone relaxation increases the number of calls to the oracle, the cpu time is significantly improved. This is because semidefinite cuts are stronger than second-order cone cuts but a lot more expensive.

It appears that using a partial relaxation (p SOCC) on SDC has a computational advantage over the full relaxation. This relaxation provides a balance between the strength and the price of the cuts. Applying the full relaxation requires adding too many cuts simultaneously. This necessitates

Table 6: $m = 500$, $n = 50$, $d = 99\%$

ratio	calls	LC	SDC	SOCC	dim(cut)	h^*	CPU (sec)
	28	5	23	-	77	7.0673	1170
.500	35	6	-	29	191	7.0679	474
	37	7	-	30	83	7.0681	224
	42	42	0	-	42	7.0898	24
.005	40	39	-	1	41	7.0659	97
	40	39	-	1	41	7.0659	97

more Newton steps to recover centrality (see Theorem 5), and therefore escalates the cpu time. Surprisingly, using the full relaxation does not reduce the number of calls to the oracle.

References

[1] F. Alizadeh and D. Goldfarb, *Second-order cone programming*, Mathematical Programming, 95(1), 2003, pp. 3–51.

[2] D. S. Atkinson and P. M. Vaidya, *A cutting plane algorithm that uses analytic centers*, Mathematical Programming, series B, 69 (1995), pp. 1–43.

[3] V. L. Basescu, *An analytic center cutting plane method in conic optimization*, Doctoral Thesis, Rensselaer Polytechnic Institute, Troy, NY 12180, August 2003.

[4] S. K. Chua, K. C. Toh, and G. Y. Zhao, *An analytic center cutting plane method with deep cuts for semidefinite feasibility problems*, JOTA, to appear.

[5] S. Elhedhli and J.-L. Goffin, *The integration of an interior point cutting plane method with branch and price algorithm*, Mathematical Programming, Volume 100, Number 2, June 2004 pp. 267–294.

[6] J.-L. Goffin, J. Gondzio, R. Sarkissian, and J.-P. Vial, *Solving nonlinear multicommodity flow problems by the analytic center cutting plane method*, Mathematical programming, 76(1) (1997), pp. 131–154.

[7] J.-L. Goffin, Z.-Q. Luo, and Y. Ye, *Complexity analysis of an interior cutting plane for convex feasibility problems*, SIAM Journal on Optimization, 6 (1996), pp. 638–652.

[8] J.-L. Goffin and J.-P. Vial, *Multiple cuts in the analytic center cutting plane methods*, SIAM Journal on Optimization, 11 (2000), pp. 266–288.

[9] K. Krishnan and J. E. Mitchell, *Semidefinite cut-and-price approaches for the maxcut problem*, AdvOL-Report No. 2004/3, March 2004, Hamilton, Ontario, Canada.

[10] S. Kim, M. Kojima and M. Yamashita, *Second Order Cone Programming Relaxation of a Positive Semidefinite Constraint*, Optimization Methods and Software, Vol.18 (5) 535-541 (2003).

[11] A. S. Lewis and M. L. Overton, *Eigenvalue Optimization*, Acta Numerica, (1996), pp. 149–190.

[12] Z.-Q. Luo and J. Sun, *An analytic center based on column generation algorithm for convex quadratic feasibility problems*, SIAM Journal on Optimization, 9(1) (1998) pp. 217–235.

[13] H.J. Lüthi and B Büeler, *The Analytic Center Quadratic Cut Method (ACQCM) for Strongly Monotone Variational Inequality Problems*, SIAM Journal on Optimization, 10(2) (2000), pp. 415–426.

[14] J. E. Mitchell, *Computational experience with an interior point cutting plane algorithm*, SIAM Journal on Optimization, 10(4) (2000), pp. 1212–1227.

[15] J. E. Mitchell, *Polynomial interior point cutting plane methods*, Optimization Methods and Software, 18(5):507–534, 2003.

[16] J. E. Mitchell and M.J. Todd, *Solving combinatorial optimization problems using Karmarkar's algorithm*, Mathematical Programming, 56 (1992), pp. 245–284.

[17] M. R. Oskoorouchi and J. L. Goffin, *The analytic center cutting plane method with semidefinite cuts*, SIAM Journal on Optimization, 13 (4) (2003), 1029–1053.

[18] M. R. Oskoorouchi and J. L. Goffin, *An interior point cutting plane method for the convex feasibility problem with second-order cone inequalities*, Mathematics of Operations Research, forthcoming.

[19] M. R. Oskoorouchi and J. L. Goffin, *A matrix generation approach for eigenvalue optimization*, Technical report, CoBA, Cal State San Marcos, September 2004.

[20] M. L. Overton, *Large-scale optimization of eigenvalues*, SIAM Journal on Optimization, 2 (1992), pp. 88–120

[21] F. Sharifi Mokhtarian and J.-L. Goffin, *An analytic center quadratic cut method for the convex quadratic feasibility problem*, Math. Program., Ser. A 93 (2002) 2, pp. 305–325.

[22] G. Sonnevend, *New algorithms in convex programming based on a notation of center and on rational extrapolations*, International Series of Numerical Mathematics, Birkhauser Verlag, Basel, Switzerland, 84 (1988), pp. 311–327.

[23] J. Sun, K. C. Toh, and G. Y. Zhao, *An analytic center cutting plane method for semidefinite feasibility problems*, Mathematics of Operations Research, Volume: 27 (2), May 2002 pp. 332–346

[24] K. C. Toh, G. Y. Zhao, and J. Sun, *A multiple-cut analytic center cutting plane method for semidefinite feasibility problems*, SIAM Journal on Optimization, 12 (2002) 4, pp. 1126–1146

[25] Y. Ye, *A potential reduction algorithm allowing column generation*, SIAM Journal on Optimization, 2 (1992), pp. 7–20.

[26] Y. Ye, *Complexity analysis of the analytic center cutting plane method that uses multiple cuts*, Mathematical Programming, 78 (1997), pp. 85–104.