McMaster University

Advanced Optimization Laboratory

Title:

A conic interior point decomposition approach for semidefinite programming

Authors:

Kartik K. Sivaramakrishnan, Gema Plaza
and Tamás Terlaky

AdvOL-Report No. 2005/22

December 2005, Hamilton, Ontario, Canada

Mathematical Programming manuscript No.
(will be inserted by the editor)

Kartik K. Sivaramakrishnan[1] · Gema Plaza[2] · Tamás Terlaky[3]

A conic interior point decomposition approach for semidefinite programming

Received: December 2005

Abstract. We describe a conic interior point decomposition approach for solving a large scale semidefinite programs (SDP) whose primal feasible set is bounded. The idea is to solve such an SDP using existing primal-dual interior point methods, in an iterative fashion between a *master problem* and a *subproblem*. In our case, the master problem is a mixed conic problem over linear and smaller sized semidefinite cones. The subproblems is smaller structured semidefinite program that either returns a column or a small sized matrix depending on the multiplicity of the minimum eigenvalue of the dual slack matrix associated with the semidefinite cone. We motivate and develop our conic decomposition methodology on semidefinite programs and also discuss various issues involved in an efficient implementation. Computational results on several well known classes of semidefinite programs are presented.

Kartik K. Sivaramakrishnan: Department of Mathematics, P.O. Box 8205, North Carolina State University, Raleigh, NC 27695-8205, USA. e-mail: `kksivara@ncsu.edu`. Research supported by SHARCNET and a startup grant from North Carolina State University.

Gema Plaza: Department of Statistics & Operations Research, University of Alicante, Apartado de Correos 99, 03080 Alicante, Spain. e-mail: `gema@optlab.mcmaster.ca`. Research supported by Grant CTBPRB/2003/165 from FPI Program of GVA, Spain.

Tamás Terlaky: Advanced Optimization Laboratory, Department of Computing & Software, McMaster University, Hamilton, Ontario L8S 4K1, Canada. e-mail: `terlaky@mcmaster.ca`. Research supported by NSERC, Canada Research Chair Program, and MITACS.

Mathematics Subject Classification (1991): 20E28, 20G40, 20C20

Correspondence to: Kartik K. Sivaramakrishnan

1. Introduction

Conic programming is a natural extension of linear programming and is the problem of minimizing a linear function over a feasible set given by the intersection of an affine subspace and a closed convex cone. Three important classes of conic programming include linear programming (LP), second order cone programming (SOCP), and semidefinite programming (SDP). SDP is the most important problem in this class and it also has a variety of applications in control; approximation algorithms for combinatorial optimization; planning under uncertainty including robust optimization; and more recently in polynomial optimization and lift-and-project schemes for solving integer and nonconvex problems. The books by Ben Tal & Nemirovskii [3] and Boyd & Vandenberghe [7] discuss a variety of applications of conic programming in science and engineering. Other references for SDP include the SDP handbook edited by Wolkowicz et al. [54], the monograph by De Klerk [11], and the habilitation thesis of Helmberg [17]. A recent survey of SDP based approaches in combinatorial optimization can be found in Krishnan & Terlaky [32].

Interior point methods (IPMs) are currently the most powerful techniques for solving large scale conic problems. A nice exposition of the theoretical foundations can be found in Renegar [50]. In addition to the ongoing theoretical work that derived complexity estimates and convergence guarantees/rates for such algorithms (De Klerk [11] and Peng et al. [49]), many groups of researchers have also implemented these algorithms and developed public domain software packages that are capable of solving mixed conic optimization problems over linear, second order, and semidefinite cones of ever increasing size and diversity. Two recent solvers include SeDuMi developed by Sturm and further developed by the AdvOL group at McMaster [52], and SDPT3 by Tütüncü et al. [53]. However, a note is due with respect to the sizes of linear, second order, and semidefinite problems that can be solved using IPMs in practice. Let m denote the number of equality constraints in the conic problem and n be the number of variables. While, very large LPs ($m \leq 100,000$, $n \leq 10,000,000$) and large SOCPs ($m \leq 100,000$, $n \leq 1,000,000$) are routinely solved using IPMs, current IPM technology can only handle SDPs that have both the size of the semidefinite cone and $m \leq 10,000$; this roughly translates to $n \leq 50,000,000$. In each iteration of an IPM for a conic problem, one needs to form the Schur matrix $M = AD^2A^T$, and having done so, solve a system of linear equations with M as the coefficient matrix. In the LP and SOCP cases, forming the Schur matrix is fairly cheap requiring at most $O(m^2n)$ flops. However, in the SDP case, forming the Schur matrix is the most time consuming operation since the variables are symmetric matrices. Moreover, in the LP and SOCP cases with small sized cones, the Schur matrix inherits the sparsity of the original coefficient matrix and one can employ sparse Cholesky solvers to quickly factorize this matrix. Similarly, in the SOCP case with large cones, the Schur matrix can be expressed as the sum of a sparse matrix and some low rank updates where the rank of the update depends on the number of large second order cones; in this case one can still employ the Sherman-Morrison-Woodbury update formula when solving the Schur complement equation (see Andersen et al. [2] and Tütüncü et al. [53]).

No such techniques are available in the general SDP case where the Schur matrix is always dense (regardless of the sparsity in A). All this makes the SDP a more difficult problem to solve! A recent survey of IPMs and several first order approaches for SDP can be found in Monteiro [39].

We present a conic interior point based decomposition methodology in this paper to solve a semidefinite program, over one semidefinite cone and whose primal feasible region is bounded, in an iterative fashion between a master problem and a subproblem in the usual Dantzig-Wolfe setting (see Bertsekas [4], Lasdon [34], and Nesterov [41]). Our master problem is a mixed conic problem over linear and smaller sized semidefinite cones that is solved using existing primal-dual IPMs. The subproblem is a smaller semidefinite program with a special structure that computes the smallest eigenvalues and associated eigenvectors of the dual slack matrix associated with the semidefinite cone. The master problem and the subproblem are much smaller than the original problem. Hence, even if the decomposition algorithm requires several iterations to reach the solution, it is usually faster and requires less memory than algorithms that attack the original undecomposed problem. In the future, we will extend our approach to structured semidefinite programs, and we present a discussion of our approach on *block-angular* semidefinite programs in Section 9.

Our conic interior point approach in this paper extends the polyhedral interior point decomposition approach developed in Krishnan [27] and Krishnan & Mitchell [28,29]. It also complements two other nonpolyhedral interior point decomposition approaches for SDP: the spectral bundle method of Helmberg and

Rendl [19] and the matrix generation ACCPM approach of Oskoorouchi and Goffin [43]. Excellent computational results with the spectral bundle method were reported in Helmberg [18]. The aforementioned approaches differ in the procedure used in computing the dual prices, and the proposals returned by the subproblems. A comparison of these approaches can be found in Krishnan and Mitchell [30]. Now we list some of the major differences between our decomposition method, the spectral bundle method, and ACCPM.

1. **Comparisons with the spectral bundle approach:** In each iteration of our algorithm, we maintain a feasible solution for the primal SDP. The spectral bundle approach is essentially a dual approach, where one is approximately solving a Lagrangian relaxation of the primal problem using the bundle method for nonsmooth optimization, and the method generates a feasible solution for the primal SDP only in the limit. In combinatorial applications (see Goemans-Williamson [14]) and branch-cut-price algorithms for integer programming, where the semidefinite relaxations are solved approximately, it is useful to maintain a feasible solution to the primal SDP in every iteration to obtain bounds.
2. **Comparisons with the matrix generation ACCPM approach:** The matrix generation ACCPM approach is a primal only approach, where the primal iterates are updated at each iteration and the dual prices computed only approximately. Thus, our primal-dual approach enjoys inherent advantages over the ACCPM approach since the dual prices are given to the various

subproblems at each iteration, and thus they are directly responsible for the columns/matrices generated in the primal master problem.

Our paper is organized as follows: Section 2 introduces the SDP and its optimality conditions. Section 3 develops the conic decomposition methodology. Section 4 presents an overview of our stabilized column/matrix generation technique aimed at improving the convergence of the decomposition approach introduced in Section 3. We present warm-start procedures in Section 5 that enable us to restart the primal and dual master problems with strictly feasible starting points after column/matrix generation. Some aggregation schemes to keep the size of the master problem small are discussed in Section 6. The overall primal-dual IPM based decomposition approach is presented in Section 7. We present computational results on several well known classes of SDPs in Section 8. Extensions to our decomposition approach to block-angular semidefinite programs are considered in Section 9. Finally, we conclude with our observations and future work in Section 10.

2. The semidefinite programming problem

Consider the semidefinite program

$$\begin{aligned} \min\ & C \bullet X \\ \text{s.t.}\ & \mathcal{A}(X) = b, \\ & I \bullet X = 1, \\ & X \succeq 0, \end{aligned} \tag{SDP}$$

with dual

$$\begin{array}{ll} \max & b^T y + z \\ \text{s.t.} & \mathcal{A}^T y + zI + S = C, \\ & S \succeq 0. \end{array} \tag{SDD}$$

A quick word on notation: $X, S \in \mathcal{S}^n$ the space of real symmetric $n \times n$ matrices; $b \in I\!R^m$; $C \bullet X = \text{trace}(CX) = \sum_{i,j=1}^{n} C_{ij} X_{ij}$ is the Frobenius inner product of matrices in $\mathcal{S}^n$. The linear operator $\mathcal{A} : \mathcal{S}^n \to I\!R^m$ and its adjoint $\mathcal{A}^T : I\!R^m \to \mathcal{S}^n$ are:

$$\mathcal{A}(X) = \begin{bmatrix} A_1 \bullet X \\ \vdots \\ A_m \bullet X \end{bmatrix} \text{ and } \mathcal{A}^T y = \sum_{i=1}^{m} y_i A_i,$$

where the matrices $A_i \in \mathcal{S}^n$, $i = 1, \ldots, m$, and $C \in \mathcal{S}^n$ are the given problem parameters. The constraints $X \succeq 0$, $S \succeq 0$, ($X \succ 0$, $S \succ 0$) are the only nonlinear constraints in the problem requiring that X and S are symmetric positive semi-definite (positive definite) matrices. Finally, $\bar{n} = \frac{n(n+1)}{2}$ and svec : $\mathcal{S}^n \to I\!R^{\bar{n}}$ is the linear operator

$$\text{svec}(X) = \left[X_{11}, \sqrt{2}X_{12}, X_{22}, \ldots, \sqrt{2}X_{1n_i}, \ldots, \sqrt{2}X_{n-1,n} X_{n,n}\right]^T.$$

A detailed discussion of matrix analysis and related notation can be found in Horn & Johnson [22].

We assume that our SDP satisfies the following assumptions

Assumption 1 *The matrices A_i, $i = 1, \ldots, m$ are linearly independent in $\mathcal{S}^n$.*

In particular, this implies $m \leq \binom{n+1}{2}$.

Assumption 2 *Both (SDP) and (SDD) have strictly feasible points, namely the sets* $\{X \in \mathcal{S}^n : \mathcal{A}(X) = b, X \succ 0\}$ *and* $\{(y, z, S) \in I\!R^m \times I\!R \times \mathcal{S}^n : \mathcal{A}^T y + zI + S = C, S \succ 0\}$ *are nonempty.*

Assumption 2 guarantees that both (SDP) and (SDD) attain their optimal solutions X^* and (y^*, z^*, S^*), and their optimal values are equal, i.e. $C \bullet X^* = b^T y^* + z^*$. Thus, at optimality the duality gap $X^* \bullet S^* = 0$.

Assumption 3 *One of the constraints in the primal problem is* $I \bullet X = 1$.

Although, Assumption 3 appears restrictive, we must mention that our decomposition principle can also be applied when the trace constraint is not explicitly present in (SDP) but is implied by the other constraints in $\mathcal{A}(X) = b$. In fact, any SDP whose primal feasible set is bounded can be reformulated to satisfy Assumption 3 (see Helmberg [17]). The assumption also enables one to rewrite (SDD) as an eigenvalue optimization problem as follows.

We can write down the Lagrangian dual to (SDP) transferring all the equality constraints other than $I \bullet X = 1$ into the objective function via Lagrangian multipliers y_i, $i = 1, \ldots, m$. This gives the following problem

$$\max_y \{b^T y \;+ \min_{I \bullet X = 1, X \succeq 0} (C - \mathcal{A}^T y) \bullet X\}. \tag{1}$$

Assumption 2 ensures that this problem is equivalent to (SDP). Using the variational characterization of the minimum eigenvalue function, the quantity in the inner minimization can be expressed as $\lambda_{\min}(C - \mathcal{A}^T y)$. We can therefore rewrite (1) as

$$\max_y \{b^T y + \lambda_{\min}(C - \mathcal{A}^T y)\}. \tag{2}$$

This is an eigenvalue optimization problem. We shall return to the formulation (2), when we introduce the conic decomposition scheme for (SDP) in Section 3. We now introduce some shorthand notation. Let

$$f(y) = \lambda_{\min}(C - \mathcal{A}^T y)$$

and let $\theta(y) = b^T y + f(y)$. Thus, (SDD) can be written as the following eigenvalue optimization problem

$$\max_{y \in I\!R^m} \theta(y). \tag{3}$$

The function $f(y)$ is concave. It is also non-differentiable, precisely at those points where $f(y)$ has a multiplicity greater than one. Let us consider a point y, where $f(y)$ has a multiplicity r. Let p_i, $i = 1, \ldots, r$ be an orthonormal set of eigenvectors corresponding to $f(y)$. Also, $P \in I\!R^{n \times r}$ with $P^T P = I_r$ is the matrix, whose ith column is p_i. Any normalized eigenvector p corresponding to $f(y)$ can be expressed as $p = Px$, where $x \in I\!R^r$, with $x^T x = 1$. The superdifferential of $f(y)$ at this point is then given by

$$\begin{aligned} \partial f(y) &= \text{conv}\{-\mathcal{A}(pp^T) : p = Px, \ \ x^T x = 1\} \\ &= \{-\mathcal{A}(PVP^T) : V \in \mathcal{S}^r, \ \ I \bullet V = 1, \ \ V \succeq 0\} \end{aligned} \tag{4}$$

Here conv denotes the convex hull operation. The equivalence of the two expressions can be found in Overton [45]. Each member of (4) is called a supergradient.

Using the expression (4) one can write the directional derivative $f'(y,h)$ for $f(y)$ along the direction $h \in I\!R^m$ as follows

$$\begin{aligned} f'(y,h) &= \min_{s \in \partial f(y)} s^T h \\ &= \min_{V \succeq 0,\ I \bullet V = 1} (-P^T \mathcal{A}^T h P) \bullet V \\ &= \lambda_{\min}(-P^T \mathcal{A}^T h P). \end{aligned} \tag{5}$$

A discussion of these results for the minimum eigenvalue function can be found in the books by Bonnans and Shapiro [5] and Hiriart-Urruty and Lemaréchal [21].

We conclude this section with a discussion of the optimality conditions for SDP and the rank of an optimal solution.

Theorem 1. *Let X^* and (y^*, z^*, S^*) be feasible solutions in (SDP) and (SDD), respectively. They are optimal if and only if they satisfy the complementary slackness conditions $X^* S^* = 0$.*

The complementary slackness conditions imply that X^* and S^* share a common eigenspace at optimality (Alizadeh et al. [1]). Also, the range space of X^* is the null space of S^*. Theorem 2, due to Pataki [47], gives an upper bound on the dimension of this subspace.

Theorem 2. *Let $X^* = PVP^T$ be an optimal solution to (SDP) with $P \in I\!R^{n \times r^*}$ and $V \succ 0$. We have*

$$r^* \le \lfloor \frac{-1 + \sqrt{1 + 8m}}{2} \rfloor.$$

3. A conic decomposition approach for SDP

We will develop a cutting plane algorithm for (SDD) based on its eigenvalue formulation given in (2).

Since $f(y)$ is a concave function, we have the following set of inequalities

$$\begin{aligned} f(y) &\leq f(\hat{y}) + \lambda_{\min}(-P^T \mathcal{A}^T h P) \\ &\leq \lambda_{\min}(P^T(C - \mathcal{A}^T y)P), \end{aligned}$$

where $h = (y - \hat{y}) \in I\!R^m$, and $P \in I\!R^{n \times r}$ is the eigenspace corresponding to $f(\hat{y})$ with r as its multiplicity.

Given a set of points y^i, $i = 1, \ldots, k$, one obtains the following overestimate for $f(y)$ in the kth iteration.

$$f^k(y) = \min_{i=1,\ldots,k} \lambda_{\min}(P_i^T(C - \mathcal{A}^T y)P_i), \tag{6}$$

where $P_i \in I\!R^{n \times r_i}$ corresponds to the eigenspace of $f(y^i)$, $i = 1, \ldots, k$. We have $f^k(y) \geq f(y)$, with equality holding when $y = y^i$, $i = 1, \ldots, k$. Also, let $\theta^k(y) = b^T y + f^k(y)$.

In the kth iteration of the cutting plane method we solve the following problem

$$\max_{y \in I\!R^m} \theta^k(y), \tag{7}$$

where we maximize the overestimate instead of $\theta(y)$ for y^k.

One can rewrite (7) as the following SDP problem

$$\begin{array}{ll} \max & b^T y + z \\ \text{s.t.} & P_j^T(C - \mathcal{A}^T y)P_j \succeq zI, \;\; j = 1, \ldots, k, \end{array} \tag{8}$$

with dual

$$
\begin{aligned}
\min\ & \sum_{j=1}^{k}(P_j^T C P_j) \bullet V_j \\
\text{s.t.}\ & \sum_{j=1}^{k}(P_j^T A_i P_j) \bullet V_j = b_i,\ i = 1,\ldots,m, \\
& \qquad \sum_{j=1}^{k} I \bullet V_j = 1, \\
& \qquad\qquad V_j \succeq 0,\ j = 1,\ldots,k.
\end{aligned} \tag{9}
$$

The dual master problem (8) is clearly a relaxation of (SDD). It requires the matrix $S = (C - \mathcal{A}^T y - zI)$ to be positive semidefinite w.r.t the subspaces spanned by the columns in P_j, $j = 1,\ldots,k$ instead of the whole of $\mathbb{R}^n$. Similarly, (9) is a constrained version of (SDP), i.e., its feasible region is contained within that of (SDP). This can be seen from the following interpretation of (9) in the Dantzig-Wolfe setting: Consider the convex set

$$\Omega = \{X \in \mathcal{S}^n : I \bullet X = 1,\ \ X \succeq 0\}$$

obtained by taking the last equality and conic constraints in (SDP). The extreme points of Ω are all rank one matrices of the form $X = dd^T$ with $d^T d = 1$ (see Overton [45]). One can write the following expressions for any $X \in \Omega$ in terms of these extreme points:

1. Consider $X = \sum_j \lambda_j d_j d_j^T$ where $\sum_j \lambda_j = 1, \lambda_j \geq 0$. Here, X is written as a convex combination of the extreme points of Ω in a *linear* fashion. Substituting this expression for X in (SDP) gives a *semi-infinite* LP formulation for this semidefinite program. If one takes only a finite number of terms in the convex combination, we get an LP which is also a *constrained* version of (SDP).

2. Consider $X = \sum_j P_j V_j P_j^T$ where $\sum_j \text{trace}(\mathrm{V_j}) = 1, V_j \in \mathcal{S}_+^{r_j}$, and $P_j = \big[d_1, \ldots, d_{r_j}\big] \in I\!R^{n \times r_j}$ is an orthonormal matrix satisfying $P_j^T P_j = I_{r_j}$. Here X is written as a convex combination of the extreme points of Ω in a *conic* fashion. Different conic formulations are possible depending on the size of r_j. Substituting this expression for X in (SDP) gives a *semi-infinite* block SDP formulation of (SDP). As long as $r_j < n$, taking a finite number of terms in the convex combination also ensures that this block SDP is a *constrained* version of (SDP).

Solving (8) and (9) for (y^k, z^k, V^k), where $V^k = \begin{pmatrix} V_1^k & & \\ & \ddots & \\ & & V_k^k \end{pmatrix}$ gives

$$\begin{aligned} \theta^k(y^k) &= b^T y^k + z^k \\ &= \sum_{j=1}^{k} (P_j^T C P_j) \bullet V_j^k \end{aligned}$$

We note that

$$\theta(y) \leq \theta^k(y) \leq \theta^k(y^k), \, \forall y \in I\!R^m. \tag{10}$$

To check whether the obtained y^k does maximize $\theta(y)$, we compute the true value $\theta(y^k)$. We assume we have an oracle (Algorithm 1) to compute $\theta(y^k)$.

Algorithm 1 (Separation Oracle/Subproblem)

Step 1: Given (y^k, z^k)*, a feasible point in (8), as input. Solve the following subproblem*

$$\begin{aligned} \min \ & (C - \mathcal{A}^T y^k) \bullet W \\ \text{s.t.} \ & I \bullet W = 1, \\ & W \succeq 0, \end{aligned} \tag{11}$$

for $W^k = P_k M^k P_k^T$ *with* $M^k \succ 0$*. The optimal value of this problem is* $\lambda_{\min}(C - \mathcal{A}^T y^k)$*.*

Step 2: If $\lambda_{\min}(C - \mathcal{A}^T y^k) = z^k$*, the oracle reports feasibility. STOP!. Else, go to Step 3.*

Step 3: If $\lambda_{\min}(C - \mathcal{A}^T y^k) < z^k$*, the oracle returns the cut*

$$\sum_{i=1}^{m} y_i (P_k^T A_i P_k) + zI \preceq P_k^T C P_k, \tag{12}$$

which is added to the dual master problem (8).

We briefly summarize the workings of this oracle.

1. The oracle computes the value of $\theta(y^k)$ in Step 1. We have $\theta(y^k) = b^T y^k + \lambda_{\min}(C - \mathcal{A}^T y^k)$, where the 2nd term is the objective value of (11).
2. If $\theta(y^k) = \theta^k(y^k)$, i.e., $\lambda_{\min}(C - \mathcal{A}^T y^k) = z^k$, then the oracle reports feasibility in Step 2. In this case, the inequality (10) ensures that y^k maximizes $\theta(y)$.
3. If $\theta(y^k) < \theta^k(y^k)$, i.e., $\lambda_{\min}(C - \mathcal{A}^T y^k) < z^k$, the oracle returns the cut (12) in Step 3. This is added to (8) and it cuts off the earlier point (y^k, z^k) which is not in the feasible region of (SDD).

4. The following special cases could occur in the kth iteration depending on the multiplicity r_k.
   (a) When $r_k = 1$, i.e., $P_k = d$, the oracle returns the linear cut
$$\sum_{i=1}^{m} y_i(d^T A_i d) + z \leq (d^T C d).$$
   (b) When $r_k \geq 2$, the oracle returns an r_k dimensional semidefinite cut.
5. Numerically one is never going to observe the higher multiplicities of $\lambda_{\min}(C - \mathcal{A}^T y^k)$. So, in our implementation of Algorithm 1, we simply compute the r smallest eigenvalues of $(C - \mathcal{A}^T y^k)$ using an iterative scheme such as *Lanczos* method (see Parlett [46]). Here $r \leq r^*$ with r^* given by Theorem 2. We then add the semidefinite cut given in (12) with $P_k \in I\!R^{n \times r_k}$, where r_k is the number of these eigenvalues that were negative and P_k is their corresponding eigenspace.

The algorithm can be stopped when the largest cut violation is *sufficiently small*, i.e., when
$$\begin{aligned}\frac{\theta^k(y^k) - \theta(y^k)}{\mid \theta^k(y^k) \mid} &= \frac{z^k - \lambda_{\min}(C - \mathcal{A}^T y^k)}{|\theta^k(y^k)|} \\ &\leq \epsilon\end{aligned}$$
for some $\epsilon > 0$. Since we have strong duality between (9) and (8), the above stopping criterion can also be written as
$$\begin{aligned}\mathrm{SC}(V^k, y^k) &= \frac{\displaystyle\sum_{j=1}^{k} (P_j^T C P_j) \bullet V_j^k - \theta(y^k)}{\left| \displaystyle\sum_{j=1}^{k} (P_j^T C P_j) \bullet V_j^k \right|} \\ &\leq \epsilon\end{aligned} \tag{13}$$

However, the sequence $\theta(y^k)$ is not monotone and it is more efficient to use the best dual iterate

$$\hat{y} = \text{argmax}\{\theta(y^i) : i = 1, \ldots, k\}, \tag{14}$$

and stop whenever

$$\begin{aligned} \text{SC}(V^k, \hat{y}) &= \frac{\displaystyle\sum_{j=1}^{k} (P_j^T C P_j) \bullet V_j^k - \theta(\hat{y})}{\left| \displaystyle\sum_{j=1}^{k} (P_j^T C P_j) \bullet V_j^k \right|} \\ &\leq \qquad \epsilon. \end{aligned} \tag{15}$$

We are now ready to introduce the *master problem* in the Dantzig-Wolfe setting. Utilizing the separation oracle described in Algorithm 1, one can rewrite (9) and (8) as the following mixed conic problems over linear and lower dimensional semidefinite cones.

$$\begin{aligned} \min \quad & \sum_{i=1}^{n_l} c_{li} x_{li} + \sum_{k=1}^{n_s} c_{sk}^T \text{svec}(X_{sk}) \\ \text{s.t.} \quad & \sum_{i=1}^{n_l} A_{li} x_{li} + \sum_{k=1}^{n_s} A_{sk} \text{svec}(X_{sk}) = b, \\ & \sum_{i=1}^{n_l} x_{li} + \sum_{k=1}^{n_s} \text{trace}(\text{X}_{\text{sk}}) = 1, \\ & x_{li} \geq 0, \quad i = 1, \ldots, n_l, \\ & X_{sk} \succeq 0, \quad k = 1, \ldots, n_s, \end{aligned}$$

and its dual

$$
\begin{array}{lrl}
\max & & b^T y + z \\
\text{s.t.} & A_{li}^T y + z + s_{li} = c_{li}, & i = 1, \ldots, n_l \\
& A_{sk}^T y + z\text{svec}(I_{r_k}) + \text{svec}(S_{sk}) = c_{sk}, & k = 1, \ldots, n_s \\
& s_{li} \geq 0, & i = 1, \ldots, n_l, \\
& S_{sk} \succeq 0, & k = 1, \ldots, n_s,
\end{array}
$$

respectively. We list the notation below:

1. n_l, n_s denote the number of linear and semidefinite cones in the master problem.
2. For $i = 1, \ldots, n_l$, we define $c_{li} \in I\!R$ and $A_{li} \in I\!R^{m\times 1}$ as follows:

$$
\begin{aligned}
c_{li} &= d_i^T C d_i, \\
A_{li} &= \begin{pmatrix} d_i^T A_1 d_i \\ \vdots \\ d_i^T A_m d_i \end{pmatrix}.
\end{aligned}
$$

3. For $k = 1, \ldots, n_s$, we define $c_{sk} \in I\!R^{\bar{r}_k}$ and $A_{sk} \in I\!R^{m\times \bar{r}_k}$ as follows:

$$
\begin{aligned}
c_{sk} &= \text{svec}(P_k^T C P_k), \\
A_{sk} &= \begin{pmatrix} \text{svec}(P_k^T A_1 P_k)^T \\ \vdots \\ \text{svec}(P_k^T A_m P_k)^T \end{pmatrix}.
\end{aligned}
$$

We will henceforth adopt the following shorthand notation for the primal and dual master problems.

$$\begin{aligned} \min\ & \bar{c}_l^T x_l + \bar{c}_s^T x_s \\ \text{s.t.}\ & \bar{A}_l x_l + \bar{A}_s x_s = \bar{b}, \\ & x_l \geq 0, \\ & x_s \succeq 0, \end{aligned} \tag{16}$$

$$\begin{aligned} \max\ & \bar{b}^T \bar{y} \\ \text{s.t.}\ \ & s_l = \bar{c}_l - \bar{A}_l^T \bar{y} \geq 0, \\ & s_s = \bar{c}_s - \bar{A}_s^T \bar{y} \succeq 0, \end{aligned} \tag{17}$$

where $\bar{y} = \begin{pmatrix} y \\ z \end{pmatrix}$ and

$$x_l = \begin{pmatrix} x_{l1} \\ \vdots \\ x_{ln_l} \end{pmatrix}, \ \ x_s = \begin{pmatrix} \text{svec}(X_{s1}) \\ \vdots \\ \text{svec}(X_{sn_s}) \end{pmatrix},$$

respectively. s_l and s_s are defined similarly. Let

$$m_{\text{mast}} = m + 1, \ \ n_{\text{mast}} = n_l + \sum_{i=1}^{n_s} \bar{r}_i,$$

denote the number of constraints and the dimension of the resultant cone in the master problem. Finally,

$$x = \begin{pmatrix} x_l \\ x_s \end{pmatrix}, \ \ s = \begin{pmatrix} s_l \\ s_s \end{pmatrix} \in I\!R^{n_{\text{mast}}}.$$

A short note is in order: Although, x_s is a vector, the notation $x_s \succeq 0$ will denote that the matrix associated with this vector is positive semidefinite. A similar interpretation holds for operations such as trace($\mathrm{x_s}$) and det(x_s).

Theorem 3. *Let (x_l^*, x_s^*) and (y^*, z^*, s_l^*, s_s^*) be optimal solutions to (16) and (17) respectively and let $\hat{y}$ be given by (14).*

1. *The solution (x_l^*, x_s^*) generates an X^* feasible in (SDP). Also, $X^* \bullet S^* = 0$, where $S^* = (C - \mathcal{A}^T y^* - z^* I)$.*
2. *If $SC(x^*, \hat{y}, \hat{z}) = 0$, i.e., $c^T x^* = \theta(\hat{y})$ with $\hat{z} = \lambda_{\min}(C - \mathcal{A}^T \hat{y})$, then X^* and $\hat{S} = (C - \mathcal{A}^T \hat{y} - \hat{z} I)$ are optimal solutions to (SDP) and (SDD) respectively.*

Proof. The primal master problem (16) is a constrained version of (SDP), and any feasible solution (x_l^*, x_s^*) in (16) gives an X^* that is feasible in (SDP), i.e., $\mathcal{A}(X^*) = b$, $I \bullet X^* = 1$, and $X^* \succeq 0$. Consider $S^* = (C - \mathcal{A}^T y^* - z^* I)$. Now, $X^* \bullet S^* = {x_l^*}^T s_l^* + {x_s^*}^T s_s^* = 0$; the 2nd equality follows from the complementary slackness conditions at optimality for the master problem.

Now, consider $c^T x^* = b^T \hat{y} + \lambda_{\min}(C - \mathcal{A}^T \hat{y})$. Adding $\hat{z}$ to both sides of this equality and rearranging terms gives $c^T x^* - (b^T \hat{y} + \hat{z}) = -\hat{z} + \lambda_{\min}(C - \mathcal{A}^T \hat{y})$. Since, x^* and $\hat{y}, \hat{z}$ are feasible solutions in (16) and (17) respectively, the quantity on the left hand side of the equality is non-negative (weak duality). Thus $\hat{z} \leq \lambda_{\min}(C - \mathcal{A}^T \hat{y})$, and so $\lambda_{\min}(\hat{S}) \geq 0$, where $\hat{S} = (C - \mathcal{A}^T \hat{y} - \hat{z} I)$. Therefore $\hat{S} \succeq 0$. Now $X^* \bullet \hat{S} = C \bullet X^* - \hat{y}^T \mathcal{A}(X^*) - \hat{z}(I \bullet X^*) = c^T x^* - b^T \hat{y} - \hat{z} = 0$ for $\hat{z} = \lambda_{\min}(C - \mathcal{A}^T \hat{y})$. Finally, $X^*, \hat{S} \succeq 0$ and $X^* \bullet \hat{S} = 0$ together imply $X^* \hat{S} = 0$.

Here is a summary of the overall decomposition approach:

1. **Initialize:** Start with an initial master problem, such that the dual master problem is a relaxation of (SDD), and its objective value is bounded.
2. **Generate dual prices:** Solve the current (16) and (17) for the new primal and dual iterates. The optimal objective value of the master problem provides an upper bound while the objective value at the best feasible point ($\theta(\hat{y}) = b^T\hat{y} + f(\hat{y})$) gives a lower bound on the SDP objective value.
3. **Check for termination:** If the criterion (13) is satisfied, STOP. Else, proceed to Step 4.
4. **Call the separation oracle:** The current dual iterate is input to the separation oracle (Algorithm 1), which returns either a linear or a semidefinite cut.
5. Update the master problems and return to Step 2.

The overall decomposition approach in the primal setting is the Dantzig-Wolfe approach [10] for large scale linear programming. In this case, the master problem is a linear program, while the separation oracle returns columns which are appended to the new restricted master problem; this is commonly known as *column generation*. In our approach, the separation oracle may also return matrices, corresponding to semidefinite cuts, which are added to the new restricted master problem; this corresponds to *matrix generation* in the primal setting. Viewed in the dual setting, the approach is the cutting plane method of Kelley [23].

We conclude this section with a discussion of various issues needed for an effective implementation of the conic decomposition approach. These are:

1. The approach requires an initial master problem to start. The initialization procedure is problem specific and we discuss it in detail in Section 8.
2. In the original Dantzig-Wolfe/Kelley approach, the prices were the optimal solutions to the dual master problem. However, it is well known that this approach has a very poor rate of convergence. There are examples in Nemirovskii and Yudin [40] (see also Hiriart-Urruty and Lemaréchal [21]) which show that standard column generation can be desperately slow. We will look at a *stabilized* column generation scheme in Section 4. The stopping criterion (13) for the stabilized scheme also needs to be modified accordingly.
3. It must be emphasized that we use a primal-dual IPM to solve the conic master problem. This is unlike the Dantzig-Wolfe decomposition approach for LP which uses the simplex method to solve the master problem, and where reoptimization with the dual simplex method after column generation is possible. An issue in the conic interior point decomposition approach is to restart the new master problem with strictly feasible primal and dual iterates after column/matrix generation. This will be the topic of Section 5.
4. Finally, it is imperative to detect the less important linear and semidefinite cones to speed up the solution of the master problem. We discuss this procedure in Section 6.

4. Stabilized column generation

In this section we will describe a stabilized column generation procedure to compute dual prices in Step 2 of the algorithm. The discussion in this section aims at improving the convergence of the conic decomposition approach.

In the kth iteration of the original column generation approach, one assumes the current overestimate $\theta^k(y)$ is a good approximation to $\theta(y)$, so maximizing $\theta^k(y)$ should also maximize $\theta(y)$. Finding the exact maximizer y^k of $\theta^k(y)$ amounts to solving (17) to optimality. However, we may have $\theta(y^k) \ll \theta^k(y^k)$ (see the example in Section 4.3.6 of Nemirovskii and Yudin [40]). Therefore, it is advisable to adopt more conservative strategies, commonly known as *stabilized column generation* (see du Merle et al. [12]). Here, one only maximizes $\theta^k(y)$ approximately in every iteration providing a more *central* point in the feasible region of the dual master problem.

Several stabilized column generation approaches have been proposed in the literature. We review some of the common schemes below:

1. **Stabilization via penalty:** The aim is to maximize $\hat{\theta}^k(y) = \theta^k(y) - u||y - \hat{y}||$ instead, where $u > 0$ is some penalty term, and $\hat{y}$ is the point corresponding to the best $\theta(y)$ value. Different norms can be used and these choices are discussed in Briant et al. [8]. The 2 norm, in particular, gives the bundle methods of Kiwiel, Lemaréchal et al. ([21]). Such a stabilizing procedure was also employed in the spectral bundle scheme [19]. The term $u||y - \hat{y}||$ serves as a regularization term preventing one from moving too far from the best

iterate $\hat{y}$. Typically, u is reduced if the current iterate achieves a *sufficient* reduction in the objective value.

2. **Stabilization via more central iterates:** This technique is adopted in Analytic center cutting plane methods (ACCPM) (see Goffin et al. [15]). Consider the set of localization

$$\text{SOL} = \{(y, z) : s_l(y, z) \geq 0, \;\; s_s(y, z) \succeq 0, \;\; \text{and} \;\; \alpha z \geq \alpha\theta(\hat{y})\},$$

where s_l, s_s are the dual slacks for the linear and semidefinite cones in (17) and $\alpha > 1$ is a suitable scaling parameter. This set contains the constraints in (8), and a suitably scaled lower bound constraint on the objective value of this problem. In ACCPM, one finds the analytic center of SOL, which is the unique point maximizing the logarithmic barrier function defined on this set. Typically, α is increased as the algorithm proceeds to keep the current iterate away from the $z \geq \theta(\hat{y})$ constraint.

We adopt the following stabilized column generation approach which is similar in spirit to the two approaches mentioned above: The idea is to approximately maximize $\theta^k(y)$ in the beginning when $\theta^k(y)$ is a poor approximation to $\theta(y)$; this amounts to solving (17) to a low tolerance. We tighten this tolerance, i.e., solve (17) (maximize $\theta^k(y)$) more accurately, as the piecewise overestimate $\theta^k(y)$ better approximates $\theta(y)$. This approach is described below:

Algorithm 2 (Stabilized Column Generation)

Step 0: Solve the initial master problem to a tight tolerance $TOL = 1e - 3$ *to obtain* (x^1, y^1, z^1, s^1)*, where* $x^1 = (x^1_l, x^1_s)$ *and* $s^1 = (s^1_l, s^1_s)$*. Set* $k = 1$*.*

Step 1: In the kth iteration compute the following parameters

$$\begin{aligned} GAPTOL(x^k, y^k, z^k, s^k) &= \frac{{x^k}^T s^k}{\max\{1, \frac{1}{2}\,|c^T x^k + b^T y^k + z^k|\}} \\ INF(x^k, y^k, z^k) &= \frac{\max\{z^k - \lambda_{\min}(C - \mathcal{A}^T y^k), 0\}}{\max\{1, \frac{1}{2}\,|c^T x^k + b^T y^k + z^k|\}} \\ OPT(x^k, y^k, z^k, s^k) &= \max\{GAPTOL(x^k, y^k, z^k, s^k), INF(x^k, y^k, z^k)\} \end{aligned} \tag{18}$$

Step 2: If $k = 1$ set $TOL = \mu \times OPT(x^1, y^1, z^1, s^1)$. Else, go to Step 3.

Step 3: If $OPT(x^k, y^k, z^k, s^k) < TOL$, set $TOL = \mu \times OPT(x^k, y^k, z^k, s^k)$, where $0 < \mu < 1$. Else, TOL remains unchanged.

Step 4: Solve the current master problem using a primal-dual IPM to tolerance TOL for $(x^{k+1}, y^{k+1}, z^{k+1}, s^{k+1})$. Set $k = k + 1$ and return to Step 1.

Given (x^k, y^k, z^k, s^k), GAPTOL(x^k, y^k, z^k, s^k) is the duality gap between the primal and dual master problems and is scaled by the average of their absolute values. INF(x^k, y^k, z^k) is the current infeasibility measure, and is the difference between the objective value of the dual master problem and the value of the eigenvalue function $\theta(y)$ scaled by the same factor as GAPTOL. If $z^k < \lambda_{\min}(C - \mathcal{A}^T y^k)$, then we can always set $z^k = \lambda_{\min}(C - \mathcal{A}^T y^k)$ and still get a feasible iterate (y^k, z^k) in the dual master problem. So, we can assume $z^k \geq \lambda_{\min}(C - \mathcal{A}^T y^k)$ without loss of generality. The tolerance update is clear: In each iteration we want to lower the complementary slackness (GAPTOL) while reducing the dual infeasibility (INF) by a corresponding amount too. We obviously have control on GAPTOL since this quantity depends on the tolerance to which we solve the master problem. Once we have attained a certain GAPTOL we retain this tolerance until the INF decreases by a corresponding amount (as we add cuts in the dual master problem). When these twin objectives are achieved, the tolerance is again lowered and the process is repeated.

Our new stopping criterion in the kth iteration is

$$\begin{aligned} \mathrm{SC}(x^k, y^k, z^k) &= \frac{c^T x^k - \theta(y^k)}{\max\{1, \frac{1}{2}\,|c^T x^k + b^T y^k + z^k|\}} \\ &\leq \epsilon \end{aligned} \tag{19}$$

for some $\epsilon > 0$. Note that we use the difference between the objective value of the *primal* master problem and the eigenvalue function, since it is only the objective value of the primal master problem that provides the upper bound. Also, this value is scaled by the average of the primal and the dual master problem objective values.

The upper bounds need not vary monotonically, and we can compute the best primal iterate

$$\hat{x} = \operatorname{argmin}\{c^T x^i : i = 1, \ldots, k\}. \tag{20}$$

and use the modified stopping criterion

$$\begin{aligned} \mathrm{SC}(\hat{x}, \hat{y}, \hat{z}) &= \frac{c^T \hat{x} - \theta(\hat{y})}{\max\{1, \frac{1}{2}\,|c^T \hat{x} + b^T \hat{y} + \hat{z}|\}} \\ &\leq \epsilon \end{aligned} \tag{21}$$

with $\hat{x}, \hat{y}$ given in (20) and (14) and $\hat{z} = \lambda_{\min}(C - \mathcal{A}^T \hat{y})$.

Theorem 4. *We have*

$$SC(x^*, y^*, z^*) < \epsilon$$

for some $\epsilon > 0$ *whenever*

$$OPT(x^*, y^*, z^*, s^*) < \frac{\epsilon}{2}$$

Proof. Consider

$$\begin{aligned}
\mathrm{SC}(x^*, y^*, z^*) &= \frac{c^T x^* - b^T y^* - \lambda_{\min}(C - \mathcal{A}^T y^*)}{\max\{1, \frac{1}{2}\,|c^T x^* + b^T y^* + z^*|\}} \\
&= \frac{c^T x^* - b^T y^* - z^* + z^* - \lambda_{\min}(C - \mathcal{A}^T y^*)}{\max\{1, \frac{1}{2}\,|c^T x^* + b^T y^* + z^*|\}} \\
&= \frac{{x^*}^T s^*}{\max\{1, \frac{1}{2}\,|c^T x^* + b^T y^* + z^*|\}} + \frac{z^* - \lambda_{\min}(C - \mathcal{A}^T y^*)}{\max\{1, \frac{1}{2}\,|c^T x^* + b^T y^* + z^*|\}} \\
&\leq \frac{{x^*}^T s^*}{\max\{1, \frac{1}{2}\,|c^T x^* + b^T y^* + z^*|\}} + \frac{\max\{z^* - \lambda_{\min}(C - \mathcal{A}^T y^*), 0\}}{\max\{1, \frac{1}{2}\,|c^T x^* + b^T y^* + z^*|\}} \\
&= \mathrm{GAPTOL}(x^*, y^*, z^*, s^*) + \mathrm{INF}(x^*, y^*, z^*) \\
&\leq \frac{\epsilon}{2} + \frac{\epsilon}{2} \\
&= \epsilon
\end{aligned}$$

Theorem 4 suggests that we can use OPT instead as an optimality measure and stop whenever OPT $< \epsilon$. This is employed as a stopping criterion in Algorithm 4.

Theorem 5. *Let $\hat{x}$ and $\hat{y}$ be given as in (20) and (14) respectively.*

1. *The $\hat{x}$ generates an $\hat{X}$ feasible in (SDP).*
2. *If $SC(\hat{x}, \hat{y}, \hat{z}) = 0$, i.e., $c^T\hat{x} = \theta(\hat{y})$, then $\hat{X}$ and $\hat{S} = (C - \mathcal{A}^T\hat{y} - \hat{z}I)$ with $\hat{z} = \lambda_{\min}(C - \mathcal{A}^T\hat{y})$ are optimal solutions for (SDP) and (SDD) respectively.*

Proof. The proof relies on the feasibility of $\hat{x}$ and $\hat{y}, \hat{z}$ in (16) and (17), and is along the same lines as Theorem 3.

Our stopping criterion (21), implies that $\hat{x}, \hat{y}$ and $\hat{z} = \lambda_{\min}(C - \mathcal{A}^T\hat{y})$ are ϵ optimal solutions to (SDP) and (SDD) respectively.

4.1. Solving the mixed conic problem

Our master problem is a mixed conic problem over linear and semidefinite cones. We solve the master problem using a primal-dual interior point method in a *feasible* setting. We used SDPT3 by Tütüncü et al. [53] since it gave us the flexibility to provide an initial starting point.

More details on the primal-dual IPM can be found in Tütüncü et al. [53]. We will summarize some of our settings with this code:

1. A number of primal-dual IPM directions (see Zhang [56]) are available for conic programs. SDPT3 provides two such directions: the H..K..M direction (see Helmberg et al. [20], Kojima et al. [26], and Monteiro [38]) and the NT direction (see Nesterov & Todd [42]). The H..K..M. direction requires less computational effort on conic problems involving only linear and semidefinite cones (see the discussion and the computational results in Tütüncü et al. [53]). We employed the H..K..M. direction in our computational results in Section 8.
2. We found it was better to use SDPT3 as a long step path following algorithm rather than the usual Mehrotra predictor-corrector setting.
3. We used SDPT3 in a feasible IPM setting. We use the warm-start procedure described in Section 5 to restart the primal and the dual problems with strictly feasible starting points in each iteration.

4.2. Upper and lower bounds

We describe a procedure of generating upper and lower bounds on the SDP objective value in our stabilized column generation approach.

Since, we do not solve the master problem to optimality in every iteration, an upper bound is only given by the objective value of the primal master problem (16), which equals the objective value of the dual master problem (17) plus the duality gap. Moreover, these upper bounds need not be monotonically decreasing, so one will need to maintain the best (lowest) upper bound.

A lower bound is obtained as follows: Let (y^*, z^*) be the iterate in the dual master feasible region (17). Consider the current dual slack matrix $S^* = (C - \mathcal{A}^T y^* - z^* I)$ at this point. Compute $\lambda^* = \lambda_{\min}(S^*)$. It follows from the definition of the minimum eigenvalue that $S^{lb} = (C - \mathcal{A}^T y^* - z^* I - \lambda^* I) \succeq 0$. Let $S^{lb} = (C - \mathcal{A}^T y^{lb} - z^{lb} I)$ where $y^{lb} = y^*$ and $z^{lb} = z^* + \lambda^*$. Since (y^{lb}, z^{lb}) is within the feasible region of (SDD), the objective value $b^T y^{lb} + z^{lb}$ gives a lower bound on the optimal objective value to (SDP). A similar procedure was employed in the spectral bundle method (see Helmberg [17]). Once again, these lower bounds need not be monotonically increasing, so one needs to maintain the best (greatest) lower bound.

5. Warm start

We consider the issue of restarting the primal and dual master problems with a strictly feasible iterate after one adds either a linear or semidefinite cutting plane to the dual master problem. The cutting plane corresponds to a column (linear case) or matrix (semidefinite case) which is added to the new primal master problem.

Consider the restricted master problem

$$\begin{aligned} \min\ & \bar{c}_l^T x_l + \bar{c}_s^T x_s \\ \text{s.t.}\ & \bar{A}_l x_l + \bar{A}_s x_s = \bar{b}, \\ & x_l \geq 0, \\ & x_s \succeq 0, \end{aligned} \tag{22}$$

and its dual

$$\begin{aligned} \max\ & \bar{b}^T \bar{y} \\ \text{s.t.}\quad & s_l = \bar{c}_l - \bar{A}_l^T \bar{y} \geq 0, \\ & s_s = \bar{c}_s - \bar{A}_s^T \bar{y} \succeq 0, \end{aligned} \tag{23}$$

in the usual shorthand notation, where $\bar{y} = \begin{pmatrix} y \\ z \end{pmatrix}$. Let $(x_l^*, x_s^*, \bar{y}^*, s_l^*, s_s^*)$ be the current feasible solution in this master problem where $\bar{y}^* = \begin{bmatrix} y^* \\ z^* \end{bmatrix}$. Also, $S^* = (C - \mathcal{A}^T y^* - z^* I)$ is the dual slack matrix at $\bar{y}^*$.

5.1. Warm starting after adding a linear cutting plane

We consider the case when one adds a linear cutting plane $a_l^T \bar{y} \leq d$ to (23). The cut corresponds to the eigenspace of $\lambda_{\min}(S^*)$, which currently has multiplicity 1.

The new restricted master problem is

$$\begin{aligned} \min\ & \bar{c}_l^T x_l + \bar{c}_s^T x_s + d^T \beta \\ \text{s.t.}\ & \bar{A}_l x_l + \bar{A}_s x_s + a_l \beta = \bar{b}, \\ & x_l \geq 0, \\ & x_s \succeq 0, \\ & \beta \geq 0, \end{aligned} \tag{24}$$

with dual

$$\begin{aligned} \max\ & \bar{b}^T \bar{y} \\ \text{s.t.}\ & s_l = \bar{c}_l - \bar{A}_l^T \bar{y} \geq 0, \\ & s_s = \bar{c}_s - \bar{A}_s^T \bar{y} \succeq 0, \\ & \gamma = d - a_l^T \bar{y} \geq 0. \end{aligned} \tag{25}$$

We want a strictly feasible starting point $(x_l^{st}, x_s^{st}, \beta^{st}, \bar{y}^{st}, s_l^{st}, s_s^{st}, \gamma^{st})$ for the new master problem. The new constraint cuts off the earlier iterate $\bar{y}^*$, i.e., $\gamma^* = d - a_l^T \bar{y}^* < 0$.

One can perturb $\bar{y}^*$ using the technique discussed in Section 4.2 to generate a $\bar{y}^{lb}$ within the feasible region of (SDD), and hence, within the feasible region of (25). Let $s_l^{lb} = \bar{c}_l - \bar{A}_l^T \bar{y}^{lb}$ and $s_s^{lb} = \bar{c}_s - \bar{A}_s^T \bar{y}^{lb}$. We note that $\gamma^{lb} = d - a_l^T \bar{y}^{lb} = 0$.

The perturbed point $(x_l^*, x_s^*, \beta^*, \bar{y}^{lb}, s_l^{lb}, s_s^{lb}, \gamma^{lb})$ is feasible in the new master problem with $\beta^* = 0$. We want to increase β^* and γ^{lb} from their current zero values, while limiting the variation in the other variables x_l, x_s, s_l, s_s at the same time. To do this, one solves the following problems (see Goffin and Vial [16])

$$\begin{array}{lrl} \max & \log \beta & \\ \text{s.t.} & \bar{A}_l \Delta x_l + \bar{A}_s \Delta x_s + a_l \beta & = 0, \\ & \sqrt{||D_l^{-1} \Delta x_l||^2 + ||D_s^{-1} \Delta x_s||^2} & \leq 1, \\ & \beta & \geq 0, \end{array} \tag{26}$$

and

$$\begin{array}{lrl} \max & \log \gamma & \\ \text{s.t.} & a_l^T \Delta \bar{y} + \gamma & = 0, \\ & \sqrt{||D_l \bar{A}_l^T \Delta \bar{y}||^2 + ||D_s \bar{A}_s^T \Delta \bar{y}||^2} & \leq 1, \\ & \gamma & \geq 0, \end{array} \tag{27}$$

for $(\Delta x_l, \Delta x_s, \beta)$ and $(\Delta \bar{y}, \gamma)$ respectively. Here, D_l and D_s are appropriate primal-dual scaling matrices for the linear and semidefinite cones at $(x_l^*, x_s^*, s_l^{lb}, s_s^{lb})$. Finally, let

$$\Delta s_l = -\bar{A}_l^T \Delta \bar{y},$$
$$\Delta s_s = -\bar{A}_s^T \Delta \bar{y}.$$

These directions are also commonly known as the Mitchell-Todd restart directions in the primal-dual setting. The solution to problems (26) and (27) is given by

$$\begin{aligned} \Delta x_l &= -D_l^2 \bar{A}_l^T (\bar{A}_l D_l^2 \bar{A}_l^T + \bar{A}_s D_s^2 \bar{A}_s^T)^{-1} a_l \beta \\ \Delta x_s &= -D_s^2 \bar{A}_s^T (\bar{A}_l D_l^2 \bar{A}_l^T + \bar{A}_s D_s^2 \bar{A}_s^T)^{-1} a_l \beta \\ \Delta \bar{y} &= -(\bar{A}_l D_l^2 \bar{A}_l^T + \bar{A}_s D_s^2 \bar{A}_s^T)^{-1} a_l \beta \\ \gamma &= V \beta \end{aligned} \tag{28}$$

where β is the unique solution to

$$\begin{array}{ll} \min & \frac{1}{2}V\beta^2 - \log\beta \\ \text{s.t.} & \beta \geq 0 \end{array} \tag{29}$$

and $V = a_l^T(\bar{A}_l D_l^2 \bar{A}_l^T + \bar{A}_s D_s^2 \bar{A}_s^T)^{-1} a_l$. The solution to (29) can be obtained in closed form as $\beta = \frac{1}{\sqrt{V}}$. The strictly feasible starting point is then given by

$$\begin{aligned} (x_l^{st}, x_s^{st}, \beta^{st}) &= (x_l^* + \kappa\alpha_{max}^p \Delta x_l, x_s^* + \kappa\alpha_{max}^p \Delta x_s, \kappa\alpha_{max}^p \beta) \\ \bar{y}^{st} &= \bar{y}^{lb} + \kappa\alpha_{max}^d \Delta\bar{y} \\ (s_l^{st}, s_s^{st}, \gamma^{st}) &= (s_l^{lb} + \kappa\alpha_{max}^d \Delta s_l, s_s^{lb} + \kappa\alpha_{max}^d \Delta s_s, \kappa\alpha_{max}^d \gamma) \end{aligned} \tag{30}$$

where $\kappa \in (0,1)$. Moreover, α_{max}^p and α_{max}^d are the maximal steps that can be taken along the primal and dual directions respectively; these quantities are computed using the usual minimum ratio test for a linear cone.

5.2. Warm starting after adding a semidefinite cut

We now consider the case when one adds one semidefinite cut $B_s^T \bar{y} \preceq d$ where $B_s \in I\!R^{\bar{p} \times m+1}$ and $d \in I\!R^{\bar{p}}$ in the dual master problem. The cut corresponds to the eigenspace of $\lambda_{\min}(S^*)$, which currently has multiplicity $p \geq 3$. We note that $\bar{p} = \frac{p(p+1)}{2}$.

The new master problem is

$$\begin{array}{ll} \min & \bar{c}_l^T x_l + \bar{c}_s^T x_s + d^T \beta \\ \text{s.t.} & \bar{A}_l x_l + \bar{A}_s X_s + B_s \beta = \bar{b}, \\ & x_l \geq 0, \\ & x_s \succeq 0, \\ & \beta \succeq 0, \end{array} \tag{31}$$

with dual

$$\begin{aligned} &\max \bar{b}^T \bar{y} \\ &\text{s.t.} \quad s_l = \bar{c}_l - \bar{A}_l^T \bar{y} \geq 0, \\ &\qquad\;\; s_s = \bar{c}_s - \bar{A}_s^T \bar{y} \succeq 0, \\ &\qquad\;\; \gamma = d - B_s^T \bar{y} \succeq 0, \end{aligned} \tag{32}$$

where $B_s \in I\!R^{\bar{p} \times m+1}$ and $\beta, \gamma, d \in I\!R^{\bar{p}}$. Our aim is to generate a strictly feasible starting point $(x_l^{st}, x_s^{st}, \beta^{st}, \bar{y}^{st}, s_l^{st}, s_s^{st}, \gamma^{st})$ for the new master problem. The new constraint cuts off the earlier iterate $\bar{y}^*$, i.e., $\gamma^* = d - B_s^T \bar{y}^* \not\succeq 0$.

As in the linear case, we perturb $\bar{y}^*$ using the lower bounding technique in Section 4.2 to generate $\bar{y}^{lb}$ which is in the feasible region of (SDD) and hence within the feasible region of (32). Let $s_l^{lb} = \bar{c}_l - \bar{A}_l^T \bar{y}^{lb}$ and $s_s^{lb} = \bar{c}_s - \bar{A}_s^T \bar{y}^{lb}$. Also, $\gamma^{lb} = d - B_s^T \bar{y}^{lb} \succeq 0$. The perturbed point $x_l^*, x_s^*, \beta^*, \bar{y}^{lb}, s_l^{lb}, s_s^{lb}, \gamma^{lb})$ is feasible in the new master problem, with $\beta^* = 0$. We want to increase β^* and γ^{lb} until they strictly lie within the new semidefinite cone, while limiting the variation in the other variables x_l, x_s, s_l, s_s at the same time.

To do this, one solves the following problems

$$\begin{aligned} &\max && \log\det\beta \\ &\text{s.t.} && \bar{A}_l \Delta x_l + \bar{A}_s \Delta x_s + B_s \beta = 0, \\ &&& \sqrt{\|D_l^{-1}\Delta x_l\|^2 + \|D_s^{-1}\Delta x_s\|^2} \leq 1, \\ &&& \beta \succeq 0, \end{aligned} \tag{33}$$

and

$$\begin{aligned} \max \quad & \log\det\gamma \\ \text{s.t.} \quad & B_s^T \Delta\bar{y} + \gamma = 0, \\ & \sqrt{||D_l \bar{A}_l^T \Delta\bar{y}||^2 + ||D_s \bar{A}_s^T \Delta\bar{y}||^2} \leq 1, \\ & \gamma \succeq 0, \end{aligned} \tag{34}$$

for $(\Delta x_l, \Delta x_s, \beta)$ and $(\Delta\bar{y}, \gamma)$ respectively. Here D_l and D_s are appropriate primal-dual scaling matrices for the linear and semidefinite cones at $(x_l^*, x_s^*, s_l^{lb}, s_s^{lb})$. Finally

$$\begin{aligned} \Delta s_l &= -\bar{A}_l^T \Delta\bar{y}, \\ \Delta s_s &= -\bar{A}_s^T \Delta\bar{y}. \end{aligned}$$

We note that similar directions in the primal setting were proposed by Oskoorouchi and Goffin [43]. The solution to problems (33) and (34) is given by

$$\begin{aligned} \Delta x_l &= -D_l^2 \bar{A}_l^T (\bar{A}_l D_l^2 \bar{A}_l^T + \bar{A}_s D_s^2 \bar{A}_s^T)^{-1} B_s \beta, \\ \Delta x_s &= -D_s^2 \bar{A}_s^T (\bar{A}_l D_l^2 \bar{A}_l^T + \bar{A}_s D_s^2 \bar{A}_s^T)^{-1} B_s \beta, \\ \Delta y &= -(\bar{A}_l D_l^2 \bar{A}_l^T + \bar{A}_s D_s^2 \bar{A}_s T)^{-1} B_s \beta, \\ \gamma &= V\beta \end{aligned} \tag{35}$$

where β is the unique solution to

$$\begin{aligned} \min \quad & \frac{p}{2}\beta^T V \beta - \log\det\beta \\ \text{s.t.} \quad & \beta \succeq 0 \end{aligned} \tag{36}$$

and $V \in \mathcal{S}^{\bar{p}}$ is a symmetric positive definite matrix given by

$$V = B_s^T (\bar{A}_l D_l^2 \bar{A}_l^T + \bar{A}_s D_s^2 \bar{A}_s^T)^{-1} B_s.$$

The strictly feasible starting point is then given by

$$
\begin{aligned}
(x_l^{st}, x_s^{st}, \beta^{st}) &= (x_l^* + \kappa\alpha_{max}^p \Delta x_l, x_s^* + \kappa\alpha_{max}^p \Delta x_s, \kappa\alpha_{max}^p \beta) \\
\bar{y}^{st} &= \bar{y}^{lb} + \kappa\alpha_{max}^d \Delta\bar{y} \\
(s_l^{st}, s_s^{st}, \gamma^{st}) &= (s_l^{lb} + \kappa\alpha_{max}^d \Delta s_l, s_s^{lb} + \kappa\alpha_{max}^d \Delta s_s, \kappa\alpha_{max}^d \gamma)
\end{aligned}
\tag{37}
$$

where $\kappa \in (0,1)$. Moreover, α_{max}^p and α_{max}^d are the maximal steps that can be taken along the primal and dual directions respectively; there exist closed form formulas to compute these quantities for semidefinite cones (see Tütüncü et al. [53]).

Problem (36) can be reformulated as a standard conic problem over linear, second order, and semidefinite cones, and solved within SDPT3 in a primal-dual IPM framework as follows:

Consider the Cholesky factorization $V = R^T R$ of the positive definite matrix $V \in \mathcal{S}^{\bar{p}}$, with $R \in I\!R^{\bar{p}\times\bar{p}}$. The problem (36) can be reformulated as

$$
\begin{aligned}
\min \quad & \frac{p}{2}t - \log\det\beta \\
\text{s.t.} \quad & R\beta - Is = 0, \\
& u - \begin{pmatrix} 1 & 0 \\ 0 & 2I \\ 1 & 0 \end{pmatrix} \begin{pmatrix} t \\ s \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \\
& u \succeq_Q 0, \\
& \beta \succeq 0.
\end{aligned}
\tag{38}
$$

This problem is solved within the standard primal-dual IPM framework in SDPT3 where the variables $t \in I\!R$ and $s \in I\!R^{\bar{p}}$ are treated as free variables. The barrier parameter for the semidefinite cone $\beta \succeq 0$ is held constant at 1 whenever it falls below this value, while the corresponding parameter for the second order cone $u \succeq_Q 0$ is driven to zero.

6. Detecting and aggregating unimportant cones

We will consider procedures to detect unimportant constraints in Section 6.1 and a technique to aggregate them in Section 6.2. Aggregation helps in keeping the size of the master problem bounded and this will reduce the time taken to solve these problems.

6.1. Detecting unimportant cones

We employ a primal-dual scaling measure to detect unimportant cones in this section. For the ith linear cone, the scaling measure is $\frac{x_{li}}{s_{li}}$, and cones with a small measure are considered unimportant. For the jth semidefinite cone, the measure is $\frac{\mathrm{trace}(\mathrm{x_{sj}})}{\mathrm{trace}(\mathrm{s_{sj}})}$.

One can safely drop linear and semidefinite cones for which these indicators are zero without affecting the solution. On the other hand, we aggregate those cones for which the indicators are small, but, nonzero in an aggregate linear block of size 1. We describe this procedure in Section 6.2.

6.2. Dropping and aggregating unimportant cones

Consider the master problem written in the following format:

$$\begin{aligned} \min \quad & \bar{c}_{\text{agg}} x_{\text{agg}} + \bar{c}_l^T x_l + \bar{c}_s^T x_s \\ \text{s.t.} \quad & \bar{A}_{\text{agg}} x_{\text{agg}} + \bar{A}_l x_l + \bar{A}_s x_s = \bar{b} \\ & x_{\text{agg}} \geq 0, \\ & x_l \geq 0, \\ & x_s \succeq 0, \end{aligned} \tag{39}$$

with dual

$$\begin{aligned} \max \ & \bar{b}^T \bar{y} \\ \text{s.t.} \quad & s_l = \bar{c}_l - \bar{A}_l^T \bar{y} \geq 0, \\ & s_s = \bar{c}_s - \bar{A}_s^T \bar{y} \succeq 0, \\ & s_{\text{agg}} = \bar{c}_{\text{agg}} - \bar{A}_{\text{agg}}^T \bar{y} \geq 0, \end{aligned} \tag{40}$$

where $x_{\text{agg}}, s_{\text{agg}} \in I\!R$ represent an aggregate linear block of size 1 in the primal and dual problems. The idea is to aggregate the unimportant linear and semidefinite cones within this aggregate linear block. These unimportant cones are detected using the techniques discussed in Section 6.1. Our aggregation scheme is described below:

Algorithm 3 (Drop and Aggregate)

Step 0: Set $\bar{c}_{\text{agg}}, \bar{A}_{\text{agg}} = 0$.

Step 1: Solve the master problem for $(x_l^*, x_s^*, x_{\text{agg}}^*, \bar{y}^*, s_l^*, s_s^*, s_{\text{agg}}^*)$*. Use the criteria discussed in Section 6.1 to detect unimportant cones. Drop the linear and semidefinite cones for which the indicators are zero. Let LD and SD* *denote the index set of linear and semidefinite cones, respectively for which these indicators are small, but nonzero.* **Note:** *The cones in the initial master problem are never dropped nor aggregated.*

Step 2: Carry out the updates in the order mentioned:

$$\begin{aligned}
\bar{A}_{\text{agg}} &= \frac{x^*_{\text{agg}}\bar{A}_{\text{agg}} + \sum\limits_{i\in LD}\bar{A}_{li}x^*_{li} + \sum\limits_{j\in SD}\bar{A}_{sj}x^*_{sj}}{x^*_{\text{agg}} + \sum\limits_{i\in LD}x^*_{li} + \sum\limits_{j\in SD}\text{trace}(\text{x}^*_{\text{sj}})} \\
\bar{c}_{\text{agg}} &= \frac{x^*_{\text{agg}}\bar{c}_{\text{agg}} + \sum\limits_{i\in LD}\bar{c}^T_{li}x^*_{li} + \sum\limits_{j\in SD}\bar{c}^T_{sj}x^*_{sj}}{x^*_{\text{agg}} + \sum\limits_{i\in LD}x^*_{li} + \sum\limits_{j\in SD}\text{trace}(\text{x}^*_{\text{sj}})} \\
x^*_{\text{agg}} &= x^*_{\text{agg}} + \sum_{i\in LD}x^*_{li} + \sum_{j\in SD}\text{trace}(\text{x}^*_{\text{sj}}) \\
s^*_{\text{agg}} &= \bar{c}_{\text{agg}} - \bar{A}^T_{\text{agg}}\bar{y}^*
\end{aligned}$$

Update $x^*_l, s^*_l, \bar{A}_l, \bar{c}_l$ and $X^*_s, S^*_s, \bar{A}_s, \bar{c}_s$ by removing the indices in LD and SD respectively. The new iterate $(x^*_l, x^*_s, x^*_{\text{agg}}, \bar{y}^*, s^*_l, s^*_s, s^*_{\text{agg}})$ is strictly feasible in the aggregated master problem.

7. The complete decomposition algorithm

Algorithm 4 (The overall IPM conic decomposition algorithm)

1. **Initialize:** *Construct an initial master problem. Solve this master problem to a tight tolerance (1e-3) to get $(x^1, \bar{y}^1, s^1)$. Set $k = 1$, $\epsilon =$ 1e-3, and $TOL = \mu \times OPT(x^1, \bar{y}^1, s^1)$ using (18). Update $\hat{x}$ via (20), $\hat{y}$ via (14), and LB,UB using the procedure in Section 4.2.*
2. **Call separation oracle:** *Call the separation oracle (Algorithm 1) at the point $\bar{y}^k$. Update $OPT(x^k, \bar{y}^k, s^k)$ via (18).*

3. **Termination check:** *If* $OPT(x^k, \bar{y}^k, s^k) < \epsilon$*, go to Step 8. Else, continue.*

4. **Tolerance update:** *Update the tolerance TOL using Algorithm 2.*

5. **Update and warm-start master problem:** *If the oracle returned cutting planes, update the primal and dual master problems. Use the warm-start procedure described in Section 5 to obtain a strictly feasible starting point. Else, if the oracle reported feasibility in Step 2, the previous iterate* $(x^k, \bar{y}^k, z^k)$ *serves as the warm-start point.*

6. **Solve master problem:** *Solve the master problem to a tolerance TOL using a primal-dual IPM conic solver with the settings discussed in Section 4.1, and the warm-start point obtained in Step 5. Let* $(x^{k+1}, \bar{y}^{k+1}, s^{k+1})$ *be the solution obtained. Update* $\hat{x}$*,* $\hat{y}$*, LB and UB.*

7. **Aggregate unimportant constraints:** *Detect unimportant constraints using the scaling measures described in Section 6.1. Aggregate the unimportant linear and semidefinite cones within an aggregate linear block of size* 1 *and update* (x^k, s^k) *using Algorithm 3. Set* $k = k + 1$ *and return to Step 2.*

8. **Terminate with solution:** *Construct an* ϵ *optimal solution for (SDP) and (SDD) from* $(\hat{x}, \hat{y}, \hat{z})$ *with* $\hat{z} = \lambda_{\min}(C - \mathcal{A}^T \hat{y})$*.*

The overall approach is summarized in Figure 1.

8. Computational Experiences with the algorithm

Our test problems are presented in Sections 8.1, 8.2, 8.3, and 8.4. A detailed discussion of our computational experiences appears in Section 8.5.

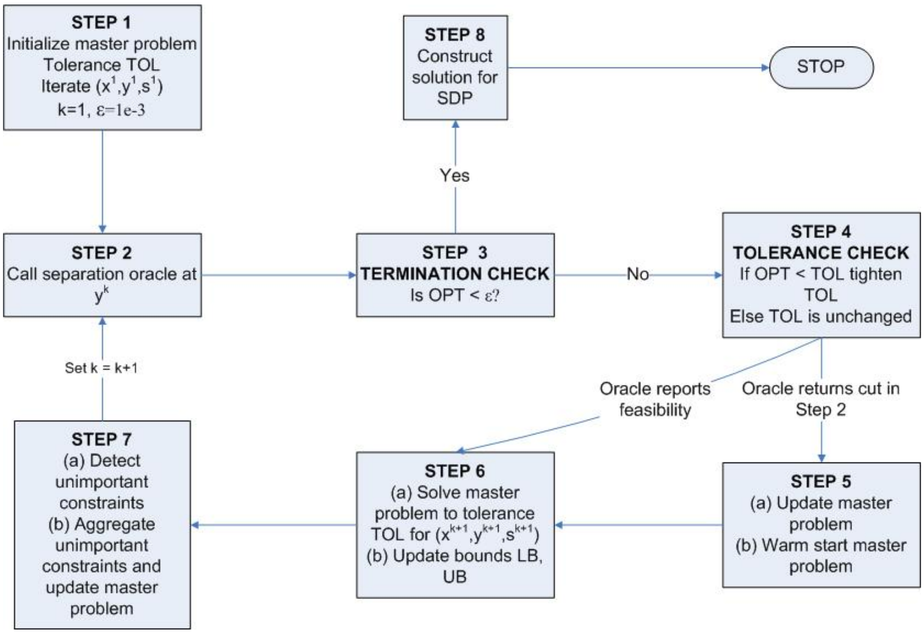

Fig. 1. The complete algorithm

8.1. The maxcut SDP

The SDP relaxation for the maxcut problem reads as:

$$
\begin{aligned}
&\min -\frac{L}{4} \bullet X \\
&\text{s.t.}\ \ \operatorname{diag}(X) = e, \\
&\qquad\qquad\quad X \succeq 0,
\end{aligned}
$$

with dual

$$
\begin{aligned}
&\max e^T y \\
&\text{s.t.} \qquad S = -\frac{L}{4} - \operatorname{Diag}(y), \\
&\qquad\qquad S \succeq 0,
\end{aligned}
$$

where L is the Laplacian matrix associated with the graph $G = (V, E)$. Also, $\operatorname{diag}(X)$ is a n dimensional vector containing the diagonal entries of X and

$\mathrm{Diag}(y)$ is its adjoint operator that forms a diagonal matrix with the components of y. Note that $I \bullet X = n$, where n is the number of nodes in the graph, although, this constraint is not explicitly present in the original SDP.

Our starting dual master problem is

$$\begin{array}{ll} \max & e^T y \\ \text{s.t.} & y \leq -\frac{1}{4}\mathrm{diag}(L). \end{array}$$

The constraints correspond to $\mathrm{diag}(S) \geq 0$.

8.2. The graph partitioning SDP

The SDP relaxation for the graph partitioning problem reads as:

$$\begin{array}{ll} \min & \frac{L}{4} \bullet X \\ \text{s.t.} & \mathrm{diag}(X) = e, \\ & ee^T \bullet X \leq 0.05n^2, \\ & X \succeq 0, \end{array}$$

with dual

$$\begin{array}{ll} \max & e^T y + 0.05n^2 y_0 \\ \text{s.t.} & S = \frac{L}{4} - \mathrm{Diag}(y) - y_0(ee^T), \\ & S \succeq 0, \\ & y_0 \leq 0, \end{array}$$

where L is the Laplacian matrix associated with the graph $G = (V, E)$. Once again $I \bullet X = n$, where n is the number of nodes in the graph.

Our starting dual master problem is

$$\begin{aligned} \max\ & e^T y + 0.05 n^2 y_0 \\ \text{s.t.}\ & y_0 e + y \leq \frac{1}{4}\mathrm{diag}(L), \\ & n^2 y_0 + e^T y \leq 0, \\ & y_0 \leq 0. \end{aligned}$$

The first set of constraints results from $\mathrm{diag}(S) \geq 0$. The constraint in the second set is obtained from $e^T S e \geq 0$, where e is the all ones vector.

8.3. The Lovasz theta SDP

The SDP relaxation for the Lovasz theta problem reads as:

$$\begin{aligned} \min\ & -ee^T \bullet X \\ \text{s.t.}\ & \frac{1}{2}(e_i e_j^T + e_j e_i^T) \bullet X = 0, \forall \{i,j\} \in E \\ & I \bullet X = 1, \\ & X \succeq 0, \end{aligned}$$

with dual

$$\begin{aligned} \max\ & y_0 \\ \text{s.t.}\ & S = -ee^T - \sum_{\{i,j\} \in E} \frac{y_{ij}}{2}(e_i e_j^T + e_j e_i^T) - y_0 I \\ & S \succeq 0. \end{aligned}$$

Note that $I \bullet X = 1$ is explicitly present in the primal problem.

We will assume that our graph $G = (V, E)$ is not complete. Our starting dual master problem is

$$\begin{array}{rl} \max & y_0 \\ \text{s.t.} & y_0 \leq -2, \\ y_0 - \frac{1}{2} y_{ij} \leq 0, & \{i,j\} \in E, \\ y_0 + \frac{1}{2} y_{ij} \leq -2, & \{i,j\} \in E. \end{array}$$

The first linear constraint is obtained by requiring any 2×2 submatrix of S not corresponding to an edge in the graph be psd. The latter two constraints are obtained by requiring every 2×2 submatrix of S corresponding to an edge in the graph to be positive semidefinite.

8.4. Randomly generated semidefinite programs

We generated random semidefinite programs of the form

$$\begin{array}{rl} \max & b^T y + z \\ \text{s.t.} & \sum_{i=1}^{m} y_i A_i + zI \preceq C, \\ & y \leq u, \\ & y \geq l, \end{array}$$

where the symmetric matrices A_i, $i = 1, \ldots, m$ are randomly chosen from a normal distribution with a prescribed sparsity. We use the MATLAB command sprandsym(n, density) to generate these matrices. The bounds l and u are also chosen from a normal distribution with $l < u$.

Our starting master problem was

$$\begin{aligned} \max\ & b^T y + z \\ \text{s.t.}\ & y \geq l, \\ & y \leq u. \end{aligned}$$

8.5. Detailed discussion of numerical results

All computations are done in the MATLAB environment on a 2×2.4GHz processor with 1 GB of memory. We use SDPT3, a primal-dual IPM solver (Tütüncu et al. [53]), to solve our master problem.

We use SDPT3 in a *feasible* IPM setting by providing it with a strictly feasible starting point in every iteration. The H..K..M primal-dual direction is used in the search direction. Our subproblem employs MATLAB's Lanczos solver *eigs*, where one computes the $r = \lfloor \frac{m}{2} \rfloor$ smallest eigenvalues of S in every iteration. This choice for r follows from Theorem 2. Let p be the number of these eigenvalues that are negative. If $p = 1$, the oracle returns a linear cut. Else, if $p \geq 2$ the oracle returns a semidefinite cut of size p.

Initially, we solve the master problem to a tolerance $1e - 3$, and set TOL to be the infeasibility measure at this point. Whenever TOL $<$ OPT, we set TOL $= \frac{1}{2}$OPT. We run each problem for m iterations, or, until OPT $< 1e - 3$, whichever comes earlier.

The warm start procedure described in Section 5 is used to restart our master problem with a strictly feasible starting point in every iteration. We use the primal-dual scaling measure (see Section 6) to aggregate constraints. In each iteration, we compute the average of all the scaling measures and aggregate those constraints whose scaling measure is less than $1e-3$ times this average value. The unimportant constraints are aggregated within a linear block of size 1.

We compute the upper and lower bounds in each iteration as described in Section 4.2. Figure 2 shows the variation of these bounds for maxG11, a semidefinite program with $n, m = 800$. We present our computational results for maxcut,

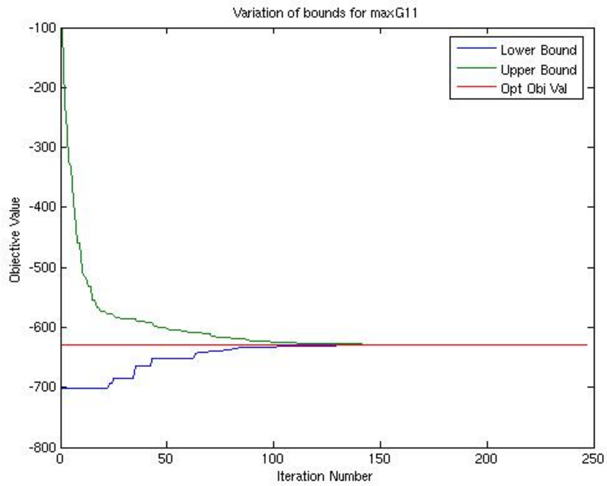

Fig. 2. Variation of bounds for the maxG11 problem

graph partitioning, and Lovasz theta problems in Table 1, 2, and 3 respectively. For the maxcut simulations, the first 13 problems are from the SDPLIB repository maintained by Borchers [6], while the last 2 problems are from the 7th DIMACS Implementation Challenge [48]. The problem bm1 is from the DIMACS set while all other graph partitioning and Lovasz theta problems are from the SDPLIB set.

The columns in Tables 1-3 represent the following:

1. **Problem:** Problem Name.
2. n, m**:** The dimensions of the SDP.
3. **Optimal Value:** The optimal objective value of the SDP.
4. **LP,SDP cones:** The number of linear and semidefinite cones in the final master problem. For the semidefinite cones, the notation $n_s(s)$ refers to a block diagonal semidefinite cone of size n_s where there are s blocks in all. The dimension of each block is bounded above by r.
5. **Upper,Lower bound:** The best lower and upper bounds obtained during the algorithm.
6. ϵ**:** Our optimality tolerance given by

$$\epsilon = \frac{c^T \hat{x} - \theta(\hat{y})}{\max\{1, \frac{1}{2}\,|(c^T \hat{x} + \theta(\hat{y}))|\}} \tag{41}$$

7. **Time (hh:mm:ss)** The total time taken by the algorithm to attain an optimality tolerance of TOL $\leq 1e-3$, or run m iterations, whichever comes earlier.

We also present our computational experiences on randomly generated semidefinite programs in Table 4. We compare our lower bounds and solution times with those obtained with running SDPT3 directly on the semidefinite programs. Our stopping criterion is TOL $\leq 1e-3$ or $\max\{m.n\}$ iterations, whichever, comes earlier. To achieve a fair comparison, the stopping criterion for SDPT3 is

Table 1. Computational results on maxcut problems

Problem	n	Opt value	LP cones	SDP cones	Lower bound	ϵ	Time (h:m:s)
mcp100	100	-226.16	101	152(31)	-226.32	1.6e-3	16
mcp124-1	124	-141.99	125	187(48)	-141.87	1.5e-3	34
mcp124-2	124	-269.88	125	173(36)	-270.05	1.4e-3	25
mcp124-3	124	-467.75	125	174(35)	-468.07	1.5e-3	22
mcp124-4	124	-864.41	125	189(41)	-864.99	1.5e-3	21
mcp250-1	250	-317.26	251	339(84)	-317.46	1.3e-3	2:03
mcp250-2	250	-531.93	251	343(51)	-532.35	1.7e-3	1:05
mcp250-3	250	-981.17	251	307(44)	-981.86	1.6e-3	49
mcp250-4	250	-1681.96	251	301(43)	-1683.1	1.6e-3	47
mcp500-1	500	-598.15	501	735(115)	-598.56	1.6e-3	12:29
mcp500-2	500	-1070.06	501	634(70)	-1070.70	1.5e-3	6:18
mcp500-3	500	-1847.97	501	570(57)	-1849.20	1.6e-3	4:33
mcp500-4	500	-3566.73	501	523(49)	-3569.10	1.6e-3	2:51
toruspm3-8-50	512	-527.80	513	681(84)	-528.10	1.4e-3	9:28
torusg3-8	512	-457.36	513	605(89)	-457.60	1.4e-3	13.11
maxG11	800	-629.16	801	734(84)	-629.39	1.4e-3	1:49:55
maxG51	1000	-4003.81	1001	790(54)	-4008.90	1.7e-3	32:24

OPTIONS.gaptol $\leq 1e-3$. We observe that SDPT3 is faster on problems that are sparse and small, but as the problem size and density grow, the decomposition approach is much faster than SDPT3. For instance on problem test-8, with $m = n = 300$ and density $= 90\%$, we are almost twice as fast as SDPT3.

In figure 3 we plot $\log \epsilon$, where ϵ is given by (41), versus the iteration count. The plot illustrates that our algorithm converges almost linearly.

Table 2. Computational results on graph partitioning problems

Problem	n	Opt value	LP cones	SDP cones	Lower bound	ϵ	Time (h:m:s)
gpp100	100	39.91	103	80(23)	39.89	2e-3	50
gpp124-1	124	5.31	127	187(52)	5.26	2.2e-2	1:35
gpp124-2	124	42.35	127	121(39)	42.33	2.1e-3	1:24
gpp124-3	124	141.36	127	189(42)	141.29	1.4e-3	50
gpp124-4	124	387.7	127	188(39)	387.43	1.4e-3	43
gpp250-1	250	12.25	253	231(66)	12.18	1.7e-2	10:24
gpp250-2	250	73.33	253	182(49)	73.28	2.5e-3	10:11
gpp250-3	250	278.72	253	321(50)	278.53	1.6e-3	3:55
gpp250-4	250	695.83	251	358(53)	695.42	1.3e-3	3:49
gpp500-1	500	18.78	503	299(92)	18.67	2e-2	1:48:23
gpp500-2	500	138.98	503	277(65)	138.89	2.5e-3	2:11:08
gpp500-3	500	476.96	503	522(63)	476.72	1.3e-3	40:21
gpp500-4	500	1467.92	503	592(56)	1466.90	1.6e-3	26:22
bm1	882	17.65	885	1272(339)	17.47	4.3e-2	26:31:13

Table 3. Computational results on Lovasz theta problems

Problem	n	m	Opt value	LP blocks	SDP blocks	Lower bound	ϵ	Time (h:m:s)
theta-1	50	104	-23	208	141(35)	-23.01	1.7e-3	1:09
theta-2	100	498	-32.87	996	396(37)	-32.90	1.5e-3	38:20
theta-3	150	1106	-42.17	2212	608(40)	-42.23	3e-3	5:15:52

The distribution of solution times for the various components of the algorithm on the maxG11 problem are illustrated in figure 4. A final note on our computational experiences is in order: We also considered approximating a semidefinite

Table 4. Computational results on random semidefinite programs

Problem	n	m	Density	LP blocks	SDP blocks	Decomp bound	Decomp (h:m:s)	SDPT3 bound	SDPT3 (h:m:s)
test-1	100	501	0.1	1100	45(9)	424.68	1:06	424.97	43
test-2	100	501	0.9	1100	140(28)	181.91	5:21	182.02	1:32
test-3	500	101	0.1	700	50(25)	33.36	4:33	33.38	4:31
test-4	500	101	0.9	700	116(29)	-37.99	4:49	-37.96	6:35
test-5	1000	51	0.1	1100	24(12)	7.99	18:40	7.99	16:03
test-6	1000	51	0.9	1100	86(19)	-13.33	13:25	-13.32	17.30
test-7	300	300	0.1	900	105(21)	147.51	2:39	147.56	4:13
test-8	300	300	0.9	900	142(18)	22.91	4:52	22.92	8:11

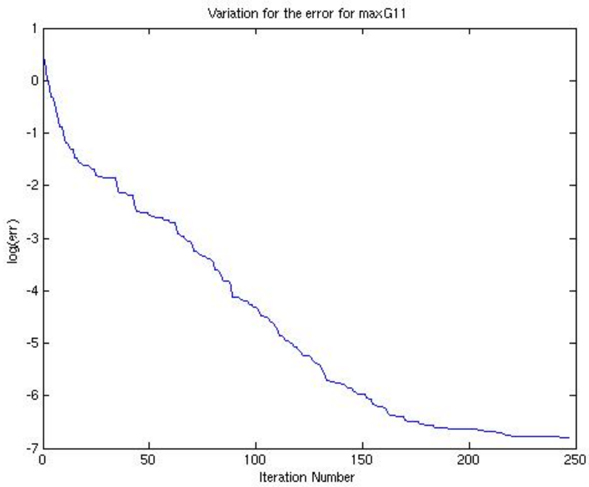

Fig. 3. Variation of $\log \epsilon$ for maxG11

cut of size r by $\binom{r}{2}$ second order cuts of size 3 in our computational experiments (see also Kim et al. [24] and Oskoorouchi & Mitchell [44] for related work). In this case, our master problem is a mixed conic problem over linear and second order

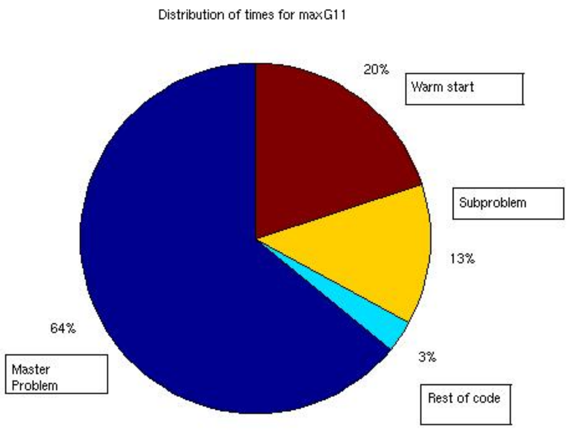

Fig. 4. Distribution of solution times for maxG11

cones and we solved the problem in SDPT3 using the Nesterov-Todd (NT) [42] search direction. We tested this approach on all the problems mentioned in this paper. However, our conic decomposition scheme with the linear and semidefinite cones was consistently better than the scheme with linear and second-order cones and we have chosen not to present these results here.

9. Future extensions to structured semidefinite programs

The decomposition approach is especially appealing when the underlying semidefinite program has a special structure such as the *block angular* structure (see Bertsekas [4] and Lasdon [34]). We consider extensions of our decomposition approach to block angular semidefinite programs in this section. Consider the semidefinite programming problem

$$\begin{aligned} \min \ & \sum_{i=1}^{J} C_i \bullet X_i \\ \text{s.t.} \ & \sum_{i=1}^{J} A_i(X_i) = b, \\ & X_i \in \mathcal{C}_i, \ i = 1, \ldots, J. \end{aligned} \tag{42}$$

where the components $C_i, X_i \in \mathcal{S}^{n_i}$ are symmetric matrices, the $A_i : \mathcal{S}^{n_i} \to I\!R^m$ are linear operators, and $b \in I\!R^m$. The sets $\mathcal{C}_i$ are bounded convex feasible sets for semidefinite programs that are described by linear matrix inequalities (LMI's). The constraint set of (42) comprises of J independent sets $\mathcal{C}_i$, $i = 1, \ldots, J$, where each set contains its own (private) variables X_i. The objective function $C \bullet X = \sum_{i=1}^{J} C_i \bullet X_i$ is also block separable. The constraints $\sum_{i=1}^{J} A_i(X_i) = b$ represent common resources shared by the r subsystems, and couple the components X_i. Recently, SDPs with such a structure have appeared in the distributed analysis of networked systems (see Langbort et al. [33]), and two stage stochastic semidefinite programs with recourse (see Mehrotra & Özevin [36,37]).

Our conic decomposition approach solves (42) in an iterative fashion between a coordinating master problem of the form

$$\theta(y) = b^T y + \sum_{i=1}^{J} \min_{X_i \in \mathcal{C}_i} \{C_i \bullet X_i - y^T(A_i(X_i))\}, \tag{43}$$

where $y \in I\!R^m$ is a set of dual multipliers for the coupling constraints; and a set of J subproblems, where the ith subproblem is

$$\theta_i(y) = \min_{X_i \in \mathcal{C}_i} \{C_i \bullet X_i - y^T(A_i(X_i))\}, \tag{44}$$

and $\theta(y) = b^T y + \sum_{i=1}^{J} \theta_i(y)$. The master problem (43) can be set up as a mixed conic problem over linear and semidefinite cones in the form (9). The subproblems (44) are smaller semidefinite programs and they are solved in parallel on different processors. In the future, we will test our algorithm on SDPs of the form (43) in a high performance parallel and distributed computing environment.

10. Conclusions and future work

We described an interior point conic decomposition approach for solving large scale semidefinite programs in this paper. The approach solves a semidefinite program over one semidefinite cone in an iterative fashion between a master problem which is a mixed conic program over linear and semidefinite cones and a subproblem which is a simple separation oracle for the semidefinite cone. Depending on the multiplicity of the minimum eigenvalue of the dual slack matrix, the oracle returns either a column or a matrix in the primal master problem. The master problem is solved with a conic solver in a feasible primal-dual IPM setting. We also discussed several issues required for an efficient implementation of the approach, including a) a stabilized matrix generation approach for improving convergence, b) a technique to warm-start the master problem after matrix generation, and c) a procedure to detect and aggregate unimportant linear and semidefinite cones in the master problem.

Our computational experiments indicate that our algorithm has the potential to outperform IPMs on large and dense semidefinite programs. Most of the time in each iteration of our decomposition algorithm is spent in solving the master problem. In fact, this problem is solved more quickly as the iterations proceed: this is because the master problem becomes a better approximation of the original SDP and the warm-start procedure helps in quickly re-optimizing the master problem after column/matrix generation. The aggregation procedure also plays an important role in keeping the size of this problem small. A potential bottleneck in our algorithm is the choice of the Lanczos scheme in solving the subproblem. Initially the eigenvalues of the dual slack matrix are well-separated and the Lanczos scheme works well. However, as the iterations proceed, the smaller eigenvalues of the dual slack matrix tend to coalesce together (see Pataki [47]), and this slows down the Lanczos scheme. In our computational experiments, the slowdowns of the Lanczos scheme were especially prominent on some of the randomly generated SDPs. We are currently investigating techniques to speed up the Lanczos scheme. The feasible region of the dual SDPs is usually unbounded and this presents a difficulty in the choice of the initial master problem. In the future, we will incorporate the trust region philosophy (see Conn et al. [9] and Schramm & Zowe [51]) within our algorithm to ensure that intermediate dual master problems and associated dual prices are bounded.

We also conclude with some avenues of future research:

1. The convergence and the complexity of the decomposition scheme is a topic of future research.
2. Extensions of our algorithm to solve structured semidefinite programs (42) with a block angular structure, in a parallel & distributed high performance computing environment. This will also serve as an alternative to recent efforts

in parallelizing IPMs (see Kojima [25] and Yamashita et al. [55]) for solving large scale semidefinite programs.

3. The *matrix completion* procedure of Fukuda et al. [13] (see also Lu et al. [35]) allows one to process a semidefinite program, whose data matrices are sparse and which is not already in the block-angular form (42), into an equivalent formulation with this structure. We will use the matrix completion approach as a preprocessing phase in our overall decomposition algorithm for solving unstructured semidefinite programs in the future.
4. Our computational results indicate that conic decomposition approach is capable of solving semidefinite relaxations of combinatorial optimization problems quickly, albeit to a limited accuracy. It can be used in the pricing phase of an SDP based conic branch-cut-price approach for mixed-integer and nonconvex programs. We described an SDP based polyhedral cut-price algorithm for the maxcut problem in Krishnan and Mitchell [31]; a conic branch-cut-price algorithm can be viewed as an extension of this approach.

References

1. F. Alizadeh, J.P.A. Haeberly, and M.L. Overton, *Complementarity and Nondegeneracy in Semidefinite Programming*, Mathematical Programming, 77, 111-128, 1997.
2. E.D. Andersen, C. Roos, and T. Terlaky, *A primal-dual interior-point method for conic quadratic optimization*, Mathematical Programming, 95(2003), pp. 249-277.
3. A. Ben Tal and A. Nemirovskii, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2001.

4. D.P. Bertsekas, *Nonlinear Programming*, 2nd edition, Athena Scientific, Belmont, Massachusetts, 1999.

5. J.F. Bonnans and A. Shapiro, *Perturbation analysis of optimization problems*, Springer, 2000.

6. B. Borchers, *SDPLIB 1.2, a library of semidefinite programming test problems*, Optimization Methods and Software, 11 & 12 (1999), pp. 683-690. Available at `http://infohost.nmt.edu/~sdplib/`.

7. S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

8. O. Briant, C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck, *Comparison of bundle and classical column generation*, Rapport de researche no 5453, INRIA, January 2005. Available at `http://www.inria.fr/rrrt/rr-5453.html`.

9. A.R. Conn, N.I.M. Gould, and P.L. Toint, *Trust-Region Methods*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2000.

10. G.B. Dantzig and P. Wolfe, *The decomposition principle for linear programming*, Operations Research, 8(1960), pp. 101-111.

11. E. De Klerk, *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*, Applied Optimization Series, Vol. 65, Kluwer Academic Publishers, May 2002.

12. O. du Merle, D. Villeneuve, J. Desroisers, and P. Hansen, *Stabilized column generation*, Discrete Mathematics, 194(1999), pp. 229-237.

13. M. Fukuda, M. Kojima, K. Murota, and K. Nakata, *Exploiting sparsity in semidefinite programming via matrix. completion I: General framework*, SIAM Journal on Optimization, 11(2000), pp. 647-674.

14. M. Goemans and D.P. Williamson, *Improved approximation algorithms for max cut and satisfiability problems using semidefinite programming*, Journal of the A.C.M, 42 (1995), pp. 1115-1145.

15. J.L. Goffin, A. Haurie, and J.P. Vial, *Decomposition and nondifferentiable optimization with the projective algorithm*, Management Science, 38(1992), pp. 284-302.

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

17. C. Helmberg, *Semidefinite Programming for Combinatorial Optimization*, Habilitation Thesis, ZIB-Report ZR-00-34, Konrad-Zuse-Zentrum Berlin, October 2000. Available at `http://www.zib.de/PaperWeb/abstracts/ZR-00-34/`.

18. C. Helmberg, *Numerical evaluation of spectral bundle method*, Mathematical Programming, 95(2003), pp. 381-406.

19. C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM Journal on Optimization, 10(2000), pp. 673-696.

20. C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz, *An interior point method for semidefinite programming*, SIAM Journal on Optimization, 6(1996), pp. 342-361.

21. J.B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*, Springer-Verlag, 1993.

22. R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

23. J.E. Kelley, *The cutting plane method for solving convex programs*, Journal of the SIAM, 8(1960),pp. 703-712.

24. S. Kim, M. Kojima, and M. Yamashita, *Second order cone programming relaxation of a positive semidefinite constraint*, Optimization Methods and Software, 18(2003), pp. 535-541.

25. M. Kojima, *Parallel computing for semidefinite programming*, Plenary talk, 2005 SIAM Conference on Optimization, Stockholm, Sweden, 2005. Available at `http://www.is.titech.ac.jp/~kojima/articles/SIOPT05Web.pdf`.

26. M. Kojima, S. Shindoh, and S. Hara, *Interior point methods for the monotone linear complementarity problem in symmetric matrices*, SIAM Journal on Optimization, 7(1997), pp. 86-125.

27. K. Krishnan Sivaramakrishnan, *Linear Programming Approaches to Semidefinite Programming Problems*, Ph.D. thesis, Department of Mathematical Sciences, Rensselaer Polytechnic Institute, July 2002. Available at `http://www4.ncsu.edu/~kksivara/publications/rpithes.pdf`.

28. K. Krishnan Sivaramakrishnan and J.E. Mitchell, *Properties of a cutting plane algorithm for semidefinite programming*, Technical Report, Department of

Mathematical Sciences, Rensselaer Polytechnic Institute, May 2003. Available at http://www4.ncsu.edu/~kksivara/publications/properties.pdf.

29. K. Krishnan Sivaramakrishnan and J.E. Mitchell, *Semi-infinite linear programming approaches to semidefinite programming problems*, Novel Approaches to Hard Discrete Optimization, edited by P.M. Pardalos and H. Wolkowicz, Fields Institute Communications Series, American Mathematical Society, 2003, pp. 123-142.

30. K. Krishnan Sivaramakrishnan and J.E. Mitchell, *An unifying framework for several cutting plane algorithms for semidefinite programming*, Optimization Methods and Software, 21(2006), pp. 57-74.

31. K. Krishnan Sivaramakrishnan and J.E. Mitchell, *A semidefinite programming based polyhedral cut and price algorithm for the maxcut problem*, to appear in Computational Optimization and Applications, 2006. Available at http://www4.ncsu.edu/~kksivara/publications/coap-final.pdf.

32. K. Krishnan Sivaramakrishnan and T. Terlaky, *Interior point and semidefinite approaches in combinatorial optimization*, GERAD 25th Anniversary Volume on *Graph Theory and Combinatorial Optimization*, edited by D. Avis, A. Hertz, and O. Marcotte, Springer, 2005, pp. 101-157.

33. C. Langbort, L. Xiao, R. D'Andrea, and S. Boyd, *A decomposition approach to distributed analysis of networked systems*, Proceedings of the 43rd IEE Conference on Decision and Control, pp. 3980-3985. Available at http://www.stanford.edu/~boyd/dist_contr_cdc04.html.

34. L. Lasdon, *Optimization Theory for Large Systems*, Dover Publications Inc., 2002 (Dover reprint of the original 1970 Macmillan edition).

35. Z. Lu, A. Nemirovskii, and R.D.C. Monteiro, *Large scale semidefinite programming via saddle point mirror-prox algorithm*, Technical report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, November 2004. Available at http://www.optimization-online.org/DB_HTML/2004/11/998.html.

36. S. Mehrotra and M.G. Özevin, *Two-stage stochastic semidefinite programming and decomposition based interior point methods: Theory*, Technical Report 04-016, Department of Industrial Engineering & Management Sciences, Northwestern University, December 2004. Available at http://www.optimization-online.org/DB_HTML/2005/01/1040.html.

37. S. Mehrotra and M.G. Özevin, *On the implementation of interior point decomposition algorithms for two-stage stochastic programs*, Technical Report 05-04, Department of Industrial Engineering & Management Sciences, Northwestern University, October 2005. Available at http://www.iems.northwestern.edu/images/PDF/SDP_computational.pdf.

38. R.D.C. Monteiro, *Primal-dual path following algorithms for semidefinite programming*, SIAM Journal on Optimization, 7(1997), pp. 663-678.

39. R.D.C. Monteiro, *First and second order methods for semidefinite programming*, Mathematical Programming, 97(2003), pp. 209-244.

40. A.S. Nemirovskii and D.B. Yudin, *Problem Complexity and Method Efficiency in Optimization*, John Wiley, Chichester, 1983.

41. Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, Boston, 2004.

42. Y. Nesterov and M.J. Todd, *Self scaled barriers and interior point methods in convex programming*, Mathematics of Operations Research, 22(1997), pp. 1-42.

43. M.R. Oskoorouchi and J.L. Goffin, *A matrix generation approach for eigenvalue optimization*, Technical report, College of Business Administration, California State University, San Marcos, September 2004. Available at http://www.optimization-online.org/DB_HTML/2004/04/856.html.

44. M.R. Oskoorouchi and J.E. Mitchell, *A second order cone cutting surface method: complexity and application*, Technical report, California State University San Marcos, San Marcos, California, May 2005. Available at http://www.optimization-online.org/DB_HTML/2005/05/1129.html.

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

46. B. Parlett, *The Symmetric Eigenvalue Problem*, Classics in Applied Mathematics, SIAM, Philadelphia, 1998.

47. G. Pataki, *On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues*, Mathematics of Operations Research, 23(1998), pp. 339-358.

48. G. Pataki and S. Schmieta, *The DIMACS library of mixed semidefinite, quadratic, and linear programs*, Available at http://dimacs/rutgers.edu/Challenge/Seventh/Instances.

49. J. Peng, C. Roos, and T. Terlaky, *Self Regularity: A New Paradigm for Primal-Dual Interior-Point Algorithms*, Princeton University Press, NJ, 2002.

50. J. Renegar, *A Mathematical View of Interior Point Methods in Convex Optimization*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2001.

51. H. Schramm and J. Zowe, *A version of the bundle method for minmizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results*, SIAM Journal on Optimization, 2(1992), pp. 121-152.

52. J.F. Sturm, *Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11 & 12(1999), pp. 625-653. Available at `http://sedumi.mcmaster.ca`.

53. R.H. Tütüncü, K.C. Toh, and M.J. Todd, *Solving semidefinite-quadratic-linear programs using SDPT3*, Mathematical Programming, 95(2003), pp. 189-217. Available at `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`.

54. H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook on Semidefinite Programming*, Kluwer Academic Publishers, Boston, 2000.

55. M. Yamashita, K. Fujisawa, and M. Kojima, *SDPARA: SemiDefinite Programming Algorithm PARAllel version*, Parallel Computing, 29(2003), pp. 1053-1067. Available at `http://grid.r.dendai.ac.jp/sdpa/sdpa.6.00/sdpara.index.html`.

56. Y. Zhang, *On extending some primal-dual interior point methods from linear programming to semidefinite programming*, SIAM Journal on Optimization, 8(1998), pp. 365-386.