A matrix generation approach for eigenvalue optimization *

M. R. Oskoorouchi† J.-L. Goffin‡

March 2004

Abstract

We study the extension of a column generation technique to eigenvalue optimization. In our approach we utilize the method of analytic center to obtain the query points at each iteration. A restricted master problem in the primal space is formed corresponding to the relaxed dual problem. At each step of the algorithm, an oracle is called to return the necessary columns to update the restricted master. Since eigenvalue optimization yields to a nonpolyhedral model, at some query points the oracle generates matrices, rather than traditional columns. In this case, we update the restricted master problem by enlarging the matrix variable by a block-diagonal element. We discuss the issues of recovering feasibility after the restricted master is updated by a column or a matrix. The numerical result of implementing the algorithm on randomly generated problems is reported.

Keywords: Column generation, cutting plane technique, eigenvalue optimization, analytic center, semidefinite inequality.

---

*This work has been completed with the support of a research grant from the College of of Business Administration, California State University San Marcos.

†College of Business Administration, California State University San Marcos, San Marcos, California 92096, Email: moskooro@csusm.edu

‡GERAD/Faculty of Management, McGill University, 1001 Sherbrooke West, Montreal, QC, H3A 1G5, Canada, Email: jlg@crt.umontreal.ca

1 Introduction

Many applications in Engineering and Management can be modeled as optimization problems and solved using computational algorithms. Depending on the mathematical model, an appropriate optimization technique should be employed in order to enhance efficiency. Essentially, a large class of applications, due to their complicated disposition, give raise to models that have huge number of variables or constraints or both. These models are often too large or too complex to be handled by classical optimization techniques. Fortunately these problems often have special structures.

The idea of decomposition and *column generation*, introduced by Dantzig and Wolfe [4] in the late fifties, provides an efficient approach to large problems with special structure. Column generation technique was first implemented by Gilmore and Gomory [7, 8] in the context of cutting stock problem. Since then, enormous number of applications used this technique to solve models with large number of variables.

In linear programming context, classical decomposition technique is applied to problems with special structure and exponentially large number of constraints. In this approach, the original problem is transformed into an identical model with substantially fewer constraints but exponentially large number of columns (variables), called the *master problem*. The master problem is then restricted to a reasonably small subset of variables. The dual of the restricted master is solved by simplex and the optimal dual is used to determine new columns to add to the restricted master problem.

In the economics literature, this appears under the headings of decentralization and coordination by prices. The idea of this principle is that if a model is too large, or a function is too complex, it is better not to know it entirely, but to discover it as needed by intelligent queries.

Many methodologies for linear and integer programs, that make use of the column generation idea, have been developed over the past four decades. Kelley's cutting plane method [14], Benders decomposition [2], Bundle method [15], Lagrangean relaxation [6, 12], and analytic center cutting plane method [27] are amongst the most efficient techniques of this nature. Although all of these techniques are similar in principle, there is a key difference between them. The disparity is in their tactic in selecting the query points or the columns to add to the restricted master. One interesting methodology is the cutting plane method that uses analytic center.

Analytic center cutting plane method (ACCPM) combines the decomposition principle with the interior point methodologies (discovered by Narendra Karmarkar in 1984). This method was introduced by Sonnevend [27], and developed by Goffin, Haurie, and Vial [10], and Ye [30]. ACCPM has been very successful in practice. Goffin, Gondzio, Sarkissian and Vial [9] discuss the implementation of this technique to multicommodity flow problem, Mitchell [20, 21, 22] applies ACCPM to combinatorial applications, and Elhedhli and Goffin[5] integrate ACCPM with branch-and-bound.

Recently some applications gave raise to models where vector variables are replaced by matrices and nonnegativity constraint is replaced by positive semidefiniteness. The fundamental difference between the new class of optimization problems and LP models is that the feasible region of these models is nonpolyhedral. Classical optimization techniques for nonpolyhedral models were discovered in early 90s. Although these techniques are polynomial in their theoretical complexity, but they are not very efficient when applied to large problems.

Nonpolyhedral models have been recently combined with decomposition techniques. Helmberg and Rendl [13] implement a Spectral Bundle method to the semidefinite relaxation arising from the max-cut problem and report nice numerical results. Oskoorouchi and Goffin [24] combine semidefinite programming and analytic center cutting plane method and report a fully polynomial approximation scheme. In a separate paper Oskoorouchi and Goffin [25] employ second-order cone cuts into the cutting plane techniques. Sun, Toh, and Zhao [28], Toh, Zhao, and Sun [29], and Chua, Toh, and Zhao [3] study ACCPM for convex semidefinite feasibility problem; Krishnan and Mitchell [16] use LP cutting plane scheme for semidefinite programming; and Basescu [1] studies a general form that is applied to conic optimization. Two survey papers by Krishnan and Mitchell [17] and Krishnan and Terlaky [18] include a summary of these techniques.

In this paper we implement an extension of column generation technique to nonpolyhedral models. We employ the method of centers and use the analytic center to obtain a query point at each step of the algorithm. Since we are dealing with matrix variables, at each query point our algorithm generates a matrix, rather than the traditional column, to add to the restricted master problem. The new matrix is added to the restricted master as a block diagonal. We call this extension a *matrix generation* algorithm. We show that the dual of the restricted master problem is a convex nondifferentiable optimization problem whose epigraph consists of semidefinite surfaces. We closely follow the primal and dual problems. The dual space is used to give a geometric insight to the reader. However, we keep our search directions and query points in the primal space.

Theoretical issues and convergence of the algorithm were studied in [24], where a fully polynomial approximation scheme is established. In this paper we explore the implementation issues of this algorithm. Although we implement matrix generation technique to eigenvalue optimization with bounds, however, the algorithm can be naturally extended to solve unbounded problems.

The paper is organized as follows: In Section 2 we discuss the mathematical properties of eigenvalue optimization. Section 3 deals with the transformation of the eigenvalue optimization into a feasibility problem. We also present the framework of the algorithm in the dual space in this section. In Section 4 we present the KKT optimality conditions and the primal algorithm for computing an approximate weighted analytic center. Section 5 presents a procedure to recover feasibility when a matrix is generated and the restricted master is updated. Finally Section 6 presents the matrix generation algorithm and numerical results when applied to eigenvalue optimization.

Notations: We use lower case letters for vectors and upper case letters for matrices. The space of $n \times n$ symmetric matrices is denoted by $\mathcal{S}^n$, trace of matrix X is indicated by $\mathbf{tr}(X)$, and for two matrices X and Y $X \bullet Y = \mathbf{tr}(X^T Y)$. We use $X \succeq 0$ ($X \succ 0$) to show that matrix X is positive semidefinite (positive definite). Operator $\mathcal{A}$ is a linear operator from $\mathcal{S}^n$ to $\Re^m$ defined by $(\mathcal{A}X)_i = A_i \bullet X$, for $A_i \in \mathcal{S}^n$, and $\mathcal{A}^T : \Re^m \to \mathcal{S}^n$ is its adjoint operator defined by

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

2 Eigenvalue optimization

In this section we introduce the eigenvalue optimization problem with bound and linear constraints and study the properties of the maximum eigenvalue function in detail.

Consider the following optimization problem:

$$\begin{array}{ll} \min & \lambda_{\max}(F(y)) \\ s.t & \\ & G^T y \leq h \\ & l \leq y \leq u, \end{array} \tag{1}$$

where $F(y) = F_0 + \sum_{i=1}^{m} y_i F_i$, and $F_i \in \mathcal{S}^n$ are linearly independent, $G \in \Re^{m \times r}$, $r \leq m$, is a full rank matrix, and $h \in \Re^r$. The m-vector y is the vector variable which is bounded from above by u and from below by l. We call m the size of the problem and n the dimension. We assume that the feasible set of Problem (1) satisfies a Slater's condition. That is there exists a point y that strictly satisfies all inequalities. We refer to such a point as an interior point of the feasible set.

The maximum eigenvalue of the affine combination of symmetric matrices is well studied in the literature. In a survey paper, Lewis and Overton [19] provide a comprehensive analysis of the mathematics of eigenvalue functions. To keep the paper self-contained we present the most important facts selected from the above paper and from Overton [26]. Let

$$f(y) = \lambda_{\max}(F_0 + \sum_{i=1}^{m} y_i F_i)$$

It is known that the maximum eigenvalue of a symmetric matrix is a convex nondifferentiable function and can be written as a semidefinite programming problem. More precisely, if A is a symmetric matrix, then the largest eigenvalue of A can be obtained via

$$\lambda_{\max}(A) = \max \left\{ A \bullet U : \mathbf{tr}(U) = 1, U \succeq 0 \right\}, \tag{2}$$

The following example gives a geometric insight of the maximum eigenvalue function.

Example 1 *Let* $F_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, $F_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, *and* $F_0 = I$, *then*

$$F(y) = \begin{pmatrix} 1 + y_1 & y_2 \\ y_2 & 1 - y_1 \end{pmatrix},$$

and $\lambda_{\max}(F(y)) = 1 + \sqrt{y_1^2 + y_2^2}$.

Figure 1 plots the maximum eigenvalue function of Example 1. It illustrates a clear pictorial view of the properties of the maximum eigenvalue function that were mentioned above. In this example $f(y)$ is a convex cone that is nondifferentiable at $y = 0$ which happens to be the minimum. In general, f is differentiable at y if the maximum eigenvalue has multiplicity one. However, in practice we are dealing with functions that do not possess this smooth property. In such situations, one can work with the set of subgradients of f at each query point rather than the gradient.

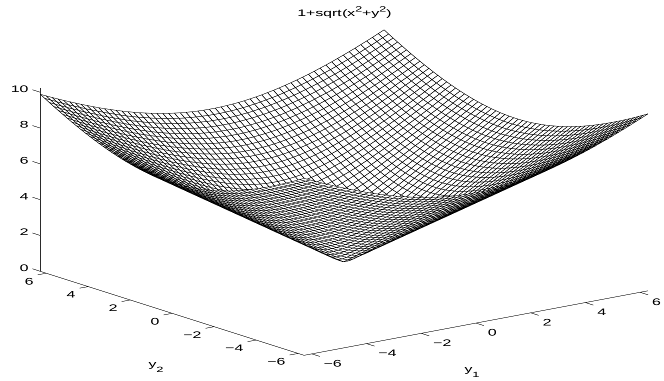

Figure 1: Graph of the maximum eigenvalue of the affine combination of symmetric matrices F_0, F_1 and F_2

The subdifferential of function f at point $\bar{y}$ can be obtained using the chain rule and the Clark generalized gradient:

Theorem 2 *Let $\bar{y}$ be in domain of f and the maximum eigenvalue of $f(\bar{y})$ has multiplicity p, with a corresponding orthonormal basis of eigenvectors $\bar{Q} = [\bar{q_1}, ..., \bar{q_p}]$. The generalized gradient of f at $\bar{y}$ is*

$$\partial f(\bar{y}) = \{w \in \Re^m : w_i = F_i \bullet \bar{Q}U\bar{Q}^T : U \succeq 0, \mathbf{tr}(U) = 1\}. \tag{3}$$

The eigenvector matrix $\bar{Q}$ plays a key role in our algorithm. In the next section we discuss the transformation of the optimization problem into a feasibility problem.

3 From optimization to feasibility

In this section we first transform the eigenvalue optimization into a convex feasibility problem by means of subgradients of the maximum eigenvalue function. We then present the framework of our algorithm in the dual space. The matrix generation algorithm is presented in detail in Section 6. Let

$$\mathcal{F} = \left\{ y \in \Re^m : G^T y \leq h, \text{ and } l \leq y \leq u \right\}$$

be the feasible set of Problem (1). $\mathcal{F}$ is a convex set described by linear inequalities, which is bounded and contains a strictly feasible point as assumed. Now consider the objective function of Problem (1). In view of (2) we have

$$f(y) = \max\left\{F(y) \bullet U : \mathbf{tr}(U) = 1, U \succeq 0\right\}. \tag{4}$$

Problem (4) is a semidefinite programming problem. By restricting positive semidefinite matrix U to a subcone generated by the faces of the cone at a feasible point $\bar{y} \in \mathcal{F}$, we obtain a lower bound for $f(y)$. Let $\bar{y} \in \mathcal{F}$ and the maximum eigenvalue of $F(\bar{y})$ has multiplicity p and $\bar{Q} \in \Re^{n\times p}$ be a matrix whose columns form a basis for the eigenspace of the maximum eigenvalue. That is

$$\left(F_0 + \sum \bar{y}_i F_i\right)\bar{Q} = \lambda_{\max}\left(F_0 + \sum \bar{y}_i F_i\right)\bar{Q} \tag{5}$$

and define

$$\begin{aligned}\bar{f}(y) &= \max\left\{F(y)\bullet \bar{Q}U\bar{Q}^T : \mathbf{tr}(U) = 1, U \succeq 0\right\}\\ &= \lambda_{\max}\left(\bar{Q}^T F(y)\bar{Q}\right)\end{aligned} \tag{6}$$

$\bar{f}(y)$ is a convex function that establishes a lower bound on $f(y)$. That is

$$\bar{f}(y) \le f(y), \text{ for all } y \in \Re^m.$$

Moreover $\bar{f}(\bar{y}) = f(\bar{y})$.

Let $\bar{\theta}$ and $\underline{\theta}$ be upper and lower bounds on the optimal objective value of Problem (1) respectively, and let

$$\Omega_D = \left\{y \in \mathcal{F}, z \in \Re : \underline{\theta} \le z \le \bar{\theta}, \bar{f}(y) \le z\right\}.$$

Note that Ω_D is the area bounded by the feasible region of Problem (1), a set of subgradients of f at $\bar{y}$ and the hyperplanes $z = \bar{\theta}$ and $z = \underline{\theta}$. Furthermore, since

$$\bar{f} \le f^* = \min f(y) \le \bar{\theta},$$

Ω_D contains the optimal solution of the eigenvalue optimization.

Let us study the inequality $\bar{f}(y) \le z$ more closely. From (6)

$$\lambda_{\max}\left(\bar{Q}^T F(y)\bar{Q}\right) \le z.$$

This implies that

$$\bar{Q}^T\left(F_0 + \sum y_i F_i\right)\bar{Q} \preceq zI,$$

or

$$\sum_{i=1}^{m} y_i\left(\bar{Q}^T F_i \bar{Q}\right) - zI \preceq -\bar{Q}^T F_0 \bar{Q}.$$

Thus the set Ω_D can be expressed as

$$\Omega_D = \left\{\hat{y} = (y,z) \in \Re^{m+1} : G^T y \le h,\ \ l \le y \le u,\ \ \underline{\theta} \le z \le \bar{\theta},\ \ \mathcal{A}^T\hat{y} \preceq C\right\},$$

where $\mathcal{A}$ is the linear operator defined in Section 1: $\mathcal{A}^T\hat{y} = \sum_{i=1}^{m+1} \hat{y}_i A_i$. Here

$$A_i = \bar{Q}^T F_i \bar{Q}, \quad i = 1, ..., m, \qquad \text{and} \qquad A_{m+1} = -I,$$

and $C = -\bar{Q}^T F_0 \bar{Q}$.

We call $\mathcal{A}^T\hat{y} \preceq C$ a *semidefinite cut (SDC)* of dimension p, and refer to A_i as the *semidefinite cut matrices*. Note that $\hat{y}$ is an $(m+1)$-vector composed of $y \in \Re^m$ and $z \in \Re$. Let us write the linear inequalities of Ω_D (except for the upper bound cut) in the matrix form:

$$\Omega_D = \left\{ \hat{y} \in \Re^{m+1} : \mathcal{A}^T\hat{y} \preceq C, \;\; A^T\hat{y} \leq c, \;\; \hat{y}_{m+1} \leq \bar{\theta} \right\}$$

where $A^T\hat{y} \leq c$ represents the feasible region of Problem (1) including the bound constraints and the lower bound on the objective function ($z \geq \underline{\theta}$). We refer to A as the *linear cut matrix*. The set Ω_D, also known as *the set of localization*, is a compact convex set with nonempty interior composed of linear inequalities and linear matrix inequalities. Moreover Ω_D contains the optimal solution set Ω^* of Problem (1). The eigenvalue optimization problem is therefore reduced to a convex feasibility problem. We develop a matrix generation algorithm based on a weighted analytic center cutting surface method to find a point in Ω^*. The algorithm starts from a strictly feasible point of Ω_D. At each iteration k, $\hat{y}^k$ a weighted analytic center of Ω_D^k is computed and a separation oracle is called. The oracle determines if either $\hat{y}^k$ is in Ω^* or returns a cut which contains the optimal solution set. If f is differentiable at $\hat{y}^k$, the oracle returns the gradient of f as a single *linear cut (LC)* and the set of localization is updated by adding this cut as a column of matrix A (traditional column generation technique). Otherwise the new cut is an SDC and we update Ω_D by adding the SDC to the diagonal of the semidefinite cut matrix (new matrix generation technique). The analytic center is recovered for the updated set by obtaining an optimal updating direction. The detail follows.

Consider the set of localization at the k^{th} iteration as

$$\Omega_D^k = \left\{ \hat{y} \in \Re^{m+1} : (\mathcal{A}^k)^T\hat{y} \preceq C^k, \;\; (A^k)^T\hat{y} \leq c^k, \;\; \hat{y}_{m+1} \leq \theta^k \right\}$$

If $\hat{y}^k$ is not in the solution set the oracle evaluates f at $\hat{y}^k$ and returns an orthonormal matrix $Q^k \in \Re^{n \times p_k}$, where p_k is the multiplicity of the maximum eigenvalue of $F(y^k)$. The matrix Q^k is used to create a new cut and to update the set of localization Ω_D^k. If $p_k > 1$, then the new cut is a semidefinite cut and the set of localization is updated via

$$\Omega_D^{k+1} = \Omega_D^k \cap \left\{ \hat{y} \in \Re^{m+1} : (\mathcal{B}^k)^T \hat{y} \preceq D^k,\ \hat{y}_{m+1} \leq \theta^{k+1} \right\},$$

where

$$(\mathcal{B}^k)^T \hat{y} = \sum_{i=1}^{m+1} \hat{y}_i B_i^k,$$

$$B_i^k = (Q^k)^T F_i Q^k, \quad i = 1, ..., m, \text{ and } B_{m+1}^k = -I,$$

$$D^k = -(Q^k)^T F_0 Q^k,$$

and

$$\theta^{k+1} = \min\{\theta^k, f(y^k)\}. \tag{7}$$

The semidefinite cut matrices A_i^k and the matrix C^k, in this case are updated by

$$A_i^{k+1} = \begin{pmatrix} A_i^k & 0 \\ 0 & B_i^k \end{pmatrix}, \text{ and } C^{k+1} = \begin{pmatrix} C^k & 0 \\ 0 & D^k \end{pmatrix},$$

That is, the dimension of the semidefinite cut matrices A_i^k is enlarged, by p_k when adding a p_k-dimensional SDC as a block diagonal.

If $p_k = 1$, i.e., f is differentiable at y^k, then B_i^k, for $i = 1, ..., m+1$ are scalars and therefore the new cut is a single linear cut. In this case we store B_i^k, for $i = 1, ..., m+1$ in a column vector $b^k \in \Re^{m+1}$ and update the linear cut matrix A^k via

$$A^{k+1} = \begin{bmatrix} A^k & b^k \end{bmatrix}.$$

Likewise, D^k is a scalar and is used to update c^k:

$$c^{k+1} = \begin{pmatrix} c^k \\ D^k \end{pmatrix}$$

Thus the updated set of localization in the case of linear cut is

$$\Omega_D^{k+1} = \Omega_D^k \cap \left\{ \hat{y} \in R^{m+1} : (A^{k+1})^T \hat{y} \leq c^{k+1},\ \hat{y}_{m+1} \leq \theta^{k+1} \right\},$$

where θ^{k+1} is defined as in (7).

In practice, as k increases the dimension of the cut matrix also increases. This, makes the analytic center of Ω_D^k to get close to the upper bound θ^k and results in loss of centrality. To overcome this difficulty, we put a weight on the upper bound cut $z \leq \theta^k$, i.e., we repeat this constraint ρ times, ($\rho \geq 1$), and compute the weighted analytic center. By trial and error we found the best value for ρ to be equal to the dimension of the current cut matrix.

In the next section we introduce the weighted analytic center and derive its optimality conditions and present a computational algorithm.

4 Weighted analytic center: optimality conditions and computational algorithms

Let n_{sd} and n_l be the number of semidefinite cuts (matrices generated) and linear cuts (columns generated) respectively and let $N_{sd} = \sum_{j=1}^{n_{sd}} p_j$ be the dimension of the current semidefinite cut matrix. For the sake of simplicity let us indicate $\hat{y}$ by y and drop the index in the dual set of localization

$$\Omega_D = \left\{ y \in \Re^{m+1} : \mathcal{A}^T y \preceq C, \;\; A^T y \leq c, \;\; y_{m+1} \leq \theta \right\},$$

where $\mathcal{A}^T y \preceq C$, with $A_i \in \mathcal{S}^{N_{sd}}$ and $C \in \mathcal{S}^{N_{sd}}$ represents an SDC with the dimension of N_{sd}; $A \in \Re^{(m+1)\times n_{lp}}$ is a matrix whose columns correspond to the linear cuts, $c \in \Re^{m+1}$; and $y_{m+1} \leq \theta$ is the upper bound cut.

The weighted analytic center of Ω_D is defined as the minimizer of the weighted dual potential function over the interior of Ω_D. The weighted dual potential function is defined via

$$\phi_D(y) = -\log\det(C - \mathcal{A}^T y) - \sum_{j=1}^{n_{lp}} \log(c_j - a_j^T y) - \rho \log(\theta - y_{m+1}),$$

where a_j is the j^{th} column of matrix A, and $\rho \geq 1$ is the weight on the upper bound cut. We sometimes denote the dual potential function by

$$\phi_D(S, s, \sigma) = -\log\det S - \sum_{j=1}^{n_{lp}} \log s_j - \rho \log \sigma,$$

where $S = C - \mathcal{A}^T y$, $s = c - A^T y$, and $\sigma = \theta - y_{m+1}$. Notice that ϕ_D is a strictly convex function on the interior of Ω_D and therefore the analytic center is uniquely defined by

$$\min_{y \in \Omega_D} \phi_D(y). \tag{8}$$

The first order optimality conditions of Problem (8) are

$$\begin{pmatrix} A_1 \bullet S^{-1} \\ \vdots \\ A_m \bullet S^{-1} \\ A_{m+1} \bullet S^{-1} \end{pmatrix} + As^{-1} + \rho\sigma^{-1} e_{m+1} = 0.$$

Let $X = S^{-1}$, $x = s^{-1}$ and $\xi = \sigma^{-1}$, then the optimality conditions read

$$\begin{array}{c} \mathcal{A}X + Ax + \rho\xi e_{m+1} = 0 \\ \mathcal{A}^T y + S = C \\ A^T y + s = c \\ y_{m+1} + \sigma = \theta \\ XS = I \\ xs = e \\ \sigma\xi = 1. \end{array} \tag{9}$$

Note that xs is the coordinate-wise product of vectors x and s, and s^{-1} is the component-wise inverse of vector s.

The optimality conditions for the weighted analytic center can be alternatively derived by the weighted primal potential function. Let

$$\Omega_P = \left\{ X \in \mathcal{S}_+^{N_{sd}}, x \in \Re_+^{n_l}, \xi \in \Re_+ : \mathcal{A}X + Ax + \rho\xi e_{m+1} = 0 \right\},$$

and let

$$\phi_P(X, x, \xi) = C \bullet X + c^T x + \rho\theta\xi - \log\det X - \sum_{j=1}^{n_l} \log x_j - \rho \log \xi.$$

Then the optimal solution of the following problem

$$\begin{array}{ll} \min & \phi_P(X, x, \xi) \\ s.t. & \\ & \mathcal{A}X + Ax + \rho\xi e_{m+1} = 0 \\ & X \succeq 0, \quad x \geq 0, \quad \xi \geq 0, \end{array}$$

satisfies system (9).

Abusing notation we sometimes denote the analytic center by (X, x, ξ) or $(X, x, \xi, y, S, s, \sigma)$.

Let

$$\eta(X, x, \xi) = \sqrt{\|XS - I\|^2 + \|xs - e\|^2 + \rho|\sigma\xi - 1|^2}, \tag{10}$$

be a measure of proximity around the center. We call $(\bar{X}, \bar{x}, \bar{\xi}, \bar{y}, \bar{S}, \bar{s}, \bar{\sigma})$ a τ-approximate analytic center if it satisfies the linear equalities of (9) and

$$\eta(X, x, \xi) \leq \tau < 1.$$

We employ the Newton method to compute an approximate analytic center of Ω_P. Consider the quadratic approximation of the the primal potential function

$$\begin{aligned}
&\phi_P(X + dX, x + dx, \xi + d\xi) = \\
&\qquad C \bullet (X + dX) + c^T(x + dx) + \rho\theta(\xi + d\xi) - \log\det(X + dX) \\
&\qquad - \sum \log(x_j + dx_j) - \rho \log(\xi + d\xi), \\
&= \quad \phi_P(X, x, \xi) + (C - X^{-1}) \bullet dX + (c - x^{-1})^T dx + \rho(\theta - \xi^{-1}) d\xi \\
&\qquad + \frac{1}{2} \mathbf{tr} X^{-1}(dX) X^{-1}(dX) + \frac{1}{2} dx^T X_{lp}^{-2} dx + \frac{\rho}{2} \xi^{-2} d\xi^2,
\end{aligned}$$

where X_{lp} is a diagonal matrix made up of x. Let (X, x, ξ) be a strictly feasible point of Ω_P. Feasible directions dX, dx and $d\xi$ should satisfy

$$\mathcal{A}(X + dX) + A(x + dx) + \rho(\xi + d\xi) e_{m+1} = 0,$$

or

$$\mathcal{A} dX + A dx + \rho d\xi e_{m+1} = 0. \tag{11}$$

Thus, we solve

$$\begin{array}{ll}
\min & \phi_P(X + dX, x + dx, \xi + d\xi) \\
s.t. & \\
& \mathcal{A} dX + A dx + \rho d\xi e_{m+1} = 0.
\end{array} \tag{12}$$

Using the KKT optimality conditions, y, dX, dx, and $d\xi$ are optimal if and only if

$$C - X^{-1} + X^{-1}(dX)X^{-1} - \mathcal{A}^T y \quad = \quad 0 \tag{13}$$

$$c - x^{-1} + X_{lp}^{-2} dx - A^T y = 0 \tag{14}$$

$$\rho\theta - \rho\xi^{-1} + \rho\xi^{-2} d\xi - \rho y_{m+1} \quad = \quad 0. \tag{15}$$

Note that y is indeed a function of X, x, and ξ. However, to simplify the notations we ignore the arguments.

By multiplying X from the right side and from the left side to (13) and then applying operator $\mathcal{A}$ one has

$$\mathcal{A}(XCX) - \mathcal{A}X + \mathcal{A}dX - (\mathcal{A}_P\mathcal{A}_P^T)y = 0. \tag{16}$$

where $\mathcal{A}_P : \mathcal{S}^n \to \Re^{m+1}$ is a linear operator and $\mathcal{A}_P^T : \Re^{m+1} \to \mathcal{S}^n$ is its adjoint operator, defined via

$$\mathcal{A}_P Y = \begin{pmatrix} X^{.5}A_1X^{.5} \bullet Y \\ \vdots \\ X^{.5}A_mX^{.5} \bullet Y \end{pmatrix}, \text{ and } \mathcal{A}_P^T y = \sum_{i=1}^{m+1} y_i X^{.5}A_iX^{.5},$$

Note that $(\mathcal{A}_P\mathcal{A}_P^T) \in \mathcal{S}^{m+1}$ with $(\mathcal{A}_P\mathcal{A}_P^T)_{ij} = \mathbf{tr}A_iXA_jX$. Since the matrices A_i are linearly independent then $\mathcal{A}_P\mathcal{A}_P^T \succ 0$.

By multiplying AX_{lp}^2 to (14) from the left side, we have

$$AX_{lp}^2c - Ax + Adx - AX_{lp}^2A^Ty = 0, \tag{17}$$

and by multiplying $\xi^2 e_{m+1}$ to (15), we have

$$\rho\theta\xi^2e_{m+1} - \rho\xi e_{m+1} + \rho d\xi e_{m+1} - \rho y_{m+1}\xi^2 e_{m+1} = 0. \tag{18}$$

Let

$$G = \mathcal{A}_P\mathcal{A}_P^T + A(X_{lp})^2A^T + \rho\xi^2 e_{m+1}e_{m+1}^T, \tag{19}$$

and

$$g = \mathcal{A}_P C_P + AX_{lp}^2c + \rho\theta\xi^2 e_{m+1}, \tag{20}$$

where $C_P = X^{.5}CX^{.5}$. Then summing up (16), (17) and (18), implies

$$Gy = g, \tag{21}$$

and thus

$$y = G^{-1}g.$$

We refer to G as the primal Gram matrix.

Substituting y into (16), (17) and (18) we derive the primal directions for computing the weighted analytic center

$$\begin{aligned} dX &= (X\mathcal{A}^TX)y + X - XCX \\ &= X - XSX \end{aligned} \tag{22}$$

$$dx = x - X_{lp}^2 s \tag{23}$$

$$d\xi = \xi - \xi^2\sigma, \tag{24}$$

where $S = C - \mathcal{A}^T y$, $s = c - A^T y$, and $\sigma = \theta - y_{m+1}$ are functions of X, x, and ξ.

Feasible directions dX, dx, and $d\xi$ should satisfy (11). This condition may not be satisfied due to the computational round-off error. We therefore project dX, dx, and $d\xi$ back to the primal null space:

Let $\bar{G}$ be the same as G where $\rho = 1$. That is

$$\bar{G} = \mathcal{A}_P \mathcal{A}_P^T + A(X_{lp})^2 A^T + \xi^2 e_{m+1} e_{m+1}^T, \tag{25}$$

and let $q \in \Re^{m+1}$ be defined via

$$q = \bar{G}^{-1} \left(\mathcal{A} dX + A dx + d\xi e_{m+1} \right). \tag{26}$$

The projection of the directions dX, dx, and $d\xi$ onto the primal null space can be derived as

$$\bar{dX} = dX - X(\mathcal{A}^T q)X, \tag{27}$$

$$\bar{dx} = dx - X_{lp}^2 A^T q, \tag{28}$$

$$\bar{d\xi} = d\xi - \xi^2 q. \tag{29}$$

Observe that

$$\begin{aligned}
&\mathcal{A}\bar{dX} + A\bar{dx} + \bar{d\xi} e_{m+1} \\
&= \mathcal{A} dX + A dx + d\xi e_{m+1} - (\mathcal{A}_P \mathcal{A}_P^T) q - (A X_{lp}^2 A^T) q - (\xi^2 e_{m+1}) q \\
&= \mathcal{A} dX + A dx + d\xi e_{m+1} - \left(\mathcal{A}_P \mathcal{A}_P^T + A X_{lp}^2 A^T + \xi^2 e_{m+1} e_{m+1}^T \right) q \\
&= \mathcal{A} dX + A dx + d\xi e_{m+1} - \bar{G} q \\
&= 0.
\end{aligned}$$

Note that $\bar{G}$ differs from G in only one element

$$\bar{G} = G - \left(\frac{(\rho + 1)\xi^2}{\rho} \right) e_{m+1} e_{m+1}^T. \tag{30}$$

and hence computing $\bar{G}$ is not an expensive task.

The following lemma shows that $S(X, x, \xi)$, $s(X, x, \xi)$, and $\sigma(X, x, \xi)$ can be characterized as a least square problem:

Lemma 3 *Let (X, x, ξ) be a strictly feasible point of Ω_P. Then the dual solutions $S(X, x, \xi)$, $s(X, x, \xi)$, and $\sigma(X, x, \xi)$ are the minimizer of the following least square problem:*

$$\begin{array}{ll} \min & \sqrt{\|X^{.5}SX^{.5} - I\|^2 + \|xs - e\|^2 + \rho|\sigma\xi - 1|^2} \\ s.t. & \\ & \mathcal{A}^T y + S = C \\ & A^T y + s = c \\ & y_{m+1} + \sigma = \theta. \end{array} \tag{31}$$

Proof. First observe that

$$\begin{aligned} & \|X^{.5}SX^{.5} - I\|^2 \\ = \ & \mathbf{tr}XSXS - 2\mathbf{tr}XS + \mathbf{tr}I \\ = \ & \mathbf{tr}X(C - \mathcal{A}^T y)X(C - \mathcal{A}^T y) - 2\mathbf{tr}X(C - \mathcal{A}^T y) + n \\ = \ & \mathbf{tr}(C_P - \mathcal{A}_P^T y)^2 - 2\mathbf{tr}(C_P - \mathcal{A}_P^T y) + n \\ = \ & y^T(\mathcal{A}_P\mathcal{A}_P^T)y - 2y^T\mathcal{A}_P C_P + 2\mathbf{tr}\mathcal{A}_P^T y + \mathbf{tr}C_P^2 - 2\mathbf{tr}C_P + n. \end{aligned} \tag{32}$$

and

$$\begin{aligned} & \|xs - e\|^2 \\ = \ & (c - A^T y)^T X_{lp}^2 (c - A^T y) - 2(c - A^T y)^T x + n \\ = \ & y^T(AX_{lp}^2 A^T)y - 2y^T AX_{lp}^2 c + c^T X_{lp}^2 c - 2c^T x + 2y^T Ax + n \end{aligned} \tag{33}$$

and

$$\begin{aligned} \rho|\sigma\xi - 1|^2 &= \rho\xi^2\sigma^2 - 2\rho\xi\sigma + \rho \\ &= \rho\xi^2 y_{m+1}^2 - 2\rho\theta\xi^2 y_{m+1} + 2\rho\xi y_{m+1} + \rho(\theta\xi - 2)\theta\xi + \rho. \end{aligned} \tag{34}$$

The proof follows by multiplying (34) by e_{m+1} and adding it up with (32) and (33) and noting that $(X, x, \xi) \in \Omega_P$, and taking the first order optimality conditions. ■

Next lemma shows the strict feasibility of the primal direction and the rate of convergence when $\eta(X, x, \xi) < 1$.

Lemma 4 *If $\eta(X, x, \xi) \le \alpha < 1$, then,*

$$\eta(X^+, x^+, \xi^+) \le \eta^2(X, x, \xi) < 1,$$

and

$$X^+ \succ 0, \quad x > 0, \quad \xi > 0.$$

Proof. Let $P_X = X^{.5}SX^{.5} - I$, $p_x = xs - e$, and $p_\xi = \xi\sigma - 1$. In view of (22)– (24)

$$dX = -X^{.5}P_X X^{.5}, \quad dx = -X_{lp}p_x, \quad \text{and} \quad d\xi = -\xi p_\xi,$$

and since $\|P_X\| < 1$, $\|p_x\| < 1$, and $|p_\xi| < 1$, then

$$X^+ = X + dX = X^{.5}(I - P_X)X^{.5} \succ 0,$$

$x^+ = x + dx = X_{lp}(e - p_x) > 0$, and $\xi^+ = \xi + d\xi = \xi(1 - p_\xi) > 0$.

To prove the rate of convergence, first observe that from Lemma 3

$$\begin{aligned} &\|P_{X^+}\|^2 + \|p_{x^+}\|^2 + \rho|p_{\xi^+}|^2 \le \\ &\quad \|(X^+)^{.5}S(X,x,\xi)(X^+)^{.5} - I\|^2 + \|X_{lp}^+ s(X,x,\xi) - e\|^2 \\ &\quad +\rho|\xi^+\sigma(X,x,\xi) - 1|^2. \end{aligned} \tag{35}$$

Now since

$$dX = -X^{.5}P_X X^{.5} = X - XS(X,x,\xi)X,$$

then $X^+ = 2X - XS(X,x,\xi)X$ and (in what follows we denote $S(X,x,\xi)$ by S)

$$\begin{aligned} \|S^{.5}X^+S^{.5} - I\|^2 &= \|S^{.5}(2X - XSX)S^{.5} - I\|^2 \\ &= \|(S^{.5}XS^{.5} - I)^2\|^2 \\ &= \mathbf{t}r(S^{.5}XS^{.5} - I)^4 \\ &= \sum(\lambda_j(S^{.5}XS^{.5}) - 1)^4 \\ &\le (\sum(\lambda_j(S^{.5}XS^{.5}) - 1)^2)^2 \\ &= \left(\|S^{.5}XS^{.5} - I\|^2\right)^2. \end{aligned}$$

Thus

$$\|(X^+)^{.5}S(X,x,\xi)(X^+)^{.5} - I\| \le \|P_X\|^2.$$

Similarly, we can prove that $\|X_{lp}^+ s(X,x,\xi) - e\| \le \|p_x\|^2$. The lemma now follows from (35). ■

Lemma 4 guarantees quadratic convergence within the primal Dikin ellipsoid. Given a strictly feasible point in Ω_P one can prove that the Newton algorithm reduces the primal potential function by a constant amount in each iteration and after a finite number of steps the measure of proximity $\eta(X,x,\xi)$ lies in the quadratic convergence region (see Oskoorouchi [23]).

In our algorithm we apply a step size α to move as far as possible along the primal direction while respecting strict primal feasibility. Consider the quadratic approximation of the weighted primal potential function with the step size α:

$$\begin{aligned}
&\phi_P(X + \alpha dX, x + \alpha dx, \xi + \alpha d\xi) \\
&= C \bullet (X + \alpha dX) + c^T(x + \alpha dx) + \rho\theta(\xi + \alpha d\xi) - \log\det(X + \alpha dX) \\
&\quad - \sum \log(x_j + \alpha dx_j) - \rho \log(\xi + \alpha d\xi), \\
&= C \bullet X + \alpha C \bullet dX + c^T x + \alpha c^T dx + \rho\theta\xi + \rho\alpha\theta d\xi - \log\det X \\
&\quad - \sum \log(1 + \alpha\lambda_j) - \sum \log x_j - \sum \log(1 + \alpha\gamma_j) \\
&\quad - \rho \log \xi - \rho \log(1 + \alpha\xi^{-1} d\xi),
\end{aligned}$$

where λ_js are the eigenvalues of the symmetric matrix $X^{-.5}(dX)X^{-.5}$ and $\gamma_j = x_j^{-1} dx_j$. Taking the derivative of ϕ_P with respect to α and setting it to zero one has

$$C \bullet dX + c^T dx + \rho\theta d\xi - \sum \frac{\lambda_j}{1 + \alpha\lambda_j} - \sum \frac{\gamma_j}{1 + \alpha\gamma_j} - \frac{\rho\xi^{-1} d\xi}{1 + \alpha\xi^{-1} d\xi} = 0.$$

We solve the above equation by a line search, where we initiate the step size α to be in the feasible interval $[0, \alpha_{\max}]$. The upper bound of the step size interval will be determined such that the updated matrix $X(\alpha)$ is positive definite and the updated vector $x(\alpha)$ and $\xi(\alpha)$ are positive.

Lemma 5 *Let $X(\alpha) = X + \alpha dX$, $x(\alpha) = x + \alpha dx$ and $\xi(\alpha) = \xi + \alpha d\xi$, where dX, dx and $d\xi$ are primal directions computed by the primal algorithm, and let λ_i and γ_j be defined as above. Then $X(\alpha) \succ 0$, $x(\alpha) > 0$ and $\xi(\alpha) > 0$ for any $\alpha \in [0, \alpha_{\max}]$, where*

$$\alpha_{\max} = \frac{-1}{\min_{i,j}(\lambda_i, \gamma_j, \xi^{-1} d\xi)} \tag{36}$$

Proof. Note that

$$X(\alpha) = X + \alpha dX = X^{.5}(1 + \alpha X^{-.5} dX X^{-.5}) X^{.5},$$

and $X(\alpha) \succ 0$ only if $\alpha\lambda_i > -1$, for all $i = 1, ..., n_{sd}$, or

$$\alpha(\min_i \lambda_i) > -1.$$

Similarly,

$$x(\alpha) = x + \alpha dx = x(e + \alpha x^{-1}dx),$$

and $x(\alpha) > 0$ only if $\alpha\gamma_j > -1$, for all $j = 1, ..., n_{lp}$, or

$$\alpha(\min_j \gamma_j) > -1,$$

and finally, $\xi(\alpha) = \xi + \alpha d\xi = \xi(1 + \xi^{-1}d\xi) > 0$, if

$$\alpha\xi^{-1}d\xi > -1.$$

Thus a feasible step size should satisfy

$$\alpha \left(\min_{i,j} \left(\lambda_i, \gamma_j, \xi^{-1}d\xi \right) \right) > -1$$

The proof is immediate now. ■

We now formally present the primal algorithm to compute an approximate weighted analytic center.

Algorithm 1 (Primal Algorithm) *Given $X^0 \succ 0$, $x^0 > 0$, $\xi^0 > 0$, and $\rho \geq 1$, let $\varepsilon = 0.25$ and k=0.*

Step 1. *Compute G^k and g^k from (19) and (20), and use the Cholesky factorization of G^k to compute y^k from (21).*

Step 2. *Compute $S^k = C - \mathcal{A}^T y^k$, $s^k = c - A^T y^k$ and $\sigma^k = \theta - y^k_{m+1}$.*

Step 3. *Calculate $\eta(X^k, x^k, \xi^k)$ from (10). If $\eta(X^k, x^k, \xi^k) < \varepsilon$, stop. Otherwise, continue.*

Step 4. *Compute $\bar{G}$ from (30) and vector q from (26).*

Step 5. *Compute the projection directions $d\bar{X}$, $\bar{dx}$ and $\bar{d\xi}$ from (27)–(29)*

Step 6. *Compute $\alpha_{\max}$ from (36) and perform a line search to obtain the step size α.*

Step 7. *Let $X^{k+1} = X^k + \alpha dX$, $x^{k+1} = x^k + \alpha dx$ and $\xi^{k+1} = \xi^k + \alpha d\xi$.*

Step 8. *Set $k = k + 1$ and return to step 1.*

So far, we showed that Problem (1) can be transformed into a feasibility problem and discussed the issues related to the computational algorithm for the weighted analytic center in primal setting.

At each iteration, the algorithm calls an oracle to return a single linear or a semidefinite cut. We must recover centrality after updating the set of localization by adding cuts. To this end, we need a strictly feasible point of Ω_P as an initial point for the Newton algorithm. We address this issue in the next section.

5 Recovering feasibility

In this section we deal with the issues in recovering the primal feasibility after adding a column or a matrix to the restricted master problem. Theoretically, a column (linear cut) can be considered as a matrix (semidefinite cut). However, in practice the distinction between the two categories is important. Treating a linear cut as a semidefinite cut would drastically increase the computation time of the algorithm. It is mostly due to the computation of the Gram matrix G. We first discuss adding a matrix:

5.1 Adding a p- dimensional matrix

Let

$$\Omega_D = \left\{ y \in \Re^{m+1} : \mathcal{A}^T y \preceq C, \;\; A^T y \leq c, \;\; y_{m+1} \leq \theta \right\},$$

be the current dual set of localization and $(\bar{X}, \bar{x}, \bar{\xi}, \bar{y}, \bar{S}, \bar{s}, \bar{\sigma})$ be an approximate analytic center where

$$\begin{array}{c} \mathcal{A}\bar{X} + A\bar{x} + \bar{\xi} e_{m+1} = 0 \\ \mathcal{A}^T \bar{y} + \bar{S} = C \\ A^T \bar{y} + \bar{s} = c \\ \bar{y}_{m+1} + \bar{\sigma} = \theta \\ \bar{X}\bar{S} = I \\ \bar{x}\bar{s} = e \\ \bar{\sigma}\bar{\xi} = \rho. \end{array} \tag{37}$$

Assume that the oracle returns a p-dimensional semidefinite cut $\mathcal{B}^T y \preceq D$ at the current iteration. The semidefinite cut is called *deep* (respectively *shallow*) if $D \prec \mathcal{B}^T \bar{y}$ (respectively $D \succ \mathcal{B}^T \bar{y}$). If $D = \mathcal{B}^T \bar{y}$, the semidefinite cut is called *central*. In any other situation, the cut is called *partially deep*.

The analytic center can efficiently be recovered after adding a shallow or a central cut to the dual set of localization as it is done in [24] for semidefinite cuts and in [11] and [31] for linear cuts. However, there is no efficient way to recover centrality in the case of deep or partially deep cuts when working in the dual space. This is the main reason that we work in the primal space and focus on the matrix generation approach. The advantage of the primal algorithm is that computing the analytic center of the updated primal set of localization can be performed efficiently, regardless of the depth of the cut. The corresponding primal set of localization reads

$$\Omega_P = \left\{ X \in \mathcal{S}_+^{N_{sd}}, x \in \Re_+^{n_l}, \xi \in \Re_+ : \mathcal{A}X + Ax + \xi e_{m+1} = 0 \right\}.$$

Let

$$\Omega_P^+ = \left\{ \begin{pmatrix} X & \\ & T \end{pmatrix} \in \mathcal{S}_+^{N_{sd}+p}, x \in \Re_+^{n_{lp}}, \xi' \in \Re_+ : \mathcal{A}X + \mathcal{B}T + Ax + \xi e_{m+1} = 0 \right\}$$

be the updated primal set of localization after adding the semidefinite cut $\mathcal{B}^T y \preceq D$. To obtain a strictly feasible point of Ω_P^+, we compute the optimal updating direction $\tilde{dX}$ by maximizing $\log\det$ of the new slack matrix over a neighborhood of the primal feasible region.

$$\begin{array}{ll} \max & \log\det T \\ s.t. & \\ & \mathcal{A}dX + \mathcal{B}T + Adx + d\xi e_{m+1} = 0 \\ & \sqrt{\|X^{-1}dX\|^2 + \|X_{lp}^{-1}dx\|^2 + |\xi^{-1}d\xi|^2} \leq 1 \\ & T \succeq 0. \end{array} \tag{38}$$

The optimality conditions of problem (38) are

$$-\tilde{T}^{-1} + \mathcal{B}^T v = 0 \tag{39}$$

$$\mathcal{A}^T v + \mu X^{-1}(dX)X^{-1} = 0 \tag{40}$$

$$A^T v + \mu X_{lp}^{-2} dx = 0 \tag{41}$$

$$v + \mu \xi^{-2} d\xi = 0 \tag{42}$$

$$\mu(1 - \|X^{-1}dX\|^2 + \|X_{lp}^{-1}dx\|^2 + |\xi^{-1}d\xi|^2) = 0 \tag{43}$$

$$\mathcal{A}dX + \mathcal{B}T + Adx + d\xi e_{m+1} = 0, \tag{44}$$

where $\mu \geq 0$ is the Lagrange multiplier associated with the norm constraint. By multiplying equation (40) from the left and from the right by X and then applying the operator $\mathcal{A}$ we have,

$$(\mathcal{A}_P \mathcal{A}_P^T)v + \mu \mathcal{A}dX = 0. \tag{45}$$

Applying the same strategy to (41)

$$(AX_{lp}^2 A^T)v + \mu A dx = 0. \tag{46}$$

Multiplying Equation (42) by $\xi^2 e_{m+1}$ and adding it up to (45) and (46) and in view of (44) one has

$$v = \mu \bar{G}^{-1} \mathcal{B} \tilde{T},$$

where $\bar{G}$ is defined as in (25).

Now the primal directions dX, dx and $d\xi$ from (40)– (42) can be derived as

$$\tilde{dX} = -X\mathcal{A}^T \bar{G}^{-1} \mathcal{B} \tilde{T} X, \tag{47}$$

$$\tilde{dx} = -X_{lp}^2 A^T \bar{G}^{-1} \mathcal{B} \tilde{T}, \tag{48}$$

$$\tilde{d\xi} = -\xi^2 \bar{G}^{-1} \mathcal{B} \tilde{T} e_{m+1}, \tag{49}$$

Notice that $\tilde{dX}$ is symmetric since $\mathcal{A}^T \bar{G}^{-1} \mathcal{B} \tilde{T}$ is symmetric. Finally from (39), $\tilde{T}$ is the unique solution of the following optimization problem:

$$\tilde{T} = \arg \min_{T \succeq 0} \{ \frac{p}{2} \mathbf{t} r T \mathcal{V} T - \log \det T \}, \tag{50}$$

where $\mathcal{V} = \mathcal{B}^T \bar{G}^{-1} \mathcal{B}$.

Let

$$X^+ = \begin{pmatrix} X + \alpha \tilde{dX} & \\ & \alpha \tilde{T} \end{pmatrix},$$

$x^+ = x + \alpha \tilde{dx}$ and $\xi^+ = \xi + \alpha \tilde{d\xi}$, for $\alpha < 1$. Starting from (X^+, x^+, ξ^+), a strictly feasible point of Ω_P^+, Algorithm (1) is employed to compute an approximate analytic center of Ω_P.

Problem (50) can be solved using the Newton method. Let

$$F(T) = \frac{p}{2} \mathbf{tr} T \mathcal{V}(T) - \log \det T.$$

Let $T \succ 0$ be given. For small symmetric dT

$$F(T + dT) = \frac{p}{2} \mathbf{tr}(T + dT) \mathcal{V}(T + dT) - \log \det(T + dT).$$

Using the quadratic approximation of $\log \det(T + dT)^{-1}$, one has

$$F(T + dT) - F(T) =$$
$$p \mathbf{tr}(dT) \mathcal{V}(T) + \frac{p}{2} \mathbf{tr}(dT) \mathcal{V}(dT) - \mathbf{tr} T^{-1} dT + \frac{1}{2} \mathbf{tr} T^{-1} (dT) T^{-1} (dT).$$

Note that

$$\begin{aligned}
\mathbf{tr}T\mathcal{V}(dT) &= \mathbf{tr}T\mathcal{B}^T(\bar{G})^{-1}\mathcal{B}dT \\
&= (\mathcal{B}T)^T(\bar{G})^{-1}\mathcal{B}dT \\
&= (\mathcal{B}dT)^T(\bar{G})^{-1}\mathcal{B}T \\
&= \mathbf{tr}(dT)\mathcal{B}^T(\bar{G})^{-1}\mathcal{B}T \\
&= \mathbf{tr}(dT)\mathcal{V}(T).
\end{aligned}$$

The Newton step dT is obtained by setting the gradient of $F(T + dT)$ with respect to dT, to zero. That is

$$p\mathcal{V}(T) + p\mathcal{V}(dT) - T^{-1} + T^{-1}(dT)T^{-1} = 0.$$

By multiplying the above equation from the left side and from the right side by T, we have

$$pT\mathcal{V}(T)T + pT\mathcal{V}(dT)T - T + dT = 0. \tag{51}$$

An explicit form of dT cannot be obtained from (51) and therefore computing an exact Newton direction seems to be impossible. To over pass this problem, in our algorithm, we approximate the quadratic term $\mathbf{tr}T\mathcal{V}T$ in $F(T)$ by a linear term. That is we ignore $\mathbf{tr}(dT)\mathcal{V}(dT)$ in $F(T + dT)$. Consequently (51) becomes

$$pT\mathcal{V}(T)T - T + dT = 0,$$

and hence

$$dT = T - pT\mathcal{V}(T)T.$$

This approximation does not significantly change the direction. Our numerical results show that the rate of convergence is still quadratic in most cases and super linear in some. As an initial point we use T^0 defined in [24, Theorem 6.2] and apply a line search to compute the step size.

5.2 Adding a column

Let Ω_P, the current set of localization, be defined as in previous Section 5.1 and assume that the oracle returns a single linear cut $b^T y \leq d$. We update Ω_P by adding a column

$$\Omega_P^+ = \left\{ X \in \mathcal{S}_+^{N_{sd}}, x \in \Re_+^{n_{lp}+1}, \xi \in R_+ : \mathcal{A}X + Ax + bx + \xi e_{m+1} = 0 \right\}.$$

To find a strictly feasible point of Ω_P^+, we maximize $\log x$ subject to the primal feasibility

$$\begin{array}{ll} \max & \log x \\ s.t. & \\ & \mathcal{A}dX + Adx + bx + d\xi e_{m+1} = 0 \\ & \sqrt{\|X^{-1}dX\|^2 + \|X_{lp}^{-1}dx\|^2 + |\xi^{-1}d\xi|^2} \leq 1 \\ & x \geq 0. \end{array} \tag{52}$$

Similar to the previous section, one can derive the optimal updating directions $\tilde{dX}$, $\tilde{dx}$, and $\tilde{d\xi}$ as

$$\tilde{dX} = -X\mathcal{A}^T\bar{G}^{-1}b\tilde{x}X, \tag{53}$$

$$\tilde{dx} = -X_{lp}^2A^T\bar{G}^{-1}b\tilde{x}, \tag{54}$$

$$\tilde{d\xi} = -\xi^2\bar{G}^{-1}b\tilde{x}e_{m+1}, \tag{55}$$

where

$$\tilde{x} = \left(\sqrt{b^T\bar{G}^{-1}b}\right)^{-1}.$$

6 Numerical results

In this section we present the matrix generation algorithm and its numerical results. In our test problems we assume that Problem (1) consists of only the bound constraints. Clearly, problems with additional linear constraints are solved slightly faster (with less cuts) since the presence of additional linear constrains results in smaller initial set of localization and therefore faster convergence.

Consider

$$\begin{array}{ll} \min & \lambda_{\max}(F(y)) \\ s.t & \\ & l \leq y \leq u, \end{array} \tag{56}$$

where $F(y) = F_0 + \sum_{i=1}^m y_iF_i$, and F_i are linearly independent. Let us formally present the algorithm.

Algorithm 2 *Let* $\varepsilon = 5 \times 10^{-3}$, $\rho = 1$, $\bar{\theta}^0$ *and* $\underline{\theta}$ *be large positive and negative numbers respectively, and* $k = 0$.

Input: F_i, $i = 0, 1, ..., m$, *where* $F_i \in \mathcal{S}^n$*; limits* $l \in \Re^m$ *and* $u \in \Re^m$.

Step 0. *Compute initial points m-vector* $y^0 = \frac{1}{2}(l + u)$*,* $(2m + 1)$*-vector* $x^0 = [(u - y^0)^{-1}; \frac{1}{2}(\bar{\theta}^0 - \underline{\theta})^{-1}; (y^0 - l)^{-1}]$*, and scalars* $z^0 = \frac{1}{3}(\bar{\theta}^0 + \underline{\theta})$ *and* $\xi^0 = \frac{1}{2}(\bar{\theta}^0 - \underline{\theta})^{-1}$*.*

Step 1. *Evaluate* $F(y^k)$ *and obtain the maximum eigenvalue with multiplicity* p_k *and its orthonormal matrix* Q^k*.*

Step 2. *If* $F(y^k) < \bar{\theta}^k$*, then* $\bar{\theta}^{k+1} = F(y^k)$*.*

Step 3. *If* $p_k = 1$*, then update the set of localization as in Section 5.2, and calculate a strictly primal feasible directions* $\tilde{dX}$*,* $\tilde{dx}$*, and* $\tilde{d\xi}$ *via (53)–(55).*

Step 4. *If* $p_k > 1$*, then compute the primal directions* $\tilde{dX}$*,* $\tilde{dx}$*, and* $\tilde{d\xi}$ *via (47)–(49).*

Step 5. *Project the primal directions back to the primal null space by (27)–(29).*

Step 6. *Set* $\rho = \sum_{j=0}^{k} p_j$ *and implement Algorithm 1 to compute an approximate weighted analytic center of the updated set of localization, starting from the strictly feasible point derived in Step 5.*

Step 7. *If* $\bar{\theta}^k - z^k < \varepsilon z^k$*, stop.*

Step 8. *Set* $k = k + 1$ *and return to step 1.*

We use the mathematical package MATLAB 6.0 to code Algorithm 2. The data for the test problems were generated using the MATLAB function *RANDN* that generates Normally distributed random numbers. The program was run on an 1.60 GHz Intel Pentium 4 with 384 MB of RAM.

Figure 2 shows the convergence of Algorithm 2 when implemented on a problem of size 200 and dimension 100 with full density. The upper curve plots the values of $\bar{\theta}^k$ at each iteration and the lower curve shows the values of z^k. The algorithm stops when the relative error between the two points is less than ε as it is specified by Step 7 of the algorithm.

Note that the lower bound $\underline{\theta}$ on the objective function will never get updated. This is because there is no efficient way to approximate a good lower bound on the objective function. We overcome this difficulty by using the weighted analytic center. That is, since we compute the weighted analytic center at each iteration, z^k gets close to the upper bound cut only when the

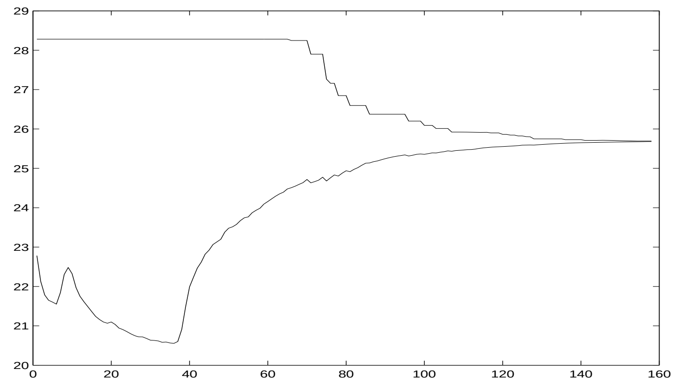

Figure 2: The convergence behavior of Algorithm 2

set of localization is small enough, thus solving the problem. We use this strategy as the stopping criterion.

Figure 2 gives a clear pictorial view of the behavior of Algorithm 2. In the beginning, the upper bound cut $z \leq \bar{\theta}^k$ is not updated very often. Speaking in dual terminologies, the reason is that the set of localization is large at the start of the algorithm. As the iterations advance, the oracle creates more cuts and the set of localization is shrunk. Once the localization set is small enough, the algorithm starts generating query points that yield to a lower objective and therefore updating the upper bound cut. Obviously, problems with more constraints have a smaller set of localization to start with, and the upper bound cut starts getting updated sooner, which of course, results in faster convergence. This is the reason to our earlier statement that the algorithm performs better on problems with more linear constraints.

Tables 1–4 present numerical results of Algorithm 2 when applied to problems with various sizes, dimensions and densities. The first two columns indicate the size (m) and dimension (n) of the problem. The third column shows the density of the problem. The density is determined when the random data is generated by the MATLAB function SPRANDN. For instance, an $m \times n$ matrix with the density of 5% means that the number of nonzero entrees of the matrix is approximately $0.05 \times m \times n$. We use problems with 5%, 50%, and 100% density. The next column represents n_l, the number of columns generated (linear cuts). The number of matrices generated (semidefinite cuts) throughout the algorithm is given by n_{sd} in the fifth column. Finally, the last column shows the CPU time in the format *minute:second*.

A close study of the numerical results presented in Tables 1–4 reveals that

Table 1: Performance of the matrix generation algorithm for $m = 50$ and $n = 50, 100, 200, 300,$ and 500.

size m	dimension n	density %	columns n_l	matrices n_{sd}	CPU time *mm:ss*
		5	97	23	00:24
	50	50	108	22	00:31
		100	8	0	00:00
		5	82	33	00:51
	100	50	89	34	01:05
		100	26	0	00:03
		5	79	40	02:19
50	200	50	104	35	02:59
		100	44	21	00:37
		5	76	50	04:30
	300	50	87	34	05:56
		100	97	36	03:29
		5	99	42	12:52
	500	50	66	54	16:31
		100	75	53	11:33

problems with higher density need less cuts to optimize. This observation suggests that Algorithm 2 is more effective when applied to dense problems. This is even more visible when the algorithm is applied to larger problems. For instance in Table 2, a problem with $m = 100$ and $n = 100$ and density of 5% needs 189 columns and 47 matrices, whereas the same problem with 100% density needs only 39 columns and 3 matrices. and substantially less CPU time.

For problems where the bound constraint is active, the CPU time dramatically increases. For example, consider a problem with size $m = 800$ and dimension $n = 50$ in Table 4. When the density of this problem is 5% the bound constraint for some of the y_i's becomes active. In this case, since the upper bound and the value of z^k decrease simultaneously (Figure 3), the algorithm needs to add too many cuts before the set of localization is small

Table 2: Performance of the matrix generation algorithm for $m = 100$ and $n = 50, 100, 200, 300$ and 500.

size m	dimension n	density %	columns n_l	matrices n_{sd}	CPU time mm:ss
		5	200	36	02:05
	50	50	200	42	02:22
		100	13	0	00:01
		5	189	47	03:25
	100	50	181	52	04:25
		100	39	3	00:09
		5	158	64	08:25
100	200	50	170	57	10:02
		100	101	31	02:51
		5	153	71	16:49
	300	50	161	72	22:05
		100	147	75	12:12
		5	137	89	43:12
	500	50	283	27	69:19
		100	132	92	34:03

enough to satisfy the stopping criterion.

Note that, in this paper we are dealing with the eigenvalue optimization with bound constrains. therefore, the initial set of localization is bounded. Algorithm 2 can also be implemented for unbounded problems. For such problems one can define an artificial bound on the variables, also known as box constrains. When an iteration gets close to the boundary of the box constrain, the bound is moved away to give flexibility to the algorithm.

One weakness of Algorithm 2 is the fact that we have to compute the Gram matrix G several times at Step 6 (see Algorithm 1). As we mentioned earlier, when the maximum eigenvalue has multiplicity more than 1 the oracle returns a semidefinite cut. To update the set of localization in this case, each matrix A_i is enlarged by adding the semidefinite cut matrices to

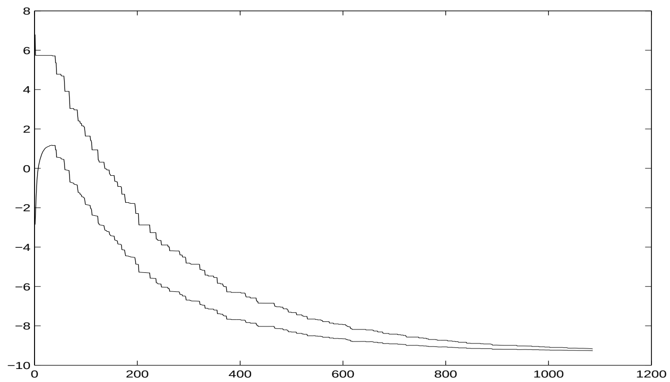

Figure 3: The convergence behavior of Algorithm 2 for problems with active bound constraint

its diagonal. Although we exploit the block-diagonal structure of A_i's, as the number of semidefinite cuts increases the computation time of $\mathcal{A}_P\mathcal{A}_P^T$ grows drastically. The use of MATLAB makes this problem even worse. MATLAB is extremely vulnerable in large loops. To overcome this difficulty, we used "C^{++}" to compute the Gram matrix and then imported it to MATLAB by writing MEX files. This modification reduced the computation time of G by a factor of 20! But it is still very expensive for large problems.

One remedy, which is to be tested in the future, is the use of weaker but less expensive cuts. One candidate is second-order cone relaxations on semidefinite inequalities. Using a set of second-order cone cuts to relax a semidefinite cut can significantly improve the computational time. However, the theoretical issues of this integration should be explored first.

7 Conclusion

We extended the column generation technique to nonpolyhedral models and implemented this extension to eigenvalue optimization. In the algorithm described in this paper, the method of analytic center is used to obtain query points at each step and update the restricted master problem. We showed that the restricted master problem is updated by a single column when the objective function in the dual space is differentiable at the query point. In the cases where this function if nondifferentiable, the oracle returns a set of subgradients and the restricted master is updated by adding a matrix to its diagonal.

Table 3: Performance of the matrix generation algorithm for $m = 200$ and 300 and $n = 50, 100, 200,$ and 300.

size m	dimension n	density %	columns n_l	matrices n_{sd}	CPU time hh:mm:ss
200	50	5	409	72	00:12:45
		50	401	76	00:10:22
		100	22	1	00:00:06
	100	5	335	96	00:20:31
		50	375	83	00:20:35
		100	74	11	00:00:48
	200	5	326	106	00:36:14
		50	323	108	00:43:22
		100	184	55	00:10:04
	300	5	283	119	01:02:49
		50	294	128	01:23:05
		100	317	121	00:49:12
300	50	5	595	99	00:40:35
		50	581	108	00:47:12
		100	43	9	00:00:55
	100	5	541	124	01:04:34
		50	570	123	00:59:15
		100	85	25	00:03:08
	200	5	460	156	01:44:25
		50	495	155	02:01:02
		100	275	71	00:27:31
	300	5	406	176	02:43:40
		50	418	52	01:16:33
		100	458	171	02:06:22

The numerical results of implementing this method on randomly generated problems illustrate that the matrix generation technique is more efficient on problems whose coefficient matrix is full rather than on sparse problems.

This technique can be improved and extended in several ways. The use of the second-order cone inequalities as an alternative to semidefinite cuts provides one line of approach. such a relaxation could improve the computational time of the algorithm. It is particularly useful when the matrix generation algorithm is integrated with branch-and-price or branch-and-cut techniques to solve integer programming problems to optimality. Such integration would be another extension to this paper.

References

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

[2] J. F. Benders, *Partitioning procedures for solving mixed-variables programming problems*, J. Soc. Ind. Appl. Math, 8 (1961), pp. 703–712.

[3] S. K. Chua, K. C. Toh, and G. Y. Zhao, *An analytic center cutting plane method with deep cuts for semidefinite feasibility problems*, Working paper, Department of Mathematics, National University of Singapore, (2002).

[4] G. B. Dantzig and P. Wolfe, *Decomposition principle for linear programs*, Operations Research, 8 (1960), pp. 101–111.

[5] S. Elhedhli and J.-L. Goffin, *The integration of an interior point cutting plane method with branch and price algorithm*, Mathematical Programming, to appear.

[6] M. L. Fisher, *The Lagrangean relaxation method for solving integer programming problems*, Management Science 27 (1981), pp. 1–18.

[7] P. C. Gilmore and R. E. Gomory, *A linear programming approach to the cutting-stock problem*, Operations Research, 9 (1961), pp. 849–859.

[8] P. C. Gilmore and R. E. Gomory, *A linear programming approach to the cutting-stock problem–Part II*, Operations Research, 11 (1963), pp. 863–888.

Table 4: Performance of the matrix generation algorithm for $m = 500, 800$ and 1000, and $n = 10, 20$ and 50.

size m	dimension n	density %	columns n_l	matrices n_{sd}	CPU time hh:mm:ss
500	10	5	88	0	00:02:28
		100	23	0	00:00:39
	20	5	203	0	00:06:46
		100	30	0	00:00:51
	50	5	1418	90	01:12:05
		100	59	2	00:01:59
800	10	5	68	8	00:07:50
		100	16	1	00:01:36
	20	5	297	1	00:31:23
		100	52	0	00:04:33
	50	5	1311	99	07:18:23
		100	73	0	00:05:08
1000	10	5	57	12	00:12:16
		100	27	2	00:03:07
	20	5	231	13	00:45:19
		100	56	0	00:09:24
	50	5	1187	73	06:22:10
		100	105	0	00:17:41

[9] 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.

[10] J.-L. Goffin, A. Haurie, and J.-P. Vial, *Decomposition and non-differentiable optimization with the projective algorithm*, Management Science, 38 (1992), pp. 284–302.

[11] 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.

[12] M. Guignard and S. Kim, *Lagrangean decomposition: A model yielding stronger lagrangean bounds*, Mathematical Programming, 39 (1987), pp. 215–228.

[13] C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM Journal on Optimization, 10(3) (2000) pp. 673–696.

[14] J. E. Kelley Jr., *The cutting-plane method for solving convex programs*, J. Soc. Ind. Appl. Math, 8 (1961), pp. 703–712.

[15] K. C. Kiwiel, *An aggregate subgradient method for nonsmooth convex minimization*, Mathematical Programming, 27 (1983), pp. 320–341.

[16] K. Krishnan and J. Mitchell, *Semi-infinite linear programming approaches to semidefinite programming*, Novel Approaches to Hard Discrete Optimization (P. M. Pardalos and H. Wolkowicz editors), Fields Institute Communication Series, AMS, (2003) pp. 123–142.

[17] K. Krishnan and J. E. Mitchell, *A unifying framework for several cutting plane methods for semidefinite programming*, Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, November 2002. Revised: December 15, 2003.

[18] K. Krishnan and T. Terlaky, *Interior point and semidefinite approaches in combinatorial optimization*, Technical Report, Advanced Optimization Laboratory, AdvOL-Report No. 2004/2, January 2004.

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

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

[21] J. E. Mitchell, *Restarting after branching in the SDP approach to MAX-CUT and similar combinatorial optimization problems*, Journal of Combinatorial Optimization, 5(2):151–166, 2001.

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

[23] M. R. Oskoorouchi, *The analytic center cutting plane method with semidefinite cuts*, Ph.D. Dissertation, Faculty of Management, McGill University, Montreal, Canada July 2002.

[24] 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.

[25] M. R. Oskoorouchi and J. L. Goffin, *An interior point cutting plane method for the convex feasibility problem with second-order cone inequalities*, Technical Report, California State University San Marcos, San Marcos, California 92096, April 2003 submitted to *Mathematics of Operations Research*. Revised: October 2003 and March 2004.

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

[27] 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.

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

[29] 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), pp. 669–691.

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

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