An Interior-Point Method for Semidefinite Programming

Christoph Helmberg * Franz Rendl † Robert J. Vanderbei ‡
Henry Wolkowicz §

November 8, 1994

Program in Statistics & Operations Research
Princeton University
Princeton, NJ 08544

November 8, 1994

Revised October 1994

Abstract

We propose a new interior point based method to minimize a linear function of a matrix variable subject to linear equality and inequality constraints over the set of positive semidefinite matrices. We show that the approach is very efficient for graph bisection problems, such as max-cut. Other applications include max-min eigenvalue problems and relaxations for the stable set problem.

Key words: semidefinite programming, interior-point methods, max-cut relaxations, max-min eigenvalue problems.

AMS 1991 Subject Classification: Primary 65F15, Secondary 49M35, 90C48

---

*Technische Universität Graz, Institut für Mathematik, Kopernikusgasse 24, A-8010 Graz, Austria. Research support by Christian Doppler Laboratorium für Diskrete Optimierung.

†Technische Universität Graz, Institut für Mathematik, Kopernikusgasse 24, A-8010 Graz, Austria. Research support by Christian Doppler Laboratorium für Diskrete Optimierung.

‡Research support by AFOSR through grant AFOSR-91-0359.

§Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ont., Canada. This author thanks the Department of Civil Engineering and Operations Research, Princeton University, for their support during his stay while on research leave. Thanks also to the National Science and Engineering Research Council Canada for their support.

1 Introduction.

The continuously rising success of interior point techniques applied to Linear Programming has stimulated research in various related fields. One possible line of generalization consists in looking at linear programs over non-polyhedral cones. This type of generalization is studied in the present paper.

To be specific, let $\mathcal{M}_n$ denote the vector space of symmetric $n \times n$ matrices. Suppose $A : \mathcal{M}_n \mapsto \Re^k$, $B : \mathcal{M}_n \mapsto \Re^m$ are two linear operators and $C \in \mathcal{M}_n, a \in \Re^k, b \in \Re^m$.

We study the following optimization problem:

$$
\text{(SDP)} \qquad \begin{array}{rl} \text{maximize} & \text{tr } CX \\ \text{subject to} & a - A(X) = 0 \\ & b - B(X) \geq 0 \\ & X \succeq 0. \end{array}
$$

This is a *semidefinite linear program*, because we optimize a linear function subject to linear inequality and equality constraints over positive semidefinite matrices X. We note that the inequality constraints $b - B(X) \geq 0$ could be transformed into equality constraints by introducing slack variables. In this case X has to be replaced by a direct sum of X and 1×1 matrices, corresponding to the slacks.

Our main motivation to study this kind of problem comes from applications in discrete optimization. In particular, we will investigate a new, powerful, and tractable relaxation of the *max-cut* problem as well as other graph bisection problems. Also *max-min eigenvalue problems* over an affine parameter space fall into the present framework and can be handled with our approach.

The main contribution of the paper is the following: we will propose a primal-dual interior point algorithm for problem (SDP). Moreover, we will discuss implementation details and present some computational experiments indicating that the approach is also highly efficient in practice.

We close this section by describing research related to our work. Alizadeh, Haeberly, Jarre and Overton [2, 3, 4, 14, 15, 21, 22] consider a problem similar to ours. Algorithmically, these authors use mostly interior point based techniques to solve the problem. Alizadeh proposes a potential reduction method and shows a polynomial running time to find an ϵ-optimal solution. Jarre uses a barrier approach and works directly on the dual. Finally Overton [21, 22] studies the problem under nonlinear equality constraints. The formulations in [14, 21, 22, 4] are not in the form above, but it is an easy exercise to transform them into our model. Vandenberghe and Boyd study primal-dual potential reduction algorithms for semidefinite programs, see [27]. In [16], the monotone linear complementarity problem for symmetric matrices is investigated. The authors present several interior point approaches to this type of semidefinite program. Finally, a general framework for interior point methods applied to convex programs can be found in the monograph [20].

1.1 Preliminaries

We first collect some preliminary results and notation. We work mainly in the space $\mathcal{M}_n$ of symmetric $n \times n$ matrices, endowed with *inner product*

$$\langle U, V \rangle := \mathrm{tr}(UV^T).$$

The curly inequality symbol refers to the *Löwner partial order* induced by the cone of positive semidefinite matrices; i.e., $M_1 \preceq M_2$ ($M_1 \prec M_2$) means that $M_2 - M_1$ is positive semidefinite (positive definite, respectively). In contrast, the usual inequality symbol, $v \leq w$, refers to the partial order induced by the cone of nonnegative vectors. The *maximum eigenvalue* of M is denoted by $\lambda_{\max}(M)$. For two matrices $U = (u_{ij})$ and $V = (v_{ij})$ of the same size, $U \circ V$ denotes the Hadamard product, i.e. $(U \circ V)_{ij} = u_{ij} \cdot v_{ij}$.

Associated with any linear operator $A : \mathcal{M}_n \mapsto \Re^k$ is another linear operator, which we denote by A^T and which is defined by the adjoint relation

$$\langle A(X), y \rangle = \langle X, A^T(y) \rangle, \quad \text{for all } X \in \mathcal{M}_n, y \in \Re^k.$$

Here we have used the angle bracket both for the inner product in $\mathcal{M}_n$ and in $\Re^k$.

For any convex cone S in R^n, we let S^* denote the *dual cone*; i.e.,

$$S^* = \{ y \in \Re^n : x^T y \geq 0 \quad \text{for all } x \in S \}.$$

In a slight abuse of notation, given a real function f defined on a real domain and a vector v in $\Re^n$, we will occasionally write $f(v)$ for $(f(v_1), \ldots, f(v_n))$. Here, f will usually be the logarithm or a power function.

Finally, for X in $\mathcal{M}_n$, we let $\mathrm{diag}(X)$ denote the vector in $\Re^n$ consisting of the diagonal elements of X. Analogously, for a vector x in $\Re^n$, we let $\mathrm{Diag}(x)$ denote the diagonal matrix in $\mathcal{M}_n$ whose diagonal elements are obtained from x.

2 Duality

The general duality theory for problems such as (SDP) has been thoroughly studied, see e.g. [30].

We derive the dual to (SDP) directly using Lagrangian methods. Indeed, let ω^* denote the optimal objective value for (SDP). Introducing Lagrange multipliers $y \in \Re^k$ and $t \in \Re^m_+$ for the equality and inequality constraints, respectively, we see that

$$\begin{aligned} \omega^* &= \max_{X \succeq 0} \min_{t \geq 0, y} \quad \mathrm{tr}\; CX + y^T(a - A(X)) + t^T(b - B(X)) \\ &\leq \min_{t \geq 0, y} \max_{X \succeq 0} \quad \mathrm{tr}(C - A^T(y) - B^T(t))X + a^T y + b^T t. \end{aligned}$$

Now note that the inner maximization over X is bounded from above only if

$$A^T(y) + B^T(t) - C \succeq 0.$$

In this case the maximum occurs if complementarity holds, i.e.

$$\mathrm{tr}(C - A^T(y) - B^T(t))X = 0.$$

Thus we get the (weak) dual (DSDP).

$$\text{(DSDP)} \qquad \begin{array}{rl} \text{minimize} & a^T y + b^T t \\ \text{subject to} & A^T(y) + B^T(t) - C \succeq 0 \\ & y \in \Re^k, \quad t \in \Re^m_+. \end{array}$$

The duality gap from interchanging max and min vanishes only under additional assumptions on the Lagrangian. We tacitly assume that both problems have feasible solutions. If a constraint qualification holds, then it can be shown that both problems form a pair of dual problems and strong duality holds; i.e., the minimum attained in (SDP) coincides with the maximum attained in (DSDP), see e.g. [30]. A sufficient condition for strong duality to hold is the existence of strictly feasible interior points for both the primal and the dual problem. (Weak duality, the max being less than or equal to the min, holds by construction.)

In our applications, we will focus mostly on the special case where $A(X) = \mathrm{diag}(X)$, in which case $A^T(y) = \mathrm{Diag}(y)$.

Before developing an algorithm for (SDP), we first show in the next section that this type of problem provides a strong machinery for deriving tight bounds to several basic NP-hard optimization problems.

3 Applications

3.1 Max-Cut Problem

The max-cut problem is the problem of partitioning the node set of an edge-weighted undirected graph into two parts so as to maximize the total weight of edges *cut* by the partition. We tacitly assume that the graph in question is complete (if not, nonexisting edges can be given weight 0 to complete the graph). Mathematically, the problem can be formulated as follows (see e.g [19]). Let the graph be given by its weighted adjacency matrix A. Define the matrix $L := \mathrm{Diag}(Ae) - A$, where e is the vector of all ones. (The matrix L is called the *Laplacian matrix* associated with the graph.) If a cut S is represented by a vector x where $x_i \in \{-1, 1\}$ depending on whether or not $i \in S$, we get the following formulation for the max-cut problem.

$$\text{(MC)} \qquad \begin{array}{rl} \text{maximize} & \frac{1}{4} x^T L x \\ \text{subject to} & x \in \{-1, 1\}^n. \end{array}$$

Using $X := \frac{1}{4} x x^T$, this is equivalent to

$$\begin{array}{rl} \text{maximize} & \mathrm{tr}\, LX \\ \text{subject to} & \mathrm{diag}(X) = \frac{1}{4} e \\ & \mathrm{rank}(X) = 1 \\ & X \succeq 0. \end{array}$$

Dropping the rank condition we obtain a problem of the form (SDP) with no inequalities, $a = \frac{1}{4}e$ and $A(X) = \mathrm{diag}(X)$.

This relaxation of max-cut is well known and studied e.g. in [9, 11, 24]. Goemans and Williamson [11] have recently shown that the optimal value of this relaxation is at most 14% above the value of the maximum cut, provided $A \geq 0$, i.e. no negative edge weights exist.

The variable X can be interpreted as being defined on the edge set of the (complete) graph. Therefore we can add further linear constraints, that are satisfied by all edge vectors representing cuts. One such class of constraints is obtained by the following trivial observation. Consider an arbitrary triangle with vertices $i < j < k$ in the graph G. Then any partition cuts either 0 or 2 of its edges. Translated into our model this leads to

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_{ij} \\ x_{ik} \\ x_{jk} \end{pmatrix} + \frac{1}{4} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \geq 0.$$

(Note that in our model an edge is cut if $x_i x_j = -1$, thus $x_{ij} = -\frac{1}{4}$.) The first constraint states that at most 2 edges can be cut. The other 3 constraints state that if one edge is cut, then there must be at least one other edge which must lie in the cut. Let us now collect these constraints for all triangles in the operator B. Then this leads to

$$B(X) - b \geq 0,$$

where $\dim(B) = 4\binom{n}{3}$ and b is a vector. Thus we get the following stronger relaxation for max-cut, first proposed in [24]

$$\begin{array}{rl} \text{maximize} & \mathrm{tr}\, LX \\ \text{subject to} & \mathrm{diag}(X) - a = 0 \\ & B(X) - b \geq 0 \\ & X \succeq 0. \end{array}$$

Dropping the semidefiniteness constraint on X we obtain an ordinary Linear Program which is again a relaxation for max-cut. This relaxation is usually called the *Metric Relaxation*, because the polyhedron

$$\{X : B(X) - b \geq 0, \mathrm{diag}(X) = a\}$$

is often referred to as the *metric polytope*, see e.g. [17]. We point out that this LP has $\binom{n}{2}$ variables and roughly $\frac{2}{3}n^3$ (very sparse) constraints. This polyhedron turns out to be highly degenerate, so that it is still considered a computational challenge to optimize an arbitrary linear function over this polytope for say $n \approx 40$. (If the graph is planar, then the metric relaxation already provides the max-cut, see e.g. [5].)

3.2 Graph Bisection

Graph bisection is similar to the max-cut problem, but here we seek a partition (S,T) of the node set V such that the two sets have prespecified cardinalities, say $|S| = k$ and $|T| = n - k$ for some given k. An important special case occurs if $k = n/2$. In this case, one looks for a partition of the node set into two sets of equal size, so as to minimize the weight of the cut (this problem is usually formulated as a minimization rather than a maximization problem.) The additional cardinality constraint $|S| = n/2$ translates into $\sum_i x_i = 0$. Thus we have

$$\text{(BS)} \qquad \begin{array}{rl} \text{minimize} & \frac{1}{4}x^T Lx \\ \text{subject to} & x_i \in \{-1, 1\} \\ & e^T x = 0. \end{array}$$

In analogy with max-cut, we obtain the following relaxation:

$$\begin{array}{rl} \text{minimize} & \text{tr } LX \\ \text{subject to} & \text{diag}(X) = a \\ & \text{tr } XJ = 0 \\ & X \succeq 0. \end{array}$$

Here $J = ee^T$ is the matrix of all ones. Note that the constraint tr $JX = 0$ is obtained by squaring the cardinality constraint: $0 = (e^T x)^2 = \text{tr } JX$. This relaxation was also studied in [10] where it was treated as a min-max eigenvalue problem using nonsmooth optimization techniques. A more theoretical investigation of this bound is given by Boppana [6].

3.3 Maximum cliques in graphs

Semidefinite programs are also used in conjunction with stable set and clique problems in graphs, see [18]. Suppose a graph G on n vertices is given by its edge set E. Define $E_{ij} := e_i e_j^t + e_j e_i^t$, where e_j is column j of the identity matrix I_n of size n. As above, $J = ee^t$ is the matrix of all ones of size n. The following semidefinite program was introduced in [18] and provides an upper bound on the largest clique in G. The optimal objective function of this program is usually denoted by $\vartheta(G)$.

$$\begin{array}{rl} \text{maximize} & \text{tr } JX \\ \text{subject to} & \text{tr } E_{ij}X = 0 \quad \forall (ij) \notin E \\ & \text{tr } X = 1 \\ & X \succeq 0. \end{array}$$

Note that the number of equality constraints is $\binom{n}{2} - |E| + 1$. Therefore, even if n is small, the dimension of the dual may be quite large.

3.4 Min-Max Eigenvalue Problems

The following min-max eigenvalue problem is studied, e.g., in [14, 21, 22]:

$$\lambda^* := \min_{y \in \Re^k} \lambda_{\max}(C + A(y)).$$

Here, $C \in \mathcal{M}_n$ and $A : \Re^k \mapsto \mathcal{M}_n$ is a linear operator. Overton [21, 22] actually considers a more general case allowing A to be nonlinear. It is well known that this problem can be reformulated as an (SDP) problem:

$$\begin{array}{rl} \text{minimize} & \lambda \\ \text{subject to} & \lambda I - C - A(y) \succeq 0 \\ & y \in \Re^k, \quad \lambda \in \Re. \end{array}$$

The dual is

$$\begin{array}{rl} \text{maximize} & \operatorname{tr} CX \\ \text{subject to} & A^T(X) = 0 \\ & \operatorname{tr} X = 1 \\ & X \succeq 0. \end{array}$$

Note that complementary slackness at optimality implies that $ZX = 0$, with $Z := \lambda I - C - A(y)$. Dual feasibility then implies that the eigenvectors for the optimal eigenvalue are found as the columns of X. If strict complementary slackness holds, i.e. $\operatorname{rank}(X) + \operatorname{rank}(Z) = n$, then $\operatorname{rank}(X)$ equals the multiplicity of the optimal eigenvalue.

Applications to other combinatorial optimization problems are described in [3].

4 An Interior-Point Method for SDP

In this section we will develop a primal-dual interior point method that solves (SDP) and (DSDP) simultaneously. The nature of this approach requires that there exists an X strictly satisfying the inequalities of the primal problem, i.e.

$$b - B(X) > 0 \quad \text{and} \quad X \succ 0.$$

Furthermore we assume without loss of generality that the equality constraints on X are linearly independent, $\operatorname{rank}(A(\cdot)) = k$. Since we will have to apply operators A and B to nonsymmetric matrices as well, we extend their definition by mapping the skew-symmetric part to zero. This implies

$$A(M) = A(M^T) \quad \text{and} \quad B(M) = B(M^T). \tag{4.1}$$

We follow the usual derivation of primal-dual interior point methods in Linear Programming and first introduce the associated *barrier problem* for (DSDP), which we call the *dual barrier problem*:

$$\text{(DBP)} \qquad \begin{array}{rl} \text{minimize} & a^T y + b^T t - \mu(\log\det Z + e^T \log t) \\ \text{subject to} & A^T(y) + B^T(t) - C = Z \\ & t \geq 0, Z \succeq 0. \end{array} \tag{4.2}$$

Here μ is a positive real number called the *barrier parameter*.

For each $\mu > 0$, there is a corresponding Lagrangian:

$$\begin{aligned} L_\mu(X,y,t,Z) = \; & a^T y + b^T t - \mu(\log\det Z + e^T \log t) + \\ & \left\langle Z + C - A^T(y) - B^T(t), X \right\rangle . \end{aligned} \tag{4.3}$$

The first-order optimality conditions for the saddle point of this Lagrangian are obtained easily using the adjoint identity for A and B:

$$\nabla_X L_\mu = Z + C - A^T(y) - B^T(t) = 0 \tag{4.4}$$

$$\nabla_y L_\mu = a - A(X) = 0 \tag{4.5}$$

$$\nabla_t L_\mu = b - B(X) - \mu t^{-1} = 0 \tag{4.6}$$

$$\nabla_Z L_\mu = X - \mu Z^{-1} = 0. \tag{4.7}$$

The strict concavity of $\log\det Z$ and $\log t_i$ implies that there exists a unique solution $(X_\mu, y_\mu, t_\mu, Z_\mu)$ to these optimality conditions. The one-parameter family $\{(X_\mu, y_\mu, t_\mu, Z_\mu) : 0 \leq \mu \leq \infty\}$ is called the *central trajectory*. Given a point (X,y,t,Z) on the central trajectory it is easy to determine its associated μ value using (4.6) and/or (4.7):

$$\mu = \frac{\operatorname{tr}(ZX)}{n} = \frac{t^T(b - B(X))}{m} = \frac{\operatorname{tr}(ZX) + t^T(b - B(X))}{n+m}. \tag{4.8}$$

We note that if the point is a feasible solution of the primal and the dual problem, $\operatorname{tr}(ZX) + t^T(b - B(X))$ is the gap between the primal and the dual objective value. We shall use (4.8) to associate μ values with quadruples (X,y,t,Z) even when these quadruples don't belong to the central trajectory.

Our interior-point algorithm is derived as follows. We start with a quadruple (X,y,t,Z) for which $X \succ 0$, $Z \succ 0$, $t > 0$, and $b - B(X) > 0$ but which is otherwise arbitrary. From this point we estimate the current μ value using (4.8) and divide it by two:

$$\mu = \frac{\operatorname{tr}(ZX) + t^T(b - B(X))}{2(n+m)}. \tag{4.9}$$

(Experience from linear programming indicates that this simple heuristic performs very well, even though it does not guarantee monotonic decrease in μ, see [28].) We next attempt to find directions $(\Delta X, \Delta y, \Delta t, \Delta Z)$ such that the new point $(X + \Delta X, y + \Delta y, t + \Delta t, Z + \Delta Z)$ lies on the central trajectory at this value of μ. However, since not all the defining equations, (4.4)–(4.7), are linear, it is not possible to solve this system directly. In fact, only (4.6) and (4.7) are nonlinear. They can be written in several equivalent forms, each form giving rise to a different linearization. We will give a more detailed discussion of the possibilities in the next section, though restricted to (4.7). For our algorithm we use the linearization of the form

$$ZX - \mu I = 0.$$

For simplicity of notation, we rewrite (4.4) to (4.7) as the function

$$F_\mu(s) = F_\mu(X,y,t,Z) := \begin{pmatrix} Z + C - A^T(y) - B^T(t) \\ a - A(X) \\ t \circ (b - B(X)) - \mu e \\ ZX - \mu I \end{pmatrix} =: \begin{pmatrix} F_d \\ F_p \\ F_{tB} \\ F_{ZX}. \end{pmatrix} \tag{4.10}$$

The solution s^* to $F_\mu(s) = 0$ satisfies the Karush-Kuhn-Tucker conditions (4.4) to (4.7) and is the optimal solution to the barrier problem. To find a direction $\Delta s = (\Delta X, \Delta y, \Delta t, \Delta Z)$ toward s^*, we use Newton's Method, which says that Δs must satisfy

$$F_\mu + \nabla F_\mu(\Delta s) = 0.$$

The direction Δs is therefore the solution of the system

$$\begin{aligned}
\Delta Z - A^T(\Delta y) - B^T(\Delta t) &= -F_d & (4.11)\\
-A(\Delta X) &= -F_p & (4.12)\\
\Delta t \circ (b - B(X)) - t \circ B(\Delta X) &= -F_{tB} & (4.13)\\
Z\Delta X + \Delta Z X &= -F_{ZX}. & (4.14)
\end{aligned}$$

This linear system can now be solved for $(\Delta X, \Delta y, \Delta t, \Delta Z)$. Indeed, first we solve (4.11) for an obviously symmetric ΔZ (in terms of Δy and Δt),

$$\Delta Z = -F_d + A^T(\Delta y) + B^T(\Delta t), \tag{4.15}$$

and then substitute this expression into (4.14) to get

$$\Delta\tilde{X} = \mu Z^{-1} - X + Z^{-1}F_d X - Z^{-1}(A^T(\Delta y) + B^T(\Delta t))X. \tag{4.16}$$

Evidently, $\Delta\tilde{X}$ is not symmetric in general. Substituting this expression for ΔX into (4.12), we get the first equation for Δy and Δt

$$O_{11}(\Delta y) + O_{12}(\Delta t) = v_1, \tag{4.17}$$

where O_{11} and O_{12} are the linear operators defined by

$$\begin{aligned}
O_{11}(\cdot) &:= A(Z^{-1}A^T(\cdot)X) & (4.18)\\
O_{12}(\cdot) &:= A(Z^{-1}B^T(\cdot)X) & (4.19)
\end{aligned}$$

and v_1 is the vector

$$v_1 := \mu A(Z^{-1}) - a + A(Z^{-1}F_d X). \tag{4.20}$$

Finally we substitute (4.16) into (4.13) and get

$$O_{21}(\Delta y) + O_{22}(\Delta t) = v_2, \tag{4.21}$$

where O_{21} and O_{22} are the linear operators defined by

$$\begin{aligned}
O_{21}(\cdot) &:= B(Z^{-1}A^T(\cdot)X) & (4.22)\\
O_{22}(\cdot) &:= (b - B(X)) \circ t^{-1} \circ (\cdot) + B(Z^{-1}B^T(\cdot)X) & (4.23)
\end{aligned}$$

and v_2 is the vector

$$v_2 := \mu t^{-1} - b + \mu B(Z^{-1}) + B(Z^{-1}F_d X). \tag{4.24}$$

Because of (4.1) operators O_{11} and O_{22} are self adjoint and operator O_{12} is the adjoint operator to O_{21}. Equations (4.17) and (4.21) form a symmetric linear system for Δy and Δt. In fact, this system is even positive definite. To show this, we define a new operator O that maps $\mathcal{M}_n$ into $\Re^{k+m}$:

$$O(X) = \begin{pmatrix} A(X) \\ B(X) \end{pmatrix}.$$

The adjoint operator is given by the adjoint identity

$$\left\langle O(X), \begin{pmatrix} y \\ t \end{pmatrix} \right\rangle = \left\langle X, A^T(y) + B^T(t) \right\rangle.$$

The system can now be written as

$$O(Z^{-1}O^T\begin{pmatrix} \Delta y \\ \Delta t \end{pmatrix}X) + \begin{pmatrix} 0 \\ (b - B(X)) \circ t^{-1} \circ (\Delta t) \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}. \tag{4.25}$$

Observe that the second summand adds positive coefficients to the main diagonal entries corresponding to the inequality constraints. In fact this increment on the main diagonal is the only difference between inequality and equality constraints. The second summand clearly forms a positive semidefinite operator on $\binom{\Delta y}{\Delta t}$. It is positive definite for all vectors having at least one component $\Delta t_i \neq 0$. For the first summand we get

$$\begin{aligned} & \left\langle O(Z^{-1}O^T(v)X), v \right\rangle & (4.26) \\ = \; & \left\langle Z^{-1}O^T(v)X, O^T(v) \right\rangle & (4.27) \\ = \; & \operatorname{tr}\left(Z^{-\frac{1}{2}}O^T(v)X^{\frac{1}{2}}X^{\frac{1}{2}}O^T(v)Z^{-\frac{1}{2}}\right) & (4.28) \\ = \; & \left\langle Z^{-\frac{1}{2}}O^T(v)X^{\frac{1}{2}}, Z^{-\frac{1}{2}}O^T(v)X^{\frac{1}{2}} \right\rangle \geq 0. & (4.29) \end{aligned}$$

Since X and Z^{-1} are positive definite and the equality constraints are linearly independent, equality is possible if and only if $v = 0$ or at least one $\Delta t_i \neq 0$. It follows that the system is indeed positive definite. It can be solved efficiently in $O((k+m)^3)$. Observe that an equivalent representation of operator $O(X)$ is given by $(\operatorname{tr}(A_1 X), \ldots \operatorname{tr}(A_{k+m}X))^T$ with the A_i being appropriately chosen symmetric matrices. Then the ij-th element of the matrix describing $O(Z^{-1}O^T(\cdot)X)$ reads $\operatorname{tr}(A_i Z^{-1} A_j X)$.

The solution yields a quadruple with $\Delta\tilde{X}$ not necessarily symmetric. This is mended by using the symmetric part of $\Delta\tilde{X}$ only:

$$\Delta X = \frac{\Delta\tilde{X}^T + \Delta\tilde{X}}{2}. \tag{4.30}$$

To summarize, we solve for the quadruple $\Delta s = (\Delta X, \Delta y, \Delta t, \Delta Z)$ by first solving (4.17) and (4.21) for Δy and Δt and then substituting this into (4.15) to solve for ΔZ and finally substituting that into (4.14) to solve for $\Delta\tilde{X}$ of which we take the symmetric part only. In section 5 we will show that this indeed yields a descent direction.

Having determined the desired quadruple, $(\Delta X, \Delta y, \Delta t, \Delta Z)$, of directions, we would step to the new quadruple $(X+\Delta X, y+\Delta y, t+\Delta t, Z+\Delta Z)$ except that it might violate the nonnegativity of t and $b - B(X)$ and the positive definiteness property required of the two matrices. Hence, we perform a line search to find constants α_p and α_d such that $t + \alpha_d \Delta t$ and $b - B(X + \alpha_p \Delta X)$ are strictly positive and $X + \alpha_p \Delta X$ and $Z + \alpha_d \Delta Z$ are positive definite. Given α_p and α_d, we step to the new point

$$\begin{gathered} X + \alpha_p \Delta X \\ y + \alpha_d \Delta y \\ t + \alpha_d \Delta t \\ Z + \alpha_d \Delta Z. \end{gathered}$$

We update μ using (4.9) and repeat. The algorithm continues until the current quadruple (X, y, t, Z) satisfies primal feasibility, dual feasibility and the duality gap is sufficiently small. This completes the description of our interior-point algorithm.

5 Descent Direction

We have found a modified Newton direction for the optimality conditions. To shed some light on the quality of this direction $(\Delta X, \Delta y, \Delta t, \Delta Z)$ we will prove that it forms a descent direction with respect to an appropriately defined merit function. We measure the progress of the algorithm using the following merit function. (This type of merit function was also used in [1].)

$$\begin{aligned} f_\mu(X, y, t, Z) \quad = \quad & \langle Z, X \rangle - \mu \log \det(XZ) + \\ & t^T(b - B(X)) - \mu e^T \log(t \circ (b - B(X))) + \\ & \frac{1}{2}||F_p||^2 + \frac{1}{2}||F_d||^2 \end{aligned} \tag{5.1}$$

For feasible points the merit function is the difference between the objective values of the dual and primal barrier functions. Therefore f_μ is convex over the set of feasible points. As the minimum of $(x - \mu \log x)$ for $x > 0$ is attained at $x = \mu$, function f_μ is bounded below by $(n+m)\mu(1 - \log \mu)$. We note that $F_\mu(s) = 0 \iff f_\mu = (n+m)\mu(1 - \log \mu)$. f_μ is continuously differentiable on the interior ($X \succ 0$, $t > 0$, $Z \succ 0$) and grows towards infinity on the boundary. In the following Lemma we prove that Δs (as defined on page 9) is a descent direction for f_μ.

Lemma 1 *The directional derivative of f_μ in the direction Δs satisfies*

$$\langle \nabla_s f_\mu, \Delta s \rangle \leq 0 \tag{5.2}$$

with equality holding if and only if $F_\mu(s) = 0$.

Proof. We have to prove that

$$\langle \nabla_X f_\mu, \Delta X\rangle + \langle \nabla_y f_\mu, \Delta y\rangle + \langle \nabla_t f_\mu, \Delta t\rangle + \langle \nabla_Z f_\mu, \Delta Z\rangle \le 0$$

The partial derivatives of f_μ are

$$\begin{aligned}
\nabla_X f_\mu &= F_p^T \nabla_X F_p + F_d^T \nabla_X F_d + Z - \mu X^{-1} - B^T(t) - \mu B^T((b - B(X))^{-1})\\
\nabla_y f_\mu &= F_p^T \nabla_y F_p + F_d^T \nabla_y F_d\\
\nabla_t f_\mu &= F_p^T \nabla_t F_p + F_d^T \nabla_t F_d + b - B(X) - \mu t^{-1}\\
\nabla_Z f_\mu &= F_p^T \nabla_Z F_p + F_d^T \nabla_Z F_d + X - \mu Z^{-1}.
\end{aligned}$$

Summing the directional derivative over all partial derivatives of F_p we get

$$\begin{aligned}
&\left\langle F_p^T \nabla_X F_p, \Delta X\right\rangle + \left\langle F_p^T \nabla_y F_p, \Delta y\right\rangle + \left\langle F_p^T \nabla_t F_p, \Delta t\right\rangle + \left\langle F_p^T \nabla_Z F_p, \Delta Z\right\rangle =\\
&\quad= \langle F_p, -A(\Delta X)\rangle\\
&\quad= \left\langle F_p, -A(\Delta \tilde{X})\right\rangle\\
&\quad= -||F_p||^2,
\end{aligned}$$

where the third line follows from $A(M) = A(M^T)$ and the fourth from (4.12). Analogously we get

$$\left\langle F_d^T \nabla_X F_d, \Delta X\right\rangle + \left\langle F_d^T \nabla_y F_d, \Delta y\right\rangle + \left\langle F_d^T \nabla_t F_d, \Delta t\right\rangle + \left\langle F_d^T \nabla_Z F_d, \Delta Z\right\rangle = -||F_d||^2.$$

For the next step we observe that for a symmetric matrix S, $\langle S, \Delta X\rangle = \left\langle S, \Delta\tilde{X}\right\rangle$ since the skew–symmetric part of $\Delta\tilde{X}$ is orthogonal to symmetric matrices.

$$\begin{aligned}
&\left\langle Z - \mu X^{-1}, \Delta X\right\rangle + \left\langle X - \mu Z^{-1}, \Delta Z\right\rangle =\\
&\quad= \left\langle Z - \mu X^{-1}, \Delta\tilde{X}\right\rangle + \left\langle X - \mu Z^{-1}, \Delta Z\right\rangle\\
&\quad= \mathrm{tr}(I - \mu X^{-1}Z^{-1})(Z\Delta X' + \Delta Z X)\\
&\quad= -\mathrm{tr}(I - \mu X^{-1}Z^{-1})(ZX - \mu I)\\
&\quad= -\mathrm{tr}(ZX - 2\mu I + \mu^2 (ZX)^{-1})\\
&\quad= -\sum_{i=1}^n \lambda_i(ZX)\left(1 - \mu\lambda_i^{-1}(ZX)\right)^2.
\end{aligned}$$

As Z and X are positive definite, all eigenvalues of ZX are strictly positive. The last expression is less or equal to zero; and it equals zero if and only if $ZX = \mu I$.

In quite the same manner we get

$$\begin{aligned}
&(b - B(X) - \mu t^{-1})^T \Delta t + (t - \mu(b - B(X))^{-1})^T B(\Delta X) =\\
&\quad= -(t \circ (b - B(X)))^T (e - \mu t^{-1} \circ (b - B(X))^{-1})^2.
\end{aligned}$$

Again $t \circ (b - B(X))$ is strictly positive, so the expression is less or equal to zero; and it equals zero if and only if $t \circ (b - B(X)) = \mu e$.

Summing up we have

$$\begin{aligned} \frac{\partial f_\mu}{\partial s}\Delta s &= -||F_p||^2 - ||F_d||^2 - \sum_{i=1}^{n} \lambda_i(ZX)\left(1 - \mu\lambda_i^{-1}(ZX)\right)^2 \\ &\quad -(t \circ (b - B(X)))^T(e - \mu t^{-1} \circ (b - B(X))^{-1})^2 \leq 0 \end{aligned} \tag{5.3}$$

with equality holding if and only if $F_\mu(s) = 0$ holds. □

Inaccurate line search with respect to the merit function and a feasible starting point is sufficient to guarantee convergence, an elementary proof can be found in [13]. This need not be true for arbitrarily chosen infeasible starting points. Some computational evidence for this claim is contained in [4].

Very strong results on the rate of convergence can be obtained by tightly restricting the choice of μ and the stepsize. In an independent paper Kojima, Shindoh, and Hara [16] propose the very same algorithmical framework for the semidefinite complementarity problem. Within this framework they give several variants of algorithms leading to polynomial convergence. The first of these methods is very close to our setting and we will give some details. A complementarity problem asks for a point $(X, Z) \in \mathcal{M}_n^2$ which lies within a given $n(n+1)/2$ dimensional affine subspace $\mathcal{F}$ of $\mathcal{M}_n^2$ and satisfies the complementarity condition $\mathrm{tr}(ZX) = 0$, i.e. find a pair $(Z, X) \in \mathcal{M}_n^2$ such that

$$\text{(LCP)} \qquad (X, Z) \in \mathcal{F},\ X \succeq 0,\ Z \succeq 0,\ \text{and } \mathrm{tr}(XZ) = 0.$$

Let $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of XZ. By

$$\mathcal{N}(\gamma) = \left\{(X, Z) \in \mathcal{F}, X \succ 0, Z \succ 0 \,\middle|\, \left(\sum_{i=1}^{n}(\lambda_i - \mu)^2\right)^{\frac{1}{2}} \leq \gamma\mu \ \text{ with } \ \mu = \frac{\mathrm{tr}(ZX)}{n}\right\}$$

a horn neighborhood of the central trajectory is introduced. The width of this neighborhood is controlled by γ. For $\gamma \in (0, 0.1]$ this neighborhood has the convenient property (Theorem 4.1 in [16]) that by reducing μ by a factor of $(1 - \frac{\gamma}{\sqrt{n}})$ the "Newton step" $(X + \Delta X, Z + \Delta Z)$ is again in $\mathcal{N}(\gamma)$. Furthermore the new μ–value of the new point is smaller than the old value of μ by at least a factor of $(1 - \frac{\gamma}{2\sqrt{n}})$. So if $(X^0, Z^0) \in \mathcal{N}(\gamma)$ the algorithm yields a sequence $(X^j, Z^j) \in \mathcal{N}(\gamma)$ such that for every $j = 0, 1, \ldots$

$$\mathrm{tr}(X^j, Z^j) \leq \left(1 - \frac{\gamma}{2\sqrt{n}}\right)^j \mathrm{tr}(X^0 Z^0).$$

Thus, if we require $\mathrm{tr}(X^j Z^j) < \varepsilon$ for a given $\varepsilon > 0$ then the algorithm will stop in $O(\sqrt{n}\log\frac{\mathrm{tr}(X^0 Z^0)}{\varepsilon})$ iterations. In practical applications the strict requirements for the choice of μ have to be violated for the benefit of efficiency.

We will now give a short discussion of the linearization stemming from equivalent formulations of the condition $ZX - \mu I = 0$. For ease of notation we will restrict our investigation to the equality constraints and drop the inequality operator B. We have the following possibilities.

$$\mu I - Z^{1/2} X Z^{1/2} = 0 \tag{5.4}$$

$$\mu I - X^{1/2} Z X^{1/2} = 0 \tag{5.5}$$

$$\mu Z^{-1} - X = 0 \tag{5.6}$$

$$\mu X^{-1} - Z = 0 \tag{5.7}$$

$$ZX - \mu I = 0 \tag{5.8}$$

$$XZ - \mu I = 0 \tag{5.9}$$

$$ZX + XZ - 2\mu I = 0. \tag{5.10}$$

The first two linearizations are very popular in linear programming, in which case X and Z are diagonal matrices. The two forms are then the same and do not really involve matrix square roots. However, in the semidefinite programming case they do involve matrix square roots, which makes them computationally less attractive.

The linearization of the third form (5.6) leads to Jarre's algorithm [14]. In this case the defining equation for Δy reads,

$$A(Z^{-1} A^T(\Delta y) Z^{-1}) = A(Z^{-1} + Z^{-1} F_d Z^{-1}) - \frac{a}{\mu}.$$

Note that no information about the primal variable is contained in this equation. This is a pure dual approach. The resulting direction for the primal variable is poor. This was confirmed by practical experiments.

Analogously, the linearization of (5.7) leads to

$$A(XA^T(\Delta y)X) = A(\mu X + XCX - XA^T(y)X) - \mu F_p$$

This formulation is used in [25]. This time the step is mainly based on the primal variable. This linearization can be considered as a pure primal approach.

The linearization of (5.8) is the choice of this paper. It is easy to see that (5.9) results in the same step. Indeed, $\Delta \tilde{X}$ of the latter linearization is just the transpose of $\Delta \tilde{X}$ of our algorithm. In both cases the direction found contains the information of the primal and dual variables at an equal degree. Both linearizations are especially well suited for Mehrotra's LP predictor–corrector method as described in [7].

The latter two statements also apply to the linearization of (5.10) as discussed in [4]. An advantage of this linearization is that it preserves symmetry. Furthermore Alizadeh, Haeberly, and Overton present some evidence of good numerical properties. However, as the system is more difficult to solve, computation time per iteration can be considerably higher.

6 Computational Results

6.1 Interior Point Approach for Max–Cut

In this section we show how the Max–Cut relaxation given in Section 3 can be implemented efficiently within this framework. We first look at the relaxation without the triangle inequalities. The resulting program is very simple and can be solved for quite large n very fast, see Table 1.

The cost matrix is given by the Laplacian L of the weighted adjacency matrix of the graph. The SDP forming the relaxation reads

$$\begin{array}{rl} \text{maximize} & \operatorname{tr} LX \\ \text{subject to} & \operatorname{diag}(X) = \frac{1}{4}e =: a \\ & X \succeq 0. \end{array} \tag{6.1}$$

Since Diag is the adjoint of diag, the dual SDP reads

$$\begin{array}{rl} \text{minimize} & a^T y \\ \text{subject to} & \operatorname{Diag}(y) - L = Z \\ & Z \succeq 0. \end{array} \tag{6.2}$$

The starting point

$$X := \operatorname{Diag}(a) \tag{6.3}$$

$$y := 1.1 \cdot \operatorname{abs}(L)e \tag{6.4}$$

$$Z := \operatorname{Diag}(y) - L \tag{6.5}$$

is feasible and in the interior. If we look at the definig equation for Δy we get

$$O_{11}(\Delta y) = \operatorname{diag}(Z^{-1}\operatorname{Diag}(\Delta y)X) = (Z^{-1} \circ X)\Delta y \tag{6.6}$$

$$v_1 = \mu \operatorname{diag}(Z^{-1}) - a. \tag{6.7}$$

This means that at each iteration we solve the system by computing

$$\Delta y := (Z^{-1} \circ X)^{-1}(\mu \operatorname{diag}(Z^{-1}) - a) \tag{6.8}$$

$$\Delta Z := \operatorname{Diag}(\Delta y) \tag{6.9}$$

$$\Delta \tilde{X} := \mu Z^{-1} - X - Z^{-1}\Delta Z X \tag{6.10}$$

$$\Delta X := (\Delta \tilde{X}^T + \Delta \tilde{X})/2. \tag{6.11}$$

To further emphasize the simplicity and ease of implementation of this approach, we include in the next section a MATLAB function that solves this relaxation. For problems of sizes up to $n \approx 200$ this is a very efficient code and the reader is encouraged to test it.

The numerical results of Table 1 were computed on a 486 66 Mhz, PC using a C encoded version of this algorithm. We point out that the number of iterations does not seem to depend significantly on n. In all tables hh gives the number of hours, mm the number of

size n	iterations	hh:mm:ss
100	14	48
150	12	2:30
200	12	6:30
250	13	14:24
300	14	30:24
400	14	1:24:36
500	14	2:13.18

Table 1: SDP relaxation for max-cut, computed on a 486, 66 Mhz PC

minutes, ss the seconds. The instances were generated as random unweighted graphs with edge probability $p = 0.5$. We set the stopping condition to 6 digits of accuracy, as in the MATLAB routine from the appendix.

As a second experiment we looked at the max-cut relaxation formed by the intersection of all the $4\binom{n}{3}$ triangle inequalities with the semidefiniteness constraints, see Section 3.1. When we included triangle constraints it turned out to be favorable to use Mehrotra's predictor – corrector approach, which is closely investigated in [7]. In this case the system is solved twice with the same factorization but with different right hand sides. First the right hand side is chosen as to minimize the objective, i.e $\mu = 0$ is used. The second step computes a good centering direction for the predicted point, using an estimate of μ from the previous iteration. The triangle inequalities were included successively according to their amount of violation, for a detailed description of the algorithm the reader is referred to [13]. Some results for random graphs are given in Table 2. Most of the time the best cut was already found by the relaxation without triangles and could not be improved later on. It was constructed from the rounded rows of X giving an initial cut. We then applied a local improvement strategy to this cut by considering swapping any single node from one side to the other side of the cut. If any such swap improved the cut, we chose the swap with highest gain. We stopped if no further improvement could be obtained this way.

We note that these semidefinite programs have n equality and $4\binom{n}{3}$ inequality constraints in the dual program. The final solution satisfies all these constraints. The number of iterations necessary to obtain these results is astonishingly small, see the last column of Table 2. To be explicit, the solution of problems with $n = 100$ satisfies $4\binom{100}{3} = 646,800$ inequality constraints. Our approach does not seem to depend on structural properties of the underlying graph. It significantly extends purely polyhedral techniques for max-cut. Further and more substantial computational results with this approach, applied to max-cut, can be found in [13].

6.2 The ϑ–Function

In this section we give computational results for the ϑ–function as defined in Section 3.3. For the convenience of the reader we give the corresponding primal dual pair of semidefinite

random graphs, edge-weights $\in \{-10, \ldots, 10\}$				
size n	cut	upper bnd (%)	hh:mm:ss	iterations
30	330	330.0001 (0.0)	54	59
40	619	619.0000 (0.0)	2:02	60
50	708	708.0001 (0.0)	1:04:22	166
70	1215	1248.7142 (2.8)	4:42:21	220
100	2440	2540.2528 (4.1)	37:56:05	266
random unweighted graphs, edge probability 0.5				
size n	cut	upper bnd (%)	hh:mm:ss	iterations
30	143	143.0000 (0.0)	56	62
40	245	245.6864 (0.3)	59:19	247
50	374	375.3689 (0.4)	1:54:18	250
70	723	726.0399 (0.4)	5:57:22	257

Table 2: Solutions to Max–Cut relaxation with triangle inequalities. The number of nodes in the graph is n, *cut* refers to the best cut found, *upper bnd* is the value of the relaxation, % is the percentage of the gap between relaxation and best cut found. The next column gives the computation times on a DEC-station 5000/240. Note that the number of iterations is almost independent of n if the gap is not closed.

programs,

$$
\begin{array}{ll}
\text{maximize} & \operatorname{tr} JX \\
\text{subject to} & \operatorname{tr} E_{ij}X = 0 \quad \forall (ij) \notin E \\
& \operatorname{tr} X = 1 \\
& X \succeq 0.
\end{array}
\qquad
\begin{array}{ll}
\text{minimize} & \lambda \\
\text{subject to} & Z = \lambda I - J + \sum_{(ij)\notin E} E_{ij} y_{ij} \\
& Z \succeq 0, \; y_{ij} \in \Re \quad \forall (ij) \notin E,
\end{array}
$$

As starting point we used the feasible interior point

$$
\begin{aligned}
X &:= \frac{1}{n} I \\
\lambda &:= 1.1 \cdot n \\
y_{ij} &:= 0 \quad \forall (ij) \notin E \\
Z &:= \lambda I - J.
\end{aligned}
$$

We tested the code for random graphs of the class $G_{n;p}$, i.e. graphs with n vertices and each edge (ij) being in the set E with probability p. In order for the reader to be able to reproduce the graphs, we include the source code, written in C++, that we used to generate these pseudo-random graphs. For generating the adjacency matrix A, we called the following random number generator with parameters n, *seed*, and *prob*.

```
double r=(4.*double(seed)+1.)/16384./16384.;
for(int i=0;i<n;i++){
    for(int j=i+1;j<n;j++) {
        r=fmod(r*41475557.,1.);
        if (r<1.-prob) A(i,j)=1.;
        else A(i,j)=0.;
    }
}
```

The results for several choices of *seed*, *prob*, and n are displayed in Table 3. The code employed the predictor–corrector approach using separate primal and dual stepsizes. The program was written in C++ and executed on a DEC-station 5000/240. The number of primal constraints or dual variables is the number of *nonedges* plus one. Computing times strongly depend on the latter number but are almost independent of n as long as *nonedges* $\gg n$. Again, the number of iterations seems to be quite independent of both n and the number of constraints.

6.3 Min-Max Eigenvalue Problems

Consider the min-max eigenvalue problem:

$$
\begin{array}{ll}
\text{minimize} & \lambda_{max}(C - \operatorname{Diag}(v)) \\
\text{subject to} & e^T v = 0.
\end{array}
\tag{6.12}
$$

seed	prob	size n	nonedges	ϑ	hh.mm.ss	iterations
1	0.5	50	593	7.9233	7:00	9
2	0.8	50	237	16.0012	42	14
3	0.9	50	124	21.0910	10	11
4	0.8	100	1018	21.9283	44:05	10
5	0.9	100	511	32.4967	5:29	10
6	0.9	150	1130	41.6814	1:04:23	10
7	0.95	150	574	56.4224	10:56	11
8	0.95	200	972	70.5405	44:46	10
9	0.97	200	585	85.0430	17:39	12
10	0.97	250	915	98.5259	51:10	11
11	0.98	250	604	114.6005	25:43	11
12	0.97	300	1310	112.4511	2:30:46	11

Table 3: The ϑ–function for random graphs.

The objective function in (6.12) is not differentiable when the multiplicity of the largest eigenvalue exceeds one. In fact, a singleton eigenvalue characterizes differentiability. Since the largest eigenvalue is a convex function, subgradient approaches can be used to solve (6.12) (see, e.g., [8]). More recently, it has been shown that Newton-based algorithms with local quadratic convergence exist (see, e.g., [23]) but the local convergence depends on correctly identifying the multiplicity of the largest eigenvalue. We present computational experiments showing that our interior-point method is indeed robust in the presence of high multiplicity.

By minimizing λ over the set $\left\{\lambda I \succeq C - \mathrm{Diag}(v), e^T v = 0\right\}$ and substituting $y := v + \lambda e$ we see that (6.12) is indeed a special case (with $a = e/n$) of the more general problem

$$\begin{array}{rl} \text{minimize} & a^T y \\ \text{subject to} & \mathrm{Diag}(y) - C \succeq 0. \end{array} \tag{6.13}$$

The MATLAB code given in the appendix also applies to this problem. To test the code on problem instances that exhibit given multiplicity at the optimal solution, we developed a special generator which we now describe.

To generate positive semidefinite programs we generate the elements of a uniformly on some interval of the nonnegative half-line (the primal problem is clearly infeasible if any component of a is negative). For the experiments described below, we used $a = e/n$.

Given a, we generate C as follows. First, we generate an $n \times m$ random matrix A and apply row scaling to make all squared row norms equal to the corresponding elements of a. That is,

$$\mathrm{diag}(AA^T) = a. \tag{6.14}$$

We denote the columns of A by $v_1, \ldots, v_m$. We then construct $n - m$ additional random n-vectors $v_{m+1}, \ldots, v_n$ and apply Gram-Schmidt orthonormalization to $v_1, \ldots, v_n$ to produce

n	m	k	Time (hh:mm:ss.s) BT	IP
10	1	1	0.0	0.0
20	1	1	0.1	0.3
30	1	1	0.3	1.0
50	1	1	1.2	4.2
100	1	1	12.6	35.6
200	1	1	1:56.9	6:17.4

Table 4: Statistics for problems with multiplicity 1. BT refers to the Bundle Trust method and IP refers to our Interior-Point method.

n	m	k	Time (hh:mm:ss.s) BT	IP	Comments
20	3	3	0.4	0.4	
20	5	5	2.8	0.3	
20	5	12	2.8	0.3	
20	8	8	2.7	0.3	
20	12	12	3.6	0.3	
30	3	3	1.5	1.0	
30	3	6	1.5	0.9	
30	6	6	18.2	0.8	
30	10	10	4.0	1.0	Num. trouble in BT
50	5	5	>20:00.0	4.3	5 sig. fig. in BT
100	3	3	18.7	33.9	
100	6	6	>15:00.0	36.9	5 sig. fig. in BT
500	50	50	-	2:02:47.0	No attempt at BT

Table 5: Statistics for problems with higher built-in multiplicity. BT refers to the Bundle Trust method and IP refers to our Interior-Point method.

an $n \times n$ orthogonal matrix Q whose first m columns span the same space as $v_1, \ldots, v_m$. Finally, we set

$$C = Q\Lambda Q^T, \tag{6.15}$$

where Λ is a diagonal matrix whose first $k \geq m$ entries are all set to $\lambda_{\max}$ (which is a constant that can be chosen arbitrarily – we used 5) and the remaining diagonal entries are generated uniformly on some interval that is strictly smaller than $\lambda_{\max}$. For such a matrix C, we claim that

$$\begin{aligned} X &= AA^T \\ y &= \lambda_{max} e \\ Z &= \mathrm{Diag}(y) - C \end{aligned}$$

is optimal. Indeed, it follows from (6.14) that X is feasible for the primal and it is clear from (6.15) that (y, Z) is feasible for the dual. Finally, optimality follows from the absence of a duality gap:

$$\mathrm{tr}(ZX) = \mathrm{tr}\{(\lambda_{max} I - C)AA^T\} = 0$$

The last equality follows from the fact that the columns of A are eigenvectors of C associated with the maximal eigenvalue.

Table 4 shows the comparison between the bundle trust method, see [26], and our interior-point method when the optimal eigenvalue is a singleton ($k = 1$). For these problems, the bundle trust method is three to four times faster (computing times are given for a Silicon Graphics Indigo workstation R 4000). However, this situation never arises in practice. Indeed, for $k = 1$ in our construction above, we see that we are requiring the vector of all ones to be a maximal eigenvector of C. This is clearly an unlikely event in real applications.

Table 5 shows comparisons for higher multiplicities. Here the results look much better for the interior-point method. In fact, it is clear that the bundle trust method completely breaks down rather rapidly as the multiplicity increases.

7 Matlab Implementation

The following MATLAB function solves the semidefinite programming problem described above with $\mathcal{A}(X) = \mathrm{diag}(X)$ and no inequality constraints. We include it to further emphasize the simplicity of our interior point approach. We assume that MATLAB's version 4 is available in the positive definiteness test, using MATLAB's built in function `chol`. Other than that the program could also be run under older versions of MATLAB. We have somewhat arbitrarily set the stopping condition to 6 digits of accuracy. Thus, after successful termination, we return primal and dual feasible solutions, whose objective values agree to approximately 6 digits.

In all our practical experiments it proved unnecessary to check the decrease of the merit function. Therefore this feature is not included in the present algorithm. We also mention that in case of 'large steps', i.e. $\alpha_p + \alpha_d \geq 1.8$ we select the new value of μ to be $\mathrm{tr}XZ/(4n)$, and otherwise $\mathrm{tr}XZ/(2n)$.

```
function [phi, X, y] = psd_ip( L);
% solves: max trace(LX) s.t. X psd, diag(X) = b;  b = ones(n,1)/4
%         min b'y       s.t. Diag(y) - L psd, y unconstrained,
% input:  L ... symmetric matrix
% output: phi ... optimal value of primal, phi =trace(LX)
%         X   ... optimal primal matrix
%         y   ... optimal dual vector
% call:   [phi, X, y] = psd_ip( L);

digits = 6;                        % 6 significant digits of phi
[n, n1] = size( L);                % problem size
b = ones( n,1 ) / 4;               % any b>0 works just as well
X = diag( b);                      % initial primal matrix is pos. def.
y = sum( abs( L))' * 1.1;          % initial y is chosen so that
Z = diag( y) - L;                  % initial dual slack Z is pos. def.
phi = b'*y;                        % initial dual
psi = L(:)' * X( :);               % and primal costs
mu = Z( :)' * X( :)/( 2*n);        % initial complementarity
iter=0;                            % iteration count

disp(['     iter    alphap    alphad      gap     lower      upper']);

while phi-psi > max([1,abs(phi)]) * 10^(-digits)

      iter = iter + 1;             % start a new iteration
      Zi = inv( Z);                % inv(Z) is needed explicitly
      Zi = (Zi + Zi')/2;
      dy =  (Zi.*X) \ (mu * diag(Zi) - b);      % solve for dy
      dX = - Zi * diag( dy) * X + mu * Zi - X;  % back substitute for dX
      dX = ( dX + dX')/2;          % symmetrize

% line search on primal
      alphap = 1;                  % initial steplength
      [dummy,posdef] = chol( X + alphap * dX );  % test if pos.def
      while posdef > 0,
              alphap = alphap * .8;
              [dummy,posdef] = chol( X + alphap * dX );
              end;
      if alphap < 1, alphap = alphap * .95; end; % stay away from boundary
% line search on dual; dZ is handled implicitly: dZ = diag( dy);
      alphad = 1;
      [dummy,posdef] = chol( Z + alphad * diag(dy) );
      while posdef > 0;
              alphad = alphad * .8;
              [dummy,posdef] = chol( Z + alphad * diag(dy) );
              end;
      if alphad < 1, alphad = alphad * .95; end;
% update
      X = X + alphap * dX;
      y = y + alphad * dy;
      Z = Z + alphad * diag(dy);
      mu = X( :)' * Z( :) / (2*n);
      if alphap + alphad > 1.8, mu = mu/2; end; % speed up for long steps
      phi = b' * y; psi = L( :)' * X( :);
% display current iteration
        disp([ iter alphap alphad (phi-psi) psi  phi ]);

        end;            % end of main loop
```

Acknowledgement. We thank the anonymous referees and the associate editor for their constructive comments on earlier versions of this paper. Their critical remarks helped to improve this paper.

References

[1] K.M. ANSTREICHER, J.-P. VIAL. *On the Convergence of an Infeasible Primal-Dual Interior-Point Method for Convex Programming.* Technical Report 1993.16, Départment d'économie commerciale et industrielle, Université de Genève, Switzerland, 1993.

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

[3] F. ALIZADEH. Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization. to appear in *SIAM J. Optimization.*

[4] F. ALIZADEH, J.A. HAEBERLY, and M.L. OVERTON. Primal-dual interior point methods for semidefinite programming. Technical Report, 1994.

[5] F. BARAHONA. The max-cut problem in graphs not contractible to K_5. *Operations Research Letters*, 2:107–111, 1983.

[6] R.B. BOPPANA. Eigenvalues and graph bisection: an average case analysis. In *Proceedings of the 28th Annual Symposium on Computer Science*, 280–285, IEEE, 1987.

[7] T.J. CARPENTER, I.J. LUSTIG, J.M. MULVEY, and D.F.SHANNO. Higher-order predictor-corrector interior point methods with application to quadratic objectives. *SIAM J. Optimization*, 3:696–725, 1993.

[8] J. CULLUM, W.E. DONATH, and P.WOLFE. The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices. *Mathematical Programming Study*, 3:35–55, 1975.

[9] C. DELORME and S. POLJAK. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62:557–574, 1993.

[10] J. FALKNER, F. RENDL, and H. WOLKOWICZ. A computational study of graph partitioning. Technical Report CORR, Department of Combinatorics and Optimization, Waterloo, Ont, 1992. to appear in *Math. Programming.*

[11] M.X. GOEMANS and D.P. WILLIAMSON. .878-approximation algorithm for max–cut and max–2sat. Technical Report, MIT, 1993.

[12] F. O. HADLOCK. Finding a maximum cut in a planar graph in polynomial time. *SIAM J. Computing*, 4:221–225, 1975.

[13] C. HELMBERG. An interior point method for semidefinite programming and max–cut bounds. PhD Thesis, Graz University of Technology, Oct. 1994.

[14] F. JARRE. An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices. *SIAM J. Control and Optimization*, 31:1360-1377, 1993.

[15] F. JARRE. Interior-point methods via self-concordance or relative Lipschitz condition. Habilitationsschrift, Würzburg University, March 1994.

[16] M. KOJIMA, S. SHINDOH, and S. HARA. Interior–point methods for the monotone linear complementarity problem in symmetric matrices. Technical Report B–282, Department of Information Sciences, Tokyo Institute of Technology, Apr. 1994.

[17] M. LAURENT and S. POLJAK. The Metric Polytope. Proceedings of IPCO 2, (E.Balas, G.Cornuejols and R.Kannan eds.), 1992, 274–286.

[18] L. LOVÁSZ. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25:1–7, 1979.

[19] B. MOHAR and S. POLJAK. Eigenvalues in combinatorial optimization, in "Combinatorial and graph-theoretic problems in linear algebra", R. Brualdi, S. Friedland, V. Klee, Eds., IMA Volumes in Mathematics and Its Applications, Vol. 50, Springer-Verlag, 1993.

[20] Y. E. NESTEROV and A. S. NEMIROVSKII. *Interior Point Polynomial Methods in Convex Programming : Theory and Algorithms.* SIAM Publications. SIAM, Philadelphia, USA, 1993.

[21] M.L. OVERTON. On minimizing the maximum eigenvalue of a symmetric matrix. *SIAM J. Matrix Analysis and Applications*, 9:256–268, 1988.

[22] M.L. OVERTON. Large-scale optimization of eigenvalues. *SIAM J. Optimization*, 2:88–120, 1992.

[23] M.L. OVERTON and R.S. WOMERSLEY. Second derivatives for optimizing eigenvalues of symmetric matrices. Technical Report 627, Computer Science Department, NYU, 1993. to appear in Siam J. Matrix Analysis and Applications.

[24] S. POLJAK and F. RENDL. Nonpolyhedral relaxations of graph bisection problems. Technical Report, University of Technology Graz, 1992. To appear in *SIAM J. Optimization.*

[25] F. RENDL, R. J. VANDERBEI, and H. WOLKOWICZ. Max-min eigenvalue problems, primal-dual interior point algorithms, and trust region subproblems. Technical Report CORR 93-30, Department of Combinatorics and Optimization, Waterloo, Ont. To appear in Proceedings of NATO conference on Nonlinear Programming, Il-Ciocco, Italy, 1993.

[26] H. SCHRAMM and J. ZOWE. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM J. Optimization*, 2:121–152, 1992.

[27] L. VANDENBERGHE and S. BOYD. Primal-dual potential reduction method for problems involving matrix inequalities. Technical report, Electrical Engineering Department, Stanford University, Stanford, CA 94305, 1993. To appear in *Mathematical Programming, Series B.*

[28] R.J. VANDERBEI and T.J. CARPENTER. Symmetric indefinite systems for interior-point methods. *Mathematical Programming*, 58:1–32, 1993.

[29] J.-P. VIAL. *Computational Experience with a Primal-Dual Interior-Point Method for Smooth Convex Programming.* to appear in Optimization Methods and Software.

[30] H. WOLKOWICZ. Some applications of optimization in matrix theory. *Linear Algebra and its Applications*, 40:101–118, 1981.