Incorporating Inequality Constraints in the Spectral Bundle Method

C. Helmberg[1], K.C. Kiwiel[2], and F. Rendl[3]

[1] Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7, 14195 Berlin, Germany
helmberg@zib.de, http://www.zib.de/helmberg

[2] Systems Research Institute, Polish Academy of Sciences
Newelska 6, 01-447 Warsaw, Poland
kiwiel@ibspan.waw.pl

[3] Technische Universität Graz, Institut für Mathematik
Steyrergasse 30, A-8010 Graz, Austria
rendl@opt.math.tu-graz.ac.at

Abstract. Semidefinite relaxations of quadratic 0-1 programming or graph partitioning problems are well known to be of high quality. However, solving them by primal-dual interior point methods can take much time even for problems of moderate size. Recently we proposed a spectral bundle method that allows to compute, within reasonable time, approximate solutions to structured, large equality constrained semidefinite programs if the trace of the primal matrix variable is fixed. The latter property holds for the aforementioned applications. We extend the spectral bundle method so that it can handle inequality constraints without seriously increasing computation time. This makes it possible to apply cutting plane algorithms to semidefinite relaxations of real world sized instances. We illustrate the efficacy of the approach by giving some preliminary computational results.

1 Introduction

Since the landmark papers [9, 4, 10, 2] it is well known that semidefinite programming allows to design powerful relaxations for constrained quadratic 0-1 programming and graph partitioning problems. The most commonly used algorithms for solving these relaxations, primal-dual interior point methods, offer little possibilities to exploit problem structure. Typically their runtime is governed by the factorization of a dense symmetric positive definite matrix in the number of constraints and by the line search that ensures the positive definiteness of the matrix variables. Computation times for problems with more than 3000 constraints or matrix variables of order 500, say, are prohibitive. Very recently, a pure dual approach has been proposed in [1] that is able to exploit the sparsity of the cost matrix in the case of the max-cut relaxation with diagonal constraints. It is not yet clear whether these results extend to problems with a huge number of less structured constraints.

The spectral bundle method [6] works on a reformulation of semidefinite relaxations as eigenvalue optimization problems and was developed to provide approximate solutions to structured problems fast. In contrast to standard bundle methods [7, 11], a non-polyhedral semidefinite cutting plane model is constructed from the subgradients. Reinterpreted in terms of the original semidefinite program, the semidefinite model ensures positive semidefiniteness of the dual matrix variable on a subspace only. Subgradients correspond to eigenvectors to negative eigenvalues and are used to correct the subspace. By means of an aggregate subgradient the dimension of the subspace can be kept small, thus ensuring efficient solvability of the subproblems. Lanczos methods (see e.g. [3]) allow to compute a few extremal eigenvalues and their eigenvectors efficiently by a series of matrix-vector multiplications which do not require the matrix in explicit form. Thus structural properties of cost and coefficient matrices can be exploited.

Like most first order methods, the spectral bundle method exhibits fast progress in the beginning, but shows a strong tailing off effect as the optimal solution is approached. Fortunately many semidefinite relaxations do not have to be solved exactly. Rather an approximate solution is used to improve, e.g., by cutting planes, the current relaxation, which is then resolved.

In its original form the spectral bundle method is designed for equality constraints only because sign constraints on the dual variables may increase computation times for the semidefinite subproblem significantly. In this paper we employ Lagrangian relaxation to approximate the solution of a sign constrained semidefinite subproblem. Surprisingly just one update of Lagrange multipliers per function evaluation suffices to ensure convergence. The semidefinite subproblem can be solved as efficiently as in the unconstrained case, thus rendering this method an attractive choice for large scale semidefinite cutting plane algorithms.

Section 2 introduces some notation and explains the connection to semidefinite programming. This is followed by a very brief review of important properties of the maximal eigenvalue function. Section 4 explains the extension of the spectral bundle method to inequality constraints. In Section 5 we discuss efficiency aspects of the subproblem solution. Section 6 gives computational results.

2 Semidefinite Programs

Let $\mathcal{S}_n$ denote the space of symmetric matrices of order n. The inner product in this space is the usual matrix inner product, $\langle A, B\rangle = \operatorname{tr}(B^T A)$ for $A, B \in \mathbb{R}^{m\times n}$. Let $\mathcal{S}_n^+$ denote the set of symmetric positive semidefinite matrices. $\mathcal{S}_n^+$ is a pointed closed convex cone. Except for its apex $\{0\}$, a face F of this cone can be described as

$$F = \left\{PVP^T : V \in \mathcal{S}_r^+\right\},$$

where $P \in \mathbb{R}^{n\times r}$ is some fixed matrix with orthonormal columns (w.l.o.g.). The dimension of such a face F is $\binom{r+1}{2}$. For $A, B \in \mathcal{S}_n$, $A \succeq B$ refers to the Löwner partial order induced by the cone $\mathcal{S}_n^+$ ($A \succeq B \iff A - B \in \mathcal{S}_n^+$).

In the following $\mathcal{A} : \mathcal{S}_n \to \mathbb{R}^m$ is a linear operator and $\mathcal{A}^T(\cdot)$ is the corresponding adjoint operator satisfying $\langle \mathcal{A}(X), y\rangle = \langle X, \mathcal{A}^T(y)\rangle$ for all $X \in \mathcal{S}_n$ and

$y \in \mathbb{R}^m$. The operators are of the form

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

We will denote a subset $J \subset \{1, \ldots, m\}$ of the rows of $\mathcal{A}$ by $\mathcal{A}_J$. Likewise we will speak of $\mathcal{A}_J^T$ and y_J for some vector $y \in \mathbb{R}^m$.

We consider semidefinite programs of the following form with $J \subset \{1, \ldots, m\}$ and $\bar{J} = \{1, \ldots, m\} \setminus J$,

$$\text{(P)} \quad \begin{array}{ll} \max & \langle C, X \rangle \\ \text{s.t.} & \mathcal{A}_{\bar{J}}(X) = b_{\bar{J}} \\ & \mathcal{A}_J(X) \leq b_J \\ & X \succeq 0. \end{array} \qquad \text{(D)} \quad \begin{array}{ll} \min & \langle b, y \rangle \\ \text{s.t.} & Z = \mathcal{A}^T(y) - C \\ & Z \succeq 0, y_J \geq 0. \end{array}$$

Under a constraint qualification (which we tacitly assume to hold), any optimal solution X^* of (P) and any optimal solution (y^*, Z^*) of (D) satisfy $X^* Z^* = 0$.

We assume that there exists $\hat{y} \in \mathbb{R}^m$ with $I = \mathcal{A}_{\bar{J}}^T(\hat{y})$, which is equivalent to the requirement that $\operatorname{tr}(X) = a = b_{\bar{J}}^T \hat{y}_{\bar{J}}$ for all feasible X. Most semidefinite relaxations of combinatorial optimization problems satisfy this requirement or can be scaled appropriately. To avoid the trivial case of $X = 0$, assume $a > 0$.

Under these assumptions it is well known that (D) is equivalent to the eigenvalue optimization problem (see e.g. [8])

$$\min_{y \in \mathbb{R}^m, y_J \geq 0} a \lambda_{\max}(C - \mathcal{A}^T(y)) + \langle b, y \rangle .$$

To simplify notation we drop, without loss of generality, the coefficient a and assume that $|J| = m$, i.e., there are inequality constraints only. Thus we concentrate on the following problem

$$\min_{y \geq 0} f(y) \qquad \text{with} \qquad f(y) = \lambda_{\max}(C - \mathcal{A}^T(y)) + \langle b, y \rangle , \tag{1}$$

or equivalently $\min f_S := f + \imath_S$ with $\imath_S$ being the indicator function of $S = \mathbb{R}_+^m$ ($\imath_S(y) = 0$ if $y \in S$, ∞ otherwise).

3 The Function $\lambda_{\max}(X)$

The maximal eigenvalue $\lambda_{\max}(X)$ of a matrix is a nonsmooth convex function. The kinks of $\lambda_{\max}(X)$ appear at points where the maximal eigenvalue has multiplicity at least two. If, for some $\hat{X} \in \mathcal{S}_n$, the columns of $P \in \mathbb{R}^{n \times r}$ form an orthonormal basis of the eigenspace of the maximal eigenvalue of $\hat{X}$, then the subdifferential (the set of subgradients) of $\lambda_{\max}(\cdot)$ at $\hat{X}$ is

$$\begin{aligned} \partial \lambda_{\max}(\hat{X}) &= \left\{ PVP^T : \operatorname{tr}(V) = 1, V \in \mathcal{S}_r^+ \right\} \qquad (2) \\ &= \operatorname{conv} \left\{ vv^T : \; v^T \hat{X} v = \lambda_{\max}(\hat{X}), \|v\| = 1 \right\} . \end{aligned}$$

It is a well known fact that

$$\lambda_{\max}(X) = \max\left\{\langle W, X\rangle : \operatorname{tr}(W) = 1, W \succeq 0\right\}.$$

Any subset $\widehat{\mathcal{W}} \subset \{W \succeq 0 : \operatorname{tr}(W) = 1\}$ gives rise to a non-polyhedral cutting plane model minorizing $\lambda_{\max}(X)$,

$$\lambda_{\max}(X) \geq \max\left\{\langle W, X\rangle : W \in \widehat{\mathcal{W}}\right\}. \tag{3}$$

4 The Spectral Bundle Method with Inequality Constraints

It follows from the properties of $\lambda_{\max}(\cdot)$ that (1) is a nonsmooth convex minimization problem. A standard approach to solve such problems is the proximal bundle method [7]. We first sketch the method in general and give the corresponding symbols that we will use for the spectral bundle method. Afterwards we specialize the algorithm to the eigenvalue problem (1).

At iteration k the algorithm generates a test point $y^{k+1} = \arg\min_S \hat{f}^k + \frac{u}{2}\|\cdot - x^k\|^2$, where $\hat{f}^k$ is an accumulated cutting plane model of f and the weight $u > 0$ keeps y^{k+1} near the current iterate x^k. A *serious step* $x^{k+1} = y^{k+1}$ occurs if $f(x^k) - f(y^{k+1}) \geq m_L[f(x^k) - \hat{f}^k(y^{k+1})]$, where $m_L \in \left(0, \frac{1}{2}\right)$. Otherwise a *null step* $x^{k+1} = x^k$ is made but the new subgradient computed at y^{k+1} improves the next model.

The spectral bundle method is tailored for the eigenvalue problem (1) in that, instead of the usual polyhedral $\hat{f}^k$, it uses a semidefinite cutting plane model of the form (cf. (3))

$$\hat{f}^k(y) = \max_{W \in \widehat{\mathcal{W}}^k} \left\langle W, C - \mathcal{A}^T(y)\right\rangle + \langle b, y\rangle. \tag{4}$$

We will specify the choice of $\widehat{\mathcal{W}}^k$ and discuss the computation of $\hat{f}^k$ in detail in Sect. 5. The usual bundle subproblem of finding

$$y_*^{k+1} = \arg\min_{y \geq 0} \hat{f}^k(y) + \frac{u}{2}\|y - x^k\|^2 \tag{5}$$

turns out to be rather expensive computationally because of the sign constraints on y. Instead we compute, by means of Lagrangian relaxation, an approximation y^{k+1} to y_*^{k+1}. To this end we define the Lagrangian of (4)–(5)

$$L(y; W, \eta) = \langle C, W\rangle + \langle b - \mathcal{A}(W), y\rangle - \langle \eta, y\rangle + \frac{u}{2}\|y - x^k\|^2$$

for $W \in \widehat{\mathcal{W}}^k$ and $\eta \in \mathbb{R}_+^m$. It has the form $L(\cdot; W, \eta) = \bar{f}_{W,\eta}(\cdot) + \frac{u}{2}\|\cdot - x^k\|^2$, where

$$\bar{f}_{W,\eta}(\cdot) = \langle C, W\rangle + \langle b - \eta - \mathcal{A}(W), \cdot\rangle$$

is an affine function minorizing $\hat{f}_S^k = \hat{f}^k + \imath_S$ and hence f_S. The dual function $\varphi(W, \eta) = \min L(\cdot; W, \eta)$, computed by finding

$$y_{W,\eta} = \arg\min_y \bar{f}_{W,\eta}(y) + \frac{u}{2}\|y - x^k\|^2 = x^k - \frac{b - \eta - \mathcal{A}(W)}{u},$$

has the form

$$\varphi(W, \eta) = \langle C, W \rangle + \left\langle b - \eta - \mathcal{A}(W), x^k \right\rangle - \frac{1}{2u}\|b - \eta - \mathcal{A}(W)\|^2.$$

Denote by η^k the multipliers of the previous iteration. We find an approximate maximizer (W^{k+1}, η^{k+1}) to φ over $\widehat{\mathcal{W}}^k \times \mathbb{R}_+^m$ by first computing

$$W^{k+1} \in \operatorname{Arg} \max_{W \in \widehat{\mathcal{W}}^k} \varphi(W, \eta^k) \tag{6}$$

via an interior point algorithm, and then setting

$$\eta^{k+1} = \arg\max_{\eta \geq 0} \varphi(W^{k+1}, \eta) = -u \min\left\{0, x^k - \frac{b - \mathcal{A}(W)}{u}\right\}. \tag{7}$$

It is not difficult to check that

$$y^{k+1} := y_{W^{k+1},\eta^{k+1}} \geq 0 \qquad \text{and} \qquad \left\langle \eta^{k+1}, y^{k+1} \right\rangle = 0.$$

At the test point y^{k+1} the function is evaluated and a new subgradient is computed.

By the observations above, we have

$$\bar{f}^{k+1} = \bar{f}_{W^{k+1},\eta^{k+1}} \leq \hat{f}_S^k \leq f_S.$$

This and the following lemma motivate a stopping criterion of the form

$$f(x^k) - \bar{f}^{k+1}(y^{k+1}) \leq \varepsilon_{\text{opt}}(|f(x^k)| + 1).$$

Lemma 1. *$f(x^k) \geq \bar{f}^{k+1}(y^{k+1})$, and if $f(x^k) = \bar{f}^{k+1}(y^{k+1})$ then x^k is optimal.*

Proof. Since $f(x^k) \geq \bar{f}^{k+1}(x^k)$ and

$$\begin{aligned}\bar{f}^{k+1}(x^k) - \bar{f}^{k+1}(y^{k+1}) &= \left\langle b - \eta^{k+1} - \mathcal{A}(W^{k+1}), x^k - y^{k+1} \right\rangle \\ &= \frac{1}{u}\|b - \eta^{k+1} - \mathcal{A}(W^{k+1})\|^2,\end{aligned}$$

the inequality follows. If equality holds then $f(x^k) = \bar{f}^{k+1}(x^k)$ and $0 = b - \eta^{k+1} - \mathcal{A}(W^{k+1}) = \nabla \bar{f}^{k+1} \in \partial f_S(x^k)$. □

For practical reasons we state the algorithm with an inner loop allowing for several repetitions of the two "coordinatewise" maximization steps (6) and (7).

Algorithm 1.
Input: $y^0 \in \mathbb{R}^m_+$, $\varepsilon_{\text{opt}} \geq 0$, $\varepsilon_{\text{M}} \in (0, \infty]$, *an improvement parameter* $m_L \in (0, \frac{1}{2})$, *a weight* $u > 0$.

1. *(Initialization) Set* $k = 0$, $x^0 = y^0$, $\eta^0 = 0$, $f(x^0)$ *and* $\widehat{\mathcal{W}}^0$ *(cf. Sect. 5).*
2. *(Direction finding) Set* $\eta^+ = \eta^k$.
   (a) *Find* $W^+ \in \operatorname{Arg\,max}_{W \in \widehat{\mathcal{W}}^k} \varphi(W, \eta^+)$.
   (b) *Compute* $\eta^+ = \arg\max_{\eta \geq 0} \varphi(W^+, \eta)$.
   (c) *Compute* $y^+ = y_{W^+, \eta^+}$.
   (d) *(Termination) If* $f(x^k) - \bar{f}_{W^+,\eta^+}(y^+) \leq \varepsilon_{\text{opt}}(|f(x^k)| + 1)$ *then STOP.*
   (e) *If* $\hat{f}^k(y^+) - \bar{f}_{W^+,\eta^+}(y^+) > \varepsilon_{\text{M}}[f(x^k) - \bar{f}_{W^+,\eta^+}(y^+)]$ *then go to (a).*
   Set $\eta^{k+1} = \eta^+$, $W^{k+1} = W^+$, $y^{k+1} = y^+$.
3. *(Evaluation) Compute* $f(y^{k+1})$,
   find $W_S^{k+1} \in \operatorname{Arg\,max} \left\{ \left\langle C - \mathcal{A}^T(y^{k+1}), W \right\rangle : \operatorname{tr}(W) = 1, W \succeq 0 \right\}$.
4. *(Model updating) Determine* $\widehat{\mathcal{W}}^{k+1} \supseteq \operatorname{conv}\left\{W^{k+1}, W_S^{k+1}\right\}$ *(cf. Sect. 5).*
5. *(Descent test) If* $f(x^k) - f(y^{k+1}) \geq m_L[f(x^k) - \bar{f}^{k+1}(y^{k+1})]$ *then set* $x^{k+1} = y^{k+1}$ *(serious step); otherwise set* $x^{k+1} = x^k$ *(null step).*
6. *Increase* k *by* 1 *and go to Step 2.*

For $\varepsilon_{\text{M}} = \infty$ the algorithm performs exactly one inner iteration as described before. This single inner iteration suffices to guarantee convergence.

Theorem 1. *Let* $\varepsilon_{\text{opt}} = 0$ *and* $\varepsilon_{\text{M}} = \infty$. *If* $\operatorname{Arg\,min} f_S \neq \emptyset$ *then* $x^k \to \bar{x} \in \arg\min f_S$, *otherwise* $\|x^k\| \to \infty$. *In both cases* $f(x^k) \downarrow \inf f_S$.

The proof is rather technical and we refer the reader to the full version of the paper for details. To sketch the main idea only, assume that test point y^k has caused a null step. Then y^k is the (unconstrained) minimizer of the strongly convex function $\bar{f}^k(\cdot) + \frac{u}{2}\|\cdot - x^k\|^2$ with modulus u and therefore, since $W^k \in \widehat{\mathcal{W}}^k$,

$$L(y_{W^{k+1},\eta^k}; W^{k+1}, \eta^k) \geq L(y^k; W^k, \eta^k) + \frac{u}{2}\|y_{W^{k+1},\eta^k} - y^k\|^2.$$

Likewise we obtain

$$L(y^{k+1}; W^{k+1}, \eta^{k+1}) \geq L(y_{W^{k+1},\eta^k}; W^{k+1}, \eta^k) + \frac{u}{2}\|y^{k+1} - y_{W^{k+1},\eta^k}\|^2.$$

Using these relations and the fact that the subgradients generated by the algorithm remain locally bounded one can arrive at Theorem 1 along a similar line of arguments as given in [7].

5 $\widehat{\mathcal{W}}^k$ and the Quadratic Semidefinite Subproblem

Whereas the convergence analysis of the algorithm imposes only very few requirements on $\widehat{\mathcal{W}}^k$ and its update to $\widehat{\mathcal{W}}^{k+1}$, its actual realization is of utmost importance for the efficiency of the algorithm. The spectral bundle method uses

$$\widehat{\mathcal{W}}^k = \left\{ W = P^k V (P^k)^T + \alpha \overline{W}^k : \operatorname{tr}(V) + \alpha = 1, V \in S_r^+, \alpha \geq 0 \right\},$$

where $P^k \in \mathbb{R}^{n \times r}$ is some fixed matrix with orthonormal columns and $\overline{W}^k \in S_n^+$ satisfies $\operatorname{tr}(\overline{W}^k) = 1$. P^k should span at least partially the eigenspace belonging to the largest eigenvalues of matrices $C - \mathcal{A}^T(y)$ for y in the vicinity of x^k. $\overline{W}^k$ serves to aggregate subgradient information that cannot be represented within the set $\{P^k V (P^k)^T : \operatorname{tr}(V) = 1, V \in S_r^+\}$. The use of $\overline{W}^k$ ensures convergence of the spectral bundle method even if the number of columns of P^k is restricted to $r = 1$. The quality of the semidefinite model as well as the computation time for solving (6) depend heavily on r.

The special structure of $\widehat{\mathcal{W}}^k$ allows us to evaluate $\hat{f}^k$ directly:

$$\hat{f}^k(y) = \max\left\{\lambda_{\max}((P^k)^T(C - \mathcal{A}^T(y))P^k), \left\langle \overline{W}^k, C - \mathcal{A}^T(y) \right\rangle\right\} + \langle b, y \rangle .$$

Observe that the eigenvalue computation involved is cheap since the argument is an $r \times r$ symmetric matrix.

By the choice of $\widehat{\mathcal{W}}^k$, for a fixed $\eta = \eta^k$, (6) is equivalent to the quadratic semidefinite program

$$\begin{array}{ll} \min & \frac{1}{2u}\|b - \eta - \mathcal{A}(P^k V (P^k)^T + \alpha \overline{W}^k)\|^2 \\ & \quad - \left\langle P^k V (P^k)^T + \alpha \overline{W}^k, C - \mathcal{A}^T(x^k) \right\rangle - \left\langle b - \eta, x^k \right\rangle \\ \text{s.t.} & \operatorname{tr}(V) + \alpha = 1 \\ & V \succeq 0, \alpha \geq 0. \end{array} \tag{8}$$

Its optimal solution (V^*, α^*) gives rise to $W^{k+1} = P^k V^* (P^k)^T + \alpha^* \overline{W}^k$ and is also used to update P^k and $\overline{W}^k$ in Step 4 of the algorithm. Let $Q \Lambda Q^T$ be an eigenvalue decomposition of V^*. Then the 'important' part of the spectrum of W^{k+1} is spanned by the eigenvectors associated with the 'large' eigenvalues of V^*. Thus the eigenvectors of Q are split into two parts $Q = [Q_1 Q_2]$ (with corresponding spectra Λ_1 and Λ_2), Q_1 containing as columns the eigenvectors associated to 'large' eigenvalues of V^* and Q_2 containing the remaining columns. P^{k+1} then contains an orthonormal basis of the columns of $P^k Q_1$ and at least one eigenvector to the maximal eigenvalue of $C - \mathcal{A}^T(y^{k+1})$ computed in Step 3 of the algorithm. The next aggregate matrix is

$$\overline{W}^{k+1} = \alpha^* \overline{W}^k + P^k Q_2 \Lambda_2 Q_2^T (P^k)^T / (\alpha^* + \operatorname{tr}(\Lambda_2)). \tag{9}$$

This ensures $W^{k+1} \in \widehat{\mathcal{W}}^{k+1}$. For $\widehat{\mathcal{W}}^0$ we choose P^0 to contain a set of orthonormalized eigenvectors to large eigenvalues of $C - \mathcal{A}(y^0)$. We do not use the aggregate in the first iteration.

(8) has to be solved for each update of η in the inner loop or, if $\varepsilon_{\mathrm{M}} = \infty$, at least once per outer iteration. Thus the efficiency of the whole algorithm hinges on the speed of this computation. We briefly explain the most important issues that arise in this context.

Using the svec-operator described in [12] to expand symmetric matrices from $\mathcal{S}_r$ into column vectors of length $\binom{r+1}{2}$ and by ignoring all constants, (8) can be brought into the following form (recall that, for $A, B \in \mathcal{S}_r$, $\langle A, B\rangle = \operatorname{svec}(A)^T \operatorname{svec}(B)$ and that $\operatorname{tr}(V) = \langle I, V\rangle$)

$$\begin{aligned} \min\ & \tfrac{1}{2}\operatorname{svec}(V)^T Q_{11} \operatorname{svec}(V) + \alpha q_{12}^T \operatorname{svec}(V) + \tfrac{1}{2} q_{22}\alpha^2 + c_1^T \operatorname{svec}(V) + c_2\alpha \\ \text{s.t. } & \alpha + s_I^T \operatorname{svec}(V) = 1 \\ & \alpha \geq 0, V \succeq 0, \end{aligned} \tag{10}$$

where

$$\begin{aligned} Q_{11} &= \tfrac{1}{u} \textstyle\sum_{i=1}^m \operatorname{svec}(P^T A_i P) \operatorname{svec}(P^T A_i P)^T \\ q_{12} &= \tfrac{1}{u} \operatorname{svec}(P^T \mathcal{A}^T(\mathcal{A}(\overline{W}))P) \\ q_{22} &= \tfrac{1}{u} \left\langle \mathcal{A}(\overline{W}), \mathcal{A}(\overline{W}) \right\rangle \\ c_1 &= -a \operatorname{svec}(P^T(\tfrac{1}{u}\mathcal{A}^T(b-\eta) + C - \mathcal{A}^T(y))P) \\ c_2 &= -a(\left\langle \tfrac{1}{u}(b-\eta) - y, \mathcal{A}(\overline{W}) \right\rangle + \left\langle C, \overline{W} \right\rangle) \\ s_I &= \operatorname{svec}(I). \end{aligned}$$

This problem has $\binom{r+1}{2}+1$ variables and can be solved quite efficiently by interior point methods if r is not too large, smaller than 25, say. Since convergence is guaranteed even for $r = 1$, it is possible to run this algorithm for problems with a huge number of constraints m. Several remarks are in order.

First, it is not necessary to have $\overline{W}$ available as a matrix, it suffices to store the m-vector $\mathcal{A}(\overline{W})$ and the scalar $\left\langle C, \overline{W} \right\rangle$. These values are easily updated whenever $\overline{W}$ is changed.

Second, almost all computations involve the projected matrices $P^T A_i P$. These have to be computed only once for each evaluation of (10), they are symmetric of size $r \times r$, and only one such projected matrix has to be kept in memory if the values of Q_{11} to c_2 are accumulated.

Third, the most expensive operation in computing the cost coefficients is the accumulation of Q_{11}, which involves the summation of m dyadic products of vectors of size $\binom{r+1}{2}$ for a total of $O(mr^4)$ operations. Even for rather small r but sufficiently large m, this operation takes longer than solving the reduced quadratic semidefinite program.

Finally, changes in η do not affect the quadratic cost matrices Q_{11} to q_{22}, but only the linear cost coefficients c_1 and c_2. Within the inner loop it suffices to update the linear cost coefficients. The dominating cost of computing Q_{11} can be avoided. If the inner iteration yields only small changes in η then the optimal solution will also change only slightly. This can be exploited in restarting strategies.

We solve (10) by the primal-dual interior point code used in [6]. It maintains primal and dual feasibility throughout (U is the dual variable of V, β is the dual variable of α, and t is the dual variable to the primal equality constraint),

$$Q_{11}\operatorname{svec}(V) + \alpha q_{12} + c_1 - t s_I - \operatorname{svec}(U) = 0, \tag{11}$$

$$\alpha q_{22} + q_{12}^T \operatorname{svec}(V) + c_2 - t - \beta = 0, \tag{12}$$

$$1 - \alpha - s_I^T \operatorname{svec}(V) = 0. \tag{13}$$

We employ a restart procedure for reoptimizing the quadratic program in the inner loop. Let $(U^*, V^*, \alpha^*, \beta^*, t^*)$ denote the optimal solution of the latest problem, and let Δc_1 and Δc_2 denote the update (caused by the change in η) to be added to c_1 and c_2. We determine a starting point $(U^0, V^0, \alpha^0, \beta^0, t^0)$ by the following steps. First, note that $(V, \alpha) = (I, 1)/(r+1)$ is the analytic center of the primal feasible set. Therefore any point on the straight line segment $\xi(V^*, \alpha^*) + (1-\xi)(I, 1)/(r+1)$ with $\xi \in [0,1]$ is again primal feasible. Furthermore it can be expected to be a point close to the central path if the new optimal solution is relatively close to the old optimal solution. We determine ξ by $\xi = \min\{.99999, \max[.9, \sqrt{1 - \frac{\|(\Delta c_1, \Delta c_2)\|}{\|(c_1, c_2)\|}}]\}$.

In (11) and (12) the change from (V^*, α^*) to the starting point (V^0, α^0) determines the changes in the dual variables up to a diagonal shift that can be applied through t. This diagonal shift is chosen so that the changes in U and β are positive definite. Thus U^0 (β^0) is strictly positive definite but not too far from U^* (β^*). Experience shows that this restarting heuristic reduces the number of iterations to two thirds down to one half depending on the size of the changes in η.

6 Implementation and Combinatorial Applications

The algorithm has been implemented as a direct extension of the algorithm in [6] and uses essentially the same parameter settings. In particular, $m_L = 0.1$, $r = 25$, $\varepsilon_{\text{opt}} = 5 \cdot 10^{-4}$, and the weight u is updated dynamically by safeguarded quadratic interpolation. We present some computational results for $\varepsilon_{\text{M}} = .6$ and $\varepsilon_{\text{M}} = \infty$. The choice of $\varepsilon_{\text{M}} = .6$ should keep the number of inner iterations small, at the same time avoiding null steps enforced by the bad quality of the relaxation. Our computational experiments were carried out on a Sun sparc Ultra 1 with a Model 140 UltraSPARC CPU and 64 MB RAM. Computation times are given in hours:minutes:seconds.

We apply the code to the max-cut instances G_1 to G_{21} on 800 vertices from [6]. Graphs G_1 to G_5 are unweighted random graphs with a density of 6% (approx. 19000 edges). G_6 to G_{10} are the same graphs with random edge weights from $\{-1, 1\}$. G_{11} to G_{13} are toroidal grids with random edge weights from $\{-1, 1\}$ (approx. 1200 edges). G_{14} to G_{17} are unweighted 'almost' planar graphs having as edge set the union of two (almost maximal) planar graphs (approx. 4500 edges). G_{18} to G_{21} are the same almost planar graphs with random edge weights from $\{-1, 1\}$. In all cases the cost matrix C is the Laplace matrix of the graph divided by four, i.e., let A denote the (weighted) adjacency matrix of G, then $C = (\text{Diag}(Ae) - A)/4$.

To the basic semidefinite relaxation $(\text{diag}(X) = e)$ we add 1600 triangle inequalities. These were determined by our code from [5], which is a primal-dual interior point code. We applied the separation routine of this code to the optimal solution of the semidefinite relaxation without inequality constraints. We also used the code to solve the relaxation with the same triangle constraints and will compare these results to the spectral bundle method.

Table 1. Comparison of the spectral bundle and interior point approaches

Problem	IP-sol	spectral bundle		interior point	
		1%	0.1%	1%	0.1%
G_1	12049.33	19	3:51	2:16:14	4:05:13
G_2	12053.88	20	5:37	2:15:57	4:04:42
G_3	12048.56	17	4:13	2:44:06	4:06:09
G_4	12071.59	17	3:11	2:42:53	4:04:20
G_5	12059.68	15	3:21	3:39:56	5:02:24
G_6	2615.55	1:24	14:47	3:37:52	4:59:34
G_7	2451.61	1:28	11:55	3:37:48	4:59:28
G_8	2463.68	1:42	19:19	3:37:47	4:59:27
G_9	2493.51	1:38	16:44	3:38:45	5:00:46
G_{10}	2447.79	1:22	12:17	3:35:53	4:57:31
G_{11}	623.49	9:25	54:55	4:55:20	6:10:01
G_{12}	613.02	9:32	47:01	4:57:09	6:43:04
G_{13}	636.45	8:52	39:00	4:31:17	6:46:56
G_{14}	3181.35	1:19	10:13	3:36:13	5:24:18
G_{15}	3161.89	1:09	11:51	3:35:00	5:22:31
G_{16}	3164.95	1:24	12:55	4:07:35	5:57:37
G_{17}	3161.78	1:20	12:32	3:35:22	5:23:04
G_{18}	1147.61	4:10	15:45	4:30:04	5:51:05
G_{19}	1064.57	4:23	17:17	4:59:07	6:47:54
G_{20}	1095.46	4:31	14:58	4:30:13	5:51:18
G_{21}	1087.89	3:22	12:40	4:56:09	6:16:55

For the bundle algorithm the diagonal of C is removed. This does not change the problem because the diagonal elements of X are fixed to one. The offset $e^T(Ae - \operatorname{diag}(A))/4$ is not used internally in the spectral bundle code. However, after each serious step we check externally whether the stopping criterion is satisfied if the offset is added. If so, we terminate the algorithm. As starting vector y^0 we choose the zero vector.

For inequality constrained problems our spectral bundle code exhibits an even stronger tailing off effect than for equality constrained problems. To solve the relaxation exactly seems to be out of reach. However, within the first few iterations the objective value gets already very close to the optimal value. In cutting plane approaches it does not make sense to solve relaxations to optimality if much faster progress can be achieved by adding new inequalities. Therefore the quick initial convergence is a desirable property for these applications.

In this spirit we give the time needed to get within 1% and 0.1% of the optimum in Table 1 for both the spectral bundle code with $\varepsilon_{\mathrm{M}} = .6$ and the interior point code. The first column specifies the problem instance, the second (IP-sol) gives the optimal value of the relaxation with triangle inequalities. Columns three and five (four and six) display the times needed by both codes to get within 1% (0.1%, resp.) of the optimum. (The entries for the interior point code are estimates, since its output only gives the overall computation time.)

Table 2. Performance of the spectral bundle code for $\varepsilon_{\text{opt}} = 5 \cdot 10^{-4}$ and $\varepsilon_{\text{M}} = .6$

Problem	f^*	$f(x^k)$	$\frac{f(x^k)-f^*}{f^*}$	total time	eig_time[%]	# calls	# serious
G_1	12049.33	12055.79	$5.4 \cdot 10^{-4}$	5:10	62.9	15	11
G_2	12053.88	12057.64	$3.1 \cdot 10^{-4}$	12:06	53.99	25	16
G_3	12048.56	12051.59	$2.5 \cdot 10^{-4}$	13:15	53.21	27	18
G_4	12071.59	12078.58	$5.8 \cdot 10^{-4}$	4:26	61.28	15	11
G_5	12059.68	12065.88	$5.1 \cdot 10^{-4}$	6:28	56.96	18	13
G_6	2615.55	2617.04	$5.7 \cdot 10^{-4}$	26:17	45.21	49	22
G_7	2451.61	2453.27	$6.8 \cdot 10^{-4}$	15:38	45.95	34	21
G_8	2463.68	2464.86	$4.8 \cdot 10^{-4}$	31:33	46.01	59	22
G_9	2493.51	2494.79	$4.8 \cdot 10^{-4}$	28:06	45.37	54	22
G_{10}	2447.79	2449.71	$7.8 \cdot 10^{-4}$	15:20	43.8	35	22
G_{11}	623.49	623.95	$7.3 \cdot 10^{-4}$	1:11:04	68.64	94	30
G_{12}	613.02	613.61	$9.6 \cdot 10^{-4}$	47:42	78.13	90	35
G_{13}	636.45	637.03	$9.1 \cdot 10^{-4}$	39:00	79.74	82	35
G_{14}	3181.35	3182.53	$3.7 \cdot 10^{-4}$	19:46	64.67	43	29
G_{15}	3161.89	3163.24	$4.3 \cdot 10^{-4}$	19:51	66.16	40	27
G_{16}	3164.95	3165.98	$3.3 \cdot 10^{-4}$	29:32	64.73	50	28
G_{17}	3161.78	3163.52	$5.5 \cdot 10^{-4}$	18:48	67.55	41	28
G_{18}	1147.61	1148.25	$5.6 \cdot 10^{-4}$	21:54	55.48	48	31
G_{19}	1064.57	1065.06	$4.6 \cdot 10^{-4}$	22:33	54.99	48	28
G_{20}	1095.46	1095.94	$4.4 \cdot 10^{-4}$	20:05	56.02	46	29
G_{21}	1087.89	1088.72	$7.6 \cdot 10^{-4}$	14:48	59.57	40	29

The fast progress of the bundle code on examples G_1 to G_5 can be explained by the large offset value (9500) that is added externally. For the other classes the offset is around 0 for G_6 to G_{13} and G_{18} to G_{21}, being about 2300 for G_{14} to G_{17}. The eigenvalue structure of the solutions of G_{10} to G_{13} suggests that the rather poor performance of the algorithm on these problems is caused by a very flat objective.

Table 2 provides more detailed information on the overall performance of the spectral bundle code for $\varepsilon_{\text{M}} = .6$. Column f^* refers to the value of the interior point solution, $f(x^k)$ gives the objective value at termination, $\frac{f(x^k)-f^*}{f^*}$ is the relative accuracy with respect to f^*, *total time* gives the computation time, *eig_time* is the percentage of time spent in the Lanczos code (eigenvalue and eigenvector computation), *# calls* counts the number of objective evaluations and *# serious* the number of serious steps.

Somewhat surprisingly the spectral bundle method also yields reasonable results for $\varepsilon_{\text{M}} = \infty$. The corresponding values are displayed in Table 3.

References

1. S. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. Working paper, Department of Management Science, University of Iowa, IA, 52242, USA, Sept. 1997.

Table 3. Performance of the spectral bundle code for $\varepsilon_{\mathrm{opt}} = 5 \cdot 10^{-4}$ and $\varepsilon_{\mathrm{M}} = \infty$

Problem	f^*	$f(x^k)$	$\frac{f(x^k)-f^*}{f^*}$	total time	eig_time[%]	# calls	# serious
G_1	12049.33	12055.79	$5.4{\cdot}10^{-4}$	5:14	62.42	15	11
G_2	12053.88	12057.31	$5.4{\cdot}10^{-4}$	13:13	54.22	27	16
G_3	12048.56	12053.27	$3.9{\cdot}10^{-4}$	11:54	54.06	25	17
G_4	12071.59	12078.58	$5.8{\cdot}10^{-4}$	4:23	61.98	15	11
G_5	12059.68	12065.88	$5.1{\cdot}10^{-4}$	6:26	57.51	18	13
G_6	2615.55	2616.89	$5.1{\cdot}10^{-4}$	26:32	45.79	52	24
G_7	2451.61	2453.27	$6.8{\cdot}10^{-4}$	15:14	46.50	34	21
G_8	2463.68	2464.86	$4.7{\cdot}10^{-4}$	32:02	44.59	59	22
G_9	2493.51	2494.69	$4.7{\cdot}10^{-4}$	30:14	46.14	58	23
G_{10}	2447.79	2449.51	$7.0{\cdot}10^{-4}$	14:32	44.84	34	22
G_{11}	623.49	623.89	$6.4{\cdot}10^{-4}$	1:30:05	71.3	113	33
G_{12}	613.02	613.69	$1.1{\cdot}10^{-3}$	53:17	74.8	95	33
G_{13}	636.45	637.05	$9.4{\cdot}10^{-4}$	43:31	79.1	87	36
G_{14}	3181.35	3182.68	$4.1{\cdot}10^{-4}$	19:23	64.49	43	28
G_{15}	3161.89	3163.18	$4.1{\cdot}10^{-4}$	19:29	66.47	40	27
G_{16}	3164.95	3165.98	$3.3{\cdot}10^{-4}$	28:55	65.42	50	28
G_{17}	3161.78	3163.46	$5.3{\cdot}10^{-4}$	18:35	67.89	41	28
G_{18}	1147.61	1148.28	$5.8{\cdot}10^{-4}$	20:16	58.31	47	30
G_{19}	1064.57	1065.04	$4.4{\cdot}10^{-4}$	23:24	59.69	51	28
G_{20}	1095.46	1095.94	$4.3{\cdot}10^{-4}$	18:39	59.34	46	28
G_{21}	1087.89	1088.63	$6.8{\cdot}10^{-4}$	15:09	61.17	42	29

2. M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995.
3. G. H. Golub and C. F. van Loan. *Matrix Computations.* The Johns Hopkins University Press, 2^{nd} edition, 1989.
4. M. Grötschel, L. Lovász, and A. Schrijver. Polynomial algorithms for perfect graphs. *Annals of Discrete Mathematics*, 21:325–356, 1984.
5. C. Helmberg and F. Rendl. Solving quadratic (0,1)-problems by semidefinite programs and cutting planes. ZIB Preprint SC-95-35, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Nov. 1995. To appear in Math. Programming.
6. C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. ZIB Preprint SC-97-37, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Aug. 1997. Revised October 1997.
7. K. C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Math. Programming*, 46:105–122, 1990.
8. A. S. Lewis and M. L. Overton. Eigenvalue optimization. *Acta Numerica*, pages 149–190, 1996.
9. L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, IT-25(1):1–7, Jan. 1979.
10. L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1(2):166–190, May 1991.

11. H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM J. Optim.*, 2:121–152, 1992.
12. M. J. Todd, K. C. Toh, and R. H. Tütüncü. On the Nesterov-Todd direction in semidefinite programming. Technical Report TR 1154, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 14853, Mar. 1996. Revised May 1996.