Optimization Methods and Software
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/goms20

The spectral bundle method with second-order information

C. Helmberg[a], M.L. Overton[b] & F. Rendl[c]

[a] Fakultät für Mathematik, Technische Universität Chemnitz, Chemnitz, Germany

[b] Courant Institute of Mathematical Sciences, New York University, New York, NY, USA

[c] Institut für Mathematik, Alpen-Adria Universität Klagenfurt, Klagenfurt, Austria

Accepted author version posted online: 18 Dec 2013.Published online: 07 Feb 2014.

To cite this article: C. Helmberg, M.L. Overton & F. Rendl (2014) The spectral bundle method with second-order information, Optimization Methods and Software, 29:4, 855-876, DOI: 10.1080/10556788.2013.858155

To link to this article: http://dx.doi.org/10.1080/10556788.2013.858155

Optimization Methods & Software, 2014
Vol. 29, No. 4, 855–876, http://dx.doi.org/10.1080/10556788.2013.858155

Taylor & Francis
Taylor & Francis Group

The spectral bundle method with second-order information

C. Helmberg[a*], M.L. Overton[b] and F. Rendl[c]

[a]*Fakultät für Mathematik, Technische Universität Chemnitz, Chemnitz, Germany;* [b]*Courant Institute of Mathematical Sciences, New York University, New York, NY, USA;* [c]*Institut für Mathematik, Alpen-Adria Universität Klagenfurt, Klagenfurt, Austria*

(*Received 10 September 2012; accepted 15 October 2013*)

The spectral bundle (SB) method was introduced by Helmberg and Rend [*A spectral bundle method for semidefinite programming*. SIAM J. Optim. 10 (2000), pp. 673–696] to solve a class of eigenvalue optimization problems that is equivalent to the class of semidefinite programs with the constant trace property. We investigate the feasibility and effectiveness of including full or partial second-order information in the SB method, building on work of Overton [*On minimizing the maximum eigenvalue of a symmetric matrix*. SIAM J. Matrix Anal. Appl. 9(2) (1988), pp. 256–268] and Overton and Womersley [*Second derivatives for optimizing eigenvalues of symmetric matrices*. SIAM J. Matrix Anal. Appl. 16 (1995), pp. 697–718]. We propose several variations that include second-order information in the SB method and describe efficient implementations. One of these, namely diagonal scaling based on a low-rank approximation of the second-order model for $\lambda_{\max}$, improves the standard SB method both with respect to accuracy requirements and computation time.

Keywords: semidefinite optimization; bundle methods; eigenvalue optimization

1. Introduction

Given $C, A_1, \ldots, A_m \in S^n$, the space of $n \times n$ real symmetric matrices, and a vector $b \in \mathbb{R}^m$, consider the optimization problem

$$\min_{y \in \mathbb{R}^m} \lambda_{\max}\left(C - \sum_i y_i A_i\right) + b^{\mathrm{T}} y, \tag{1}$$

where $\lambda_{\max}$ denotes the largest eigenvalue. The function

$$f(y) := \lambda_{\max}\left(C - \sum_i y_i A_i\right) + b^{\mathrm{T}} y$$

is convex but nonsmooth. Eigenvalue optimization problems of this kind have attracted much research over the past few decades, including the work of Cullum *et al.* [2], Overton [20,21], Schramm and Zowe [25], Jarre [15], Overton and Womersley [23], Shapiro and Fan [26], Helmberg and Rendl [13] and Oustry [18,19]. It is well known that the class of problems of the form (1)

*Corresponding author. Email: helmberg@mathematik.tu-chemnitz.de

is equivalent to the class of semidefinite programs (SDPs) with the constant trace property, as briefly discussed in the next section. It is this equivalence, together with the continued emerging importance of SDP and its applications—with the constant trace property holding in many cases—that largely motivates our work.

In order to be able to present our ideas more clearly, we summarize a somewhat simplified version of the (SB) algorithm of Helmberg and Rendl [13] in Section 3. In Section 4, we summarize the second-order method of Overton and Womersley [23]. Then in Section 5, we explain how to incorporate the second-order model into the SB method. A closely related algorithm, also incorporating such second-order information for the maximum eigenvalue function into a first-order bundle method, was proposed and analysed by Oustry [19]. However, this method has not been used much in practice. The difficulty is that the introduction of second-order approximations substantially raises the computational cost per iteration, resulting in an algorithm that is simply not competitive with interior-point methods for SDP.

The same is true of the second-order method that we introduce at the beginning of Section 5, but this is not the method that we advocate. Instead, we develop several much less computationally intensive variants in Section 5.4, after first discussing two important technical issues (how to estimate the multiplicity of the maximum eigenvalue and how to collect the corresponding active subspace during the bundle update) in Sections 5.2 and 5.3. These variants are based on low-rank approximations of this matrix, which is then approximated itself and finally reduced to its diagonal. The necessity to consider the entire active subspace has consequences for the scope of such scaled SB methods as explained in Section 5.5. The effectiveness of the methods is illustrated by numerical results that are reported in Section 6. Finally, we make some conclusions in Section 7.

2. Eigenvalue optimization and constant trace SDPs

Consider the primal SDP

$$\max_{X \in S^n} \langle C, X \rangle \quad \text{such that } \mathcal{A}X = b,\ X \succeq 0 \tag{2}$$

and its dual

$$\min_{u \in \mathbb{R}^m} b^{\mathrm{T}} u \quad \text{such that } \mathcal{A}^{\mathrm{T}} u - C \succeq 0. \tag{3}$$

As usual, $X \succeq 0$ means that X is in S^n_+, the cone of symmetric, positive semidefinite matrices. The notation $\langle \cdot, \cdot \rangle$ refers to the trace inner product on S^n, and $\mathcal{A}X$ represents the vector with components $\langle A_i, X \rangle$ for $i = 1, \ldots, m$, with $\mathcal{A}^{\mathrm{T}} y = \sum_i y_i A_i$ defining its adjoint. It is well known [29] that if both (2) and (3) have strictly feasible points, then their optimal values are the same and are attained by optimizers X and u satisfying the complementarity equation $X(\mathcal{A}^{\mathrm{T}} u - C) = 0$.

We say that the operator $\mathcal{A}$ has the *constant trace property* if the identity matrix I is in the range of $\mathcal{A}^{\mathrm{T}}$, i.e. $\exists \eta$ such that $\mathcal{A}^{\mathrm{T}} \eta = I$. The constant trace property implies constant trace of primal feasible matrices, that is

$$\mathcal{A}X = b \text{ implies } \operatorname{tr}(X) = \langle I, X \rangle = \langle \mathcal{A}^{\mathrm{T}} \eta, X \rangle = \langle \eta, \mathcal{A}X \rangle = \eta^T b.$$

It is shown in [13] that, if the constant trace property holds, then there is a simple relationship between the solution sets of (3) and the problem $\min_{y \in \mathbb{R}^m} (\eta^{\mathrm{T}} b) \lambda_{\max}(C - \mathcal{A}^{\mathrm{T}} y) + b^{\mathrm{T}} y$, which is (1) for $\eta^{\mathrm{T}} b = 1$.

The subdifferential of f at a given point y is given by

$$\partial f(y) = \{b - \mathcal{A}(W) : \langle W, C - \mathcal{A}^{\mathrm{T}} y\rangle = \lambda_{\max}(C - \mathcal{A}^{\mathrm{T}} y),\ W \in \mathcal{W}_n\}$$
$$\text{with} \quad \mathcal{W}_n := \{W \in S^n : \operatorname{tr} W = 1, W \succeq 0\}.$$

Since f is convex, y^* solves (1) if and only if $0 \in \partial f(y^*)$. This can be rewritten as follows. Let $\mathcal{O}_{n,r}$ denote the $n \times r$ matrices with orthonormal columns, i.e. $P \in \mathcal{O}_{n,r}$ satisfies $P^{\mathrm{T}}P = I$, the identity matrix of order r. Suppose y^* solves (1), and let $\lambda^* = \lambda_{\max}(C - \mathcal{A}^{\mathrm{T}} y^*)$ have multiplicity r^*. Then there exists $P^* \in \mathcal{O}_{n,r^*}$ and $U^* \in \mathcal{W}_r$ satisfying the following conditions:

- $\lambda^* I \succeq C - \mathcal{A}^{\mathrm{T}} y^*$ and $(C - \mathcal{A}^{\mathrm{T}} y^*)P^* = \lambda^* P^*$, i.e. λ^* is indeed the largest eigenvalue of $C - \mathcal{A}^{\mathrm{T}} y^*$ and it has multiplicity r^*.
- $\mathcal{A}(P^* U^* (P^*)^{\mathrm{T}}) = b$, i.e. $0 \in \partial f(y^*)$.

We say that P^* and U^* satisfying these conditions furnish an *optimality certificate* of y^* for (1).

3. The SB method

We now recall the main idea of bundle methods [14] and more concretely of the SB method [13].

Given a first-order oracle of a nonsmooth convex function f, that is a routine returning the function value and a subgradient of f at a given point y, bundle methods use the subgradient information to form a minorizing model of f. A next candidate y^+ is determined with respect to a current *centre of stability* $\hat{y}$ by minimizing the model augmented by a quadratic term $(t/2)\|y - \hat{y}\|^2$ where t is a *weight* controlling the distance from the candidate to the centre as in a trust region approach. If evaluation of f at the candidate exhibits sufficient decrease, the methods perform a *descent step* by moving the centre to the candidate. Otherwise, in a *null step*, the centre is not modified but the new subgradient information is used to improve the model.

Second-order information can be incorporated in bundle methods by replacing the augmenting term $\|y - \hat{y}\|^2$ by a general quadratic term $\|y - \hat{y}\|^2_{H_t} = \langle y - \hat{y}, H_t(y - \hat{y})\rangle$ with

$$H_t = H + tI \succ 0$$

for some $H \succeq 0$ and $t > 0$. This is often called general scaling and is central to the approach presented here. Therefore we describe the most important steps of the SB algorithm for general scaling, deferring discussion of what choice to use for H until the later sections.

Note that for any $W \in \mathcal{W}_n$ the function

$$f_W(y) := \langle C - \mathcal{A}^{\mathrm{T}} y, W\rangle + b^{\mathrm{T}} y = \langle C, W\rangle + \langle b - \mathcal{A}W, y\rangle$$

is a linear minorant of f. The SB method uses the maximum over a subset $\hat{\mathcal{W}} \subseteq \mathcal{W}_n$ of the minorants to describe a cutting model

$$g_{\hat{\mathcal{W}}}(y) := \max_{W \in \hat{\mathcal{W}}} f_W(y) \le g_{\mathcal{W}_n}(y) = f(y).$$

To simplify the notation in what follows, we focus on the case where $\hat{\mathcal{W}}$ is defined by some $P \in \mathcal{O}_{n,k}$ as

$$\hat{\mathcal{W}} = \{PUP^{\mathrm{T}} : U \in S^k, \operatorname{tr} U = 1, U \succeq 0\} \tag{4}$$

although in practice, it is necessary to consider a slightly more general set

$$\{PUP^{\mathrm{T}} + \alpha \bar{W} : \operatorname{tr} U + \alpha = 1, U \in S^r, U \succeq 0, \alpha \in \mathbb{R}, \alpha \ge 0\},$$

where $\bar{W} \in S^n_+$ is used to 'aggregate' residual information in order to allow for fewer columns in P. A key point of the SB method is that the columns of the matrix P are chosen to be approximate

eigenvectors for the largest eigenvalues of $C - \mathcal{A}^{\mathrm{T}}\hat{y}$ at the current or previous values of $\hat{y}$. Given a *bundle of approximate eigenvectors* P, a *scaling matrix* $H \succeq 0$, a *weight* $t > 0$ and a *centre of stability* $\hat{y}$, the next candidate is

$$y^+ = \operatorname*{argmin}_{y\in\mathbb{R}^m} \left\{ g_{\hat{\mathcal{W}}}(y) + \frac{1}{2}\|y-\hat{y}\|^2_{H_t} \right\} = \operatorname*{argmin}_{y\in\mathbb{R}^m} \max_{W\in\hat{\mathcal{W}}} \left\{ \langle C, W\rangle + \langle b - \mathcal{A}W, y\rangle + \frac{1}{2}\|y-\hat{y}\|^2_{H_t} \right\}. \tag{5}$$

Compactness and convexity of the set $\hat{\mathcal{W}}$ and strong convexity of the augmented function in y ensure the existence of saddle points, so we may exchange min with max. For any given W, the minimizing y is

$$y(W) := \hat{y} - H_t^{-1}(b - \mathcal{A}W), \tag{6}$$

so, substituting this into the right-hand side of (5), we must maximize the dual functional

$$\langle C, W\rangle + \langle b - \mathcal{A}W, \hat{y}\rangle - \tfrac{1}{2}\|b - \mathcal{A}W\|^2_{H_t^{-1}}.$$

For $\hat{\mathcal{W}}$ as in (4), a maximizing $W^+ = PU^+P^{\mathrm{T}}$ is defined by

$$U^+ \in \operatorname{argmax}\{\langle C, PUP^{\mathrm{T}}\rangle + \langle b - \mathcal{A}PUP^{\mathrm{T}}, \hat{y}\rangle - \tfrac{1}{2}\|b - \mathcal{A}PUP^{\mathrm{T}}\|^2_{H_t^{-1}} : \operatorname{tr} U = 1, U \succeq 0\}$$

or equivalently

$$U^+ \in \operatorname{argmax}\{\langle P^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}}\hat{y})P, U\rangle + b^{\mathrm{T}}\hat{y} - \tfrac{1}{2}\|b - \mathcal{A}PUP^{\mathrm{T}}\|^2_{H_t^{-1}} : \operatorname{tr} U = 1, U \succeq 0\}. \tag{7}$$

This convex optimization problem in S^k, with a quadratic objective and a semidefinite constraint, is called a *quadratic SDP*. We assume that k is small enough that it can be solved efficiently by a standard interior-point method. Having determined U^+, the new candidate is given by $y^+ = y(W^+) = y(PU^+P^{\mathrm{T}})$. If the progress predicted by the model value $g_{\hat{\mathcal{W}}}(y^+) = f_{W^+}(y^+)$ is small in relative scale, i.e. if for given $\varepsilon_{\mathrm{opt}} > 0$

$$f(\hat{y}) - f_{W^+}(y^+) \le \frac{\varepsilon_{\mathrm{opt}}}{\max\{1, \operatorname{tr} H/n\}}(|f(\hat{y})| + 1), \tag{8}$$

then the algorithm stops. Here the denominator $\max\{1, \operatorname{tr} H/n\}$ compensates for the influence of H on the step size in (6). Otherwise, f is evaluated at y^+ and actual progress $f(\hat{y}) - f(y^+)$ is compared to the predicted progress $f(\hat{y}) - f_{W^+}(y^+)$. If this ratio is good, say $f(\hat{y}) - f(y^+) > \kappa[f(\hat{y}) - f_{W^+}(y^+)]$ for some $\kappa \in (0,1)$, the method performs a *descent step* by moving its centre of stability to the candidate, that is, setting $\hat{y} \leftarrow y^+$. Otherwise, in a *null step*, the centre of stability is left unchanged and the model $\hat{\mathcal{W}}$ is corrected by updating P. Summarizing, we have the following basic version of the SB algorithm.

ALGORITHM 1 (Spectral bundle method)

Input: $\hat{y} \in \mathbb{R}^m$, $\varepsilon_{\mathrm{opt}} \ge 0$, $\kappa \in (0,1)$, $\bar{\kappa} \in (\kappa, 1)$, $H_t^{\min}$ and $H_t^{\max}$ with $0 \prec H_t^{\min} \preceq H_t^{\max}$.

SB0 (Initialization).
Compute $f(\hat{y})$ and initialize P to contain some approximate eigenvectors for the largest eigenvalues of $C - \mathcal{A}^{\mathrm{T}}\hat{y}$. Initialize $H_t \succ 0$ so that $H_t^{\min} \preceq H_t \preceq H_t^{\max}$.

Iteration: repeat the following steps

SB1 (Candidate Finding).
Compute U^+ and $W^+ = PU^+P^{\mathrm{T}}$ by solving (7) and set $y^+ \leftarrow y(W^+)$ using (6). If

$$f(\hat{y}) - f_{W^+}(y^+) \leq \frac{\varepsilon_{\mathrm{opt}}}{\max\{1, \operatorname{tr} H/n\}}(|f(\hat{y})| + 1),$$

stop.

SB2 (Evaluation and Descent Test).
For $B^+ := C - \mathcal{A}^{\mathrm{T}}y^+$ compute a Ritz vector v, with $\|v\| = 1$, so that at least one of the following cases applies:

SB3a (Null Step).

$$f(\hat{y}) - f_{vv^{\mathrm{T}}}(y^+) \leq \bar{\kappa}[f(\hat{y}) - f_{W^+}(y^+)]$$

In this case, leave the centre of stability $\hat{y}$ unchanged.

SB3b (Descent Step).
Here we assume v to satisfy $f_{vv^{\mathrm{T}}}(y^+) = f(y^+)$; see the remark below.

$$f(\hat{y}) - f_{vv^{\mathrm{T}}}(y^+) > \kappa[f(\hat{y}) - f_{W^+}(y^+)]$$

In this case set $\hat{y} \leftarrow y^+$. The candidate solution becomes the new centre of stability.

SB4 (Update Bundle and Scaling Matrix).
Update the bundle P. Details will be given later, but if a null step was taken (Step SB3a), the update must ensure that $\hat{\mathcal{W}}^+ \supseteq \operatorname{conv}\{W^+, vv^{\mathrm{T}}\}$ and $H_t \preceq H_t^+ \preceq H_t^{\max}$. If a descent step was taken (SB3b), update H_t so that $H_t^{\min} \preceq H_t \preceq H_t^{\max}$.

Note that it is possible that the conditions for a null step and a serious step are both satisfied. This gives us some flexibility in the eigenvalue computation: in particular, as soon as the null step criterion applies, we may terminate the eigenvalue computation and continue with a null step. Only in the case where the null step criterion fails do we need to compute $\lambda_{\max}$ to full precision.

Remark 1 The main work in evaluating f and updating P is the computation of $\lambda_{\max}(C - \mathcal{A}^{\mathrm{T}}y^+)$. In this computation, sparsity or other structural properties of $C - \mathcal{A}^{\mathrm{T}}y^+$ are exploited by an iterative method of Lanczos type. It generates a sequence of *Ritz pairs* consisting of *Ritz vectors* $v_i \in \mathbb{R}^n$, $\|v_i\| = 1$, and corresponding *Ritz values* $v_i^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}}y^+)v_i$ that converge to $\lambda_{\max}(C - \mathcal{A}^{\mathrm{T}}y^+)$ from below. Note that via $W_i = v_iv_i^{\mathrm{T}} \in \mathcal{W}_n$ each Ritz vector generates a linear minorant satisfying $f_{W_i}(y^+) \leq f(y^+)$. As soon as some Ritz value gives rise to a value $f_{W_i}(y^+)$ fulfilling the null step criterion (see Step SB3a above), the Ritz vector v_i provides sufficient information to proceed with a null step of the bundle method and neither the precise value $\lambda_{\max}(C - \mathcal{A}^{\mathrm{T}}y^+)$ nor a corresponding eigenvector needs to be computed. Otherwise the process is continued until the maximum eigenvalue is well approximated together with a corresponding eigenvector; see step SB3b.

To summarize, the evaluation of f in step SB2 results in a matrix $V = (v_1, \ldots)$ of Ritz vectors with $v = v_1$ and associated Ritz values

$$v_1^{\mathrm{T}}B^+v_1 \geq v_2^{\mathrm{T}}B^+v_2 \geq \cdots$$

and $V^{\mathrm{T}}V = I$. Regardless of the null step or descent step decision, the matrix P is then updated. For details we refer to [11,13]. In theory, satisfying the condition $\hat{\mathcal{W}}^+ \supseteq \operatorname{conv}\{W^+, vv^{\mathrm{T}}\}$ during

null steps suffices to ensure $f(\hat{y}) \to \inf f$ over all descent steps by the standard analysis of bundle methods [11].

In order to motivate our second-order enhancements to this algorithm, we first describe in the following section a second-order method to minimize $f(y)$. Then we will explain the changes to Algorithm 1 which are needed to incorporate second-order information.

4. A second-order method to minimize f

It is well known that the maximum eigenvalue function $\lambda_{\max}$ is differentiable around a given matrix $X \in S^n$ if and only if the maximum eigenvalue of X is simple. In this case, the formula for the second derivative of $\lambda_{\max}$ can be found, for example, in [17] and (less explicitly) in [16]. If, on the other hand, the maximum eigenvalue of X has multiplicity $r > 1$, the maximum eigenvalue function is smooth near X only if it is restricted to the submanifold of S^n consisting of matrices whose maximum eigenvalue has multiplicity r. Thus, the key idea for second-order methods is to model the second-order behaviour of the maximum eigenvalue function on such a manifold.

Using this idea, a second-order method to solve (1) (for the case $b = 0$) was given in [20], based on a parameterization used for inverse eigenvalue problems by Friedland et al. [7] and also, less directly, a second-order model for semidefinite constraints due to Fletcher [6]. Overton and Womersley [23] and Shapiro and Fan [26] independently analysed the algorithm of Overton [20] (extended to the case where the matrix depends smoothly, not necessarily affinely, on parameters), establishing its local quadratic convergence under nondegeneracy assumptions. These two approaches to proving quadratic convergence used quite different techniques; a third approach may be found in [4]. Oustry [19] introduced the same quadratic model into a bundle method for (1), proving global and local quadratic convergence under nondegeneracy assumptions, using yet another analytical technique based on $\mathcal{U}$-Lagrangian theory [18].

In order to understand the enhancements of the standard SB method, which are the main topic of this paper, we now provide a brief description of Iteration 4 from Overton and Womersley [23], which we rephrase in the terminology of this paper, and which we call the OW method. Let y^* be a unique minimizer of $f(y)$ and let

$$C - \mathcal{A}^{\mathrm{T}} y^* = Q^* \Lambda^* (Q^*)^{\mathrm{T}}$$

be the spectral decomposition at y^*. Assume that $\lambda_{\max}(C - \mathcal{A}^{\mathrm{T}} y^*)$ has multiplicity r.

One iteration of the OW method can be described as follows. Let $\hat{y}$ be the current iterate, assumed to be close enough to y^* such that $\lambda_{\max}(C - \mathcal{A}^{\mathrm{T}}\hat{y})$ has approximate multiplicity r.

ALGORITHM 2 (Second-order iteration from Overton and Womersley [23])

OW1 Compute the spectral decomposition

$$C - \mathcal{A}^{\mathrm{T}}\hat{y} = \hat{Q}\hat{\Lambda}\hat{Q}^{\mathrm{T}}$$

with $\hat{\lambda}_1 \approx \cdots \approx \hat{\lambda}_r > \hat{\lambda}_{r+1} \geq \cdots \geq \hat{\lambda}_n$ and $\hat{Q}\hat{Q}^{\mathrm{T}} = I$, with $\hat{Q} = [\hat{Q}_1 \ \hat{Q}_2]$, where the columns of $\hat{Q}_1$ are eigenvectors corresponding to $\hat{\lambda}_1, \ldots, \hat{\lambda}_r$ and the columns of $\hat{Q}_2$ are eigenvectors for the remaining eigenvalues.

OW2 Solve the least-squares problem

$$\tilde{U} = \operatorname*{argmin}_{U \in S^r} \{\|b - \mathcal{A}(\hat{Q}_1 U \hat{Q}_1^{\mathrm{T}})\|^2 : \operatorname{tr} U = 1\}.$$

OW3 Define the $m \times m$ second-order matrix $H(\tilde{U})$ by

$$H(\tilde{U})_{ij} = 2\,\mathrm{tr}(A_i\hat{Q}_1\tilde{U}\hat{Q}_1^{\mathrm{T}}A_j\hat{Q}_2(\hat{\lambda}_1 I - D_2)^{-1}\hat{Q}_2^{\mathrm{T}}), \tag{9}$$

where $D_2 = \mathrm{diag}(\hat{\lambda}_{r+1}, \ldots, \hat{\lambda}_n)$.

OW4 Compute the new iterate y from

$$\min_{y\in\mathbb{R}^m, \delta\in\mathbb{R}} \frac{1}{2}\|y - \hat{y}\|^2_{H(\tilde{U})} + b^{\mathrm{T}}y + \delta \quad \text{such that } \delta I = \hat{Q}_1^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}}y)\hat{Q}_1$$

and set $\hat{y} := y$.

In Theorem 7 from Overton and Womersley [23] it is shown that under some regularity assumptions, this iteration converges quadratically to y^*, provided that the starting point $\hat{y}$ is close enough to y^*.

We close this section with two remarks.

Remark 2 The matrix $H(\tilde{U})$ in (9) corresponds to the second-order formula from Overton and Womersley [23], or more precisely the variant $\tilde{W}$ discussed there, and is also the formula used in [20] (for a slightly different problem) and in [19]. Using the vec operator and the Kronecker product, it is equivalent to

$$H(\tilde{U}) = 2\bar{\mathcal{A}}(Q_1 \otimes Q_2)(\tilde{U} \otimes (\hat{\lambda}_1 I - D_2)^{-1})(Q_1 \otimes Q_2)^{\mathrm{T}}\bar{\mathcal{A}}^{\mathrm{T}}, \tag{10}$$

where $\bar{\mathcal{A}}^{\mathrm{T}} = [\mathrm{vec}(A_1), \ldots, \mathrm{vec}(A_m)]$ and $D_2 = \mathrm{diag}(\hat{\lambda}_{r+1}, \ldots, \hat{\lambda}_n)$.

Note, as pointed out in [12], the similarity of the structure of this matrix to that of the system matrix that must be formed in primal–dual interior-point methods for SDP [28], the key difference being that $H(\tilde{U})$ is well defined in the limit as $y \to y^*$ since the quantities being inverted in the central factor do *not* converge to zero as long as the multiplicity r is estimated correctly. Note also, however, that $H(\tilde{U})$ will be singular whenever the rank of $\tilde{U}$ is smaller than r.

Remark 3 Below, we relax step OW4 as follows. The equation

$$\delta I = \hat{Q}_1^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}}y)\hat{Q}_1$$

imposes, to first-order, an eigenvalue of $C - \mathcal{A}^{\mathrm{T}}y$ with multiplicity r and value δ. We change it to the semidefinite constraint

$$\delta I \succeq \hat{Q}_1^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}}y)\hat{Q}_1$$

requiring that δ is at least as large as the largest eigenvalue of the matrix on the right-hand side. Making this substitution, step OW4 becomes

$$\min_{y\in\mathbb{R}^m} \frac{1}{2}\|y - \hat{y}\|^2_{H(\tilde{U})} + b^{\mathrm{T}}y + \lambda_{\max}(\hat{Q}_1^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}}y)\hat{Q}_1),$$

which can be rewritten as

$$\min_{y\in\mathbb{R}^m} \max_{U\in\mathcal{S}^k} \left\{ \langle \hat{Q}_1^{\mathrm{T}}(C - \mathcal{A}^{\mathrm{T}}y)\hat{Q}_1, U\rangle + b^{\mathrm{T}}y + \frac{1}{2}\|y - \hat{y}\|^2_{H(\tilde{U})} : \mathrm{tr}(U) = 1, U \succeq 0 \right\}. \tag{11}$$

This is now a problem of the form (5) with $t = 0$ and $\hat{Q}_1$ taking the role of the bundle P.

Step OW2 is based on the assumption that $\hat{y}$ is close enough to the optimum y^*, so that r, the multiplicity of the largest eigenvalue, is known *a priori* and the minimizing matrix $\tilde{U}$ is

positive definite by continuity (given a regularity assumption). In (11) we impose the semidefinite constraint on U explicitly, following the philosphy of the SB method and allowing us to change the multiplicity estimate r for the largest eigenvalue dynamically, as described below.

The full second-order iteration involves several operations which are acceptable only for small problems. A full spectral decomposition, as required in OW1, limits the size n of the primal space to $n \approx 1000$. The second-order matrix $H(\tilde{U})$ of order $m \times m$ is generically dense, even if the A_i are sparse. This puts a limit on m, as is the case for interior-point methods. In the next section we describe an extension of the SB method that incorporates second-order information efficiently.

5. Incorporating second-order information into the SB method

In this section we give several ways to define the scaling matrix H_t in the SB method (Algorithm 1) using second-order information, inspired by the OW method (Algorithm 2).

The SB method is driven by the $n \times k$ bundle matrix P, which is used in step SB1 to solve (7) yielding U^+ and the new candidate solution $y^+ = y(PU^+P^{\mathrm{T}})$. In constrast the OW iterations are based on the spectral decomposition of $C - \mathcal{A}^{\mathrm{T}}\hat{y}$, given by the orthogonal eigenvector matrix $\hat{Q}$. In order to incorporate second-order information into the SB method, we aim at using $\hat{Q}$ to define the scaling matrix H_t. To maintain computational efficiency we also would like to avoid a full factorization to get $\hat{Q}$. Thus we extend the original SB method by including, in addition to the bundle matrix P, a matrix Q of order $n \times \ell$, where $k \le \ell \le n$, which contains approximate eigenvectors of $C - \mathcal{A}^{\mathrm{T}}y$. The matrix Q will play the role of a truncated approximation to $\hat{Q}$ in the OW method. The modified SB method is therefore driven by the bundle P and the matrix Q of approximate eigenvectors of $C - \mathcal{A}^{\mathrm{T}}y$ at the candidate solution y. Both P and Q will be updated in each iteration to yield P^+ and Q^+.

We now provide an overview of the modifications to the SB method which allow us to include second-order information in the scaling matrix H_t. Mathematical justifications and implementation details will be described in the following subsections. At the beginning of each iteration of the modified spectral bundle (MSB) method in Algorithm 3 below we have the following data:
$\hat{y}$ ... current centre of stability with objective function value $f(\hat{y})$,
P ... $n \times k$ bundle matrix with $P^{\mathrm{T}}P = I$,
Q ... $n \times \ell$ eigenvector estimates of $C - \mathcal{A}^{\mathrm{T}}y$ with $Q^{\mathrm{T}}Q = I$ at the current candidate solution y,
$H_t \succ 0$ ... scaling matrix.
At the beginning we set $y \leftarrow \hat{y}, Q \leftarrow P$.

ALGORITHM 3 (MSB method)

MSB1 (Determine a new candidate solution y^+).
Compute U^+ as in (7). Set $W^+ = PU^+P^{\mathrm{T}}$ and $y^+ = \hat{y} - H_t^{-1}(b - \mathcal{A}W^+)$; see (6).

MSB2 (Evaluation of $f(y^+)$).
For $B^+ := C - \mathcal{A}^{\mathrm{T}}y^+$ use a Lanczos type method to generate Ritz vectors $V = (v_1, \ldots)$ and Ritz values $v_1^{\mathrm{T}}B^+v_1 \ge v_2^{\mathrm{T}}B^+v_2 \ge \ldots$.

MSB3 (Null step or descent step).
Decide on whether to take a null step or a descent step as in Algorithm 1.

MSB4 (Update P, Q and H_t).

MSB4a (Estimate eigenvectors $\bar{q}_i$ and eigenvalues $\bar{\lambda}_i$ of B^+).
Let $\bar{V}$ be an orthonormal basis of $[Q\ V]$. Compute an eigenvalue decomposition of $\bar{V}^{\mathrm{T}}B^+\bar{V} =$

$S\bar{\Lambda}S^{\mathrm{T}}$ with $S^{\mathrm{T}}S = I$ and $\bar{\Lambda} = \mathrm{diag}(\bar{\lambda}_i)$ with $\bar{\lambda}_1 \geq \bar{\lambda}_2 \geq \cdots$ and set $\bar{Q} = (\bar{q}_1, \ldots) = \bar{V}S$. Thus $\bar{\lambda}_i = \bar{q}_i^{\mathrm{T}} B^+ \bar{q}_i$. For details see Section 5.1.

MSB4b (Multiplicity estimate).
Determine an estimate for the multiplicity r of $\lambda_{\max}(B^+)$. Details are given in Section 5.2.

MSB4c (Update P and Q).
Use P and $\bar{Q}$ to get an update P^+ for P and use $\bar{Q}$ to get an update Q^+ for Q. Set $P \leftarrow P^+, Q \leftarrow Q^+$. Details are given in Section 5.3.

MSB4d (Minimum norm approximate subgradient).
Partition $Q = [Q_1\ Q_2]$ where Q_1 is $n \times r$. Solve

$$\tilde{U} = \operatorname*{argmin}_{U \in S^r} \{\|b - \mathcal{A}(Q_1 U Q_1^{\mathrm{T}})\|^2 : \mathrm{tr}\, U = 1, U \succeq 0\}. \tag{12}$$

This corresponds to step OW2 of Algorithm 2, except that a semidefinite constraint is included in (12); see Remark 3.

MSB4e (Update the scaling matrix H_t).
Set H_t to an approximation of the second-order matrix $H(\tilde{U})$ defined in (10) as explained below in Section 5.4. This corresponds to step OW3, as explained in Remark 2. The matrix H_t is then used in the next iteration in step MSB1; see also Remark 3.

Summarizing, the new candidate solution y^+ is determined through the bundle P and the scaling matrix H_t which mimicks the second-order term $H(\tilde{U})$ in step OW4; compare in particular (11) and (5). A quadratic SDP of the form (7) has to be solved. In contrast to Algorithm 2 we avoid a full spectral decomposition of B^+, approximating only the largest eigenvalues of B^+; see MSB4a. The update of H_t requires the solution of an additional quadratic SDP to get the matrix $\tilde{U}$ which forms the basis for the second-order matrix $H(\tilde{U})$; see (10), and the variants described in Section 5.4 below.

5.1 *Approximate eigenvalues and eigenvectors*

The evaluation of f at y^+ is done approximately, as explained in Section 3, using a Lanczos-type algorithm. It produces an approximation to $\lambda_{\max}(B^+)$ together with a set of Ritz vectors, which are collected in the matrix V, where $V^{\mathrm{T}}V = I$. We combine the new Ritz vectors and the matrix Q into a new matrix $\bar{Q}$ as follows. First, let $\bar{V}$ form an orthonormal basis of $[Q\ V]$. We determine the eigenvalue decomposition of the projected matrix $\bar{V}^{\mathrm{T}} B^+ \bar{V}$, given as

$$S\bar{\Lambda}S^{\mathrm{T}} = \bar{V}^{\mathrm{T}} B^+ \bar{V},$$

with $S^{\mathrm{T}}S = I$, $\bar{\Lambda} = \mathrm{diag}(\bar{\lambda}_i)$ and $\bar{\lambda}_1 \geq \bar{\lambda}_2 \geq \cdots$, and set $\bar{Q} = \bar{V}S$. Thus $\bar{\Lambda} = \bar{Q}^{\mathrm{T}} B^+ \bar{Q}$. Note that, if the iterative eigenvalue solver returned a true eigenvector for $\lambda_{\max}(B^+)$, then $\bar{\lambda}_1 = \lambda_{\max}(B^+)$. In any case, by continuity of the eigenspaces, the largest $\{\bar{\lambda}_i\}$ will become highly accurate estimates of the largest eigenvalues of B^+ whenever y converges. Thus, these values are employed as estimates for $\lambda_i(B^+)$.

5.2 *Estimating the eigenvalue multiplicity*

A key challenge is to devise a stable approach for determining a good estimate of the multiplicity r of the maximum eigenvalue. In theory (see [12]), once the algorithm is close enough to an optimal solution y^*, there will be a gap of significant relative size between λ_r and λ_{r+1}.

In practice, however, even quite complex schemes based on observing the ratio $(\lambda_r - \lambda_{r+1})/(\lambda_1 - \lambda_{r+1})$ fail quite regularly. For example, it might happen that the multiplicity estimate r stabilizes at a certain value for several iterations and then suddenly drops to $r = 1$. Such misclassifications have dire consequences for the performance of the bundle method, increasing the number of null steps dramatically. Instead, we estimate the multiplicity using the following two ingredients.

As a first estimate, r should at least embrace all eigenvalues of B^+ within a relative precision, say τ, of $\lambda_{\max}$ (We used $\tau = 10^{-6}$.). The resulting lower bound based on the $\bar{\lambda}_i$ is

$$\underline{r} = \max\{j \in \{1, \ldots, k\} : \bar{\lambda}_1 - \bar{\lambda}_i \leq \tau(|\bar{\lambda}_1| + 1)\ \forall i = 1, \ldots, j\}.$$

Secondly, we use the optimizer U^+ of (7) in step SB1 based on the following intuition. Once we are close enough to an optimal solution y^*, the matrix $W^+ = PU^+P^{\mathrm{T}}$ approaches an optimal solution of (2), so the columns of P approach the eigenspace of $\lambda_{\max}(C - \mathcal{A}^{\mathrm{T}}y^*)$. The rank of W^+, or equivalently U^+, therefore serves as another estimate for the multiplicity r. As long as k, the number of columns in P, is at least r^*, the actual multiplicity of $\lambda_{\max}(C - \mathcal{A}^{\mathrm{T}}y^*)$, this provides another reasonable estimate for r. In order to identify the nonzero eigenvalues of U^+ we make use of the idea of Tapia indicators [5] as follows. In solving (7) by an interior-point method, let U' be the last iterate before the algorithm terminates with the solution U^+ and denote the corresponding eigenvalues by $\lambda_i(U')$ and $\lambda_i(U^+)$ sorted nonincreasingly for $i = 1, \ldots, k$. Generically, the 'active' eigenvalues converge to some fixed positive value while inactive ones converge to zero with the same speed as the barrier parameter, so the criterion estimates the decrease from $\lambda_i(U')$ to $\lambda_i(U^+)$. In our implementation the barrier parameter is typically reduced by some value in $(0, 0.3]$, leading to the rather simple estimate

$$\bar{r} = \max\{j \in \{1, \ldots, k\} : \lambda_i(U^+) \geq 0.8\,\lambda_i(U')\ \forall i = 1, \ldots, j\}.$$

The final multiplicity estimate is then

$$r = \max\{\bar{r}, \underline{r}\}.$$

5.3 *The update mechanism for P and Q in step MSB4c*

We recall that in the SB method the columns of P should contain approximate eigenvectors for the largest eigenvalue of $C - \mathcal{A}^{\mathrm{T}}y$ at the current and possibly previous iterates. Thus, to update P we use the old bundle P and also the eigenvector estimates for B^+ in $\bar{Q}$. Here are the details for the update of the $n \times k$ bundle matrix P to a matrix P^+.

(1) First we include $\min(r + 3, k)$ eigenvectors w_i of PU^+P^{T} corresponding to its largest eigenvalues.
(2) Second we also consider including additional eigenvectors w_i of PU^+P^{T}, by investigating their contribution to the second-order matrix H. For inclusion of the eigenvector w_i in P^+, we consider as an indicator for the importance of the ith eigenvector to the model the contribution of the corresponding $\bar{q}_i$ to the trace of H if $\bar{q}_i$ appears as a column of $\bar{Q}_2$,

$$\rho_i = \sum_{h=1}^{m} \bar{q}_i^{\mathrm{T}} A_h P U^+ P^{\mathrm{T}} A_h \bar{q}_i (\bar{\lambda}_1 - \bar{\lambda}_i)^{-1}.$$

We include w_i in P^+ if ρ_i is large enough. Assuming prescaled $\|A_h\| = 1$ for all $h = 1, \ldots, m$ we include w_i in P^+ if $\rho_i > m$, i.e. if the average contribution to each diagonal element of H is at least one.

(3) Finally, we include 5 columns of $\bar{Q}$ corresponding to the largest eigenvalue estimates $\bar{\lambda}_i$.

The total number of columns included from P in (1) and (2) is denoted k_P and will be used in Section 5.4.3, and the total number of columns of P and Q added from (1), (2) and (3) is denoted k^+. Since the columns from Q are not orthogonal to the ones from P, the resulting set must be orthogonalized.

A straightforward update for Q would simply be to use $\bar{Q}$. Since we also have computational efficiency in mind, we select only a subset of the columns of $\bar{Q}$, based on the following intuition. In view of the definition of $H(\tilde{U})$, it seems reasonable to discard eigenvectors corresponding to eigenvalues significantly smaller than the largest.

The update of Q will be denoted Q^+, and is thus formed by taking some, but not all, vectors of $\bar{Q}$ as follows. The updated matrix Q^+ is chosen to contain the first k^+ and at least n_a further columns of the matrix $\bar{Q}$ for some adaptive parameter $n_a \in \mathbb{N}$ described in Section 5.4.2. Furthermore, we drop all indices $i > k^+ + n_a$ with

$$\bar{\lambda}_i < \bar{\lambda}_1 - \min\{10^{-2}(1 + |\bar{\lambda}_1|), 10(\bar{\lambda}_1 - \bar{\lambda}_{r+1})\}.$$

Of the remaining ones we keep the first few with contribution $\rho_i > m/10$ to H. Because the computation of the ρ_i is quite involved, this is restricted to the update on descent steps.

5.4 *Four choices of H inspired by the second-order method*

We now describe four variants for choosing H, the first one being the full Newton method which is then, for the sake of computational efficiency, approximated and simplified, the final simplification being a diagonal scaling heuristic.

The SB method is started without scaling, i.e. initially $H = 0$ so $H_t = tI$ with t being updated as described in [11]. From the beginning, however, the bundle update scheme of Section 5.3 is employed, so that all required information is available once scaling is started. Scaling is used once a relative precision of 10^{-2} has been reached, i.e. when $f(\hat{y}) - f_{W^+}(y^+) \leq 10^{-2}(|f(\hat{y})| + 1)$. From then on, a new scaling matrix H is formed at every descent step and H_t is set to $H + tI$. During null steps, however, $H_t \preceq H_t^+$ is required to ensure convergence. To meet this, H is not altered during null steps.

5.4.1 *The full second-order model*

This variant implements the full second-order model as explained in Section 4. We compute the full spectral decomposition $C - \mathcal{A}^{\mathrm{T}}\hat{y} = \hat{Q}\hat{\Lambda}\hat{Q}^{\mathrm{T}}$. We also compute $\tilde{U}$ as well as $H = H(\tilde{U})$ as defined in (12) and (9). Also, the bundle P is replaced by $P = \hat{Q}_1$, so in fact the bundle update scheme of the previous section would only be needed for null steps.

Even though $H(\tilde{U})$ may be singular, $H_t = H + tI$ is positive definite because $t > 0$ provides the necessary regularization. Still, a large t might be appropriate when $H = 0$ but may hinder progress in the presence of a full Newton matrix H. Therefore, when H is nonzero for the first time, we reinitialize t by the following heuristic. Let $\tilde{t}$ denote the minimal value of t over all iterations up to this point, then $t \leftarrow \max\{10^{-3} \cdot \min_i H(\tilde{U})_{ii}, \min\{10^{-3}, \tilde{t}/10\}\}$. During subsequent null steps the heuristic for choosing t as described in [11] is used but the t of the next descent step is not allowed to exceed 10 times the value of the previous t.

The eigenvalue decomposition requires $O(n^3)$ operations and takes roughly 5 times the computation time of a dense Cholesky factorization. For large structured problems this exceeds the work required for computing an extremal eigenvalue via Lanczos methods significantly. The cost of computing $H(\tilde{U})$ is comparable in cost to forming the system matrix in semidefinite interior-point

methods and is typically by far the most expensive step unless m is small or the A_i have very special structure. The Cholesky factorization of $H(\tilde{U})$ is also required, to define the coefficient matrices in the quadratic SDP which must be solved in (11). This amounts to $O(m^3)$ arithmetic operations. Because a new $H(\tilde{U})$ and its factorization must be computed for each descent step, the iterations of interior-point methods can be expected to be at least as fast. Thus, from a computational perspective the bundle method with full second-order scaling is not suitable for large-scale semidefinite optimization and cannot be expected to be able to compete with interior-point methods even for medium scale problems.

5.4.2 *A low-rank variant of the second-order model*

In practice, as explained in Section 3, it is impractical to compute all eigenvalues of $B^+ = C - \mathcal{A}^{\mathrm{T}}y^+$, so we assume in this subsection that only the matrix $Q = (q_1, \ldots, q_\ell)$ (with $\ell < n$) together with the corresponding approximate eigenvalues $\bar{\lambda}_i = q_i^{\mathrm{T}} B^+ q_i$ is available. This eliminates the possibility of using the full second-order model. However, partitioning $Q = [Q_1, Q_2]$ with Q_1 consisting of the first r columns and Q_2 consisting of the remaining columns and splitting $\Lambda = \mathrm{diag}(\bar{\lambda})$ into D_1 and D_2 correspondingly, suggests how to replace the full second-order model by a low-rank approximation. Using Q_1, the matrix $\tilde{U}$ in (12) can be computed as before. The computation of $H(\tilde{U})$ in (10) now reads

$$H(\tilde{U}) = 2\bar{\mathcal{A}}(Q_1 \otimes Q_2)(\tilde{U} \otimes (\bar{\lambda}_1 I - D_2)^{-1})(Q_1 \otimes Q_2)^{\mathrm{T}} \bar{\mathcal{A}}^{\mathrm{T}},$$

which is a low-rank approximation because Q has less than n columns. Even if the dimension of D_2 is kept small, it is tempting to reduce the rank further by eliminating small eigenvalues of $\tilde{U} \otimes (\bar{\lambda}_1 I - D_2)^{-1}$. These are $\lambda_i(\tilde{U})/(\bar{\lambda}_1 - \bar{\lambda}_j)$ which may be small in comparison to the largest choice with $i = 1$ and $j = r + 1$.

Computationally, however, the approximation seems to profit more from first computing the QR decomposition of

$$\bar{\mathcal{A}}(Q_1 \otimes Q_2) =: Q_{\bar{\mathcal{A}}} R_{\bar{\mathcal{A}}} \tag{13}$$

and then computing the spectral factorization

$$2R_{\bar{\mathcal{A}}}(\tilde{U} \otimes (\bar{\lambda}_1 I - D_2)^{-1}) R_{\bar{\mathcal{A}}}^{\mathrm{T}} = Q'_{\bar{H}} \Lambda_{\bar{H}} Q'^{\mathrm{T}}_{\bar{H}}.$$

Instead of using

$$\bar{H} = Q_{\bar{H}} \Lambda_{\bar{H}} Q_{\bar{H}}^{\mathrm{T}} \quad \text{with } Q_{\bar{H}} := Q_{\bar{\mathcal{A}}} Q'_{\bar{H}},$$

we use a low rank approximation obtained by deleting from $\Lambda_{\bar{H}}$ all eigenvalues $(\Lambda_{\bar{H}})_{ii} < \tilde{\delta}\lambda_{\max}(\bar{H})$ for a parameter $\tilde{\delta} \in (0, 1)$ (we use $\tilde{\delta} = 10^{-6}$). Calling the corresponding submatrices $\Lambda_{\tilde{H}}$ and $Q_{\tilde{H}}$ we finally have the approximation

$$H(\tilde{U}) \approx \tilde{H} = Q_{\tilde{H}} \Lambda_{\tilde{H}} Q_{\tilde{H}}^{\mathrm{T}}.$$

In this scaling approach, the bundle is updated as described in Section 5.3 also after descent steps.

Note that the dimension of D_2 and the number of columns in Q_2 depend directly on the number of columns ℓ provided by this update. Due to the high computational cost involved in a large ℓ, in particular in view of the QR decomposition (13), the rules for including more columns in the update Q^+ are rather stringent but work sufficiently well initially. In some cases, in particular if higher accuracy levels are required, the rules are too restrictive leading to a poor scaling matrix $\tilde{H}$ which leads to a large number of null steps between consecutive descent steps. In such cases the lower bound n_a on the number of columns in Q_2 is increased by the following heuristic rule.

Initially $n_a = 5$, and whenever at least $h > 20$ null steps proceed a descent step, n_a is increased by $\lfloor h/20 \rfloor$ as long as n_a does not exceed $n/10$.

Regarding the regularization in $H_t = \tilde{H} + tI$ by $t > 0$ for low-rank scaling, we first reinitialize t to the minimal value of t over all previous iterations before $\tilde{H}$ is computed for the first time. During subsequent iterations the heuristic of Helmberg and Kiwiel [11] for choosing t is used, but to increase stability in view of a less accurate H, the t of the next descent step is not allowed to exceed the previous descent step value by a factor of $\frac{4}{3}$ or to be reduced by more than a factor of $\frac{2}{3}$.

Because $\tilde{H}$ is already given by its eigenvalue decomposition, the inverse of H_t could be applied explicitly in a numerically stable way via a representation of $Q_{\tilde{H}}$ by Householder vectors. In practice, however, exploiting the low rank structure by a Sherman-Morrison variant proved computationally more efficient on our test instances.

5.4.3 *A low-rank approximation with PU^+P^{T} replacing $Q_1\tilde{U}Q_1^{\mathrm{T}}$*

For large r, the quadratic SDPs to determine U^+ in step SB1 and to compute $\tilde{U}$ in (12) are computationally quite involved. We will now argue that the solution U^+ of SB1 can be used to construct increasingly accurate solutions to (12) without actually solving (12).

While [22] stresses the importance of computing $\tilde{U}$, it can be shown as in [8] that all cluster points of $Q_1\tilde{U}Q_1^{\mathrm{T}}$ as well as of those PU^+P^{T} that result in descent steps are optimal solutions to the primal program (2). Indeed, because $b - \mathcal{A}(Q_1\tilde{U}Q_1)$ as well as $b - \mathcal{A}(PU^+P)$ goes to zero, all cluster points are feasible for (2). Therefore optimality for (2) follows from complementarity as Q_1 and P converge to the active subspace of optimal solutions of (3). If the primal optimal solution is unique rewriting (10) allows us to conclude

$$\begin{aligned} H(\tilde{U}) =& 2\bar{\mathcal{A}}[(Q_1\tilde{U}Q_1^{\mathrm{T}}) \otimes (Q_2(\bar{\lambda}_1 I - D_2)^{-1}Q_2^{\mathrm{T}})]\bar{\mathcal{A}}^{\mathrm{T}} \\ \approx& 2\bar{\mathcal{A}}[(PU^+P^{\mathrm{T}}) \otimes (Q_2(\bar{\lambda}_1 I - D_2)^{-1}Q_2^{\mathrm{T}})]\bar{\mathcal{A}}^{\mathrm{T}} \end{aligned}$$

once $\|Q_1\tilde{U}Q_1^{\mathrm{T}} - PU^+P^{\mathrm{T}}\|$ is small enough. Instead of solving (12) it might therefore suffice to approximate $Q_1\tilde{U}Q_1^{\mathrm{T}}$ by PU^+P^{T}. Recall that we include k_P columns from P in P^+, see Section 5.3.

In this variant we therefore replace $Q_1\tilde{U}Q_1^{\mathrm{T}}$ by $P(\sum_{i=1}^{k_P} \lambda_i(U^+)u_iu_i^{\mathrm{T}})P^{\mathrm{T}}$, where the u_i denote the eigenvectors for $\lambda_i(U^+)$. Equivalently, Q_1 is replaced by $\tilde{Q}_1 = P \cdot [u_1, \ldots, u_{k_P}]$. The matrix Q_2 is formed from Q by extracting the columns $k_P + 1$ through ℓ as the gain of including in H second-order information with respect to directions already contained in the bundle P seems to be negligible: the nonpolyhedral bundle model P gives this information already.

This variant basically makes no use of the first k_P columns of Q. Otherwise, we proceed exactly as in Section 5.4.2.

5.4.4 *Using the diagonal of the low-rank approximation*

In terms of computational efficiency diagonal scaling has many advantages even compared to the low-rank approach. Indeed, the diagonal of a low rank approximation can be computed directly without forming the matrix $\bar{\mathcal{A}}(Q_1 \otimes Q_2)$, so the need for the computationally involved QR-factorization (13) is eliminated. Also, the coefficients of the quadratic SDP (7) can be determined much faster. Furthermore, a diagonal H allows us to employ the highly efficient approach of Helmberg and Kiwiel [11] for implementing box constraints on y. Because of these advantages we decided to also test diagonal scaling based on the diagonal of the low rank approximation. In particular, using the notation of Section 5.4.3, the diagonal element for $i \in \{1, \ldots, m\}$

is computed by

$$H_{ii} = 2\mathrm{tr}\left(A_i P\left(\sum_{i=1}^{k_P} \lambda_i(U^+) u_i u_i^{\mathrm{T}}\right) P^{\mathrm{T}} A_i Q_2 (\bar{\lambda}_1 I - D_2)^{-1} Q_2^{\mathrm{T}}\right),$$

where we now use all eigenvalues and eigenvectors of U^+ that are kept in the bundle in the hope of improving the quality of the approximation PU^+P^{T} of $Q_1\tilde{U}Q_1$. The choice of all other parameters is identical to that of the previous subsection. Surprisingly, the same scheme for choosing t again seems to produce very good results.

5.5 *Computational limitations with a large bundle*

Let us look more closely at the linear algebra involved in solving the quadratic SDPs (7). The primal objective may be expressed as

$$d + c_P^{\mathrm{T}} u - \tfrac{1}{2} u^{\mathrm{T}} Q_P u,$$

where

$$d = b^{\mathrm{T}}\hat{y} - \tfrac{1}{2} b^{\mathrm{T}} H_t^{-1} b, \quad c_P = A_P H_t^{-1} b + \mathrm{vec}(P^{\mathrm{T}} C P - P^{\mathrm{T}} \mathcal{A}^{\mathrm{T}} \hat{y} P), \quad Q_P = A_P^T H_t^{-1} A_P,$$

and

$$u = \mathrm{vec}(U), \quad A_P = [(\mathrm{vec}(P^{\mathrm{T}} A_i P))^{\mathrm{T}}]_{i=1,\ldots,m}.$$

In order for the scaled variants of the SB method to be effective it is important that k, the number of columns of P, be at least r^*, the optimal multiplicity. However, this means that each step of the interior-point method solving the quadratic SDPs involves factorizing a positive-definite matrix of order $k(k+1)/2$ (recall that the original m dual variables have been eliminated in (7) by using (6)). For m constraints a reasonable estimate of the dimension of the optimal subspace is $r^* \approx \sqrt{m}$, because there always exists an optimal primal solution whose rank is bounded by this number [1,24]. Solving the primal–dual KKT-system of a standard primal–dual interior-point method for semidefinite programming requires factorizing a matrix of order m. So, if the cost of one interior-point iteration is dominated by this factorization and if $r^* \approx \sqrt{m}$, solving one quadratic SDP is almost as expensive as solving the original problem by an interior point method. In our computational experiments the update scheme for P of Section 5.3 leads to a moderate increase of k towards r^* over time, so that the fast initial progress of bundle methods is preserved, i.e. solutions of moderate accuracy are obtained significantly faster than by interior-point methods. However, good progress at higher accuracy levels seems to require $k \geq r^*$.

Therefore the scope of problems where the proposed scaled versions of the SB method may indeed outperform interior-point methods in computing rather accurate solutions is restricted to problems where either the optimal multiplicity r^* is known to be small or where m is small and the cost of interior-point methods is dominated by the cost of factorizing the primal and dual matrix variables. This can also be observed for our test instances in the next section.

6. Numerical results

The scaling methods were implemented within the ConicBundle (CB) callable library [10] in C++ and tested on Intel(R) Core(TM) i7 CPU 920 machines with 8 MB cache and 12 GB RAM under openSUSE Linux 11.1 (x86_64) in single processor mode. As test instances we generated

several general random sparse SDP problems satisfying the constant trace property as well as semidefinite relaxations of max-cut problems corresponding to Ising spin glasses [27] on three dimensional toroidal grids with edge weights chosen randomly from $\{-1, 1\}$.

The general random sparse instances were generated for given n (order of X and Z), m (number of constraint matrices A_i), p (number of nonzeros per row) and different seeds for the random number generators. Starting with the weighted adjacency matrix A of a connected random graph with an expected number of np edges and edge weights uniformly distributed in $[-\frac{1}{2}, \frac{1}{2}]$, and a uniform random vector $d_Z \in [0, 10]^n + \mathbf{1}$, a dual slack matrix Z is set to $Z = \mathrm{Diag}(d_Z + A\mathbf{1}) - A$ (Z is not enforced to be positive definite). A primal matrix X is formed by $X = EE^{\mathrm{T}} + \mathrm{Diag}(d_X)$ for a uniform random vector $d_x \in [0, 10]^n + \mathbf{1}$, where each element of $E \in \mathbb{R}^{n \times \lceil n/4 \rceil}$ is drawn from the standard normal distribution. This X determines the right-hand side $b_0 = \mathrm{tr}\, X$ of the trace constraint $A_0 = I$ in $\langle A_0, X\rangle = b_0$. For each A_i, $i \in \{1, \ldots, m\}$, a principal submatrix is selected consisting of p distinct indices from $1, \ldots, n$ uniformly at random; its elements are chosen uniformly at random from $-100, \ldots, 100$; the A_i are then normalized to Frobenius norm 1, so that there should be no obvious improvement by pure diagonal scaling. The right-hand side is set to $b = \mathcal{A}X$ ensuring primal feasibility. Finally a vector $y \in \mathbb{R}^m$ is drawn from the standard normal distribution and the cost matrix is set to $C = Z + \mathcal{A}^{\mathrm{T}} y$. Dual feasibility, a duality gap of zero and primal attainment are guaranteed by the trace constraint. The computational results indicate that all problems have dual optimal solutions.

For comparison the problems were also solved with SDPT3-4.0 beta [30] and the old version of the SB code SB described in [9]. SDPT3 is a primal dual interior point package which provides special support for sparsity and solves the Newton system by a preconditioned conjugate gradient approach achieving excellent results also for rather large-scale instances.

To illustrate the results, Figures 3–9 give performance profiles in the style suggested in [3] for comparing the cumulative number of problems that have been solved by each method to precision 10^{-4} (left) and 10^{-6} (right), respectively, within the time given on the abscissa. The methods are

- the conic bundle code (CB-ns) with the bundle update of Section 5.3 but no scaling ($H_t = tI$),
- the full Newton version (CB-fN) described in Section 5.4.1,
- the low rank approximation of Newton (CB-lrN) of Section 5.4.2,
- its approximated variant (CB-alrN) of Section 5.4.3,
- the diagonal version (CB-diag) employing the diagonal scaling heuristic of Section 5.4.4,
- the interior-point code (SDPT3),
- and the old SB code .

In order to circumvent, in these comparisons, the inherent difficulties of bundle methods to terminate precisely at a desired precision on the basis of the rather weak stopping criterion (8), we let all codes solve the problems to a relative precision $\varepsilon_{\mathrm{opt}} = 10^{-8}$ and then use the best objective value f^* over all codes as a reference value in order to determine afterwards for each code the computation time needed until the first descent step or iterate satisfies $f(y) - f^* \le \varepsilon(|f^*| + 1)$ for $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-6}$, respectively. For $\varepsilon = 10^{-6}$ more detailed information is given in Tables 1–6. These list, for each method and a canonical grouping of the instances, mean and variance of computation time in seconds, the number of descent steps and the total number of function evaluations/iterations.

Before discussing these statistics, it is worth taking a detailed look at the relative performance of codes (CB-fn)–(CB-diag) on a rather typical example with $n = 500$ and $m = 500$ as provided in Figures 1 and 2. Figure 1 displays the evolution of the relative gap $(f(\hat{y}) - f^*)/(|f^*| + 1)$ versus bundle iterations and CPU-seconds, respectively. For the full Newton version (CB-fn), second-order convergence may be observed around iteration 60, but the computation time of each single iteration is excessive in comparison to the other variants. There is no indication of superlinear

Table 1. Small random SDPs: average and variance of computation time rounded to seconds for reaching a relative precision of 10^{-6} over 15 instances per row (5 for each $p \in \{3, 5, 7\}$).

n	m	CB-ns	CB-fN	CB-lrN	CB-alrN	CB-diag	SDPT3	SB
100	100	0.6 (0.246)	0.3 (0.132)	0.4 (0.109)	0.4 (0.127)	0.5 (0.473)	0.6 (0.424)	1.9* (1.82)
300	100	2.3 (1.07)	1.9 (0.679)	1.4 (0.58)	1.5 (0.742)	1.2 (0.602)	1.3 (0.313)	2.5 (1.22)
500	100	7.0 (3.54)	7.7 (4.01)	4.2 (2.67)	4.3 (3)	3.8 (4.22)	3.7 (0.515)	7.2 (5.78)
100	500	6.1 (1.7)	6.1 (3.09)	8.2 (1.97)	6.1 (1.36)	5.6 (1.99)	2.7 (1.53)	1319* (1.32×10^3)
300	500	9 (4.95)	22 (24)	9 (2.5)	8 (2.43)	5 (1.03)	6 (4.29)	12 (4.01)
500	500	14 (7.24)	33 (13.8)	11 (3.41)	10 (3.15)	7 (1.96)	9 (4.91)	16 (9.59)
100	1000	47 (21)	94 (47.8)	61 (19.9)	46 (15.7)	45 (21.7)	7 (3.34)	2794* (1.04×10^3)
300	1000	26 (6.93)	90 (64.3)	35 (5.96)	27 (5.02)	19 (2.53)	19 (16.1)	383* (473)
500	1000	32 (19.7)	146 (71.7)	36 (14.1)	30 (12.6)	19 (2.97)	25 (17.9)	300* (521)

Note: *Not all instances achieved the required precision.

Table 2. Small random SDPs: average and variance of the number of descent steps for reaching a relative precision of 10^{-6} over 15 instances per row (5 for each $p \in \{3, 5, 7\}$).

n	m	CB-ns	CB-fN	CB-lrN	CB-alrN	CB-diag	SDPT3	SB
100	100	37 (6.11)	20 (3.44)	33 (6.86)	33 (6.09)	38 (21.3)	11 (0.573)	43* (10.4)
300	100	43 (5.96)	22 (4.7)	38 (8.5)	39 (9.86)	37 (8.65)	13 (0.49)	53 (10.2)
500	100	58 (12.7)	27 (6.67)	50 (11.1)	51 (11.2)	52 (20)	14 (0.611)	69 (25.1)
100	500	42 (5.44)	27 (3.07)	42 (5.56)	42 (5.3)	50 (15.4)	11 (0.499)	48* (3.35)
300	500	59 (11.1)	34 (5.04)	56 (10.2)	57 (11.3)	57 (11.6)	13 (0.806)	54 (6.3)
500	500	66 (11.5)	37 (5.23)	62 (12.2)	63 (12.4)	59 (15.9)	14 (0.596)	64 (15.6)
100	1000	51 (7)	32 (3.25)	50 (8.13)	49 (8.26)	60 (17.9)	10 (0.249)	55* (2.46)
300	1000	59 (6.76)	36 (5.84)	59 (6.81)	59 (6.31)	60 (7.8)	12 (0.442)	55* (3.26)
500	1000	67 (10.8)	42 (5.44)	67 (11.2)	67 (11.1)	67 (10.5)	13 (0.442)	58* (3.64)

Note: *Not all instances achieved the required precision.

Table 3. Small random SDPs: average and variance of the number of oracle calls for reaching a relative precision of 10^{-6} over 15 instances per row (5 for each $p \in \{3, 5, 7\}$).

n	m	CB-ns	CB-fN	CB-lrN	CB-alrN	CB-diag	SDPT3	SB
100	100	75 (25.6)	44 (24.9)	49 (15.8)	52 (15.6)	54 (31.3)	11 (0.573)	255 (504)
300	100	155 (60.4)	75 (44.2)	104 (41.4)	110 (49.1)	86 (29.3)	13 (0.49)	279 (171)
500	100	314 (135)	95 (44.6)	195 (102)	199 (108)	163 (132)	14 (0.611)	464 (399)
100	500	83 (18.8)	68 (27.8)	69 (13.7)	68 (12.6)	76 (20.7)	11 (0.499)	119,453* (1.03×10^5)
300	500	178 (110)	142 (132)	125 (46.3)	127 (54.4)	107 (32.3)	13 (0.806)	289 (207)
500	500	295 (211)	180 (129)	187 (99.9)	188 (99.8)	143 (75.5)	14 (0.596)	532 (462)
100	1000	117 (35.6)	90 (25.4)	96 (21)	97 (22.6)	113 (34)	10 (0.249)	213,306* (3.65×10^4)
300	1000	151 (41.8)	110 (59.8)	123 (23.3)	124 (24.1)	118 (19.9)	12 (0.442)	25,553* (3.58×10^4)
500	1000	238 (159)	152 (65.1)	177 (83.8)	178 (86.7)	148 (37)	13 (0.442)	15,803* (3.12×10^4)

Note: *Not all instances achieved the required precision.

convergence of the low rank and diagonal scaling variants, yet in terms of computation time they clearly outperform the full Newton approach, with the diagonal approach requiring the fewest iterations of the three. The differences in the multiplicity estimates, illustrated in Figure 2, are negligible. All methods arrive at the correct estimate within roughly 60 iterations and do not deviate from this in the sequel.

Figures 3–5 display small to medium sized problems with $n \in \{100, 300, 500\}$, where the full second-order approach does not take too long to run, the results being grouped so that instances with $m = 100$ are shown in Figure 3 (for these the observed multiplicity r ranged from 4 to 7), $m = 500$ in Figure 4 ($9 \le r \le 18$), and $m = 1000$ in Figure 5 ($16 \le r \le 28$). This grouping is

Table 4. Large random SDPs: Average and variance of computation time in seconds for reaching a relative precision of 10^{-6} over 15 instances per row (5 for each $p \in \{3, 4, 5\}$).

n	m	CB-ns	CB-lrN	CB-alrN	CB-diag	SDPT3	SB
1000	1000	53 (22.2)	52 (20.5)	46 (16.3)	28 (7.78)	35 (6.02)	94 (48.2)
2000	1000	169 (78.9)	113 (35.7)	108 (34.6)	65 (13.6)	183 (27.6)	196 (105)
3000	1000	852* (764)	359 (238)	347 (240)	147 (69.3)	567 (133)	1109* (1.05×10^3)
4000	1000	1019 (576)	479 (218)	468 (226)	307 (407)	1228 (172)	1195 (764)
5000	1000	2216* (1.51×10^3)	1026 (866)	1098 (973)	479 (376)	2459 (390)	3366* (2.7×10^3)
6000	1000	3891* (3.43×10^3)	2016 (2.3×10^3)	2091 (2.44×10^3)	659 (609)	3934 (499)	4915* (4.35×10^3)
1000	3000	351 (92.3)	509 (138)	409 (102)	257 (58.9)	105 (41.2)	8561* (3.76×10^3)
2000	3000	593 (398)	619 (276)	529 (249)	285 (103)	278 (86)	5374* (5.57×10^3)
3000	3000	837 (381)	786 (240)	679 (211)	347 (90.4)	726 (167)	12,868* (4.65×10^3)
4000	3000	1819 (1.6×10^3)	1229 (614)	1107 (600)	502 (199)	1515 (384)	9809* (4.74×10^3)
5000	3000	1826 (867)	1394 (595)	1272 (571)	596 (259)	2638 (364)	7610* (4.9×10^3)
6000	3000	2624 (1.61×10^3)	1651 (831)	1592 (854)	736 (274)	4631 (935)	5104* (2.54×10^3)
1000	5000	1635 (506)	2216 (425)	1723 (341)	1178 (205)	251 (102)	17,719* (7.22×10^3)
2000	5000	1908 (1.62×10^3)	2083 (950)	1721 (913)	958 (248)	482 (157)	14,859* (5.9×10^3)
3000	5000	1420 (505)	1675 (449)	1363 (362)	869 (247)	1062 (286)	26,909* (1.69×10^4)
4000	5000	2059 (923)	2224 (998)	1829 (640)	1037 (404)	1746 (269)	39,761* (1.69×10^4)
5000	5000	3214 (2.18×10^3)	3207 (2.76×10^3)	2753 (2.35×10^3)	1201 (421)	3073 (448)	51,811* (1.56×10^4)
6000	5000	4084 (2.88×10^3)	3796 (2.66×10^3)	3079 (1.55×10^3)	1463 (514)	5277 (1.13×10^3)	68,310* (3.3×10^4)

Note: *Not all instances achieved the required precision.

Table 5. Max-Cut on $h \times h \times h$ grids: average and variance of computation time in seconds for reaching a relative precision of 10^{-6} over five instances for each value of h.

h	$n = m$	CB-ns	CB-lrN	CB-alrN	CB-diag	SDPT3	SB
10	1000	3 (0.554)	5 (0.533)	4 (0.546)	3 (0.345)	14 (0.0326)	3 (0.388)
15	3375	41 (4.45)	94 (6.17)	69 (4.31)	37 (2.57)	411 (1.25)	42 (5.25)
20	8000	308 (27.3)	882 (36.9)	727 (40.1)	273 (22.7)	4668 (6.02)	268 (34.8)
25	15,625	1821 (260)	5142 (389)	3916 (247)	1395 (88.6)	52,917 (499)	1602 (200)

Table 6. Max-Cut on $h \times h \times h$ grids: average and variance of the number of oracle calls for reaching a relative precision of 10^{-6} over 5 instances for each value of h.

h	$n = m$	CB-ns	CB-lrN	CB-alrN	CB-diag	SDPT3	SB
10	1000	53 (9.41)	46 (3.01)	44 (2.45)	52 (5.88)	11 (0)	55 (8.91)
15	3375	136 (8.13)	101 (2.33)	98 (2.94)	123 (3.32)	12 (0)	171 (28.9)
20	8000	251 (7.86)	189 (4.12)	186 (3.72)	227 (5.71)	12 (0)	290 (38)
25	15,625	452 (49.1)	307 (10.1)	307 (8)	373 (6.99)	13 (0)	555 (78.1)

motivated by the fact that for relative precision requirements beyond 10^{-3} the decisive parameter for the performance of the SB method relative to interior-point methods is the number of constraints m. In particular, the plots of Figure 5 show that for most SB variants the order of the semidefinite matrix n is less relevant than m; in part this might also be due to the surprising observation that for these instances and constant m the values of r decrease with increasing n. The plots also confirm that for increasing m and increasing precision the interior-point approach SDPT3 becomes more attractive. While CB-fN is reasonably competitive for instances with small $n \in \{100, 300\}$ and $m = 100$, it performs poorly in terms of computation time in spite of its rather small number of oracle calls (Table 3). On the other hand CB-diag clearly outperforms CB-alrN and CB-lrN in computation time, the real surprise being that it also needs fewer oracle calls quite regularly (Table 3). CB-alrN is a bit faster than CB-lrN, both need roughly the same number of oracle

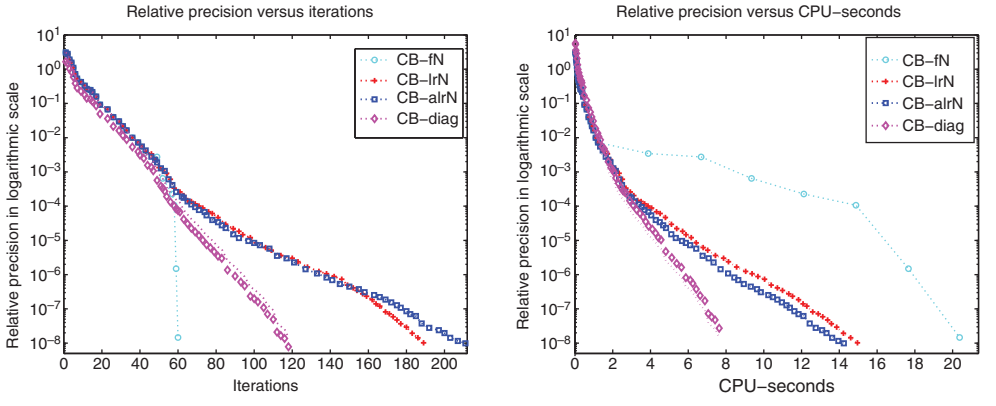

Figure 1. Evolution of the relative optimality gap for an example with $n = 500$ and $m = 500$.

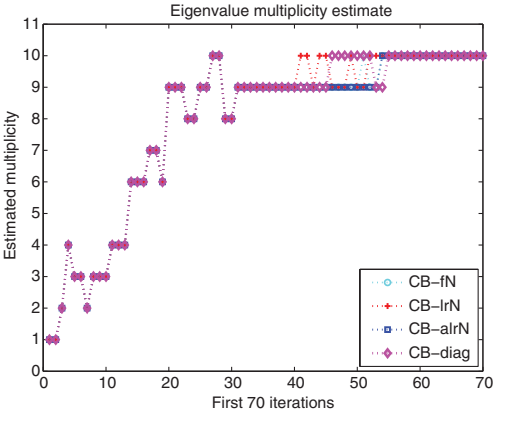

Figure 2. Evolution of the multiplicity estimate during the first 70 iterations.

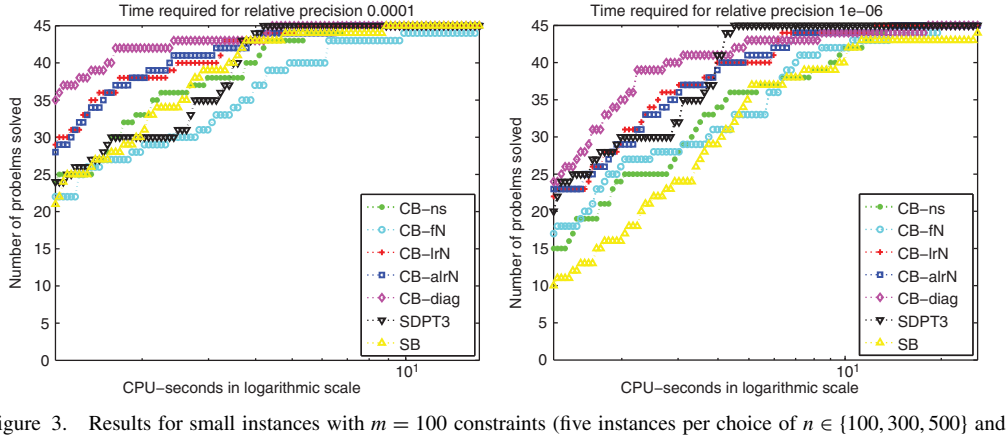

Figure 3. Results for small instances with $m = 100$ constraints (five instances per choice of $n \in \{100, 300, 500\}$ and $p \in \{3, 5, 7\}$).

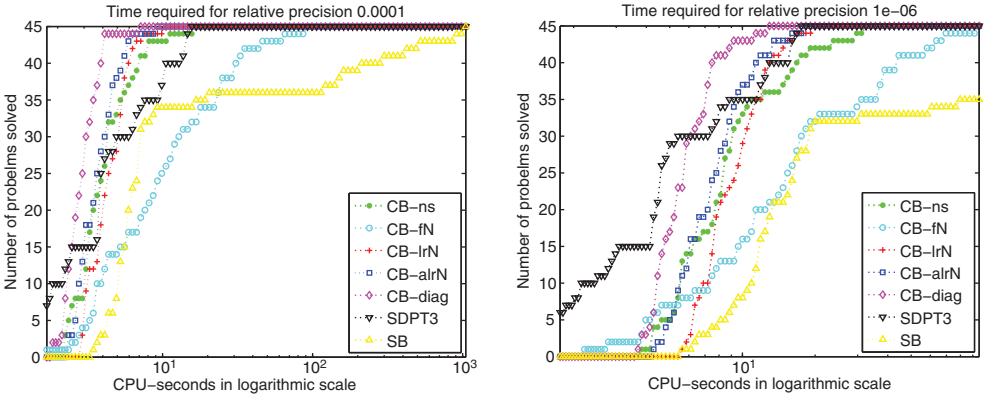

Figure 4. Results for medium instances with $m = 500$ constraints (five instances per choice of $n \in \{100, 300, 500\}$ and $p \in \{3, 5, 7\}$).

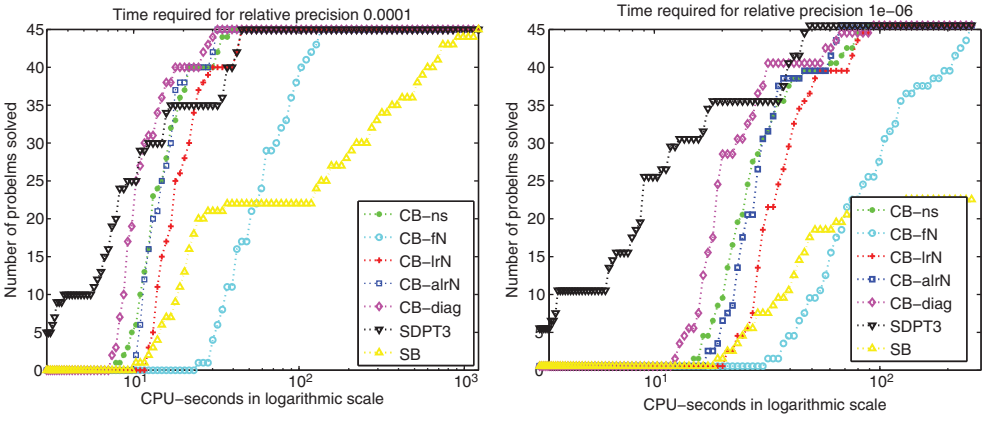

Figure 5. Results for medium instances with $m = 1000$ constraints (five instances per choice of $n \in \{100, 300, 500\}$ and $p \in \{3, 5, 7\}$).

calls. Note that CB-diag is clearly better than CB-ns with respect to time and calls, but CB-alrN may well be outperformed by CB-ns as m increases in spite of the significant difference in oracle calls, as the additional cost of low-rank scaling is considerable. In general, all CB variants seem to be preferable to the old SB code whose performance deteriorates quickly for higher precision requirements and increasing m.

Figures 6–8 illustrate the development of computation time for increasing matrix sizes $n \in \{1000 \cdot i : i = 1, \ldots, 6\}$ grouped by instances with $m = 1000$ (Figure 6, $8 \le r \le 16$), $m = 3000$ (Figure 7, $17 \le r \le 29$), and $m = 5000$ (Figure 7, $23 \le r \le 40$). For these instances CB-fN is no longer an option and therefore it is excluded from these tests. The advantage of CB-diag, however, becomes even more apparent as the order of the matrices increases. The additional effort of low rank scaling already pays off for rather moderate precision requirements. This is even more so if the number of constraints is increased, as can be seen in Figure 7, where $m = 3000$ and matrix sizes range in $n \in \{1000 \cdot i : i = 1, \ldots, 6\}$. SB is not competitive, failing to obtain the desired precision almost all the time, while once again we see confirmed that the advantage of SDPT3 for larger m is diminishing with increasing matrix order n.

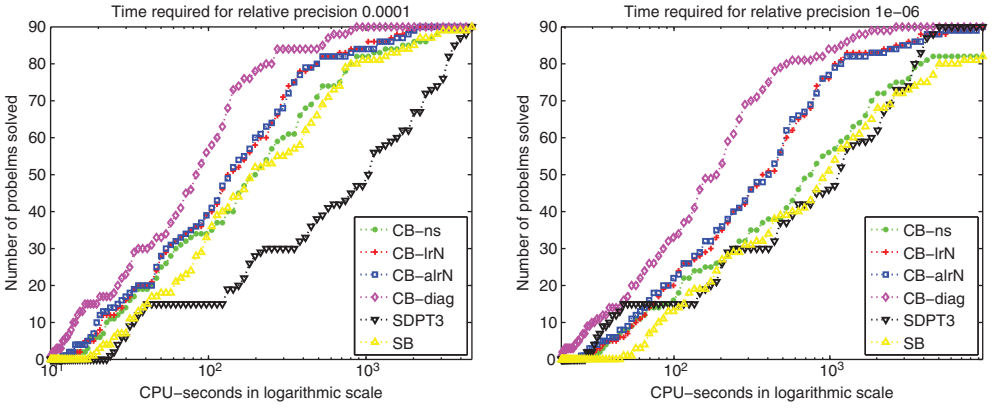

Figure 6. Results for big instances with $m = 1000$ constraints (five instances per choice of $n \in \{1000 \cdot i : i = 1, \ldots, 6\}$ and $p \in \{3, 4, 5\}$).

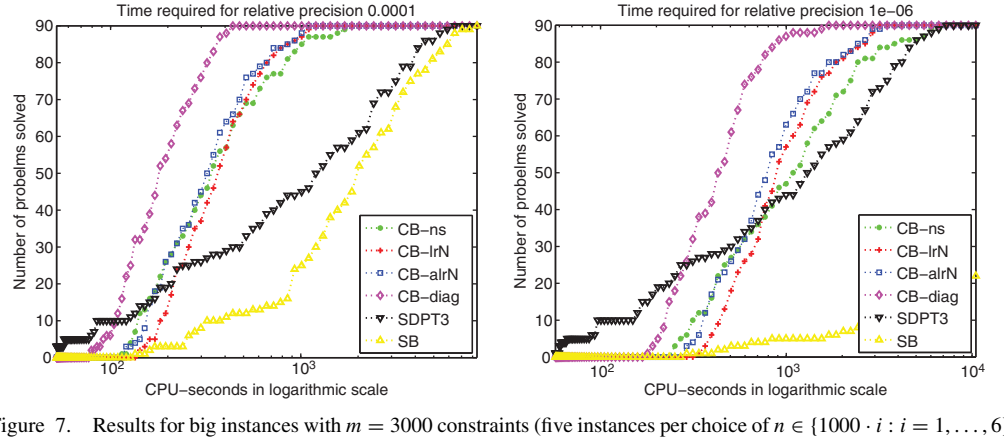

Figure 7. Results for big instances with $m = 3000$ constraints (five instances per choice of $n \in \{1000 \cdot i : i = 1, \ldots, 6\}$ and $p \in \{3, 4, 5\}$).

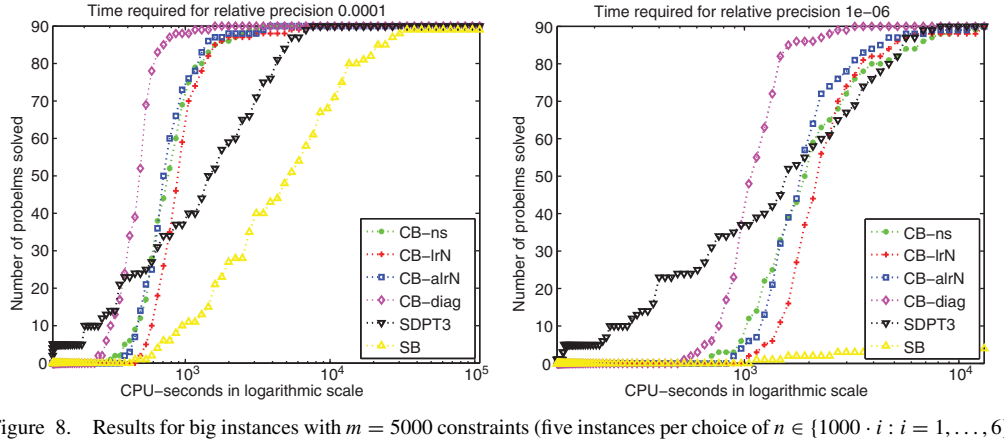

Figure 8. Results for big instances with $m = 5000$ constraints (five instances per choice of $n \in \{1000 \cdot i : i = 1, \ldots, 6\}$ and $p \in \{3, 4, 5\}$).

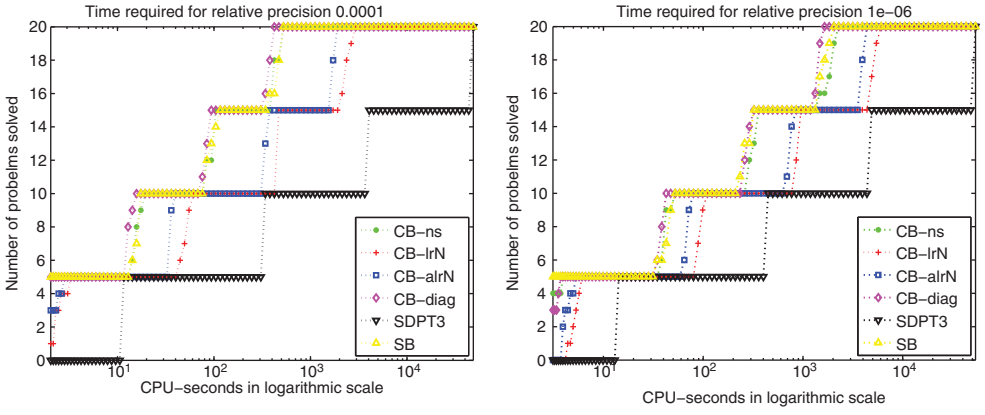

Figure 9. Results for SDP-relaxations of max-cut instances of Ising spin glasses on toroidal $n = h \times h \times h$ grids (five instances per $h \in \{10, 15, 20, 25\}$).

Finally, Figure 9 and Tables 5 and 6 present the results for computing the SDP relaxation of max-cut for Ising spin glasses on $h \times h \times h$ grids for $h \in \{10, 15, 20, 25\}$ (for $h = 30$, SDPT3 failed due to memory problems); the observed value of r was roughly h for instances corresponding to h. For these instances the code SB was known to perform quite well and the main purpose here is to show that none of this good performance is lost in the case of CB-diag. SDPT3 cannot compete with the bundle approaches, but also CB-lrN and CB-alrN fall off considerably in comparison to the diagonal variants in spite of their smaller number of oracle calls (Table 6).

7. Conclusions

The proposed diagonal scaling technique based on a low rank approximation of the second-order approach of Overton and Womersley [23] significantly improves the performance of the SB method [13] and, surprisingly, also requires fewer evaluations on the instances considered than our approaches based on the low-rank approximation itself. It allows computing solutions within relative precision of 10^{-6} routinely. It appears to be faster than the original SB method even for precision requirements of 10^{-4}. In comparison with the excellent package SDPT3 [30] (a primal dual interior-point method employing a preconditioned conjugate gradient solver) the scaled SB approach seems to be competitive to superior in computing solutions with a precision requirement of 10^{-6} whenever the number of constraints m is not significantly bigger than the order n of the matrix. The advantage turns toward SDPT3 if the relative size of m increases while it turns towards the scaled SB approach when precision requirements decrease. The main advantages of the SB method are its quick computation of low precision approximations to optimal solutions, its applicability to very large-scale problems and its suitability for combinatorial cutting plane algorithms due to its advantageous restart properties. The new diagonal scaling variant achieves higher precision and increased robustness and efficiency as well.

Acknowledgement

We thank Henry Wolkowicz for the encouragement to experiment with Tapia indicators for identifying the multiplicity r.

Funding

This work was supported by the National Science Foundation [grant number DMS-1016325].

References

[1] A.I. Barvinok, *Problems of distance geometry and convex properties of quadratic maps*, Discret Comput. Geom. 13 (1995), pp. 189–202.
[2] J. Cullum, W.E. Donath, and P. Wolfe, *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices*, Math. Program. Study 3 (1975), pp. 25–55.
[3] E. Dolan and J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program. 91(2) (2002), pp. 201–213.
[4] A. Edelman, T.A. Arias, and S.T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl. 20(2) (1999), pp. 303–353.
[5] A.S. El-Bakry, R.A. Tapia, and Y. Zhang. *A study of indicators for identifying zero variables in interior-point methods*, SIAM Rev. 36(1) (1994), pp. 45–72.
[6] R. Fletcher, *Semi-definite matrix constraints in optimization*, SIAM J. Control Optim. 23(4) (1985), pp. 493–513.
[7] S. Friedland, J. Nocedal, and M.L. Overton, *The formulation and analysis of numerical methods for inverse eigenvalue problems*, SIAM J. Num. Anal. 24 (1987), pp. 634–667.
[8] C. Helmberg, Semidefinite programming for combinatorial optimization. Habilitations-schrift TU Berlin, Jan. 2000; ZIB-Report ZR 00-34, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, 14195 Berlin, Germany, October 2000.
[9] C. Helmberg, `Numerical evaluation of SBmethod`, Math. Program. 95(2) (2003), pp. 381–406.
[10] C. Helmberg, *ConicBundle 0.3*. Fakultät für Mathematik, Technische Universität Chemnitz, 2009. Available at http://www.tu-chemnitz.de/~helmberg/ConicBundle.
[11] C. Helmberg and K.C. Kiwiel, *A spectral bundle method with bounds*, Math. Program. 93(2) (2002), pp. 173–194.
[12] C. Helmberg and F. Oustry, *Bundle methods to minimize the maximum eigenvalue function*, in *Handbook of Semidefinite Programming: Theory, Algorithms and Applications*, R. Saigal H. Wolkowicz, and L. Vandenberghe, eds., Kluwer, Boston, MA, 2000, pp. 307–337.
[13] C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM J. Optim. 10 (2000), pp. 673–696.
[14] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms I+II*, Grundlehren der mathematischen Wissenschaften, Vol. 305/306, Springer, Berlin, 1993.
[15] F. Jarre, *An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices*, SIAM J. Control Optim. 31(5) (1993), pp. 1360–1377.
[16] T. Kato, *Perturbation Theory for Linear Operators*, Grundlehren der mathematischen Wissenschaften, Vol. 132, 2nd corr. print. of the 2nd ed., Springer, Berlin, 1984.
[17] P. Lancaster, *On eigenvalues of matrices dependent on a parameter*, Numer. Math. 6 (1964), pp. 377–387.
[18] F. Oustry, *The U-Lagrangian of the maximum eigenvalue function*, SIAM J. Optim. 9 (1999), pp. 526–549.
[19] F. Oustry, *A second-order bundle method to minimize the maximum eigenvalue function*, Math. Program. 89 (2000), pp. 1–33.
[20] M.L. Overton, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. Matrix Anal. Appl. 9(2) (1988), pp. 256–268.
[21] M.L. Overton, *Large-scale optimization of eigenvalues*, SIAM J. Optim. 2(1) (1992), pp. 88–120.
[22] M.L. Overton and R.S. Womersley, *On the sum of the largest eigenvalues of a symmetric matrix*, SIAM J. Matrix Anal. Appl. 13 (1992), pp. 41–45.
[23] M.L. Overton and R.S. Womersley, *Second derivatives for optimizing eigenvalues of symmetric matrices*, SIAM J. Matrix Anal. Appl. 16 (1995), pp. 697–718.
[24] G. Pataki, *On the rank of extreme matrices in semidefinite programming and the multiplicity of optimal eigenvalues*, Math. Oper. Res. 23(2) (1998), pp. 339–358.
[25] 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 (1992), pp. 121–152.
[26] A. Shapiro and M.K.H. Fan, *On eigenvalue optimization*, SIAM J. Optim. 5(3) (1995), pp. 552–569.
[27] C. De Simone, M. Diel, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi, *Exact ground states of Ising spin glasses: New experimental results with a branch-and-cut algorithm*. J. Stat. Phys. 80 (1995), pp. 487–496.
[28] M.J. Todd, *A study of search directions in primal-dual interior-point methods for semidefinite programming*, Optim. Methods Softw. 11 (1999), pp. 1–46.
[29] M.J. Todd, *Semidefinite optimization*, Acta Numer. 10 (2001), pp. 515–560.
[30] K.-C. Toh, M.J. Todd, and R.H. Tütüncü, *SDPT3 version 4.0 beta*, National University of Singapore, February 2009; software available at http://www.math.nus.edu.sg/~mattohkc/sdpt3.html (May 21, 2010).