Math. Prog. Comp. (2014) 6:103–150
DOI 10.1007/s12532-013-0062-7

FULL LENGTH PAPER

A first-order block-decomposition method for solving two-easy-block structured semidefinite programs

Renato D. C. Monteiro · Camilo Ortiz · Benar F. Svaiter

Received: 6 September 2012 / Accepted: 14 October 2013 / Published online: 10 November 2013

Abstract In this paper, we consider a first-order block-decomposition method for minimizing the sum of a convex differentiable function with Lipschitz continuous gradient, and two other proper closed convex (possibly, nonsmooth) functions with easily computable resolvents. The method presented contains two important ingredients from a computational point of view, namely: an adaptive choice of stepsize for performing an extragradient step; and the use of a scaling factor to balance the blocks. We then specialize the method to the context of conic semidefinite programming (SDP) problems consisting of two easy blocks of constraints. Without putting them in standard form, we show that four important classes of graph-related conic SDP problems automatically possess the above two-easy-block structure, namely: SDPs for θ-functions and θ_+-functions of graph stable set problems, and SDP relaxations of binary integer quadratic and frequency assignment problems. Finally, we present computational results on the aforementioned classes of SDPs showing that our method outperforms the three most competitive codes for large-scale conic semidefinite programs, namely:

The work of R. D. C. Monteiro was partially supported by NSF Grants CCF-0808863, CMMI-0900094 and CMMI- 1300221, and ONR Grant ONR N00014-11-1-0062.

The work of B. F. Svaiter was partially supported by CNPq Grants No. 303583/2008-8, 302962/2011-5, 480101/2008-6, 474944/2010-7, FAPERJ Grants E-26/102.821/2008, E-26/102.940/2011.

R. D. C. Monteiro · C. Ortiz (✉)
School of Industrial and Systems Engineering, Georgia Institute of Technology,
Atlanta, GA 30332-0205, USA
e-mail: camiort@gatech.edu

R. D. C. Monteiro
e-mail: monteiro@isye.gatech.edu

B. F. Svaiter
IMPA, Estrada Dona Castorina 110, Rio de Janeiro 22460-320, Brazil
e-mail: benar@impa.br

the boundary point (BP) method introduced by Povh et al., a Newton-CG augmented Lagrangian method, called SDPNAL, by Zhao et al., and a variant of the BP method, called the SPDAD method, by Wen et al.

Keywords Complexity · Proximal · Extragradient · Block-decomposition · Convex optimization · Conic optimization · Semidefinite programming

Mathematics Subject Classification (2000) 49M27 · 49M37 · 90C06 · 90C22 · 90C30 · 90C35 · 90C90 · 65K05 · 90-08

1 Introduction

Let $\mathbb{R}$ denote the set of real numbers, $\mathbb{R}^n$ denote the n-dimensional Euclidean space, $\mathbb{R}^n_+$ denote the cone of nonnegative vectors in $\mathbb{R}^n$, $\mathcal{S}^n$ denote the set of all $n \times n$ symmetric matrices and $\mathcal{S}^n_+$ denote the cone of $n \times n$ symmetric positive semidefinite matrices. Let $\mathcal{X}$ and $\mathcal{W}$ be finite dimensional vector spaces and consider the conic programming problem

$$\min\{c(x) \ : \ \mathcal{A}x = b, \ x \in \mathcal{K}\}, \tag{1}$$

where $\mathcal{A} : \mathcal{X} \to \mathcal{W}$ and $c : \mathcal{X} \to \mathbb{R}$ are linear mappings, $b \in \mathcal{W}$ and $\mathcal{K} \subset \mathcal{X}$ is a closed convex cone. Several papers [9,10,12,21,22] in the literature discuss methods/codes for solving large-scale conic semidefinite programming (SDP) problems, i.e., special cases of (1) in which

$$\mathcal{X} = \mathbb{R}^{n_u+n_l} \times \mathcal{S}^{n_s}, \quad \mathcal{W} = \mathbb{R}^m, \quad \mathcal{K} = \mathbb{R}^{n_u} \times \mathbb{R}^{n_l}_+ \times \mathcal{S}^{n_s}_+. \tag{2}$$

Presently, the most efficient methods/codes for solving large-scale conic SDP problems are the first-order projection-type discussed in [10,12,21,22] (see also [15] for a slight variant of [10]).

More specifically, augmented Lagrangian approaches have been proposed for the dual formulation of (1) with $\mathcal{X}, \mathcal{W}$ and $\mathcal{K}$ as in (2) for the case when m, n_u and n_l are large (up to a few millions) and n_s is moderate (up to a few thousands). In [10,15], a boundary point method for solving (1) is proposed which can be viewed as variants of the alternating direction method of multipliers introduced in [6,7] applied to the dual formulation of (1). In [22], an inexact augmented Lagrangian method is proposed which solves a reformulation of the augmented Lagrangian subproblem via a semismooth Newton approach combined with the conjugate gradient method. Using the theory developed in [11], an implementation of a first-order block-decomposition (BD) algorithm, based on the hybrid proximal extragradient (HPE) method [18], for solving standard form conic SDP problems is discussed in [12], and numerical results are presented showing that it generally outperforms the methods of [10,22]. In [21], an efficient variant of the BP method is discussed and numerical results are presented showing its impressive ability to solve important classes of large-scale graph-related SDP problems. It should be observed though that the implementation in [21], as well as the one described in this work, are very specific in the sense that they both take advantage of each SDP problem class structure so as to keep the number of variables and/or constraints as small as possible. This contrasts with the codes described in

[10,21] and [22], which always introduce additional variables and/or constraints into the original SDP formulation to bring it into the required standard form input.

Our goal in this paper is to study the performance of a BD method based on the BD-HPE framework in [11] for solving conic optimization problems, not necessarily in standard form, with two "easy" blocks of constraints. (We will simply say that these problems have the "two-easy-block" structure.) We first present a first-order BD method for minimizing the sum of a convex differentiable function with Lipschitz continuous gradient, and two other proper closed convex (possibly, nonsmooth) functions with easily computable resolvents. The method presented contains two important ingredients from a computational point of view, namely: an adaptive choice of stepsize for performing an extragradient step; and the use of a scaling factor to balance the blocks. We discuss its specialization to the context of conic SDP problems possessing the "two-easy-block" structure. Then, we apply it to solve four important classes of graph-related conic SDP problems which have the two-easy-block structure, namely: SDPs for θ-functions and θ_+-functions of graph stable set problems, and SDP relaxations of binary integer quadratic and frequency assignment problems. Finally, we present computational results on several instances of the aforementioned classes of conic SDPs showing that our method substantially outperforms the codes in [12,21] and [22]. Since the code in this paper works directly in the conic optimization problem as given, and hence works with a formulation with less number of variables, it is not surprising that it also outperforms the BD method of [12], which in contrast requires as input an SDP problem in standard form.

Our paper is organized as follows. Section 2 reviews some facts about the ε-subdifferential of a convex function and the ε-enlargement of a monotone operator. Section 3 presents an adaptive block-decomposition HPE (A-BD-HPE) framework in the context of block-structured monotone inclusion problems, similar to the one presented in [11], but with an adaptive choice of stepsize for performing the extragradient step. Section 4 presents a first-order instance of the A-BD-HPE framework, and corresponding iteration-complexity results, for solving a minimization problem whose objective function is the sum of a finite everywhere convex function with Lipschitz continuous gradient and two proper closed convex (possibly, nonsmooth) functions with easily computable resolvents. Section 5 discusses the specialization of the method of Sect. 4 to the context of conic optimization problems with a two-easy-block structure. Section 6 describes a practical variant of the BD method of Sect. 5 which incorporates a dynamic update of the scaling factor to balance the blocks. Section 7 presents numerical results comparing the latter variant of the BD method to the method discussed in [21]. Section 8 briefly compares this variant of the BD method with the methods in [12] and [22]. Finally, Sect. 9 presents some final remarks.

2 The ε-subdifferential and ε-enlargement of monotone operators

In this section, we review some properties of the ε-subdifferential of a convex function and the ε-enlargement of a monotone operator.

Let $\mathcal{Z}$ denote a finite dimensional inner product space with inner product and associated norm denoted by $\langle \cdot, \cdot \rangle_{\mathcal{Z}}$ and $\| \cdot \|_{\mathcal{Z}}$. A point-to-set operator $T : \mathcal{Z} \rightrightarrows \mathcal{Z}$ is a relation $T \subseteq \mathcal{Z} \times \mathcal{Z}$ and

$$T(z) = \{v \in \mathcal{Z} \mid (z, v) \in T\}.$$

Alternatively, one can consider T as a multi-valued function of $\mathcal{Z}$ into the family $\wp(\mathcal{Z}) = 2^{(\mathcal{Z})}$ of subsets of $\mathcal{Z}$. Regardless of the approach, it is usual to identify T with its graph defined as

$$Gr(T) = \{(z, v) \in \mathcal{Z} \times \mathcal{Z} \mid v \in T(z)\}.$$

The domain of T, denoted by Dom T, is defined as

$$\text{Dom}\, T := \{z \in \mathcal{Z} : T(z) \neq \emptyset\}.$$

An operator $T : \mathcal{Z} \rightrightarrows \mathcal{Z}$ is *affine* if its graph is an affine manifold. An operator $T : \mathcal{Z} \rightrightarrows \mathcal{Z}$ is *monotone* if

$$\langle v - \tilde{v}, z - \tilde{z}\rangle_{\mathcal{Z}} \geq 0, \quad \forall (z, v), (\tilde{z}, \tilde{v}) \in Gr(T),$$

and T is *maximal monotone* if it is monotone and maximal in the family of monotone operators with respect to the partial order of inclusion, i.e., $S : \mathcal{Z} \rightrightarrows \mathcal{Z}$ monotone and $Gr(S) \supset Gr(T)$ implies that $S = T$. The following result states Moreau's identity.

Lemma 2.1 (Moreau's identity; see Lemma 6.3 in [11]) *Let $\lambda > 0, z \in \mathcal{Z}$ and $T : \mathcal{Z} \rightrightarrows \mathcal{Z}$ be a point to set maximal monotone operator. Then,*

$$z = (I + \lambda T)^{-1}(z) + \lambda(I + \lambda^{-1}T^{-1})^{-1}(\lambda^{-1}z).$$

In [1], Burachik, Iusem and Svaiter introduced the ε-enlargement of maximal monotone operators. In [13] this concept was extended to a generic point-to-set operator in $\mathcal{Z}$ as follows. Given $T : \mathcal{Z} \rightrightarrows \mathcal{Z}$ and a scalar ε, define $T^{\varepsilon} : \mathcal{Z} \rightrightarrows \mathcal{Z}$ as

$$T^{\varepsilon}(z) = \{v \in \mathcal{Z} \mid \langle z - \tilde{z}, v - \tilde{v}\rangle_{\mathcal{Z}} \geq -\varepsilon, \; \forall \tilde{z} \in \mathcal{Z}, \; \forall \tilde{v} \in T(\tilde{z})\}, \quad \forall z \in \mathcal{Z}.$$

We now state a few useful properties of the operator T^{ε} that will be needed in our presentation.

Proposition 2.2 *Let $T, T' : \mathcal{Z} \rightrightarrows \mathcal{Z}$. Then,*

(a) *if $\varepsilon_1 \leq \varepsilon_2$, then $T^{\varepsilon_1}(z) \subseteq T^{\varepsilon_2}(z)$ for every $z \in \mathcal{Z}$;*
(b) *$T^{\varepsilon}(z) + (T')^{\varepsilon'}(z) \subseteq (T + T')^{\varepsilon+\varepsilon'}(z)$ for every $z \in \mathcal{Z}$ and $\varepsilon, \varepsilon' \in \mathbb{R}$;*
(c) *T is monotone if and only if $T \subseteq T^0$;*
(d) *T is maximal monotone if and only if $T = T^0$;*

We refer the reader to [2,19] for further discussion on the ε-enlargement of a maximal monotone operator.

For a scalar $\varepsilon \geq 0$, the ε-subdifferential of a function $f : \mathcal{Z} \to [-\infty, +\infty]$ is the operator $\partial_{\varepsilon} f : \mathcal{Z} \rightrightarrows \mathcal{Z}$ defined as

$$\partial_{\varepsilon} f(z) = \{v \mid f(\tilde{z}) \geq f(z) + \langle \tilde{z} - z, v\rangle_{\mathcal{Z}} - \varepsilon, \; \forall \tilde{z} \in \mathcal{Z}\}, \quad \forall z \in \mathcal{Z}.$$

When $\varepsilon = 0$, the operator $\partial_\varepsilon f$ is simply denoted by ∂f and is referred to as the subdifferential of f. The operator ∂f is trivially monotone if f is proper. If f is a proper lower semi-continuous convex function, then ∂f is maximal monotone [17].

The conjugate f^* of f is the function $f^* : \mathcal{Z} \to [-\infty, \infty]$ defined as

$$f^*(v) = \sup_{z \in \mathcal{Z}} \langle v, z \rangle_{\mathcal{Z}} - f(z), \quad \forall v \in \mathcal{Z}.$$

The following result lists some useful properties about the ε-subdifferential of a proper convex function.

Proposition 2.3 *Let $f : \mathcal{Z} \to (-\infty, \infty]$ be a proper convex function. Then,*

(a) $\partial_\varepsilon f(z) \subseteq (\partial f)^\varepsilon(z)$ *for any* $\varepsilon \geq 0$ *and* $z \in \mathcal{Z}$;
(b) *if f is closed, then* $\partial(f^*) = (\partial f)^{-1}$;
(c) *if* $v \in \partial f(z)$ *and* $f(\tilde{z}) < \infty$, *then* $v \in \partial_\varepsilon f(\tilde{z})$, *for every* $\varepsilon \geq f(\tilde{z}) - [f(z) + \langle \tilde{z} - z, v \rangle]$.

The *indicator function* of a closed convex set $Z \subseteq \mathcal{Z}$ is the function $\delta_Z : \mathcal{Z} \to [0, \infty]$ defined as

$$\delta_Z(z) = \begin{cases} 0, & z \in Z, \\ \infty, & \text{otherwise}. \end{cases}$$

For a closed convex cone $\mathcal{K} \subseteq \mathcal{Z}$, we have the following characterization about the ε-subdifferential of $\delta_{\mathcal{K}}$.

Proposition 2.4 *Let $\mathcal{K} \subseteq \mathcal{Z}$ be a (nonempty) closed convex cone. Then, for any $\varepsilon \geq 0$, the pair $(z, w) \in \mathcal{Z} \times \mathcal{Z}$ satisfies $w \in -\partial_\varepsilon \delta_{\mathcal{K}}(z)$ if and only if $z \in \mathcal{K}$, $w \in \mathcal{K}^*$ and $\langle z, w \rangle_{\mathcal{Z}} \leq \varepsilon$, where $\mathcal{K}^*$ is dual cone of $\mathcal{K}$ defined as*

$$\mathcal{K}^* := \{w \in \mathcal{Z} : \langle z, x \rangle \geq 0, \ \forall x \in \mathcal{K}\}.$$

3 The A-BD-HPE framework

In this section, we review the A-BD-HPE framework with adaptive stepsize for solving a special type of monotone inclusion problem consisting of the sum of a continuous monotone map and a point-to-set maximal monotone operator with a separable two-block-structure.

Let $\mathcal{X}$ and $\mathcal{Y}$ be finite dimensional inner product spaces with associated inner products denoted by $\langle \cdot, \cdot \rangle_{\mathcal{X}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{Y}}$, respectively, and associated norms denoted by $\| \cdot \|_{\mathcal{X}}$ and $\| \cdot \|_{\mathcal{Y}}$, respectively. We endow the product space $\mathcal{X} \times \mathcal{Y}$ with the canonical inner product $\langle \cdot, \cdot \rangle_{\mathcal{X},\mathcal{Y}}$ and associated canonical norm $\| \cdot \|_{\mathcal{X},\mathcal{Y}}$ defined as

$$\langle (x, y), (x', y') \rangle_{\mathcal{X},\mathcal{Y}} := \langle x, x' \rangle_{\mathcal{X}} + \langle y, y' \rangle_{\mathcal{Y}}, \quad \|(x, y)\|_{\mathcal{X},\mathcal{Y}} := \sqrt{\langle (x, y), (x, y) \rangle_{\mathcal{X},\mathcal{Y}}}, \tag{3}$$

for all $(x, y), (x', y') \in \mathcal{X} \times \mathcal{Y}$.

Our problem of interest in this section is the monotone inclusion problem of finding $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that

$$(0, 0) \in [F + (C \otimes D)](x, y), \tag{4}$$

where

$$F(x,y)=(F_1(x,y),F_2(x,y))\in\mathcal{X}\times\mathcal{Y},\quad (C\otimes D)(x,y)=C(x)\times D(y)\subseteq\mathcal{X}\times\mathcal{Y}$$

and the following conditions are assumed:

(A.1) $C:\mathcal{X}\rightrightarrows\mathcal{X}$ and $D:\mathcal{Y}\rightrightarrows\mathcal{Y}$ are maximal monotone;
(A.2) F : Dom $F\subseteq\mathcal{X}\times\mathcal{Y}\to\mathcal{X}\times\mathcal{Y}$ is a continuous map such that Dom $F\supset$ cl(Dom C) $\times\,\mathcal{Y}$;
(A.3) F is monotone on cl(Dom C) $\times$ cl(Dom D);
(A.4) there exists $L>0$ such that

$$\|F_1(x,y')-F_1(x,y)\|_{\mathcal{X}}\le L\|y'-y\|_{\mathcal{Y}},\quad \forall x\in \mathrm{Dom}\,C,\ \forall y,y'\in\mathcal{Y}.$$

It is trivial to check that $C\otimes D$ is maximal monotone. Moreover, in view of Proposition A.1 of [14], it follows that $F+(C\otimes D)$ is maximal monotone. Note that problem (4) is equivalent to

$$0\in F_1(x,y)+C(x),\quad 0\in F_2(x,y)+D(y).$$

We now state the A-BD-HPE framework. Its statement uses $\lambda_{\max}(\cdot)$ to denote the maximum eigenvalue function of a symmetric matrix.

A-BD-HPE Framework: An adaptive block-decomposition HPE framework

0) Let $(x^0,y^0)\in\mathcal{X}\times\mathcal{Y}$, $\sigma\in(0,1]$, $\sigma_x\in[0,1]$ and $\tilde\sigma_x,\sigma_y\in[0,1)$ be given and set $k=1$;
1) choose $\tilde\lambda_k>0$ such that

$$\sigma_k:=\lambda_{\max}\left(\begin{bmatrix}\sigma_x^2 & \tilde\lambda_k\tilde\sigma_x L\\ \tilde\lambda_k\tilde\sigma_x L & \sigma_y^2+\tilde\lambda_k^2L^2\end{bmatrix}\right)^{1/2}\le\sigma; \tag{5}$$

2) compute $\tilde x^k, c^k\in\mathcal{X}$ and $\varepsilon_k'\ge 0$ such that

$$c^k\in C^{\varepsilon_k'}(\tilde x^k),\quad \|\tilde\lambda_k[F_1(\tilde x^k,y^{k-1})+c^k]+\tilde x^k-x^{k-1}\|_{\mathcal{X}}^2+2\tilde\lambda_k\varepsilon_k'\le\sigma_x^2\|\tilde x^k-x^{k-1}\|_{\mathcal{X}}^2, \tag{6}$$

$$\|\tilde\lambda_k[F_1(\tilde x^k,y^{k-1})+c^k]+\tilde x^k-x^{k-1}\|_{\mathcal{X}}\le\tilde\sigma_x\|\tilde x^k-x^{k-1}\|_{\mathcal{X}}; \tag{7}$$

3) compute $\tilde y^k, d^k\in\mathcal{Y}$ and $\varepsilon_k''\ge 0$ such that

$$d^k\in D^{\varepsilon_k''}(\tilde y^k),\quad \|\tilde\lambda_k[F_2(\tilde x^k,\tilde y^k)+d^k]+\tilde y^k-y^{k-1}\|_{\mathcal{Y}}^2+2\tilde\lambda_k\varepsilon_k''\le\sigma_y^2\|\tilde y^k-y^{k-1}\|_{\mathcal{Y}}^2; \tag{8}$$

4) choose λ_k to be the largest $\lambda>0$ such that

$$\begin{aligned}&\|\lambda[F(\tilde x^k,\tilde y^k)+(c^k,d^k)]+(\tilde x^k,\tilde y^k)-(x^{k-1},y^{k-1})\|_{\mathcal{X},\mathcal{Y}}^2+2\lambda(\varepsilon_k'+\varepsilon_k'')\\&\quad\le\sigma^2\|(\tilde x^k,\tilde y^k)-(x^{k-1},y^{k-1})\|_{\mathcal{X},\mathcal{Y}}^2;\end{aligned} \tag{9}$$

5) set

$$(x^k,y^k)=(x^{k-1},y^{k-1})-\lambda_k[F(\tilde x^k,\tilde y^k)+(c^k,d^k)],$$

$k\leftarrow k+1$, and go to step 1.

The following result follows from Proposition 3.1 of [11] and Proposition 2.2 of [12].

Proposition 3.1 *Consider the sequences* $\{\lambda_k\}$ *and* $\{\tilde{\lambda}_k\}$ *generated by the A-BD-HPE framework. Then, for every* $k \in \mathbb{N}$, $\lambda = \tilde{\lambda}_k$ *satisfies* (9). *As a consequence* $\lambda_k \geq \tilde{\lambda}_k$.

The following point-wise convergence result for the A-BD-HPE framework follows from Theorem 3.2 of [11] and Theorem 2.3 of [12].

Theorem 3.2 *Assume that* $\sigma < 1$ *and consider the sequences* $\{(\tilde{x}^k, \tilde{y}^k)\}$, $\{(c^k, d^k)\}$, $\{\lambda_k\}$ *and* $\{(\varepsilon'_k, \varepsilon''_k\}$ *generated by the A-BD-HPE framework and let* d_0 *denote the distance of the initial point* $(x^0, y^0) \in \mathcal{X} \times \mathcal{Y}$ *to the solution set of* (4). *Then, for every* $k \in \mathbb{N}$, *there exists* $i \leq k$ *such that*

$$\|F(\tilde{x}^i, \tilde{y}^i) + (c^i, d^i)\|_{\mathcal{X},\mathcal{Y}} \leq d_0 \sqrt{\frac{1+\sigma}{1-\sigma}\left(\frac{1}{\lambda_i \sum_{j=1}^k \lambda_j}\right)},$$

$$\varepsilon'_i + \varepsilon''_i \leq \frac{\sigma^2 d_0^2}{2(1-\sigma^2)\sum_{j=1}^k \lambda_j}.$$

4 A BD algorithm for a class of structured convex optimization

This section presents a first-order BD algorithm, and corresponding complexity results, for solving a minimization problem whose objective function is the sum of a finite everywhere convex function with Lipschitz continuous gradient and two proper closed convex (possibly, nonsmooth) functions with easily computable resolvents.

Throughout this section, $\mathcal{X}$ denotes a finite dimensional inner product space with corresponding inner product and norm denoted by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively. We are concerned with the optimization problem

$$\begin{aligned} \min \quad & f(x) + h_1(x) + h_2(x) \\ \text{s.t.} \quad & x \in \mathcal{X}, \end{aligned} \tag{10}$$

where:

(B.1) $f, h_1, h_2 : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ are convex lower semicontinuous proper functions;
(B.2) f is differentiable on $\mathcal{X}$ and its gradient is L_f-Lipschitz continuous, that is,

$$\|\nabla f(x) - \nabla f(x')\| \leq L_f \|x - x'\|, \quad \forall x, x' \in \mathcal{X}; \tag{11}$$

(B.3) the intersection of the relative interiors of the effective domains of h_1 and h_2 is non-empty.

In view of the above assumptions and [16, Theorem 23.8], we have $\partial(f + h_1 + h_2) = \nabla f + \partial h_1 + \partial h_2$. Therefore, x^* is an optimal solution of (10) if and only if

$$0 \in \nabla f(x^*) + \partial h_1(x^*) + \partial h_2(x^*). \tag{12}$$

Using Proposition 2.3(b), it then follows that x^* is an optimal solution of (10) if and only if there exists $y^* \in \mathbb{R}^n$ such that

$$0 \in \nabla f(x^*) + \partial h_1(x^*) + y^*, \qquad 0 \in \partial h_2^*(y^*) - x^*. \tag{13}$$

It is interesting to note that the above inclusion problem is associated with the Lagrangian $\mathcal{L} : \mathcal{X} \times \mathcal{X} \to [-\infty, \infty]$ defined as

$$\mathcal{L}(x, y) = f(x) + h_1(x) + \langle x, y\rangle - h_2^*(y), \quad \forall (x, y) \in \mathcal{X} \times \mathcal{X},$$

in that it can be simply expressed as

$$0 \in \partial_x \mathcal{L}(x, y), \quad 0 \in \partial_y(-\mathcal{L})(x, y), \tag{14}$$

where the two partial derivatives are with respect to the same inner product $\langle \cdot, \cdot \rangle$ on $\mathcal{X}$. Although one can apply the A-BD-HPE framework directly to the above system with $C = \partial(f + h_1)$ and $D = \partial h_2^*$, and $F(x, y) = (y, -x)$ for all $(x, y) \in \mathcal{X} \times \mathcal{X}$, it is more efficient from a computational point of view to introduce a scale factor to balance the two inclusions in (14).

Indeed, let $\theta > 0$ be given and consider the scaled inner product $\langle \cdot, \cdot \rangle_\theta$ in $\mathcal{X}$ defined as

$$\langle x, x'\rangle_\theta := \theta^{-1}\langle x, x'\rangle, \quad \forall x, x' \in \mathcal{X},$$

and observe that the associated inner product norm, denoted by $\| \cdot \|_\theta$, satisfies

$$\| \cdot \|_\theta = \frac{1}{\sqrt{\theta}} \| \cdot \|. \tag{15}$$

Also, denote the gradient and ε-subdifferential of an arbitrary function $\phi : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ with respect to $\langle \cdot, \cdot \rangle_\theta$ by $\nabla^\theta \phi$ and $\partial_\varepsilon^\theta \phi$, respectively. It is trivial to see that

$$\nabla^\theta \phi = \theta(\nabla \phi), \quad \partial_\varepsilon^\theta \phi = \theta(\partial_\varepsilon \phi). \tag{16}$$

It turns out that the monotone inclusion problem (14) is equivalent to

$$0 \in \partial_x^\theta \mathcal{L}(x, y), \quad 0 \in \partial_y(-\mathcal{L}(x, y)), \tag{17}$$

or equivalently,

$$\begin{aligned} &0 \in \theta(\nabla f(x) + \partial h_1(x) + y), \\ &0 \in \partial h_2^*(y) - x. \end{aligned} \tag{18}$$

We note that the use of (13), or more generally (18), as a way of solving (12) is well known (see for example the methods described in [4,9,11]).

The above system (17) is determined by $\mathcal{L}$ and the inner product norm on $\mathcal{X} \times \mathcal{X}$ defined as

$$\|(x, y)\|_{\theta,1} = \sqrt{\|x\|_\theta^2 + \|y\|^2}, \quad \forall (x, y) \in \mathcal{X} \times \mathcal{X}. \tag{19}$$

Note that this norm is the one given by (3) with $\mathcal{X} = \mathcal{Y}$, $\|\cdot\|_{\mathcal{X}} = \|\cdot\|_\theta$ and $\|\cdot\|_{\mathcal{Y}} = \|\cdot\|$. In order to view (17), or equivalently (18), as a special case of (3) and (4), the latter observation motivates us to define in this section $\mathcal{Y} := \mathcal{X}$, the inner products as

$$\langle \cdot, \cdot \rangle_{\mathcal{X}} := \langle \cdot, \cdot \rangle_\theta, \quad \langle \cdot, \cdot \rangle_{\mathcal{Y}} := \langle \cdot, \cdot \rangle, \tag{20}$$

and the operators $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$, $C : \mathcal{X} \rightrightarrows \mathcal{X}$, and $D : \mathcal{Y} \rightrightarrows \mathcal{Y}$ as

$$\begin{aligned} F(x, y) &:= (\theta y, -x), \quad C(x) := \partial^\theta (f + h_1)(x) = \theta(\nabla f(x) + \partial h_1(x)), \\ D(y) &:= \partial h_2^*(y). \end{aligned} \tag{21}$$

The following simple result summarizes the main properties of the scaled reformulation (18) (or equivalently, (17)) of (12).

Proposition 4.1 *The spaces $\mathcal{X}$ and $\mathcal{Y} := \mathcal{X}$ with corresponding inner products given by* (20)*, and the operators F, C and D defined by* (21)*, satisfy conditions A.1–A.4 with $L = \sqrt{\theta}$. Moreover, the inclusion problem* (18) *is equivalent to the maximal monotone inclusion problem* (4).

Our goal now will be to state an instance of the A-BD-HPE framework for solving (18), and hence (10), under the assumption that the resolvents of both ∂h_1 and ∂h_2, that is, the maps $(I + \lambda \partial h_i)^{-1}$ for every $\lambda > 0$ and $i = 1, 2$, can be easily evaluated at any given $x \in \mathcal{X}$. In other words, we assume that the optimal solutions of minimization subproblems of the form

$$\min_{\bar{x} \in \mathcal{X}} h_i(x) + \frac{1}{2\lambda} \|\bar{x} - x\|^2$$

can be easily computed for any $x \in \mathcal{X}$, $\lambda > 0$ and $i = 1, 2$.

Algorithm 1 : Scaled A-BD-HPE method for (10)

---

0) Let $(x^0, y^0) \in \mathcal{X} \times \mathcal{X}$, $\theta > 0$, $\sigma_1 \in (0, 1)$ and $\sigma \in [\sigma_1, 1]$ be given, and set $k = 1$ and

$$\tilde{\lambda} := \min \left\{ \frac{\sigma_1^2}{\theta L_f}, \frac{\sigma}{\sqrt{\theta}} \right\}; \tag{22}$$

1) set $\tilde{x}^k := \left(I + \tilde{\lambda}\theta \partial h_1\right)^{-1} \left(x^{k-1} - \tilde{\lambda}\theta \left(\nabla f(x^{k-1}) + y^{k-1}\right)\right)$;

2) set $\tilde{y}^k := (I + \tilde{\lambda} \partial h_2^*)^{-1}(y^{k-1} + \tilde{\lambda}\tilde{x}^k)$;

3) choose λ_k to be the largest $\lambda > 0$ such that

$$\|\lambda(v_1^k, v_2^k) + (\tilde{x}^k, \tilde{y}^k) - (x^{k-1}, y^{k-1})\|_{\theta,1}^2 + 2\lambda\varepsilon_k \le \sigma^2 \|(\tilde{x}^k, \tilde{y}^k) - (x^{k-1}, y^{k-1})\|_{\theta,1}^2,$$

where

$$v_1^k := \frac{1}{\tilde{\lambda}} \left(x^{k-1} - \tilde{x}^k\right) + \theta \left(\tilde{y}^k - y^{k-1}\right), \quad v_2^k := \frac{1}{\tilde{\lambda}}(y^{k-1} - \tilde{y}^k), \quad \varepsilon_k := \frac{L_f}{2}\|\tilde{x}^k - x^{k-1}\|^2; \tag{23}$$

4) set $(x^k, y^k) = (x^{k-1}, y^{k-1}) - \lambda_k(v_1^k, v_2^k)$ and $k \leftarrow k + 1$, and go to step 1.

We now make two remarks about Algorithm 1. First, when $L_f = 0$, it follows from (22) that $\tilde{\lambda} = \sigma/\sqrt{\theta}$ and hence that Algorithm 1 does not depend on the choice of σ_1. Second, the formula for computing $\tilde{y}^k$ in step 2 of Algorithm 1 involves the resolvent of ∂h_2^*, instead of ∂h_2. Using Lemma 2.1 with $T = \partial h_2^*$ and the fact that $(\partial h_2^*)^{-1} = \partial h_2$, it follows that $\tilde{y}^k$ can also be computed as

$$\tilde{y}^k = y^{k-1} + \tilde{\lambda}\tilde{x}^k - \tilde{\lambda}\left(I + \tilde{\lambda}^{-1}\partial h_2\right)^{-1}\left(\tilde{\lambda}^{-1}y^{k-1} + \tilde{x}^k\right). \tag{24}$$

Clearly, depending on the function h_2, one of these resolvents might be easier to compute than the other one, and hence is the better one for computing $\tilde{y}^k$. Using Lemma 2.1 with $T = \partial h_1$, it is also possible to give an expression for computing $\tilde{x}^k$ in terms of the resolvent of ∂h_1^*. Again, which one to use computationally will depend on the function h_1. We have chosen the formulae in steps 1 and 2 of Algorithm 1 due to their symmetry and the fact that they are more convenient for our theoretical presentation.

The following result shows that Algorithm 1 is a special instance of the A-BD-HPE framework applied to (17).

Lemma 4.2 *Consider the sequences* $\{(x^k, y^k)\}$, $\{(\tilde{x}^k, \tilde{y}^k)\}$, $\{(v_1^k, v_2^k)\}$ *and* $\{\varepsilon_k\}$ *generated by Algorithm* 1 *and, for every* $k \in \mathbb{N}$, *define*

$$\tilde{\lambda}_k := \tilde{\lambda}, \quad \varepsilon'_k := \varepsilon_k, \quad \varepsilon''_k := 0, \tag{25}$$

and

$$c^k := \frac{x^{k-1} - \tilde{x}^k}{\tilde{\lambda}} - \theta y^{k-1}, \quad d^k := \frac{y^{k-1} - \tilde{y}^k}{\tilde{\lambda}} + \tilde{x}^k. \tag{26}$$

Then, for every $k \in \mathbb{N}$, *the following statements hold with respect to the A-BD-HPE framework with*

$$\sigma_x = \sigma_1, \quad \tilde{\sigma}_x = 0, \quad \sigma_y = 0, \tag{27}$$

and the inner products $\langle\cdot,\cdot\rangle_{\mathcal{X}}$ *and* $\langle\cdot,\cdot\rangle_{\mathcal{Y}}$ *and operators* F, C *and* D *defined as in* (20) *and* (21)*:*

(a) $\tilde{\lambda}_k$ *satisfies* (5)*;*
(b) $\tilde{\lambda}_k$, x^{k-1} *and the triple* $(\tilde{x}^k, c^k, \varepsilon'_k)$ *satisfies* (6) *and* (7) *and*

$$\theta^{-1}c^k \in \nabla f(x^{k-1}) + \partial h_1(\tilde{x}^k) \subseteq (\partial_{\varepsilon'_k} f + \partial h_1)(\tilde{x}^k); \tag{28}$$

(c) $\tilde{\lambda}_k$, y^{k-1} *and the triple* $(\tilde{y}^k, d^k, \varepsilon''_k)$ *satisfies* (8)*, and*

$$d^k \in \partial h_2^*(\tilde{y}^k); \tag{29}$$

(d) $(v_1^k, v_2^k) = F(\tilde{x}^k, \tilde{y}^k) + (c^k, d^k)$.

As a consequence, Algorithm 1 *applied to* (10) *is a special instance of the A-BD-HPE framework for solving* (4) *where the inner products* $\langle\cdot,\cdot\rangle_{\mathcal{X}}$ *and* $\langle\cdot,\cdot\rangle_{\mathcal{Y}}$ *and operators* F, C *and* D *are given by* (20) *and* (21).

Proof Statement (a) follows immediately from condition (22), the definitions of $\sigma_x, \tilde{\sigma}_x, \sigma_y$ and $\tilde{\lambda}_k$ in (25) and (27), and the fact that $L = \sqrt{\theta}$ and $\sigma_1 \le \sigma$, in view of step 0 of Algorithm 1 and Proposition 4.1.

Now, it follows from (26) and the definitions of F and $\tilde{\lambda}_k$ in (21) and (25), respectively, that

$$\tilde{\lambda}_k[F_1(\tilde{x}^k, y^{k-1}) + c^k] + \tilde{x}^k - x^{k-1} = \tilde{\lambda}[\theta y^{k-1} + c^k] + \tilde{x}^k - x^{k-1} = 0$$

and

$$\tilde{\lambda}_k[F_2(\tilde{x}^k, \tilde{y}^k) + d^k] + \tilde{y}^k - y^{k-1} = \tilde{\lambda}[-\tilde{x}^k + d^k] + \tilde{y}^k - y^{k-1} = 0.$$

Clearly, these identities and the definition of ε''_k in (25) imply that $\tilde{\lambda}_k$, x^{k-1}, y^{k-1} and the triples $(\tilde{x}^k, c^k, \varepsilon'_k)$ and $(\tilde{y}^k, d^k, \varepsilon''_k)$ satisfy (7) and the inequality in (8). They also satisfy the inequality in (6) due to the fact that, the definitions of ε_k, $\tilde{\lambda}_k$, ε'_k and σ_x in (23), (25) and (27), and relations (15), (22) and (20), imply that

$$\begin{aligned} 2\tilde{\lambda}_k\varepsilon'_k &= 2\tilde{\lambda}\varepsilon_k = L_f\tilde{\lambda}\|\tilde{x}^k - x^{k-1}\|^2 \le \frac{\sigma_1^2}{\theta}\|\tilde{x}^k - x^{k-1}\|^2 = \sigma_x^2\|\tilde{x}^k - x^{k-1}\|_\theta^2 \\ &= \sigma_x^2\|\tilde{x}^k - x^{k-1}\|_{\mathcal{X}}^2. \end{aligned}$$

We will now show that the inclusions in (6) and (8) hold. It is well-known that Assumption B.2 implies that

$$f(\tilde{x}^k) - f(x^{k-1}) - \langle\nabla f(x^{k-1}), \tilde{x}^k - x^{k-1}\rangle \le \frac{L_f}{2}\|\tilde{x}^k - x^{k-1}\|^2 = \varepsilon_k = \varepsilon'_k,$$

where the last two equalities follow from the definitions of ε_k and ε'_k in (23) and (25), respectively. Using the last conclusion, the fact that $\nabla f(x^{k-1}) \in \partial f(x^{k-1})$, Proposition 2.3(c) with $v = \nabla f(x^{k-1})$, $z = x^{k-1}$ and $\tilde{z} = \tilde{x}^k$, we then conclude that $\nabla f(x^{k-1}) \in \partial_{\varepsilon'_k} f(\tilde{x}^k)$. Now, using the definition of $\tilde{x}^k$ in step 1 of Algorithm 1, c^k in (26) and C in (21), the last conclusion, relation (16), and Proposition 2.3(a), we conclude that

$$\begin{aligned} c^k &\in \theta[\nabla f(x^{k-1}) + \partial h_1(\tilde{x}^k)] \subseteq \theta(\partial_{\varepsilon'_k} f + \partial h_1)(\tilde{x}^k) \subseteq \theta\left[\partial_{\varepsilon'_k}(f + h_1)(\tilde{x}^k)\right] \\ &= \partial^\theta_{\varepsilon'_k}(f + h_1)(\tilde{x}^k) \subseteq [\partial^\theta(f + h_1)]^{\varepsilon'_k}(\tilde{x}^k) = C^{\varepsilon'_k}(\tilde{x}^k), \end{aligned}$$

which shows that (28) and the inclusion in (6) hold. Also, the definitions of $\tilde{y}^k$ in step 2 of Algorithm 1, d^k in (26), D in (21) and ε''_k in (25), and Proposition 2.2(d), imply that

$$d^k \in \partial h_2^*(\tilde{y}^k) = D(\tilde{y}^k) = D^{\varepsilon''_k}(\tilde{y}^k),$$

which shows that (29) and the inclusion in (8) hold. We have thus shown statements (b) and (c).

Statement (d) follows immediately from the definitions of F, v_1^k, v_2^k, c^k and d^k in (21), (23) and (26).

The last claim of the lemma immediately follows from statements (a)–(d) and the descriptions of Algorithm 1 and the A-BD-HPE framework. □

It follows from Lemma 4.2 that Algorithm 1 is a special instance of the A-BD-HPE framework. Hence, the convergence result described in Theorem 3.2 applies to it. In what follows, we will describe the implications of this result towards the behavior of Algorithm 1.

However, we first make some observations regarding the distance of the initial point (x^0, y^0) to the solution set of (17) with respect to the norm $\|(\cdot,\cdot)\|_{\theta,1}$. First observe that the solution sets of (14) and (17) are the same. Second, by the saddle-point theory, this set is of the form $X^* \times Y^* \subseteq \mathcal{X} \times \mathcal{X}$. Third, the distance d_0^θ of the initial point (x^0, y^0) to the solution set of (17) with respect to the norm $\|(\cdot,\cdot)\|_{\theta,1}$ can be expressed as

$$d_0^\theta := \sqrt{\theta^{-1} d_{x,0}^2 + d_{y,0}^2}, \tag{30}$$

where

$$d_{x,0} := \min\{\|x - x^0\| : x \in X^*\}, \quad d_{y,0} := \min\{\|y - y^0\| : y \in Y^*\}. \tag{31}$$

The following theorem shows how Algorithm 1 nearly solves the optimality conditions (13) (or equivalently, (12)).

Theorem 4.3 *Consider the sequences* $\{(x^k, y^k)\}$, $\{(\tilde{x}^k, \tilde{y}^k)\}$, $\{(v_1^k, v_2^k)\}$ *and* $\{\varepsilon_k\}$ *generated by Algorithm* 1 *under the assumption that* $\sigma < 1$ *and, for every* $k \in \mathbb{N}$*, define*

$$r_1^k := \theta^{-1} v_1^k + \nabla f(\tilde{x}^k) - \nabla f(x^{k-1}), \tag{32}$$

$$r_2^k := v_2^k = \frac{1}{\tilde{\lambda}}(y^{k-1} - \tilde{y}^k). \tag{33}$$

Then, for every $k \in \mathbb{N}$,

$$r_1^k \in \nabla f(\tilde{x}^k) + \partial h_1(\tilde{x}^k) + \tilde{y}^k, \quad r_2^k \in -\tilde{x}^k + \partial h_2^*(\tilde{y}^k), \tag{34}$$

and there exists $i \le k$ *such that*

$$\sqrt{\theta\|r_1^i\|^2 + \|r_2^i\|^2} \le \max\left\{\frac{1}{\sigma}, \frac{\sqrt{\theta}L_f}{\sigma_1^2}\right\}\left(\frac{1+\sigma+\sigma\min\left\{\sigma_1, \theta^{1/4}\sqrt{\sigma L_f}\right\}}{\sqrt{1-\sigma^2}}\right) \times \frac{\sqrt{\theta}}{\sqrt{k}}\sqrt{\theta^{-1}d_{x,0}^2 + d_{y,0}^2}.$$

Proof Consider the sequences $\{c^k\}$ and $\{d^k\}$ defined in (26). It follows from the definition of v_1^k and v_2^k in (23) that

$$\theta^{-1} v_1^k = \tilde{y}^k + \theta^{-1} c^k, \quad v_2^k = -\tilde{x}^k + d^k. \tag{35}$$

Hence, (34) follows from the above two identities, the first inclusion in (28), and relations (29), (32) and (33). Moreover, Lemma 4.2(d), the definitions of ε_k' and ε_k'' in (25), together with Theorem 3.2 imply the existence of $i \le k$ such that

$$\|(v_1^i, v_2^i)\|_{\theta,1} \le d_0^\theta \sqrt{\frac{1+\sigma}{1-\sigma}\left(\frac{1}{\lambda_i \sum_{j=1}^k \lambda_j}\right)} \le \sqrt{\frac{1+\sigma}{1-\sigma}}\,\frac{d_0^\theta}{\tilde{\lambda}\sqrt{k}}, \tag{36}$$

$$\varepsilon_i = \varepsilon_k' + \varepsilon_k'' \le \frac{\sigma^2 \left(d_0^\theta\right)^2}{2(1-\sigma^2)\sum_{j=1}^k \lambda_j} \le \frac{\sigma^2 \left(d_0^\theta\right)^2}{2(1-\sigma^2)\tilde{\lambda} k}, \tag{37}$$

where the last inequalities in (36) and (37) follow from Proposition 3.1 and the definition of $\tilde{\lambda}_k$ in (25). Moreover, using the definitions of $\|\cdot\|_\theta$, $\|(\cdot,\cdot)\|_{\theta,1}$, ε_k, r_1^k and r_2^k in (15), (19), (23), (32) and (33), respectively, the the triangular inequality for norms and (11), we conclude that

$$\begin{aligned}
\sqrt{\theta\|r_1^i\|^2 + \|r_2^i\|^2} &= \|(\theta r_1^i, r_2^i)\|_{\theta,1} = \left\|\left(v_1^i + \theta[\nabla f(\tilde{x}^i) - \nabla f(x^{i-1})], v_2^i\right)\right\|_{\theta,1} \\
&\le \|(v_1^i, v_2^i)\|_{\theta,1} + \theta\left\|\left(\nabla f(\tilde{x}^i) - \nabla f(x^{i-1}), 0\right)\right\|_{\theta,1} \\
&= \|(v_1^i, v_2^i)\|_{\theta,1} + \sqrt{\theta}\|\nabla f(\tilde{x}^i) - \nabla f(x^{i-1})\| \\
&\le \|(v_1^i, v_2^i)\|_{\theta,1} + \sqrt{\theta} L_f \|\tilde{x}^i - x^{i-1}\| = \|(v_1^i, v_2^i)\|_{\theta,1} + \sqrt{2\theta L_f \varepsilon_i}.
\end{aligned} \tag{38}$$

Now, combining (36), (37) and (38), we have

$$\sqrt{\theta\|r_1^i\|^2 + \|r_2^i\|^2} \le \left(\frac{1+\sigma+\sigma\sqrt{\theta\tilde{\lambda}L_f}}{\sqrt{1-\sigma^2}}\right)\frac{d_0^\theta}{\tilde{\lambda}\sqrt{k}},$$

which, together with (30) and the definition of $\tilde{\lambda}$ in (22), imply the last conclusion of the theorem. □

Note that the point-wise iteration-complexity bound in Theorem 4.3 is $\mathcal{O}(1/\sqrt{k})$. Appendix derives an $\mathcal{O}(1/k)$ iteration-complexity ergodic bound for Algorithm 1 as an immediate consequence of Theorem 3.3 of [11] and Theorem 2.4 of [12].

5 Specialization of Algorithm 1 to conic optimization

In this section, we discuss the specialization of Algorithm 1 to the context of conic optimization problems possessing a two-easy-block structure.

More specifically, let $\mathcal{X}$ be as in Sect. 4 and, for $i = 1, 2$, let $\mathcal{W}_i$ be an inner product space whose inner product and associated norm is denoted by $\langle \cdot, \cdot \rangle_{\mathcal{W}_i}$ and $\| \cdot \|_{\mathcal{W}_i}$. We consider the conic optimization problem of the form

$$
\begin{aligned}
z_P^* := \min \quad & \langle c, x \rangle \\
\text{s.t.} \quad & \mathcal{A}_1 x - b_1 \in \mathcal{K}_1 \\
& \mathcal{A}_2 x - b_2 \in \mathcal{K}_2,
\end{aligned} \tag{39}
$$

where $c \in \mathcal{X}, b_1 \in \mathcal{W}_1, b_2 \in \mathcal{W}_2, \mathcal{A}_1 : \mathcal{X} \to \mathcal{W}_1$ and $\mathcal{A}_2 : \mathcal{X} \to \mathcal{W}_2$ are linear maps, and $\mathcal{K}_1 \subseteq \mathcal{W}_1$ and $\mathcal{K}_2 \subseteq \mathcal{W}_2$ are nonempty closed convex cones. Observe that (39) is a special of (10) in which

$$
f(\cdot) = \langle c, \cdot \rangle, \qquad h_i(\cdot) = \delta_{\mathcal{M}_i}(\cdot) = \delta_{\mathcal{K}_i}(\mathcal{A}_i(\cdot) - b_i), \quad i = 1, 2, \tag{40}
$$

and

$$
\mathcal{M}_i := \{x \in \mathcal{X} : \mathcal{A}_i x - b_i \in \mathcal{K}_i\}, \quad i = 1, 2. \tag{41}
$$

Throughout this section, we make the following assumptions on (39):

(C.1) (39) has an optimal solution, and hence $z_P^* \in \mathbb{R}$;
(C.2) (39) has a Slater point, i.e., there exists $x \in \mathcal{X}$ such that $\mathcal{A}_i x - b_i \in \operatorname{ri} \mathcal{K}_i$ for $i = 1, 2$.

We will also need another assumption related to our ability to evaluate the resolvents $(I + \lambda \partial h_i)^{-1}, i = 1, 2$, at any given $x \in \mathcal{X}$. In the case of (39) with h_i defined as in (40), evaluating the resolvent of ∂h_i at x is equivalent to projecting x onto $\mathcal{M}_i$, i.e.,

$$
\begin{aligned}
(I + \lambda \partial h_i)^{-1}(x) \; &= \; \Pi_{\mathcal{M}_i}(x) \\
&:= \arg\min_{\tilde{x} \in \mathcal{X}} \left\{ \frac{1}{2} \|\tilde{x} - x\|^2 : \mathcal{A}_i \tilde{x} - b_i \in \mathcal{K}_i \right\}, \quad \forall \lambda > 0, \; \forall x \in \mathcal{X}.
\end{aligned} \tag{42}
$$

Observe that $(I + \lambda \partial h_i)^{-1}(x)$ does not depend on the value of λ. The optimality conditions of the optimization problem above, Assumption C.2, the fact that $\delta_{\mathcal{M}_i}(\cdot) = \delta_{\mathcal{K}_i}(\mathcal{A}_i(\cdot) - b_i)$ and the well-known chain rule property of the subdifferential imply that p_i is the optimal solution of (42) if and only if $p_i \in \mathcal{M}_i$ and

$$
p_i - x \in -\partial \delta_{\mathcal{M}_i}(p_i) = -\mathcal{A}_i^* \left[\left(\partial \delta_{\mathcal{K}_i} \right) (\mathcal{A}_i p_i - b_i) \right] = -\mathcal{A}_i^* N_{\mathcal{K}_i}(\mathcal{A}_i p_i - b_i),
$$

where $\mathcal{A}_i^*$ is the adjoint of $\mathcal{A}_i$ and $N_{\mathcal{K}_i}$ denotes the normal cone operator for $\mathcal{K}_i$. Hence, in view of the characterization of the normal cone, we conclude that for every $\xi > 0, x \in \mathcal{X}$ and $i = 1, 2$, p_i is the optimal solution of (42) if and only if there exists a dual variable $w_i \in \mathcal{W}_i$ such that

$$
\mathcal{A}_i p_i - b_i \in \mathcal{K}_i, \quad \mathcal{A}_i^* w_i = \xi (p_i - x), \quad w_i \in \mathcal{K}_i^*, \quad \langle w_i, \mathcal{A}_i p_i - b_i \rangle_{\mathcal{W}_i} = 0, \tag{43}
$$

where $\mathcal{K}_i^*$ is the dual cone of $\mathcal{K}_i$. We further assume that:

(C.3) for any given $\xi > 0, x \in \mathcal{X}$ and $i = 1, 2$, it is easy to compute a pair $(p_i, w_i) \in \mathcal{M}_i \times \mathcal{W}_i$ satisfying (43).

It should be observed that the application of Algorithm 1 to a conic programming instance strongly depends on the possibility of splitting its constraints into two blocks $\mathcal{M}_1$ and $\mathcal{M}_2$ as in (41) such that C.3 is satisfied. In this respect, the constraints of all instances used in the benchmarks of Sects. 7 and 8 can be partitioned so as to satisfy C.3 without the need of reformulating them. Another possibility of solving a general conic SDP instance (39) is to reformulate it in standard form (1) and apply Algorithm 1 with the partition given by the blocks $\mathcal{M}_1 = \mathcal{K}$ and $\mathcal{M}_2 = \mathcal{M} := \{x : \mathcal{A}x = b\}$. In fact, the latter approach is the one used by the DSA-BD method developed in [12].

The dual of (39) is the conic optimization problem given by

$$\begin{aligned} z_D^* := \max \;\; & \langle b_1, w_1\rangle_{\mathcal{W}_1} + \langle b_2, w_2\rangle_{\mathcal{W}_2} \\ \text{s.t.} \;\; & \mathcal{A}_1^* w_1 + \mathcal{A}_2^* w_2 = c, \\ & w_1 \in \mathcal{K}_1^*, \;\; w_2 \in \mathcal{K}_2^*. \end{aligned} \tag{44}$$

It is well-known that assumptions C.1 and C.2 imply that: (i) the dual of (39) has an optimal solution and $z_P^* = z_D^*$; and (ii) $x^* \in \mathcal{X}$ is an optimal solution of (39) and the pair $(w_1^*, w_2^*) \in \mathcal{W}_1 \times \mathcal{W}_2$ is an optimal solution of (44) if and only if

$$\begin{aligned} & c - \mathcal{A}_1^* w_1^* - \mathcal{A}_2^* w_2^* = 0, \\ & \mathcal{A}_i x^* - b_i \in \mathcal{K}_i, \quad w_i^* \in \mathcal{K}_i^*, \quad \langle w_i^*, \mathcal{A}_i x^* - b_i\rangle_{\mathcal{W}_i} = 0, \qquad i = 1, 2. \end{aligned}$$

For the sake of clarity we explicitly state below an specialization of Algorithm 1 to the context of (39), i.e., the special case of Algorithm 1 in which f, h_1 and h_2 are given by (40), $L_f = 0$ and the iterate $\tilde{y}^k$ in step 2 is computed using the alternative formula (24). In addition, steps 1 and 2 include the computation of a sequence of dual variables $\{(w_1^k, w_2^k)\} \subseteq \mathcal{K}_1^* \times \mathcal{K}_2^*$, which can be easily obtained in view of Assumption C.3.

Algorithm 2 : Scaled A-BD-HPE method for (39)

0) Let $(x^0, y^0) \in \mathcal{X} \times \mathcal{X}$, $\theta > 0$ and $\sigma \in (0, 1]$ be given, and set $k = 1$ and $\tilde{\lambda} := \sigma/\sqrt{\theta}$;

1) set $\tilde{x}^k := \Pi_{\mathcal{M}_1}\left(x^{k-1} - \tilde{\lambda}\theta\left(c + y^{k-1}\right)\right)$, or equivalently, compute a pair $(\tilde{x}^k, w_1^k) \in \mathcal{X} \times \mathcal{W}_1$ such that

$$\mathcal{A}_1\tilde{x}^k - b_1 \in \mathcal{K}_1, \quad \mathcal{A}_1^* w_1^k = \frac{\tilde{x}^k - x^{k-1}}{\tilde{\lambda}\theta} + c + y^{k-1}, \quad w_1^k \in \mathcal{K}_1^*, \quad \langle w_1^k, \mathcal{A}_1\tilde{x}^k - b_1\rangle_{\mathcal{W}_1} = 0; \tag{45}$$

2) compute $\tilde{u}^k = \Pi_{\mathcal{M}_2}\left(\tilde{\lambda}^{-1} y^{k-1} + \tilde{x}^k\right)$, or equivalently, a pair $(\tilde{u}^k, w_2^k) \in \mathcal{X} \times \mathcal{W}_2$ satisfying

$$\mathcal{A}_2\tilde{u}^k - b_2 \in \mathcal{K}_2, \quad \mathcal{A}_2^* w_2^k = \tilde{\lambda}(\tilde{u}^k - \tilde{x}^k) - y^{k-1}, \quad w_2^k \in \mathcal{K}_2^*, \quad \langle w_2^k, \mathcal{A}_2\tilde{u}^k - b_2\rangle_{\mathcal{W}_2} = 0, \tag{46}$$

and set $\tilde{y}^k = y^{k-1} + \tilde{\lambda}(\tilde{x}^k - \tilde{u}^k)$;

3) choose λ_k to be the largest $\lambda > 0$ such that

$$\|\lambda(v_1^k, v_2^k) + (\tilde{x}^k, \tilde{y}^k) - (x^{k-1}, y^{k-1})\|_{\theta,1} \le \sigma\|(\tilde{x}^k, \tilde{y}^k) - (x^{k-1}, y^{k-1})\|_{\theta,1}$$

where v_1^k and v_2^k are defined as in (23);

4) set $(x^k, y^k) = (x^{k-1}, y^{k-1}) - \lambda_k(v_1^k, v_2^k)$ and $k \leftarrow k + 1$, and go to step 1.

Observe that the condition in step 1 is equivalent to

$$\xi = (\tilde{\lambda}\theta)^{-1}, \quad x = x^{k-1} - \tilde{\lambda}\theta(c + y^{k-1}), \quad p_1 = \tilde{x}^k, \quad w_1 = w_1^k$$

satisfying (43) with $i = 1$. Moreover, the condition in step 2 is equivalent to

$$\xi = \tilde{\lambda}, \quad x = \tilde{\lambda}^{-1} y^{k-1} + \tilde{x}^k, \quad p_2 = \tilde{u}^k = \tilde{x}^k + (y^{k-1} - \tilde{y}^k)/\tilde{\lambda}, \quad w_2 = w_2^k$$

satisfying (43) with $i = 2$.

The following result specializes Theorem 4.3 to the context of (39) and also shows how Algorithm 2 solves the dual problem (44) in the limit.

Theorem 5.1 *Consider the sequences* $\{(\tilde{x}^k, \tilde{y}^k)\}$, $\{(x^k, y^k)\}$, $\{(w_1^k, w_2^k)\}$, $\{\tilde{u}^k\}$ *and* $\{(v_1^k, v_2^k)\}$ *generated by Algorithm 2 with* $\sigma < 1$ *and, for every* $k \in \mathbb{N}$, *define*

$$r^k := c - \mathcal{A}_1^* w_1^k - \mathcal{A}_2^* w_2^k. \tag{47}$$

Then, Algorithm 2 *is a special case of Algorithm* 1 *with* f, h_1 *and* h_2 *given by* (40) *and* $L_f = 0$. *Moreover, for every* $k \in \mathbb{N}$, *in addition to* (45) *and* (46), *the following statements hold:*

(a) $v_1^k = \theta r^k$ *and* $v_2^k = \tilde{u}^k - \tilde{x}^k$;

(b) *the duality gap* $dg_k := \langle c, \tilde{x}^k\rangle - \langle b_1, w_1^k\rangle_{\mathcal{W}_1} - \langle b_2, w_2^k\rangle_{\mathcal{W}_2}$ *can be alternatively computed as*

$$dg_k = \langle r^k, \tilde{x}^k\rangle + \langle v_2^k, \tilde{y}^k\rangle;$$

(c) *there exists* $i \le k$ *such that*

$$\max\left\{\sqrt{\theta}\|r^k\|,\ \|v_2^k\|\right\} \le \frac{\sqrt{\theta}}{\sigma\sqrt{k}}\sqrt{\left(\frac{1+\sigma}{1-\sigma}\right)\left(\theta^{-1}d_{x,0}^2 + d_{y,0}^2\right)},$$

where $d_{x,0}$ *and* $d_{y,0}$ *are defined in* (31).

Proof The first part of the theorem follows from (24), (40) and (42). To show (a), note that the definition of $\tilde{y}^k$ in step 2 of Algorithm 2 and the definition of v_2^k in (23) imply that $v_2^k = \tilde{u}^k - \tilde{x}^k$. Moreover, in view of the definition of v_1^k in (23), the second relations in (45) and (46), and the definition of $\tilde{y}^k$ in step 2 of Algorithm 2, we have

$$\mathcal{A}_1^* w_1^k = c - \theta^{-1} v_1^k + \tilde{y}^k, \quad \mathcal{A}_2^* w_2^k = -\tilde{y}^k. \tag{48}$$

The first identity in (a) follows from the definition of r^k and the above two relations. To show (b), note that the definition of r^k and the last relations in (45) and (46) imply that

$$
\begin{aligned}
&\langle c, \tilde{x}^k \rangle - \langle b_1, w_1^k \rangle_{\mathcal{W}_1} + \langle b_2, w_2^k \rangle_{\mathcal{W}_2} \\
&\quad = \langle r^k + \mathcal{A}_1^* w_1^k + \mathcal{A}_2^* w_2^k, \tilde{x}^k \rangle - \langle \tilde{x}^k, \mathcal{A}_1^* w_1^k \rangle - \langle \tilde{u}^k, \mathcal{A}_2^* w_2^k \rangle \\
&\quad = \langle r^k, \tilde{x}^k \rangle - \langle \tilde{u}^k - \tilde{x}^k, \mathcal{A}_2^* w_2^k \rangle = \langle r^k, \tilde{x}^k \rangle + \langle v_2^k, \tilde{y}^k \rangle,
\end{aligned}
$$

where the last equality follows from the second identities in (48) and (a). Finally, (c) follows from Theorem 4.3 with f, h_1 and h_2 given by (40) and $L_f = 0$, and the fact that r_1^k and r_2^k defined in (32) and (33) are equal to r^k and v_2^k, respectively, in view of (a) and the fact that $\nabla f(\cdot) = c$. □

We now make some observations about Algorithm 2 and Theorem 5.1. First, Theorem 5.1 shows that $\tilde{x}^k$ and its perturbation $\tilde{u}^k = \tilde{x}^k + v_2^k$ satisfy the first and second blocks $\mathcal{A}_1 x - b_1 \in \mathcal{K}_1$ and $\mathcal{A}_2 x - b_2 \in \mathcal{K}_2$, respectively. Second, Algorithm 2 can be implemented without generating the dual sequence $\{(w_1^k, w_2^k)\}$. In such case, (a) and (b) of Theorem 5.1 show that the quantities $c - \mathcal{A}_1^* w_1^k - \mathcal{A}_2^* w_2^k$ and $\langle c, \tilde{x}^k \rangle - \langle b_1, w_1^k \rangle_{\mathcal{W}_1} - \langle b_2, w_2^k \rangle_{\mathcal{W}_2}$, commonly used to monitor the progress of algorithms for solving (39) and (44), can be computed in terms of $\tilde{x}^k$ and $\tilde{y}^k$ only, and hence do not require (w_1^k, w_2^k). Third, Theorem 5.1(c) sheds light on how the scaling parameter θ might affect the sizes of the residuals $r^k = c - \mathcal{A}_1^* w_1^k - \mathcal{A}_2^* w_2^k$ and $v_2^k = \tilde{u}^k - \tilde{x}^k$. Roughly speaking, viewing all quantities in the bound of Theorem 5.1(c), with the exception of θ, as constants, we see that

$$
\|r^k\| = \mathcal{O}\left(\max\left\{1, \theta^{-1/2}\right\}\right), \quad \|v_2^k\| = \mathcal{O}\left(\max\left\{1, \theta^{1/2}\right\}\right).
$$

Hence, the ratio $\|v_2^k\|/\|r^k\|$ can grow significantly as $\theta \to \infty$, while it can become very small as $\theta \to 0$. This suggests that this ratio increases (resp., decreases) as θ increases (resp., decreases). In fact, we have observed in our computational experiments that this ratio behaves just as described.

6 A practical dynamically scaled BD method

In this section, we describe three measures that quantify the optimality of an approximate solution of (39), namely: the primal infeasibility measure; the dual infeasibility measure; and the relative duality gap. We also describe two important refinements of Algorithm 2, whose goal is to balance the magnitudes of the primal and dual infeasibility measures. More specifically, we describe: (i) a scheme for choosing the initial scaling parameter θ; and (ii) a procedure to dynamically update the scaling parameter θ for balancing the sizes of the primal and dual infeasibility measures as the algorithm progresses.

Let $\mathcal{X}$, $\mathcal{W}_1$ and $\mathcal{W}_2$ be as in Sect. 5. For the purpose of describing a stopping criterion for Algorithm 2, for $i = 1, 2$, denote the distance of a point $w \in \mathcal{W}_i$ to the cone $\mathcal{K}_i$ as

$$
d_i(w) := \min\{\|w - \tilde{w}\|_{\mathcal{W}_i} : \tilde{w} \in \mathcal{K}_i\} \quad \forall w \in \mathcal{W}_i,
$$

and the primal infeasibility measure as

$$\epsilon_P(x) := \frac{\sqrt{d_1(\mathcal{A}_1 x - b_1)^2 + d_2(\mathcal{A}_2 x - b_2)^2}}{1 + \sqrt{\|b_1\|^2_{\mathcal{W}_1} + \|b_2\|^2_{\mathcal{W}_2}}} \quad \forall x \in \mathcal{X}. \tag{49}$$

Also, define the dual infeasibility measure as

$$\epsilon_D(w_1, w_2) := \frac{\|c - \mathcal{A}_1^* w_1 - \mathcal{A}_2^* w_2\|}{\|c\| + 1} \quad \forall (w_1, w_2) \in \mathcal{W}_1 \times \mathcal{W}_2. \tag{50}$$

Finally, define the relative duality gap as

$$\epsilon_G(x, w_1, w_2) := \frac{\langle c, x\rangle - \langle b_1, w_1\rangle_{\mathcal{W}_1} - \langle b_2, w_2\rangle_{\mathcal{W}_2}}{|\langle c, x\rangle| + |\langle b_1, w_1\rangle + \langle b_2, w_2\rangle| + 1}$$
$$\forall x \in \mathcal{X},\ \forall (w_1, w_2) \in \mathcal{W}_1 \times \mathcal{W}_2. \tag{51}$$

For given tolerances $\bar{\epsilon}, \bar{\nu} > 0$, we stop Algorithm 2 whenever

$$\max\{\epsilon_{P,k}, \epsilon_{D,k}\} \le \bar{\epsilon}, \quad |\epsilon_{G,k}| \le \bar{\nu}, \tag{52}$$

where

$$\epsilon_{P,k} := \epsilon_P(\tilde{x}^k), \quad \epsilon_{D,k} := \epsilon_D(w_1^k, w_2^k), \quad \epsilon_{G,k} := \epsilon_G(\tilde{x}^k, w_1^k, w_2^k).$$

We now make some observations about the stopping criteria (52). First, in view of Theorem 5.1, the first inclusion in (45) and the definition of r^k in (47), we have that

$$\epsilon_{P,k} = \frac{d_2(\mathcal{A}_2 \tilde{x}^k - b_2)}{1 + \sqrt{\|b_1\|^2_{\mathcal{W}_1} + \|b_2\|^2_{\mathcal{W}_2}}}, \quad \epsilon_{D,k} = \frac{\|r^k\|}{\|c\| + 1},$$
$$\epsilon_{G,k} = \frac{\langle r^k, \tilde{x}^k\rangle + \langle v_2^k, \tilde{y}^k\rangle}{|\langle c, \tilde{x}^k\rangle| + |\langle r^k - c, \tilde{x}^k\rangle + \langle v_2^k, \tilde{y}^k\rangle| + 1}. \tag{53}$$

Second, since Theorem 5.1, (45) and (46) imply that zero is a cluster value of the sequences $\{\epsilon_{P,k}\}$, $\{\epsilon_{D,k}\}$ and $\{\epsilon_{G,k}\}$ as $k \to \infty$, Algorithm 2 with the termination criteria (52) will eventually terminate. Third, another possibility is to terminate Algorithm 2 based on the quantities $\epsilon'_{P,k} = \epsilon_P(\tilde{u}^k), \epsilon_{D,k}$ and $\epsilon'_{G,k} := \epsilon_G(\tilde{u}^k, w_1^k, w_2^k)$, which also approach zero (in a cluster sense) due to Theorem 5.1. Our current implementation of Algorithm 2 ignores the latter possibility and terminates based on (52). Fourth, the above termination criteria do not contain a violation measure with respect to the constraint $(w_1, w_2) \in \mathcal{K}_1^* \times \mathcal{K}_2^*$. In fact, our benchmarks of Sects. 7 and 8 disregard this measure due to the fact that all the codes tested generate the sequence $\{(w_1^k, w_2^k)\}$ inside the cone $\mathcal{K}_1^* \times \mathcal{K}_2^*$. Finally, Theorem 5.1(a) and the first inclusion in (46) imply that

$$\epsilon_{P,k} \le \frac{\|(\mathcal{A}_2\tilde{x}^k - b_2) - (\mathcal{A}_2\tilde{u}^k - b_2)\|_{\mathcal{W}_2}}{1 + \sqrt{\|b_1\|^2_{\mathcal{W}_1} + \|b_2\|^2_{\mathcal{W}_2}}} = \frac{\|\mathcal{A}_2 v_2^k\|_{\mathcal{W}_2}}{1 + \sqrt{\|b_1\|^2_{\mathcal{W}_1} + \|b_2\|^2_{\mathcal{W}_2}}}. \quad (54)$$

We now discuss two important refinements of Algorithm 2 whose goal is to balance the magnitudes of the primal and dual infeasibility measures $\epsilon_{P,k}$ and $\epsilon_{D,k}$. First note that (53) and (54) imply that $\epsilon_{P,k}/\epsilon_{D,k} = \mathcal{O}(\|v_2^k\|/\|r^k\|)$. Hence, in view of the last observation in the paragraph following Theorem 5.1, the latter ratio can grow significantly as $\theta \to \infty$, while it can become very small as $\theta \to 0$. This suggests that this ratio increases (resp., decreases) as θ increases (resp., decreases). Indeed, our computational experiments indicate that the ratio $\epsilon_{P,k}/\epsilon_{D,k}$ behaves in this manner.

In the following, let θ_k denote the dynamic value of θ at the kth iteration of Algorithm 2. Observe that, in view of (53) and (23), the measures $\epsilon_{P,k}$ and $\epsilon_{D,k}$ depend on $\tilde{x}^k$ and $\tilde{y}^k$, whose values in turn depend on the choice of θ_k, in view of steps 1 and 2 of Algorithm 2. Hence, these two measures are indeed functions of θ, which are denoted as $\epsilon_{P,k}(\theta)$ and $\epsilon_{D,k}(\theta)$.

We first describe a scheme for choosing the initial scaling parameter θ_1. Let a constant $\rho > 1$ be given and set $\theta = 1$. If $\epsilon_{P,1}(\theta)/\epsilon_{D,1}(\theta) > \rho$ (resp., $\epsilon_{P,1}(\theta)/\epsilon_{D,1}(\theta) < \rho^{-1}$), we successively divide (resp., successively multiply) the current value of θ by 2 until $\epsilon_{P,1}(\theta)/\epsilon_{D,1}(\theta) \le \rho$ (resp., $\epsilon_{P,1}(\theta)/\epsilon_{D,1}(\theta) \ge \rho^{-1}$) is satisfied, and set $\theta_1 = \theta_1^*$, where θ_1^* is the last value of θ. Since there is no guarantee that this procedure will terminate, we specify an upper bound on the number of times that θ can be updated. In our implementation, we set this upper bound to be 20.

We next describe a procedure for dynamically updating the scaling parameter θ so as to balance the sizes of the two measures $\epsilon_{P,k}(\theta)$ and $\epsilon_{D,k}(\theta)$. As in [12], we use the heuristic of changing θ every time a specified number of iterations have been performed. More specifically, given an integer $\bar{k} \ge 1$, and scalars $\gamma > 1$ and $0 < \tau < 1$, we use the following dynamic scaling procedure for updating θ_k,

$$\theta_k = \begin{cases} \theta_{k-1}, & k \not\equiv 0 \mod \bar{k} \text{ or } \gamma^{-1} \le \bar{\epsilon}_{P,k-1}/\bar{\epsilon}_{D,k-1} \le \gamma \\ \tau^2\theta_{k-1}, & k \equiv 0 \mod \bar{k} \text{ and } \bar{\epsilon}_{P,k-1}/\bar{\epsilon}_{D,k-1} > \gamma \\ \tau^{-2}\theta_{k-1}, & k \equiv 0 \mod \bar{k} \text{ and } \bar{\epsilon}_{P,k-1}/\bar{\epsilon}_{D,k-1} < \gamma^{-1} \end{cases} \quad \forall k \ge 2, \quad (55)$$

where

$$\bar{\epsilon}_{P,k-1} = \left(\prod_{i=k-\bar{k}}^{k-1} \epsilon_{P,i}\right)^{1/\bar{k}}, \quad \bar{\epsilon}_{D,k-1} = \left(\prod_{i=k-\bar{k}}^{k-1} \epsilon_{D,i}\right)^{1/\bar{k}} \quad \forall k > \bar{k}. \quad (56)$$

Roughly speaking, the above dynamic scaling procedure changes the value of θ at most a single time in the right direction so as to balance the sizes of the residuals based on the information provided by their values at the previous $\bar{k}$ iterations. We observe that a dynamic scaling procedure using $\epsilon_{P,k-1}$ and $\epsilon_{D,k-1}$ in place of $\bar{\epsilon}_{P,k-1}$ and $\bar{\epsilon}_{D,k-1}$ in (55), respectively, is proposed in [12]. However, the more conservative procedure based on the aggregated measures in (56) have performed better in our computational experiments.

In our computational experiments, we will refer to the variant of Algorithm 2 which incorporates the two aforementioned schemes as the *two-easy-block-decomposition*

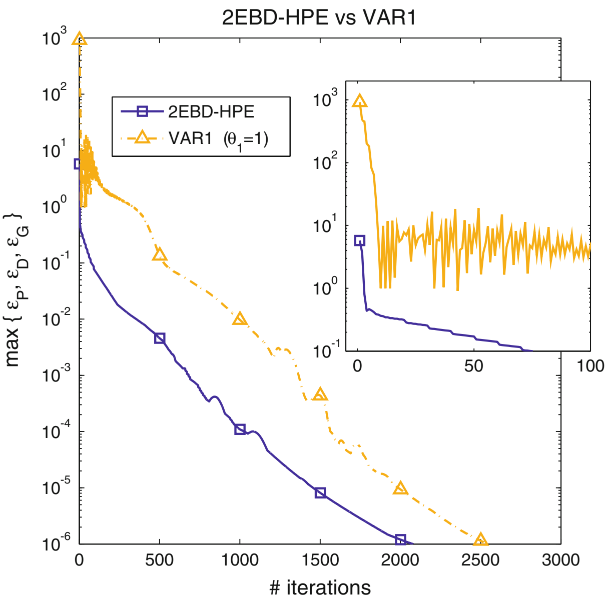


Fig. 1 This example (BIQ-be200.8.8) illustrates how the scheme for choosing the initial scaling parameter θ_1 can help Algorithm 2 to start with an error at least 2 orders of magnitude smaller

HPE (2EBD-HPE) method. To illustrate the importance of the above two schemes, we have chosen an instance of a conic optimization problem to compare the performance of 2EBD-HPE against the performance of its two variants obtained by removing exactly one of the two schemes. Indeed, Fig. 1 compares the performance of 2EBD-HPE against its variant VAR1 in which θ_1 is initialized as 1 instead of θ_1^*. Figure 2 compares the performance of 2EBD-HPE against its variant VAR2 in which dynamic scaling is removed (i.e., θ_k set to θ_1^* , for every $k \geq 1$).

In addition, to illustrate the importance of adaptively choosing the stepsize λ_k in Algorithm 2, Fig. 3 compares the performance of 2EBD-HPE against its variant VAR3 in which the stepsize λ_k is chosen as $\tilde{\lambda} = \sigma/\sqrt{\theta_k}$ for every $k \geq 1$.

Finally, Fig. 4 compares the performance of 2EBD-HPE against the following three variants: (i) VAR2, namely, the one that removes the dynamic scaling (i.e., set $\theta_k = \theta_1^*$, for every $k \geq 1$); (ii) VAR4, namely, the one that removes the dynamic scaling and the initialization scheme for θ_1 (i.e., set $\theta_k = 1$, for every $k \geq 1$); and (iii) VAR5, namely, the one that removes these latter two refinements and the use of adaptive stepsize (i.e., set $\theta_k = 1$ and $\lambda_k = \tilde{\lambda} = \sigma$, for every $k \geq 1$).

7 Numerical results: part I

In this section, we compare the 2EBD-HPE method described in Sect. 6 with a variant of the boundary point method, namely SDPAD, presented in [21]. More specifically, we compare these two methods on four important classes of graph-related SDP problems, namely: SDP relaxations of binary integer quadratic (BIQ) and frequency assignment (FAP) problems, and SDPs for θ-functions and θ_+-functions of graph stable set

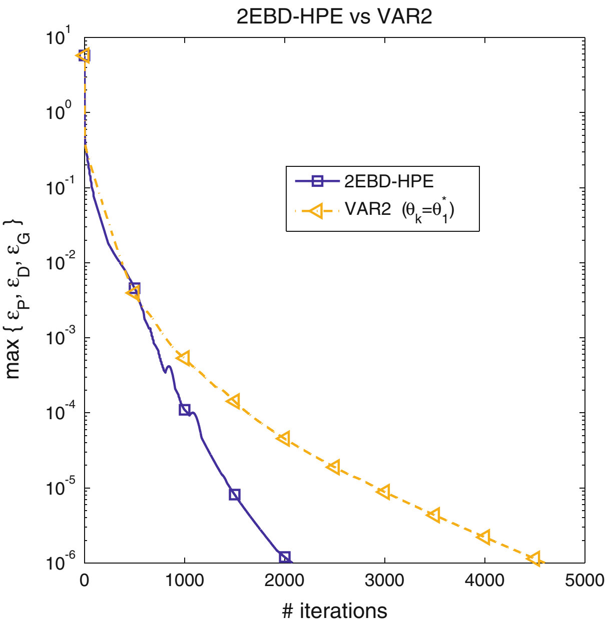


Fig. 2 This example (BIQ-be200.8.8) illustrates how the dynamic scaling improves the performance of Algorithm 1 considerably

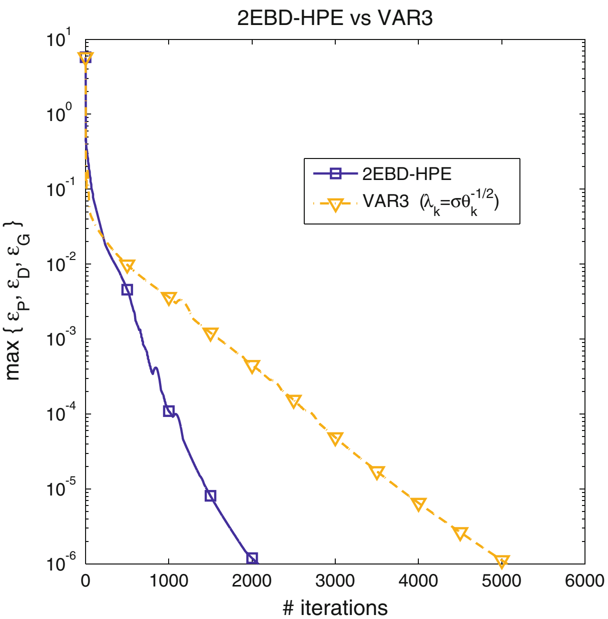


Fig. 3 This example (BIQ-be200.8.8) illustrates how the adaptive stepsize improves the performance of Algorithm 1 considerably

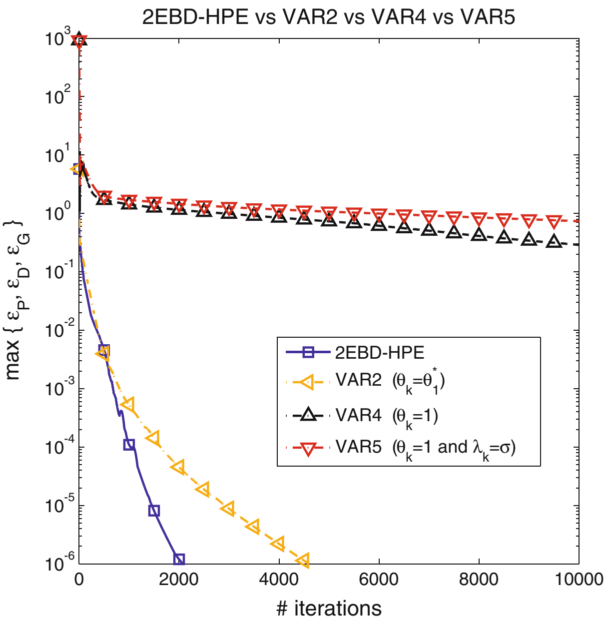


Fig. 4 This example (BIQ-be200.8.8) illustrates how all the refinements made in the application of the BD-HPE framework to conic optimization helped improve the performance of the algorithm

problems. This section contains three subsections. The first subsection considers SDP relaxations of BIQ problems, the second one deals with SDP relaxations of FAPs, and the third one discusses SDPs corresponding to the θ-functions and θ_+-functions of graph stable set problems.

For the four problem classes above, $\mathcal{X}$, $\mathcal{W}_1$ and $\mathcal{W}_2$ are Cartesian products of Euclidean spaces and/or spaces of symmetric matrices, which are endowed with the natural canonical inner product consisting of the sum of Euclidean and/or Frobenius inner products associated with the spaces comprising the products.

We have implemented 2EBD-HPE for solving (39) in a MATLAB code. We have used the beta2 MATLAB implementation of SDPAD[1] released on December, 2012. The computational results for all the conic SDP instances were obtained on a single core of a server with 2 Xeon X5520 processors at 2.27 GHz and 48 GB RAM. For each one of the above classes of conic SDP problems, both methods generate primal and dual sequences $\{\tilde{x}^k\}$ and $\{(w_1^k, w_2^k)\} \subseteq \mathcal{K}_1^* \times \mathcal{K}_1^*$, and stop whenever (52) with $\bar{\epsilon} = 10^{-6}$ and $\bar{\nu} = 10^{-5}$ is satisfied.

For all classes of conic SDP problems considered the sequence $\{\tilde{x}^k\}$ lies in $\mathcal{S}^n$ for some $n \geq 1$, and evaluation of $\epsilon_{P,k}$ requires the computation of the distance from $\tilde{x}^k$ to $\mathcal{S}^n_+$, which in turn requires an eigenvalue decomposition of $\tilde{x}^k$. The 2EBD-HPE method has the nice feature that it generates $\{\tilde{x}^k\}$ inside $\mathcal{S}^n_+$. On the other hand, we have observed that SDPAD may generate elements of the sequence $\{\tilde{x}^k\}$ outside $\mathcal{S}^n_+$, but that this sequence eventually approaches $\mathcal{S}^n_+$ as $k \to \infty$ (as proved in Subsection 3.3 of

[1] Available at http://math.sjtu.edu.cn/faculty/zw2109/code/SDPAD-release-beta2.zip.

[21]). For the purpose of this benchmark, we have assumed that SDPAD generates $\tilde{x}^k$ inside $\mathcal{S}^n_+$ so as to avoid computing an extra eigenvalue decomposition in the evaluation $\epsilon_{P,k}$ at every iteration.

We now make some general remarks about how the results are reported on the tables given below. Tables 1, 3, 5 and 7 compare 2EBD-HPE against SDPAD (each one of these tables corresponds to one of the four problem classes considered). The time (in seconds) taken by any of the two methods for any particular instance is marked in italics, and also with an asterisk (*), whenever it cannot solve the instance by the required accuracy, in which case the residual (i.e., the maximum between the infeasibility measures and the relative duality gap) reported is the one obtained at the last iteration of the method. Also, the times marked in bold in a row is the best one among the ones listed in that row with the convention that, when a method cannot solve the instance, the corresponding time is assumed to be ∞. Tables 2, 4, 6 and 8 report more detailed computational results obtained at the last iteration of 2EBD-HPE, such as the primal and dual objective function values, number of iterations, the primal and dual infeasibility measures and the relative duality gaps as described in (49), (50) and (51), respectively (each one of these tables corresponds to one of the four problem classes considered).

Finally, Figs. 5, 6, 7 and 8 plot the performance profiles (see [5]) of 2EBD-HPE and SDPAD methods for each of the four problem classes. We recall the following definition of a performance profile. For a given instance, a method A is said to be at most x times slower than method B, if the number of iterations performed by method A is at most x times the number of iterations performed by method B. A point (x, y) is in the performance profile curve of a method if it can solve exactly $(100y)$ % of all the tested instances x times slower than any other competing method.

7.1 Binary integer quadratic problems

This subsection gives more details of our implementation of 2EBD-HPE for solving SDP relaxations of BIQ problems and summarizes its computational performance against SDPAD on a collection of 134 such instances.

The SDP relaxation of the BIQ problem can be described as follows (see for example Section 7 in [22]). Given an $n \times n$ symmetric matrix Q, the BIQ problem can be formulated as

$$\min\{z^T Qz : z \in \{0, 1\}^n\}.$$

By representing the binary set $\{0, 1\}^n$ as $\{z \in \mathbb{R}^n | z_i^2 - z_i = 0\}$, we obtain the following SDP relaxation

$$\begin{aligned} \min\ & Q \bullet Z \\ \text{s.t. } & x := \begin{bmatrix} Z & z \\ z^T & \alpha \end{bmatrix} \succeq 0, && \text{(57a)} \\ & \operatorname{diag}(Z) - z = 0,\ \alpha = 1,\ Z \geq 0,\ z \geq 0, && \text{(57b)} \end{aligned}$$

where $Z \in \mathcal{S}^n$, $z \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$.

Table 1 Comparison of the methods on BIQ problems

Problem		$\max\{\epsilon_P, \epsilon_D\}$		ϵ_G		Time	
Instance	n_s—m	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD
be100.1	101—5252	9.69 −7	9.96 −7	−1.17 −6	−2.38 −7	**8.1**	10.2
be100.10	101—5252	9.99 −7	1.00 −6	+5.75 −7	−4.82 −7	**7.1**	8.6
be100.2	101—5252	1.00 −6	9.98 −7	+1.23 −7	+3.44 −9	**6.9**	13.7
be100.3	101—5252	9.98 −7	9.99 −7	+2.53 −7	−1.33 −7	**9.6**	11.7
be100.4	101—5252	9.98 −7	9.99 −7	−1.32 −7	−2.86 −7	**10.4**	19.7
be100.5	101—5252	1.00 −6	9.99 −7	+2.93 −7	−6.37 −7	**7.2**	10.7
be100.6	101—5252	1.00 −6	9.98 −7	−5.03 −7	−5.47 −7	**8.1**	13.8
be100.7	101—5252	9.91 −7	9.99 −7	−1.08 −6	−1.40 −7	**7.5**	11.8
be100.8	101—5252	9.95 −7	9.89 −7	−9.80 −7	+5.54 −7	**7.3**	8.7
be100.9	101—5252	9.95 −7	9.99 −7	−2.56 −7	−2.33 −7	**7.4**	13.0
be120.3.1	121—7502	9.99 −7	9.97 −7	−2.27 −7	−3.99 −7	**12.3**	18.6
be120.3.10	121—7502	9.96 −7	1.00 −6	−4.58 −7	−6.97 −7	**9.8**	16.6
be120.3.2	121—7502	9.98 −7	9.99 −7	−6.31 −7	−1.08 −7	**13.6**	23.8
be120.3.3	121—7502	9.96 −7	9.98 −7	−5.51 −7	−8.88 −7	**11.2**	14.3
be120.3.4	121—7502	9.95 −7	9.98 −7	−1.23 −6	−2.02 −6	**12.2**	14.6
be120.3.5	121—7502	1.00 −6	1.00 −6	+1.91 −9	+2.25 −8	**27.1**	35.1
be120.3.6	121—7502	9.98 −7	9.98 −7	−4.74 −7	−2.67 −7	**15.4**	21.6
be120.3.7	121—7502	1.00 −6	9.99 −7	−1.33 −7	−1.75 −7	**27.6**	46.1
be120.3.8	121—7502	1.00 −6	1.00 −6	−2.17 −7	−1.22 −7	**20.1**	32.1
be120.3.9	121—7502	1.00 −6	1.00 −6	−2.58 −7	−4.54 −7	**18.5**	54.2
be120.8.1	121—7502	9.95 −7	9.96 −7	+7.08 −7	−2.38 −7	**8.9**	13.1
be120.8.10	121—7502	9.97 −7	1.00 −6	−1.87 −7	−4.31 −7	**10.7**	19.6
be120.8.2	121—7502	1.00 −6	9.99 −7	−1.87 −7	−1.71 −7	**19.8**	35.3
be120.8.3	121—7502	9.96 −7	9.98 −7	−3.19 −8	−3.89 −7	**11.5**	15.5
be120.8.4	121—7502	9.98 −7	1.00 −6	−2.02 −7	−4.79 −8	**14.2**	24.1
be120.8.5	121—7502	9.96 −7	9.99 −7	−3.26 −8	−2.07 −8	**12.1**	21.2
be120.8.6	121—7502	1.00 −6	9.97 −7	−1.66 −7	−3.36 −7	**11.6**	21.0
be120.8.7	121—7502	9.97 −7	9.94 −7	+3.12 −8	−4.51 −7	11.7	**11.5**
be120.8.8	121—7502	9.98 −7	1.00 −6	−3.22 −7	−4.03 −7	**9.4**	11.7
be120.8.9	121—7502	9.99 −7	9.97 −7	+3.53 −9	−2.13 −7	**10.0**	12.6
be150.3.1	151—11627	9.97 −7	9.98 −7	−8.52 −7	−6.45 −7	**22.3**	26.0
be150.3.10	151—11627	9.99 −7	9.99 −7	−3.00 −7	−1.34 −7	**30.0**	63.5
be150.3.2	151—11627	9.98 −7	9.98 −7	−5.18 −7	−4.05 −7	**21.4**	33.7
be150.3.3	151—11627	9.95 −7	1.00 −6	−6.71 −7	−3.33 −7	**18.8**	25.0
be150.3.4	151—11627	9.97 −7	9.98 −7	−5.55 −7	−2.94 −7	**22.3**	32.4
be150.3.5	151—11627	1.00 −6	9.94 −7	−1.33 −8	−6.97 −8	**27.6**	29.7
be150.3.6	151—11627	9.98 −7	1.00 −6	−4.99 −7	−1.06 −7	**20.3**	29.1
be150.3.7	151—11627	9.98 −7	9.97 −7	−4.94 −7	−5.11 −7	**21.2**	29.4
be150.3.8	151—11627	9.99 −7	1.00 −6	−1.01 −7	−1.56 −7	**26.6**	34.1
be150.3.9	151—11627	9.96 −7	9.98 −7	−7.71 −7	−8.63 −7	**12.3**	18.8

Table 1 continued

Problem		max{ϵ_P, ϵ_D}		ϵ_G		Time	
Instance	n_s—m	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD
be150.8.1	151—11627	9.95 −7	9.99 −7	−2.49 −7	+2.11 −7	**15.9**	20.3
be150.8.10	151—11627	9.99 −7	9.99 −7	−2.07 −7	−1.79 −7	**22.2**	30.5
be150.8.2	151—11627	9.98 −7	9.98 −7	−9.87 −7	−4.89 −7	**16.5**	23.6
be150.8.3	151—11627	9.98 −7	9.98 −7	−1.91 −7	−4.82 −7	**19.9**	26.7
be150.8.4	151—11627	9.98 −7	9.99 −7	−3.01 −7	−2.38 −7	**20.3**	37.9
be150.8.5	151—11627	9.95 −7	1.00 −6	−5.86 −7	−4.96 −7	**22.3**	30.7
be150.8.6	151—11627	9.61 −7	1.00 −6	−1.52 −7	−1.70 −8	**26.8**	47.1
be150.8.7	151—11627	9.98 −7	9.99 −7	−8.41 −7	−2.00 −7	**28.3**	44.1
be150.8.8	151—11627	9.95 −7	1.00 −6	−4.89 −7	−3.57 −7	**27.1**	41.6
be150.8.9	151—11627	9.99 −7	1.00 −6	−2.79 −7	−3.28 −7	**30.6**	44.6
be200.3.1	201—20502	9.97 −7	9.99 −7	−9.07 −7	−8.88 −7	**41.2**	43.5
be200.3.10	201—20502	9.99 −7	1.00 −6	−2.37 −7	−2.61 −7	**45.6**	54.7
be200.3.2	201—20502	9.97 −7	9.98 −7	−5.55 −7	−5.00 −7	**43.1**	53.6
be200.3.3	201—20502	1.00 −6	1.00 −6	−2.95 −7	−3.61 −7	**65.8**	100.0
be200.3.4	201—20502	9.98 −7	9.99 −7	−6.64 −7	−2.99 −7	**43.7**	52.8
be200.3.5	201—20502	9.99 −7	9.99 −7	−5.50 −7	−2.38 −7	**60.5**	78.9
be200.3.6	201—20502	1.00 −6	9.99 −7	−4.07 −7	−8.00 −8	**33.4**	44.2
be200.3.7	201—20502	9.99 −7	9.97 −7	−3.89 −7	−2.44 −7	**53.3**	58.4
be200.3.8	201—20502	9.97 −7	9.99 −7	−7.34 −7	+5.36 −8	**41.8**	56.6
be200.3.9	201—20502	9.99 −7	9.99 −7	−9.61 −7	−8.34 −7	**74.1**	93.4
be200.8.1	201—20502	9.98 −7	1.00 −6	−6.96 −7	−3.49 −7	**58.5**	67.2
be200.8.10	201—20502	9.99 −7	9.99 −7	−5.35 −7	−3.54 −7	**43.0**	61.5
be200.8.2	201—20502	9.97 −7	9.98 −7	−5.82 −7	−4.65 −7	**35.4**	41.7
be200.8.3	201—20502	9.99 −7	1.00 −6	−5.89 −7	−3.40 −7	**48.9**	64.8
be200.8.4	201—20502	9.99 −7	9.89 −7	−6.81 −7	−6.44 −7	**41.0**	51.3
be200.8.5	201—20502	9.99 −7	9.99 −7	−1.45 −7	−1.92 −7	**41.6**	54.7
be200.8.6	201—20502	9.98 −7	1.00 −6	−1.15 −7	−1.53 −7	**60.3**	85.5
be200.8.7	201—20502	9.99 −7	1.00 −6	−2.26 −6	−2.48 −6	**39.9**	53.7
be200.8.8	201—20502	9.99 −7	9.98 −7	−4.88 −9	−9.53 −9	**55.2**	60.2
be200.8.9	201—20502	9.96 −7	9.98 −7	−2.23 −7	−1.76 −7	**57.3**	68.8
be250.1	251—31877	9.99 −7	1.00 −6	−2.39 −7	−1.29 −7	**128.4**	171.1
be250.10	251—31877	1.00 −6	1.00 −6	−3.26 −7	−2.49 −7	**168.8**	232.0
be250.2	251—31877	9.99 −7	1.00 −6	−8.47 −7	−4.63 −7	**107.8**	129.3
be250.3	251—31877	1.00 −6	1.00 −6	−1.95 −7	−1.81 −7	**104.8**	115.2
be250.4	251—31877	9.96 −7	1.00 −6	−6.92 −7	−1.01 −6	**210.2**	239.4
be250.5	251—31877	1.00 −6	1.00 −6	−5.84 −7	−2.47 −7	**121.7**	147.8
be250.6	251—31877	9.98 −7	9.99 −7	−5.34 −7	−4.31 −7	**92.6**	113.0
be250.7	251—31877	1.00 −6	9.99 −7	−1.39 −7	−1.75 −8	**117.7**	153.0
be250.8	251—31877	9.99 −7	1.00 −6	−1.90 −7	−1.67 −7	**109.5**	129.2
be250.9	251—31877	9.90 −7	9.99 −7	−3.96 −7	−2.72 −7	**146.1**	169.4

Table 1 continued

Problem		max{ϵ_P, ϵ_D}		ϵ_G		Time	
Instance	n_s—m	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD
bqp100-1	101—5252	9.99 −7	9.96 −7	−2.66 −7	−4.23 −7	**7.9**	11.4
bqp100-10	101—5252	1.00 −6	9.92 −7	−1.95 −7	−3.28 −7	**24.6**	31.7
bqp100-2	101—5252	9.98 −7	1.00 −6	−4.71 −7	−5.04 −7	**16.5**	21.7
bqp100-3	101—5252	1.00 −6	1.00 −6	−6.10 −8	−2.17 −7	**23.1**	37.1
bqp100-4	101—5252	9.99 −7	9.99 −7	+9.60 −8	−1.44 −7	**13.6**	29.1
bqp100-5	101—5252	1.00 −6	9.99 −7	−7.86 −8	−2.19 −7	**19.1**	28.8
bqp100-6	101—5252	9.96 −7	9.99 −7	+8.45 −7	−2.50 −7	**7.2**	10.0
bqp100-7	101—5252	9.95 −7	9.95 −7	−6.41 −7	−7.52 −7	**10.1**	13.5
bqp100-8	101—5252	1.00 −6	1.00 −6	−8.85 −8	−3.52 −7	**16.9**	27.7
bqp100-9	101—5252	9.99 −7	9.99 −7	+7.87 −9	+2.75 −8	**15.9**	23.3
bqp250-1	251—31877	9.99 −7	1.00 −6	−4.68 −7	−1.85 −7	**119.2**	138.6
bqp250-10	251—31877	9.98 −7	9.99 −7	−1.09 −6	−1.24 −6	**84.0**	89.4
bqp250-2	251—31877	9.99 −7	9.99 −7	−7.66 −7	−7.04 −7	119.9	119.9
bqp250-3	251—31877	9.99 −7	9.96 −7	−1.75 −6	−4.02 −7	**116.4**	126.6
bqp250-4	251—31877	9.99 −7	9.92 −7	−8.32 −7	−1.25 −7	**79.6**	108.5
bqp250-5	251—31877	1.00 −6	9.99 −7	−8.75 −7	−4.82 −7	**166.5**	230.2
bqp250-6	251—31877	9.99 −7	9.99 −7	−5.26 −7	−6.67 −7	**128.0**	153.5
bqp250-7	251—31877	1.00 −6	9.99 −7	−8.46 −7	−6.95 −7	**107.4**	132.7
bqp250-8	251—31877	9.98 −7	9.99 −7	−4.61 −7	−6.12 −7	95.9	**85.3**
bqp250-9	251—31877	9.99 −7	9.99 −7	−4.00 −7	−2.11 −7	**118.5**	162.5
bqp500-1	501—126252	9.99 −7	9.99 −7	−1.32 −6	−3.06 −7	992.8	**755.9**
bqp500-10	501—126252	9.99 −7	9.93 −7	−1.47 −6	−1.32 −6	**1042.5**	932.9
bqp500-2	501—126252	9.98 −7	9.99 −7	−9.21 −7	−1.01 −7	**1113.0**	1140.2
bqp500-3	501—126252	1.00 −6	9.99 −7	−1.47 −6	+3.48 −7	1032.3	**925.9**
bqp500-4	501—126252	1.00 −6	9.97 −7	−1.24 −6	−3.88 −7	970.1	**926.0**
bqp500-5	501—126252	9.98 −7	1.00 −6	−7.52 −8	−2.16 −8	**1155.1**	1201.4
bqp500-6	501—126252	9.99 −7	9.99 −7	−7.37 −7	−8.01 −7	981.4	**777.0**
bqp500-7	501—126252	9.99 −7	9.99 −7	−1.07 −6	−1.56 −7	1116.2	**914.2**
bqp500-8	501—126252	9.99 −7	9.99 −7	−6.63 −7	−8.08 −7	1053.0	**780.6**
bqp500-9	501—126252	9.99 −7	9.99 −7	−1.05 −6	−1.54 −8	967.3	**890.8**
gka10b	126—8127	9.98 −7	9.97 −7	−4.19 −6	−8.90 −6	23.0	**20.7**
gka10d	101—5252	9.93 −7	9.93 −7	+6.59 −7	−7.02 −7	**8.4**	9.4
gka1d	101—5252	1.00 −6	9.99 −7	−2.33 −7	−1.60 −7	**15.5**	31.6
gka1e	201—20502	1.00 −6	1.00 −6	−2.34 −7	−3.41 −7	**63.9**	74.7
gka1f	501—126252	9.99 −7	1.00 −6	−8.67 −7	−5.93 −7	1012.0	**871.1**
gka2d	101—5252	9.98 −7	1.00 −6	−1.16 −7	−2.69 −7	**8.1**	16.3
gka2e	201—20502	9.98 −7	9.99 −7	−7.46 −7	−8.47 −7	**49.1**	57.7
gka2f	501—126252	9.99 −7	1.00 −6	−7.54 −7	−1.30 −6	1076.2	**922.3**
gka3d	101—5252	9.99 −7	1.00 −6	+1.66 −8	−4.88 −8	**15.0**	25.9
gka3e	201—20502	9.99 −7	1.00 −6	−1.55 −9	−1.40 −7	94.9	**93.0**

Table 1 continued

Problem		$\max\{\epsilon_P, \epsilon_D\}$		ϵ_G		Time	
Instance	n_s—m	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD
gka3f	501—126252	1.00 −6	1.00 −6	−9.60 −7	−6.97 −8	**983.9**	1023.6
gka4d	101—5252	9.93 −7	1.00 −6	+2.27 −7	−1.41 −7	**7.0**	17.7
gka4e	201—20502	1.00 −6	1.00 −6	−5.14 −7	−3.06 −7	**67.7**	83.7
gka4f	501—126252	1.00 −6	1.00 −6	−4.53 −7	−2.42 −7	**1093.4**	1165.5
gka5d	101—5252	9.93 −7	9.97 −7	−1.55 −7	−1.14 −7	**7.1**	10.2
gka5e	201—20502	9.99 −7	1.00 −6	−1.95 −8	−2.69 −8	**80.6**	99.1
gka5f	501—126252	1.00 −6	1.00 −6	−6.74 −7	−7.72 −7	973.9	**746.1**
gka6d	101—5252	9.97 −7	9.97 −7	−1.58 −8	−1.89 −8	**10.0**	15.5
gka7c	101—5252	9.96 −7	9.99 −7	−5.27 −7	−6.47 −7	**15.2**	44.6
gka7d	101—5252	9.94 −7	9.94 −7	−1.28 −6	−3.04 −7	**7.0**	9.2
gka8a	101—5252	9.99 −7	9.94 −7	+2.31 −7	+8.65 −7	87.0	**40.8**
gka8d	101—5252	9.97 −7	9.99 −7	−3.41 −7	−8.11 −8	**13.6**	27.9
gka9b	101—5252	9.74 −7	6.85 −7	+3.64 −7	−8.09 −6	**4.0**	7.1
gka9d	101—5252	1.00 −6	9.96 −7	+1.45 −6	−7.58 −8	**7.2**	8.2

There is more than one way of viewing (57) as a special case of the two-easy-block structure formulation (39). For our current implementation, we have used the following formulation. Let $\mathcal{X} = \mathcal{W}_1 := \mathcal{S}^{n+1}, \mathcal{W}_2 = \mathbb{R}^n \times \mathbb{R} \times \mathcal{S}^n \times \mathbb{R}^n, \mathcal{K}_1 = \mathcal{S}^{n+1}_+$ and $\mathcal{K}_2 = \mathbf{0}_n \times \mathbf{0}_1 \times \mathbb{R}^{n(n+1)/2}_+ \times \mathbb{R}^n_+$, where $\mathbf{0}_n$ denotes an n dimensional vector of all zeros. With these definitions, we can easily see that (57) can be viewed as having the two-easy-block structure (39) if (57a) is chosen as $\mathcal{M}_1$ and (57b) are chosen as $\mathcal{M}_2$. Note that, in view of the first inclusion in (45), the constraint $x \succeq 0$ is always satisfied by 2EBD-HPE, while SDPAD approaches it in the limit.

Table 1 compares the two methods on a collection of 134 SDP relaxations of BIQ problems. For the purpose of this comparison, we have run 2EBD-HPE with $\sigma = 0.99$ and the values of γ, τ and $\bar{k}$ in the dynamic scaling rule (55) set to $\gamma = 1.5$, $\tau = 0.9$ and $\bar{k} = 10$. Table 2 gives more detailed computational results obtained by 2EBD-HPE (see the second paragraph preceding Sect. 7.1 for an explanation on this table). Figure 5 plots the performance profiles of both methods on this collection of 134 SDP relaxations of BIQ problems.

Note that 2EBD-HPE solves 119 (out of a total of 134) problems faster than SDPAD. Moreover, 2EBD-HPE solves about 9 of them at least 2 times faster than SDPAD.

7.2 Frequency assignment problems

This subsection gives more details of our implementation of 2EBD-HPE for solving SDP relaxations of FAPs and summarizes its computational performance against SDPAD on a collection of 7 such instances generated using a subroutine from SDPT3 described in [20].

Table 2 2EBD-HPE results on BIQ problems

Instance	n_s—m	$\langle c, x\rangle$	$\langle b, w\rangle$	Iterations	ϵ_P	ϵ_D	Time
be100.1	101—5252	−2.002134 +4	−2.002129 +4	1511	5.14 −7	9.69 −7	8.1
be100.10	101—5252	−1.640851 +4	−1.640853 +4	1232	1.20 −7	9.99 −7	7.1
be100.2	101—5252	−1.798870 +4	−1.798870 +4	1381	1.00 −6	4.79 −7	6.9
be100.3	101—5252	−1.823105 +4	−1.823106 +4	1619	9.98 −7	8.62 −7	9.6
be100.4	101—5252	−1.984180 +4	−1.984179 +4	1927	9.98 −7	4.85 −7	10.4
be100.5	101—5252	−1.688870 +4	−1.688871 +4	1286	1.00 −6	8.34 −7	7.2
be100.6	101—5252	−1.814822 +4	−1.814820 +4	1463	9.53 −7	1.00 −6	8.1
be100.7	101—5252	−1.970085 +4	−1.970080 +4	1379	2.91 −7	9.91 −7	7.5
be100.8	101—5252	−1.994642 +4	−1.994638 +4	1360	8.60 −7	9.95 −7	7.3
be100.9	101—5252	−1.426337 +4	−1.426336 +4	1191	7.74 −7	9.95 −7	7.4
be120.3.1	121—7502	−1.380356 +4	−1.380355 +4	1775	9.99 −7	3.21 −7	12.3
be120.3.10	121—7502	−1.293086 +4	−1.293085 +4	1394	9.96 −7	5.78 −7	9.8
be120.3.2	121—7502	−1.362663 +4	−1.362661 +4	1964	9.98 −7	6.80 −7	13.6
be120.3.3	121—7502	−1.298791 +4	−1.298789 +4	1551	8.05 −7	9.96 −7	11.2
be120.3.4	121—7502	−1.451125 +4	−1.451122 +4	1694	6.80 −7	9.95 −7	12.2
be120.3.5	121—7502	−1.199191 +4	−1.199191 +4	3558	1.00 −6	5.64 −7	27.1
be120.3.6	121—7502	−1.343206 +4	−1.343205 +4	2130	9.98 −7	7.95 −7	15.4
be120.3.7	121—7502	−1.456411 +4	−1.456411 +4	3809	1.00 −6	5.67 −7	27.6
be120.3.8	121—7502	−1.530302 +4	−1.530302 +4	2708	9.50 −7	1.00 −6	20.1
be120.3.9	121—7502	−1.124132 +4	−1.124131 +4	2616	4.26 −7	1.00 −6	18.5
be120.8.1	121—7502	−2.019393 +4	−2.019396 +4	1257	8.22 −7	9.95 −7	8.9
be120.8.10	121—7502	−2.002400 +4	−2.002400 +4	1551	2.75 −7	9.97 −7	10.7
be120.8.2	121—7502	−2.007413 +4	−2.007412 +4	2803	1.00 −6	4.13 −7	19.8
be120.8.3	121—7502	−2.050590 +4	−2.050590 +4	1524	9.96 −7	8.17 −7	11.5
be120.8.4	121—7502	−2.177980 +4	−2.177979 +4	2027	9.98 −7	2.79 −7	14.2
be120.8.5	121—7502	−2.131628 +4	−2.131628 +4	1743	9.96 −7	7.35 −7	12.1
be120.8.6	121—7502	−1.967696 +4	−1.967695 +4	1632	1.00 −6	3.41 −7	11.6
be120.8.7	121—7502	−2.373238 +4	−2.373238 +4	1677	6.16 −7	9.97 −7	11.7
be120.8.8	121—7502	−2.120478 +4	−2.120476 +4	1297	5.09 −7	9.98 −7	9.4
be120.8.9	121—7502	−1.928441 +4	−1.928441 +4	1274	4.72 −7	9.99 −7	10.0
be150.3.1	151—11627	−1.984918 +4	−1.984915 +4	2116	6.95 −7	9.97 −7	22.3
be150.3.10	151—11627	−1.923092 +4	−1.923091 +4	2768	9.99 −7	9.23 −7	30.0
be150.3.2	151—11627	−1.886485 +4	−1.886483 +4	2094	9.32 −7	9.98 −7	21.4
be150.3.3	151—11627	−1.804372 +4	−1.804370 +4	1757	8.70 −7	9.95 −7	18.8
be150.3.4	151—11627	−2.065267 +4	−2.065264 +4	2027	7.36 −7	9.97 −7	22.3
be150.3.5	151—11627	−1.776865 +4	−1.776865 +4	2589	1.00 −6	3.35 −8	27.6
be150.3.6	151—11627	−1.805069 +4	−1.805068 +4	1944	6.59 −7	9.98 −7	20.3
be150.3.7	151—11627	−1.910131 +4	−1.910129 +4	1947	9.43 −7	9.98 −7	21.2
be150.3.8	151—11627	−1.969806 +4	−1.969806 +4	2510	9.99 −7	2.79 −7	26.6
be150.3.9	151—11627	−1.410337 +4	−1.410335 +4	1190	4.40 −7	9.96 −7	12.3

Table 2 continued

Instance	n_s—m	$\langle c, x\rangle$	$\langle b, w\rangle$	Iterations	ϵ_P	ϵ_D	Time
be150.8.1	151—11627	−2.914369 +4	−2.914367 +4	1573	6.68 −7	9.95 −7	15.9
be150.8.10	151—11627	−3.004798 +4	−3.004796 +4	2043	9.99 −7	4.34 −7	22.2
be150.8.2	151—11627	−2.882110 +4	−2.882105 +4	1520	6.93 −7	9.98 −7	16.5
be150.8.3	151—11627	−3.106037 +4	−3.106036 +4	1821	9.98 −7	9.36 −7	19.9
be150.8.4	151—11627	−2.872930 +4	−2.872928 +4	2035	9.98 −7	3.56 −7	20.3
be150.8.5	151—11627	−2.948207 +4	−2.948204 +4	1991	9.91 −7	9.95 −7	22.3
be150.8.6	151—11627	−3.143723 +4	−3.143722 +4	2711	9.17 −7	9.61 −7	26.8
be150.8.7	151—11627	−3.325211 +4	−3.325206 +4	2470	8.81 −7	9.98 −7	28.3
be150.8.8	151—11627	−3.159999 +4	−3.159996 +4	2553	9.95 −7	6.77 −7	27.1
be150.8.9	151—11627	−2.711073 +4	−2.711071 +4	2931	9.99 −7	4.75 −7	30.6
be200.3.1	201—20502	−2.771609 +4	−2.771604 +4	2069	6.75 −7	9.97 −7	41.2
be200.3.10	201—20502	−2.576069 +4	−2.576068 +4	2345	9.99 −7	6.19 −7	45.6
be200.3.2	201—20502	−2.676079 +4	−2.676076 +4	2178	7.71 −7	9.97 −7	43.1
be200.3.3	201—20502	−2.947864 +4	−2.947862 +4	3554	1.00 −6	5.87 −7	65.8
be200.3.4	201—20502	−2.910621 +4	−2.910617 +4	2284	9.97 −7	9.98 −7	43.7
be200.3.5	201—20502	−2.807299 +4	−2.807296 +4	3289	6.87 −7	9.99 −7	60.5
be200.3.6	201—20502	−2.792835 +4	−2.792832 +4	1843	7.63 −7	1.00 −6	33.4
be200.3.7	201—20502	−3.162050 +4	−3.162048 +4	2638	5.92 −7	9.99 −7	53.3
be200.3.8	201—20502	−2.924429 +4	−2.924425 +4	2256	9.97 −7	8.32 −7	41.8
be200.3.9	201—20502	−2.643705 +4	−2.643700 +4	3923	9.06 −7	9.99 −7	74.1
be200.8.1	201—20502	−5.086949 +4	−5.086942 +4	2913	5.63 −7	9.98 −7	58.5
be200.8.10	201—20502	−4.574306 +4	−4.574301 +4	2131	9.99 −7	7.21 −7	43.0
be200.8.2	201—20502	−4.433604 +4	−4.433599 +4	1869	5.93 −7	9.97 −7	35.4
be200.8.3	201—20502	−4.625398 +4	−4.625392 +4	2656	8.07 −7	9.99 −7	48.9
be200.8.4	201—20502	−4.662125 +4	−4.662119 +4	2196	6.94 −7	9.99 −7	41.0
be200.8.5	201—20502	−4.427124 +4	−4.427122 +4	2257	9.99 −7	4.79 −7	41.6
be200.8.6	201—20502	−5.121888 +4	−5.121887 +4	3105	9.98 −7	3.75 −7	60.3
be200.8.7	201—20502	−4.935288 +4	−4.935266 +4	2099	6.28 −7	9.99 −7	39.9
be200.8.8	201—20502	−4.768917 +4	−4.768917 +4	2836	9.99 −7	2.61 −7	55.2
be200.8.9	201—20502	−4.549560 +4	−4.549558 +4	2820	9.96 −7	4.44 −7	57.3
be250.1	251—31877	−2.511946 +4	−2.511945 +4	4221	9.99 −7	4.95 −7	128.4
be250.10	251—31877	−2.435502 +4	−2.435501 +4	5469	1.00 −6	4.79 −7	168.8
be250.2	251—31877	−2.368149 +4	−2.368145 +4	3459	9.95 −7	9.99 −7	107.8
be250.3	251—31877	−2.400000 +4	−2.399999 +4	3443	1.00 −6	2.37 −7	104.8
be250.4	251—31877	−2.572032 +4	−2.572028 +4	6762	8.56 −7	9.96 −7	210.2
be250.5	251—31877	−2.237471 +4	−2.237468 +4	3996	1.00 −6	7.86 −7	121.7
be250.6	251—31877	−2.401885 +4	−2.401882 +4	3301	8.25 −7	9.98 −7	92.6
be250.7	251—31877	−2.511896 +4	−2.511895 +4	3844	1.00 −6	6.02 −7	117.7
be250.8	251—31877	−2.502040 +4	−2.502039 +4	3616	9.99 −7	3.32 −7	109.5
be250.9	251—31877	−2.139706 +4	−2.139704 +4	4891	9.90 −7	9.34 −7	146.1
bqp100-1	101—5252	−8.380388 +3	−8.380384 +3	1287	7.54 −7	9.99 −7	7.9

Table 2 continued

Instance	n_s—m	$\langle c, x\rangle$	$\langle b, w\rangle$	Iterations	ϵ_P	ϵ_D	Time
bqp100-10	101—5252	−1.298027 +4	−1.298027 +4	4546	1.00 −6	3.63 −7	24.6
bqp100-2	101—5252	−1.148926 +4	−1.148925 +4	3023	9.06 −7	9.98 −7	16.5
bqp100-3	101—5252	−1.315318 +4	−1.315318 +4	4229	1.00 −6	9.45 −7	23.1
bqp100-4	101—5252	−1.073189 +4	−1.073189 +4	2573	9.99 −7	9.13 −7	13.6
bqp100-5	101—5252	−9.487027 +3	−9.487026 +3	3577	1.00 −6	4.01 −7	19.1
bqp100-6	101—5252	−1.082474 +4	−1.082476 +4	1275	9.62 −7	9.96 −7	7.2
bqp100-7	101—5252	−1.068915 +4	−1.068913 +4	1819	7.07 −7	9.95 −7	10.1
bqp100-8	101—5252	−1.176999 +4	−1.176999 +4	3043	1.00 −6	6.38 −7	16.9
bqp100-9	101—5252	−1.173325 +4	−1.173325 +4	2747	9.99 −7	2.69 −7	15.9
bqp250-1	251—31877	−4.766311 +4	−4.766306 +4	3850	9.99 −7	4.12 −7	119.2
bqp250-10	251—31877	−4.301452 +4	−4.301442 +4	2741	6.28 −7	9.98 −7	84.0
bqp250-2	251—31877	−4.722238 +4	−4.722231 +4	3693	9.99 −7	9.96 −7	119.9
bqp250-3	251—31877	−5.107673 +4	−5.107655 +4	3636	4.83 −7	9.99 −7	116.4
bqp250-4	251—31877	−4.331256 +4	−4.331249 +4	2802	9.00 −7	9.99 −7	79.6
bqp250-5	251—31877	−5.000433 +4	−5.000424 +4	5559	6.93 −7	1.00 −6	166.5
bqp250-6	251—31877	−4.366886 +4	−4.366881 +4	4037	9.99 −7	6.28 −7	128.0
bqp250-7	251—31877	−4.892173 +4	−4.892164 +4	3543	1.00 −6	6.27 −7	107.4
bqp250-8	251—31877	−3.877955 +4	−3.877951 +4	2860	4.58 −7	9.98 −7	95.9
bqp250-9	251—31877	−5.149755 +4	−5.149751 +4	3988	9.99 −7	8.96 −7	118.5
bqp500-1	501—126252	−1.259642 +5	−1.259639 +5	6205	5.13 −7	9.99 −7	992.8
bqp500-10	501—126252	−1.385344 +5	−1.385340 +5	6299	5.69 −7	9.99 −7	1042.5
bqp500-2	501—126252	−1.360111 +5	−1.360108 +5	6821	5.55 −7	9.98 −7	1113.0
bqp500-3	501—126252	−1.384534 +5	−1.384530 +5	6359	4.56 −7	1.00 −6	1032.3
bqp500-4	501—126252	−1.393284 +5	−1.393280 +5	6435	4.21 −7	1.00 −6	970.1
bqp500-5	501—126252	−1.340921 +5	−1.340921 +5	7302	9.98 −7	1.22 −7	1155.1
bqp500-6	501—126252	−1.307644 +5	−1.307642 +5	6066	5.31 −7	9.99 −7	981.4
bqp500-7	501—126252	−1.314915 +5	−1.314912 +5	6503	5.11 −7	9.99 −7	1116.2
bqp500-8	501—126252	−1.334898 +5	−1.334897 +5	6567	5.18 −7	9.99 −7	1053.0
bqp500-9	501—126252	−1.302883 +5	−1.302880 +5	5911	7.50 −7	9.99 −7	967.3
gka10b	126—8127	−1.555721 +2	−1.555708 +2	3275	9.98 −7	1.53 −8	23.0
gka10d	101—5252	−2.010856 +4	−2.010859 +4	1423	8.89 −7	9.93 −7	8.4
gka1d	101—5252	−6.528429 +3	−6.528426 +3	2606	1.00 −6	7.59 −7	15.5
gka1e	201—20502	−1.706982 +4	−1.706981 +4	3380	1.00 −6	7.48 −7	63.9
gka1f	501—126252	−6.555908 +4	−6.555896 +4	6266	5.92 −7	9.99 −7	1012.0
gka2d	101—5252	−6.990710 +3	−6.990708 +3	1506	4.79 −7	9.98 −7	8.1
gka2e	201—20502	−2.491764 +4	−2.491760 +4	2471	8.77 −7	9.98 −7	49.1
gka2f	501—126252	−1.079318 +5	−1.079316 +5	6725	9.99 −7	6.80 −7	1076.2
gka3d	101—5252	−9.734332 +3	−9.734332 +3	2852	9.99 −7	5.44 −7	15.0
gka3e	201—20502	−2.689874 +4	−2.689874 +4	5080	9.99 −7	6.77 −7	94.9
gka3f	501—126252	−1.501510 +5	−1.501508 +5	5842	7.08 −7	1.00 −6	983.9
gka4d	101—5252	−1.127841 +4	−1.127842 +4	1341	9.48 −7	9.93 −7	7.0

Table 2 continued

Instance	n_s—m	$\langle c, x\rangle$	$\langle b, w\rangle$	Iterations	ϵ_P	ϵ_D	Time
gka4e	201—20502	−3.722515 +4	−3.722511 +4	3579	1.00 −6	8.10 −7	67.7
gka4f	501—126252	−1.870879 +5	−1.870877 +5	6421	1.00 −6	5.41 −7	1093.4
gka5d	101—5252	−1.239886 +4	−1.239886 +4	1334	9.93 −7	5.28 −7	7.1
gka5e	201—20502	−3.800231 +4	−3.800231 +4	4227	9.99 −7	1.63 −7	80.6
gka5f	501—126252	−2.069143 +5	−2.069140 +5	5573	5.08 −7	1.00 −6	973.9
gka6d	101—5252	−1.492936 +4	−1.492936 +4	1841	9.97 −7	1.77 −7	10.0
gka7c	101—5252	−7.316449 +3	−7.316441 +3	2924	9.96 −7	9.66 −7	15.2
gka7d	101—5252	−1.537582 +4	−1.537578 +4	1253	4.44 −7	9.94 −7	7.0
gka8a	101—5252	−1.119721 +4	−1.119722 +4	14987	5.59 −7	9.99 −7	87.0
gka8d	101—5252	−1.700536 +4	−1.700535 +4	2613	8.86 −7	9.97 −7	13.6
gka9b	101—5252	−1.369999 +2	−1.370000 +2	736	9.74 −7	5.34 −8	4.0
gka9d	101—5252	−1.653387 +4	−1.653391 +4	1270	8.73 −7	1.00 −6	7.2

Table 3 Comparison of the methods on FAPs

Problem		$\max\{\epsilon_P, \epsilon_D\}$		ϵ_G		Time	
Instance	n_s—m	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD
fap08	120—7260	9.30 −7	9.99 −7	−3.23 −6	−1.93 −6	**5.7**	6.4
fap09	174—15225	9.94 −7	9.98 −7	−3.07 −6	+5.10 −8	7.4	**6.8**
fap10	183—14479	1.96 −7	8.28 −7	−9.73 −6	−9.83 −6	**76.2**	174.9
fap11	252—24292	1.36 −7	1.12 −7	−9.99 −6	−1.00 −5	**170.6**	424.6
fap12	369—26462	1.57 −7	8.36 −8	−9.97 −6	−1.00 −5	**556.4**	1733.1
fap25	2118—322924	5.89 −7	1.42 −7	−9.68 −6	−1.00 −5	**89519.7**	258593.0
fap36	4110—1154467	6.62 −7	4.19 −7	−9.91 −6	−1.00 −5	**293007.5**	720433.8

Table 4 2EBD-HPE results on FAPs

Instance	n_s—m	$\langle c, x\rangle$	$\langle b, w\rangle$	Iterations	ϵ_P	ϵ_D	Time
fap08	120—7260	+2.436266 +0	+2.436285 +0	956	9.30 −7	7.29 −7	5.7
fap09	174—15225	+1.079777 +1	+1.079784 +1	666	9.59 −7	9.94 −7	7.4
fap10	183—14479	+9.668432 −3	+9.678346 −3	4610	1.96 −7	1.63 −7	76.2
fap11	252—24292	+2.976395 −2	+2.977454 −2	5350	1.36 −7	9.90 −8	170.6
fap12	369—26462	+2.732333 −1	+2.732487 −1	8072	1.57 −7	7.39 −8	556.4
fap25	2118—322924	+1.287731 +1	+1.287757 +1	10270	5.89 −7	1.30 −7	89519.7
fap36	4110—1154467	+6.985624 +1	+6.985764 +1	5440	6.62 −7	4.10 −7	293007.5

The SDP relaxation of the FAP can be described as follows (see for example Subsection 2.4 in [3]). Given a network represented by a graph G with n nodes and an edge-weight matrix W, the frequency assignment problem on G can be formulated as a κ-cut problem

Table 5 Comparison of the methods on $\theta(G)$

Problem		$\max\{\epsilon_P, \epsilon_D\}$		ϵ_G		Time	
Instance	n_s—m	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD
1dc.1024	1024—24064	1.00 −6	1.00 −6	−9.34 −6	−6.57 −6	**7437.6**	10208.3
1dc.128	128—1472	1.00 −6	*1.82 −6**	−6.21 −6	+6.13 −6	**86.9**	*464.9**
1dc.2048	2048—58368	7.45 −7	7.63 −7	−1.00 −5	−1.00 −5	**107634.6**	134043.4
1dc.256	256—3840	9.53 −7	9.98 −7	−3.14 −6	+5.16 −6	**60.2**	170.3
1dc.512	512—9728	1.00 −6	1.00 −6	−8.58 −6	−8.78 −6	**1192.0**	1341.1
1et.1024	1024—9601	9.41 −7	9.51 −7	−8.64 −6	−1.00 −5	**5685.5**	9460.2
1et.128	128—673	9.92 −7	8.54 −7	−3.92 −7	+1.64 −6	3.7	**3.3**
1et.2048	2048—22529	*3.62 −3**	1.00 −6	*+3.18 −2**	−9.37 −6	*159336.8**	110786.1
1et.256	256—1665	9.99 −7	9.99 −7	−2.53 −6	−1.25 −6	**60.8**	136.3
1et.512	512—4033	9.95 −7	9.61 −7	−5.04 −6	−2.80 −6	**504.2**	1254.1
1tc.1024	1024—7937	9.93 −7	1.00 −6	−6.35 −6	−9.09 −6	**14874.0**	19483.4
1tc.128	128—513	8.96 −7	9.79 −7	−4.07 −7	−7.04 −8	**2.5**	6.3
1tc.2048	2048—18945	*2.44 −3**	9.94 −7	*+2.22 −2**	−1.00 −5	*156775.0**	**156132.9**
1tc.256	256—1313	1.00 −6	1.00 −6	−1.49 −6	−2.36 −6	**171.3**	284.4
1tc.512	512—3265	9.98 −7	1.00 −6	−4.95 −6	−6.28 −6	**3233.4**	3565.0
1zc.1024	1024—16641	8.74 −7	9.17 −7	−8.10 −6	−1.58 −6	1520.1	**1194.7**
1zc.128	128—1121	7.88 −7	9.43 −7	+9.82 −6	+1.81 −6	2.0	**1.9**
1zc.256	256—2817	5.25 −7	8.23 −7	+7.53 −6	−2.35 −6	**7.3**	**10.2**
1zc.512	512—6913	7.48 −7	9.32 −7	+7.20 −6	+1.55 −6	**83.2**	120.9
2dc.1024	1024—169163	3.04 −7	*2.45 −6**	*+1.48 −5**	*+6.20 −5**	*23295.4**	*157809.8**
2dc.512	512—54896	9.94 −7	9.64 −7	−1.58 −6	+1.25 −6	**2713.1**	17410.2
G43	1000—9991	9.96 −7	9.76 −7	−1.61 −6	+1.81 −6	969.0	**894.2**
G44	1000—9991	9.93 −7	9.31 −7	−1.19 −6	+8.75 −7	1024.8	**949.8**
G45	1000—9991	9.94 −7	9.61 −7	−1.18 −6	−1.77 −6	**1044.2**	1102.5
G46	1000—9991	9.96 −7	9.88 −7	−1.12 −6	+1.55 −6	**997.4**	1014.9
G47	1000—9991	9.90 −7	9.48 −7	−9.87 −7	−8.88 −7	1114.4	**894.1**
G51	1000—5910	1.00 −6	*1.11 −6**	−5.19 −7	+4.26 −7	**6164.0**	*20923.4**
G52	1000—5917	*2.30 −6**	*4.14 −6**	*+2.20 −5**	*+2.24 −5**	*21841.7**	*17074.8**
G53	1000—5915	*2.40 −6**	*4.08 −6**	*+1.79 −5**	*+3.24 −5**	*21511.2**	*19416.6**
G54	1000—5917	9.99 −7	1.00 −6	−4.81 −7	−1.46 −6	**4554.3**	9943.4
brock200-1	200—5067	9.69 −7	9.28 −7	−7.52 −7	+2.19 −7	5.7	**5.4**
brock200-4	200—6812	9.84 −7	9.86 −7	−8.09 −7	−3.25 −8	5.2	**4.3**
brock400-1	400—20078	9.56 −7	9.40 −7	−8.14 −7	−1.06 −6	**27.7**	32.3
c-fat200-1	200—18367	9.74 −7	9.89 −7	−2.20 −6	+3.47 −6	**3.4**	28.5
hamming-10-2	1024—23041	9.76 −7	9.63 −7	−8.99 −6	−2.18 −6	1189.1	**1123.1**
hamming-7-5-6	128—1793	9.85 −7	8.77 −7	+5.53 −6	+1.43 −6	**1.3**	4.0
hamming-8-3-4	256—16129	4.95 −7	5.27 −7	−8.92 −6	−9.57 −7	**3.5**	13.7
hamming-8-4	256—11777	8.46 −7	7.71 −7	+9.97 −6	−2.22 −6	**6.4**	7.6
hamming-9-5-6	512—53761	9.37 −7	9.76 −7	−9.11 −6	+1.81 −6	**25.7**	914.7

Table 5 continued

Problem		$\max\{\epsilon_P, \epsilon_D\}$		ϵ_G		Time	
Instance	n_s—m	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD
hamming-9-8	512—2305	7.41 −7	9.70 −7	+9.97 −6	−5.63 −7	**160.4**	477.8
keller4	171—5101	9.82 −7	9.88 −7	−8.69 −7	+1.65 −6	**3.7**	5.2
p-hat300-1	300—33918	9.98 −7	9.98 −7	−4.03 −6	−2.98 −6	**34.4**	236.0
san200-0.7-1	200—5971	9.93 −7	9.92 −7	−2.52 −6	−2.37 −6	**3.0**	135.9
sanr200-0.7	200—6033	9.72 −7	9.33 −7	−6.04 −7	+2.00 −7	5.4	**4.9**
theta10	500—12470	9.86 −7	9.87 −7	−8.00 −7	+1.36 −6	**64.6**	113.2
theta102	500—37467	9.69 −7	9.10 −7	−1.04 −6	−1.70 −6	**55.5**	61.3
theta103	500—62516	9.75 −7	9.88 −7	−1.35 −6	−2.01 −7	**50.4**	102.9
theta104	500—87245	9.85 −7	9.54 −7	−3.61 −6	−8.34 −7	**49.4**	239.1
theta12	600—17979	9.79 −7	8.52 −7	−7.26 −7	−1.01 −6	114.4	**99.3**
theta123	600—90020	9.86 −7	9.19 −7	−2.53 −6	−2.23 −7	**98.6**	201.4
theta32	150—2286	9.89 −7	9.94 −7	−4.64 −7	−5.04 −7	3.9	**3.7**
theta4	200—1949	9.84 −7	9.58 −7	−6.42 −7	+5.93 −7	8.6	**6.2**
theta42	200—5986	9.70 −7	9.93 −7	−6.14 −7	−2.70 −7	6.0	**4.8**
theta6	300—4375	9.79 −7	9.49 −7	−5.25 −7	+1.12 −6	18.6	**15.3**
theta62	300—13390	9.83 −7	9.65 −7	−1.31 −6	−1.09 −6	13.9	**12.2**
theta8	400—7905	9.70 −7	8.25 −7	−6.52 −7	−9.81 −7	36.4	**31.5**
theta82	400—23872	9.85 −7	9.56 −7	−9.01 −7	−1.96 −6	**28.7**	30.1
theta83	400—39862	9.67 −7	9.18 −7	−1.21 −6	−2.65 −7	**29.7**	51.7

$$
\begin{aligned}
\max_{X\in\mathcal{S}^n}\quad & \left[\left(\frac{\kappa-1}{2\kappa}\right)L(G,W)-\frac{1}{2}\mathrm{Diag}(We)\right]\bullet X\\
\text{s.t.}\quad & -E^{ij}\bullet X\le 2/(\kappa-1),\quad \forall\,(i,j),\\
& -E^{ij}\bullet X = 2/(\kappa-1),\quad \forall\,(i,j)\in U\subseteq E,\\
& \mathrm{diag}(X)=e,\ X\succeq 0,\ \mathrm{rank}(X)=\kappa,
\end{aligned}
$$

where $\kappa > 1$ is an integer, $L(G, W) := \mathrm{Diag}(We) - W$ is the Laplacian matrix, $E^{ij} = e_i e_j^T + e_j e_i^T$ with $e_i \in \mathbb{R}^n$ the vector with all zeros except in the ith position and $e \in \mathbb{R}^n$ is the vector with all ones. An SDP relaxation of the problem above is obtained by dropping the rank restriction and the inequality constraint for the non-edges to obtain the following formulation

$$
\begin{aligned}
\max_{X\in\mathcal{S}^n}\quad & \left[\left(\frac{\kappa-1}{2\kappa}\right)L(G,W)-\frac{1}{2}\mathrm{Diag}(We)\right]\bullet X\\
\text{s.t.}\quad & X\succeq 0, && \text{(58a)}\\
& -E^{ij}\bullet X\le 2/(\kappa-1)\quad \forall\,(i,j)\in E\backslash U, && \text{(58b)}\\
& -E^{ij}\bullet X = 2/(\kappa-1)\quad \forall\,(i,j)\in U\subseteq E,\ \mathrm{diag}(X)=e. && \text{(58c)}
\end{aligned}
$$

Table 6 2EBD-HPE results on $\theta(G)$

Instance	n_s—m	$\langle c, x\rangle$	$\langle b, w\rangle$	Iterations	ϵ_P	ϵ_D	Time
1dc.1024	1024—24064	−9.598719 +1	−9.598538 +1	6657	1.00 −6	6.18 −7	7437.6
1dc.128	128—1472	−1.684216 +1	−1.684194 +1	10375	9.79 −7	1.00 −6	86.9
1dc.2048	2048—58368	−1.747333 +2	−1.747298 +2	14660	7.45 −7	5.55 −7	107634.6
1dc.256	256—3840	−3.000019 +1	−3.000000 +1	2001	9.53 −7	9.01 −7	60.2
1dc.512	512—9728	−5.303177 +1	−5.303085 +1	8332	1.00 −6	8.53 −7	1192.0
1et.1024	1024—9601	−1.842298 +2	−1.842266 +2	6477	8.48 −7	9.41 −7	5685.5
1et.128	128—673	−2.923093 +1	−2.923091 +1	544	9.73 −7	9.92 −7	3.7
1et.2048	2048—22529	−3.233531 +2	−3.445938 +2	20000	6.45 −4	3.62 −3	159336.8
1et.256	256—1665	−5.511451 +1	−5.511422 +1	2274	9.28 −7	9.99 −7	60.8
1et.512	512—4033	−1.044250 +2	−1.044240 +2	3303	9.95 −7	8.55 −7	504.2
1tc.1024	1024—7937	−2.063072 +2	−2.063046 +2	13504	7.76 −7	9.93 −7	14874.0
1tc.128	128—513	−3.800005 +1	−3.800002 +1	414	8.96 −7	7.48 −7	2.5
1tc.2048	2048—18945	−3.598373 +2	−3.762285 +2	20000	5.82 −4	2.44 −3	156775.0
1tc.256	256—1313	−6.340007 +1	−6.339988 +1	5784	7.68 −7	1.00 −6	171.3
1tc.512	512—3265	−1.134015 +2	−1.134003 +2	19047	8.16 −7	9.98 −7	3233.4
1zc.1024	1024—16641	−1.286689 +2	−1.286668 +2	1891	5.61 −7	8.74 −7	1520.1
1zc.128	128—1121	−2.066625 +1	−2.066666 +1	344	7.88 −7	6.18 −7	2.0
1zc.256	256—2817	−3.799942 +1	−3.800000 +1	335	5.25 −7	4.10 −7	7.3
1zc.512	512—6913	−6.874906 +1	−6.875005 +1	592	6.39 −7	7.48 −7	83.2
2dc.1024	1024—169163	−1.863852 +1	−1.863795 +1	20000	3.04 −7	2.51 −7	23295.4
2dc.512	512—54896	−1.176785 +1	−1.176781 +1	15790	1.83 −7	9.94 −7	2713.1
G43	1000—9991	−2.806255 +2	−2.806246 +2	877	3.98 −7	9.96 −7	969.0
G44	1000—9991	−2.805839 +2	−2.805832 +2	930	4.01 −7	9.93 −7	1024.8
G45	1000—9991	−2.801858 +2	−2.801852 +2	922	3.98 −7	9.94 −7	1044.2
G46	1000—9991	−2.798376 +2	−2.798370 +2	893	4.05 −7	9.96 −7	997.4
G47	1000—9991	−2.818945 +2	−2.818940 +2	933	4.22 −7	9.90 −7	1114.4
G51	1000—5910	−3.490004 +2	−3.490001 +2	6110	1.00 −6	8.58 −7	6164.0
G52	1000—5917	−3.484100 +2	−3.483946 +2	20000	2.30 −6	1.87 −6	21841.7
G53	1000—5915	−3.483644 +2	−3.483519 +2	20000	1.86 −6	2.40 −6	21511.2
G54	1000—5917	−3.410003 +2	−3.410000 +2	4092	6.51 −7	9.99 −7	4554.3
brock200-1	200—5067	−2.745668 +1	−2.745664 +1	288	3.94 −7	9.69 −7	5.7
brock200-4	200—6812	−2.129351 +1	−2.129348 +1	264	3.86 −7	9.84 −7	5.2
brock400-1	400—20078	−3.970197 +1	−3.970190 +1	300	3.20 −7	9.56 −7	27.7
c-fat200-1	200—18367	−1.200003 +1	−1.199998 +1	286	4.73 −7	9.74 −7	3.4
hamming-10-2	1024—23041	−1.024019 +2	−1.024001 +2	1197	1.57 −7	9.76 −7	1189.1
hamming-7-5-6	128—1793	−4.266616 +1	−4.266664 +1	262	9.85 −7	7.85 −7	1.3
hamming-8-3-4	256—16129	−2.560046 +1	−2.559999 +1	173	4.95 −7	2.20 −7	3.5
hamming-8-4	256—11777	−1.599969 +1	−1.600002 +1	284	3.28 −7	8.46 −7	6.4
hamming-9-5-6	512—53761	−8.533495 +1	−8.533339 +1	203	8.59 −7	9.37 −7	25.7
hamming-9-8	512—2305	−2.239954 +2	−2.239999 +2	1280	7.41 −7	3.72 −7	160.4
keller4	171—5101	−1.401226 +1	−1.401223 +1	330	4.35 −7	9.82 −7	3.7

Table 6 continued

Instance	n_s—m	$\langle c, x\rangle$	$\langle b, w\rangle$	Iterations	ϵ_P	ϵ_D	Time
p-hat300-1	300—33918	−1.006806 +1	−1.006797 +1	700	9.98 −7	6.63 −7	34.4
san200-0.7-1	200—5971	−3.000000 +1	−2.999985 +1	169	4.93 −8	9.93 −7	3.0
sanr200-0.7	200—6033	−2.383619 +1	−2.383616 +1	269	3.85 −7	9.72 −7	5.4
theta10	500—12470	−8.380611 +1	−8.380597 +1	401	3.48 −7	9.86 −7	64.6
theta102	500—37467	−3.839063 +1	−3.839055 +1	297	3.23 −7	9.69 −7	55.5
theta103	500—62516	−2.252863 +1	−2.252857 +1	287	3.10 −7	9.75 −7	50.4
theta104	500—87245	−1.333624 +1	−1.333614 +1	302	5.41 −7	9.85 −7	49.4
theta12	600—17979	−9.280182 +1	−9.280169 +1	410	3.07 −7	9.79 −7	114.4
theta123	600—90020	−2.466878 +1	−2.466865 +1	299	5.00 −7	9.86 −7	98.6
theta32	150—2286	−2.757160 +1	−2.757157 +1	352	6.23 −7	9.89 −7	3.9
theta4	200—1949	−5.032129 +1	−5.032122 +1	469	4.71 −7	9.84 −7	8.6
theta42	200—5986	−2.393174 +1	−2.393171 +1	292	4.90 −7	9.70 −7	6.0
theta6	300—4375	−6.347716 +1	−6.347709 +1	401	3.82 −7	9.79 −7	18.6
theta62	300—13390	−2.964133 +1	−2.964125 +1	284	5.32 −7	9.83 −7	13.9
theta8	400—7905	−7.395367 +1	−7.395357 +1	389	3.42 −7	9.70 −7	36.4
theta82	400—23872	−3.436696 +1	−3.436689 +1	287	3.37 −7	9.85 −7	28.7
theta83	400—39862	−2.030194 +1	−2.030189 +1	278	3.20 −7	9.67 −7	29.7

There is more than one way of viewing (58) as a special case of formulation (39). For our current implementation, we have used the following two-easy-block structure formulation. Let $\mathcal{X} = \mathcal{W}_1 := \mathcal{S}^n$, $\mathcal{W}_2 = \mathbb{R}^{|E\setminus U|} \times \mathbb{R}^{|U|} \times \mathbb{R}^n$, $\mathcal{K}_1 = \mathcal{S}^n_+$ and $\mathcal{K}_2 = \mathbb{R}^{|E\setminus U|}_+ \times \mathbf{0}_{|U|} \times \mathbf{0}_n$, where $\mathbf{0}_n$ denotes an n dimensional vector of all zeros. With these definitions, we can easily see that (58) can be viewed as having the two-easy-block structure (39) if (58a) is chosen as $\mathcal{M}_1$, and (58b) and (58c) are chosen as $\mathcal{M}_2$. Note that, in view of the first inclusion in (45), the constraint $X \succeq 0$ is always satisfied by 2EBD-HPE, while SDPAD approaches it in the limit.

Table 3 compares the two methods on a collection of 7 SDP relaxations of FAPs. For the purpose of this comparison, we have run 2EBD-HPE with $\sigma = 0.99$ and the values of γ, τ and $\bar{k}$ in the dynamic scaling rule (55) set to $\gamma = 1.5$, $\tau = 0.75$ and $\bar{k} = 5$. Table 4 gives more detailed computational results obtained by 2EBD-HPE (see the second paragraph preceding Sect. 7.1 for an explanation on this table). Figure 5 plots the performance profiles of both methods on this collection of 7 SDP relaxations of FAPs.

Note that 2EBD-HPE solves 6 (out of a total of 7) problems faster than SDPAD. Moreover, 2EBD-HPE solves about 5 of them, including the two largest ones (i.e., `fap25` and `fap36`), at least 2 times faster than SDPAD.

7.3 SDPs arising from relaxation of maximum stable set problems

This subsection gives more details of our implementation of 2EBD-HPE for solving SDPs corresponding to θ-functions and θ_+-functions of graph stable set problems

Table 7 Comparison of the methods on $\theta_+(G)$

Problem		$\max\{\epsilon_P, \epsilon_D\}$		ϵ_G		Time	
Instance	n_s—m	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD
1dc.1024	1024—24064	1.00 −6	1.00 −6	−3.32 −6	−1.41 −7	**3819.0**	11958.7
1dc.128	128—1472	9.89 −7	9.99 −7	−2.89 −6	−2.93 −7	**8.1**	46.3
1dc.2048	2048—58368	1.00 −6	*3.35 −6**	−4.91 −6	+7.44 −7	**59560.7**	*235634.3**
1dc.256	256—3840	8.19 −7	9.94 −7	−3.31 −6	−5.14 −6	**11.2**	234.7
1dc.512	512—9728	9.99 −7	9.99 −7	−1.65 −6	−1.37 −7	**400.7**	1616.0
1et.1024	1024—9601	1.00 −6	9.99 −7	−3.74 −6	−3.81 −7	**2429.3**	7186.4
1et.128	128—673	9.87 −7	9.85 −7	−7.40 −8	+1.97 −8	**3.6**	4.7
1et.2048	2048—22529	1.00 −6	1.00 −6	−6.22 −6	−2.56 −7	**30816.3**	149449.2
1et.256	256—1665	*3.69 −6**	1.00 −6	+8.63 −7	−1.06 −7	*689.6**	**106.9**
1et.512	512—4033	9.99 −7	9.98 −7	−3.17 −6	−2.78 −7	**176.6**	593.3
1tc.1024	1024—7937	9.95 −7	*2.57 −6**	−3.59 −6	+4.25 −7	**6879.1**	*29837.5**
1tc.128	128—513	8.90 −7	9.94 −7	−3.04 −6	+3.48 −8	**1.6**	6.9
1tc.2048	2048—18945	9.97 −7	*3.57 −6**	−5.20 −6	+6.42 −7	**55981.8**	*168205.8**
1tc.256	256—1313	1.00 −6	1.00 −6	−1.02 −6	−1.36 −7	**84.7**	168.4
1tc.512	512—3265	1.00 −6	1.00 −6	−2.14 −6	−9.91 −8	**560.0**	2448.3
1zc.1024	1024—16641	9.83 −7	8.94 −7	+8.40 −6	+5.38 −7	1834.9	**1220.2**
1zc.128	128—1121	9.01 −7	9.81 −7	+7.75 −6	−1.36 −6	**1.6**	2.1
1zc.256	256—2817	8.97 −7	9.96 −7	−6.56 −6	−1.20 −6	**6.8**	10.8
1zc.512	512—6913	8.22 −7	9.89 −7	+1.14 −6	+1.11 −10	**140.7**	338.0
2dc.1024	1024—169163	9.96 −7	1.00 −6	−4.66 −6	−2.73 −7	**2394.4**	85897.6
2dc.512	512—54896	9.66 −7	1.00 −6	−9.99 −6	−2.77 −7	**326.9**	4058.2
G43	1000—9991	9.98 −7	9.37 −7	−1.71 −6	−1.67 −6	**772.3**	1002.7
G44	1000—9991	9.98 −7	9.91 −7	−1.45 −6	−1.62 −6	**804.6**	1091.7
G45	1000—9991	9.98 −7	9.10 −7	−1.40 −6	+6.19 −7	**801.9**	1025.9
G46	1000—9991	9.98 −7	9.29 −7	−1.23 −6	+5.70 −7	**948.0**	1125.1
G47	1000—9991	9.97 −7	9.76 −7	−1.11 −6	+4.96 −7	**840.7**	1039.9
G51	1000—5910	9.99 −7	*5.63 −6**	−2.91 −7	+5.19 −8	**6521.5**	*20902.5**
G52	1000—5917	9.94 −7	*4.35 −5**	−4.91 −7	+1.16 −6	**9781.8**	*22086.4**
G53	1000—5915	1.00 −6	*5.15 −5**	−3.31 −6	+8.46 −6	**21959.6**	*22189.0**
G54	1000—5917	9.92 −7	9.99 −7	−9.40 −7	−1.11 −8	**3274.6**	10437.7
brock200-1	200—5067	9.98 −7	9.83 −7	−9.48 −7	−3.33 −8	**6.1**	7.1
brock200-4	200—6812	9.98 −7	9.82 −7	−1.23 −6	−5.42 −8	5.7	5.7
brock400-1	400—20078	9.68 −7	9.97 −7	−1.18 −6	+2.37 −7	31.7	**27.8**
c-fat200-1	200—18367	8.71 −7	9.45 −7	−2.43 −6	−9.66 −7	**3.5**	21.9
hamming-10-2	1024—23041	7.82 −7	8.55 −7	+7.96 −6	−3.49 −6	1276.1	**947.9**
hamming-7-5-6	128—1793	6.20 −7	9.17 −7	+9.41 −6	−4.53 −7	**4.0**	7.3
hamming-8-3-4	256—16129	9.68 −7	9.99 −7	+9.60 −6	+1.10 −6	**3.9**	10.4
hamming-8-4	256—11777	9.31 −7	9.66 −7	−8.06 −6	−2.13 −6	**5.4**	5.9
hamming-9-5-6	512—53761	9.20 −7	9.99 −7	+8.37 −6	+1.88 −6	**74.3**	247.4

Table 7 continued

Problem		$\max\{\epsilon_P, \epsilon_D\}$		ϵ_G		Time	
Instance	n_s—m	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD	2EBD-HPE	SDPAD
hamming-9-8	512—2305	8.60 −7	9.16 −7	−7.73 −6	+5.12 −7	**115.4**	383.5
keller4	171—5101	9.86 −7	9.95 −7	−1.19 −6	+1.31 −7	**6.7**	20.9
p-hat300-1	300—33918	9.94 −7	9.99 −7	−1.03 −6	−9.93 −8	**34.0**	697.3
san200-0.7-1	200—5971	9.73 −7	9.61 −7	−1.17 −6	+4.02 −6	**2.7**	101.7
sanr200-0.7	200—6033	9.63 −7	9.99 −7	−4.25 −7	−5.65 −8	**6.1**	7.2
theta10	500—12470	9.80 −7	9.18 −7	−9.45 −7	−1.39 −6	68.6	**65.5**
theta102	500—37467	9.96 −7	9.74 −7	−1.49 −6	−2.65 −7	**52.0**	70.3
theta103	500—62516	9.84 −7	9.80 −7	−1.80 −6	−1.33 −7	**52.8**	109.9
theta104	500—87245	9.87 −7	9.97 −7	−2.18 −6	−2.15 −7	**49.3**	262.8
theta12	600—17979	9.65 −7	9.90 −7	−9.33 −7	−1.04 −6	128.3	**104.1**
theta123	600—90020	9.96 −7	9.82 −7	−1.73 −6	−1.17 −7	**93.2**	226.3
theta32	150—2286	9.93 −7	9.97 −7	−3.62 −7	−2.34 −8	**4.2**	5.9
theta4	200—1949	9.76 −7	9.91 −7	−5.28 −7	−1.28 −7	**8.7**	10.3
theta42	200—5986	9.87 −7	9.92 −7	−4.35 −7	−5.31 −8	**6.3**	8.4
theta6	300—4375	1.00 −6	9.93 −7	−3.83 −7	−5.71 −7	21.6	**21.0**
theta62	300—13390	9.96 −7	9.84 −7	−9.46 −7	−1.09 −7	**14.8**	15.9
theta8	400—7905	9.88 −7	8.97 −7	−9.37 −7	−1.67 −6	39.2	**32.9**
theta82	400—23872	9.98 −7	9.93 −7	−1.33 −6	+5.39 −8	**29.7**	31.7
theta83	400—39862	9.75 −7	9.92 −7	−1.59 −6	−1.41 −7	**27.8**	46.1

and summarizes its computational performance against SDPAD on a collection of 58 θ-function SDP instances and the corresponding collection of 58 θ_+-function SDP instances.

The SDPs for θ-functions and θ_+-functions of graph stable set problems can be described as follows. Given a graph G with n nodes and an edge set E, the SDP relaxations $\theta(G)$ and $\theta_+(G)$ of the maximum stable set problem are defined as

$$\begin{aligned}
\theta(G) := \max\ & C \bullet X & \qquad \theta_+(G) := \max\ & C \bullet X & \\
\text{s.t } & X \succeq 0, & \text{s.t } & X \succeq 0, & \quad (59a)\\
& I \bullet X = 1, & & I \bullet X = 1, & \quad (59b)\\
& X_{ij} = 0,\ (i,j) \in E, & & X_{ij} = 0,\ (i,j) \in E,\ X \geq 0, & \quad (59c)
\end{aligned}$$

where $C = ee^T$, $X \in \mathcal{S}^n$ and $e \in \mathbb{R}^n$ is the vector with all ones.

There is more than one way of viewing the $\theta(G)$ and $\theta_+(G)$ problems as special cases of formulation (39). For our current implementation, we have used the following two-easy-block structure formulations. For the case of the $\theta(G)$ (resp. $\theta_+(G)$) problem, we let $\mathcal{X} = \mathcal{S}^n$, $\mathcal{W}_1 := \mathcal{S}^n \times \mathbb{R}$, $\mathcal{W}_2 = \mathbb{R} \times \mathbb{R}^{|E|}$, $\mathcal{K}_1 = \mathcal{S}^n_+ \times \mathbf{0}_1$ and $\mathcal{K}_2 = \mathbf{0}_1 \times \mathbf{0}_{|E|}$ (resp. $\mathcal{K}_2 = \mathbf{0}_1 \times \mathbf{0}_{|E|} \times \mathbb{R}^{n(n+1)/2}_+$). With these definitions, we can easily see that the $\theta(G)$ and $\theta_+(G)$ problems can be viewed as having the two-easy-block structure (39)

Table 8 2EBD-HPE results on $\theta_+(G)$

Instance	n_s—m	$\langle c, x\rangle$	$\langle b, w\rangle$	Iterations	ϵ_P	ϵ_D	Time
1dc.1024	1024—24064	−9.555185 +1	−9.555121 +1	3058	5.86 −7	1.00 −6	3819.0
1dc.128	128—1472	−1.667839 +1	−1.667829 +1	945	5.93 −7	9.89 −7	8.1
1dc.2048	2048—58368	−1.742593 +2	−1.742575 +2	6634	4.87 −7	1.00 −6	59560.7
1dc.256	256—3840	−3.000003 +1	−2.999983 +1	374	5.46 −7	8.19 −7	11.2
1dc.512	512—9728	−5.269531 +1	−5.269514 +1	2350	4.35 −7	9.99 −7	400.7
1et.1024	1024—9601	−1.820729 +2	−1.820715 +2	2194	4.07 −7	1.00 −6	2429.3
1et.128	128—673	−2.923091 +1	−2.923090 +1	517	9.87 −7	7.92 −7	3.6
1et.2048	2048—22529	−3.381694 +2	−3.381652 +2	4125	5.50 −7	1.00 −6	30816.3
1et.256	256—1665	−5.446508 +1	−5.446499 +1	20000	3.69 −6	3.22 −6	689.6
1et.512	512—4033	−1.035499 +2	−1.035492 +2	1221	4.47 −7	9.99 −7	176.6
1tc.1024	1024—7937	−2.042055 +2	−2.042041 +2	6510	6.84 −7	9.95 −7	6879.1
1tc.128	128—513	−3.800018 +1	−3.799995 +1	221	6.03 −7	8.90 −7	1.6
1tc.2048	2048—18945	−3.704924 +2	−3.704886 +2	6472	6.29 −7	9.97 −7	55981.8
1tc.256	256—1313	−6.324048 +1	−6.324035 +1	2627	3.91 −7	1.00 −6	84.7
1tc.512	512—3265	−1.125343 +2	−1.125338 +2	3093	5.11 −7	1.00 −6	560.0
1zc.1024	1024—16641	−1.279977 +2	−1.279999 +2	1852	2.58 −7	9.83 −7	1834.9
1zc.128	128—1121	−2.066632 +1	−2.066665 +1	250	8.24 −7	9.01 −7	1.6
1zc.256	256—2817	−3.733380 +1	−3.733330 +1	270	5.73 −7	8.97 −7	6.8
1zc.512	512—6913	−6.799987 +1	−6.800002 +1	938	4.02 −7	8.22 −7	140.7
2dc.1024	1024—169163	−1.771014 +1	−1.770997 +1	2348	3.01 −7	9.96 −7	2394.4
2dc.512	512—54896	−1.138372 +1	−1.138349 +1	2092	4.96 −7	9.66 −7	326.9
G43	1000—9991	−2.797370 +2	−2.797360 +2	777	5.68 −7	9.98 −7	772.3
G44	1000—9991	−2.797469 +2	−2.797461 +2	814	5.75 −7	9.98 −7	804.6
G45	1000—9991	−2.793184 +2	−2.793176 +2	819	5.55 −7	9.98 −7	801.9
G46	1000—9991	−2.790332 +2	−2.790325 +2	792	5.69 −7	9.98 −7	948.0
G47	1000—9991	−2.808923 +2	−2.808917 +2	827	5.70 −7	9.97 −7	840.7
G51	1000—5910	−3.490001 +2	−3.489999 +2	5834	5.55 −7	9.99 −7	6521.5
G52	1000—5917	−3.483865 +2	−3.483862 +2	7511	5.21 −7	9.94 −7	9781.8
G53	1000—5915	−3.482137 +2	−3.482114 +2	18374	6.22 −7	1.00 −6	21959.6
G54	1000—5917	−3.410004 +2	−3.409997 +2	3260	6.05 −7	9.92 −7	3274.6
brock200-1	200—5067	−2.719677 +1	−2.719672 +1	299	6.09 −7	9.98 −7	6.1
brock200-4	200—6812	−2.112113 +1	−2.112108 +1	271	5.96 −7	9.98 −7	5.7
brock400-1	400—20078	−3.933102 +1	−3.933092 +1	307	5.08 −7	9.68 −7	31.7
c-fat200-1	200—18367	−1.200004 +1	−1.199998 +1	263	7.46 −7	8.71 −7	3.5
hamming-10-2	1024—23041	−8.533190 +1	−8.533327 +1	1119	1.87 −7	7.82 −7	1276.1
hamming-7-5-6	128—1793	−3.599932 +1	−3.600001 +1	666	6.20 −7	2.04 −7	4.0
hamming-8-3-4	256—16129	−2.559948 +1	−2.559998 +1	180	9.68 −7	6.31 −7	3.9
hamming-8-4	256—11777	−1.600024 +1	−1.599998 +1	225	2.52 −7	9.31 −7	5.4

Table 8 continued

Instance	n_s—m	$\langle c, x\rangle$	$\langle b, w\rangle$	Iterations	ϵ_P	ϵ_D	Time
hamming-9-5-6	512—53761	−5.866561 +1	−5.866660 +1	593	6.08 −7	9.20 −7	74.3
hamming-9-8	512—2305	−2.240033 +2	−2.239998 +2	968	5.36 −7	8.60 −7	115.4
keller4	171—5101	−1.346593 +1	−1.346590 +1	519	6.15 −7	9.86 −7	6.7
p-hat300-1	300—33918	−1.002023 +1	−1.002021 +1	697	4.52 −7	9.94 −7	34.0
san200-0.7-1	200—5971	−3.000000 +1	−2.999993 +1	152	9.17 −8	9.73 −7	2.7
sanr200-0.7	200—6033	−2.363331 +1	−2.363329 +1	293	3.25 −7	9.63 −7	6.1
theta10	500—12470	−8.314916 +1	−8.314900 +1	414	5.04 −7	9.80 −7	68.6
theta102	500—37467	−3.806637 +1	−3.806625 +1	308	5.18 −7	9.96 −7	52.0
theta103	500—62516	−2.237750 +1	−2.237742 +1	288	4.84 −7	9.84 −7	52.8
theta104	500—87245	−1.328267 +1	−1.328261 +1	287	4.48 −7	9.87 −7	49.3
theta12	600—17979	−9.209105 +1	−9.209088 +1	421	4.43 −7	9.65 −7	128.3
theta123	600—90020	−2.449524 +1	−2.449515 +1	293	4.32 −7	9.96 −7	93.2
theta32	150—2286	−2.729164 +1	−2.729162 +1	346	4.23 −7	9.93 −7	4.2
theta4	200—1949	−4.986907 +1	−4.986902 +1	458	3.91 −7	9.76 −7	8.7
theta42	200—5986	−2.373823 +1	−2.373821 +1	291	3.99 −7	9.87 −7	6.3
theta6	300—4375	−6.296189 +1	−6.296185 +1	431	3.31 −7	1.00 −6	21.6
theta62	300—13390	−2.937800 +1	−2.937794 +1	286	4.48 −7	9.96 −7	14.8
theta8	400—7905	−7.340799 +1	−7.340785 +1	400	5.49 −7	9.88 −7	39.2
theta82	400—23872	−3.406444 +1	−3.406435 +1	300	5.32 −7	9.98 −7	29.7
theta83	400—39862	−2.016717 +1	−2.016711 +1	279	4.85 −7	9.75 −7	27.8

if $\mathcal{M}_1$ (resp. $\mathcal{M}_2$) is chosen to be the set of $X \in \mathcal{S}^n$ satisfying (59a) and (59b) (resp. (59b) and (59c)). Note that (59b) is used to define both $\mathcal{M}_1$ and $\mathcal{M}_2$. Note that, in view of the first inclusion in (45), the constraints $X \succeq 0$ and $I \bullet X = 1$ are always satisfied by 2EBD-HPE, while SDPAD approaches them in the limit.

Tables 5 and 7 compare the two methods on a collection of 58 $\theta(G)$ instances and the corresponding collection of 58 $\theta_+(G)$ instances, respectively. For the purpose of this comparison, we have run 2EBD-HPE with $\sigma = 0.9$ and the values of γ, τ and $\bar{k}$ in the dynamic scaling rule (55) set to $\gamma = 1.5$, $\tau = 0.75$ and $\bar{k} = 5$. For the $\theta(G)$ problems, we have used the safeguard that the dynamic scaling scheme is not performed at those iterations k for which the first inequality in (52) is satisfied with $\bar{\epsilon} = 10^{-5}$. Tables 6 and 8 give more detailed computational results obtained by 2EBD-HPE (see the second paragraph preceding Sect. 7.1 for an explanation on this table). Figures 7 and 8 plot the performance profiles of both methods for solving $\theta(G)$ and $\theta_+(G)$, respectively, on this collection of 58 $\theta(G)$ instances and the corresponding collection of 58 $\theta_+(G)$ instances.

Note that 2EBD-HPE solves 36 (out of a total of 58) $\theta(G)$ and 49 (out of a total of 58) $\theta_+(G)$ problems faster than SDPAD. Moreover, 2EBD-HPE solves about 7 $\theta(G)$ and 12 $\theta_+(G)$ problems at least 4 times faster than SDPAD. Note also that 2EBD-HPE fails to solve 5 $\theta(G)$ and 1 $\theta_+(G)$ instances while SDPAD fails to solve 5 $\theta(G)$ and 6 $\theta_+(G)$ instances.

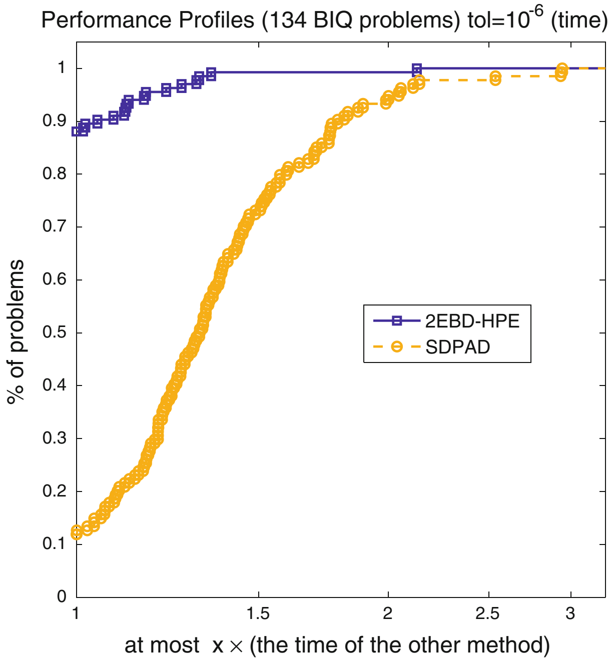


Fig. 5 Performance profiles of 2EBD-HPE and SDPAD for solving 134 SDP relaxations of BIQ problems with accuracy $\bar{\epsilon} = 10^{-6}$

8 Numerical results: part II

In this section, we briefly compare 2EBD-HPE with the SDPNAL method presented in [22] and a BD method presented in [12], namely DSA-BD. We use for this comparison the same four problem classes described in Sect. 7.

In contrast to 2EBD-HPE, the methods DSA-BD and SDPNAL always require as input a conic optimization problem given in standard form, i.e., as in (1). Hence, it is necessary for the latter two codes it is necessary (except for the θ-function SDP problems) to add additional variables and/or constraints to the original conic optimization problem (39) in order to obtain a standard form formulation. Thus, the number of variables and/or constraints handled by the latter two codes are usually larger than the number of variables and/or constraints handled by 2EBD-HPE. As the computational results of this section show, this has a negative effect on the performance of DSA-BD and SDPNAL compared to 2EBD-HPE. In fact, the main goal of the benchmark presented in this section is to show that taking advantage of any special structure of the original conic SDP formulation of the problem results in much more efficient codes both in terms of computation time and RAM.

We have used the MATLAB implementation of SDPNAL[2] version 0.1. For the 2EBD-HPE, DSA-BD and SDPNAL methods, the computational results for the SDP

[2] Downloaded in 2010 at http://www.math.nus.edu.sg/~mattohkc/SDPNAL.html.

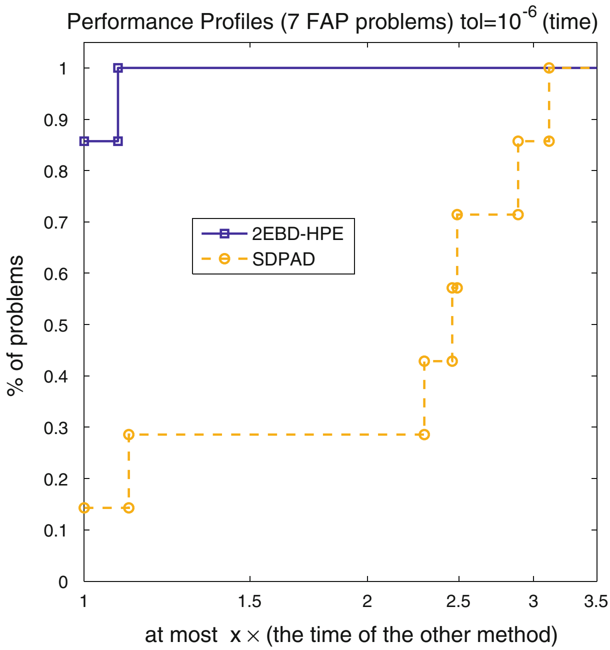


Fig. 6 Performance profiles of 2EBD-HPE and SDPAD for solving 7 SDP relaxations of FAPs with accuracy $\bar{\epsilon} = 10^{-6}$

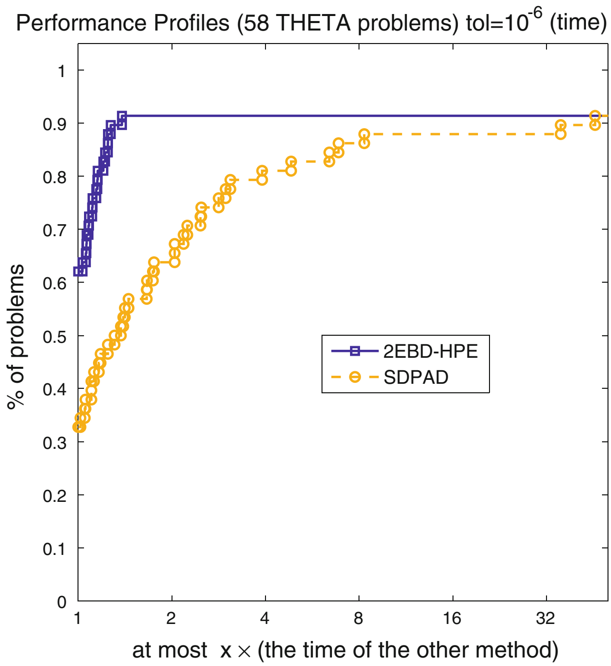


Fig. 7 Performance profiles of 2EBD-HPE and SDPAD for solving 58 $\theta(G)$ problems with accuracy $\bar{\epsilon} = 10^{-6}$

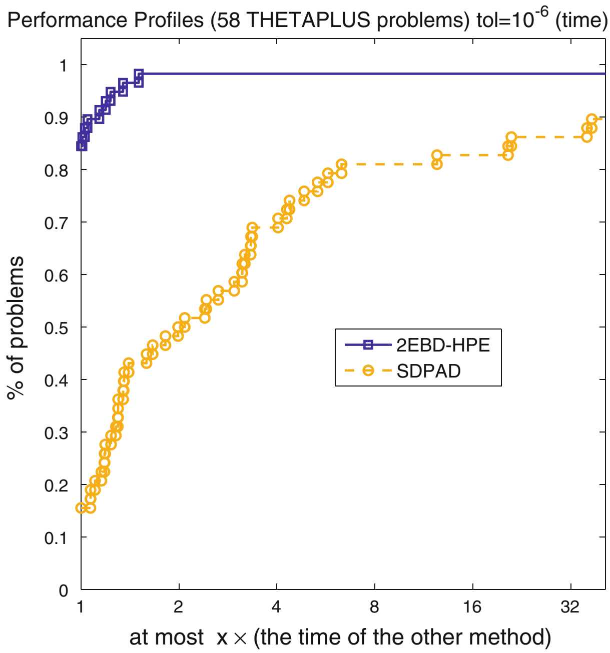


Fig. 8 Performance profiles of 2EBD-HPE and SDPAD for solving 58 $\theta_+(G)$ problems with accuracy $\bar{\epsilon} = 10^{-6}$

relaxations of BIQs and FAPs were obtained on a server with 2 Xeon X5460 processors at 3.16 GHz and 32 GB RAM, and the ones corresponding to the SDPs for θ-functions and θ_+-functions of graph stable set problems were obtained on a single core of a server with 2 Xeon X5520 processors at 2.27 GHz and 48 GB RAM.

For this benchmark, we have adopted the same stopping criterion as the one used in [10,12,15] and [22] to compare the three methods. More specifically, all methods are stopped whenever

$$\max\{\epsilon_{P,k}, \epsilon_{D,k}\} \leq \bar{\epsilon},$$

with $\bar{\epsilon} = 10^{-6}$. Even though we could have incorporated $\epsilon_{G,k}$ in the termination criterion for this benchmark, we decided to leave it out as has been done in the benchmarks of [10,12,15] and [22].

For the sake of shortness, we only report the performance profiles and exclude the detailed tables as the ones reported in Sect. 7. Figures 9, 10 and 11 plot the performance profiles of 2EBD-HPE, DSA-BD and SDPNAL for the SDP relaxations of BIQ problems, the SDP relaxations of FAPs, and the SDPs for θ-functions and θ_+-functions of graph stable set problems, respectively. Note that based on these performance profiles, 2EBD-HPE outperforms DSA-BD and SDPNAL in every problem class.

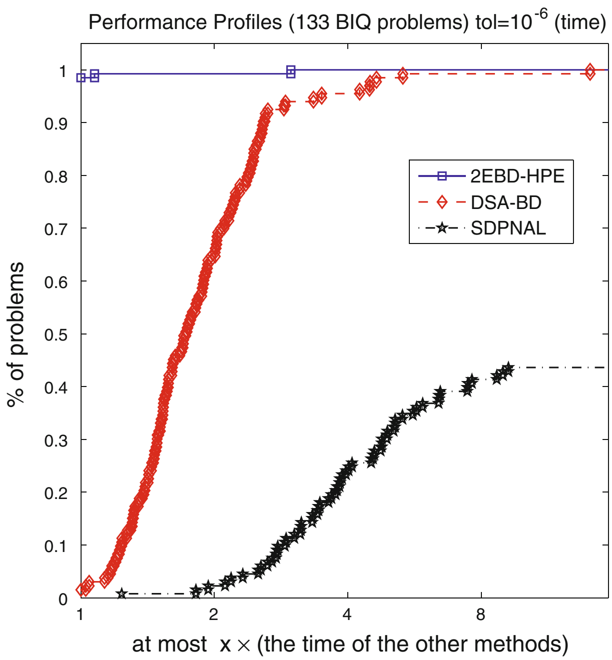


Fig. 9 Performance profiles of 2EBD-HPE, the BD method in [12] and SDPNAL for solving 133 SDP relaxations of BIQ problems with accuracy $\bar{\epsilon} = 10^{-6}$

9 Concluding remarks

Note that when applying the A-BD-HPE framework to (18), it is necessary to first specify the first and second blocks, namely $0 \in F_1(x, y) + C(x)$ and $0 \in F_2(x, y) + D(x)$, respectively. We have seen that Algorithm 1 corresponds to applying the A-BD-HPE framework to (18) by choosing the first and second blocks to be the first and second inclusions in (18), respectively. Clearly, a variant of Algorithm 1 can be obtained by changing the choice of the first and second blocks to be the second and first inclusions in (18), respectively. The resulting method can be easily shown to possess similar convergence properties as those of Algorithm 1. We observe that $\tilde{\lambda}$ for this variant should be chosen as

$$\tilde{\lambda} := \min\left\{ \frac{\sigma_1^2}{\theta L_f}, \frac{(\sigma^2 - \sigma_1^2)^{1/2}}{\sqrt{\theta}} \right\}.$$

The approach in Sect. 4 can be easily extended to the convex problem

$$\begin{aligned} \min \quad & f(x) + \sum_{i=0}^{m} h_i(x) \\ \text{s.t.} \quad & x \in \mathcal{X}, \end{aligned} \tag{60}$$

which is equivalent to solving the inclusion problem

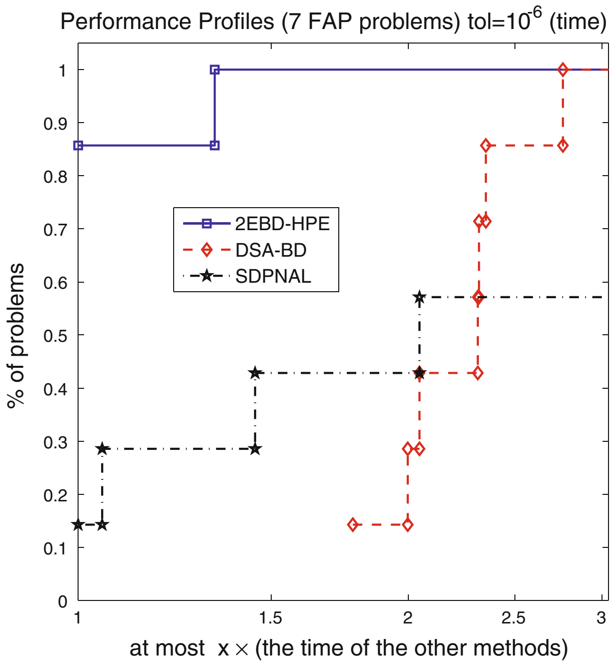
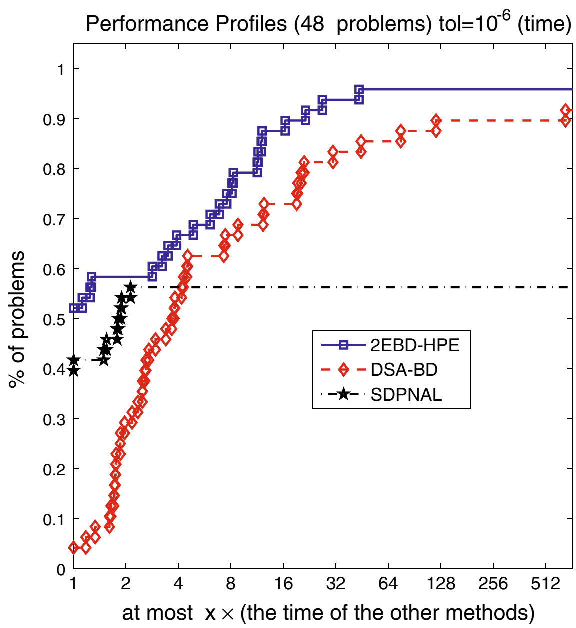


Fig. 10 Performance profiles of 2EBD-HPE, the BD method in [12] and SDPNAL for solving 7 SDP relaxations of FAPs with accuracy $\bar{\epsilon} = 10^{-6}$

Fig. 11 Performance profiles of 2EBD-HPE, the BD method in [12] and SDPNAL for solving 48 $\theta(G)$ and $\theta_+(G)$ problems with accuracy $\bar{\epsilon} = 10^{-6}$



$$0 \in \nabla f(x) + \partial h_0(x) + \sum_{i=1}^{m} y_i,$$
$$0 \in \theta_i[-x + \partial h_i^*(y_i)], \quad i = 1, \ldots, m,$$

where $\theta_i > 0, i = 1, \ldots, m$, are scaling factors. Even though, this inclusion system has $m + 1$ blocks of inclusions, it can be viewed as having two blocks for the purpose of applying the A-BD-HPE framework to it. Indeed, the first block would be the first inclusion and the second block would consist of the other m inclusions. Note that once $\tilde{x}^k$ is obtained from the proximal equation associated with the first block, it can be updated in the proximal equations corresponding to the other inclusions, and the $\tilde{y}_{i,k}$ can all be computed simultaneously. Convergence results similar to the ones obtained in Sect. 4 can be derived for (60) using the general convergence theory for BD type methods developed in [11].

Finally, our implementation of 2EBD-HPE can be found at http://www.isye.gatech.edu/~cod3/COrtiz/software/. Although in this work we have only reported computational results for problems of the form (39), it should be mentioned that the current version of 2EBD-HPE is capable of solving problems of the general form (10).

Appendix: Ergodic convergence results

This appendix derives an ergodic iteration-complexity bound for Algorithm 1.

We start by stating the weak transportation formula for the ε-subdifferential.

Proposition 10.1 (Proposition 1.2.10 in [8]) *Suppose that $f : \mathcal{Z} \rightrightarrows [-\infty, \infty]$ is a closed proper convex function. Let $z^i, v^i \in \mathcal{Z}$ and $\varepsilon_i, \alpha_i \in \mathbb{R}_+$, for $i = 1, \ldots, k$, be such that*

$$v^i \in \partial_{\varepsilon_i} f(z^i), \quad i = 1, \ldots, k, \qquad \sum_{i=1}^{k} \alpha_i = 1,$$

and define

$$z_a := \sum_{i=1}^{k} \alpha_i z^i, \quad v_a := \sum_{i=1}^{k} \alpha_i v^i,$$
$$\varepsilon_a := \sum_{i=1}^{k} \alpha_i [\varepsilon_i + \langle z^i - z_a, v^i - v_a \rangle_{\mathcal{Z}}] = \sum_{i=1}^{k} \alpha_i [\varepsilon_i + \langle z^i - z_a, v^i \rangle_{\mathcal{Z}}].$$

Then, $\varepsilon_a \geq 0$ and $v_a \in \partial_{\varepsilon_a} f(z_a)$.

Theorem 10.2 *Consider the sequences $\{(x^k, y^k)\}, \{(\tilde{x}^k, \tilde{y}^k)\}, \{(v_1^k, v_2^k)\}$ and $\{\varepsilon_k\}$ generated by Algorithm 1, and the sequences $\{c^k\}$ and $\{d^k\}$ defined in (26). For every $k \in \mathbb{N}$, define*

$$\Lambda_k := \sum_{i=1}^{k} \lambda_i, \quad (\tilde{x}_a^k, \tilde{y}_a^k) := \Lambda_k^{-1} \sum_{i=1}^{k} \lambda_i (\tilde{x}^i, \tilde{y}^i),$$

$$(v_{1,a}^k, v_{2,a}^k) := \Lambda_k^{-1} \sum_{i=1}^{k} \lambda_i (v_1^k, v_2^k), \quad (c_a^k, d_a^k) := \Lambda_k^{-1} \sum_{i=1}^{k} \lambda_i (c^k, d^k)$$

and

$$\varepsilon_k^{1,a} := \Lambda_k^{-1} \sum_{i=1}^{k} \lambda_i [\varepsilon_k + \langle \theta^{-1} c^i, \tilde{x}^i - \tilde{x}_a^k \rangle], \quad \varepsilon_k^{2,a} := \Lambda_k^{-1} \sum_{i=1}^{k} \lambda_i \langle d^i, \tilde{y}^i - \tilde{y}_a^k \rangle,$$
$$\varepsilon_k^a := \varepsilon_k^{1,a} + \varepsilon_k^{2,a}. \tag{61}$$

Then, for every $k \in \mathbb{N}$,

$$(\theta^{-1} v_{1,a}^k, v_{2,a}^k) \in \left[\partial_{\varepsilon_k^{1,a}} \left(f + h_1 + \langle \tilde{y}_a^k, \cdot \rangle\right)(\tilde{x}_a^k)\right] \times \left[\partial_{\varepsilon_k^{2,a}} \left(h_2^* - \langle \tilde{y}_a^k, \cdot \rangle\right)(\tilde{y}_a^k)\right]$$
$$\subseteq \partial_{\varepsilon_k^a} [\mathcal{L}(\cdot, \tilde{y}_a^k) - \mathcal{L}(\tilde{x}_a^k, \cdot)](\tilde{x}_a^k, \tilde{y}_a^k) \tag{62}$$

and

$$\sqrt{\theta^{-1} \|v_{1,a}^k\|^2 + \|v_{2,a}^k\|^2} \le \max\left\{\frac{1}{\sigma}, \frac{\sqrt{\theta} L}{\sigma_1^2}\right\} \left(\frac{2\sqrt{\theta}}{k}\right) \sqrt{\theta^{-1} d_{x,0}^2 + d_{y,0}^2}, \tag{63}$$

$$\varepsilon_k^a \le \max\left\{1, \frac{\sqrt{\theta} L \sigma}{\sigma_1^2}\right\} \left[\frac{8\sqrt{\theta}}{(1-\sigma_1)k}\right] \left(\theta^{-1} d_{x,0}^2 + d_{y,0}^2\right), \tag{64}$$

where $d_{x,0}$ and $d_{y,0}$ are defined in (31).

Proof Let $k \in \mathbb{N}$ be given. Note that by (35) and the definition of $\langle \cdot, \cdot \rangle_\theta$, we have

$$\varepsilon_k^{1,a} = \Lambda_k^{-1} \sum_{i=1}^{k} \lambda_i [\varepsilon_k + \langle c^i, \tilde{x}^i - \tilde{x}_a^k \rangle_\theta], \quad \varepsilon_k^{2,a} = \Lambda_k^{-1} \sum_{i=1}^{k} \lambda_i \langle d^i, \tilde{y}^i - \tilde{y}_a^k \rangle.$$

Then, in view of Lemma 4.2 and Theorem 2.4 in [12], we have

$$\|F(\tilde{x}_a^k, \tilde{y}_a^k) + (c_a^k, d_a^k)\|_{\theta,1} \le 2\frac{d_0^\theta}{\Lambda_k}, \quad \varepsilon_k^a = \varepsilon_k^{1,a} + \varepsilon_k^{2,a} \le \left(\frac{8\sigma}{1-\sigma_1}\right) \frac{(d_0^\theta)^2}{\Lambda_k}.$$

Hence, it follows from the above relations, Lemma 4.2(d) and the fact that $\lambda_k \ge \tilde{\lambda}$, that

$$\|(v_{1,a}^k, v_{2,a}^k)\|_{\theta,1} = \|F(\tilde{x}_a^k, \tilde{y}_a^k) + (c_a^k, d_a^k)\|_{\theta,1} \le 2\frac{d_0^\theta}{\Lambda_k} \le 2\frac{d_0^\theta}{k\tilde{\lambda}}, \quad \varepsilon_k^a \le \left(\frac{8\sigma}{1-\sigma_1}\right) \frac{(d_0^\theta)^2}{k\tilde{\lambda}}.$$

Using the definition of $\|(\cdot,\cdot)\|_{\theta,1}$, (30) and the definition of $\tilde{\lambda}$ in (22), we easily see that the above two inequalities imply (63) and (64). Now, (28), (29), (35), (61) and Proposition 10.1 imply that

$$\theta^{-1}v^k_{1,a} \in \partial_{\varepsilon_k^{1,a}}(f+h_1)(\tilde{x}^k_a)+\tilde{y}^k_a, \quad v^k_{2,a} \in \partial_{\varepsilon_k^{2,a}}(h_2^*)(\tilde{y}^k_a)-\tilde{x}^k_a.$$

and hence that

$$\theta^{-1}v^k_{1,a} \in (\partial_{x,\varepsilon_k^{1,a}}\mathcal{L})(\tilde{x}^k_a,\tilde{y}^k_a), \quad v^k_{2,a} \in (\partial_{y,\varepsilon_k^{2,a}}\mathcal{L})(\tilde{x}^k_a,\tilde{y}^k_a).$$

The above four inclusions are easily seen to imply (62). □

References

1. Burachik, R.S., Iusem, A.N., Svaiter, B.F.: Enlargement of monotone operators with applications to variational inequalities. Set Valued Anal. **5**, 159–180 (1997). doi:10.1023/A:1008615624787
2. Burachik, R.S., Svaiter, B.F.: Maximal monotone operators, convex functions and a special family of enlargements. Set Valued Anal. **10**, 297–316 (2002). doi:10.1023/A:1020639314056
3. Burer, S., Monteiro, R.D.C., Zhang, Y.: A computational study of a gradient-based log-barrier algorithm for a class of large-scale SDPs. Math. Program. **95**, 359–379 (2003). doi:10.1007/s10107-002-0353-7
4. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**, 120–145 (2011). doi:10.1007/s10851-010-0251-1
5. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles (2002). doi:10.1007/s101070100263
6. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Comput. Math. Appl. **2**(1), 17–40 (1976). doi:10.1016/0898-1221(76)90003-1
7. Glowinski, R., Marrocco, A.: Sur l'approximation, par éléments finis d'ordre un, et la résolution, par penalisation-dualité, d'une classe de problèmes de dirichlet non linéaires. RAIRO Anal. Numér. **2**, 41–76 (1975)
8. Lemaréchal, C.: Extensions diverses des méthodes de gradient et applications. Tech. rep., Thèse d'Etat, Université de Paris IX (1980)
9. Ma, S., Yin, W., Zhang, Y., Chakraborty, A.: An efficient algorithm for compressed mr imaging using total variation and wavelets. In: IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, pp. 1–8 (2008). doi:10.1109/CVPR.2008.4587391
10. Malick, J., Povh, J., Rendl, F., Wiegele, A.: Regularization methods for semidefinite programming. SIAM J. Optim. **20**(1), 336–356 (2009). doi:10.1137/070704575
11. Monteiro, R.C.D., Svaiter, B.F.: Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. SIAM J. Optim. **23**(1), 475–507 (2013). doi:10.1137/110849468
12. Monteiro, R.D.C., Ortiz, C., Svaiter, B.F.: Implementation of a block-decomposition algorithm for solving large-scale conic semidefinite programming problems. Comput. Optim. Appl., pp. 1–25 (2013). doi:10.1007/s10589-013-9590-3
13. Monteiro, R.D.C., Svaiter, B.F.: On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. SIAM J. Optim. **20**(6), 2755–2787 (2010). doi:10.1137/090753127
14. Monteiro, R.D.C., Svaiter, B.F.: Complexity of variants of Tseng's modified F-B splitting and Korpelevich's methods for hemivariational inequalities with applications to saddle-point and convex optimization problems. SIAM J. Optim. **21**(4), 1688–1720 (2011). doi:10.1137/100801652
15. Povh, J., Rendl, F., Wiegele, A.: A boundary point method to solve semidefinite programs. Computing **78**, 277–286 (2006). doi:10.1007/s00607-006-0182-2
16. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)
17. Rockafellar, R.T.: On the maximal monotonicity of subdifferential mappings. Pac. J. Math. **33**, 209–216 (1970)

18. Solodov, M.V., Svaiter, B.F.: A hybrid approximate extragradient-proximal point algorithm using the enlargement of a maximal monotone operator. Set Valued Anal. **7**(4), 323–345 (1999)
19. Svaiter, B.F.: A family of enlargements of maximal monotone operators. Set Valued Anal. **8**, 311–328 (2000). doi:10.1023/A:1026555124541
20. Toh, K.C., Todd, M., Tütüncü, R.H.: Sdpt3—a matlab software package for semidefinite programming. Optim. Methods Softw. **11**, 545–581 (1999)
21. Wen, Z., Goldfarb, D., Yin, W.: Alternating direction augmented lagrangian methods for semidefinite programming. Math. Program. Comput. **2**, 203–230 (2010). doi:10.1007/s12532-010-0017-1
22. Zhao, X.Y., Sun, D., Toh, K.C.: A Newton-CG augmented lagrangian method for semidefinite programming. SIAM J. Optim. **20**(4), 1737–1765 (2010). doi:10.1137/080718206