# Implementation of a block-decomposition algorithm for solving large-scale conic semidefinite programming problems

Renato D.C Monteiro* Camilo Ortiz† Benar F. Svaiter‡

May 12, 2011 (Revised: June ??, 2012)

**Abstract**

In this paper, we consider block-decomposition first-order methods for solving large-scale conic semidefinite programming problems given in standard form. Several ingredients are introduced to speed-up the method in its pure form such as: an aggressive choice of stepsize for performing the extragradient step; use of scaled inner products in the primal and dual spaces; dynamic update of the scaled inner product in the primal space for properly balancing the primal and dual relative residuals; and proper choices of the initial primal and dual iterates, as well as the initial parameter for the primal scaled inner product. Finally, we present computational results showing that our method outperforms the two most competitive codes for large-scale conic semidefinite programs, namely: the boundary point method introduced by Povh et al. and the Newton-CG augmented Lagrangian method by Zhao et al.

## 1 Introduction

Let $\mathcal{X}$ and $\mathcal{Y}$ be finite dimensional inner product spaces, with inner products and associated norms denoted by $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$, respectively. The conic programming problem is

$$\min\{\langle c,x\rangle \ : \ \mathcal{A}x=b,\ x\in K\}, \tag{1}$$ eq:problem_gene

where $\mathcal{A}:\mathcal{X}\to\mathcal{Y}$ is a linear map, $b\in\mathcal{Y}$, $c\in\mathcal{X}$ and $K\subset\mathcal{X}$ is a closed convex cone. The corresponding dual problem is

$$\max\{\langle b,y\rangle \ : \ c-\mathcal{A}^*y\in K^*\}, \tag{2}$$ eq:problem-gene

where $\mathcal{A}^*$ denotes the adjoint of $\mathcal{A}$ and $K^*$ is the dual cone of $K$ defined as

$$K^*:=\{v\in\mathcal{X}|\langle x,v\rangle\geq 0,\ \forall x\in K\}. \tag{3}$$ eq:dual-cone-1

Let $\mathbb{R}^n$ denote the $n$ dimensional Euclidean space and $\mathbb{R}^n_+$ denote the cone of nonnegative vectors in $\mathbb{R}^n$. Also, let $\mathcal{S}^n$ denote the linear space of all $n\times n$ symmetric matrices and $\mathcal{S}^n_+$ denote the cone of $n\times n$ symmetric positive semidefinite matrices. In this paper, we report our computational experience with a first-order block-decomposition (BD) method for solving large-scale conic semidefinite programming problems (1), where

$$\mathcal{X}=\mathbb{R}^{n_u+n_l}\times\mathcal{S}^{n_s},\qquad \mathcal{Y}=\mathbb{R}^m,\qquad K=\mathbb{R}^{n_u}\times\mathbb{R}^{n_l}_+\times\mathcal{S}^{n_s}_+, \tag{4}$$ eq:def-of-space

*School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205 (email: monteiro@isye.gatech.edu). The work of this author was partially supported by NSF Grants CCF-0808863 and CMMI-0900094 and ONR Grant ONR N00014-11-1-0062.

†School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205 (email: camiort@gatech.edu).

‡IMPA, Estrada Dona Castorina 110, 22460-320 Rio de Janeiro, Brazil (email:benar@impa.br). The work of this author was partially supported by CNPq grants no. 474944/2010-7, 303583/2008-8 and FAPERJ grant E-26/110.821/2008.

and the inner products in $\mathcal{X}$ and $\mathcal{Y}$ are the standard Euclidean/Frobenius inner products. Iteration complexity bounds for this method have been studied in [4, 11]. In particular, paper [10] derives the iteration-complexity of this method by using the fact that it is a special case of the Hybrid Proximal Extragradient (HPE) method introduced in [16, 17] by Solodov and Svaiter, and whose complexity is derived in [12, 11]. Moreover, as will be seen later on, the use of the HPE framework to analyze BD methods results in some crucial ideas towards improving their practical performance.

Though the BD method described in [4, 10] is simple and has nice convergence properties, its implementation in its pure form is far from being efficient. This paper introduces several ingredients to the method in its pure form to obtain a highly efficient algorithm for solving (1). The first ingredient is the use of an aggressive choice of stepsize based on a certain error criterion for performing the extragradient step. The second important idea is the implementation of the method with $\mathcal{X}$ and $\mathcal{Y}$ endowed with scaled inner products. The third idea is to allow the scaled inner product in the $\mathcal{X}$ space to dynamically change as the algorithm progresses, with the aim of properly balancing the sizes of the primal and dual relative residuals so as to make them go to zero according to the same order of magnitude. The fourth idea is proper choices of the initial primal and dual iterates, as well as the initial parameter for the primal scaled inner product.

Recently, augmented Lagrangian approaches have been proposed for the dual formulation (2) with $\mathcal{X}$, $\mathcal{Y}$ and $K$ as in (4) 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 [8, 14], a boundary point method for solving (1) is proposed which can be viewed as variants of the alternating minimization augmented Lagrangian method introduced in [6, 7] applied to (2). In [21], an inexact augmented Lagrangian method is proposed which solves a reformulation of the augmented Lagrangian subproblem involving only the $y$ variable via a semismooth Newton approach combined with the conjugate gradient method. Moreover, [8, 14] and [21] report numerical results indicating that their methods are currently the best alternatives for solving large-scale conic programming problems of the form (1) and (4). In this paper, we present computational results showing that our method outperforms the ones in [8, 14] and [21] in most large-scale instances used in our benchmark.

It should be noted that another highly efficient variant of the BP method, which performs a more aggressive Lagrange multiplier update, has been studied and implemented by Wen at el. in [20]. The resulting package, namely SDPAD, contains in fact a number of codes designed to solve different classes of graph-related SDP relaxations. Moreover, each code is written in such a way as to exploit the special structure of each SDP class, without requiring the input to be given in standard form. Since SDPAD is not a general-purpose package (in the sense that it accepts any standard form conic SDP as input) such as the ones mentioned in the previous paragraph, we have not included it in our present computational experiments. However, in a follow-up paper [9], we have compared SDPAD with a specialized version of our BD method for solving different classes of graph-related SDP relaxations, and have found that the latter one outperforms the first one in all problem classes.

This paper is organized as follows. Section 2 reviews the HPE framework and its corresponding convergence rate bounds. Section 3 presents an adaptive BD HPE framework in the context of block-structured monotone inclusion problems, similar to the one presented in [10], but with an aggressive choice of stepsize for performing the extragradient step. A specialization of the method for solving a class of convex optimization problems are also described in this section. The application to conic programming problems (1) with scaled inner products in the spaces $\mathcal{X}$ and $\mathcal{Y}$ is described in Section 4. Section 5 describes in detail all the ingredients needed to speed-up the pure form of the adaptive BD HPE method for (1), and presents numerical results demonstrating the efficiency of the resulting algorithm for solving many large instances of (1) and (4).

notation

### 1.1 Notation

The norm of a linear operator $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ is defined as

$$\|\mathcal{A}\| := \sup_{\|x\| \leq 1} \|\mathcal{A}x\|.$$

The norm of the pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is defined as

$$\|(x, y)\| = \sqrt{\|x\|^2 + \|y\|^2}. \tag{5}$$ eq:XYnorm

Let a closed convex cone $K \subset \mathcal{X}$ be given. The dual cone of $K$ is defined as

$$K^* = \{v \in \mathbb{R}^n | \langle x, v\rangle \geq 0,\ \forall x \in K\} \tag{6}$$ eq:dual-cone

and the projection operator $\Pi_K : \mathcal{X} \to K$ onto $K$ is defined by

$$\Pi_K(x) = \arg\min_{\tilde{x} \in K} \{\|x - \tilde{x}\|\}, \quad \forall x \in \mathcal{X}.$$

Finally, the indicator function $\delta_K : \mathcal{X} \to \bar{\mathbb{R}}$ of $K$ is defined as

$$\delta_K(x) = \begin{cases} 0, & x \in K, \\ \infty, & x \notin K, \end{cases}$$

and the normal cone operator $N_K : \mathcal{X} \rightrightarrows \mathcal{X}$ for $K$ is the point-to-set map given by

$$N_K(x) = \begin{cases} \emptyset, & x \notin K, \\ \{w \in \mathcal{X} : \langle \tilde{x} - x, w\rangle \leq 0, \forall \tilde{x} \in K\}, & x \in K. \end{cases}$$

Clearly, the normal cone operator $N_K$ of $K$ can be expressed in terms of its indicator function as $N_K = \partial \delta_K$ (see Subsection 2.1 for the definition of the subdifferential of a map).

adient-method

# 2 Hybrid proximal extragradient method

In this section, we review the hybrid proximal extragradient (HPE) method introduced in [16, 17]. This section contains two subsections. The first subsection reviews some basic definitions and facts about $\varepsilon$-subdifferentials of functions and $\varepsilon$-enlargements of monotone operators. The second subsection reviews the HPE framework and its corresponding convergence rate results.

differential-

## 2.1 The $\varepsilon$-subdifferential and $\varepsilon$-enlargement of monotone operators.

Let $\mathcal{Z}$ denote a finite dimensional inner product space. A point-to-set operator $T : \mathcal{Z} \rightrightarrows \mathcal{Z}$ is a relation $T \subset \mathcal{Z} \times \mathcal{Z}$ and

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

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

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

The domain of $T$ , denoted by $\mathrm{Dom}\, T$ , is defined as

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

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

$$\langle v - \tilde{v}, z - \tilde{z}\rangle \geq 0, \qquad \forall (z, v), (\tilde{z}, \tilde{v}) \in Gr(T),$$

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 $S = T$ .

In [1], Burachik, Iusem and Svaiter introduced the $\varepsilon$-enlargement of maximal monotone operators. In [11] 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 $\varepsilon$, define $T^\varepsilon : \mathcal{Z} \rightrightarrows \mathcal{Z}$ as

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

For a scalar $\varepsilon \geq 0$, the $\varepsilon$-subdifferential of a function $f : \mathcal{X} \to \bar{\mathbb{R}}$ is the operator $\partial_\varepsilon f : \mathcal{X} \rightrightarrows \mathcal{X}$ defined as

$$\partial_\varepsilon f(x) = \{w \in \mathcal{X} | f(\tilde{x}) \geq f(x) + \langle \tilde{x} - x, w \rangle - \varepsilon, \forall \tilde{x} \in \mathcal{X}\}, \quad \forall x \in \mathcal{X}.$$

When $\varepsilon = 0$, the operator $\partial_\varepsilon f$ is simply denoted by $\partial f$ and is referred to as the subdifferential of $f$. The operator $\partial f$ is trivially monotone if $f$ is proper. If $f$ is a proper lower semi-continuous convex function, then $\partial f$ is maximal monotone [15].

We now state two technical results about the $\varepsilon$-subdifferential that will be needed in our presentation.

lem:1 **Lemma 2.1.** *If $K$ is a nonempty closed cone and $K^*$ is its dual cone, then for every $x \in K$, we have*

$$-q \in (\partial \delta_K)^\varepsilon(x) \iff q \in K^*, \ \langle x, q \rangle \leq \varepsilon.$$

lem:2 **Lemma 2.2.** *Assume that $f = \langle c, \cdot \rangle$ for some $c \in \mathcal{X}$ and $h : \mathcal{X} \to \bar{\mathbb{R}}$ is a proper convex function. Then, for any $\varepsilon \geq 0$ and $x \in \mathcal{X}$, we have $\partial_\varepsilon(f + h)(x) = c + \partial_\varepsilon h(x)$ and $\partial_\varepsilon h(x) \subset (\partial h)^\varepsilon(x)$ .*

Finally, we refer the reader to [2, 18] for further discussion on the $\varepsilon$-enlargement of a maximal monotone operator.

## 2.2 The hybrid proximal extragradient method

In this section we describe the HPE method introduced in [16, 17] and review its convergence rate results established in [12, 11].

The HPE method solves the monotone inclusion problem of finding $z \in \mathcal{Z}$ such that

$$0 \in T(z), \tag{7}$$ eq:monotone-inc

where $T : \mathcal{Z} \rightrightarrows \mathcal{Z}$ is a maximal monotone operator and $\mathcal{Z}$ is a normed vector space. We will always assume that this problem has a solution, i.e., $T^{-1}(0) \neq \emptyset$.

We now describe the HPE method.

alg:HPE **Algorithm 1** : Hybrid proximal extragradient (HPE) method for solving (7)

**0)** Let $z_0 \in \mathcal{Z}$, $0 \leq \sigma \leq 1$ be given and set $k = 1$;

**1)** choose $\lambda_k > 0$, $\tilde{z}_k, \tilde{v}_k \in \mathcal{Z}$, $\sigma_k \in [0, \sigma]$ and $\varepsilon_k \geq 0$ such that

$$\tilde{v}_k \in T^{\varepsilon_k}(\tilde{z}_k), \qquad \|\lambda_k \tilde{v}_k + \tilde{z}_k - z_{k-1}\|^2 + 2\lambda_k \varepsilon_k \leq \sigma_k^2 \|\tilde{z}_k - z_{k-1}\|^2; \tag{8}$$ eq:HPE-criteric

**2)** define $z_k = z_{k-1} - \lambda_k \tilde{v}_k$, set $k \leftarrow k + 1$, and go to step 1.

We observe that the HPE method does not specify how to choose $\lambda_k$, $\tilde{z}_k$, $\tilde{v}_k$ and $\epsilon_k$ as in (7). The particular choice for these iterates will depend on the particular implementation of the method and the properties of the operator $T$. The definition of $z_k$ at step 2 is known as the *extragradient step*.

Convergence rate behavior of Framework 1 are reviewed in the next two results. The first one, referred to as the pointwise convergence result, describes the behavior of the two residual sequences $\{\tilde{v}_k\}$ and $\{\varepsilon_k\}$ corresponding to $\{\tilde{z}_k\}$.

nvergence-HPE **Theorem 2.3** (Lemma 4.3 in [11]). *Assume that $\sigma < 1$ and let $d_0$ denote the distance of the initial point $z_0$ to $T^{-1}(0)$. Then, for every $\alpha \in \mathbb{R}$ and $k \in \mathbb{N}$, there exists $i \leq k$ such that*

$$\|\tilde{v}_i\| \leq d_0 \sqrt{\frac{(1+\sigma)}{(1-\sigma)}\left(\frac{\lambda_i^{\alpha-2}}{\sum_{j=1}^k \lambda_j^\alpha}\right)}, \quad \varepsilon_i \leq \frac{d_0^2\sigma^2}{2(1-\sigma^2)}\left(\frac{\lambda_i^{\alpha-1}}{\sum_{j=1}^k \lambda_j^\alpha}\right).$$

Convergence rate behavior of the ergodic sequences $\{\tilde{z}_k^a\}$, $\{\tilde{v}_k^a\}$ and $\{\varepsilon_k^a\}$ defined as

$$\tilde{z}_k^a = \frac{1}{\Lambda_k}\sum_{i=1}^k \lambda_i \tilde{z}_i, \qquad \tilde{v}_k^a = \frac{1}{\Lambda_k}\sum_{i=1}^k \lambda_i \tilde{v}_i, \qquad \varepsilon_k^a := \frac{1}{\Lambda_k}\sum_{i=1}^k \lambda_i \left(\varepsilon_i + \langle \tilde{z}_i - \tilde{z}_k^a, \tilde{v}_i\rangle\right),$$

where $\Lambda_k = \sum_{i=1}^k \lambda_i$, are described in the following ergodic convergence result.

**Theorem 2.4** (Lemma 4.5 and Proposition 4.6 in [10]). *For every $k \in \mathbb{N}$,*

$$\tilde{v}_k^a = \frac{1}{\Lambda_k}(z_0 - z_k) \in T^{\varepsilon_k^a}(\tilde{z}_k^a), \qquad \|\tilde{v}_k^a\| \leq \frac{2d_0}{\Lambda_k},$$

*and*

$$0 \leq \varepsilon_k^a \leq \frac{1}{2\Lambda_k}\left[2\langle \tilde{z}_k^a - z_0, z_k - z_0\rangle - \|z_k - z_0\|^2\right] \leq \frac{2d_0^2}{\Lambda_k}(1+\rho_k),$$

*where $d_0$ is the distance of $z_0$ to $T^{-1}(0)$, and*

$$\rho_k := \frac{1}{d_0}\|\tilde{z}_k^a - z_k^a\| \leq \frac{1}{d_0}\max_{i=1,\ldots,k}\|\tilde{z}_i - z_i\|, \qquad \text{where } z_k^a := \frac{1}{\Lambda_k}\sum_{i=1}^k \lambda_i z_i.$$

*Moreover, if $\sigma < 1$ then*

$$\rho_k \leq \frac{\sigma\sqrt{\tau_k}}{\sqrt{(1-\sigma^2)}}, \qquad \text{where } \tau_k = \max_{i=1,\ldots,k}\frac{\lambda_i}{\Lambda_k} \leq 1.$$

sec:A-BD-HPE # 3 An adaptive block-decomposition HPE framework

In this section, we review an adaptive BD HPE framework which is a special case of the HPE framework discussed in Section 2. We also discuss a specific instance of the adaptive BD HPE framework for solving a class of convex optimization problems. This section contains two subsections. The first subsection describes the adaptive BD HPE framework in the context of a block-structured monotone inclusion problem similar to the one in Section 3 of [10], but with an adaptive (and aggressive) stepsize choice for performing the extragradient step. The second subsection describes a specific instance of the adaptive BD HPE framework of Subsection 3.1 for solving a class of convex minimization problems.

HPE-framework ## 3.1 Adaptive block-decomposition HPE framework

In this subsection, we consider the block-structured monotone inclusion problem of finding $(x, y) \in \mathcal{X}\times\mathcal{Y}$ such that

$$0 \in [F + C \otimes D](x, y), \tag{9}$$ eq:BDinclusion

where $C : \mathcal{X} \rightrightarrows \mathcal{X}$, $D : \mathcal{Y} \rightrightarrows \mathcal{Y}$ and the operator $C \otimes D : \mathcal{X} \times \mathcal{Y} \rightrightarrows \mathcal{X} \times \mathcal{Y}$ is defined as

$$(C \otimes D)(x, y) = C(x)\times D(y), \quad \forall (x, y)\in\mathcal{X}\times\mathcal{Y}.$$

We make the following assumptions regarding (9):

**A.1** $C$ and $D$ are maximal monotone operators;

**A.2** $F : \operatorname{Dom} F \subseteq \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$ is a continuous map such that $\operatorname{Dom} F \supseteq \mathcal{X} \times \operatorname{cl}(\operatorname{Dom} D)$;

**A.3** $F$ is monotone on $\operatorname{Dom} C \times \operatorname{Dom} D$;

**A.4** there exists $L_{yx} > 0$ such that

$$\|F_y(x', y) - F_y(x, y)\| \leq L_{yx}\|x' - x\|, \quad \forall y \in \operatorname{Dom} D, \forall x, x' \in \mathcal{X}.$$

Under these assumptions, it is trivial to check that the operator $C \otimes D$ is maximal monotone.

We now state a variant of the BD-HPE framework studied in [10] in which an adaptive rule for aggressively choosing the stepsize in the extragradient step is used.

alg:A-BD-HPE

**Algorithm 2** : Adaptive block-decomposition HPE (A-BD-HPE) framework for (9)

**0)** Let $x_0 \in \mathcal{X}$, $y_0 \in \mathcal{Y}$, $\sigma \in [0, 1]$, $\sigma_x, \sigma_y \in [0, 1)$ and $\tilde{\sigma}_y \in [0, \sigma_y]$ be given and set $k = 1$;

**1)** choose $\tilde{\lambda}_k > 0$ such that

$$\sigma_k := \lambda_{\max}\left(\begin{bmatrix} \sigma_y^2 & \tilde{\lambda}_k\tilde{\sigma}_y L_{yx} \\ \tilde{\lambda}_k\tilde{\sigma}_y L_{yx} & \sigma_x^2 + \tilde{\lambda}_k^2 L_{yx}^2 \end{bmatrix}\right)^{1/2} \leq \sigma; \tag{10}$$ eq:lambda-condi

**2)** compute $\tilde{y}_k, \tilde{d}_k \in \mathcal{Y}$ and $\epsilon_k^y \geq 0$ such that

$$\tilde{d}_k \in D^{\epsilon_k^y}(\tilde{y}_k), \quad \|\tilde{\lambda}_k[F_y(x_{k-1}, \tilde{y}_k) + \tilde{d}_k] + \tilde{y}_k - y_{k-1}\|^2 + 2\tilde{\lambda}_k\epsilon_k^y \leq \sigma_y^2\|\tilde{y}_k - y_{k-1}\|^2, \tag{11}$$ eq:y-prox-prob

$$\|\tilde{\lambda}_k[F_y(x_{k-1}, \tilde{y}_k) + \tilde{d}_k] + \tilde{y}_k - y_{k-1}\| \leq \tilde{\sigma}_y\|\tilde{y}_k - y_{k-1}\|; \tag{12}$$ eq:step.1b-1

compute $\tilde{x}_k, \tilde{c}_k \in \mathcal{X}$ and $\epsilon_k^x \geq 0$ such that

$$\tilde{c}_k \in C^{\epsilon_k^x}(\tilde{x}_k), \quad \|\tilde{\lambda}_k[F_x(\tilde{x}_k, \tilde{y}_k) + \tilde{a}_k] + \tilde{x}_k - x_{k-1}\|^2 + 2\tilde{\lambda}_k\epsilon_k^x \leq \sigma_x^2\|\tilde{x}_k - x_{k-1}\|^2; \tag{13}$$ eq:x-prox-prob

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

$$\|\lambda[F(\tilde{x}_k, \tilde{y}_k) + (\tilde{c}_k, \tilde{d}_k)] + (\tilde{x}_k, \tilde{y}_k) - (x_{k-1}, y_{k-1})\|^2 + 2\lambda(\epsilon_k^x + \epsilon_k^y) \leq \sigma^2\|(\tilde{x}_k, \tilde{y}_k) - (x_{k-1}, y_{k-1})\|^2; \tag{14}$$ eq:errorcriteri

**4)** set

$$(x_k, y_k) = (x_{k-1}, y_{k-1}) - \lambda_k[F(\tilde{x}_k, \tilde{y}_k) + (\tilde{c}_k, \tilde{d}_k)], \tag{15}$$ eq:extragradien

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

It should be noted that the A-BD-HPE framework is a more aggressive version of the BD-HPE framework studied in [10]. In contrast to A-BD-HPE framework, the BD-HPE framework in [10] performs the extragradient step in (15) with $\lambda_k = \tilde{\lambda}_k$. The well-definedness of $\lambda_k$ in step 3 of Framework 2 follows from the next result proved in [10].

errorcriteria

**Proposition 3.1** (Proposition 3.1 in [10]). *Consider the sequences $\{(x_k, y_k)\}, \{(\tilde{x}_k, \tilde{y}_k)\}, \{(\tilde{c}_k, \tilde{d}_k)\}$, $\{\tilde{\lambda}_k\}$ and $\{(\epsilon_k^x, \epsilon_k^y)\}$ generated by the A-BD-HPE framework. Then, for every $k \in \mathbb{N}$,*

$$F(\tilde{x}_k, \tilde{y}_k) + (\tilde{c}_k, \tilde{d}_k) \in [F + (C \otimes D)^{\epsilon_k^x + \epsilon_k^y}](\tilde{x}_k, \tilde{y}_k) \subset (F + C \otimes D)^{\epsilon_k^x + \epsilon_k^y}(\tilde{x}_k, \tilde{y}_k) \tag{16}$$ eq:BDenlargemen

*and $\lambda = \tilde{\lambda}_k$ satisfies (14). As a consequence $\lambda_k \geq \tilde{\lambda}_k$.*

In view of the fact that $\lambda_k$ satisfies the error criterion (14), Proposition 3.1, and relation (15), it follows that A-BD-HPE framework is a special case of the HPE framework applied to inclusion problem (9).

We now present two convergence results whose proofs are analogous to those of Theorems 3.2 and 3.3 of [10] with Theorem 2.3 and Proposition 3.1 of this work replacing Theorem 2.5 and Proposition 3.1 of [10], respectively.

rgence-BD-HPE **Theorem 3.2.** *Assume that $\sigma < 1$ and consider the sequences $\{(\tilde{x}_k, \tilde{y}_k)\}, \{(\tilde{c}_k, \tilde{d}_k)\}$, $\{\lambda_k\}$ and $\{(\epsilon_k^x, \epsilon_k^y)\}$ 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 (9). Then, for every $\alpha \in \mathbb{R}$ and $k \in \mathbb{N}$,*

$$(\tilde{c}_k, \tilde{d}_k) \in C^{\epsilon_k^x}(\tilde{x}_k) \times D^{\epsilon_k^y}(\tilde{y}_k),$$

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

$$\|F(\tilde{x}_i, \tilde{y}_i) + (\tilde{c}_i, \tilde{d}_i)\| \leq d_0 \sqrt{\frac{(1+\sigma)}{(1-\sigma)} \left( \frac{\lambda_i^{\alpha-2}}{\sum_{j=1}^k \lambda_j^\alpha} \right)}, \quad \epsilon_i^x + \epsilon_i^y \leq \frac{d_0^2 \sigma^2}{2(1-\sigma^2)} \left( \frac{\lambda_i^{\alpha-1}}{\sum_{j=1}^k \lambda_j^\alpha} \right). \tag{17}$$ eq:rate-BD-HPE

rgence-BD-HPE **Theorem 3.3.** *Consider the sequences $\{(\tilde{x}_k, \tilde{y}_k)\}, \{(\tilde{c}_k, \tilde{d}_k)\}$, $\{\lambda_k\}$ and $\{(\epsilon_k^x, \epsilon_k^y)\}$ generated by the A-BD-HPE framework and define for every $k \in \mathbb{N}$:*

$$(\tilde{x}_k^a, \tilde{y}_k^a) = \frac{1}{\Lambda_k} \sum_{i=1}^k \lambda_i (\tilde{x}_i, \tilde{y}_i), \quad (\tilde{c}_k^a, \tilde{d}_k^a) = \frac{1}{\Lambda_k} \sum_{i=1}^k \lambda_i (\tilde{c}_k, \tilde{d}_k), \quad \tilde{F}_k^a := \frac{1}{\Lambda_k} \sum_{i=1}^k \lambda_i F(\tilde{x}_i, \tilde{y}_i),$$

*and*

$$\begin{aligned}
\epsilon_{k,F}^a &:= \frac{1}{\Lambda_k} \sum_{i=1}^k \lambda_i \langle (\tilde{x}_i, \tilde{y}_i) - (\tilde{x}_k^a, \tilde{y}_k^a), F(\tilde{x}_i.\tilde{y}_i) \rangle \geq 0, \\
\epsilon_{k,C}^a &:= \frac{1}{\Lambda_k} \sum_{i=1}^k \lambda_i (\epsilon_i^x + \langle \tilde{x}_i - \tilde{x}_k^a, \tilde{c}_i \rangle) \geq 0, \\
\epsilon_{k,D}^a &:= \frac{1}{\Lambda_k} \sum_{i=1}^k \lambda_i (\epsilon_i^y + \langle \tilde{y}_i - \tilde{y}_k^a, \tilde{d}_i \rangle) \geq 0.
\end{aligned}$$

*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 (9) and $\sigma_{xy} = \max\{\sigma_x, \sigma_y\}$. Then, for every $k \in \mathbb{N}$,*

$$\tilde{F}_k^a \in F^{\epsilon_{k,F}^a}(\tilde{x}_k^a, \tilde{y}_k^a), \quad (\tilde{c}_k^a, \tilde{d}_k^a) \in C^{\epsilon_{k,C}^a}(\tilde{x}_k^a) \times D^{\epsilon_{k,D}^a}(\tilde{y}_k^a),$$

*and*

$$\left\| \tilde{F}_k^a + (\tilde{c}_k^a, \tilde{d}_k^a) \right\| \leq \frac{2d_0}{\Lambda_k}, \qquad \epsilon_{k,F}^a + \epsilon_{k,C}^a + \epsilon_{k,D}^a \leq \frac{2d_0^2}{\Lambda_k}(1+\eta_k),$$

*where*

$$\eta_k := \frac{2}{1-\sigma_{xy}} \left( 1 + \frac{1}{(1-\sigma_y)^2} \right)^{1/2} \sqrt{\sigma_x^2 + \sigma_y^2 + \lambda_k^2 L_{yx}^2} \leq \frac{2\sqrt{2}\sigma}{1-\sigma_{xy}} \left( 1 + \frac{1}{(1-\sigma_y)^2} \right)^{1/2}.$$

*Moreover, if $F$ is affine, then $\tilde{F}_k^a = F(\tilde{x}_k^a, \tilde{y}_k^a)$. Also, if $C$ (resp., $D$) is affine and $\epsilon_k^x = 0$ (resp., $\epsilon_k^y = 0$) for every $k \in \mathbb{N}$, then $\tilde{c}_k^a \in C(\tilde{x}_k^a)$ (resp., $\tilde{d}_k^a \in D(\tilde{y}_k^a)$).*

## 3.2 An adaptive BD method for convex optimization

-size-applied This subsection describes a specific instance of the A-BD-HPE framework of Subsection 3.1 for solving a special class of convex minimization problems and, in particular, to problem (1). The discussion of this subsection is very similar to that of Subsection 5.3 of [10] and its main purpose is to establish the necessary notation and background for the subsequent sections.

The problem of interest in this subsection is the convex optimization problem

$$\min\{f(x)+h(x) : \mathcal{A}x = b\} \tag{18}$$ eq:bdproblem

with the following assumptions:

**B.1)** $\mathcal{A}:\mathcal{X}\to\mathcal{Y}$ is a surjective linear map and $b\in\mathcal{Y}$;

**B.2)** $f,h:\mathcal{X}\to\bar{\mathbb{R}}$ are proper closed convex functions;

**B.3)** $\mathrm{dom}\,(h)\subseteq\mathrm{dom}\,(f)$ and there exists a point $\hat{x}\in\mathrm{ri}\,(\mathrm{dom}\,h)\bigcap\mathrm{ri}\,(\mathrm{dom}\ \ f)$ such that $\mathcal{A}(\hat{x})=b$;

**B.4)** the solution set of (18) is non-empty;

**B.5)** $f$ is differentiable on $\mathrm{cl}\,(\mathrm{dom}\,h)$ and $\nabla f$ is $L$-Lipschitz continuous on $\mathrm{cl}\,(\mathrm{dom}\,h)$.

We now make a few observations about (18). First, under the above assumptions, $x^*$ is an optimal solution of (18) if, and only if, it satisfies the condition

$$0\in\partial f(x)+\partial h(x)+N_{\mathcal{M}}(x), \tag{19}$$ eq:condition

where $\mathcal{M}:=\{x\in\mathcal{X}:\mathcal{A}(x)=b\}$. Second, the above assumptions also guarantee that $\partial f+\partial g+N_{\mathcal{M}}$ is maximal monotone. Third, we can see that (1) is a special case of (18) by setting

$$f(\cdot)=\langle c,\cdot\rangle,\quad h(\cdot)=\delta_K(\cdot). \tag{20}$$ eq:transformati

It is well-known that the optimality condition (19) is equivalent to find a pair $(x,y)\in\mathcal{X}\times\mathcal{Y}$ satisfying

$$\mathcal{A}x-b=0,\quad \nabla f(x)+\partial h(x)-\mathcal{A}^*y\ni 0, \tag{21}$$ eq:convexcondit

which is a special case of the block-structured monotone inclusion problem (9) with

$$F(x,y):=\begin{pmatrix}-\mathcal{A}^*y\\ \mathcal{A}x-b\end{pmatrix},\quad C(x)=\partial f(x)+\partial h(x),\quad D(y)=0. \tag{22}$$ eq:convexmap

We now state an adaptive BD method for solving (18).

alg:A-BD

**Algorithm 3** : Adaptive block-decomposition (A-BD) method for (18).

**0)** Let $x_0\in\mathcal{X}$, $y_0\in\mathcal{Y}$ and $0<\sigma\leq 1$ be given, set $\tilde{\lambda}>0$ such that $\tilde{\lambda}L+\tilde{\lambda}^2\|\mathcal{A}\|^2=\sigma^2$ and set $k=1$.

**1)** Set $\tilde{y}_k=y_{k-1}-\tilde{\lambda}(\mathcal{A}x_{k-1}-b)$ and $\tilde{x}_k=(I+\tilde{\lambda}\partial h)^{-1}[x_{k-1}-\tilde{\lambda}(\nabla f(x_{k-1})-\mathcal{A}^*\tilde{y}_k)]$;

**2)** define

$$\tilde{v}_k=\begin{pmatrix}(x_{k-1}-\tilde{x}_k)/\tilde{\lambda}\\ \mathcal{A}\tilde{x}_k-b\end{pmatrix},\qquad \epsilon_k^x=\frac{L}{2}\|\tilde{x}_k-x_{k-1}\|^2, \tag{23}$$ eq:v_k_tilde-1

choose $\lambda_k$ to be the largest $\lambda>0$ such that

$$\left\|\lambda\tilde{v}_k+\begin{pmatrix}\tilde{x}_k\\ \tilde{y}_k\end{pmatrix}-\begin{pmatrix}x_{k-1}\\ y_{k-1}\end{pmatrix}\right\|^2+2\lambda\epsilon_k^x\leq\sigma^2\left\|\begin{pmatrix}\tilde{x}_k\\ \tilde{y}_k\end{pmatrix}-\begin{pmatrix}x_{k-1}\\ y_{k-1}\end{pmatrix}\right\|^2, \tag{24}$$ eq:convex-error

**3)** set

$$\begin{pmatrix}x_k\\ y_k\end{pmatrix}=\begin{pmatrix}x_{k-1}\\ y_{k-1}\end{pmatrix}-\lambda_k\tilde{v}_k, \tag{25}$$ eq:IBD-A-extrag

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

We now make two remarks about the A-BD method. First, when $h = \delta_Q$ for some closed convex set $Q$, the resolvent $(I + \lambda\partial h)^{-1}$ that appears in step 1 of the A-BD method reduces to the projection operator $\Pi_Q$ onto $Q$, and hence it is invariant under a change of inner product obtained by multiplying the inner product in $\mathcal{X}$ by a scalar $\theta > 0$. Second, computation of $\lambda_k$ in step 2 of the A-BD method involves the solution of a second-order equation whose coefficients are relatively cheap to compute. Observe also that there is no trial-and-error involved on this stepsize selection. The use of the pre-determined value $\tilde{\lambda}$ as in step 0) allows us to determine $\tilde{x}_k$ and $\tilde{y}_k$ in step 1, which are then used to determine the adaptive value of the stepsize $\lambda_k$ in step 2. This stepsize choice still ensures that the A-BD method is special case of the A-BD-HPE framework (see Proposition 3.4 below), and hence enjoys all its convergence properties.

Recall that the norm $\|(\cdot,\cdot)\|$ in the $\mathcal{X} \times \mathcal{Y}$ space used in (24) is defined in (5). The determination of this norm depends on specific choices of inner product norms for the spaces $\mathcal{X}$ and $\mathcal{Y}$. We will see in Section 4 that this degree of freedom can be exploited to select norms for Algorithm 3 that properly balances the primal and the dual spaces with respect to each other, and as a byproduct improves the practical behavior of the method.

The following result shows that Algorithm 3 is a special case of the A-BD-HPE framework of Subsection 3.1.

pro:HPE-IBD **Proposition 3.4.** *The A-BD method (Algorithm 3) for solving (18) is a special instance of the A-BD-HPE framework for solving the inclusion problem (9) with $F$, $C$ and $D$ given by (22),*

$$\sigma_x = (\tilde{\lambda}L)^{1/2}, \qquad \sigma_y = \tilde{\sigma}_y = 0, \qquad L_{yx} = \|\mathcal{A}\|$$

*and for every $k \in \mathbb{N}$,*

$$\tilde{\lambda}_k = \tilde{\lambda}, \quad \epsilon_k^y = 0 \quad , \tag{26}$$ eq:epsilonx

$$\tilde{d}_k = 0, \quad \tilde{c}_k = \frac{1}{\tilde{\lambda}}(x_{k-1} - \tilde{x}_k) + \mathcal{A}^*\tilde{y}_k. \tag{27}$$ eq:defak

*Proof.* Consider the block-structured inclusion problem (9) with $F$, $C$ and $D$ defined as in (22). First observe that the condition on $\tilde{\lambda}$ at step 0 of Algorithm 3 is equivalent to (10) with $\sigma_x$, $\sigma_y$, $\tilde{\sigma}_y$, $\tilde{\lambda}_k$ and $L_{yx}$ as above. Moreover, it is shown in the proof of Proposition 5.5 in [10] that $\tilde{y}_k$, $\tilde{d}_k$, $\epsilon_k^y$, $\tilde{x}_k$, $\tilde{c}_k$ and $\epsilon_k^x$ as in (26), (27) and Algorithm 3 satisfy (11)-(13) of the A-BD-HPE framework when $\lambda_k = \tilde{\lambda}$ . Using (22), (23) and (27) we conclude that

$$\tilde{v}_k = F(\tilde{x}_k, \tilde{y}_k) + (\tilde{c}_k, \tilde{d}_k), \tag{28}$$ eq:v_k-equiv

and hence steps 2 and 3 of Algorithm 3 are a special case of steps 3 and 4 of Framework 2. Therefore, the conclusion of the proposition follows. □

We now specialize Theorem 3.2 and Theorem 3.3 to the context of (18) and Algorithm 3.

vergence-A-BD **Theorem 3.5.** *Consider the sequences $\{y_k\}$, $\{(\tilde{x}_k, \tilde{y}_k)\}$ generated by the A-BD method for (18), and the sequences $\{\epsilon_k^x\}$ and $\{\tilde{c}_k\}$ as in (26) and (27), respectively. Moreover, for each $k \in \mathbb{N}$, define*

$$(\tilde{x}_k^a, \tilde{y}_k^a) = \frac{1}{k}\sum_{i=1}^{k}(\tilde{x}_i, \tilde{y}_i), \quad \tilde{c}_k^a = \frac{1}{k}\sum_{i=1}^{k}\tilde{c}_i, \tag{29}$$ eq:ergodicxy

*and*

$$\epsilon_{k,C}^a := \frac{1}{k}\sum_{i=1}^{k}(\epsilon_k^x + \langle \tilde{x}_i - \tilde{x}_k^a, \tilde{c}_i\rangle) \geq 0. \tag{30}$$ eq:ergodiceps

*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 (21). Then, for every $k \in \mathbb{N}$, the following statements hold:*

*a)* $\tilde{c}_k \in (\partial_{\epsilon_k^x} f + \partial h)(\tilde{x}_k)$, *and if* $\sigma < 1$, *there exists* $i \leq k$ *such that*

$$\|(-\mathcal{A}^* \tilde{y}_i + \tilde{c}_i, b - \mathcal{A}(\tilde{x}_i))\| \leq d_0 \sqrt{\frac{(1+\sigma)}{(1-\sigma)} \left( \frac{1}{\lambda_i \sum_{j=1}^k \lambda_j} \right)} \leq \frac{d_0}{\tilde{\lambda}\sqrt{k}} \sqrt{\frac{1+\sigma}{1-\sigma}}, \tag{31}$$

eq:upper-bound-

$$\epsilon_i^x \leq \frac{\sigma^2 d_0^2}{2(1-\sigma^2) \sum_{j=1}^k \lambda_j} \leq \frac{\sigma^2 d_0^2}{2(1-\sigma^2)\tilde{\lambda} k};$$

*b) we have*

$$\tilde{c}_k^a \in \partial_{\epsilon_{k,C}^a} (f + h)(\tilde{x}_k^a),$$

*and*

$$\|(-\mathcal{A}^* \tilde{y}_k^a + \tilde{c}_k^a, b - \mathcal{A}\tilde{x}_k^a)\| \leq \frac{2d_0}{\sum_{i=1}^k \lambda_i} \leq \frac{2d_0}{\tilde{\lambda} k}, \quad \epsilon_{k,C}^a \leq \frac{2d_0^2}{\sum_{i=1}^k \lambda_i}(1+\bar{\eta}) \leq \frac{2d_0^2}{\tilde{\lambda} k}(1+\bar{\eta}), \tag{32}$$

eq:upper-bound-

*where*

$$\bar{\eta} := \frac{2\sqrt{2}\sigma}{1-\sqrt{\lambda L}} \left( 1 + \frac{1}{(1-\sqrt{\lambda L})^2} \right)^{1/2}.$$

*Proof.* The proof of this result is similar to the one in Theorem 5.6 of [10]. It follows from Proposition 3.4, Theorem 3.2 with $\alpha = 1$, and Theorem 3.3. □

For the sake of future reference, we specialize Theorem 3.5 to the special case when $f$ is linear and $h$ is the indicator function of a closed convex cone as in (20).

e-linear-A-BD **Corollary 3.6.** *Assume that* $f$ *and* $h$ *are given by* (20)*, and consider the same sequences and* $d_0$ *as in Theorem 3.5. In addition, let the sequences* $\{\tilde{z}_k\}$ *and* $\{\tilde{z}_k^a\}$ *be defined for every* $k \in \mathbb{N}$ *by*

$$\tilde{z}_k := c - \tilde{c}_k, \qquad \tilde{z}_k^a := c - \tilde{c}_k^a. \tag{33}$$

eq:def-of-z-til

*Then, for every* $k \in \mathbb{N}$, *the following statements hold:*

*a)* $\tilde{z}_k \in K^*$, $\langle \tilde{x}_k, \tilde{z}_k \rangle = 0$, *and if* $\sigma < 1$, *there exists* $i \leq k$ *such that*

$$\|(\mathcal{A}^* \tilde{y}_i + \tilde{z}_i - c, b - \mathcal{A}(\tilde{x}_i))\| \leq d_0 \sqrt{\frac{(1+\sigma)}{(1-\sigma)} \left( \frac{1}{\lambda_i \sum_{j=1}^k \lambda_j} \right)} \leq \frac{d_0 \|\mathcal{A}\|}{\sigma\sqrt{k}} \sqrt{\frac{1+\sigma}{1-\sigma}}, \tag{34}$$

eq:linear-upper

*b)* $\tilde{z}_k^a \in K^*$ , $\langle \tilde{x}_k^a, \tilde{z}_k^a \rangle \leq \epsilon_{k,C}^x$ *and*

$$\|(\mathcal{A}^* \tilde{y}_k^a + \tilde{z}_k^a - c, b - \mathcal{A}\tilde{x}_k^a)\| \leq \frac{2d_0}{\sum_{i=1}^k \lambda_i} \leq \frac{2d_0 \|\mathcal{A}\|}{\sigma k}, \tag{35}$$

eq:linear-upper

$$\epsilon_{k,C}^a \leq \frac{d_0^2}{\sum_{i=1}^k \lambda_i}(2+8\sigma) \leq \frac{d_0^2 \|\mathcal{A}\|}{\sigma k}(2+8\sigma). \tag{36}$$

eq:linear-upper

*Proof.* Let $k \in \mathbb{N}$ be given. First note that the assumption that $f$ is linear implies that $L = 0$ and hence $\tilde{\lambda} = \sigma / \|\mathcal{A}\|$ and $\epsilon_k^x = 0$, in view of (23) and step 0 of Algorithm 3. The second part of statements a) and b) of Corollary 3.6 follows from (31), (32), and (33), and the fact that $\tilde{\lambda} = \sigma / \|\mathcal{A}\|$. To show the first part of Corollary 3.6(a), note that by Theorem 3.5(a),(20) and the fact that $\epsilon_k^x = 0$, we have

$$\tilde{z}_k := c - \tilde{c}_k \in c - (\nabla f + \partial h)(\tilde{x}_k) = c - (c + \partial \delta_K(\tilde{x}_k)) = -\partial \delta_K(\tilde{x}_k),$$

and hence that $\tilde{z}_k \in K^*$ and $\langle \tilde{x}_k, \tilde{z}_k \rangle = 0$, in view of Lemma 2.1 with $q = \tilde{z}_k$, $x = \tilde{x}_k$ and $\varepsilon = 0$. The proof of the first part of Corollary 3.6(b) follows by using a similar argument together with Lemma 2.2. □

ion-of-BD-HPE

# 4 A scaled A-BD method for conic programming

In this section, we introduce a scaled version of the A-BD method for conic programming which is essentially the A-BD method applied to (1) with different choices of inner products in the $\mathcal{X}$ and $\mathcal{Y}$ space. Even though there is nothing new in this section from the theoretical point of view, the idea of choosing different inner products for $\mathcal{X}$ and $\mathcal{Y}$ plays an important role in the practical performance of the method for solving large-scale conic programming problems.

Given $\theta > 0$ and $\mathcal{U} : \mathcal{Y} \to \mathcal{Y}$ a self adjoint positive definite linear mapping, consider the inner products in the $\mathcal{X}$ and $\mathcal{Y}$ space defined as

$$\langle \cdot, \cdot \rangle_\theta := \theta^{-1} \langle \cdot, \cdot \rangle, \qquad \langle \cdot, \cdot \rangle_{\mathcal{U}} := \langle \cdot, \mathcal{U}^{-1} \cdot \rangle, \tag{37}$$

eq:innerscaling

respectively. These inner products induce norms $\|\cdot\|_\theta$ and $\|\cdot\|_{\mathcal{U}}$ in the spaces $\mathcal{X}$ and $\mathcal{Y}$, as well as an operator norm $\|\cdot\|_{[\theta,\mathcal{U}]}$ in the space of linear maps from $\mathcal{X}$ to $\mathcal{Y}$ satisfying the following identities

$$\|x\|_\theta = \theta^{-1/2}\|x\|, \qquad \|y\|_{\mathcal{U}} = \left\|\mathcal{U}^{-1/2} y\right\|, \qquad \|\mathcal{A}\|_{[\theta,\mathcal{U}]} = \theta^{1/2} \left\|\mathcal{U}^{-1/2}\mathcal{A}\right\|, \tag{38}$$

eq:operrel-1

for every $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and linear map $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$. Moreover, the adjoint of the map $\mathcal{A}$ with respect to these new inner products, denoted by $(\mathcal{A}^*)_{[\theta,\mathcal{U}]}$, satisfies

$$(\mathcal{A}^*)_{[\theta,\mathcal{U}]} = \theta \mathcal{A}^* \mathcal{U}^{-1}. \tag{39}$$

eq:operrel-2

Also, the new inner products induce an obvious inner product and a norm $\|(\cdot,\cdot)\|_{[\theta,\mathcal{U}]}$ in the Cartesian product $\mathcal{X} \times \mathcal{Y}$ defined as

$$\|(x,y)\|_{[\theta,\mathcal{U}]} = \sqrt{\|x\|_\theta^2 + \|y\|_{\mathcal{U}}^2}, \tag{40}$$

eq:new-norm

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

We will now state the scaled A-BD method and then show in Proposition (4.1) that it is the A-BD method applied to (1) when the new inner products $\langle \cdot, \cdot \rangle_\theta$ and $\langle \cdot, \cdot \rangle_{\mathcal{U}}$ are used for both $\mathcal{X}$ and $\mathcal{Y}$, respectively.

alg:SA-BD

**Algorithm 4** : Scaled adaptive block-decomposition (SA-BD) method for solving (1) .

**0)** Let $x_0 \in \mathcal{X}$, $y_0 \in \mathcal{Y}$, $0 < \sigma \leq 1$ and $\theta > 0$ be given, and set $k = 1$ and

$$\tilde{\lambda}_x = \frac{\sigma\sqrt{\theta}}{\|\mathcal{U}^{-1/2}\mathcal{A}\|}, \qquad \tilde{\lambda}_y = \frac{\sigma}{\|\mathcal{U}^{-1/2}\mathcal{A}\|\sqrt{\theta}}; \tag{41}$$

eq:lambda-x-y

**1)** compute

$$\tilde{y}_k = y_{k-1} - \tilde{\lambda}_y(\mathcal{A}x_{k-1} - b), \qquad \tilde{x}_k = \Pi_K \left[x_{k-1} - \tilde{\lambda}_x \left(c - \mathcal{A}^*\mathcal{U}^{-1}\tilde{y}_k\right)\right]; \tag{42}$$

eq:intermediate

**2)** define

$$\tilde{v}_k = \begin{pmatrix} (x_{k-1} - \tilde{x}_k)/\tilde{\lambda}_x \\ \mathcal{A}\tilde{x}_k - b \end{pmatrix}, \tag{43}$$

eq:v_k_tilde

choose $\lambda_k$ to be the largest $\lambda > 0$ such that

$$\left\|\lambda\tilde{v}_k + \begin{pmatrix} \tilde{x}_k \\ \tilde{y}_k \end{pmatrix} - \begin{pmatrix} x_{k-1} \\ y_{k-1} \end{pmatrix}\right\|_{[\theta,\mathcal{U}]} \leq \sigma \left\|\begin{pmatrix} \tilde{x}_k \\ \tilde{y}_k \end{pmatrix} - \begin{pmatrix} x_{k-1} \\ y_{k-1} \end{pmatrix}\right\|_{[\theta,\mathcal{U}]}; \tag{44}$$

eq:errorcriteri

**3)** set

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} x_{k-1} \\ y_{k-1} \end{pmatrix} - \lambda_k \tilde{v}_k, \tag{45}$$

eq:extragradien

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

It is worth noting that the coefficients of the second-order equation which determines $\lambda_k$ in step 2 of the SA-BD method are more expensive to compute than their counterpart in the A-BD method, due to the need to evaluate the norm $\|\cdot\|_{\mathcal{U}}$ and inner product $\langle\cdot,\cdot\rangle_{\mathcal{U}}$. Note however that the latter entities are relatively easy to evaluate when $\mathcal{U}$ has a diagonal representation.

pro:SABD-ABD **Proposition 4.1.** *Algorithm 4 is equivalent to Algorithm 3 when $f$ and $h$ are given by (20), $\tilde{\lambda} = \tilde{\lambda}_y$ for all $k \in \mathbb{N}$, and the inner products in $\mathcal{X}$ and $\mathcal{Y}$ are chosen as $\langle\cdot,\cdot\rangle_\theta$ and $\langle\cdot,\cdot\rangle_{\mathcal{U}}$, respectively,*

*Proof.* First observe that the gradient of the objective function $f(\cdot) = \langle c, \cdot\rangle$ of (1) at any $x \in \mathcal{X}$ with respect to the inner product $\langle\cdot,\cdot\rangle_\theta$ is $\theta c$. Letting $\tilde{\lambda} = \tilde{\lambda}_y$, and using the observation immediately following Algorithm 3, relations (38) and (39), the fact that the Lipschitz constant $L$ of the gradient of $f$ is zero, and the fact that $\tilde{\lambda}_x = \theta\tilde{\lambda}_y$ due to (41), we easily see that Algorithm 4 is equivalent to Algorithm 3 under the conditions stated in the proposition. □

We now specialize the convergence rate results in Theorem 3.5 to the context of Algorithm 4.

ergence-SA-BD **Theorem 4.2.** *Consider the sequences $\{(x_k, y_k)\}$ and $\{(\tilde{x}_k, \tilde{y}_k)\}$ generated by SA-BD method for (1), and the sequences $\{\tilde{c}_k\}$, $\{\hat{y}_k\}$ and $\{\hat{z}_k\}$ defined for every $k \in \mathbb{N}$ by*

$$\tilde{c}_k = \frac{1}{\tilde{\lambda}_y}(x_{k-1} - \tilde{x}_k) + \theta\mathcal{A}^*\mathcal{U}^{-1}\tilde{y}_k, \qquad \hat{y}_k = \mathcal{U}^{-1}\tilde{y}_k, \qquad \hat{z}_k = c - \theta^{-1}\tilde{c}_k. \tag{46}$$ eq:def-of-y-z-h

*Moreover, define the sequences $\{(\tilde{x}^a_k, \tilde{y}^a_k)\}$, $\{\tilde{c}^a_k\}$ and $\{\epsilon^a_{k,C}\}$ as in (29) and (30) with $\epsilon^x_k = 0$, and the sequences $\{\hat{y}^a_k\}$ and $\{\hat{z}^a_k\}$ defined for every $k \in \mathbb{N}$ by*

$$\hat{y}^a_k = \mathcal{U}^{-1}\tilde{y}^a_k, \qquad \hat{z}^a_k = c - \theta^{-1}\tilde{c}^a_k.$$

*Let $\mathcal{X}^*$ and $\mathcal{Y}^*$ denote the set of optimal solutions of (1) and (2), respectively, and define*

$$d_{0,x} := \min\{\|x_0 - x^*\| : x^* \in \mathcal{X}^*\}, \qquad d_{0,y} := \min\left\{\left\|\mathcal{U}^{-1}y_0 - y^*\right\| : y^* \in \mathcal{Y}^*\right\}. \tag{47}$$ eq:x-y-distance

*Then, for every $k \in \mathbb{N}$, the following statements hold:*

*a) $\hat{z}_k \in K^*$, $\langle\tilde{x}_k, \hat{z}_k\rangle = 0$, and if $\sigma < 1$, there exists $i \le k$ such that*

$$\begin{aligned}\theta\left\|\mathcal{A}^*\hat{y}_i + \hat{z}_i - c\right\|^2 + \|\mathcal{U}\|^{-1}\left\|b - \mathcal{A}\tilde{x}_i\right\|^2 &\le \left(\frac{1+\sigma}{1-\sigma}\right)\frac{\theta^{-1}d^2_{0,x} + \|\mathcal{U}\|\, d^2_{0,y}}{\lambda_i\sum_{j=1}^k \lambda_j} \\ &\le \left(\frac{1+\sigma}{1-\sigma}\right)\left\|\mathcal{U}^{-1/2}\mathcal{A}\right\|^2\frac{d^2_{0,x} + \theta\|\mathcal{U}\|\, d^2_{0,y}}{k\sigma^2};\end{aligned} \tag{48}$$ eq:SABD-upper-b

*b) $\hat{z}^a_k \in K^*$ , $\langle\tilde{x}^a_k, \hat{z}^a_k\rangle \le \epsilon^x_{k,C}$ and*

$$\theta\left\|\mathcal{A}^*\hat{y}^a_k + \hat{z}^a_k - c\right\|^2 + \|\mathcal{U}\|^{-1}\left\|b - \mathcal{A}\tilde{x}^a_k\right\|^2 \le \left\|\mathcal{U}^{-1/2}\mathcal{A}\right\|^2\frac{d^2_{0,x} + \theta\|\mathcal{U}\|\, d^2_{0,y}}{(k\sigma)^2}, \tag{49}$$ eq:SABD-upper-b

$$\epsilon^a_{k,C} \le (2 + 8\sigma)\left\|\mathcal{U}^{-1/2}\mathcal{A}\right\|\frac{\theta^{-1/2}d^2_{0,x} + \theta^{1/2}\|\mathcal{U}\|\, d^2_{0,y}}{k\sigma}. \tag{50}$$ eq:SABD-upper-b

*Proof.* Using the fact that the dual cone with respect to the inner product $\langle\cdot,\cdot\rangle_\theta$ is $K^*$, we can easily see that the dual of (1) with respect to the inner products $\langle\cdot,\cdot\rangle_\theta$ and $\langle\cdot,\cdot\rangle_{\mathcal{U}}$ is given by

$$\max\{\langle b, y\rangle_{\mathcal{U}} \ : \ \theta c - (\mathcal{A}^*)_{[\theta,\mathcal{U}]}\, y \in K^*\} = \max\{\langle b, \mathcal{U}^{-1}y\rangle \ : \ c - \mathcal{A}^*\mathcal{U}^{-1}y \in K^*\}. \tag{51}$$ eq:problem-gene

Observe that the set of dual optimal solutions of (51) is exactly $\mathcal{U}(\mathcal{Y}^*)$. Letting $\tilde{d}_0$ denote the distance from the initial point $(x_0, y_0)$ to the scaled set of primal-dual optimal solutions $\mathcal{X}^* \times \mathcal{U}(\mathcal{Y}^*)$ with respect to the scaled norm (40), it follows from Proposition 4.1 and Corollary 3.6(a) that, for every $k \in \mathbb{N}$, we have

$$\tilde{z}_k \in K^*, \qquad \langle \tilde{x}_k, \tilde{z}_k \rangle_\theta = 0, \tag{52}$$

eq:scaled inclu

and if $\sigma < 1$, there exists $i \leq k$ such that

$$\left\| (\mathcal{A}^*)_{[\theta,\mathcal{U}]} \tilde{y}_i + \theta \hat{z}_i - \theta c \right\|_\theta^2 + \|b - \mathcal{A}\tilde{x}_i\|_{\mathcal{U}}^2 \leq \frac{(\tilde{d}_0)^2}{\lambda_i \sum_{j=1}^k \lambda_j} \left( \frac{1+\sigma}{1-\sigma} \right) \leq \frac{(\tilde{d}_0)^2 \, \|\mathcal{A}\|_{[\theta,\mathcal{U}]}^2}{\sigma^2 k} \left( \frac{1+\sigma}{1-\sigma} \right), \tag{53}$$

eq:scaled-upper

where $\hat{z}_k$ is given by (46). Observe that the first part of Theorem 4.2(a) is implied by (52). We will now show that (53) implies (48). Noting (47) and letting $(x^*, y^*) \in \mathcal{X}^* \times \mathcal{Y}^*$ be such that $d_{0,x} = \|x_0 - x^*\|$ and $d_{0,y} = \left\| \mathcal{U}^{-1} y_0 - y^* \right\|$, it follows from the definition of $\tilde{d}_0$ and (38) that

$$\begin{aligned} (\tilde{d}_0)^2 &\leq \|x_0 - x^*\|_\theta^2 + \|y_0 - \mathcal{U}y^*\|_{\mathcal{U}}^2 = \theta^{-1} \|x_0 - x^*\|^2 + \left\| \mathcal{U}^{-1/2} y_0 - \mathcal{U}^{1/2} y^* \right\|^2 \\ &\leq \theta^{-1} d_{0,x}^2 + \|\mathcal{U}\| \left\| \mathcal{U}^{-1} y_0 - y^* \right\|^2 = \theta^{-1} d_{0,x}^2 + \|\mathcal{U}\| \, d_{0,y}^2. \end{aligned}$$

Also, in view of (38) and (39), and the fact that $\hat{y}_i = \mathcal{U}^{-1}\tilde{y}_i$ , we have

$$\begin{aligned} \| (\mathcal{A}^*)_{[\theta,\mathcal{U}]} \, \tilde{y}_i + \theta \hat{z}_i - \theta c \|_\theta^2 &= \theta^{-1} \left\| \theta \mathcal{A}^* \mathcal{U}^{-1} \tilde{y}_i + \theta \hat{z}_i - \theta c \right\|^2 = \theta \left\| \mathcal{A}^* \hat{y}_i + \hat{z}_i - c \right\|^2, \\ \|b - \mathcal{A}\tilde{x}_i\|_{\mathcal{U}}^2 &= \|\mathcal{U}^{-1/2}(b - \mathcal{A}\tilde{x}_i)\|^2 \geq \|\mathcal{U}\|^{-1} \|b - \mathcal{A}\tilde{x}_i\|^2. \end{aligned}$$

Then, combining the last three relations with (53) and using the last identity in (38), we obtain (48). Hence, statement a) follows.

Statement b) follows by similar arguments as those used in the proof of a). □

Observe that the convergence rate bounds of Theorem 4.2 implies the ones of Theorem 3.5 with $\mathcal{U} = \mathcal{A}\mathcal{A}^*$ and $\theta = \|\mathcal{U}\|^{-1}$. This seems to suggest that Algorithm 4 with $\mathcal{U} = \mathcal{A}\mathcal{A}^*$ might be faster than its version with $\mathcal{U} = \mathcal{I}$. As we will see in Section 5 we will use $\mathcal{U} = \mathcal{A}\mathcal{A}^*$ in our implementation of Algorithm 4.

Observe also, that the convergence rate bounds of Theorem 4.2, suggest how the scalar $\theta$ affects the magnitude of the primal and dual residuals. For the sake of concreteness, consider the bound (48). Viewing all the quantities in this formula, with the exception of $\theta$, as constants, we easily see that the primal and dual residuals are

$$\|b - \mathcal{A}\tilde{x}_i\| = \mathcal{O}\left( \max\left\{1, \theta^{1/2}\right\} \right), \qquad \|\mathcal{A}^*\hat{y}_i + \hat{z}_i - c\| = \mathcal{O}\left( \max\left\{1, \theta^{-1/2}\right\} \right),$$

respectively. Hence, as $\theta \to 0$, the dual residual can become significantly larger than the primal residual, while, as $\theta \to \infty$, the first can become significantly larger than the later one. In fact, we have observed in our computational experiments that these residuals behave exactly as described above.

mplementation

## 5 Implementation details and numerical results

In this section, we describe all the ingredients needed to speed-up the implementation of Algorithm 4, and present numerical results demonstrating the efficiency of the resulting method for solving many large instances of (1) and (4). More specifically, we describe two important ingredients, namely: i) convenient choice of initial primal and dual iterates, and initial parameter for the scaled inner product (37) on the space $\mathcal{X}$, and; ii) dynamic change of the scaled inner product in the $\mathcal{X}$ space as the algorithm progresses, with the aim of properly balancing the sizes of the primal and dual relative residuals. This section also contains four subsections reporting computational results which compare our method with the ones in [8, 14] and [21] for various types of conic semidefinite programming problems.

For every $k \in \mathbb{N}$, define the primal and dual relative residuals as

$$\epsilon_{P,k} = \frac{\|b - \mathcal{A}(\tilde{x}_k)\|}{1 + \|b\|}, \quad \epsilon_{D.k} = \frac{\|\mathcal{A}^* \hat{y}_k + \hat{z}_k - c\|}{1 + \|c\|}, \tag{54}$$ eq:errors

where $\{\tilde{x}_k\}$ is the sequence generated by Algorithm 4, and $\{\hat{y}_k\}$ and $\{\hat{z}_k\}$ are given by (46). In our implementation, we used the stopping criterion

$$\max\{\epsilon_{P,k}, \epsilon_{D,k}\} \leq \bar{\epsilon}, \tag{55}$$ eq:criteria

where $\bar{\epsilon} > 0$ is a given tolerance. We observe that the complementarity measure is $\langle \tilde{x}_k, \hat{z}_k \rangle = 0$ for every $k \in \mathbb{N}$, in view of Theorem 4.2(a). We note that the two methods we compare our code to also use the stopping criterion (55) and satisfy the later complementarity property.

Given a self adjoint positive definite linear mapping $\mathcal{U} : \mathcal{Y} \to \mathcal{Y}$, our implementation chooses the initial iterates $x_0$ and $y_0$ as

$$x_0 = 0, \qquad y_0 = \arg\min \left\| \mathcal{A}^* \mathcal{U}^{-1} y - c \right\| = \mathcal{U}(\mathcal{A}\mathcal{A}^*)^{-1} \mathcal{A} c. \tag{56}$$ eq:initial-x-y

Another possibility would be to choose $x_0$ as the vector with minimum norm lying in the manifold $\{x \in \mathcal{X} : \mathcal{A}x = b\}$. However, the computational results reported in this paper are based on the choice of the initial iterates given by (56).

As mentioned at the end of the previous section, the operator $\mathcal{U}$ is chosen as

$$\mathcal{U} = \mathcal{A}\mathcal{A}^*. \tag{57}$$ eq:def-of-U

Note that this choice of $\mathcal{U}$ implies that $\left\| \mathcal{U}^{-1/2} \mathcal{A} \right\| = 1$, and hence

$$\tilde{\lambda}_x = \sigma \theta^{1/2}, \qquad \tilde{\lambda}_y = \sigma \theta^{-1/2}, \tag{58}$$ eq:lambda-x-y-s

in view of (41).

We will now discuss how to choose the initial value of $\theta$. Using (58), and relations (42) and (54) with $k = 1$, we easily see that

$$\epsilon_{P,1} = \epsilon_{P,1}(\theta) = \frac{\left\| \theta^{1/2} \sigma \mathcal{A} \left( \Pi_K(-s_0) \right) - b \right\|}{1 + \|b\|}, \qquad \epsilon_{D,1} = \frac{\|\Pi_K(-s_0)\|}{1 + \|c\|}, \tag{59}$$ eq:initial-erro

where $s_0 = c - \mathcal{A}^* \mathcal{U}^{-1} y_0$. Using (56) and the definition of $s_0$, we easily see that $\|s_0\| \leq \|c\|$, and hence that

$$\epsilon_{D,1} \leq \frac{\|s_0\|}{1 + \|c\|} \leq \frac{\|c\|}{1 + \|c\|} < 1.$$

Thus, $\epsilon_{D,1} = \mathcal{O}(1)$. We will choose the initial $\theta$ so as to enforce $\epsilon_{P,1}$ to be $\mathcal{O}(1)$. Observe that, by the first identity in (59), we have that $\epsilon_{P,1} \to \|b\| / (1 + \|b\|)$ as $\theta \to 0$. Given a constant $\rho \geq 1$, we check whether $\epsilon_{P,1}(1) \leq \rho$. If so, we set the initial $\theta$ to be 1, otherwise we successively divide the current value of $\theta$ by 2 until $\epsilon_{P.1}(\theta) \leq \rho$ is satisfied, and use this value as the initial $\theta$. The motivation behind this initial choice of $\theta$ is to guarantee that the initial primal relative residual $\epsilon_{P,k}$ is not too large at the first iteration of Algorithm 4.

Even though, the convergence rate bounds of Theorem 4.2 are guaranteed for a fixed value of $\theta$, we have used in our computational results the heuristic of changing $\theta$ every time a specified number $\bar{k}$ of iterations have been performed. The motivation for dynamically changing $\theta$, is that our preliminary computational experiments have suggested us that the performance of the method is improved as $\epsilon_{P,k}$ and $\epsilon_{D,k}$ are of the same order of magnitude. More specifically, if $\theta_k$ denotes the dynamic value of $\theta$ at the $k$th iteration of the algorithm, we use the following rule for updating $\theta_k$,

$$\theta_k = \begin{cases} \theta_{k-1}, & k \not\equiv 0 \bmod \bar{k} \text{ or } \gamma^{-1} \leq \epsilon_{P,k-1}/\epsilon_{D,k-1} \leq \gamma \\ \theta_{k-1} \cdot \tau, & k \equiv 0 \bmod \bar{k} \text{ and } \epsilon_{P,k-1}/\epsilon_{D,k-1} > \gamma \\ \theta_{k-1}/\tau, & k \equiv 0 \bmod \bar{k} \text{ and } \epsilon_{D,k-1}/\epsilon_{P,k-1} > \gamma \end{cases}, \qquad \forall k \geq 2, \tag{60}$$ eq:theta-k

for some prespecified integer $\bar{k} \geq 1$, and scalars $\gamma > 1$ and $0 < \tau < 1$. In our computational experiments, we have used $\bar{k} = 5$, $\gamma = 1.5$ and $\tau = 0.9$. Note that this update rule is motivated by the last observation in Section 4. In summary, the update rule changes the value of $\theta$ at most a single time in the right direction, so as to balance the sizes of the primal and dual relative residuals based on the information provided by their values at the previous iteration. We should emphasize that convergence rate bounds for Algorithm 4 endowed with this updating rule are not available, but we have observed that this variant of Algorithm 4 performs extremely well.

In our computational experiments, we will refer to the variant of Algorithm 4 described above as the *dynamically scaled adaptive block-decomposition* (DSA-BD) method for solving (1). This variant was implemented for spaces $\mathcal{X}$ and $\mathcal{Y}$, and cone $K$ given as in (4). Hence, our code is able to solve conic programming problems given in standard form (i.e., as in (1)) with $n_u$ unrestricted scalar variables, $n_l$ nonnegative scalar variables and an $n_s \times n_s$ positive semidefinite symmetric matrix variable. The inner products (before scaling) used in $\mathcal{X}$ and $\mathcal{Y}$ are the standard ones, namely: the scalar inner product in $\mathcal{Y}$ and the following inner product in $\mathcal{X}$

$$\langle x, \tilde{x} \rangle := x_v^T \tilde{x}_v + X_s \bullet \tilde{X}_s, \tag{61}$$ eq:def-inner

for every $x = (x_v, X_s) \in \mathbb{R}^{n_u+n_l} \times \mathcal{S}^{n_s}$ and $\tilde{x} = (\tilde{x}_v, \tilde{X}_s) \in \mathbb{R}^{n_u+n_l} \times \mathcal{S}^{n_s}$, where $X \bullet Y := Tr(X^T Y)$ for every $X, Y \in \mathcal{S}^{n_s}$.

We present a computational benchmark of our algorithm (DSA-BD) compared to the semismooth Newton-CG augmented Lagrangian (SDPNAL) method in [21] and the boundary point (BP) method in [8, 14]. We implemented the DSA-BD method in MATLAB using the SDPT3 data structures described in [19], but without exploiting any possible block sparsity on the semidefinite variable $X_s$. All the tests were made using a server with 2 Xeon X5460 processors at 3.16GHz and 32GB RAM.

Various large-scale SDP problems are solved to obtain this benchmark, ranging from purely random to SDP relaxations of combinatorial optimization problems such as the frequency assignment problem (FAP), the binary integer quadratic (BIQ) problem and the quadratic assignment problem (QAP). In the following subsections, we describe in detail the problems included in our computational tests but before that, we make some general remarks about how the results are reported on the several tables given below. In Table 1, we compare our method to BP and SDPNAL methods while, in Tables 3, 5, 6, 6 and 9, we compare it against SDPNAL only due to the fact that the current version of the BP method available to us only accepts conic optimization SDP problems without nonnegative scalar variables. In some of these tables, we report computational results for the same problem using two different tolerances. They are listed in two different rows of the table to the right of the name and size of the instance. We mark the time and the residual for a method in red, and also with an asterisk (*), whenever the instance cannot be solved to the required accuracy, with the convention that the time and residual reported are the ones obtained at the last iteration of the method. Also, the time marked in blue in a row is the best one among the times listed in that row under the convention that when a method cannot solve the instance, the corresponding time is assumed to be $\infty$. In Tables 2, 4, 7, 8 and 10, we report more detailed computational results obtained by our method DSA-BD.

Finally, Figure 1 plots the performance profiles (see [5]) of DSA-BD and SDPNAL methods based on all instances used in our benchmark. A point $(x, y)$ in the performance profile curve of a method indicates that $(100y)\%$ of the instances tested can be solved by the method in at most

$$2^x \times (\text{the cpu time of any other competing method}).$$

Other performance profiles based on instances belonging to the same class of conic programming problems will be reported in the subsequent subsections.

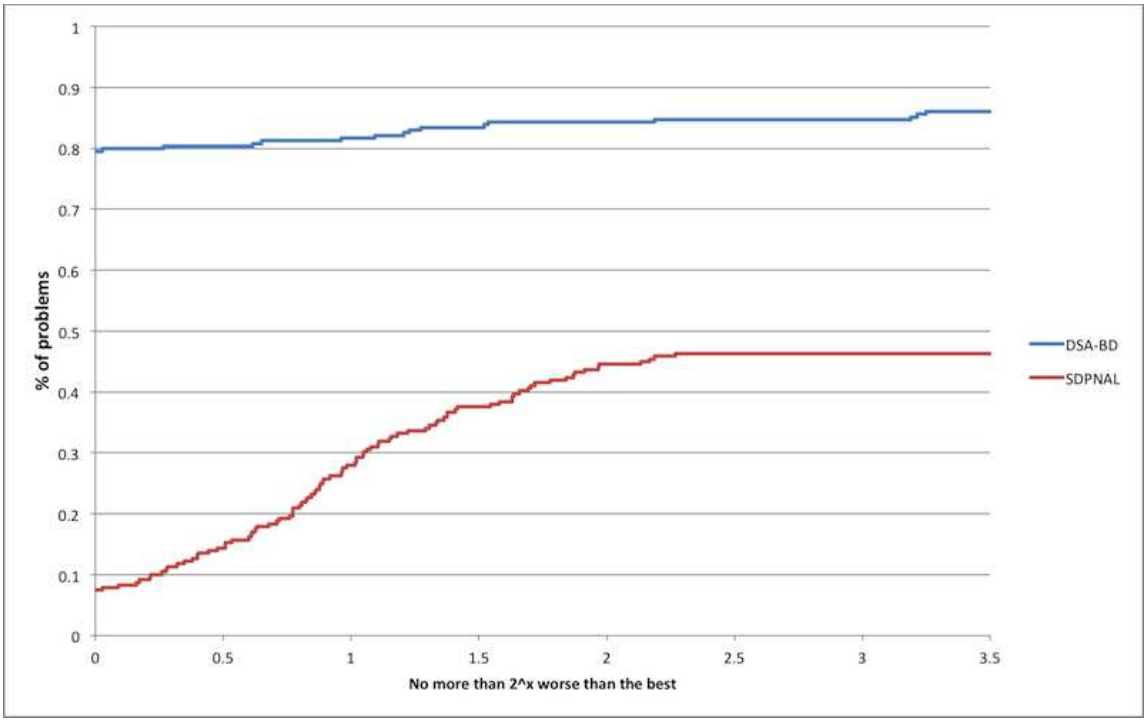



fig:Profile-6 Figure 1: Performance profiles of DSA-BD and SDPNAL for solving 229 conic semidefinite programming problems with accuracy $\bar{\epsilon} = 10^{-6}$.

### 5.1 Random SDPs

This subsection compares the performance of our method DSA-BD with that of BP and SDPNAL on a collection of random sparse SDP problems. These instances were also used in [8] to report the performance of BP introduced in [14].

Table 1 compares the three methods on a collection of random sparse SDP instances using the tolerance $\bar{\epsilon} = 10^{-6}$. Table 2 gives more detailed computational results on these instances obtained by our method DSA-BD, such as the objective values, number of iterations, and the primal and dual relative residuals. Figure 2 plots the performance profiles of the three methods based on these random sparse SDP instances only.

### 5.2 Frequency assignment problems

This subsection compares the performance of our method DSA-BD with that of SDPNAL on a collection of SDP relaxations of FAPs.

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 $G$ and an edge-weight matrix $W$, the frequency assignment problem on $G$ can be formulated as a $k$-cut problem

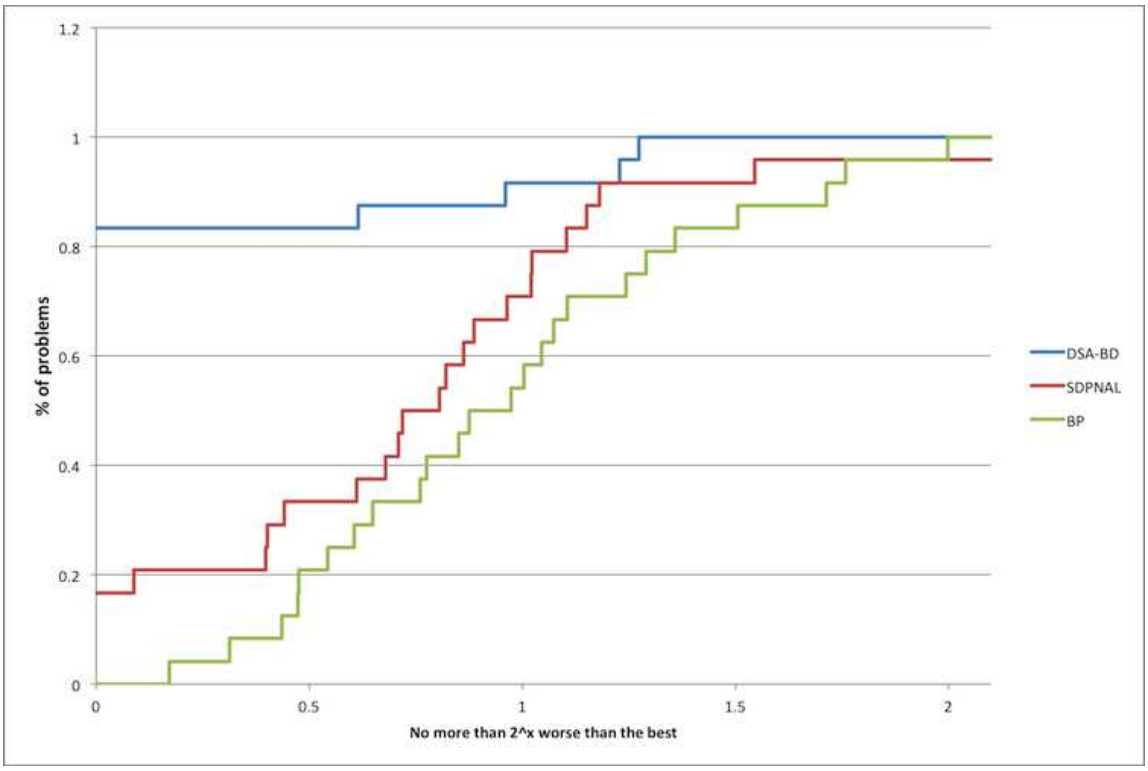
$$\begin{array}{ll} \max & \left[\left(\frac{k-1}{2k}\right) L(G,W) - \frac{1}{2}\mathrm{Diag}(We)\right] \bullet X \\ \text{s.t.} & -E^{ij} \bullet X \le 2/(k-1) \quad \forall\ (i,j) \\ & -E^{ij} \bullet X = 2/(k-1) \quad \forall\ (i,j) \in U \subseteq E \\ & \mathrm{diag}(X) = e,\ X \succeq 0,\ \mathrm{rank}(X)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

Table 1: Comparison of the methods on random SDP problems `tab:randomSDP`

| Problem | | $\max\{\epsilon_P, \epsilon_D\}$ | | | Time | | |
|---|---|---|---|---|---|---|---|
| Instance | $n_s \vert m$ | BP | DSA-BD | SDPNAL | BP | DSA-BD | SDPNAL |
| RAND-0.3k10k | 300\|10000 | 1.0 -6 | 9.4 -7 | 8.5 -7 | 29 | 27 | 14 |
| RAND-0.4k15k | 400\|15000 | 1.0 -6 | 9.8 -7 | 8.0 -7 | 61 | 52 | 22 |
| RAND-0.6k20k | 600\|20000 | 9.8 -7 | 1.0 -6 | 5.5 -7 | 144 | 55 | 36 |
| RAND-0.5k20k | 500\|20000 | 9.8 -7 | 9.7 -7 | 7.5 -7 | 110 | 76 | 32 |
| RAND-0.3k20k | 300\|20000 | 9.5 -7 | 9.8 -7 | 6.6 -7 | 26 | 18 | 39 |
| RAND-0.3k25k | 300\|25000 | 9.8 -7 | 1.0 -6 | 6.8 -7 | 73 | 65 | 69 |
| RAND-0.4k30k | 400\|30000 | 9.4 -7 | 9.7 -7 | 8.4 -7 | 37 | 24 | 54 |
| RAND-0.5k30k | 500\|30000 | 9.5 -7 | 9.6 -7 | 7.1 -7 | 52 | 29 | 53 |
| RAND-0.4k40k | 400\|40000 | 9.2 -7 | 9.4 -7 | 8.6 -6* | 115 | 92 | 134* |
| RAND-0.5k40k | 500\|40000 | 1.0 -6 | 9.6 -7 | 9.4 -7 | 53 | 31 | 57 |
| RAND-0.6k40k | 600\|40000 | 9.5 -7 | 9.8 -7 | 7.2 -7 | 87 | 41 | 68 |
| RAND-0.7k50k | 700\|50000 | 9.8 -7 | 9.7 -7 | 7.7 -7 | 140 | 65 | 88 |
| RAND-0.6k50k | 600\|50000 | 9.6 -7 | 9.6 -7 | 6.1 -7 | 82 | 42 | 122 |
| RAND-0.5k50k | 500\|50000 | 9.8 -7 | 9.6 -7 | 9.5 -7 | 89 | 66 | 100 |
| RAND-0.6k60k | 600\|60000 | 9.5 -7 | 9.2 -7 | 9.2 -7 | 96 | 57 | 101 |
| RAND-0.8k70k | 800\|70000 | 1.0 -6 | 9.8 -7 | 7.2 -7 | 196 | 80 | 131 |
| RAND-0.7k70k | 700\|70000 | 9.6 -7 | 9.9 -7 | 6.3 -7 | 126 | 63 | 127 |
| RAND-0.7k90k | 700\|90000 | 9.8 -7 | 9.5 -7 | 8.3 -7 | 202 | 145 | 192 |
| RAND-0.9k100k | 900\|100000 | 1.0 -6 | 1.0 -6 | 6.3 -7 | 263 | 102 | 200 |
| RAND-0.8k100k | 800\|100000 | 9.8 -7 | 9.6 -7 | 6.8 -7 | 203 | 113 | 250 |
| RAND-1.0k100k | 1000\|100000 | 9.7 -7 | 9.7 -7 | 7.1 -7 | 450 | 137 | 240 |
| RAND-0.8k110k | 800\|110000 | 9.9 -7 | 9.8 -7 | 3.8 -7 | 252 | 165 | 265 |
| RAND-0.9k140k | 900\|140000 | 9.3 -7 | 9.6 -7 | 9.2 -7 | 384 | 264 | 348 |
| RAND-1.0k150k | 1000\|150000 | 9.9 -7 | 9.7 -7 | 8.4 -7 | 459 | 194 | 394 |

Table 2: DSA-BD results on random SDP problems `tab:randomSDP-data`

| Instance | $n_s \vert m$ | $\langle C, X \rangle$ | $b^T y$ | Iter | $\epsilon_P$ | $\epsilon_D$ | Time |
|---|---|---|---|---|---|---|---|
| RAND-0.3k10k | 300\|10000 | 1.6597390 +2 | 1.6597440 +2 | 185 | 9.4 -7 | 4.0 -9 | 27 |
| RAND-0.4k15k | 400\|15000 | -6.5500010 +2 | -6.5500030 +2 | 202 | 9.8 -7 | 5.1 -9 | 52 |
| RAND-0.6k20k | 600\|20000 | 1.0452686 +3 | 1.0452662 +3 | 194 | 8.6 -7 | 1.0 -6 | 55 |
| RAND-0.5k20k | 500\|20000 | 3.2800420 +2 | 3.2800230 +2 | 206 | 8.6 -7 | 9.7 -7 | 76 |
| RAND-0.3k20k | 300\|20000 | 7.6135160 +2 | 7.6135210 +2 | 170 | 9.8 -7 | 9.3 -9 | 18 |
| RAND-0.3k25k | 300\|25000 | 7.3838100 +1 | 7.3838400 +1 | 264 | 1.0 -6 | 5.2 -9 | 65 |
| RAND-0.4k30k | 400\|30000 | 1.0721414 +3 | 1.0721394 +3 | 168 | 9.7 -7 | 8.9 -9 | 24 |
| RAND-0.5k30k | 500\|30000 | 1.1076268 +3 | 1.1076267 +3 | 215 | 9.6 -7 | 5.5 -9 | 29 |
| RAND-0.4k40k | 400\|40000 | 8.0576960 +2 | 8.0576860 +2 | 196 | 9.4 -7 | 6.0 -9 | 92 |
| RAND-0.5k40k | 500\|40000 | 8.1661180 +2 | 8.1661080 +2 | 183 | 9.6 -7 | 5.0 -9 | 31 |
| RAND-0.6k40k | 600\|40000 | 3.0661780 +2 | 3.0661720 +2 | 209 | 9.8 -7 | 1.0 -8 | 41 |
| RAND-0.7k50k | 700\|50000 | 3.1320370 +2 | 3.1320280 +2 | 236 | 9.7 -7 | 4.7 -9 | 65 |
| RAND-0.6k50k | 600\|50000 | -3.8641380 +2 | -3.8641360 +2 | 196 | 9.6 -7 | 8.2 -9 | 42 |
| RAND-0.5k50k | 500\|50000 | 3.6494490 +2 | 3.6494520 +2 | 177 | 9.6 -7 | 4.9 -9 | 66 |
| RAND-0.6k60k | 600\|60000 | 6.4173730 +2 | 6.4173680 +2 | 183 | 9.2 -7 | 5.6 -9 | 57 |
| RAND-0.8k70k | 800\|70000 | 2.3313969 +3 | 2.3313957 +3 | 220 | 9.8 -7 | 9.4 -9 | 80 |
| RAND-0.7k70k | 700\|70000 | -3.6955780 +2 | -3.6955870 +2 | 190 | 9.9 -7 | 6.6 -9 | 63 |
| RAND-0.7k90k | 700\|90000 | -2.6758200 +1 | -2.6755500 +1 | 183 | 9.5 -7 | 7.9 -9 | 145 |
| RAND-0.9k100k | 900\|100000 | 9.5422350 +2 | 9.5422290 +2 | 206 | 1.0 -6 | 7.9 -9 | 102 |
| RAND-0.8k100k | 800\|100000 | 2.2592888 +3 | 2.2592881 +3 | 196 | 9.6 -7 | 5.8 -9 | 113 |
| RAND-1.0k100k | 1000\|100000 | 3.0963606 +3 | 3.0963602 +3 | 235 | 9.7 -7 | 8.9 -9 | 137 |
| RAND-0.8k110k | 800\|110000 | 1.8579204 +3 | 1.8579207 +3 | 187 | 9.8 -7 | 9.3 -9 | 165 |
| RAND-0.9k140k | 900\|140000 | 2.3198295 +3 | 2.3198295 +3 | 193 | 9.6 -7 | 9.4 -9 | 264 |
| RAND-1.0k150k | 1000\|150000 | 1.0528840 +3 | 1.0528864 +3 | 205 | 9.7 -7 | 4.9 -9 | 194 |




file-6-RANDOM Figure 2: Performance profiles of DSA-BD, SDPNAL and BP for solving 24 random SDP problems with accuracy $\bar{\epsilon} = 10^{-6}$.

Table 3: Comparison of the methods on FAPs tab:FAP

| Instance | Problem $n_s; n_l \mid m$ | $\max\{\epsilon_P, \epsilon_D\}$ DSA-BD | $\max\{\epsilon_P, \epsilon_D\}$ SDPNAL | Time DSA-BD | Time SDPNAL |
|---|---|---|---|---|---|
| fap05 | 84;3263\|3570 | 9.5 -6 | 1.7 -5* | 8 | 18* |
| | | 9.9 -7 | 1.7 -5* | 14 | 18* |
| fap06 | 93;3997\|4371 | 9.9 -6 | 9.4 -6 | 8 | 10 |
| | | 1.0 -6 | 6.7 -7 | 11 | 13 |
| fap07 | 98;4139\|4851 | 1.0 -5 | 9.4 -6 | 5 | 8 |
| | | 1.0 -6 | 9.9 -7 | 10 | 13 |
| fap08 | 120;6668\|7260 | 1.0 -5 | 7.6 -6 | 9 | 6 |
| | | 9.9 -7 | 9.5 -7 | 16 | 10 |
| fap09 | 174;14025\|15225 | 9.7 -6 | 8.8 -6 | 15 | 15 |
| | | 9.9 -7 | 9.4 -7 | 19 | 19 |
| fap10 | 183;13754\|14479 | 9.9 -6 | 6.8 -6 | 38 | 17 |
| | | 1.0 -6 | 9.3 -7 | 74 | 32 |
| fap11 | 252;23275\|24292 | 9.9 -6 | 6.9 -6 | 65 | 39 |
| | | 1.0 -6 | 9.7 -7 | 132 | 78 |
| fap12 | 369;24410\|26462 | 1.0 -5 | 8.2 -6 | 103 | 79 |
| | | 1.0 -6 | 2.5 -6* | 259 | 188* |
| fap25 | 2118;311044\|322924 | 1.0 -5 | 7.4 -6 | 7444 | 14517 |
| | | 9.9 -7 | 5.1 -6* | 23572 | 29557* |
| fap36 | 4110;1112293\|1154467 | 1.0 -5 | 8.5 -6 | 49617 | 29485 |
| | | 9.8 -7 | 5.6 -6* | 148855 | 72063* |

Table 4: DSA-BD results on FAPs tab:FAP-data

| Instance | $n_s; n_l \mid m$ | $\langle C, X\rangle$ | $b^T y$ | Iter | $\epsilon_P$ | $\epsilon_D$ | Time |
|---|---|---|---|---|---|---|---|
| fap05 | 84;3263\|3570 | 3.0830000 -1 | 3.0830000 -1 | 2424 | 9.9 -7 | 7.8 -7 | 14 |
| fap06 | 93;3997\|4371 | 4.5933000 -1 | 4.5934000 -1 | 1694 | 1.0 -6 | 5.5 -7 | 11 |
| fap07 | 98;4139;\|4851 | 2.1176000 +0 | 2.1176000 +0 | 1766 | 1.0 -6 | 4.2 -7 | 10 |
| fap08 | 120;6668\|7260 | 2.4363000 +0 | 2.4363000 +0 | 1735 | 9.9 -7 | 8.2 -7 | 16 |
| fap09 | 174;14025;\|15225 | 1.0797800 +1 | 1.0797800 +1 | 1128 | 9.9 -7 | 8.6 -7 | 19 |
| fap10 | 183;13754;\|14479 | 9.6383000 -3 | 9.7289000 -3 | 3315 | 1.0 -6 | 8.0 -7 | 74 |
| fap11 | 252;23275\|24292 | 2.9713000 -2 | 2.9856000 -2 | 3380 | 1.0 -6 | 8.0 -7 | 132 |
| fap12 | 369;24410\|26462 | 2.7317000 -1 | 2.7341000 -1 | 4004 | 1.0 -6 | 8.2 -7 | 259 |
| fap25 | 2118;311044\|322924 | 1.2877800 +1 | 1.2880000 +1 | 5576 | 9.9 -7 | 8.1 -7 | 23572 |
| fap36 | 4110;1112293\|1154467 | 6.9857000 +1 | 6.9860700 +1 | 4013 | 9.8 -7 | 9.4 -7 | 148855 |

where $k > 1$ 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 $i$th 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

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

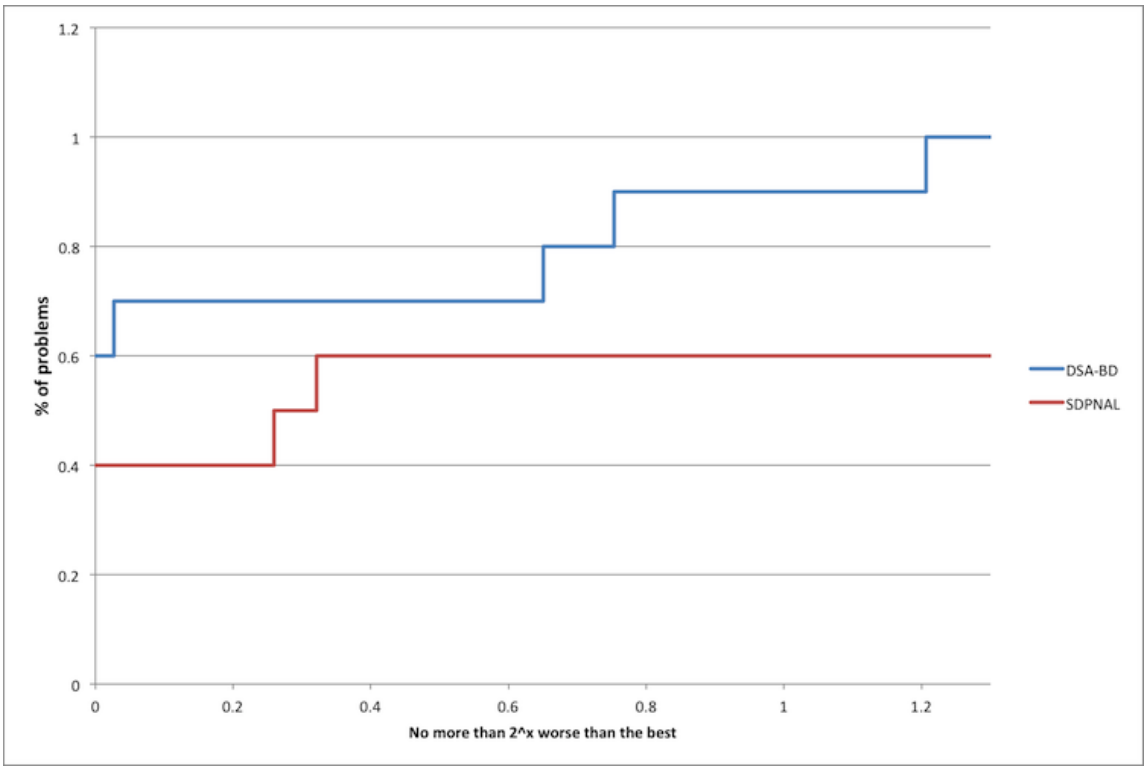
Table 3 compares the two methods on a collection of SDP relaxations of FAPs using the tolerances $\bar{\epsilon} = 10^{-5}, 10^{-6}$. In this table, computational results for each instance are reported in two rows, the first one for $\bar{\epsilon} = 10^{-5}$, and the second one for $\bar{\epsilon} = 10^{-6}$. Table 4 gives more detailed computational results on these instances obtained by our method DSA-BD, such as the objective values, number of iterations, and the primal and dual relative residuals. Figure 3 plots the performance profiles of both methods based on these SDP relaxations of FAPs.

Note that our method performs better than SDPNAL on large SDP relaxations of FAPs (i.e., `fap25` and `fap36`).

ger-Quadratic

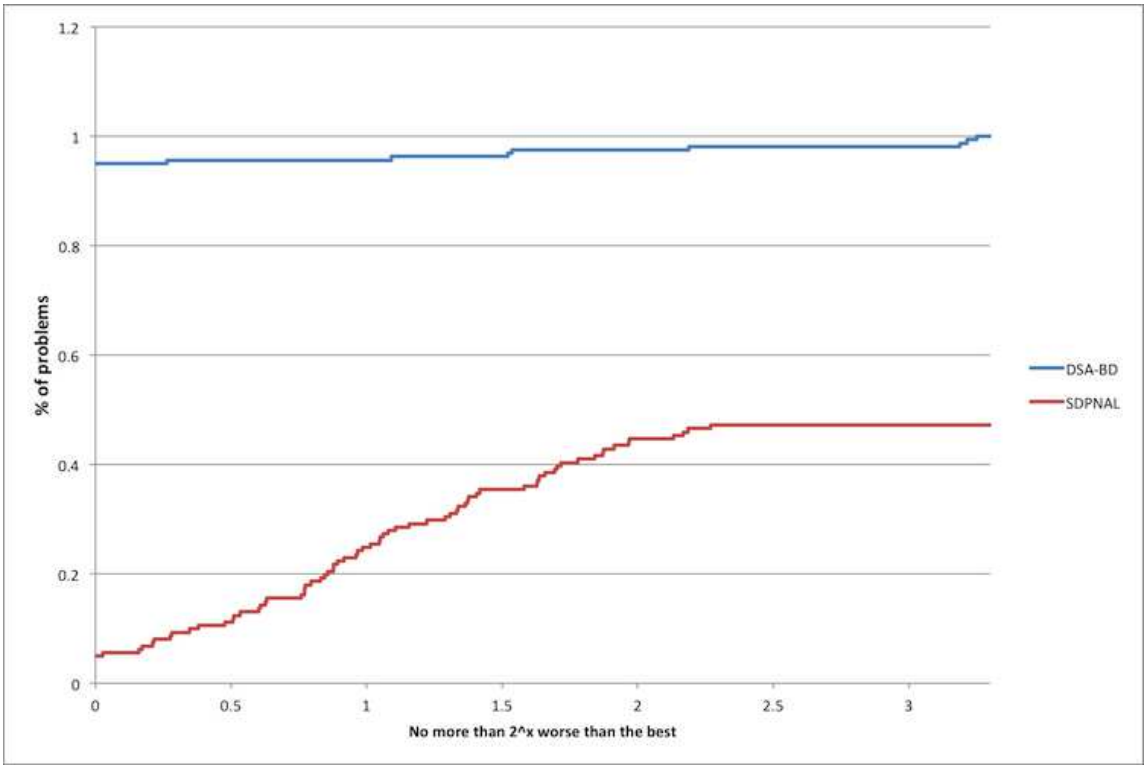
### 5.3 Binary integer quadratic problems

This subsection compares the performance of our method DSA-BD with that of SDPNAL on a collection of SDP relaxations of BIQ problems.




Profile-6-FAP

Figure 3: Performance profiles of DSA-BD and SDPNAL for solving 10 SDP relaxations of FAPs with accuracy $\epsilon = 10^{-6}$.




Profile-6-BIQ Figure 4: Performance profiles of DSA-BD and SDPNAL for solving 161 SDP relaxations of BIQ problems with accuracy $\bar{\epsilon} = 10^{-6}$.

The SDP relaxation of the BIQ problem can be described as follows (see for example Section 7 in [21]). Given an $n \times n$ symmetric matrix $Q$ the BIQ problem can be formulated as

$$\min \left\{ x^T Q x : x \in \{0,1\}^n \right\}.$$

By representing the binary set $\{0,1\}^n$ as $\left\{ x \in \mathbb{R}^n | x_i^2 - x_i = 0 \right\}$, we obtain an SDP relaxation as follows

$$\begin{array}{ll} \min & Q \bullet X \\ \text{s.t.} & \mathrm{diag}(X) - x = 0, \ \alpha = 1, \\ & \begin{bmatrix} X & x \\ x^T & \alpha \end{bmatrix} \succeq 0, \ X \geq 0, \ x \geq 0. \end{array}$$

Tables 5 and 6 compare the two methods on a collection of SDP relaxations of BIQ problems using the tolerance $\bar{\epsilon} = 10^{-6}$. Tables 7 and 8 give more detailed computational results on these instances obtained by our method DSA-BD, such as the objective values, number of iterations, and the primal and dual relative residuals. Figure 4 plots the performance profiles of both methods based on these SDP relaxations of BIQ problems.

Note that SDPNAL is outperformed by our method in almost all of the SDP relaxations of BIQ problems tested and it fails to converge to an accuracy less than or equal to $10^{-6}$ on more than half of these instances. On the other hand, our method DSA-BD converges to an accuracy less than $10^{-6}$ in all of the SDP relaxations of BIQ problems tested.

Table 5: Comparison of the methods on BIQ problems tab:BIQ1

| Problem | | $\max\{\epsilon_P, \epsilon_D\}$ | | Time | |
|---|---|---|---|---|---|
| Instance | $n_s; n_l \| m$ | DSA-BD | SDPNAL | DSA-BD | SDPNAL |
| be100.1 | 101;5151\|5252 | 1.0 -6 | 7.3 -7 | 21 | 55 |
| be100.10 | 101;5151\|5252 | 1.0 -6 | 8.3 -7 | 13 | 41 |
| be100.2 | 101;5151\|5252 | 1.0 -6 | 8.3 -7 | 17 | 51 |
| be100.3 | 101;5151\|5252 | 1.0 -6 | 9.0 -7 | 18 | 69 |
| be100.4 | 101;5151\|5252 | 1.0 -6 | 6.7 -6* | 19 | 83* |
| be100.5 | 101;5151\|5252 | 1.0 -6 | 6.8 -7 | 17 | 51 |
| be100.6 | 101;5151\|5252 | 9.9 -7 | 8.2 -6* | 14 | 68* |
| be100.7 | 101;5151\|5252 | 1.0 -6 | 8.0 -7 | 17 | 30 |
| be100.8 | 101;5151\|5252 | 9.2 -7 | 8.2 -7 | 13 | 21 |
| be100.9 | 101;5151\|5252 | 1.0 -6 | 5.0 -7 | 16 | 58 |
| be120.3.1 | 121;7381\|7502 | 9.9 -7 | 7.5 -7 | 25 | 80 |
| be120.3.10 | 121;7381\|7502 | 9.9 -7 | 6.6 -7 | 21 | 100 |
| be120.3.2 | 121;7381\|7502 | 1.0 -6 | 7.7 -6* | 25 | 88* |
| be120.3.3 | 121;7381\|7502 | 1.0 -6 | 6.0 -7 | 19 | 36 |
| be120.3.4 | 121;7381\|7502 | 9.8 -7 | 8.3 -7 | 20 | 51 |
| be120.3.5 | 121;7381\|7502 | 1.0 -6 | 1.3 -5* | 30 | 98* |
| be120.3.6 | 121;7381\|7502 | 1.0 -6 | 1.5 -5* | 29 | 93* |
| be120.3.7 | 121;7381\|7502 | 1.0 -6 | 4.1 -5* | 46 | 102* |
| be120.3.8 | 121;7381\|7502 | 1.0 -6 | 3.5 -5* | 40 | 104* |
| be120.3.9 | 121;7381\|7502 | 1.0 -6 | 5.5 -5* | 38 | 74* |
| be120.8.1 | 121;7381\|7502 | 9.9 -7 | 5.9 -7 | 21 | 52 |
| be120.8.10 | 121;7381\|7502 | 1.0 -6 | 9.9 -7 | 26 | 65 |
| be120.8.2 | 121;7381\|7502 | 1.0 -6 | 2.9 -5* | 32 | 105* |
| be120.8.3 | 121;7381\|7502 | 1.0 -6 | 7.6 -7 | 27 | 49 |
| be120.8.4 | 121;7381\|7502 | 1.0 -6 | 9.5 -6* | 25 | 82* |
| be120.8.5 | 121;7381\|7502 | 1.0 -6 | 8.9 -6* | 27 | 111* |
| be120.8.6 | 121;7381\|7502 | 9.9 -7 | 9.8 -7 | 24 | 91 |
| be120.8.7 | 121;7381\|7502 | 9.9 -7 | 6.6 -7 | 20 | 45 |
| be120.8.8 | 121;7381\|7502 | 9.7 -7 | 8.5 -7 | 19 | 19 |
| be120.8.9 | 121;7381\|7502 | 1.0 -6 | 9.9 -7 | 19 | 34 |
| be150.3.1 | 151;11476\|11627 | 1.0 -6 | 1.5 -5* | 35 | 146* |
| be150.3.10 | 151;11476\|11627 | 1.0 -6 | 1.3 -5* | 53 | 177* |
| be150.3.2 | 151;11476\|11627 | 1.0 -6 | 2.0 -5* | 45 | 125* |
| be150.3.3 | 151;11476\|11627 | 1.0 -6 | 6.1 -7 | 36 | 133 |
| be150.3.4 | 151;11476\|11627 | 1.0 -6 | 6.7 -7 | 36 | 60 |
| be150.3.5 | 151;11476\|11627 | 1.0 -6 | 9.0 -6* | 44 | 135* |
| be150.3.7 | 151;11476\|11627 | 1.0 -6 | 5.2 -7 | 36 | 94 |
| be150.3.8 | 151;11476\|11627 | 1.0 -6 | 2.2 -5* | 53 | 84* |
| be150.3.9 | 151;11476\|11627 | 9.5 -7 | 7.1 -7 | 35 | 49 |
| be150.8.1 | 151;11476\|11627 | 1.0 -6 | 8.2 -7 | 32 | 140 |
| be150.8.10 | 151;11476\|11627 | 1.0 -6 | 6.2 -7 | 38 | 80 |
| be150.8.2 | 151;11476\|11627 | 1.0 -6 | 6.9 -7 | 34 | 63 |
| be150.8.3 | 151;11476\|11627 | 1.0 -6 | 1.0 -6 | 36 | 119 |
| be150.8.4 | 151;11476\|11627 | 1.0 -6 | 5.7 -7 | 40 | 78 |
| be150.8.5 | 151;11476\|11627 | 9.9 -7 | 1.1 -5* | 35 | 146* |
| be150.8.6 | 151;11476\|11627 | 1.0 -6 | 9.4 -6* | 40 | 119* |
| be150.8.7 | 151;11476\|11627 | 1.0 -6 | 2.3 -5* | 45 | 143* |
| be150.8.8 | 151;11476\|11627 | 9.8 -7 | 1.3 -5* | 52 | 136* |
| be150.8.9 | 151;11476\|11627 | 1.0 -6 | 8.4 -7 | 46 | 108 |
| be200.3.1 | 201;20301\|20502 | 1.0 -6 | 4.8 -6* | 67 | 204* |
| be200.3.10 | 201;20301\|20502 | 1.0 -6 | 1.9 -5* | 92 | 242* |
| be200.3.2 | 201;20301\|20502 | 1.0 -6 | 6.7 -7 | 73 | 134 |
| be200.3.3 | 201;20301\|20502 | 1.0 -6 | 3.6 -5* | 138 | 271* |
| be200.3.4 | 201;20301\|20502 | 1.0 -6 | 1.6 -5* | 87 | 258* |
| be200.3.5 | 201;20301\|20502 | 1.0 -6 | 1.4 -5* | 111 | 266* |
| be200.3.6 | 201;20301\|20502 | 1.0 -6 | 7.2 -7 | 68 | 182 |
| be200.3.7 | 201;20301\|20502 | 1.0 -6 | 6.8 -7 | 84 | 261 |
| be200.3.8 | 201;20301\|20502 | 1.0 -6 | 9.1 -7 | 76 | 158 |
| be200.3.9 | 201;20301\|20502 | 1.0 -6 | 3.2 -5* | 161 | 213* |
| be200.8.1 | 201;20301\|20502 | 1.0 -6 | 3.0 -5* | 117 | 234* |
| be200.8.10 | 201;20301\|20502 | 1.0 -6 | 1.6 -5* | 77 | 243* |
| be200.8.2 | 201;20301\|20502 | 1.0 -6 | 5.3 -7 | 61 | 74 |
| be200.8.3 | 201;20301\|20502 | 1.0 -6 | 9.5 -6* | 94 | 261* |
| be200.8.4 | 201;20301\|20502 | 1.0 -6 | 5.8 -7 | 66 | 133 |
| be200.8.5 | 201;20301\|20502 | 1.0 -6 | 1.2 -5* | 81 | 255* |
| be200.8.6 | 201;20301\|20502 | 1.0 -6 | 1.7 -5* | 94 | 269* |
| be200.8.7 | 201;20301\|20502 | 1.0 -6 | 7.0 -7 | 69 | 119 |
| be200.8.8 | 201;20301\|20502 | 1.0 -6 | 6.6 -7 | 78 | 153 |
| be200.8.9 | 201;20301\|20502 | 1.0 -6 | 1.2 -5* | 87 | 246* |
| be250.1 | 251;31626\|31877 | 1.0 -6 | 1.0 -4* | 209 | 360* |
| be250.10 | 251;31626\|31877 | 1.0 -6 | 4.8 -5* | 230 | 387* |
| be250.2 | 251;31626\|31877 | 1.0 -6 | 3.0 -5* | 183 | 450* |
| be250.3 | 251;31626\|31877 | 9.9 -7 | 4.8 -5* | 161 | 420* |
| be250.4 | 251;31626\|31877 | 1.0 -6 | 6.5 -5* | 362 | 420* |
| be250.5 | 251;31626\|31877 | 1.0 -6 | 3.3 -5* | 193 | 358* |
| be250.6 | 251;31626\|31877 | 1.0 -6 | 4.2 -5* | 188 | 400* |
| be250.7 | 251;31626\|31877 | 1.0 -6 | 3.6 -5* | 207 | 367* |
| be250.8 | 251;31626\|31877 | 1.0 -6 | 1.6 -5* | 176 | 444* |
| be250.9 | 251;31626\|31877 | 1.0 -6 | 1.4 -4* | 239 | 359* |
| bqp100-1 | 101;5151\|5252 | 9.9 -7 | 7.6 -7 | 15 | 58 |
| bqp100-10 | 101;5151\|5252 | 1.0 -6 | 1.7 -5* | 27 | 72* |
| bqp100-2 | 101;5151\|5252 | 1.0 -6 | 1.4 -4* | 28 | 60* |
| bqp100-3 | 101;5151\|5252 | 1.0 -6 | 8.0 -7 | 43 | 65 |
| bqp100-4 | 101;5151\|5252 | 1.0 -6 | 7.7 -6* | 25 | 84* |
| bqp100-5 | 101;5151\|5252 | 1.0 -6 | 4.0 -5* | 26 | 64* |
| bqp100-6 | 101;5151\|5252 | 1.0 -6 | 8.6 -7 | 16 | 74 |
| bqp100-7 | 101;5151\|5252 | 1.0 -6 | 8.7 -7 | 14 | 61 |
| bqp100-8 | 101;5151\|5252 | 1.0 -6 | 2.1 -5* | 25 | 75* |

Table 6: Comparison of the methods on BIQ problems tab:BIQ2

| Problem | | $\max\{\epsilon_P, \epsilon_D\}$ | | Time | |
|---|---|---|---|---|---|
| Instance | $n_s; n_l \| m$ | DSA-BD | SDPNAL | DSA-BD | SDPNAL |
| bqp100-9 | 101;5151\|5252 | 1.0 -6 | 3.7 -5* | 35 | 76* |
| bqp250-1 | 251;31626\|31877 | 1.0 -6 | 5.2 -7 | 182 | 391 |
| bqp250-10 | 251;31626\|31877 | 1.0 -6 | 9.5 -7 | 131 | 469 |
| bqp250-2 | 251;31626\|31877 | 1.0 -6 | 6.1 -5* | 191 | 404* |
| bqp250-3 | 251;31626\|31877 | 1.0 -6 | 8.0 -7 | 209 | 266 |
| bqp250-4 | 251;31626\|31877 | 1.0 -6 | 2.2 -5* | 151 | 398* |
| bqp250-5 | 251;31626\|31877 | 1.0 -6 | 5.2 -5* | 272 | 418* |
| bqp250-6 | 251;31626\|31877 | 1.0 -6 | 4.0 -5* | 221 | 421* |
| bqp250-7 | 251;31626\|31877 | 1.0 -6 | 5.5 -7 | 193 | 400 |
| bqp250-8 | 251;31626\|31877 | 1.0 -6 | 9.4 -6* | 144 | 392* |
| bqp250-9 | 251;31626\|31877 | 1.0 -6 | 1.7 -5* | 195 | 396* |
| bqp50-1 | 51;1326\|1377 | 1.0 -6 | 1.2 -5* | 9 | 32* |
| bqp50-10 | 51;1326\|1377 | 9.5 -7 | 5.8 -7 | 6 | 8 |
| bqp50-2 | 51;1326\|1377 | 9.9 -7 | 2.5 -5* | 20 | 19* |
| bqp50-3 | 51;1326\|1377 | 1.0 -6 | 8.2 -7 | 7 | 23 |
| bqp50-4 | 51;1326\|1377 | 1.0 -6 | 4.9 -5* | 42 | 37* |
| bqp50-5 | 51;1326\|1377 | 1.0 -6 | 4.4 -5* | 9 | 19* |
| bqp50-7 | 51;1326\|1377 | 1.0 -6 | 4.9 -7 | 7 | 8 |
| bqp50-8 | 51;1326\|1377 | 9.6 -7 | 6.9 -7 | 7 | 14 |
| bqp50-9 | 51;1326\|1377 | 1.0 -6 | 6.2 -7 | 6 | 8 |
| bqp500-1 | 501;125751\|126252 | 1.0 -6 | 4.2 -7 | 1230 | 1385 |
| bqp500-10 | 501;125751\|126252 | 1.0 -6 | 9.7 -7 | 1241 | 1893 |
| bqp500-2 | 501;125751\|126252 | 1.0 -6 | 6.2 -5* | 1364 | 2113* |
| bqp500-3 | 501;125751\|126252 | 1.0 -6 | 9.2 -7 | 1232 | 1433 |
| bqp500-4 | 501;125751\|126252 | 1.0 -6 | 9.6 -7 | 1303 | 1852 |
| bqp500-5 | 501;125751\|126252 | 1.0 -6 | 6.6 -5* | 1294 | 2210* |
| bqp500-6 | 501;125751\|126252 | 1.0 -6 | 9.3 -7 | 1236 | 2198 |
| bqp500-7 | 501;125751\|126252 | 1.0 -6 | 5.2 -5* | 1280 | 1916* |
| bqp500-8 | 501;125751\|126252 | 1.0 -6 | 7.7 -7 | 1314 | 1871 |
| bqp500-9 | 501;125751\|126252 | 1.0 -6 | 5.7 -5* | 1187 | 2195* |
| gka10b | 126;8001\|8127 | 1.0 -6 | 2.3 -4* | 22 | 68* |
| gka10d | 101;5151\|5252 | 1.0 -6 | 5.8 -7 | 17 | 31 |
| gka1b | 21;231\|252 | 9.2 -7 | 2.9 -7 | 3 | 1 |
| gka1c | 41;861\|902 | 1.0 -6 | 1.4 -5* | 11 | 21* |
| gka1d | 101;5151\|5252 | 1.0 -6 | 1.6 -5* | 33 | 68* |
| gka1e | 201;20301\|20502 | 1.0 -6 | 7.8 -5* | 181 | 364* |
| gka1f | 501;125751\|126252 | 1.0 -6 | 7.4 -5* | 2195 | 3531* |
| gka2b | 31;496\|527 | 1.0 -6 | 7.9 -5* | 6 | 13* |
| gka2c | 51;1326\|1377 | 9.9 -7 | 5.4 -7 | 7 | 17 |
| gka2d | 101;5151\|5252 | 1.0 -6 | 7.7 -6* | 17 | 76* |
| gka2e | 201;20301\|20502 | 1.0 -6 | 9.2 -7 | 121 | 239 |
| gka2f | 501;125751\|126252 | 1.0 -6 | 7.9 -5* | 2488 | 3729* |
| gka3a | 71;2556\|2627 | 9.9 -7 | 7.9 -7 | 11 | 28 |
| gka3b | 41;861\|902 | 9.7 -7 | 1.6 -7 | 17 | 2 |
| gka3c | 61;1891\|1952 | 9.9 -7 | 8.8 -7 | 8 | 14 |
| gka3d | 101;5151\|5252 | 1.0 -6 | 1.2 -5* | 30 | 85* |
| gka3e | 201;20301\|20502 | 1.0 -6 | 3.6 -5* | 161 | 305* |
| gka3f | 501;125751\|126252 | 1.0 -6 | 3.7 -5* | 2222 | 3514* |
| gka4a | 81;3321\|3402 | 1.0 -6 | 5.6 -6* | 19 | 72* |
| gka4b | 51;1326\|1377 | 9.9 -7 | 5.0 -7 | 21 | 2 |
| gka4c | 71;2556\|2627 | 1.0 -6 | 7.6 -6* | 14 | 64* |
| gka4d | 101;5151\|5252 | 1.0 -6 | 7.2 -6* | 19 | 64* |
| gka4e | 201;20301\|20502 | 1.0 -6 | 1.2 -5* | 187 | 382* |
| gka4f | 501;125751\|126252 | 1.0 -6 | 3.4 -5* | 2348 | 3908* |
| gka5a | 51;1326\|1377 | 9.9 -7 | 6.0 -7 | 7 | 10 |
| gka5b | 61;1891\|1952 | 9.9 -7 | 1.4 -7 | 25 | 3 |
| gka5c | 81;3321\|3402 | 1.0 -6 | 5.6 -6* | 22 | 36* |
| gka5d | 101;5151\|5252 | 1.0 -6 | 8.6 -7 | 19 | 59 |
| gka5e | 201;20301\|20502 | 1.0 -6 | 4.0 -5* | 149 | 320* |
| gka5f | 501;125751\|126252 | 1.0 -6 | 1.8 -5* | 2210 | 3798* |
| gka6a | 31;496\|527 | 9.9 -7 | 9.4 -7 | 5 | 6 |
| gka6b | 71;2556\|2627 | 9.9 -7 | 5.3 -8 | 27 | 6 |
| gka6c | 91;4186\|4277 | 1.0 -6 | 2.7 -5* | 31 | 69* |
| gka6d | 101;5151\|5252 | 9.9 -7 | 7.0 -6* | 18 | 46* |
| gka7a | 31;496\|527 | 1.0 -6 | 9.1 -7 | 5 | 4 |
| gka7b | 81;3321\|3402 | 9.9 -7 | 5.2 -7 | 3 | 10 |
| gka7c | 101;5151\|5252 | 1.0 -6 | 5.9 -5* | 26 | 62* |
| gka7d | 101;5151\|5252 | 1.0 -6 | 5.3 -7 | 16 | 20 |
| gka8a | 101;5151\|5252 | 1.0 -6 | 4.2 -5* | 33 | 99* |
| gka8b | 91;4186\|4277 | 9.9 -7 | 8.5 -7 | 49 | 17 |
| gka8d | 101;5151\|5252 | 1.0 -6 | 7.0 -7 | 31 | 52 |
| gka9b | 101;5151\|5252 | 9.5 -7 | 3.3 -7 | 46 | 21 |
| gka9d | 101;5151\|5252 | 1.0 -6 | 7.0 -7 | 15 | 38 |

Table 7: DSA-BD results on BIQ problems `tab:BIQ-data1`

| Instance | n {s}:n {l}\|m | $\langle C, X \rangle$ | $b^T y$ | Iter | $\epsilon_P$ | $\epsilon_D$ | Time |
|---|---|---|---|---|---|---|---|
| be100.1 | 101;5151\|5252 | -2.0021326 +4 | -2.0021284 +4 | 2762 | 4.3 -7 | 1.0 -6 | 21 |
| be100.10 | 101;5151\|5252 | -1.6408506 +4 | -1.6408506 +4 | 1757 | 9.8 -7 | 1.0 -6 | 13 |
| be100.2 | 101;5151\|5252 | -1.7988702 +4 | -1.7988693 +4 | 2249 | 1.0 -6 | 6.8 -7 | 17 |
| be100.3 | 101;5151\|5252 | -1.8231054 +4 | -1.8231144 +4 | 2307 | 4.5 -7 | 1.0 -6 | 18 |
| be100.4 | 101;5151\|5252 | -1.9841800 +4 | -1.9841754 +4 | 2483 | 1.0 -6 | 9.0 -7 | 19 |
| be100.5 | 101;5151\|5252 | -1.6888702 +4 | -1.6888698 +4 | 2158 | 9.9 -7 | 1.0 -6 | 17 |
| be100.6 | 101;5151\|5252 | -1.8148225 +4 | -1.8148220 +4 | 1907 | 9.1 -7 | 9.9 -7 | 14 |
| be100.7 | 101;5151\|5252 | -1.9700850 +4 | -1.9700852 +4 | 2298 | 1.0 -6 | 8.9 -7 | 17 |
| be100.8 | 101;5151\|5252 | -1.9946409 +4 | -1.9946451 +4 | 1759 | 6.8 -7 | 9.2 -7 | 13 |
| be100.9 | 101;5151\|5252 | -1.4263373 +4 | -1.4263380 +4 | 2054 | 1.0 -6 | 7.4 -7 | 16 |
| be120.3.1 | 121;7381\|7502 | -1.3803561 +4 | -1.3803555 +4 | 2338 | 9.9 -7 | 9.7 -7 | 25 |
| be120.3.10 | 121;7381\|7502 | -1.2930857 +4 | -1.2930864 +4 | 1969 | 9.9 -7 | 9.0 -7 | 21 |
| be120.3.2 | 121;7381\|7502 | -1.3626630 +4 | -1.3626637 +4 | 2407 | 1.0 -6 | 7.6 -7 | 25 |
| be120.3.3 | 121;7381\|7502 | -1.2987898 +4 | -1.2987905 +4 | 1851 | 1.0 -6 | 6.8 -7 | 19 |
| be120.3.4 | 121;7381\|7502 | -1.4511234 +4 | -1.4511320 +4 | 1867 | 7.4 -7 | 9.8 -7 | 20 |
| be120.3.5 | 121;7381\|7502 | -1.1991909 +4 | -1.1991906 +4 | 2834 | 9.5 -7 | 1.0 -6 | 30 |
| be120.3.6 | 121;7381\|7502 | -1.3432058 +4 | -1.3432057 +4 | 2717 | 1.0 -6 | 9.3 -7 | 29 |
| be120.3.7 | 121;7381\|7502 | -1.4564113 +4 | -1.4564109 +4 | 4381 | 8.1 -7 | 1.0 -6 | 46 |
| be120.3.8 | 121;7381\|7502 | -1.5303024 +4 | -1.5303018 +4 | 3775 | 1.0 -6 | 6.7 -7 | 40 |
| be120.3.9 | 121;7381\|7502 | -1.1241319 +4 | -1.1241320 +4 | 3679 | 1.0 -6 | 6.6 -7 | 38 |
| be120.8.1 | 121;7381\|7502 | -2.0193959 +4 | -2.0193956 +4 | 1968 | 9.9 -7 | 7.5 -7 | 21 |
| be120.8.10 | 121;7381\|7502 | -2.0024004 +4 | -2.0024041 +4 | 2434 | 1.0 -6 | 9.8 -7 | 26 |
| be120.8.2 | 121;7381\|7502 | -2.0074132 +4 | -2.0074127 +4 | 3013 | 1.0 -6 | 9.4 -7 | 32 |
| be120.8.3 | 121;7381\|7502 | -2.0505900 +4 | -2.0505900 +4 | 2521 | 1.0 -6 | 8.5 -7 | 27 |
| be120.8.4 | 121;7381\|7502 | -2.1779801 +4 | -2.1779786 +4 | 2300 | 1.0 -6 | 9.2 -7 | 25 |
| be120.8.5 | 121;7381\|7502 | -2.1316282 +4 | -2.1316282 +4 | 2541 | 1.0 -6 | 6.1 -7 | 27 |
| be120.8.6 | 121;7381\|7502 | -1.9676959 +4 | -1.9676955 +4 | 2256 | 9.9 -7 | 7.4 -7 | 24 |
| be120.8.7 | 121;7381\|7502 | -2.3732388 +4 | -2.3732401 +4 | 1914 | 9.9 -7 | 7.4 -7 | 20 |
| be120.8.8 | 121;7381\|7502 | -2.1204746 +4 | -2.1204757 +4 | 1788 | 8.4 -7 | 9.7 -7 | 19 |
| be120.8.9 | 121;7381\|7502 | -1.9284427 +4 | -1.9284426 +4 | 1776 | 7.9 -7 | 1.0 -6 | 19 |
| be150.3.1 | 151;11476\|11627 | -1.9849179 +4 | -1.9849134 +4 | 2240 | 7.1 -7 | 1.0 -6 | 35 |
| be150.3.10 | 151;11476\|11627 | -1.9230920 +4 | -1.9230899 +4 | 3327 | 1.0 -6 | 6.8 -7 | 53 |
| be150.3.2 | 151;11476\|11627 | -1.8864852 +4 | -1.8864824 +4 | 2860 | 7.9 -7 | 1.0 -6 | 45 |
| be150.3.3 | 151;11476\|11627 | -1.8043715 +4 | -1.8043695 +4 | 2288 | 1.0 -6 | 7.9 -7 | 36 |
| be150.3.4 | 151;11476\|11627 | -2.0652667 +4 | -2.0652658 +4 | 2264 | 1.0 -6 | 7.1 -7 | 36 |
| be150.3.5 | 151;11476\|11627 | -1.7768649 +4 | -1.7768626 +4 | 2797 | 1.0 -6 | 8.3 -7 | 44 |
| be150.3.7 | 151;11476\|11627 | -1.9101308 +4 | -1.9101291 +4 | 2268 | 7.2 -7 | 1.0 -6 | 36 |
| be150.3.8 | 151;11476\|11627 | -1.9698060 +4 | -1.9698030 +4 | 3325 | 6.6 -7 | 1.0 -6 | 53 |
| be150.3.9 | 151;11476\|11627 | -1.4103367 +4 | -1.4103365 +4 | 2207 | 9.5 -7 | 7.5 -7 | 35 |
| be150.8.1 | 151;11476\|11627 | -2.9143686 +4 | -2.9143682 +4 | 2004 | 1.0 -6 | 7.2 -7 | 32 |
| be150.8.10 | 151;11476\|11627 | -3.0047974 +4 | -3.0047971 +4 | 2400 | 1.0 -6 | 6.8 -7 | 38 |
| be150.8.2 | 151;11476\|11627 | -2.8821096 +4 | -2.8821093 +4 | 2098 | 1.0 -6 | 7.2 -7 | 34 |
| be150.8.3 | 151;11476\|11627 | -3.1060369 +4 | -3.1060374 +4 | 2243 | 1.0 -6 | 8.5 -7 | 36 |
| be150.8.4 | 151;11476\|11627 | -2.8729295 +4 | -2.8729280 +4 | 2500 | 1.0 -6 | 7.4 -7 | 40 |
| be150.8.5 | 151;11476\|11627 | -2.9482071 +4 | -2.9482048 +4 | 2206 | 7.9 -7 | 9.9 -7 | 35 |
| be150.8.6 | 151;11476\|11627 | -3.1437234 +4 | -3.1437227 +4 | 2503 | 1.0 -6 | 7.4 -7 | 40 |
| be150.8.7 | 151;11476\|11627 | -3.3252109 +4 | -3.3252049 +4 | 2827 | 8.7 -7 | 1.0 -6 | 45 |
| be150.8.8 | 151;11476\|11627 | -3.1599994 +4 | -3.1599918 +4 | 3311 | 9.4 -7 | 9.8 -7 | 52 |
| be150.8.9 | 151;11476\|11627 | -2.7110727 +4 | -2.7110667 +4 | 2897 | 1.0 -6 | 6.9 -7 | 46 |
| be200.3.1 | 201;20301\|20502 | -2.7716091 +4 | -2.7716020 +4 | 2459 | 1.0 -6 | 9.2 -7 | 67 |
| be200.3.10 | 201;20301\|20502 | -2.5760689 +4 | -2.5760654 +4 | 3443 | 1.0 -6 | 8.4 -7 | 92 |
| be200.3.2 | 201;20301\|20502 | -2.6760792 +4 | -2.6760748 +4 | 2685 | 9.0 -7 | 1.0 -6 | 73 |
| be200.3.3 | 201;20301\|20502 | -2.9478639 +4 | -2.9478586 +4 | 5007 | 1.0 -6 | 8.8 -7 | 138 |
| be200.3.4 | 201;20301\|20502 | -2.9106208 +4 | -2.9106169 +4 | 3204 | 1.0 -6 | 7.6 -7 | 87 |
| be200.3.5 | 201;20301\|20502 | -2.8072990 +4 | -2.8072942 +4 | 4143 | 9.7 -7 | 1.0 -6 | 111 |
| be200.3.6 | 201;20301\|20502 | -2.7928345 +4 | -2.7928321 +4 | 2529 | 1.0 -6 | 8.3 -7 | 68 |
| be200.3.7 | 201;20301\|20502 | -3.1620504 +4 | -3.1620469 +4 | 3093 | 8.4 -7 | 1.0 -6 | 84 |
| be200.3.8 | 201;20301\|20502 | -2.9244286 +4 | -2.9244248 +4 | 2825 | 1.0 -6 | 6.7 -7 | 76 |
| be200.3.9 | 201;20301\|20502 | -2.6437048 +4 | -2.6436998 +4 | 5932 | 1.0 -6 | 7.0 -7 | 161 |
| be200.8.1 | 201;20301\|20502 | -5.0869496 +4 | -5.0869383 +4 | 4314 | 8.9 -7 | 1.0 -6 | 117 |
| be200.8.10 | 201;20301\|20502 | -4.5743067 +4 | -4.5742974 +4 | 2849 | 1.0 -6 | 8.3 -7 | 77 |
| be200.8.2 | 201;20301\|20502 | -4.4336053 +4 | -4.4335972 +4 | 2267 | 9.5 -7 | 1.0 -6 | 61 |
| be200.8.3 | 201;20301\|20502 | -4.6253974 +4 | -4.6253907 +4 | 3507 | 1.0 -6 | 8.1 -7 | 94 |
| be200.8.4 | 201;20301\|20502 | -4.6621241 +4 | -4.6621195 +4 | 2409 | 1.0 -6 | 9.2 -7 | 66 |
| be200.8.5 | 201;20301\|20502 | -4.4271235 +4 | -4.4271203 +4 | 2955 | 1.0 -6 | 9.5 -7 | 81 |
| be200.8.6 | 201;20301\|20502 | -5.1218884 +4 | -5.1218773 +4 | 3448 | 1.0 -6 | 6.7 -7 | 94 |
| be200.8.7 | 201;20301\|20502 | -4.9352768 +4 | -4.9352679 +4 | 2581 | 1.0 -6 | 8.2 -7 | 69 |
| be200.8.8 | 201;20301\|20502 | -4.7689168 +4 | -4.7689152 +4 | 2871 | 1.0 -6 | 7.1 -7 | 78 |
| be200.8.9 | 201;20301\|20502 | -4.5495602 +4 | -4.5495544 +4 | 3203 | 1.0 -6 | 6.8 -7 | 87 |
| be250.1 | 251;31626\|31877 | -2.5119464 +4 | -2.5119438 +4 | 5051 | 1.0 -6 | 6.9 -7 | 209 |
| be250.10 | 251;31626\|31877 | -2.4355024 +4 | -2.4354984 +4 | 5625 | 1.0 -6 | 8.0 -7 | 230 |
| be250.2 | 251;31626\|31877 | -2.3681493 +4 | -2.3681451 +4 | 4519 | 1.0 -6 | 7.3 -7 | 183 |
| be250.3 | 251;31626\|31877 | -2.4000002 +4 | -2.3999963 +4 | 3908 | 9.3 -7 | 9.9 -7 | 161 |
| be250.4 | 251;31626\|31877 | -2.5720317 +4 | -2.5720252 +4 | 8912 | 1.0 -6 | 7.4 -7 | 362 |
| be250.5 | 251;31626\|31877 | -2.2374710 +4 | -2.2374661 +4 | 4736 | 7.9 -7 | 1.0 -6 | 193 |
| be250.6 | 251;31626\|31877 | -2.4018844 +4 | -2.4018822 +4 | 4561 | 1.0 -6 | 6.7 -7 | 188 |
| be250.7 | 251;31626\|31877 | -2.5118959 +4 | -2.5118946 +4 | 5022 | 1.0 -6 | 7.5 -7 | 207 |
| be250.8 | 251;31626\|31877 | -2.5020398 +4 | -2.5020366 +4 | 4358 | 1.0 -6 | 6.8 -7 | 176 |
| be250.9 | 251;31626\|31877 | -2.1397059 +4 | -2.1396994 +4 | 5774 | 9.3 -7 | 1.0 -6 | 239 |
| bqp100-1 | 101;5151\|5252 | -8.3803844 +3 | -8.3803874 +3 | 1951 | 9.9 -7 | 9.4 -7 | 15 |
| bqp100-10 | 101;5151\|5252 | -1.2980272 +4 | -1.2980253 +4 | 3594 | 1.0 -6 | 7.0 -7 | 27 |
| bqp100-2 | 101;5151\|5252 | -1.1489258 +4 | -1.1489238 +4 | 3706 | 1.0 -6 | 8.2 -7 | 28 |
| bqp100-3 | 101;5151\|5252 | -1.3153184 +4 | -1.3153176 +4 | 5757 | 3.9 -7 | 1.0 -6 | 43 |
| bqp100-4 | 101;5151\|5252 | -1.0731890 +4 | -1.0731883 +4 | 3279 | 1.0 -6 | 9.7 -7 | 25 |
| bqp100-5 | 101;5151\|5252 | -9.4870275 +3 | -9.4870180 +3 | 3452 | 1.0 -6 | 8.3 -7 | 26 |
| bqp100-6 | 101;5151\|5252 | -1.0824760 +4 | -1.0824748 +4 | 2176 | 1.0 -6 | 7.8 -7 | 16 |
| bqp100-7 | 101;5151\|5252 | -1.0689155 +4 | -1.0689137 +4 | 1817 | 1.0 -6 | 9.1 -7 | 14 |
| bqp100-8 | 101;5151\|5252 | -1.1769990 +4 | -1.1769980 +4 | 3311 | 1.0 -6 | 7.0 -7 | 25 |

Table 8: DSA-BD results on BIQ problems `tab:BIQ-data2`

| Instance | $n_s;n_l\|m$ | $\langle C,X\rangle$ | $b^Ty$ | Iter | $\epsilon_P$ | $\epsilon_D$ | Time |
|---|---|---|---|---|---|---|---|
| bqp100-9 | 101;5151\|5252 | -1.1733253 +4 | -1.1733258 +4 | 4696 | 2.9 -7 | 1.0 -6 | 35 |
| bqp250-1 | 251;31626\|31877 | -4.7663112 +4 | -4.7662955 +4 | 4527 | 8.8 -7 | 1.0 -6 | 182 |
| bqp250-10 | 251;31626\|31877 | -4.3014529 +4 | -4.3014392 +4 | 3154 | 1.0 -6 | 9.9 -7 | 131 |
| bqp250-2 | 251;31626\|31877 | -4.7222380 +4 | -4.7222306 +4 | 4677 | 1.0 -6 | 7.2 -7 | 191 |
| bqp250-3 | 251;31626\|31877 | -5.1076751 +4 | -5.1076552 +4 | 5100 | 5.3 -7 | 1.0 -6 | 209 |
| bqp250-4 | 251;31626\|31877 | -4.3312555 +4 | -4.3312472 +4 | 3666 | 8.9 -7 | 1.0 -6 | 151 |
| bqp250-5 | 251;31626\|31877 | -5.0004327 +4 | -5.0004210 +4 | 6642 | 9.6 -7 | 1.0 -6 | 272 |
| bqp250-6 | 251;31626\|31877 | -4.3668862 +4 | -4.3668712 +4 | 5322 | 8.2 -7 | 1.0 -6 | 221 |
| bqp250-7 | 251;31626\|31877 | -4.8921743 +4 | -4.8921564 +4 | 4765 | 1.0 -6 | 8.6 -7 | 193 |
| bqp250-8 | 251;31626\|31877 | -3.8779549 +4 | -3.8779496 +4 | 3473 | 6.8 -7 | 1.0 -6 | 144 |
| bqp250-9 | 251;31626\|31877 | -5.1497554 +4 | -5.1497497 +4 | 4650 | 1.0 -6 | 9.0 -7 | 195 |
| bqp50-1 | 51;1326\|1377 | -2.1439177 +3 | -2.1439146 +3 | 2795 | 1.0 -6 | 6.9 -7 | 9 |
| bqp50-10 | 51;1326\|1377 | -3.6263711 +3 | -3.6263710 +3 | 1938 | 9.1 -7 | 9.5 -7 | 6 |
| bqp50-2 | 51;1326\|1377 | -3.7425229 +3 | -3.7425060 +3 | 6076 | 9.9 -7 | 9.2 -7 | 20 |
| bqp50-3 | 51;1326\|1377 | -4.6372395 +3 | -4.6372264 +3 | 2220 | 4.7 -7 | 1.0 -6 | 7 |
| bqp50-4 | 51;1326\|1377 | -3.5839746 +3 | -3.5839694 +3 | 12719 | 9.7 -7 | 1.0 -6 | 42 |
| bqp50-5 | 51;1326\|1377 | -4.0776076 +3 | -4.0776061 +3 | 2716 | 1.0 -6 | 8.7 -7 | 9 |
| bqp50-6 | 51;1326\|1377 | -3.7125866 +3 | -3.6959056 +3 | 20000 | 6.4 -4 | 1.4 -3 | 69 |
| bqp50-7 | 51;1326\|1377 | -4.6496912 +3 | -4.6496901 +3 | 2218 | 4.3 -7 | 1.0 -6 | 7 |
| bqp50-8 | 51;1326\|1377 | -4.2692350 +3 | -4.2692377 +3 | 2073 | 9.6 -7 | 9.6 -7 | 7 |
| bqp50-9 | 51;1326\|1377 | -3.9216432 +3 | -3.9216406 +3 | 1792 | 1.0 -6 | 9.9 -7 | 6 |
| bqp500-1 | 501;125751\|126252 | -1.2596423 +5 | -1.2596374 +5 | 7353 | 7.0 -7 | 1.0 -6 | 1230 |
| bqp500-10 | 501;125751\|126252 | -1.3853448 +5 | -1.3853384 +5 | 7411 | 7.9 -7 | 1.0 -6 | 1241 |
| bqp500-2 | 501;125751\|126252 | -1.3601108 +5 | -1.3601080 +5 | 8252 | 1.0 -6 | 7.5 -7 | 1364 |
| bqp500-3 | 501;125751\|126252 | -1.3845346 +5 | -1.3845287 +5 | 7329 | 7.7 -7 | 1.0 -6 | 1232 |
| bqp500-4 | 501;125751\|126252 | -1.3932842 +5 | -1.3932790 +5 | 7910 | 6.8 -7 | 1.0 -6 | 1303 |
| bqp500-5 | 501;125751\|126252 | -1.3409217 +5 | -1.3409177 +5 | 7790 | 1.0 -6 | 8.2 -7 | 1294 |
| bqp500-6 | 501;125751\|126252 | -1.3076439 +5 | -1.3076411 +5 | 7363 | 7.3 -7 | 1.0 -6 | 1236 |
| bqp500-7 | 501;125751\|126252 | -1.3149149 +5 | -1.3149108 +5 | 7621 | 6.9 -7 | 1.0 -6 | 1280 |
| bqp500-8 | 501;125751\|126252 | -1.3348988 +5 | -1.3348960 +5 | 7911 | 1.0 -6 | 9.8 -7 | 1314 |
| bqp500-9 | 501;125751\|126252 | -1.3028828 +5 | -1.3028790 +5 | 7166 | 1.0 -6 | 9.5 -7 | 1187 |
| gka10b | 126;8001\|8127 | -1.5557650 +2 | -1.5555950 +2 | 1822 | 6.8 -7 | 1.0 -6 | 22 |
| gka10d | 101;5151\|5252 | -2.0108576 +4 | -2.0108575 +4 | 2008 | 1.0 -6 | 8.9 -7 | 17 |
| gka1a | 51;1326\|1377 | -3.5374674 +3 | -3.5366291 +3 | 20000 | 9.4 -6 | 3.4 -5 | 74 |
| gka1b | 21;231\|252 | -1.3300000 +2 | -1.3300000 +2 | 1130 | 8.9 -7 | 9.2 -7 | 3 |
| gka1c | 41;861\|902 | -5.1138290 +3 | -5.1138227 +3 | 3866 | 5.4 -7 | 1.0 -6 | 11 |
| gka1d | 101;5151\|5252 | -6.5284315 +3 | -6.5283639 +3 | 3937 | 1.0 -6 | 9.9 -7 | 33 |
| gka1e | 201;20301\|20502 | -1.7069817 +4 | -1.7069805 +4 | 4971 | 1.0 -6 | 7.2 -7 | 181 |
| gka1f | 501;125751\|126252 | -6.5559070 +4 | -6.5558956 +4 | 7631 | 1.0 -6 | 7.7 -7 | 2195 |
| gka2b | 31;496\|527 | -1.2130600 +2 | -1.2129990 +2 | 2220 | 1.0 -6 | 7.3 -7 | 6 |
| gka2c | 51;1326\|1377 | -6.3200104 +3 | -6.3199986 +3 | 2136 | 5.2 -7 | 9.9 -7 | 7 |
| gka2d | 101;5151\|5252 | -6.9907097 +3 | -6.9907100 +3 | 2042 | 7.1 -7 | 1.0 -6 | 17 |
| gka2e | 201;20301\|20502 | -2.4917636 +4 | -2.4917596 +4 | 3231 | 1.0 -6 | 7.4 -7 | 121 |
| gka2f | 501;125751\|126252 | -1.0793177 +5 | -1.0793138 +5 | 8508 | 1.0 -6 | 9.4 -7 | 2488 |
| gka3a | 71;2556\|2627 | -6.3859990 +3 | -6.3859859 +3 | 2202 | 6.4 -7 | 9.9 -7 | 11 |
| gka3b | 41;861\|902 | -1.1799980 +2 | -1.1801510 +2 | 6358 | 9.7 -7 | 9.5 -7 | 17 |
| gka3c | 61;1891\|1952 | -6.8138990 +3 | -6.8138886 +3 | 1988 | 3.5 -7 | 9.9 -7 | 8 |
| gka3d | 101;5151\|5252 | -9.7343322 +3 | -9.7343347 +3 | 3600 | 1.0 -6 | 8.7 -7 | 30 |
| gka3e | 201;20301\|20502 | -2.6898741 +4 | -2.6898705 +4 | 4302 | 1.0 -6 | 6.8 -7 | 161 |
| gka3f | 501;125751\|126252 | -1.5015105 +5 | -1.5015061 +5 | 7522 | 9.5 -7 | 1.0 -6 | 2222 |
| gka4a | 81;3321\|3402 | -8.8809678 +3 | -8.8809583 +3 | 2936 | 2.8 -7 | 1.0 -6 | 19 |
| gka4b | 51;1326\|1377 | -1.2899980 +2 | -1.2902000 +2 | 6382 | 9.9 -7 | 9.8 -7 | 21 |
| gka4c | 71;2556\|2627 | -7.5650115 +3 | -7.5650110 +3 | 2735 | 8.6 -7 | 1.0 -6 | 14 |
| gka4d | 101;5151\|5252 | -1.1278414 +4 | -1.1278415 +4 | 2279 | 1.0 -6 | 7.6 -7 | 19 |
| gka4e | 201;20301\|20502 | -3.7225147 +4 | -3.7225072 +4 | 4524 | 9.4 -7 | 1.0 -6 | 187 |
| gka4f | 501;125751\|126252 | -1.8708790 +5 | -1.8708748 +5 | 7948 | 1.0 -6 | 7.9 -7 | 2348 |
| gka5a | 51;1326\|1377 | -5.8970505 +3 | -5.8970492 +3 | 2050 | 3.2 -7 | 9.9 -7 | 7 |
| gka5b | 61;1891\|1952 | -1.4999980 +2 | -1.5002340 +2 | 6393 | 9.8 -7 | 9.9 -7 | 25 |
| gka5c | 81;3321\|3402 | -7.5762319 +3 | -7.5762289 +3 | 3702 | 1.0 -6 | 9.7 -7 | 22 |
| gka5d | 101;5151\|5252 | -1.2398864 +4 | -1.2398866 +4 | 2234 | 1.0 -6 | 6.8 -7 | 19 |
| gka5e | 201;20301\|20502 | -3.8002313 +4 | -3.8002274 +4 | 3945 | 1.0 -6 | 8.5 -7 | 149 |
| gka5f | 501;125751\|126252 | -2.0691429 +5 | -2.0691388 +5 | 7445 | 7.0 -7 | 1.0 -6 | 2210 |
| gka6a | 31;496\|527 | -4.1032065 +3 | -4.1032066 +3 | 1951 | 1.8 -7 | 9.9 -7 | 5 |
| gka6b | 71;2556\|2627 | -1.4600020 +2 | -1.4597220 +2 | 5673 | 9.9 -7 | 9.7 -7 | 27 |
| gka6c | 91;4186\|4277 | -5.9619429 +3 | -5.9619530 +3 | 4360 | 1.0 -6 | 8.9 -7 | 31 |
| gka6d | 101;5151\|5252 | -1.4929358 +4 | -1.4929358 +4 | 2131 | 7.3 -7 | 9.9 -7 | 18 |
| gka7a | 31;496\|527 | -4.6386078 +3 | -4.6386032 +3 | 2134 | 1.0 -6 | 6.4 -7 | 5 |
| gka7b | 81;3321\|3402 | -1.6035690 +2 | -1.6035880 +2 | 493 | 9.2 -7 | 9.9 -7 | 3 |
| gka7c | 101;5151\|5252 | -7.3164493 +3 | -7.3164551 +3 | 3144 | 9.0 -7 | 1.0 -6 | 26 |
| gka7d | 101;5151\|5252 | -1.5375790 +4 | -1.5375801 +4 | 1850 | 9.5 -7 | 1.0 -6 | 16 |
| gka8a | 101;5151\|5252 | -1.1197217 +4 | -1.1197215 +4 | 3678 | 7.2 -7 | 1.0 -6 | 33 |
| gka8b | 91;4186\|4277 | -1.4499980 +2 | -1.4503580 +2 | 6797 | 9.9 -7 | 9.7 -7 | 49 |
| gka8d | 101;5151\|5252 | -1.7005361 +4 | -1.7005352 +4 | 3643 | 6.9 -7 | 1.0 -6 | 31 |
| gka9b | 101;5151\|5252 | -1.3700010 +2 | -1.3696100 +2 | 6155 | 9.5 -7 | 9.4 -7 | 46 |
| gka9d | 101;5151\|5252 | -1.6533903 +4 | -1.6533898 +4 | 1776 | 9.6 -7 | 1.0 -6 | 15 |

### 5.4 Quadratic assignment problems

This subsection compares the performance of our method DSA-BD with that of SDPNAL on a collection of SDP relaxations of QAPs.

Given the set $\Pi$ of $n \times n$ permutation matrices and $A, B \in \mathbb{R}^{n\times n}$, the quadratic assignment problem can be formulated as

$$\min \{\langle X, AXB\rangle : X \in \Pi\}.$$

For a matrix $X = [x_1, \ldots, x_n] \in \mathbb{R}^{n\times n}$, we will identify it with the $n^2$-vector $x = (x_1; \ldots; x_n)$. For a matrix $Y \in \mathbb{R}^{n^2\times n^2}$, we let $Y^{ij}$ be the $n \times n$ block corresponding to $x_i x_j^T$ in the matrix $xx^T$. In [13], it is shown that an SDP relaxation of the QAP is

$$\begin{array}{ll} \max & \langle B \otimes A, Y\rangle \\ \text{s.t.} & \sum_{i=1}^{n} Y^{ii} = I,\ \langle I, Y^{ij}\rangle = \delta_{ij}\ \forall\ 1 \le i \le j \le n, \\ & \langle E, Y^{ij}\rangle = 1\ ,\forall 1 \le i \le j \le n, \\ & Y \succeq 0,\ Y \ge 0, \end{array}$$

where $E \in \mathbb{R}^{n\times n}$ is the matrix of ones, and $\delta_{ij} = 1$ if $i = j$, and 0 otherwise.

Table 9 compares the two methods on a collection of SDP relaxations of QAPs using the tolerances $\bar{\epsilon} = 10^{-4}, 10^{-5}$. In this table, computational results for each instance are reported in two rows, the first one for $\bar{\epsilon} = 10^{-4}$, and the second one for $\bar{\epsilon} = 10^{-5}$. Table 10 gives more detailed computational results on these instances obtained by our method DSA-BD, such as the objective values, number of iterations, and the primal and dual relative residuals. Figure 5 plots the performance profiles of both methods based on these SDP relaxations of QAPs.

Note that SDPNAL is outperformed by our method and fails to converge to an accuracy less than or equal to $10^{-4}$ in almost all of the SDP relaxations of QAPs tested. On the other hand, our method DSA-BD converges to an accuracy less than $10^{-5}$ in almost all of the SDP relaxations of QAPs tested.

## References


BIS97 [1] R. S. Burachik, A. N. Iusem, and B. F. Svaiter. Enlargement of monotone operators with applications to variational inequalities. *Set-Valued Analysis*, 5:159–180, 1997. 10.1023/A:1008615624787.

BS02 [2] R. S. Burachik and B. F. Svaiter. Maximal monotone operators, convex functions and a special family of enlargements. *Set-Valued Analysis*, 10:297–316, 2002. 10.1023/A:1020639314056.

BMZ03 [3] S. Burer, R. D.C. Monteiro, and Y. Zhang. A computational study of a gradient-based log-barrier algorithm for a class of large-scale SDPs. *Mathematical Programming*, 95:359–379, 2003. 10.1007/s10107-002-0353-7.

CP10 [4] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems withÂ applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145, 2011. 10.1007/s10851-010-0251-1.

DM02 [5] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles, January 2002.

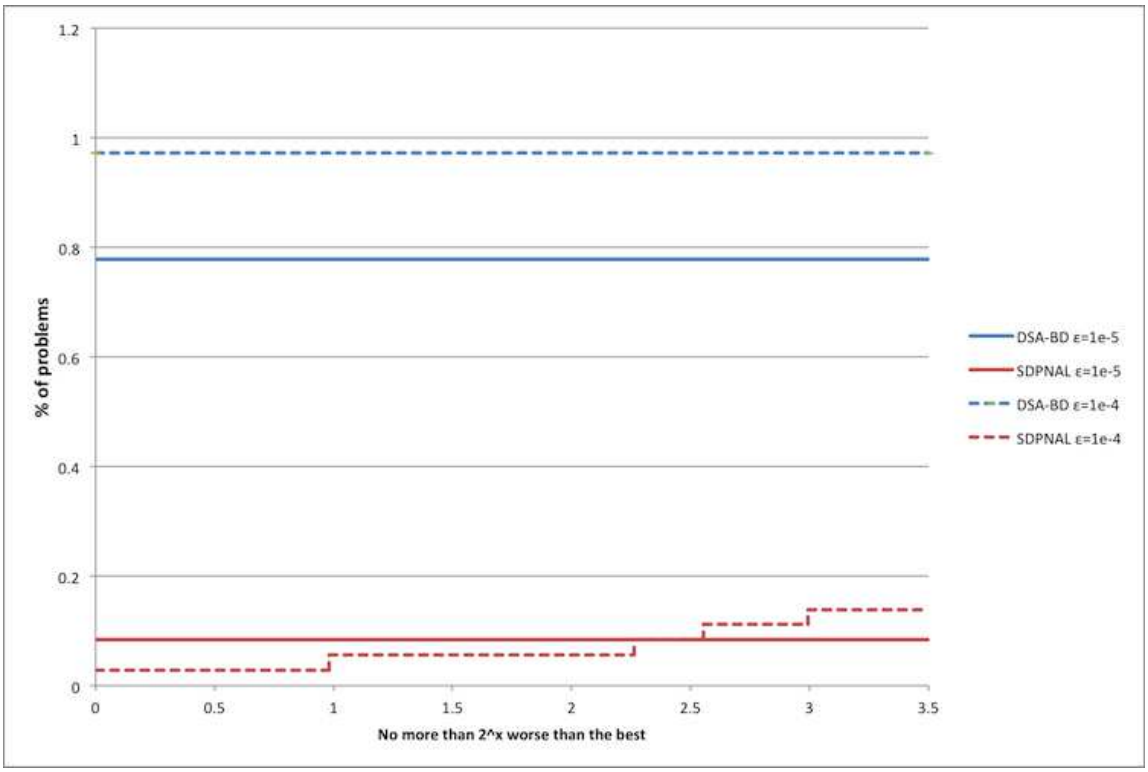
GM76 [6] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers Mathematics with Applications*, 2(1):17 – 40, 1976.

Table 9: Comparison of the methods on QAPs tab:QAP

| Problem | | $\max\{\epsilon_P, \epsilon_D\}$ | | Time | |
|---|---|---|---|---|---|
| Instance | $n_S; n_l \vert m$ | DSA-BD | SDPNAL | DSA-BD | SDPNAL |
| bur26a | 676;228826\|229877 | 1.0 -4 | 1.4 -4* | 453 | 4669* |
| | | 1.0 -5 | 1.4 -4* | 13845 | 4669* |
| bur26b | 676;228826\|229877 | 1.0 -4 | 1.5 -4* | 600 | 3577* |
| | | 1.0 -5 | 1.5 -4* | 13330 | 3577* |
| bur26c | 676;228826\|229877 | 1.0 -4 | 2.4 -4* | 641 | 5790* |
| | | 1.0 -5 | 2.4 -4* | 14684 | 5790* |
| bur26d | 676;228826\|229877 | 1.0 -4 | 2.5 -4* | 735 | 4436* |
| | | 1.0 -5 | 2.5 -4* | 16765 | 4436* |
| bur26e | 676;228826\|229877 | 1.0 -4 | 2.3 -4* | 750 | 6240* |
| | | 1.0 -5 | 2.3 -4* | 5168 | 6240* |
| bur26f | 676;228826\|229877 | 1.0 -4 | 2.1 -4* | 856 | 4509* |
| | | 1.0 -5 | 2.1 -4* | 10088 | 4509* |
| bur26g | 676;228826\|229877 | 1.0 -4 | 1.9 -4* | 311 | 4115* |
| | | 1.0 -5 | 1.9 -4* | 6814 | 4115* |
| bur26h | 676;228826\|229877 | 9.9 -5 | 3.4 -5 | 195 | 1552 |
| | | 1.0 -5 | 2.5 -5* | 1284 | 4419* |
| chr12a | 144;10440\|10672 | 1.1 -4* | 8.0 -5 | 2664* | 106 |
| | | 1.1 -4* | 4.8 -6 | 2664* | 110 |
| chr12b | 144;10440\|10672 | 8.9 -5 | 1.6 -5 | 27 | 132 |
| | | 4.3 -5* | 9.2 -6 | 2762* | 144 |
| chr12c | 144;10440\|10672 | 9.9 -5 | 1.5 -4* | 42 | 284* |
| | | 9.7 -6 | 1.5 -4* | 769 | 284* |
| chr15a | 225;25425\|25783 | 1.0 -4 | 2.8 -4* | 252 | 656* |
| | | 1.0 -5 | 2.8 -4* | 1947 | 656* |
| chr15b | 225;25425\|25783 | 1.0 -4 | 4.4 -4* | 129 | 325* |
| | | 2.4 -5* | 4.4 -4* | 5768* | 325* |
| chr15c | 225;25425\|25783 | 1.0 -4 | 4.7 -7 | 168 | 331 |
| | | 6.5 -5* | 4.7 -7 | 5534* | 331 |
| chr18a | 324;52650\|53161 | 9.9 -5 | 7.0 -4* | 477 | 1475* |
| | | 1.0 -5 | 7.0 -4* | 3488 | 1475* |
| chr18b | 324;52650\|53161 | 1.0 -4 | 4.0 -5 | 54 | 314 |
| | | 1.0 -5 | 4.0 -5* | 101 | 797* |
| chr20a | 400;80200\|80828 | 1.0 -4 | 4.1 -4* | 522 | 2133* |
| | | 1.0 -5 | 4.1 -4* | 1109 | 2133* |
| chr20b | 400;80200\|80828 | 1.0 -4 | 1.4 -4* | 941 | 1736* |
| | | 2.6 -5* | 1.4 -4* | 17235* | 1736* |
| chr20c | 400;80200\|80828 | 1.0 -4 | 4.3 -4* | 650 | 1622* |
| | | 9.8 -6 | 4.3 -4* | 5365 | 1622* |
| chr22a | 484;117370\|118127 | 1.0 -4 | 3.1 -4* | 302 | 2911* |
| | | 1.2 -5* | 3.1 -4* | 25802* | 2911* |
| chr22b | 484;117370\|118127 | 1.0 -4 | 2.2 -4* | 308 | 2273* |
| | | 1.3 -5* | 2.2 -4* | 24400* | 2273* |
| nug12 | 144;10440\|10672 | 1.0 -4 | 1.6 -4* | 31 | 46* |
| | | 1.0 -5 | 1.6 -4* | 239 | 46* |
| nug14 | 196;19306\|19619 | 1.0 -4 | 3.1 -4* | 91 | 119* |
| | | 1.0 -5 | 3.1 -4* | 1144 | 119* |
| nug15 | 225;25425\|25783 | 1.0 -4 | 2.2 -4* | 108 | 159* |
| | | 1.0 -5 | 2.2 -4* | 1019 | 159* |
| nug16a | 256;32896\|33302 | 1.0 -4 | 1.8 -4* | 203 | 336* |
| | | 1.0 -5 | 1.8 -4* | 2514 | 336* |
| nug16b | 256;32896\|33302 | 1.0 -4 | 2.7 -4* | 103 | 194* |
| | | 1.0 -5 | 2.7 -4* | 914 | 194* |
| nug17 | 289;41905\|42362 | 1.0 -4 | 1.6 -4* | 185 | 272* |
| | | 1.0 -5 | 1.6 -4* | 1742 | 272* |
| nug18 | 324;52650\|53161 | 1.0 -4 | 2.2 -4* | 206 | 297* |
| | | 1.0 -5 | 2.2 -4* | 1827 | 297* |
| nug20 | 400;80200\|80828 | 1.0 -4 | 1.8 -4* | 299 | 471* |
| | | 1.0 -5 | 1.8 -4* | 2466 | 471* |
| nug21 | 441;97461\|98152 | 1.0 -4 | 2.2 -4* | 420 | 665* |
| | | 1.0 -5 | 2.2 -4* | 3780 | 665* |
| nug21 | 441;97461\|98152 | 1.0 -4 | 2.2 -4* | 418 | 663* |
| | | 1.0 -5 | 2.2 -4* | 3814 | 663* |
| nug22 | 484;117370\|118127 | 1.0 -4 | 2.3 -4* | 659 | 961* |
| | | 1.0 -5 | 2.3 -4* | 5562 | 961* |
| nug22 | 484;117370\|118127 | 1.0 -4 | 2.3 -4* | 651 | 1030* |
| | | 1.0 -5 | 2.3 -4* | 5473 | 1030* |
| tai25a | 625;195625\|196598 | 1.0 -4 | 1.4 -4* | 471 | 1035* |
| | | 1.0 -5 | 1.4 -4* | 3688 | 1035* |
| tai25b | 625;195625\|196598 | 1.0 -4 | 5.1 -4* | 1891 | 2438* |
| | | 1.2 -5* | 5.1 -4* | 28402* | 2438* |
| tai30a | 900;405450\|406843 | 1.0 -4 | 1.0 -4* | 1059 | 2452* |
| | | 1.0 -5 | 1.0 -4* | 8223 | 2452* |

Table 10: DSA-BD results on QAPs `tab:QAP-data`

| Instance | $n_s; n_l \mid m$ | $\langle C, X \rangle$ | $b^T y$ | Iter | $\epsilon_P$ | $\epsilon_D$ | Time |
|---|---|---|---|---|---|---|---|
| bur26a | 676;228826\|229877 | 5.4263208 +6 | 5.4268401 +6 | 50000 | 5.6 -6 | 7.0 -6 | 22578 |
| bur26b | 676;228826\|229877 | 3.8172495 +6 | 3.8176224 +6 | 50000 | 6.2 -6 | 6.1 -6 | 23085 |
| bur26c | 676;228826\|229877 | 5.4270012 +6 | 5.4277912 +6 | 50000 | 7.9 -6 | 9.4 -6 | 22290 |
| bur26d | 676;228826\|229877 | 3.8206639 +6 | 3.8211360 +6 | 50000 | 6.4 -6 | 8.2 -6 | 22612 |
| bur26e | 676;228826\|229877 | 5.3874085 +6 | 5.3877112 +6 | 50000 | 3.2 -6 | 4.3 -6 | 22577 |
| bur26f | 676;228826\|229877 | 3.7820836 +6 | 3.7821498 +6 | 47127 | 6.6 -7 | 1.0 -6 | 21173 |
| bur26g | 676;228826\|229877 | 1.0117250 +7 | 1.0117274 +7 | 32438 | 1.0 -6 | 3.9 -7 | 14530 |
| bur26h | 676;228826\|229877 | 7.0986748 +6 | 7.0987800 +6 | 18086 | 9.9 -7 | 7.3 -7 | 7645 |
| chr12a | 144;10440\|10672 | 9.5602473 +3 | 9.6751178 +3 | 100000 | 2.7 -5 | 1.1 -4 | 2664 |
| chr12b | 144;10440\|10672 | 9.7420318 +3 | 9.8474864 +3 | 100000 | 1.3 -5 | 4.3 -5 | 2762 |
| chr12c | 144;10440\|10672 | 1.1156025 +4 | 1.1163345 +4 | 100000 | 5.4 -7 | 3.2 -6 | 2578 |
| chr15a | 225;25425\|25783 | 9.8957979 +3 | 9.9101536 +3 | 100000 | 5.5 -6 | 4.4 -6 | 5152 |
| chr15b | 225;25425\|25783 | 7.9898869 +3 | 7.8800531 +3 | 100000 | 3.3 -6 | 2.4 -5 | 5768 |
| chr15c | 225;25425\|25783 | 9.5039347 +3 | 9.2142006 +3 | 100000 | 1.3 -5 | 6.5 -5 | 5534 |
| chr18a | 324;52650\|53161 | 1.1097999 +4 | 1.1085506 +4 | 100000 | 2.2 -7 | 1.8 -6 | 10592 |
| chr18b | 324;52650\|53161 | 1.5339652 +3 | 1.5340022 +3 | 1493 | 9.1 -7 | 1.0 -6 | 149 |
| chr20a | 400;80200\|80828 | 2.1919994 +3 | 2.1902299 +3 | 100000 | 4.0 -7 | 1.8 -6 | 16496 |
| chr20b | 400;80200\|80828 | 2.2980107 +3 | 2.2705528 +3 | 100000 | 3.9 -6 | 2.6 -5 | 17235 |
| chr20c | 400;80200\|80828 | 1.4144561 +4 | 1.4156713 +4 | 100000 | 6.2 -7 | 1.6 -6 | 16440 |
| chr22a | 484;117370\|118127 | 6.1565369 +3 | 6.1744853 +3 | 100000 | 6.1 -6 | 1.2 -5 | 25802 |
| chr22b | 484;117370\|118127 | 6.1946119 +3 | 6.2137604 +3 | 100000 | 6.1 -6 | 1.3 -5 | 24400 |
| nug12 | 144;10440\|10672 | 5.6778580 +2 | 5.6788770 +2 | 100000 | 1.5 -6 | 1.2 -6 | 1776 |
| nug14 | 196;19306\|19619 | 1.0095760 +3 | 1.0098620 +3 | 100000 | 4.2 -6 | 2.8 -6 | 2877 |
| nug15 | 225;25425\|25783 | 1.1399857 +3 | 1.1402761 +3 | 100000 | 2.8 -6 | 2.1 -6 | 3692 |
| nug16a | 256;32896\|33302 | 1.5983130 +3 | 1.5988500 +3 | 100000 | 5.5 -6 | 3.9 -6 | 4671 |
| nug16b | 256;32896\|33302 | 1.2176912 +3 | 1.2179803 +3 | 100000 | 2.1 -6 | 1.6 -6 | 4388 |
| nug17 | 289;41905\|42362 | 1.7062557 +3 | 1.7066925 +3 | 100000 | 3.0 -6 | 2.3 -6 | 5733 |
| nug18 | 324;52650\|53161 | 1.8927180 +3 | 1.8931364 +3 | 100000 | 2.6 -6 | 1.8 -6 | 7055 |
| nug20 | 400;80200\|80828 | 2.5054307 +3 | 2.5058998 +3 | 100000 | 2.1 -6 | 1.7 -6 | 11029 |
| nug21 | 441;97461\|98152 | 2.3808229 +3 | 2.3813933 +3 | 100000 | 2.7 -6 | 2.0 -6 | 13311 |
| nug21 | 441;97461\|98152 | 2.3808229 +3 | 2.3813933 +3 | 100000 | 2.7 -6 | 2.0 -6 | 13435 |
| nug22 | 484;117370\|118127 | 3.5266774 +3 | 3.5277129 +3 | 100000 | 3.4 -6 | 2.4 -6 | 16183 |
| nug22 | 484;117370\|118127 | 3.5266774 +3 | 3.5277129 +3 | 100000 | 3.4 -6 | 2.4 -6 | 15980 |
| tai25a | 625;195625\|196598 | 1.0964898 +6 | 1.0965735 +6 | 100000 | 1.3 -6 | 8.4 -7 | 28954 |
| tai25b | 625;195625\|196598 | 3.3674678 +8 | 3.3752145 +8 | 100000 | 1.0 -5 | 1.2 -5 | 28402 |
| tai30a | 900;405450\|406843 | 1.7066136 +6 | 1.7067425 +6 | 100000 | 1.2 -6 | 8.9 -7 | 63839 |




`Profile-6-QAP` Figure 5: Performance profiles of DSA-BD and SDPNAL for solving 36 SDP relaxations of QAPs with accuracies $\bar{\epsilon} = 10^{-4}, 10^{-5}$.

GM75 [7] R. Glowinski and A. Marrocco. Sur lÕapproximation, par éléments finis dÕordre un, et la résolution, par penalisation-dualité, dÕune classe de problèmes de dirichlet non linéaires. *RAIRO Anal. Numér.*, 2:41 – 76, 1975.

MPR09 [8] J. Malick, J. Povh, F. Rendl, and A. Wiegele. Regularization methods for semidefinite programming. *SIAM J. on Optimization*, 20(1):336–356, April 2009.

MOS12 [9] R. D. C. Monteiro, C. Ortiz, and B. F. Svaiter. A first-order block-decomposition method for solving two-easy-block structured semidefinite programs. Manuscript, School of ISyE, Georgia Tech, Atlanta, GA, 30332, USA, June 2012.

MS10b [10] R. D. C. Monteiro and B. F. Svaiter. Iteration-complexity of block-decomposition algorithms and the alternating minimization augmented Lagrangian method. *Optimization-online preprint 2713*, pages 1–28, 2010.

MS10c [11] R. D. C. Monteiro and B. F. Svaiter. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. *SIAM Journal on Optimization*, 20(6):2755–2787, 2010.

MS10 [12] R. D. C. Monteiro and B. F. Svaiter. 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 Journal on Optimization*, 21(4):1688–1720, 2011.

PR06 [13] J. Povh and F. Rendl. Copositive and semidefinite relaxations of the quadratic assignment problem. *Discrete Optimization*, 6(3):231 – 241, 2009.

PRW06 [14] J. Povh, F. Rendl, and A. Wiegele. A boundary point method to solve semidefinite programs. *Computing*, 78:277–286, 2006. 10.1007/s00607-006-0182-2.

R70 [15] R. T. Rockafellar. On the maximal monotonicity of subdifferential mappings. *Pacific J. Math.*, 33:209–216, 1970.

SS99 [16] M. V. Solodov and B. F. Svaiter. A hybrid approximate extragradient – proximal point algorithm using the enlargement of a maximal monotone operator. *SetValued Analysis*, 7(4):323–345, 1999.

SS99b [17] M. V. Solodov and B. F. Svaiter. A hybrid projection-proximal point algorithm. *Journal of Convex Analysis*, 6(1):59–70, 1999.

SVA00 [18] B. F. Svaiter. A family of enlargements of maximal monotone operators. *Set-Valued Analysis*, 8:311–328, 2000. 10.1023/A:1026555124541.

TTT99 [19] K. C. Toh, M.J. Todd, and R. H. Tütüncü. Sdpt3 - a matlab software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

WGY10 [20] Z. Wen, D. Goldfarb, and W. Yin. Alternating direction augmented lagrangian methods for semidefinite programming. *Mathematical Programming Computation*, 2:203–230, 2010. 10.1007/s12532-010-0017-1.

ZST10 [21] X.-Y. Zhao, D. Sun, and K.-C. Toh. A Newton-CG augmented lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20(4):1737–1765, 2010.