Department of Policy and Planning Sciences

Discussion Paper Series

No.1378

A New Extension of Chubanov’s Method to Symmetric Cones

by

Shin-ichi KANOH and Akiko YOSHISE

October 2021

UNIVERSITY OF TSUKUBA
Tsukuba, Ibaraki 305-8573
JAPAN

A New Extension of Chubanov's Method to Symmetric Cones

Shin-ichi Kanoh[*]and Akiko Yoshise[†]

October 2021

Abstract

We propose a new variant of Chubanov's method for solving the feasibility problem over the symmetric cone by extending Roos's method (2018) for the feasibility problem over the nonnegative orthant. The proposed method considers a feasibility problem associated with a norm induced by the maximum eigenvalue of an element and uses a rescaling focusing on the upper bound of the sum of eigenvalues of any feasible solution to the problem. Its computational bound is (i) equivalent to Roos's original method (2018) and superior to Lourenço et al.'s method (2019) when the symmetric cone is the nonnegative orthant, (ii) superior to Lourenço et al.'s method (2019) when the symmetric cone is a Cartesian product of second-order cones, and (iii) equivalent to Lourenço et al.'s method (2019) when the symmetric cone is the simple positive semidefinite cone, under the assumption that the costs of computing the spectral decomposition and the minimum eigenvalue are of the same order for any given symmetric matrix.

We also conduct numerical experiments that compare the performance of our method with existing methods by generating instance in three types: (i) strongly (but ill-conditioned) feasible instances, (ii) weakly feasible instances, and (iii) infeasible instances. For any of these instances, the proposed method is rather more efficient than the existing methods in terms of accuracy and execution time.

1 Introduction

Recently, Chubanov [3, 4] proposed a new polynomial-time algorithm for solving the problem (P(A)),

$$\begin{array}{lll} \mathrm{P}(A) & \text{find} & x \in \mathbb{R}^n \\ & \text{s.t.} & Ax = \mathbf{0}, \\ & & x > \mathbf{0}, \end{array}$$

where A is a given integer (or rational) matrix and $\mathrm{rank}(A) = m$ and $\mathbf{0}$ is the n-dimensional vector of 0s. The method explores the feasibility of the following problem $\mathrm{P}_{S_1}(A)$, which is equivalent to $\mathrm{P}(A)$ and given by

$$\begin{array}{lll} \mathrm{P}_{S_1}(A) & \text{find} & x \in \mathbb{R}^n \\ & \text{s.t.} & Ax = \mathbf{0}, \\ & & \mathbf{0} < x \leq \mathbf{1}, \end{array}$$

[*]Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305-8573, and Japan Society for the Promotion of Science, 5-3-1 Kojimachi, Chiyoda-ku, Tokyo 102-0083, Japan. email: s2130104@s.tsukuba.ac.jp

[†]Corresponding author. Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan. email: yoshise@sk.tsukuba.ac.jp

where $\mathbf{1}$ is the n-dimensional vector of 1s. Chubanov's method consists of two ingredients, the "main algorithm" and the "basic procedure." The structure of the method is as follows: In the outer iteration, the main algorithm calls the basic procedure, which generates a sequence in $\mathbb{R}^n$ using projection to the set Ker$A$. The basic procedure terminates in a finite number of iterations returning one of the following:

1. a solution of problem P(A), or
2. a solution of the alternative problem of problem P(A), or
3. a cut of P(A), i.e., an index $j \in \{1, 2, \ldots, n\}$ for which $0 < x_j \leq \frac{1}{2}$ holds for any feasible solution of problem $\mathrm{P}_{S_1}(A)$.

If result 1 or 2 is returned by the basic procedure, then the feasibility of the problem P(A) can be determined, and the main procedure stops. If result 3 is returned, then the main procedure generates a diagonal matrix $D \in \mathbb{R}^{n\times n}$ with a (j, j) element of 2 and all other diagonal elements of 1 and rescales the matrix as AD^{-1}. Then it calls the basic procedure with the rescaled matrix. Chubanov's method checks the feasibility of P(A) by repeating the above procedures.

For problem P(A), several variations of Chubanov's method have been proposed and computational experiments have been conducted [9, 13, 16]. Among these, [13] proposed a tighter cut criterion of the basic procedure than the criterion used in [4]. [4] used the fact that

$$x_j \leq \frac{\sqrt{n}\|z\|_2}{y_j} \tag{1}$$

holds for any $y \in \mathbb{R}^n$ satisfying $\sum_{i=1}^n y_i = 1, y \geq 0$ and $y \notin \mathrm{Im}A^T$, $z \in \mathbb{R}^n$ obtained by projecting this y onto Ker$A$, and any feasible solution $x \in \mathbb{R}^n$ of $\mathrm{P}_{S_1}(A)$, and the basic procedure is terminated if a y is found for which $\frac{\sqrt{n}\|z\|_2}{y_j} \leq \frac{1}{2}$ holds for some index j. On the other hand, [13] showed that for $v = y - z$,

$$x_j \leq \min\left(1, \mathbf{1}^T\left[\frac{-v}{v_j}\right]^+\right) \leq \frac{\sqrt{n}\|z\|_2}{y_j} \tag{2}$$

holds if $v_j \neq 0$, where $\left[\frac{-v}{v_j}\right]^+$ is the projection of $\frac{-v}{v_j} \in \mathbb{R}^n$ on to the nonnegative orthant and $\mathbf{1}$ is the vector of ones, and the basic procedure is terminated if a y is found for which $\mathbf{1}^T\left[\frac{-v}{v_j}\right]^+ \leq \frac{1}{2}$ holds.

Chubanov's method has also been extended to the feasibility problem over the second-order cone [8] and the symmetric cone [11, 2]. The feasibility problem over the symmetric cone is of the form,

$$\begin{array}{lll} \mathrm{P}(\mathcal{A}) & \text{find} & x \\ & \text{s.t.} & \mathcal{A}(x) = \mathbf{0}, \\ & & x \in \mathrm{int}\mathcal{K}, \end{array}$$

where $\mathcal{A}$ is a linear operator, $\mathcal{K}$ is a symmetric cone, and int$\mathcal{K}$ is the interior of the set $\mathcal{K}$.

As proposed in [11, 2], for problem P($\mathcal{A}$), the structure of Chubanov's method remains the same; i.e., the main algorithm calls the basic procedure, and the basic procedure returns one of the following in a finite number of iterations:

1. a solution of problem P($\mathcal{A}$), or
2. a solution of the alternative problem of problem P($\mathcal{A}$), or
3. a recommendation of scaling problem P($\mathcal{A}$).

If result 1 or 2 is returned by the basic procedure, then the feasibility of the problem P($\mathcal{A}$) can be determined, and the main procedure stops. If result (3) is returned, the problem is scaled appropriately and the basic procedure is called again. It should be noted that the purpose of rescaling differs between [2] and [11].

In [11], the authors devised a rescaling method so that the following value becomes larger:

$$\delta(\mathrm{Ker}\mathcal{A} \cap \mathcal{K}) := \max_{x} \left\{ \det(x) \mid x \in \mathrm{Ker}\mathcal{A} \cap \mathcal{K}, \|x\|_J^2 = r \right\},$$

where $\mathrm{Ker}\mathcal{A} := \{x \mid \mathcal{A}(x) = \mathbf{0}\}$ and $\|x\|_J$ is the norm induced by the inner product $\langle x, y\rangle = \mathrm{trace}(x \circ y)$ defined in section 2.3. After showing that their algorithm terminates in $\log_{1.5} 1/\max\left(\delta(\mathrm{Ker}\mathcal{A} \cap \mathcal{K}), \delta(\mathrm{Im}\mathcal{A}^T \cap \mathcal{K})\right)$ iterations, they proposed four updating schemes to be employed in the basic procedure (the perceptron scheme, von Neuman scheme, smooth perceptron scheme, and von Neumann with away step scheme) and conducted numerical experiments to compare the effect of these schemes when the symmetric cone is the nonnegative orthant [12].

In [2], the authors assumed that the symmetric cone $\mathcal{K}$ is given by the Cartesian product of ps simple symmetric cones $\mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_p$, and they investigated the feasibility of the problem ($\mathrm{P}_{S_{1,\infty}}(\mathcal{A})$),

$$\begin{array}{lll} \mathrm{P}_{S_{1,\infty}}(\mathcal{A}) & \text{find} & x \\ & \text{s.t.} & \mathcal{A}(x) = \mathbf{0}, \\ & & \|x\|_{1,\infty} \le 1, \\ & & x \in \mathrm{int}\mathcal{K}, \end{array}$$

where for each $x = (x_1, x_2, \dots, x_p) \in \mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \mathcal{K}_p$, $\|x\|_{1,\infty}$ is defined by

$$\|x\|_{1,\infty} := \max\{\|x_1\|_1, \dots, \|x_p\|_1\},$$

and $\|x\|_1$ is the sum of the absolute values of all eigenvalues of x. Note that if $p = 1$ then problem $\mathrm{P}_{S_{1,\infty}}(\mathcal{A})$ turns out to be $\mathrm{P}_{S_1}(\mathcal{A})$ which is equivalent to P($\mathcal{A}$):

$$\begin{array}{lll} \mathrm{P}_{S_1}(\mathcal{A}) & \text{find} & x \\ & \text{s.t.} & \mathcal{A}(x) = \mathbf{0}, \\ & & \|x\|_1 \le 1, \\ & & x \in \mathrm{int}\mathcal{K}. \end{array}$$

The authors focused on the volume of the feasible region of $\mathrm{P}_{S_{1,\infty}}(\mathcal{A})$ and devised a rescaling method so that the volume becomes larger. Their method will stop when the feasibility of problem $\mathrm{P}_{S_{1,\infty}}(\mathcal{A})$ or the fact that the minimum eigenvalue of any feasible solution of problem $\mathrm{P}_{S_{1,\infty}}(\mathcal{A})$ is less than ε is determined. It stops in $\frac{r}{\varphi(2)} \log\left(\frac{1}{\varepsilon}\right) - \sum_{l=1}^{p} \frac{r_l \log(r_l)}{\varphi(2)}$ iterations, where r is the rank of $\mathcal{K}$, r_l is the rank of $\mathcal{K}_\ell (\ell = 1, 2, \dots, p)$, $\varphi(\rho) = 2 - \frac{1}{\rho} - \sqrt{3 - \frac{2}{\rho}}$, and ε is a sufficiently small positive value.

The aim of this paper is to devise a new variant of Chubanov's method for solving P($\mathcal{A}$) by extending Roos's method [13] to the following feasibility problem ($\mathrm{P}_{S_\infty}(\mathcal{A})$) over the symmetric cone $\mathcal{K}$:

$$\begin{array}{lll} \mathrm{P}_{S_\infty}(\mathcal{A}) & \text{find} & x \\ & \text{s.t.} & \mathcal{A}(x) = \mathbf{0}, \\ & & \|x\|_\infty \le 1, \\ & & x \in \mathrm{int}\mathcal{K}, \end{array}$$

where $\|x\|_\infty$ is the maximum absolute eigenvalue of x. Throughout this paper, we will assume that $\mathcal{K}$ is the Cartesian product of p simple symmetric cones $\mathcal{K}_1, \dots, \mathcal{K}_p$, i.e., $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_p$.

Our method has a feature that the main algorithm works while keeping information about the minimum eigenvalue of any feasible solution of $\mathrm{P}_{S_\infty}(\mathcal{A})$ and, in this sense, the method is closely related to Lourenço et al.'s method [2]. Using the norm $\|\cdot\|_\infty$ in problem $\mathrm{P}_{S_\infty}(\mathcal{A})$ makes it possible to

- calculate the upper bound of the minimum eigenvalue of any feasible solution of $\mathrm{P}_{S_\infty}(\mathcal{A})$,
- quantify the feasible region of $\mathrm{P}(\mathcal{A})$, and hence,
- determine whether there exists a feasible solution of $\mathrm{P}(\mathcal{A})$ whose minimum eigenvalue is greater than ε as in [2].

Note that the symmetric cone optimization includes several types of problems (linear, second-order cone, and semi-definite optimization problems) with various problem settings and the computational bound of an algorithm depends on these settings. As we will describe in section 5, the theoretical computational bound of our method is

- equivalent to Roos's original method [13] and superior to the Lourenço et al.'s method [2] when the symmetric cone is the nonnegative orthant,
- superior to Lourenço et al.'s method when the symmetric cone is a Cartesian product of second-order cones, and
- equivalent to Lourenço et al.'s method when the symmetric cone is the simple positive semidefinite cone, under the assumption that the costs of computing the spectral decomposition and of the minimum eigenvalue are of the same order for any given symmetric matrix.

Another aim of this paper is to give comprehensive numerical comparisons of the existing algorithms and our method. As described in section 6, we generate the following three types of instance:

- strongly feasible instances, i.e., $\mathrm{Ker}\mathcal{A} \cap \mathrm{int}\mathcal{K} \neq \emptyset$,
- weakly feasible instances, i.e., $\mathrm{Ker}\mathcal{A} \cap \mathrm{int}\mathcal{K} = \emptyset$, but $\mathrm{Ker}\mathcal{A} \cap \mathcal{K} \setminus \{\mathbf{0}\} \neq \emptyset$, and
- infeasible instances, i.e., $\mathrm{Ker}\mathcal{A} \cap \mathcal{K} = \{\mathbf{0}\}$

for the simple positive semidefinite cone $\mathcal{K}$, and conduct numerical experiments. The results show that our method is reliable and quite a bit faster than the existing algorithms. We focus on comparing our method with Lourenço et al.'s in section 7 and show that it can reduce the search region more efficiently than Lourenço et al.'s.

The paper is organized as follows: Section 2 contains a brief description of Euclidean Jordan algebras and their basic properties. Section 3 gives a collection of propositions which are necessary to extend Roos's method to problem $\mathrm{P}_{S_\infty}(\mathcal{A})$ over the symmetric cone. In sections 4 and 5, we explain the basic procedure and the main algorithm of our variant of Chubanov's method. In section 6, we conduct numerical experiments comparing our variant with the existing methods. Then, in section 7, we make more detailed comparisons of Lourenço et al.'s method and our method in terms of the possible reduction rate of the search region and the detection performance of an ε-feasible solution. The conclusions are summarized in section 8.

2 Euclidean Jordan algebras and their basic properties

2.1 Euclidean Jordan algebras

For a real-valued vector space $\mathbb{E}$, $(\mathbb{E}, \circ)$ is called a Jordan algebra if it satisfies

$$x \circ y = y \circ x, \quad x \circ (x^2 \circ y) = x^2 \circ (x \circ y)$$

for all $x, y \in \mathbb{E}$ and $x^2 := x \circ x$. Let e be the identity element of $(\mathbb{E}, \circ)$, i.e.,$x \circ e = e \circ x = x$ holds for any $x \in \mathbb{E}$. We denote $y \in \mathbb{E}$ as x^{-1} if y satisfies $x \circ y = e$. If a Jordan algebra $(\mathbb{E}, \circ)$ has the identity element and an inner product $\langle \cdot, \cdot \rangle$ for all $x, y, z \in \mathbb{E}$,

$$\langle x \circ y, z \rangle = \langle y, x \circ z \rangle.$$

$c \in \mathbb{E}$ is called an *idempotent* if it satisfies $c \circ c = c$, and an idempotent c is called *primitive* if it can not be written as a sum of two or more nonzero idempotents. A set of primitive idempotents $c_1, c_2, \ldots c_k$ is called a *Jordan frame* if $c_1, \ldots c_k$ satisfy

$$c_i \circ c_j = 0 \ (i \neq j), \quad c_i \circ c_i = c_i \ (i = 1, \ldots, k), \quad \sum_{i=1}^{k} c_i = e.$$

For $x \in \mathbb{E}$, the *degree* of x is the smallest integer d such that the set $\{e, x, x^2, \ldots, x^d\}$ is linearly independent. The *rank* of $\mathbb{E}$ is the maximum integer r of the degree of x over all $x \in \mathbb{E}$. The following properties are known.

Proposition 2.1 (Spectral theorem (cf. Theorem III.1.2 of [6]))**.** *Let $(\mathbb{E}, \circ)$ be a Euclidean Jordan algebra having rank r. For any $x \in \mathbb{E}$, there exist real numbers $\lambda_1, \ldots, \lambda_r$ and a Jordan frame $c_1, \ldots, c_r$ for which the following holds:*

$$x = \sum_{i=1}^{r} \lambda_i c_i.$$

The numbers $\lambda_1, \ldots, \lambda_r$ are uniquely determined eigenvalues *of x (with their multiplicities). Furthermore,*

$$\text{trace}(x) := \sum_{i=1}^{r} \lambda_i, \quad \det(x) := \prod_{i=1}^{r} \lambda_i.$$

2.2 Symmetric cone

A proper cone is symmetric if it is self-dual and homogeneous. It is known that the set of squares

$$\mathcal{K} = \{x^2 : x \in \mathbb{E}\}$$

is the symmetric cone of $\mathbb{E}$ (cf. Theorems III.2.1 and III.3.1 of [6]).

The following properties can be derived from the results in [6], as in Corollary 2.3 of [17]:

Proposition 2.2. *Let $x \in \mathbb{E}$ and let $\sum_{j=1}^{r} \lambda_j c_j$ be a decomposition of x given by Propositoin 2.1. Then*

(i) *$x \in \mathcal{K}$ if and only if $\lambda_j \geq 0$ $(j = 1, 2, \ldots, r)$,*

(ii) *$x \in \text{int}\mathcal{K}$ if and only if $\lambda_j > 0$ $(j = 1, 2, \ldots, r)$.*

A Euclidean Jordan algebra $(\mathbb{E}, \circ)$ is called *simple* if it cannot be written as any Cartesian product of non-zero Euclidean Jordan algebras. If the Euclidean Jordan algebra $(\mathbb{E}, \circ)$ associated with a symmetric cone $\mathcal{K}$ is simple, then we say that $\mathcal{K}$ is *simple.* In this paper, we will consider that $\mathcal{K}$ is given by a Cartesian product of p simple symmetric cones $\mathcal{K}_\ell$,

$$\mathcal{K} := \mathcal{K}_1 \times \cdots \times \mathcal{K}_p,$$

whose rank and identity element are r_ℓ and e_ℓ $(\ell = 1, 2, \ldots, p)$. The rank r and the identity element of $\mathcal{K}$ are given by

$$r = \sum_{\ell=1}^{p} r_\ell, \quad e = (e_1, \ldots, e_p). \tag{3}$$

In what follows, x_ℓ stands for the ℓ-th block element of $x \in \mathcal{K}$, i.e., $x = (x_1, \ldots . x_p) \in \mathcal{K}_1 \times \cdots \times \mathcal{K}_p$. For each $\ell = 1, 2, \ldots, p$, we define

$$\lambda_{\min}(x_\ell) := \min\{\lambda_1,\ \lambda_2, \ldots, \lambda_{r_\ell}\}$$

where $\lambda_1,\ \lambda_2, \cdots, \lambda_{r_\ell}$ are eigenvalues of x_ℓ. The minimum eigenvalue $\lambda_{\min}(x)$ of $x \in \mathcal{K}$ is given by

$$\lambda_{\min}(x) = \min\{\lambda_{\min}(x_1), \lambda_{\min}(x_2), \ldots, \lambda_{\min}(x_p)\}.$$

Next, we consider the *quadratic representation* $Q_v(x)$ defined by

$$Q_v(x) := 2v \circ (v \circ x) - v^2 \circ x.$$

For the cone $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_p$, the quadratic representation $Q_v(x)$ of $x \in \mathcal{K}$ is denoted by $Q_v(x) = \big(Q_{v_1}(x_1), \ldots, Q_{v_p}(x_p)\big)$. Letting I_ℓ be the identity operator of the Euclidean Jordan algebra $(\mathbb{E}_\ell, \circ_\ell)$ associated with the cone $\mathcal{K}_\ell$, we have $Q_{e_\ell} = I_\ell$ for $\ell = 1, 2, \ldots, p$.

The following properties can also be retrieved from the results in [6] as in Proposition 3 of [2]:

Proposition 2.3. *For any $v \in \mathrm{int}\mathcal{K}$, $Q_v(\mathcal{K}) = \mathcal{K}$.*

More detailed descriptions including concrete examples of symmetric cone optimization can be found in e.g., [6, 7, 14, 1].

2.3 Notation

This subsection summarizes the notation used in this paper. For any $x, y \in \mathbb{E}$, we define the inner product $\langle \cdot, \cdot \rangle$ and the norm $\| \cdot \|_J$ as follows:

$$\langle x, y \rangle := \mathrm{trace}(x \circ y), \quad \|x\|_J := \sqrt{\langle x, x \rangle}.$$

For any $x \in \mathbb{E}$ having decomposition $x = \sum_{i=1}^{r} \lambda_i c_i$ as in Proposition 2.1, we also define

$$\|x\|_1 := |\lambda_1| + \cdots + |\lambda_r|, \quad \|x\|_\infty := \max\{|\lambda_1|, \ldots, |\lambda_r|\}.$$

For $x \in \mathcal{K}$, we obtain the following equivalent representations:

$$\|x\|_1 = \langle e, x \rangle, \quad \|x\|_\infty = \lambda_{\max}(x).$$

The following is a list of other definitions and frequently used symbols in the paper.

- d: the dimension of the Euclidean space $\mathbb{E}$ corresponding to $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_p$,
- $F_{\mathrm{P}_{S_\infty}(\mathcal{A})}$: the feasible region of $\mathrm{P}_{S_\infty}(\mathcal{A})$,
- $P_{\mathcal{A}}(\cdot)$: the projection map on to $\mathrm{Ker}\mathcal{A}$,
- $Q_{\mathcal{A}}(\cdot)$: the projection map on to $\mathrm{Im}\mathcal{A}^T$,
- $\mathcal{P}_{\mathcal{K}}(\cdot)$: the projection map on to $\mathcal{K}$,
- $\lambda(x) \in \mathbb{R}^r$: an r-dimensional vector composed of the eigenvalues of $x \in \mathcal{K}$,
- $\lambda(x_\ell) \in \mathbb{R}^{r_\ell}$: an r_ℓ-dimensional vector composed of the eigenvalues of $x_\ell \in \mathcal{K}_\ell$ $(\ell = 1, 2, \ldots, p)$,
- $[\cdot]^+$: the projection map onto the nonnegative orthant, and
- $\mathcal{A}^*(\cdot)$: the adjoint operator of the linear operator $\mathcal{A}(\cdot)$, i.e., $\langle \mathcal{A}(x), y \rangle = \langle x, \mathcal{A}^*(y) \rangle$ for all $x \in \mathcal{K}$ and $y \in \mathbb{R}^m$.

3 Extension of Roos's method to the symmetric cone problem

3.1 Outline of the extended method

We focus on the feasibility of the following problem $\mathrm{P}_{S_\infty}(\mathcal{A})$, which is equivalent to $\mathrm{P}(\mathcal{A})$:

$$\begin{array}{lll} \mathrm{P}_{S_\infty}(\mathcal{A}) & \text{find} & x \\ & \text{s.t.} & \mathcal{A}(x) = \mathbf{0}, \\ & & \|x\|_\infty \leq 1, \\ & & x \in \mathrm{int}\mathcal{K}. \end{array}$$

The alternative problem $\mathrm{D}(\mathcal{A})$ of $\mathrm{P}(\mathcal{A})$ is given by

$$\begin{array}{lll} \mathrm{D}(\mathcal{A}) & \text{find} & y \\ & \text{s.t.} & y \in \mathrm{Im}\mathcal{A}^T, \\ & & y \in \mathcal{K}, y \neq \mathbf{0}, \end{array}$$

where $\mathrm{Im}\mathcal{A}^T$ is the orthogonal complement of $\mathrm{Ker}\mathcal{A}$. As we mentioned in section 2.2, we assume that $\mathcal{K}$ is given by a Cartesian product of p simple symmetric cones $\mathcal{K}_\ell(\ell = 1, 2, \ldots, p)$, i.e., $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_p$.

In our method, the upper bound of the sum of eigenvalues of a feasible solution of $\mathrm{P}_{S_\infty}(\mathcal{A})$ plays a key role, whereas the existing work focuses on the volume of the set of the feasible region [2] or the condition number of a feasible solution [11]. In the next section, we will explain our method when $\mathcal{K}$ is simple, i.e., $p = 1$. Then, we will generalize it to the case where $p \geq 2$.

3.2 Simple symmetric cone case

Let us consider the case where $\mathcal{K}$ is simple, i.e., $p = 1$. It is obvious that, for any feasible solution x of $\mathrm{P}_{S_\infty}(\mathcal{A})$, the constraint $\|x\|_\infty \leq 1$ implies the upper bound $\langle e, x\rangle \leq r$ of the sum of eigenvalues, since $x \in \mathcal{K}$. In Proposition 3.2, we show that this bound may be improved as $\langle e, x\rangle < r$ by using a point $v \in \mathrm{Im}\mathcal{A}^T \setminus \{0\}$.

The following results are used to prove Proposition 3.2.

Proposition 3.1. *Let* $(\mathbb{E}, \circ)$ *be a Euclidean Jordan Algebra with the corresponding symmetric cone* $\mathcal{K}$*. For a given* $c \in \mathbb{E}$*, consider the problem*

$$\begin{array}{ll} \max & \langle c, x\rangle \\ \text{s.t} & \mathcal{A}(x) = \mathbf{0}, \\ & \mathbf{0} \leq \lambda(x) \leq \mathbf{1}. \end{array}$$

The dual problem of the above is

$$\begin{array}{ll} \min & \langle \mathcal{P}_\mathcal{K}(c - u), e\rangle \\ \text{s.t} & u \in \mathrm{Im}\mathcal{A}^T. \end{array}$$

Proof.

Define the Lagrangian function $L(x, w)$ as

$$L(x, w) := \langle c, x\rangle - \langle w, \mathcal{A}(x)\rangle$$

where $w \in \mathbb{R}^m$ is the Lagrangian multiplier. Then we have

$$\begin{aligned}
\max_{0\le\lambda(x)\le1}\min_{w} L(x,w) &\le \min_{w}\max_{0\le\lambda(x)\le1} L(x,w)\\
&= \min_{w}\max_{0\le\lambda(x)\le1}\{\langle c,x\rangle - \langle \mathcal{A}^*(w),x\rangle\}\\
&= \min_{w}\max_{0\le\lambda(x)\le1}\{\langle c-\mathcal{A}^*(w),x\rangle\}\\
&= \min_{w}\max_{0\le\lambda(x)\le1}\langle \mathcal{P}_{\mathcal{K}}\left(c-\mathcal{A}^*(w)\right),x\rangle\\
&= \min_{w}\langle \mathcal{P}_{\mathcal{K}}\left(c-\mathcal{A}^*(w)\right),e\rangle\\
&= \min_{u\in \mathrm{Im}\mathcal{A}^T}\langle \mathcal{P}_{\mathcal{K}}\left(c-u\right),e\rangle,
\end{aligned}$$

and the dual problem is

$$\begin{array}{ll}
\min & \langle \mathcal{P}_{\mathcal{K}}\left(c-u\right),e\rangle\\
\text{s.t.} & u\in \mathrm{Im}\mathcal{A}^T.
\end{array}$$

□

The following is a key proposition that relates to the stopping criteria of our method.

Proposition 3.2. *Suppose that $v \in \mathrm{Im}\mathcal{A}^T$ is given by*

$$v=\sum_{i=1}^{r}\lambda_i c_i$$

as in Proposition 2.1. For each $i \in \{1,\dots,r\}$ and $\alpha\in\mathbb{R}$, define

$$q_i(\alpha) := [1-\alpha\lambda_i]^+ + \sum_{j\ne i}^{r}[-\alpha\lambda_j]^+.$$

Then,

$$\langle c_i,x\rangle \le \min_{\alpha\in\mathbb{R}} q_i(\alpha) = \begin{cases}\min\left\{1,\left\langle e,\mathcal{P}_{\mathcal{K}}\left(-\frac{1}{\lambda_i}v\right)\right\rangle\right\} & \text{if } \lambda_i\ne 0,\\ 1 & \text{if } \lambda_i = 0\end{cases} \tag{4}$$

hold for any $x\in F_{\mathrm{P}_{S_\infty}(\mathcal{A})}$ and $i\in\{1,\dots,r\}$.

Proof. For each $i\in\{1,2,\dots,r\}$, we have

$$\mathcal{P}_{\mathcal{K}}\left(c_i-\alpha v\right) = \mathcal{P}_{\mathcal{K}}\left(c_i-\alpha\sum_{j=1}^{r}\lambda_j c_j\right) = \mathcal{P}_{\mathcal{K}}\left((1-\alpha\lambda_i)c_i-\sum_{j\ne i}^{r}\alpha\lambda_j c_j\right),$$

and hence,

$$\langle \mathcal{P}_{\mathcal{K}}\left(c_i-\alpha v\right),e\rangle = \left\langle \mathcal{P}_{\mathcal{K}}\left((1-\alpha\lambda_i)c_i-\sum_{j\ne i}^{r}\alpha\lambda_j c_j\right),\sum_{i=1}^{r}c_i\right\rangle = [1-\alpha\lambda_i]^+ + \sum_{j\ne i}^{r}[-\alpha\lambda_j]^+ = q_i(\alpha). \tag{5}$$

Note that, since $q_i(\alpha)$ is a piece-wise linear convex function, if $\lambda_i = 0$, it attains the minimum at $\alpha = 0$ with $q_i(0) = 1$, and if $\lambda_i \ne 0$, it attains the minimum at $\alpha = 0$ with $q_i(0) = 1$ or at $\alpha = \frac{1}{\lambda_i}$ with

$$q\left(\frac{1}{\lambda_i}\right) = \sum_{j\ne i}^{r}\left[-\frac{\lambda_j}{\lambda_i}\right]^+ = \sum_{j=1}^{r}\left[-\frac{\lambda_j}{\lambda_i}\right]^+ = \left\langle e,\mathcal{P}_{\mathcal{K}}\left(-\frac{1}{\lambda_i}v\right)\right\rangle.$$

Thus, we obtain the equivalence in (4).

Since $\alpha v \in \mathrm{Im}\mathcal{A}^T$ for all $\alpha \in \mathbb{R}$, for each $i \in \{1, \ldots, r\}$, Proposition 3.1 and (5) ensure that

$$\langle c_i, x\rangle \leq \langle \mathcal{P}_{\mathcal{K}}(c_i - \alpha v), e\rangle = q_i(\alpha)$$

for all $\alpha \in \mathbb{R}$, which implies the inequality in (4). □

Since $\sum_{i=1}^r c_i = e$ holds, Proposition 3.2 allows us to compute upper bounds of the sum of eigenvalues λ_i ($i \in \{1, \ldots, r\}$). The following proposition gives us information about indexes whose upper bound of $\langle c_i, x\rangle$ in Proposition 3.2 is less than 1.

Proposition 3.3. *Suppose that $v \in \mathrm{Im}\mathcal{A}^T$ is given by*

$$v = \sum_{i=1}^r \lambda_i c_i$$

as in Proposition 2.1. If v satisfies

$$\min\left\{1, \left\langle e, \mathcal{P}_{\mathcal{K}}\left(-\frac{1}{\lambda_i} v\right)\right\rangle\right\} = \left\langle e, \mathcal{P}_{\mathcal{K}}\left(-\frac{1}{\lambda_i} v\right)\right\rangle = \xi < 1$$

for some $\xi < 1$ and for some $i \in \{1, \ldots, r\}$ for which $\lambda_i \neq 0$ holds, then λ_i has the same sign as $\langle e, v\rangle$.

Proof. First, we consider the case where $\lambda_i > 0$. Since the assumption implies that $\langle e, \mathcal{P}_{\mathcal{K}}(-v)\rangle = \lambda_i \xi$, we have

$$\langle e, v\rangle = \langle e, \mathcal{P}_{\mathcal{K}}(v)\rangle - \langle e, \mathcal{P}_{\mathcal{K}}(-v)\rangle = \langle e, \mathcal{P}_{\mathcal{K}}(v)\rangle - \lambda_i \xi \geq \lambda_i(1 - \xi) > 0.$$

For the case where $\lambda_i < 0$, since the assumption also implies that $-\langle e, \mathcal{P}_{\mathcal{K}}(-v)\rangle = -\lambda_i \xi$, we have

$$\langle e, v\rangle = \langle e, \mathcal{P}_{\mathcal{K}}(v)\rangle - \langle e, \mathcal{P}_{\mathcal{K}}(-v)\rangle = -\lambda_i \xi - \langle e, \mathcal{P}_{\mathcal{K}}(-v)\rangle \leq -\lambda_i \xi - (-\lambda_i) = (1 - \xi)\lambda_i < 0. \tag{6}$$

This completes the proof. □

The above two propositions imply that, for any $v \in \mathrm{Im}\mathcal{A}^T$ with $v = \sum_{i=1}^r \lambda_i c_i$, if we compute $\langle c_i, x\rangle$ according to Proposition 3.2 for $i \in \{1, \ldots, r\}$ having the same sign as the one of $\langle e, v\rangle$, we obtain an upper bound of the sum of eigenvalues of x over the set $F_{\mathrm{P}_{S_\infty}(\mathcal{A})}$.

Proposition 3.4. *Suppose that a nonempty index set $H \subseteq \{1, \ldots r\}$, Jordan frame $c_1, \ldots, c_r$, and $0 < \xi < 1$ satisfy*

$$\langle c_i, x\rangle \leq \xi \ (i \in H), \quad \langle c_i, x\rangle \leq 1 \ (i \notin H)$$

for any $x \in F_{\mathrm{P}_{S_\infty}(\mathcal{A})}$. Let us define $g \in \mathrm{int}\mathcal{K}$ as

$$g := \sqrt{\xi} \sum_{h \in H} c_h + \sum_{h \notin H} c_h \quad \text{i.e.,} \quad g^{-1} = \frac{1}{\sqrt{\xi}} \sum_{h \in H} c_h + \sum_{h \notin H} c_h \tag{7}$$

and set $\bar{x} := Q_{g^{-1}}(x)$. Then, any feasible solution $\bar{x}$ of the scaled problem $\mathrm{P}_{S_\infty}(\mathcal{A}Q_{g^{-1}})$ satisfies

$$\langle c_i, \bar{x}\rangle \leq 1 \ (i \in \{1, \ldots, r\})$$

and hence, $\langle e, \bar{x}\rangle \leq r$.

Proof. For $Q_{g^{-1}}$, and $c_1, \ldots, c_r$, the following equations hold:

- For any $i \in H$,

$$\begin{aligned} Q_{g^{-1}}(c_i) &= 2g^{-1} \circ (g^{-1} \circ c_i) - (g^{-1} \circ g^{-1}) \circ c_i \\ &= 2g^{-1} \circ \frac{1}{\sqrt{\xi}} c_i - \left(\frac{1}{\xi} \sum_{h \in H} c_h + \sum_{h \notin H} c_h \right) \circ c_i \\ &= \frac{2}{\xi} c_i - \frac{1}{\xi} c_i = \frac{1}{\xi} c_i. \end{aligned}$$

- For any $i \notin H$,

$$\begin{aligned} Q_{g^{-1}}(c_i) &= 2g^{-1} \circ (g^{-1} \circ c_i) - (g^{-1} \circ g^{-1}) \circ c_i \\ &= 2g^{-1} \circ c_i - \left(\frac{1}{\xi} \sum_{h \in H} c_h + \sum_{h \notin H} c_h \right) \circ c_i \\ &= 2c_i - c_i = c_i. \end{aligned}$$

Therefore, the inner products of $\bar{x} = Q_{g^{-1}}(x)$ and $c_1, \dots, c_r$ are as follows:

- For any $i \in H$,

$$\langle c_i, \bar{x} \rangle = \left\langle c_i, Q_{g^{-1}}(x) \right\rangle = \left\langle Q_{g^{-1}}(c_i), x \right\rangle = \left\langle \frac{1}{\xi} c_i, x \right\rangle \leq \frac{1}{\xi} \xi = 1.$$

- For any $i \notin H$,

$$\langle c_i, \bar{x} \rangle = \langle c_i, Q_{g^{-1}}(x) \rangle = \langle Q_{g^{-1}}(c_i), x \rangle = \langle c_i, x \rangle \leq 1.$$

□

3.3 Non-simple symmetric cone case

In this section, we consider the case where the symmetric cone is not simple; i.e., it is a Cartesian product of p simple symmetric cones $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_p$ whose rank is given by (3). Propositions 3.5 and 3.6 are extensions of Proposition 3.2 and 3.3, respectively.

Proposition 3.5. *Suppose that, for any $v \in \mathrm{Im}\mathcal{A}^T$, the ℓ-th block element v_ℓ of $v \in \mathbb{E}$ is decomposed into*

$$v_\ell = \sum_{i=1}^{r_\ell} \lambda(v_\ell)_i c(v_\ell)_i$$

as in Proposition 2.1. For each $\ell \in \{1, 2, \dots, p\}$ and $i \in \{1, 2, \dots, r_p\}$, define

$$q_{\ell,i}(\alpha) := [1 - \alpha\lambda(v_\ell)_i]^+ + \sum_{k \neq i}^{r_\ell} [-\alpha\lambda(v_\ell)_k]^+ + \sum_{j \neq \ell}^{p} \sum_{k=1}^{r_j} \left[-\alpha\lambda(v_j)_k\right]^+. \tag{8}$$

Then,

$$\langle c(v_\ell)_i, x_\ell \rangle \leq \min_{\alpha \in \mathbb{R}} q_{\ell,i}(\alpha) = \begin{cases} \min \left\{1, \left\langle e, \mathcal{P}_{\mathcal{K}} \left(-\frac{1}{\lambda(v_\ell)_i} v \right) \right\rangle \right\} & \text{if } \lambda(v_\ell)_i \neq 0, \\ 1 & \text{if } \lambda(v_\ell)_i = 0 \end{cases} \tag{9}$$

holds for any feasible solution x of $\mathrm{P}_{S_\infty}(\mathcal{A})$, $\ell \in \{1, 2, \dots, p\}$ and $i \in \{1, 2, \dots, r_p\}$.

Proof. Let $c \in \mathbb{E}$ be the element where the ℓ-th block element is $c_\ell = c(v_\ell)_i$ and the other block elements take 0. For any real number $\alpha \in \mathbb{R}$, Proposition 3.1 ensures that

$$\begin{aligned}\langle c(v_\ell)_i, x_\ell \rangle = \langle c, x \rangle &\leq \langle \mathcal{P}_\mathcal{K}(c - \alpha v), e \rangle \\ &= \langle \mathcal{P}_{\mathcal{K}_\ell}(c(v_\ell)_i - \alpha v_\ell), e_\ell \rangle + \sum_{j \neq \ell}^{p} \langle \mathcal{P}_{\mathcal{K}_j}(-\alpha v_j), e_j \rangle \\ &= [1 - \alpha \lambda(v_\ell)_i]^+ + \sum_{k \neq i}^{r_\ell} [-\alpha \lambda(v_\ell)_k]^+ + \sum_{j \neq \ell}^{p} \sum_{k=1}^{r_j} [-\alpha \lambda(v_j)_k]^+ = q_{\ell,i}(\alpha). \end{aligned} \tag{10}$$

We obtain (9) by following a similar argument to the one used in the proof of Proposition 3.2. □

The next proposition follows similarly to Proposition 3.3, by noting that $\langle e, \mathcal{P}_\mathcal{K}(-v) \rangle = \lambda(v_\ell)_i \xi$ holds if $\lambda(v_\ell)_i > 0$ and that $\langle e, \mathcal{P}_\mathcal{K}(v) \rangle = -\lambda(v_\ell)_i \xi$ if $\lambda(v_\ell)_i < 0$.

Proposition 3.6. *Suppose that, for any $v \in \mathrm{Im}\mathcal{A}^T$, each ℓ-th block element v_ℓ of v is decomposed into*

$$v_\ell = \sum_{i=1}^{r_\ell} \lambda(v_\ell)_i c(v_\ell)_i$$

as in Proposition 2.1. If v satisfies

$$\lambda(v_\ell)_i \neq 0 \quad \textit{and} \quad \left\langle e, \mathcal{P}_\mathcal{K}\left(-\tfrac{1}{\lambda(v_\ell)_i} v \right) \right\rangle = \xi_\ell < 1 \tag{11}$$

for some $\xi < 1$, $\ell \in \{1, \dots p\}$ and $i \in \{1, \dots r_\ell\}$, then $\lambda(v_\ell)_i$ has the same sign as $\langle e, v \rangle$.

From Proposition 3.5, if we obtain $v \in \mathrm{Im}\mathcal{A}^T$ satisfying (11) for a block $\ell \in \{1, \dots p\}$ with an index $i \in \{1, \dots r_\ell\}$, then the upper bound of the sum of the eigenvalues of any feasible solution x of $\mathrm{P}_{S_\infty}(\mathcal{A})$ is reduced by $\langle e, x \rangle \leq r - 1 + \xi_\ell < r$. In this case, as described below, we can find a scaling such that the sum of eigenvalues of any feasible solution of $\mathrm{P}_{S_\infty}(\mathcal{A})$ is bounded by r.

Let H_ℓ be the set of indexes i satisfying (11) for each block ℓ. According to Proposition 3.4, set $g_\ell = \sqrt{\xi_\ell} \sum_{h \in H_\ell} c(v_\ell)_h + \sum_{h \notin H_\ell} c(v_\ell)_h$ and define the linear operator Q as follows:

$$Q_\ell := \begin{cases} Q_{g_\ell^{-1}} & \text{if } |H_\ell| \neq 0, \\ I_\ell & \text{otherwise,} \end{cases}$$

$$Q(\mathbb{E}_1, \dots, \mathbb{E}_p) := (Q_1(\mathbb{E}_1), \dots, Q_p(\mathbb{E}_p)),$$

where I_ℓ is the identity operator of the Euclidean Jordan algebra $\mathbb{E}_\ell$ associated with the symmetric cone $\mathcal{K}_\ell$. From Proposition 3.4 and its proof, we can easily see that

$$Q_{g_\ell^{-1}}(c_i) = \frac{1}{\xi} c_i \ (i \in H_\ell), \quad Q_{g_\ell^{-1}}(c_i) = c_i \ (i \notin H_\ell), \tag{12}$$

and the sum of eigenvalues of any feasible solution of the scaled problem $\mathrm{P}_{S_\infty}(\mathcal{A}Q)$ is bounded by $r = \sum_{\ell=1}^{p} r_\ell$.

4 Basic procedure of the extended method

4.1 Outline of the basic procedure

In this section, we describe the details of our basic procedure. First, we introduce our stopping criteria and explain how to update y^k when the the stopping criteria is not satisfied. Next, we show that the stopping criteria is satisfied within a finite number of iterations, i.e., finite termination of the basic procedure. Our stopping criteria is new and different from the ones used in [2, 11], while the method of updating y^k is similar to the one used in [2] or in the von Neumann scheme of [11]. Algorithm 1 is a full description of our basic procedure.

4.2 Termination conditions of the basic procedure

For $z^k = P_{\mathcal{A}}(y^k)$, $v^k = y^k - z^k$ and a given $\xi \in (0,1)$, our basic procedure terminates when any of the following four cases occurs:

1. $z^k \in \text{int}\mathcal{K}$ meaning that z^k is a solution of P($\mathcal{A}$),
2. $z^k = \mathbf{0}$ meaning that y^k is feasible for D($\mathcal{A}$),
3. $y^k - z^k \in \mathcal{K}$ and $y^k - z^k \neq \mathbf{0}$ meaning that $y^k - z^k$ is feasible for D($\mathcal{A}$), or
4. there exist $\ell \in \{1, \dots, p\}$ and $i \in \{1, \dots, r_\ell\}$ for which

$$\lambda(v_\ell^k)_i \neq 0 \quad \text{and} \quad \left\langle e, \mathcal{P}_{\mathcal{K}}\left(-\frac{1}{\lambda(v_\ell^k)_i} v^k\right)\right\rangle = \xi_\ell \leq \xi < 1, \tag{13}$$

   meaning that $\langle e, x \rangle < r$ holds for any feasible solution x of $\text{P}_{S_\infty}(\mathcal{A})$ (see Proposition 3.5).

Cases 1 and 2 are direct extensions of the cases in [4], while case 3 was proposed in [8, 2]. Case 3 helps us to determine the feasibility of P($\mathcal{A}$) efficiently while we have to decompose $y^k - z^k$ for checking it.

If the basic procedure ends with case 1, 2, or 3, the feasibility of P($\mathcal{A}$) can be determined, and the basic procedure returns a solution of P($\mathcal{A}$) or D($\mathcal{A}$) to the main algorithm. If the basic procedure ends with case 4, the basic procedure returns to the main algorithm p index sets $H_1, \dots, H_p$ each of which consists of indexes i satisfying (13) and the set of primitive idempotents $C_\ell = \{c(v_\ell^k)_1, \dots, c(v_\ell^k)_{r_\ell}\}$ of v_ℓ^k for each ℓ.

4.3 Update of the basic procedure

The basic procedure updates $y^k \in \text{int}\mathcal{K}$ with $\langle y^k, e \rangle = 1$ so as to reduce the value of $\|z^k\|_J$. The following proposition is essentially the same as Proposition 13 in [2], so we will omit its proof.

Proposition 4.1 (cf. Proposition 13, [2])**.** *For $y^k \in \text{int}\mathcal{K}$ satisfying $\langle y^k, e \rangle = 1$, let $z^k = P_{\mathcal{A}}(y^k)$. If $z^k \notin \text{int}\mathcal{K}$ and $z^k \neq \mathbf{0}$, then the following hold.*

1. *There exists $c \in \mathcal{K}$ such that*

$$\langle c, z^k \rangle = \lambda_{\min}(z^k) \leq 0, \ \langle e, c \rangle = 1 \ \textit{and} \ c \in \mathcal{K}. \tag{14}$$

2. *For the above c, suppose that $p = P_{\mathcal{A}}(c)$, $p \neq \mathbf{0}$ and define*

$$\alpha = \frac{\langle p, p - z^k \rangle}{\|z^k - p\|_J^2}. \tag{15}$$

   Then, $y^{k+1} := \alpha y^k + (1-\alpha)c$ satisfies

   (a) $y^{k+1} \in \text{int}\mathcal{K}$,

(b) $\|y^{k+1}\|_{1,\infty} \geq \frac{1}{p}$,

(c) $\langle y^{k+1}, e\rangle = 1$, *and*

(d) $z^{k+1} := P_{\mathcal{A}}(y^{k+1})$ *satisfies*

$$\frac{1}{\|z^{k+1}\|_J^2} \geq \frac{1}{\|z^k\|_J^2} + 1.$$

A method of accelerating the update of y^k is provided in [13]. For $\ell \in \{1, 2, \ldots, p\}$, let $I_\ell := \{i \in \{1, 2, \ldots, r_\ell\} \mid \lambda_i(z_\ell^k) \leq 0\}$ and set $N = \sum_{\ell=1}^p |I_\ell|$. Define the ℓ-th block element of $c \in \mathcal{K}$ as

$$c_\ell = \frac{1}{N} \sum_{i \in I_\ell} c(z_\ell^k)_i.$$

and $p = P_{\mathcal{A}}(c)$. Using this p, the acceleration method computes α by (15) so as to minimize the norm of z^{k+1} and update y by

$$y^{k+1} = \alpha y^k + (1 - \alpha)c.$$

We incorporate this method in the basic procedure for our computational experiment and call it the *modified basic procedure*. A detailed description is given in Appendix A.

4.4 Finite termination of the basic procedure

In this section, we show that the basic procedure (Proposition 4.4) terminates in a finite number of iterations. To do so, we need to prove Lemma 4.2 and Proposition 4.3.

Lemma 4.2. *Let* $(\mathbb{E}, \circ)$ *be a Euclidean Jordan algebra with the corresponding symmetric cone* $\mathcal{K}$ *given by the Cartesian product of* p *simple symmetric cones, i.e.,* $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_p$. *For any* $x \in \mathbb{E}$ *and* $y \in \mathcal{K}$, *the following inequality holds:*

$$[\langle x, y\rangle]^+ \leq \langle \mathcal{P}_{\mathcal{K}}(x), y\rangle.$$

Proof. Let $x \in \mathbb{E}$ and suppose that each ℓ-th block element x_ℓ of x is given by

$$x_\ell = \sum_{i=1}^{r_\ell} \lambda(x_\ell)_i c(x_\ell)_i$$

as in Proposition 2.1. Then, we can see that

$$\begin{aligned}
[\langle x, y\rangle]^+ &= \left[\sum_{\ell=1}^{p} \left\langle \sum_{i=1}^{r_\ell} \lambda(x_\ell)_i c(x_\ell)_i, y_\ell \right\rangle\right]^+ \\
&= \left[\sum_{\ell=1}^{p} \left(\sum_{i=1}^{r_\ell} \lambda(x_\ell)_i \langle c(x_\ell)_i, y_\ell\rangle\right)\right]^+ \\
&\leq \sum_{\ell=1}^{p} \sum_{i=1}^{r_\ell} [\lambda(x_\ell)_i \langle c(x_\ell)_i, y_\ell\rangle]^+ \\
&= \sum_{\ell=1}^{p} \sum_{i=1}^{r_\ell} [\lambda(x_\ell)_i]^+ \langle c(x_\ell)_i, y_\ell\rangle \\
&= \sum_{\ell=1}^{p} \left\langle \sum_{i=1}^{r_\ell} [\lambda(x_\ell)_i]^+ c(x_\ell)_i, y_\ell \right\rangle = \langle \mathcal{P}_{\mathcal{K}}(x), y\rangle.
\end{aligned}$$

where the inequality follows from the fact that $c(x_\ell)_1, \ldots, c(x_\ell)_{r_\ell}$, and y_ℓ lie in the symmetric cone $\mathcal{K}_\ell$. □

Proposition 4.3. *For a given $y \in \mathcal{K}$, define $z = P_{\mathcal{A}}(y)$ and $v = Q_{\mathcal{A}}(y)$. Suppose that $v \neq 0$ and each ℓ-th element v_ℓ is given by $v_\ell = \sum_{i=1}^{r_\ell} \lambda(v_\ell)_i c(v_\ell)_i$, as in Proposition 2.1. Then, for any $x \in F_{\mathrm{P}_{S_\infty}(\mathcal{A})}$, $\ell \in \{1, \ldots, p\}$ and $i \in \{1, \ldots, r_\ell\}$,*

$$\langle c(v_\ell)_i, x_\ell \rangle \leq \min_\alpha q_{\ell,i}(\alpha) \leq \frac{1}{\langle y_\ell, c(v_\ell)_i \rangle} \|z\|_J \tag{16}$$

hold where $q_{\ell,i}(\alpha)$ is defined in (8).

Proof. The first inequality of (16) follows from (10) in the proof of Proposition 3.5. The second inequality is obtained by evaluating $q_{\ell,i}(\alpha)$ at $\alpha = \dfrac{1}{\langle y_\ell, c(v_\ell)_i \rangle}$, as follows:

$$\begin{aligned}
q_{\ell.i}\left(\frac{1}{\langle y_\ell, c(v_\ell)_i \rangle}\right) &= \left[1 - \frac{1}{\langle y_\ell, c(v_\ell)_i \rangle}\lambda(v_\ell)_i\right]^+ + \sum_{k\neq i}^{r_\ell}\left[-\frac{1}{\langle y_\ell, c(v_\ell)_i \rangle}\lambda(v_\ell)_k\right]^+ + \sum_{j\neq \ell}^{p}\sum_{k=1}^{r_j}\left[-\frac{1}{\langle y_\ell, c(v_\ell)_i \rangle}\lambda(v_j)_k\right]^+ \\
&= \left[1 - \frac{\langle y_\ell - z_\ell, c(v_\ell)_i \rangle}{\langle y_\ell, c(v_\ell)_i \rangle}\right]^+ + \sum_{k\neq i}^{r_\ell}\left[-\frac{\langle y_\ell - z_\ell, c(v_\ell)_k \rangle}{\langle y_\ell, c(v_\ell)_i \rangle}\right]^+ + \sum_{j\neq \ell}^{p}\sum_{k=1}^{r_j}\left[-\frac{\langle y_j - z_j, c(v_j)_k \rangle}{\langle y_\ell, c(v_\ell)_i \rangle}\right]^+ \\
&= \left[\frac{\langle z_\ell, c(v_\ell)_i \rangle}{\langle y_\ell, c(v_\ell)_i \rangle}\right]^+ + \sum_{k\neq i}^{r_\ell}\left[\frac{\langle z_\ell, c(v_\ell)_k \rangle - \langle y_\ell, c(v_\ell)_k \rangle}{\langle y_\ell, c(v_\ell)_i \rangle}\right]^+ + \sum_{j\neq \ell}^{p}\sum_{k=1}^{r_j}\left[\frac{\langle z_j, c(v_j)_k \rangle - \langle y_j, c(v_j)_k \rangle}{\langle y_\ell, c(v_\ell)_i \rangle}\right]^+ \\
&\leq \left[\frac{\langle z_\ell, c(v_\ell)_i \rangle}{\langle y_\ell, c(v_\ell)_i \rangle}\right]^+ + \sum_{k\neq i}^{r_\ell}\left[\frac{\langle z_\ell, c(v_\ell)_k \rangle}{\langle y_\ell, c(v_\ell)_i \rangle}\right]^+ + \sum_{j\neq \ell}^{p}\sum_{k=1}^{r_j}\left[\frac{\langle z_j, c(v_j)_k \rangle}{\langle y_\ell, c(v_\ell)_i \rangle}\right]^+ \\
&= \frac{1}{\langle y_\ell, c(v_\ell)_i \rangle}\left(\sum_{k=1}^{r_\ell}[\langle z_\ell, c(v_\ell)_k \rangle]^+ + \sum_{j\neq \ell}^{p}\sum_{k=1}^{r_j}[\langle z_j, c(v_j)_k \rangle]^+\right) \\
&\leq \frac{1}{\langle y_\ell, c(v_\ell)_i \rangle}\left(\sum_{k=1}^{r_\ell}\langle \mathcal{P}_{\mathcal{K}_\ell}(z_\ell), c(v_\ell)_k \rangle + \sum_{j\neq \ell}^{p}\sum_{k=1}^{r_j}\langle \mathcal{P}_{\mathcal{K}_j}(z_j), c(v_j)_k \rangle\right) \quad \text{(by Lemma 4.2)} \\
&= \frac{1}{\langle y_\ell, c(v_\ell)_i \rangle}\left(\langle \mathcal{P}_{\mathcal{K}_\ell}(z_\ell), e_\ell \rangle + \sum_{j\neq \ell}^{p}\langle \mathcal{P}_{\mathcal{K}_j}(z_j), e_j \rangle\right) \\
&= \frac{\langle \mathcal{P}_{\mathcal{K}}(z), e \rangle}{\langle y_\ell, c(v_\ell)_i \rangle} = \frac{1}{\langle y_\ell, c(v_\ell)_i \rangle}\|\mathcal{P}_{\mathcal{K}}(z)\|_1 \leq \frac{1}{\langle y_\ell, c(v_\ell)_i \rangle}\|\mathcal{P}_{\mathcal{K}}(z)\|_J \leq \frac{1}{\langle y_\ell, c(v_\ell)_i \rangle}\|z\|_J.
\end{aligned}$$

□

Proposition 4.4. *Let $r_{\max} = \max\{r_1, \ldots, r_p\}$. The basic procedure Algorithm 1 terminates in at most $\frac{p^2 r_{\max}^2}{\xi^2}$ iterations.*

Proof. Suppose that y^k is obtained at the k-th iteration of Algorithm 1. Proposition 4.1 implies that $\|y^k\|_{1,\infty} \geq \frac{1}{p}$ and an ℓ-th block element exists for which $\langle y_\ell, e_\ell \rangle \geq \frac{1}{p}$ holds. Thus, by letting $v^k = y^k - z^k$ and the ℓ-th block element v_ℓ^k of v^k be $v_\ell^k = \sum_{i=1}^{r_\ell} \lambda(v_\ell^k)_i c(v_\ell^k)_i$ as in Proposition 2.1, we have

$$\max_{i=1,\ldots,r_\ell} \langle y_\ell^k, c(v_\ell^k)_i \rangle \geq \frac{1}{pr_\ell}. \tag{17}$$

Since Proposition 4.1 ensures that $\frac{1}{\|z^k\|_J^2} \geq k$ holds at the k-th iteration, by setting $k = \frac{p^2 r_{\max}^2}{\xi^2}$, we see that

$$\xi \geq pr_{\max}\|z^k\|_J,$$

and combining this with (17), we have

$$\xi \geq pr_{\max}\|z^k\|_J \geq pr_\ell\|z^k\|_J \geq \frac{1}{\max_{i=1,\dots,r_\ell}\langle y_\ell^k, c(v_\ell)_i\rangle}\|z^k\|_J.$$

The above inequality and Proposition 4.3 imply that for any $\ell \in \{1,2,\dots,p\}$ and $i \in \{1,2,\dots,r_p\}$,

$$\langle c(v_\ell^k)_i, x_\ell\rangle \leq \min_\alpha q_{\ell,i}(\alpha) \leq \frac{1}{\langle y_\ell^k, c(v_\ell^k)_i\rangle}\|z^k\|_J \leq \xi.$$

From the equivalence in (9) and the setting $\xi \in (0,1)$, we conclude that Algorithm 1 terminates in at most $\frac{p^2 r_{\max}^2}{\xi^2}$ iterations by satisfying (13) in the fourth termination condition at an ℓ-th block and an index i. □

Here, we discuss the computational cost per iteration of Algorithm 1. At each iteration of Algorithm 1, the two most expensive operations are computing the spectral decomposition on line 5 and computing $P_{\mathcal{A}}(\cdot)$ on lines 24 and 26.

Let C_ℓ^{sd} be the computational cost of the spectral decomposition of an element of $\mathcal{K}_\ell$. For example, $C_\ell^{\mathrm{sd}} = \mathcal{O}(r_\ell^3)$ if $\mathcal{K}_\ell = \mathbb{S}_+^{r_\ell}$ and $C_\ell^{\mathrm{sd}} = \mathcal{O}(r_\ell)$ if $\mathcal{K}_\ell = \mathbb{L}_{r_\ell}$, where $\mathbb{L}_{r_\ell}$ denotes the r_ℓ-dimensional second-order cone. Then, the cost C^{sd} of computing the spectral decomposition of an element of $\mathcal{K}$ is $C^{\mathrm{sd}} = \sum_{\ell=1}^p C_\ell^{\mathrm{sd}}$. Next, let us consider the computational cost of $P_{\mathcal{A}}(\cdot)$. Recall that d is the dimension of the Euclidean space $\mathbb{E}$ corresponding to $\mathcal{K}$. As discussed in [2], we can compute $P_{\mathcal{A}} = I - \mathcal{A}^*(\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}$ by using the Cholesky decomposition of $(\mathcal{A}\mathcal{A}^*)^{-1}$. Suppose that $(\mathcal{A}\mathcal{A}^*)^{-1} = LL^*$, where L is a $m \times m$ matrix and we store the $L^*\mathcal{A}$ in main algorithm. Then, we can compute $P_{\mathcal{A}}(\cdot)$ on lines 24 and 26, which costs $\mathcal{O}(md)$. From the above discussion and Proposition 4.4, the total cost of Algorithm 1 is given by

$$\mathcal{O}\left(p^2 r_{\max}^2 \max(C^{\mathrm{sd}}, md)\right). \tag{18}$$

Algorithm 1 Basic procedure

Input: $P_{\mathcal{A}}$, $y^1 \in \text{int}\mathcal{K}$ such that $\langle y^1, e\rangle = 1$ and ξ such that $0 < \xi < 1$
Output: a solution to $\text{P}(\mathcal{A})$ or $\text{D}(\mathcal{A})$ or certificates that, for any feasible solution x to $\text{P}_{S_\infty}(\mathcal{A})$, $\langle e, x\rangle < r$
initialization : $k \leftarrow 1, z^1 \leftarrow P_{\mathcal{A}}(y^1), v^1 \leftarrow y^1 - z^1, H_1, \dots, H_p = \emptyset$
while $k \leq \frac{p^2 r_{\max}^2}{\xi^2}$ **do**
  For every $\ell \in \{1, \dots, p\}$, perform spectral decomposition: $z_\ell^k = \sum_{i=1}^{r_\ell} \lambda(z_\ell^k)_i c(z_\ell^k)_i$ and $v_\ell^k = \sum_{i=1}^{r_\ell} \lambda(v_\ell^k)_i c(v_\ell^k)_i$
  if $z^k \in \text{int}\,\mathcal{K}$ **then**
    stop basic procedure and **return** z^k (z^k is a feasible solution of $\text{P}(\mathcal{A})$)
  else if $z^k = 0$ or $v^k \in \mathcal{K} \setminus \{0\}$ **then**
    stop basic procedure and **return** y^k or v^k (y^k or v^k is a feasible solution of $D(\mathcal{A})$)
  end if
  if $\langle v^k, e\rangle > 0$ **then**
    for $\ell \in \{1, \dots, p\}$ **do**
      $I_\ell \leftarrow \left\{i \mid \lambda(v_\ell^k)_i > 0\right\}$ and then $H_\ell \leftarrow \left\{i \in I_\ell \middle| \left\langle e, \mathcal{P}_{\mathcal{K}}\left(-\frac{1}{\lambda(v_\ell^k)_i} v\right)\right\rangle \leq \xi\right\}$
    end for
  else
    for $\ell \in \{1, \dots, p\}$ **do**
      $I_\ell \leftarrow \left\{i \mid \lambda(v_\ell^k)_i < 0\right\}$ and then $H_\ell \leftarrow \left\{i \in I_\ell \middle| \left\langle e, \mathcal{P}_{\mathcal{K}}\left(-\frac{1}{\lambda(v_\ell^k)_i} v\right)\right\rangle \leq \xi\right\}$
    end for
  end if
  if $|H_1| + \dots + |H_p| > 0$ **then**
    For every $\ell \in \{1, \dots, p\}$, let C_ℓ be $\{c(v_\ell^k)_1, \dots, c(v_\ell^k)_{r_\ell}\}$.
    stop basic procedure and **return** $H_1, \dots, H_p$ and $C_1, \dots, C_p$
  end if
  Let u be an idempotent such that $\langle e, u\rangle = 1$ and $\langle z^k, u\rangle = \lambda_{\min}(z^k)$; then $p \leftarrow P_{\mathcal{A}}(u)$
  $y^{k+1} \leftarrow \alpha y^k + (1-\alpha)u$, where $\alpha = \frac{\langle p, p - z^k\rangle}{\|z^k - p\|_J^2}$
  $k \leftarrow k + 1$, $z^k \leftarrow P_{\mathcal{A}}(y^k)$ and $v^k \leftarrow y^k - z^k$
end while
return basic procedure error

5 Main algorithm of the extended method

5.1 Outline of the main algorithm

In what follows, for a given accuracy $\varepsilon > 0$, we call a feasible solution of $\text{P}_{S_\infty}(\mathcal{A})$ whose minimum eigenvalue is ε or more a *ε-feasible solution of* $\text{P}_{S_\infty}(\mathcal{A})$.

Algorithm 2 is a full description of our main algorithm. First, we calculate the corresponding projection $P_{\mathcal{A}}$ onto $\text{Ker}\mathcal{A}$ and generate an initial point as input to the basic procedure. Next, we call the basic procedure and determine whether to end the algorithm with an ε-feasible solution or to perform problem scaling according to the returned result, as follows:

1. If a feasible solution of $\text{P}(\mathcal{A})$ or $\text{D}(\mathcal{A})$ is returned from the basic procedure, the feasibility of $\text{P}(\mathcal{A})$ or $\text{D}(\mathcal{A})$ can be determined, and we stop the main algorithm.

2. If the basic procedure returns the sets of indexes $H_1, \dots, H_p$ and the sets of primitive idempotents $C_1, \dots, C_p$ which construct the corresponding Jordan frames, then

   (a) if $\mathrm{num}_\ell \geq r_\ell \frac{\log \varepsilon}{\log \xi}$ holds for some $\ell \in \{1, \dots p\}$, we determine that $\mathrm{P}_{S_\infty}(\mathcal{A})$ has no ε-feasible solution according to Proposition 5.1 and stop the main algorithm,

   (b) if $\mathrm{num}_\ell < r_\ell \frac{\log \varepsilon}{\log \xi}$ holds for any $\ell \in \{1, \dots p\}$, we rescale the problem and call the basic procedure again.

Note that our main algorithm is similar to Lourenço et al.'s method in the sense that it keeps information about the possible minimum eigenvalue of any feasible solution of the problem. In contrast, Pena and Soheili's method [11] does not keep such information.

Algorithm 2 Main algorithm

1: **Input:** $\mathcal{A}$, $\mathcal{K}$, ε and ξ such that $0 < \xi < 1$
2: **Output:** a solution to $\mathrm{P}(\mathcal{A})$ or $\mathrm{D}(\mathcal{A})$ or a certificate that there is no ε feasible solution.
3: $k \leftarrow 1$, $\mathcal{A}^1 \leftarrow \mathcal{A}$, $\mathrm{num}_\ell \leftarrow 0$, $\bar{Q}_\ell \leftarrow I_\ell$ for all $\ell \in \{1, \dots, p\}$
4: Compute $P_\mathcal{A}$ and call the basic procedure with $P_\mathcal{A}$, $\frac{1}{r}e$, ξ
5: **if** basic procedure returns z **then**
6:   **stop** main algorithm and **return** z (z is a feasible solution of $\mathrm{P}(\mathcal{A})$)
7: **else if** basic procedure returns y or v **then**
8:   **stop** main algorithm and **return** y or v (y or v is a feasible solution of $D(\mathcal{A})$)
9: **else if** basic procedure returns $H_1^k, \dots, H_p^k$ and $C_1^k, \dots, C_p^k$ **then**
10:   **for** $\ell \in \{1, \dots, p\}$ **do**
11:     **if** $|H_\ell^k| > 0$ **then**
12:       $g_\ell \leftarrow \sqrt{\xi} \sum_{h \in H_\ell^k} c^k(v_\ell)_h + \sum_{h \notin H_\ell^k}^{r_\ell} c^k(v_\ell)_h$
13:       $Q_\ell \leftarrow Q_{g_\ell}$
14:       $\mathrm{num}_\ell \leftarrow |H_\ell^k| + \mathrm{num}_\ell$
15:       **if** $\mathrm{num}_\ell \geq r_\ell \frac{\log \varepsilon}{\log \xi}$ **then**
16:         **stop** main algorithm. There is no ε feasible solution.
17:       **end if**
18:       $\bar{Q}_\ell \leftarrow Q_{g_\ell^{-1}} \bar{Q}_\ell$
19:     **else**
20:       $Q_\ell \leftarrow I_\ell$
21:     **end if**
22:   **end for**
23: **else**
24:   **return** basic procedure error
25: **end if**
26: Let $Q^k = (Q_1, \dots, Q_p)$
27: $\mathcal{A}^{k+1} \leftarrow \mathcal{A}^k Q^k$, $k \leftarrow k + 1$. Go back to line 3.

5.2 Finite termination of the main algorithm

Here, we discuss how many iterations are required until we can determine that the minimum eigenvalue $\lambda_{\min}(x)$ is less than ε for any $x \in F_{\mathrm{P}_{S_\infty}(\mathcal{A})}$.

First, we derive an upper bound for the minimum eigenvalue $\lambda_{\min}(x_\ell)$ of each ℓ-th block of x obtained after the k-th iteration of Algorithm 2. Proposition 5.2 gives an upper bound of the number of iterations of Algorithm 2.

Proposition 5.1. *After k iterations of Algorithm 2, for any feasible solution x of $\mathrm{P}_{S_\infty}(\mathcal{A})$ and $\ell \in \{1, \dots, p\}$, the ℓ-th block element x_ℓ of x satisfies*

$$r_\ell \log(\lambda_{\min}(x_\ell)) \leq \mathrm{num}_\ell \log \xi. \tag{19}$$

Proof. In Algorithm 2, the value of num_ℓ is updated only when $|H_\ell^k| > 0$. At the end of the k-th iteration, any feasible solution $\bar{x}$ of the scaled problem $\mathrm{P}_{S_\infty}(\mathcal{A}^{k+1}) = \mathrm{P}_{S_\infty}(\mathcal{A}^k Q^k)$ obviously satisfies

$$\det \bar{x}_\ell \leq \det e_\ell \quad (\ell = 1, 2, \dots, p). \tag{20}$$

Since $\bar{x}$ also satisfies $\bar{x}_\ell = \bar{Q}_\ell(x_\ell)$ $(\ell = 1, 2, \dots, p)$ for each feasible solution x of $\mathrm{P}_{S_\infty}(\mathcal{A})$, we can see that

$$\det \bar{x}_\ell = \det \bar{Q}_\ell(x_\ell) = \left(\frac{1}{\xi}\right)^{|H_\ell^k|} \times \left(\frac{1}{\xi}\right)^{|H_\ell^{k-1}|} \cdots \times \left(\frac{1}{\xi}\right)^{|H_\ell^1|} \times \det x_\ell = \left(\frac{1}{\xi}\right)^{\mathrm{num}_\ell} \det x_\ell.$$

Therefore, (20) implies

$$\det x_\ell \leq \xi^{\mathrm{num}_\ell} \det e_\ell = \xi^{\mathrm{num}_\ell}$$

and the fact $(\lambda_{\min}(x_\ell))^{r_\ell} \leq \det x_\ell$ implies $(\lambda_{\min}(x_\ell))^{r_\ell} \leq \xi^{\mathrm{num}_\ell}$. By taking the logarithm of both sides of this inequality, we obtain (19). □

Proposition 5.2. *Algorithm 2 terminates after no more than*

$$-\frac{r}{\log \xi} \log\left(\frac{1}{\varepsilon}\right) - p + 1.$$

Proof. Let us call the iteration k *good* if $|H_\ell^k| > 0$ $(\ell \in \{1, 2, \dots, p\})$ hold at the k-th iteration of Algorithm 2. Suppose that at least $-\frac{r_\ell}{\log \xi} \log\left(\frac{1}{\varepsilon}\right)$ *good* iterations are observed for a cone $\mathcal{K}_\ell$. Then, by substituting $-\frac{r_\ell}{\log \xi} \log\left(\frac{1}{\varepsilon}\right)$ into num_ℓ of inequality (19) in Proposition 5.1, we have

$$\log(\lambda_{\min}(x_\ell)) \leq -\log\left(\frac{1}{\varepsilon}\right) = \log \varepsilon$$

and hence, $\lambda_{\min}(x_\ell) \leq \varepsilon$. This implies that Algorithm 2 terminates after no more than

$$\sum_{\ell=1}^{p} \left(-\frac{r_\ell}{\log \xi} \log\left(\frac{1}{\varepsilon}\right) - 1\right) + 1 = -\frac{r}{\log \xi} \log\left(\frac{1}{\varepsilon}\right) - p + 1.$$

□

Finally, we consider the computational cost of Algorithm 2. At each iteration of Algorithm 2, the most expensive operation is computing $P_\mathcal{A}$ on line 4. Recall that d is the dimension of the Euclidean space $\mathbb{E}$ corresponding to $\mathcal{K}$. As discussed in [2], by considering $P_\mathcal{A}$ to be an $m \times d$ matrix, we find that the computational cost of $P_\mathcal{A}$ is $\mathcal{O}(m^3 + m^2 d)$. Therefore, by taking the computational cost (18) of the basic procedure and Proposition 5.2 into consideration, the cost of Algorithm 2 turns out to be

$$\mathcal{O}\left(-\frac{r}{\log \xi} \log\left(\frac{1}{\varepsilon}\right) \left(m^3 + m^2 d + \frac{1}{\xi^2} p^2 r_{\max}^2 \left(\max\left(C^{\mathrm{sd}}, md\right)\right)\right)\right) \tag{21}$$

where C^{sd} is the computational cost of the spectral decomposition of $x \in \mathbb{E}$.

Note that, in [2], the authors showed that the cost of their algorithm is

$$\mathcal{O}\left(\left(\frac{r}{\varphi(\rho)}\log\left(\frac{1}{\varepsilon}\right)-\sum_{i=1}^{p}\frac{r_i\log(r_i)}{\varphi(\rho)}\right)\left(m^3+m^2d+\rho^2p^3r_{max}^2\left(\max\left(C^{\min},md\right)\right)\right)\right)$$

where $C^{\min}$ is the cost of computing the minimum eigenvalue of $x\in\mathbb{E}$ with the corresponding idempotent.

When the symmetric cone is simple, by setting $\xi=\frac{1}{2}$ and $\rho=2$, the maximum number of iterations of the basic procedure is bounded by the same value in both algorithms. In accordance with this observation, we will compare the two computational costs (21) and (22) by supposing $\xi=\frac{1}{2}$ and $\rho=2$ (hence, $-\log\xi\simeq 0.69$ and $\varphi(\rho)\simeq 0.09$). As we can see below, the cost (21) of our method is smaller than (22) in the cases of linear programming and second-order cone problems and is equivalent to (22) in the case of semidefinite problems.

First, let us consider the case where $\mathcal{K}$ is the n-dimensional nonnegative orthant $\mathbb{R}^n_+$. Here, we see that $r=p=d=n$, $r_1=\cdots=r_p=r_{\max}=1$, and $\max\left(C^{\rm sd},md\right)=\max\left(C^{\min},md\right)=md$ hold. By substituting these values, the bounds (21) and (22) turn out to be

$$\mathcal{O}\left(\frac{n}{0.69}\log\left(\frac{1}{\varepsilon}\right)\left(m^3+m^2n+4mn^3\right)\right)$$

and

$$\mathcal{O}\left(\frac{n}{0.09}\log\left(\frac{1}{\varepsilon}\right)\left(m^3+m^2n+4mn^4\right)\right).$$

This implies that for the linear programming case, our method (which is equivalent to Roos's original method [13]) is superior to Lourenço et al.'s method [2] in terms of bounds (21) and (22) .

Next, let us consider the case where $\mathcal{K}$ is composed of p simple second-order cones $\mathbb{L}^{n_i}$ $(i=1,\dots,p)$, i.e., $\mathcal{K}=\mathbb{L}^{n_1}\times\mathbb{L}^{n_2}\times\cdots\mathbb{L}^{n_p}$. In this case, we see that $d=\sum_{i=1}^p n_i$, $r_1=\cdots=r_p=r_{\max}=2$ and $\max\left(C^{\rm sd},md\right)=\max\left(C^{\min},md\right)=md$ hold. By substituting these values, the bounds (21) and (22) turn out to be

$$\mathcal{O}\left(\frac{2p}{0.69}\log\left(\frac{1}{\varepsilon}\right)\left(m^3+m^2d+16p^2md\right)\right)$$

and

$$\mathcal{O}\left(\frac{2p}{0.09}\left(\log\left(\frac{1}{\varepsilon}\right)-\log 2\right)\left(m^3+m^2d+16p^3md\right)\right).$$

Note that ε is expected to be very small (10^{-6} or even 10^{-12} in practice) and $\frac{1}{0.69}\log\left(\frac{1}{\varepsilon}\right)\leq\frac{1}{0.09}\left(\log\left(\frac{1}{\varepsilon}\right)-\log 2\right)$ if $\varepsilon\leq 0.451$. Thus, even in this case, we may conclude that our method is superior to Lourenço et al.'s method in terms of the bounds (21) and (22) .

Finally, let us consider the case where $\mathcal{K}$ is a simple $n\times n$ positive semidefinite cone. We see that $p=1$, $r=n$, and $d=\frac{n(n+1)}{2}$ hold, and by substituting these values, the bounds (21) and (22) turn out to be

$$\mathcal{O}\left(\frac{n}{0.69}\log\left(\frac{1}{\varepsilon}\right)\left(m^3+m^2n^2+4n^2\max\left(C^{\rm sd},mn^2\right)\right)\right)$$

and

$$\mathcal{O}\left(\frac{n}{0.09}\log\left(\frac{1}{\varepsilon}\right)\left(m^3+m^2n^2+4n^2\max(C^{\min},mn^2)\right)\right).$$

If $\mathcal{O}\left(C^{\rm sd}\right)=\mathcal{O}\left(C^{\min}\right)$, then the computational bounds of two methods are equivalent.

5.3 Another criteria for ε-feasibility

Proposition 5.1 guarantees that ε-feasibility of the problem $\mathrm{P}(\mathcal{A})$ can be detected by computing $\det(\bar{x})$ of any feasible solution of $\mathrm{P}_{S_\infty}(\mathcal{A}^k Q^k)$. The following proposition, Proposition 5.3, ensures that we may use the value $\langle \bar{x}, e\rangle$ of any feasible solution of $\mathrm{P}_{S_\infty}(\mathcal{A}^k Q^k)$ to detect ε-feasibility of problem $\mathrm{P}(\mathcal{A})$, instead of $\det(\bar{x})$. Algorithm 3 is a modified version of our main algorithm, Algorithm 2, where lines 14 and 15 are replaced by new criteria in accordance with Proposition 5.3. While the analysis of the computational complexity in section 5.2 does not hold for it, the new criteria is better able to detect ε-feasibility in the numerical experiments presented in section6.3.

Proposition 5.3. *After k iterations of Algorithm 3, for any feasible solution x of $\mathrm{P}_{S_\infty}(\mathcal{A})$ and $\ell \in \{1, \dots, p\}$, the ℓ-th block element x_ℓ of x satisfies*

$$\lambda_{\min}(x_\ell) \le \frac{r_\ell}{\left(r_\ell + \left(\frac{1}{\xi} - 1\right) m_\ell\right)}. \tag{22}$$

Proof. In Algorithm 3, m_ℓ is updated only when $|H_\ell^k| > 0$. Suppose that, at the end of the k-th iteration of Algorithm 3, the last update of m_ℓ had been at the $k'(\le k)$-th iteration. Then, the stopping criteria of the basic procedure guarantees that at the beginning of k'-th iteration, $\bar{Q}_\ell$ satisfies

$$\langle x, \bar{Q}_\ell(c^{k'}(v_\ell)_i)\rangle \le \xi$$

for any index $i \in H_\ell^{k'}$. This fact gives a lower bound of $|H_\ell^{k'}|$:

$$\frac{1}{\xi}\left\langle x, \bar{Q}_\ell\left(\sum_{i\in H_\ell^{k'}} c^{k'}(v_\ell)_i\right)\right\rangle \le |H_\ell^{k'}|. \tag{23}$$

Using the fact that $x_\ell - \lambda_{\min}(x_\ell)e_\ell \in \mathcal{K}_\ell$, we obtain

$$\begin{aligned}
\lambda_{\min}(x_\ell)\langle e_\ell, \bar{Q}_\ell(e_\ell)\rangle &\le \langle x_\ell, \bar{Q}_\ell(e_\ell)\rangle \\
&\le r_\ell - |H_\ell^{k'}| + \left\langle x_\ell, \bar{Q}_\ell\left(\sum_{j\in H_\ell^{k'}} c^{k'}(v_\ell)_j\right)\right\rangle \\
&\le r_\ell - \left(\frac{1}{\xi}-1\right)\left\langle x_\ell, \bar{Q}_\ell\left(\sum_{j\in H_\ell^{k'}} c^{k'}(v_\ell)_j\right)\right\rangle \quad \text{(by (23))} \\
&\le r_\ell - \left(\frac{1}{\xi}-1\right)\lambda_{\min}(x_\ell)\left\langle e_\ell, \bar{Q}_\ell\left(\sum_{j\in H_\ell^{k'}} c^{k'}(v_\ell)_j\right)\right\rangle,
\end{aligned}$$

and hence,

$$\lambda_{\min}(x_\ell)\left(\langle e_\ell, \bar{Q}_\ell(e_\ell)\rangle + \left(\frac{1}{\xi}-1\right)\left\langle e_\ell, \bar{Q}_\ell\left(\sum_{j\in H_\ell^{k'}} c^{k'}(v_\ell)_j\right)\right\rangle\right) \le r_\ell. \tag{24}$$

Next, suppose that, at the beginning of the k-th iteration of Algorithm 3, the last update of m_ℓ had been performed at the $i(< k')$-th iteration.

Let $\bar{Q}_\ell{}^{\mathrm{pre}}$ be $\bar{Q}_\ell$ obtained at the beginning of the i-th iteration of Algorithm 3, and let $Q_{g_\ell}^{\mathrm{pre}}$ and m_ℓ^{pre} be Q_ℓ and m_ℓ obtained after the update at the i-th iteration. Note that $\bar{Q}_\ell$ at the beginning of the k'-th iteration of Algorithm 3 can be represented by $\bar{Q}_\ell = \bar{Q}_\ell{}^{\rm pre} Q^{\rm pre}_{g_\ell^{-1}}$. Thus, from (12), we see that

$$\begin{aligned}
Q^{\rm pre}_{g_\ell^{-1}}(e_\ell) &= Q^{\rm pre}_{g_\ell^{-1}}\left(\sum_{j=1}^{r_\ell} c^i(v_\ell)_j\right)\\
&= Q^{\rm pre}_{g_\ell^{-1}}\left(\sum_{j\in H_\ell^{k'}} c^i(v_\ell)_j\right) + Q^{\rm pre}_{g_\ell^{-1}}\left(\sum_{j\notin H_\ell^{k'}} c^i(v_\ell)_j\right)\\
&= \frac{1}{\xi}\sum_{j\in H_\ell^{k'}} c^i(v_\ell)_j + \sum_{j\notin H_\ell^{k'}} c^i(v_\ell)_j\\
&= e_\ell + \left(\frac{1}{\xi}-1\right)\sum_{j\in H_\ell^i} c^i(v_\ell)_j
\end{aligned}$$

and hence,

$$\begin{aligned}
\bar{Q}_\ell(e_\ell) = \bar{Q}_\ell{}^{\rm pre} Q^{\rm pre}_{g_\ell^{-1}}(e_\ell) &= \bar{Q}_\ell{}^{\rm pre}\left(e_\ell + \left(\frac{1}{\xi}-1\right)\sum_{j\in H_\ell^i} c^i(v_\ell)_j\right)\\
&= \bar{Q}_\ell{}^{\rm pre}(e_\ell) + \left(\frac{1}{\xi}-1\right)\bar{Q}_\ell{}^{\rm pre}\left(\sum_{j\in H_\ell^i} c^i(v_\ell)_j\right). \qquad (25)
\end{aligned}$$

By recursively applying (25) to $\bar{Q}_\ell{}^{\rm pre}(e_\ell)$, we finally obtain

$$\left\langle e_\ell, \bar{Q}_\ell(e_\ell)\right\rangle = r_\ell + \left(\frac{1}{\xi}-1\right) m_\ell^{\rm pre}.$$

Let $m_\ell^{k'}$ be the value of m_ℓ obtained after the update at the k'-th iteration. Then,

$$m_\ell^{k'} = m_\ell^{\rm pre} + \left\langle e_\ell, \bar{Q}_\ell\left(\sum_{j\in H_\ell^{k'}} c^{k'}(v_\ell)_j\right)\right\rangle$$

and, by (24), we obtain

$$\begin{aligned}
\lambda_{\min}(x_\ell) &\le \frac{r_\ell}{\left(r_\ell + \left(\frac{1}{\xi}-1\right) m_\ell^{\rm pre} + \left(\frac{1}{\xi}-1\right)\left\langle e_\ell, \bar{Q}_\ell\left(\sum_{j\in H_\ell^{k'}} c^{k'}(v_\ell)_j\right)\right\rangle\right)}\\
&= \frac{r_\ell}{\left(r_\ell + \left(\frac{1}{\xi}-1\right) m_\ell^{k'}\right)}.
\end{aligned}$$

Since, at the end of the k-th iteration of Algorithm 2, the last update of m_ℓ was at the k'-th iteration, we see that $m_\ell = m_\ell^{k'}$, and hence (22) holds after k iterations of Algorithm 3. □

Algorithm 3 Main algorithm using another criteria for ε-feasibility

1: **Input:** $\mathcal{A}$, $\mathcal{K}$, ε and ξ such that $0 < \xi < 1$
2: **Output:** a solution to P($\mathcal{A}$) or D($\mathcal{A}$) or a certificate that there is no ε feasible solution.
3: $k \leftarrow 1$, $\mathcal{A}^1 \leftarrow \mathcal{A}$, $m_\ell \leftarrow 0$, $\bar{Q}_\ell \leftarrow I_\ell$ for all $\ell \in \{1, \dots, p\}$
4: Compute $P_\mathcal{A}$ and call the basic procedure with $P_\mathcal{A}$, ξ , $\frac{1}{r}e$
5: **if** basic procedure returns z **then**
6:   **stop** main algorithm and **return** z (z is a feasible solution of P($\mathcal{A}$))
7: **else if** basic procedure returns y or v **then**
8:   **stop** main algorithm and **return** y or v (y or v is a feasible solution of $D(\mathcal{A})$)
9: **else if** basic procedure returns $H_1^k, \dots, H_p^k$ and $C_1^k, \dots, C_p^k$ **then**
10:   **for** $\ell \in \{1, \dots, p\}$ **do**
11:     **if** $|H_\ell^k| > 0$ **then**
12:       $g_\ell \leftarrow \sqrt{\xi} \sum_{h \in H_\ell^k} c^k(v_\ell)_h + \sum_{h \notin H_\ell^k}^{r_\ell} c^k(v_\ell)_h$
13:       $Q_\ell \leftarrow Q_{g_\ell}$
14:       $m_\ell \leftarrow \left\langle \bar{Q}_\ell \left(\sum_{h \in H_\ell^k} c^k(v_\ell)_h \right), e_\ell \right\rangle + m_\ell$
15:       **if** $\frac{r_\ell}{(r_\ell + (\frac{1}{\xi} - 1) m_\ell)} \leq \varepsilon$ **then**
16:         **stop** main algorithm. There is no ε feasible solution.
17:       **end if**
18:       $\bar{Q}_\ell \leftarrow \bar{Q}_\ell Q_{g_\ell^{-1}}$
19:     **else**
20:       $Q_\ell \leftarrow I_\ell$
21:     **end if**
22:   **end for**
23: **else**
24:   **return** basic procedure error
25: **end if**
26: Let $Q^k = (Q_1, \dots, Q_p)$
27: $\mathcal{A}^{k+1} \leftarrow \mathcal{A}^k Q^k$, $k \leftarrow k + 1$. Go back to line 3.

6 Numerical experiments

6.1 Outline of numerical implementation

Numerical experiments were performed on a positive semidefinite optimization problem with one positive semidefinite cone $\mathcal{K} = \mathbb{S}_+^n$ of the form

$$\begin{array}{lll} \mathrm{P}(\mathcal{A}) & \text{find} & X \\ & \text{s.t.} & \mathcal{A}(X) = \mathbf{0} \in \mathbb{R}^m \\ & & X \in \mathbb{S}_{++}^n \end{array}$$

where $\mathbb{S}_{++}^n$ denotes the interior of $\mathcal{K} = \mathbb{S}_+^n$. We created instances of the following three types:

- Strongly feasible instances, i.e., $\mathrm{Ker}\mathcal{A} \cap \mathbb{S}_{++}^n \neq \emptyset$.
- Weakly feasible instances, i.e., $\mathrm{Ker}\mathcal{A} \cap \mathbb{S}_{++}^n = \emptyset$, but $\mathrm{Ker}\mathcal{A} \cap \mathbb{S}_+^n \setminus \{O\} \neq \emptyset$.
- Infeasible instances, i.e., $\mathrm{Ker}\mathcal{A} \cap \mathbb{S}_+^n = O$.

We will explain how to make each type of instance in section 6.2.

In what follows, we refer to Lourenço et al.'s method [2] as Lourenço (2019), and Pena and Soheili's method [11] as Pena (2017).

We set the termination parameter as $\xi = \frac{1}{4}$ in our basic procedure, i.e., Algorithm 1. We also set the accuracy parameter as ε = 1e-12, both in our main algorithm, Algorithm 3, and in Lourenço (2019) and determined whether $\mathrm{P}_{S_\infty}(\mathcal{A})$ or $\mathrm{P}_{S_1}(\mathcal{A})$ has a solution whose minimum eigenvalue is greater than or equal to ε. Here, we call a solution whose minimum eigenvalue is ε or more a *ε-feasible solution*.

Note that [11] proposed various update methods for the basic procedure. In our numerical experiments, all methods employed the modified basic procedure (Algorithm A) with the identity matrix as the initial point in order to properly compare their performances. This implies that the basic procedures used in the three methods differ only in the termination conditions for moving to the main algorithm and that all other steps are the same.

For each method, we observed the total number of iterations of the basic procedure, the number of iterations of the main algorithm, and the total CPU time. We also examined the violation degrees of the output result, as defined below, and the residual of the constraints for the output result.

First, we classified the output results into five types: A: an interior feasible solution is found; B: no interior feasible solution is found (ver.1); C: no ε-feasible solution is found (only for Lorenço (2019) and our method); D: no interior feasible solution is found (ver.2; only for Pena (2017)); E: Out-of-time. In what follows, we briefly explain how output result type D for Pena (2017) differs from output result type B.

[11] pointed out that if $\mathrm{P}(\mathcal{A})$ has no interior feasible solution, meaning that if the main algorithm of Pena (2017) is applied to only $\mathrm{P}(\mathcal{A})$, it does not stop within a finite number of iterations. To overcome this problem, Pena et al. constructed the main algorithm in a way that it applies not only to $\mathrm{P}(\mathcal{A})$ but also to problem $\mathrm{Q}(\mathcal{A})$:

$$\begin{array}{lll} \mathrm{Q}(\mathcal{A}) & \text{find} & X \\ & \text{s.t.} & X \in \mathrm{Im}\mathcal{A}^T, \\ & & X \in \mathbb{S}^n_{++}. \end{array}$$

Accordingly, we defined output result type B as the case where a feasible solution of $\mathrm{D}(\mathcal{A})$ is obtained by applying the main algorithm to $\mathrm{P}(\mathcal{A})$ and defined output result type D as the case where a feasible solution of $\mathrm{Q}(\mathcal{A})$ is obtained by applying the main algorithm to $\mathrm{Q}(\mathcal{A})$. For each output result type of A, B and D, we defined the violation degree of the output result as follows:

- For output result type A, the violation degree is the number of eigenvalues of the output solution scaled to the solution of $\mathrm{P}(\mathcal{A})$ (i.e., the solution of the problem before scaling), whose value is less than or equal to ε.
- For output result type B, the violation degree is the number of eigenvalues of the output solution scaled to the solution of $\mathrm{D}(\mathcal{A})$ (i.e., the solution of the problem before scaling), whose value is less than or equal to 0.
- For output result type D, the violation degree is the number of eigenvalues of the output solution scaled to the solution of $\mathrm{Q}(\mathcal{A})$ (i.e., the solution of the problem before scaling), whose value is less than or equal to ε.

Morever, when output result type A is obtained, we define the residual of the constraints as the value of $\|\mathcal{A}(X)\|_2$ of the output solution X scaled to the solution of the problem before scaling.

All executions were performed using MATLAB R2018b on an Intel (R) Core (TM) i7-6700 CPU @ 3.40GHz 3.41GHz machine with 16GB of RAM.

6.2 How to generate instances

Here, we describe how the strongly feasible instances, weakly feasible instances, and infeasible instances were generated.

In what follows, for any natural numbers m, n, rand(n) is a function that returns n-dimensional real vectors whose elements are uniformly distributed in the open segment $(0, 1)$, and rand(m, n) is a function that returns $m \times n$ real matrix whose elements are uniformly distributed in the open segment $(0, 1)$. Furthermore, for any $x \in \mathbb{R}^n$ and $X \in \mathbb{R}^{m\times n}$, diag$(x) \in \mathbb{R}^{n\times n}$ is a function that returns a diagonal matrix whose diagonal elements are the elements of x, and vec$(X) \in \mathbb{R}^{mn}$ is a function that returns a vector obtained by stacking the n column vectors of X.

6.2.1 Strongly feasible instances

The strongly feasible instances were generated by extending the method of generating ill-conditioned strongly feasible instances proposed in [12] to the symmetric cone case.

Proposition 6.1. *Suppose that $\bar{x} \in \text{int}\mathcal{K}$, $\|\bar{x}\|_\infty \leq 1$ and $\bar{u} \in \mathcal{K}, \|\bar{u}\|_1 = r$ satisfy $\langle \bar{x}, \bar{u}\rangle = r$. Define the linear operator $\mathcal{A} : \mathbb{E} \to \mathbb{R}^m$ as $\mathcal{A}(x) = (\langle a_1, x\rangle, \langle a_2, x\rangle, \dots, \langle a_m, x\rangle)^T$ for which $a_1 = \bar{u} - \bar{x}^{-1}$ and $\langle a_j, \bar{x}\rangle = 0$ hold for any $j = 2, \dots, m$. Then,*

$$\bar{x} = \arg\max_x \{\det(x) : x \in \mathcal{K} \cap \ker\mathcal{A}, \|x\|_\infty = 1\}. \tag{26}$$

Proof. First, note that the assertion (26) is equivalent to

$$\bar{x} = \arg\max_{x\in\mathcal{F}} \{\log\det(x)\} \text{ where } \mathcal{F} := \{x \in \mathcal{K} \cap \ker\mathcal{A} : \|x\|_\infty \leq 1\}. \tag{27}$$

From the assumptions, we see that $\bar{x} \in \mathcal{K}$, $\|\bar{x}\|_\infty \leq 1$ and $\langle a_1, \bar{x}\rangle = \langle \bar{u} - \bar{x}^{-1}, \bar{x}\rangle = r - r = 0$; thus, $\mathcal{A}(\bar{x}) = 0$ and $\bar{x} \in \mathcal{F}$. Since $\nabla \log\det(x) = x^{-1}$, if $\bar{x}$ satisfies

$$\langle x - \bar{x}, \bar{x}^{-1}\rangle \leq 0 \text{ for any } x \in \mathcal{F} \tag{28}$$

we can conclude that (27) holds. In what follows, we show that (28) holds.

For any $x \in \mathcal{F}$, $x \in \ker\mathcal{A}$ and hence, $\langle a_1, x\rangle = \langle \bar{u} - \bar{x}^{-1}, x\rangle = \langle \bar{u} - \bar{x}^{-1}, x\rangle = 0$, i.e., $\langle \bar{u}, x\rangle = \langle \bar{x}^{-1}, x\rangle$. Thus, we obtain

$$\begin{aligned}\langle x - \bar{x}, \bar{x}^{-1}\rangle &= \langle \bar{u}, x\rangle - r\\ &\leq \langle \bar{u}, x\rangle - \|\bar{u}\|_1\|x\|_\infty \qquad (\text{by } \|\bar{u}\|_1 = r \text{ and } \|x\|_\infty \leq 1)\\ &\leq 0 \qquad (\text{by } \langle \bar{u}, x\rangle \leq \|\bar{u}\|_1\|x\|_\infty)\end{aligned}$$

which completes the proof. □

Proposition 6.1 guarantees that we can generate a linear operator $\mathcal{A}$ satisfying Ker$\mathcal{A} \cap \mathbb{S}^n_{++} \neq \emptyset$ by determining an appropriate value $\mu = \max_{X\in\mathcal{F}} \det(X)$, where $\mathcal{F} = \{X \in \mathbb{S}^n : X \in \mathbb{S}^n_{++} \cap \text{Ker}\mathcal{A}, \|X\|_\infty = 1\}$.

The details on how to generate the strongly feasible instances are in Algorithm 4. The input of Algorithm 4 consists of the rank of the semidefinite cone n and the number of constraints m, an arbitrary orthogonal matrix P, and the parameters $l, u \in \mathbb{R}_{++}$ which determine the value of μ, where $l \leq u$. We set (l, u) as

$$(l, u) \in \{(1e-50, 1e-49), (1e-100, 1e-99), (1e-150, 1e-149), (1e-200, 1e-199), (1e-250, 1e-249)\},$$

so that μ would vary around 1e-50, 1e-100, 1e-150, 1e-200, and 1e-250; i.e., the instances are strongly feasible, but ill-conditioned.

Algorithm 4 Strongly feasible instance

1: **Input:** n, m, l, u, P
2: **Output:** A
3: $d_1 \leftarrow 1$
4: **for** $i = 2$ to n **do**
5: $\quad d_i \leftarrow (l + (u - l)\,\text{rand}\,(1))^{\frac{1}{n-1}}$
6: **end for**
7: $D' \leftarrow \text{diag}(d)$ and then compute $C \leftarrow PD'P^T$ and $c \leftarrow \text{vec}(C)$
8: $u \leftarrow (n, 0_{n-1}^T)^T$ where 0_{n-1} denotes the $n-1$-dimensional vector of zeros
9: $U \leftarrow P(\text{diag}(u) - {D'}^{-1})P^T$
10: $A' \leftarrow \text{vec}(U)$
11: $R \leftarrow I - \frac{1}{\|c\|_2^2} cc^T$
12: **for** $i = 1$ to $m - 1$ **do**
13: $\quad A_i' \leftarrow \text{rand}(n, n)$ and $A_i \leftarrow \left(A_i' + (A_i')^T\right)/2$
14: $\quad A' \leftarrow \begin{pmatrix} A' \\ \text{vec}(A_i)^T \end{pmatrix}$
15: **end for**
16: $\bar{A} \leftarrow A'R$
17: $A \leftarrow \begin{pmatrix} \text{vec}(U)^T \\ \bar{A} \end{pmatrix}$

Proposition 6.2. *For any $A \in \mathbb{R}^{m \times n^2}$ returned from Algorithm 4, there exists $X \in \mathbb{S}_{++}^n$ satisfying $A\,(vec(X)) = \mathbf{0}$.*

Proof. We see that the matrix $C \in \mathbb{S}_{++}^n$ computed on line 7 of Algorithm 4 satisfies

$$A\text{vec}(C) = Ac = \begin{pmatrix} \text{vec}(U)^T c \\ \bar{A}c \end{pmatrix} = \begin{pmatrix} n - n \\ A'Rc \end{pmatrix} = \mathbf{0}.$$

□

6.2.2 Weakly feasible instances

The weakly feasible instances were generated by Algorithm 5.

Algorithm 5 Weakly feasible instance

1: **Input:** $n, m, A' = [\ ]$
2: **Output:** A
3: $B \leftarrow \mathrm{rand}(n, n)$ // B must not be O
4: $C \leftarrow \frac{B+B^T}{2}$ // $C \neq O$ must not be $C \succeq O$ or $C \preceq O$
5: $C_+ \leftarrow \mathcal{P}_{\mathbb{S}^n_+}(C)$ // $C_+ \neq O$ since $C \neq O$ is not negative semidefinite.
6: $C_- \leftarrow -\mathcal{P}_{\mathbb{S}^n_+}(-C)$ // $C_- \neq O$ since $C \neq O$ is not positive semidefinite.
7: $c_+ \leftarrow \mathrm{vec}(C_+)$ and $R \leftarrow I - \frac{1}{\|c_+\|_2^2} c_+ c_+^T$
8: **for** $i = 1$ to $m - 1$ **do**
9: $\quad A'_i \leftarrow \mathrm{rand}(n, n)$ and $A_i \leftarrow \left(A'_i + (A'_i)^T\right)/2$
10: $\quad A' \leftarrow \begin{pmatrix} A' \\ \mathrm{vec}(A_i)^T \end{pmatrix}$
11: **end for**
12: $A \leftarrow \begin{pmatrix} \mathrm{vec}(C_-)^T \\ A'R \end{pmatrix}$

Proposition 6.3. *For any $A \in \mathbb{R}^{m \times n^2}$ returned by Algorithm 5, no $X \in \mathbb{S}^n_{++}$ exists that satisfies $A(\mathrm{vec}(X)) = \mathbf{0}$, but an $X \in \mathbb{S}^n_+ \setminus \{O\}$ exists that satisfies $A(\mathrm{vec}(X)) = \mathbf{0}$.*

Proof. We first show that an $X \in \mathbb{S}^n_+ \setminus \{O\}$ exists that satisfies $A(\mathrm{vec}(X)) = \mathbf{0}$. For the matrix $C_+ \in \mathbb{S}^n_+$ computed on line 5 of Algorithm 5, we see that $C_+ \neq O$ and the following holds:

$$A(\mathrm{vec}(C_+)) = Ac_+ = \begin{pmatrix} \mathrm{vec}(C_-)^T \\ A'R \end{pmatrix} c_+ = \begin{pmatrix} \mathrm{vec}(C_-)^T c_+ \\ A'Rc_+ \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ A'(c_+ - c_+) \end{pmatrix} = \mathbf{0}.$$

Next, we show by contradiction that no $X \in \mathbb{S}^n_{++}$ exists that satisfies $A(\mathrm{vec}(X)) = \mathbf{0}$. Suppose that an $X \in \mathbb{S}^n_{++}$ satisfies $A(\mathrm{vec}(X)) = \mathbf{0}$. Since the first row of A is $\mathrm{vec}(C_-)^T$, if $A(\mathrm{vec}(X)) = \mathbf{0}$ holds, then $\mathrm{vec}(C_-)^T \mathrm{vec}(X) = 0$, i.e.,

$$\begin{aligned} \mathrm{vec}(C_-)^T \mathrm{vec}(X) &= \langle C_-, X \rangle = \langle PDP^T, QEQ^T \rangle \\ &= \langle D, P^T QEQ^T P \rangle = \sum_{i=1}^n D_{ii} \left(P^T QEQ^T P\right)_{ii} = 0 \end{aligned}$$

where $C_- = PDP^T$, $X = QEQ^T$, P are Q orthogonal matrices, and D and E are diagonal matrices. Hrer, $X \in \mathbb{S}^n_{++}$ implies $\left(P^T QEQ^T P\right)_{ii} > 0$ for any $i \in \{1, \ldots, n\}$ and hence, D should be O so that $\sum_{i=1}^n D_{ii} \left(P^T QEQ^T P\right)_{ii} = 0$, but this contradicts to $C_- \neq O$. Thus, there exists no $X \in \mathbb{S}^n_{++}$ satisfying $A(\mathrm{vec}(X)) = \mathbf{0}$. □

6.2.3 Infeasible instances

The infeasible instances were generated by Algorithm 6.

If we define the linear operator $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ as $\mathcal{A}(X) = (\langle A_1, X \rangle, \ldots, \langle A_m, X \rangle)^T$, then by choosing $A_1 \in \mathbb{S}^n_{++}$, we obtain $\mathcal{A}$ such that $\mathrm{Ker}\mathcal{A} \cap \mathbb{S}^n_+ = \{O\}$. On the basis of this observation, by introducing a parameter $\alpha > 0$, we generated a positive definite matrix A_1 whose minimum eigenvalue is a uniformly distributed random number in $(0, \alpha)$. We chose $\alpha \in \{1e-1, 1e-2, 1e-3, 1e-4, 1e-5\}$. The input of Algorithm 6 consisted of the rank of the semidefinite cone n, the number of constraints m, an arbitrary orthogonal matrix P, and the parameter $\alpha > 0$.

Algorithm 6 Infeasible instance

Input: $n, m, \alpha, P,\ A' = [\ \ ]$
Output: A
$B \leftarrow \mathrm{rand}(n, n)$
$B' \leftarrow \frac{B+B^T}{2}$ and then compute orthogonal matrix Q and diagonal matrix E such that $B' = QDQ^T$
$E_+ = \mathrm{rand}\,(1) \times \alpha I + \mathcal{P}_{\mathbb{S}^n_+}(E)$
$d \leftarrow \mathrm{rand}(n)$ and $D \leftarrow \mathrm{diag}(d)$
$B_+ \leftarrow QE_+Q^T$ and $C \leftarrow PDP^T$
$c = \mathrm{vec}(C)$ and $R \leftarrow I - \frac{1}{\|c\|_2^2}cc^T$
for $i = 1$ to $m-1$ **do**
    $A'_i \leftarrow \mathrm{rand}(n, n)$ and $A_i \leftarrow \left(A'_i + (A'_i)^T\right)/2$
    $A' \leftarrow \begin{pmatrix} A' \\ \mathrm{vec}(A_i)^T \end{pmatrix}$
end for
$A \leftarrow \begin{pmatrix} \mathrm{vec}(B_+)^T \\ A'R \end{pmatrix}$

Note that the first row of the matrix A returned by Algorithm 6 is $\mathrm{vec}(B_+)^T$. Since $B_+ \in \mathbb{S}^n_{++}$, we see that $\mathrm{vec}(B_+)^T\mathrm{vec}(X) > 0$ for any positive definite matrix $X \in \mathbb{S}^n_{++}$. Thus, there is no $X \in \mathbb{S}^n_{++}$ satisfying $A\,(\mathrm{vec}(X)) = \mathbf{0}$, which implies that the generated instance is infeasible.

6.3 Numerical results and observations

We set the size of the positive semidefinite matrix to $n = 50$ and the number of constraints m to an integer obtained by rounding the value of $\frac{n(n+1)}{2}\nu$, where $\nu \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

For each $\nu \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, we generated five instances, i.e., 25 instances for each of five strongly feasible cases (corresponding to five patterns of $\mu \simeq$ 1e-50, $\ldots$, $\mu \simeq$ 1e-250, see section 6.2.1 for details), 25 instances for a weakly infeasible case, and 25 instances for each of five infeasible cases (corresponding to five patterns of $\alpha =$ 1e-1, $\ldots$, $\alpha =$ 1e-5, see section 6.2.3 for details). Thus, we generated 125 strongly feasible instances, 25 weakly feasible instances, and 125 infeasible instances, totalling 275 instances. We set the upper limit of the execution time to 2 hours and compared the performance of our method (Algorithms 1 and 2) with those of Lourenço (2019) and Pena(2017).

Table 1 shows the results for the (ill-conditioned) strongly feasible case. The "CO-ratio" column shows the ratio of correct outputs, the "M-iter" column shows the average iteration number of each main algorithm, and the "times(s)" column shows the average CPU time of the method. The values in parentheses () in row $\mu \approx$ 1e-200 are the average CPU times of each method excluding instances for which the method ended up running out of time.

For $\mu \simeq$ 1e-50, 1e-100, there was no significant difference in performance among the three methods. For $\mu \simeq$ 1e-200, 1e-250, our method and Lourenço (2019) obtained interior feasible solutions for all problems, while Pena (2017) ended up running out of time for 10 instances (resp. 25 instances) for $\mu \simeq$ 1e-200 (resp. $\mu \simeq$ 1e-250). This is because Pena (2017) needs to call its basic procedure to find a solution of $\mathrm{Im}\mathcal{A}^T \cap \mathbb{S}^n_{++}$. Comparing our method with Lourenço (2019), we see that it is superior in terms of CPU time. This is probably because it employs a more efficient scaling at each iteration, which will be described in detail in section 7.

Table 1: Results for ill-conditioned strongly feasible instances

	Algorithms 1 and 2			Lourenço (2019)			Pena (2017)		
Instance	CO-ratio	M-iter	Time(s)	CO-ratio	M-iter	time(s)	CO-ratio	M-iter	time(s)
$\mu \simeq$ 1e-50	25/25	1	0.97	25/25	1	0.82	25/25	1	1.04
$\mu \simeq$ 1e-100	25/25	1	1.22	25/25	1	1.20	25/25	1	1.51
$\mu \simeq$ 1e-150	25/25	3	2.27	25/25	306.28	286.85	25/25	1	68.45
$\mu \simeq$ 1e-200	25/25	8.6	9.39(4.38)	25/25	768.28	667.00(802.5)	15/15	1	(5342.74)
$\mu \simeq$ 1e-250	25/25	6	4.95	25/25	1349.48	1164.32	0/0		

Table 2 summarizes the results for infeasible instances. Whereas our method and Lourenço (2019) found an element of $\mathrm{Im}\mathcal{A}^T \cap \mathbb{S}^n_+$ for all instances, Pena (2017) ended up running out of time for one instance for $\alpha =$ 1e-4 and $\alpha =$ 1e-5. Similarly to Table 1, the "CO-ratio, " "M-iter, " and "times(s)" columns respectively show the ratio of correct outputs, the average iteration number of each main algorithm, and the average CPU time of each method (the values in parentheses () in rows $\alpha =$ 1e-4 and $\alpha =$ 1e-5 are the average CPU times of each method excluding the instances for which the method ended up running out of time).

Our method obtained correct outputs for every instance in a short CPU time. This would be because it employed an efficient scaling and found an element of $\mathrm{Im}\mathcal{A}^T \cap \mathbb{S}^n_+$. Pena (2017) ended up running out of time for some instances, but its average CPU time was shorter than that of Lourenço (2019). This would be because Pena (2017) needed no scaling to find an element of $\mathrm{Im}\mathcal{A}^T \cap \mathbb{S}^n_+$ or found the element during the basic procedure for finding an element of $\mathrm{Im}\mathcal{A}^T \cap \mathbb{S}^n_{++}$ for some instances.

Table 2: Results for infeasible instances

	Algorithms 1 and 2			Lourenço (2019)			Pena (2017)		
Instance	CO-ratio	M-iter	Time(s)	CO-ratio	M-iter	time(s)	CO-ratio	M-iter	time(s)
$\alpha =$ 1e-1	25/25	1.96	1.23	25/25	3.6	2.37	25/25	1	0.79
$\alpha =$ 1e-2	25/25	5.36	4.39	25/25	29.04	37.93	25/25	1	25.99
$\alpha =$ 1e-3	25/25	7.16	5.38	25/25	53	61.61	25/25	1.08	61.55
$\alpha =$ 1e-4	25/25	9.32	7.81(3.16)	25/25	73.04	88.32(37.18)	24/25	1.52	(20.80)
$\alpha =$ 1e-5	25/25	10.04	9.08(2.98)	25/25	60.96	76.17(29.55)	24/25	1.44	(9.47)

For weakly infeasible instances, we compared our method (Algorithms 1 and 2), a modified version with another criteria for ε-feasibility (Algorithms 1 and 3), Lourenço (2019), and Pena (2017). The results are summarized in Table 3.

As described in section 6.1, we classified the output results into type A: an interior feasible solution is found; type B: no interior feasible solution is found (ver.1); type C: no ε-feasible solution is found (only for Lorenço (2019) and our methods); type D: no interior feasible solution is found (ver.2; only for Pena (2017)); type E: Out-of-time.

Note that B* indicates that the output was B, but when we converted the obtained solution to a solution of D($\mathcal{A}$), it contained a negative eigenvalue and violated the SDP constraint.

From Table 3, we can observe the following:

- For all instances, Pena (2017) exceeded 2 hours and E (Out-of-time). This would be largely because Pena (2017) needs to apply the algorithm to two problems P($\mathcal{A}$) and Q($\mathcal{A}$).
- Our method (Algorithms 1 and 2) and Lourenço (2019) sometimes obtained output type A (an interior feasible solution is found) , while the obtained solution had $0 \sim 5$ negative eigenvalues (about -1e-16) and more than 20 positive eigenvalues (less than 1e-12) when we converted it into a solution of P($\mathcal{A}$).
- Lourenço (2019) obtained output type B* (no interior feasible solution is found) but when we converted the obtained solution into a solution of D($\mathcal{A}$), it contained a negative eigenvalue and violated the SDP constraint). The obtained solution had $1 \sim 3$ negative eigenvalues (about -1e-6) and violated the SDP constraint when we converted it into a solution of D($\mathcal{A}$).
- Our modified method (Algorithms 1 and 3) was able to determine the existence of an ε-feasible solution for all instances. This implies that the criteria focusing on the total value of the eigenvalues used in Algorithm 3 is more suitable than the criteria focusing on the product of all the eigenvalues.

Table 3: Output types for weakly feasible instances

Method \ Value of ν	$\nu = 0.1$	$\nu = 0.3$	$\nu = 0.5$	$\nu = 0.7$	$\nu = 0.9$
Algorithms 1 and 2	AAAAA	AAAAA	AAAAA	AAAAA	BBBBB
Algorithms 1 and 3	CCCCC	CCCCC	CCCCC	CCCCC	CCCCC
Lourenço (2019)	AAAAA	BB*B*B*B	ABAAA	ABAB*B*	BBBBB
Pena (2017)	EEEEE	EEEEE	EEEEE	EEEEE	EEEEE

Table 4 summarizes the results for weakly feasible instances, where we omitted the column of Pena (2017), since it resulted in E (Out-of-time) for all instances. Both of our methods had a smaller average CPU time than that of Lourenço (2019). This would be largely because, at each iteration, our methods employ a more efficient scaling than that of Lourenço (2019).

Table 4: Results for weakly feasible instances (Pena (2017) resulted in out-of-time for all instances)

Method	Algorithms 1 and 2			Algorithms 1 and 3			Lourenço (2019)		
Value of ν	CO-ratio	M-iter	Time(s)	CO-ratio	M-iter	time(s)	CO-ratio	M-iter	time(s)
$\nu = 0.1$	0/5	547	536.29	5/5	462	459.75	0/5	3048.8	4216.44
$\nu = 0.3$	0/5	573.6	219.19	5/5	478.2	159.19	2/5	3362.4	2484.63
$\nu = 0.5$	0/5	544	352.68	5/5	452.6	224.01	1/5	3298.6	3007.91
$\nu = 0.7$	0/5	533.8	438.51	5/5	438.2	296.03	1/5	3315.4	3583.32
$\nu = 0.9$	5/5	443.2	525.43	5/5	386.6	370.89	5/5	3193.6	4271.58
Total	5/25	528.32	414.42	25/25	443.52	301.97	9/25	3243.76	3512.78

The detailed numerical results are in Appendix B.

7 More comparisons of basic procedures

In section 5, we showed that the bound of the computational cost of our method is superior to the one of Lourenço et al. when $\mathcal{K}$ is the n-dimensional nonnegative orthant $\mathbb{R}^n_+$ or a Cartesian product of simple second-order corns, and that their bounds are equivalent when $\mathcal{K}$ is a simple positive semidefinite cone under the assumption that the costs of computing the spectral decomposition and the minimum eigenvalue are the same for an $n \times n$ symmetric matrix. In this section, we make more detailed comparisons of these algorithms in terms of the possible reduction rate of the search region and the detectability of an ε-feasible solution. Similarly to section 6, we will refer to Lourenço et al.'s method [2] as Lourenço (2019) throughout this section.

7.1 Possible reduction rate of two basic procedures for the simple case

Here, for the sake of simplicity, we will focus on the case where the symmetric cone is simple, i.e., $p = 1$. Let $\mathbb{E}$ be the Euclidean space corresponding to the symmetric cone $\mathcal{K}$. For any given $w, v \in \mathbb{E}$, Lourenço et al. [2] defined $\mathrm{vol}(w, v)$ as the volume of the intersection $H(w, v) \cap \mathcal{K}$, where $H(w, v)$ is the half space given by

$$H(w, v) = \{x \in \mathbb{E} \mid \langle w, x\rangle \leq \langle w, v\rangle\}.$$

On the basis of the discussion in [2], we will observe how $\mathrm{vol}(w, v)$ decreases compared with $\mathrm{vol}(e, e/r)$ (the value before the scaling) in the basic procedures (Algorithm 1 and of Lourenço (2019)).

7.1.1 Possible reduction rate of Algorithm 1

Suppose that Algorithm 1 returns a result such that there exists a nonempty index set $I \subseteq \{1, \dots, r\}$ with $|I| = N$ for which

$$\langle c_i, x\rangle \leq \begin{cases} \xi & i \in I \\ 1 & i \notin I \end{cases} \tag{29}$$

holds for any feasible solution x of $\mathrm{P}_{S_\infty(A)}$, where $\{c_1, \dots, c_r\}$ are primitive idempotents that make up a Jordan frame.

Note that Algorithm 1 employs the scaling $\bar{x} = Q_{g^{-1}}(x)$ with $g^{-1} = \frac{1}{\sqrt{\xi}} \sum_{i \in I} c_i + \sum_{i \notin I} c_i$. Let us find $w, v \in \mathbb{E}$ which satisfy

$$H(e, e/r) = Q_{g^{-1}}\left(H(w, v)\right). \tag{30}$$

Since (30) and the scaling $\bar{x} = Q_{g^{-1}}(x)$ imply that

$$\begin{aligned} H(w, v) &= Q_g\left(H(e, e/r)\right) \\ &= \{Q_g(\bar{x}) \in \mathbb{E} \mid \langle \bar{x}, e\rangle \leq 1\} \\ &= \{Q_g(\bar{x}) \in \mathbb{E} \mid \langle Q_g(\bar{x}), Q_{g^{-1}}(e)\rangle \leq 1\} \\ &= \{x \in \mathbb{E} \mid \langle x, Q_{g^{-1}}(e)\rangle \leq \langle Q_{g^{-1}}(e), Q_g(e)/r\rangle = 1\}, \end{aligned}$$

by setting $w = Q_{g^{-1}}(e)$ and $v = Q_g(e)/r$, we find that the half space $H(w, v)$ is transformed to $H(e, e/r)$ after the scaling. Since $Q_{g^{-1}}(e) \in \mathrm{int}\mathcal{K}$, we can apply the following proposition to $w = Q_{g^{-1}}(e)$.

Proposition 7.1 (Proposition 6 of [2]). *Suppose that $w \in \mathrm{int}\mathcal{K}$. Then,*

$$Q_{w^{-1/2}\sqrt{\langle w, v\rangle}}\left(H(e, e/r)\right) = H(w, v),$$

$$\mathrm{vol}(w, v) = \left(\frac{\langle w, v\rangle}{\sqrt[r]{\det w}}\right)^d \mathrm{vol}(e, e/r).$$

Using the above proposition and the assumption $|I| = N$ for the set I in (29), we can see how the volume $\mathrm{vol}(Q_{g^{-1}}(e), Q_g(e)/r)$ of $H(Q_{g^{-1}}(e), Q_g(e)/r) \cap \mathcal{K}$ decreases compared with $\mathrm{vol}(e, e/r)$ as follows:

$$\begin{aligned}
\mathrm{vol}(Q_{g^{-1}}(e), Q_g(e)/r) &= \left(\frac{1}{\sqrt[r]{\det Q_{g^{-1}}(e)}}\right)^d \mathrm{vol}(e, e/r) \\
&= \left(\frac{1}{\sqrt[r]{\frac{1}{\xi^N}}}\right)^d \mathrm{vol}(e, e/r) \\
&= \left(\xi^N\right)^{\frac{d}{r}} \mathrm{vol}(e, e/r). \qquad (31)
\end{aligned}$$

7.1.2 Possible reduction rate of the basic procedure of Lourenço (2019)

The following theorem gives the possible reduction rate of the basic procedure of Lourenço (2019).

Theorem 7.2 (Theorem 10 of [2]). *Let $\rho > 1$ and $y \in \mathcal{K} \setminus \{0\}$ be such that $F_{\mathrm{P}_{S_1}(A)} \subseteq H(y, e/\rho r)$. Let*

$$\begin{aligned}
\beta &= r - \left(\frac{1}{\rho} - \frac{1}{\sqrt{\rho(3\rho - 2)}}\right), \\
w &= \frac{r - \beta}{\langle y, e\rangle} \rho r y + \beta e, \\
v &= w^{-1}.
\end{aligned}$$

Then, the following hold:

1. $F_{\mathrm{P}_S(A)} \subseteq H(y, e/\rho r) \cap H(e, e/r) \subseteq H(w, v)$
2. $Q_{\sqrt{r} w^{-1/2}} (H(e, e/r)) = H(w, v)$
3. $$\mathrm{vol}(w, v) = \left(\frac{r^r}{\det w}\right)^{\frac{d}{r}} \mathrm{vol}(e, e/r) \leq \left(\exp\left(-\varphi(\rho)\right)\right)^{\frac{d}{r}} \mathrm{vol}(e, e/r)$$

   where $\varphi(\rho) = 2 - \frac{1}{\rho} - \sqrt{3 - \frac{2}{\rho}}$.

   In particular, if $\rho \geq 2$, we have $\mathrm{vol}(w, v) < (0.918)^{\frac{d}{r}} \mathrm{vol}(e, e/r)$.

7.1.3 Comparison of possible reduction rates of two methods

The results in section 7.1.1 and 7.1.2 are summarized in Table 5, where the "UB # of iter." column shows the upper bound of the number of iterations required in the basic procedure. The "Upper bound # of iter." of Lourenço (2019) comes from Proposition 14 of [2] (where the authors showed the result by substituting $\rho = 2$) and the one of Algorithm 1 comes from Proposition 4.4 with $\ell = 1$. The "Possible reduction the search region" of Lourenço (2019) comes from Theorem 7.2 and the one of Algorithm 1 comes from (31) with $(w, v) = (Q_{g^{-1}}(e), Q_g(e)/r)$.

Table 5: Comparison of reduction rates of two algorithms: Theoretical results

Basic procedure	Upper bound # of iter.	Possible reduction of the search region
Lourenço (2019)	$\rho^2 r_{\max}^2$	$\mathrm{vol}(w,v) = \left(\frac{r^r}{\det w}\right)^{\frac{d}{r}} \mathrm{vol}(e, e/r) \le \left(e^{-\varphi(\rho)}\right)^{\frac{d}{r}} \mathrm{vol}(e, e/r)$
Algorithm 1	$\dfrac{r_{\max}^2}{\xi^2}$	$\mathrm{vol}(w,v) = \left(\xi^N\right)^{\frac{d}{r}} \mathrm{vol}(e, e/r)$

By setting $\rho = 2$ and $\xi = \frac{1}{2}$, the two bounds in"Upper bound # of iter." have the same value and in this case, the possible reduction rates turn out to be

$$\text{Lourenço (2019):} \left(\frac{r^r}{\det w}\right)^{\frac{d}{r}} \le \left(e^{-\varphi(2)}\right)^{\frac{d}{r}} \simeq (0.918)^{\frac{d}{r}}, \qquad \text{Algorithm 1:} \left(\xi^N\right)^{\frac{d}{r}} \le \left(\frac{1}{2}\right)^{\frac{d}{r}}.$$

The above comparison indicates that Algorithm 1 is superior to the basic procedure in [2] in terms of the possible reduction rate of the search region. To confirm whether similar reduction rates are observed numerically, we conducted an experiment where we used our method (Algorithms 1 and 2) with $\xi = 1/2$ and Lourenço (2019) with $\rho = 2$ to solve a weakly feasible instance with $\nu = 0.1$. At each iteration of the main algorithms, we recorded the value of $\frac{r^r}{\det w}$ of Lourenço (2019) and the value of ξ^N of our method and computed the reduction rates of the search region. The results are summarized in Table 6.

Table 6: Comparison of reduction rates of two algorithms: Numerical results

Algorithm	# of iter. of M-A	Output	Average reduction rate	Final reduction rate
Lourenço (2019)	3060	A	0.864	3.859e-195
Algorithms 1 and 3	618	C	0.357	9.114e-305

The "Average reduction rate" column shows the average value of $\frac{r^r}{\det w}$ for Lourenço (2019) and the average value of ξ^N for our method (Algorithms 1 and 2). The "Final reduction rate" column shows the value

$$\frac{r^{kr}}{\det w(1) \times \det w(2) \times \cdots \times \det w(k)}$$

for Lourenço (2019), where $w(k)$ denotes w computed from the result of the basic procedure at the k-th iteration of the main algorithm, or the value

$$\xi^{N_1 + \cdots + N_k}.$$

for our method (Algorithms 1 and 2), where N_k denotes the number of cuts obtained from the basic procedure at the k-th iteration of the main algorithm.

Here, we observed that our method (Algorithms 1 and 2) terminated at the 618-th iteration of the main algorithm with a reduction rate of 9.114e-305, while Lourenço (2019) attained a reduction rate of 5.879e-40 at the same iteration of the main algorithm. Thus, our method detected the existence of an ε-feasible solution in fewer iterations than Lourenço (2019) did.

7.2 Detection of an ε-feasible solution

Here, we discuss the capabilities of our method and Lourenço (2019) at detecting an ε-feasible solution. Both methods terminate their main algorithms by detecting the existence of an ε-feasible solution. We compared them by computing the reduction in $\log(\lambda_{\min}(x_\ell))$ per iteration for parameter settings in which the maximum numbers of iterations of the basic procedures would be the same (i.e., $\rho = 2$ in Lourenço (2019) and $\xi = \frac{1}{2}$ in our method).

In [2], for each block ℓ, Lemma 16 ensures that $\log(\lambda_{\min}(x_\ell))$ is bounded from above by ϵ_ℓ, and Theorem 17 ensures that ϵ_ℓ decreases at least $\frac{\varphi(\rho)}{r_\ell} > 0$ if a *good* iteration is obtained for the block ℓ.

For our method, Proposition 5.1 ensures that $\log(\lambda_{\min}(x_\ell))$ is bounded from above by $\frac{\mathrm{num}_\ell}{r_\ell}\log\xi$ and Proposition 5.2 ensures that $\frac{\mathrm{num}_\ell}{r_\ell}\log\xi$ decreases $-\frac{1}{r_\ell}\log\xi > 0$ in the same situation.

By substituting $\rho = 2$ and $\xi = \frac{1}{2}$ into $\varphi(\rho)$ and $-\log\xi$ so that the maximum numbers of iterations of the basic procedures are the same, we obtain

$$\begin{aligned}\varphi(2) &= 2 - \frac{1}{2} - \sqrt{2} \simeq 0.085786,\\ -\log\frac{1}{2} &= \log 2 \simeq 0.693147\end{aligned}$$

which implies that the reduction rate of the upper bound of $\log(\lambda_{\min}(x_\ell))$ of our method is superior to that of Lourenço (2019).

8 Concluding remarks

In this study, we proposed a new version of Chubanov's method for solving the feasibility problem over the symmetric cone by extending Roos's method [13] for the feasible problem over the nonnegative orthant, and we conducted comprehensive numerical experiments on the problem over the positive semidefinite cone to compare the performances of our method and the existing ones [2, 11].

Our method has the following features:

- It considers the feasibility problem $\mathrm{P}_{S_\infty}(\mathcal{A})$, which is equivalent to $\mathrm{P}(\mathcal{A})$, and uses a rescaling focusing on the upper bound of the sum of eigenvalues of any feasible solution of $\mathrm{P}_{S_\infty}(\mathcal{A})$.
- Using the norm $\|\cdot\|_\infty$ in problem $\mathrm{P}_{S_\infty}(\mathcal{A})$ makes it possible to (i) calculate the upper bound of the minimum eigenvalue of any feasible solution of $\mathrm{P}_{S_\infty}(\mathcal{A})$, (ii) quantify the feasible region of $\mathrm{P}(\mathcal{A})$, and hence (iii) determine whether there exists a feasible solution of $\mathrm{P}(\mathcal{A})$ whose minimum eigenvalue is greater than ϵ as in [2].
- The computational bound of our method is (i) equivalent to Roos's original method [13] and superior to Lourenço et al.'s method [2] when the symmetric cone is the nonnegative orthant, (ii) superior to Lourenço et al.'s when the symmetric cone is a Cartesian product of second-order cones, and (iii) equivalent to Lourenço et al.'s when the symmetric cone is the simple positive semidefinite cone, under the assumption that the costs of computing the spectral decomposition and the minimum eigenvalue are of the same order for any given symmetric matrix.

We conducted comprehensive numerical experiments comparing our method with the existing ones. We generated instances in three types: (i) strongly (but ill-conditioned) feasible instances by using Algorithm

4, (ii) weakly feasible instances by using Algorithm 5, and (iii) infeasible instances by using Algorithm 6. Our numerical results showed that

- Our method (Algorithms 1 and 2) is superior to the methods proposed in [2, 11] in terms of accuracy and execution time.
- It is considerably faster than the existing methods on ill-conditioned strongly feasible instances.
- A modified version of our method (Algorithms 1 and 3) can exactly determine whether the instance has no feasible solution whose minimum eigenvalue is less than ε = 1e-12 for all weakly feasible instances (i.e., having no interior feasible solution), which is in contrast to Lourenço et al.'s method, which sometimes returns a solution that does not satisfy the conic constraint of $\mathrm{P}(\mathcal{A})$ or $\mathrm{D}(\mathcal{A})$ and is affected by the large number of iterations of its main algorithm.

On the basis of the above numerical results, we further examined the number of iterations of Lourenço et al.'s method and our method. As a result, we found that the basic procedure of our method is superior to the one of Lourenço et al in terms of both a constant reduction rate of the volume of the detecting region and the upper bound of the minimum eigenvalue of any feasible solution.

Note that Chubanov's method can find an $x \in \mathrm{int}\mathcal{K}$ satisfying $\mathcal{A}(x) = \mathbf{0}$, but not $x \in \mathcal{K}$ close to the boundary and satisfying $\mathcal{A}(x) = \mathbf{0}$, and it can determine the feasibility of $\mathrm{P}(\mathcal{A})$ in a finite number of iterations, but not a feasible solution of $\mathrm{D}(\mathcal{A})$ in such a way.

On the other hand, once we find that $\mathrm{P}_\infty(\mathcal{A})$ and $\mathrm{P}_{1,\infty}(\mathcal{A})$ have no feasible solution whose minimum eigenvalue is greater than ε, the next issue is to find an $x \in \mathcal{K} \setminus \mathrm{int}\mathcal{K}$ satisfying $\mathcal{A}(x) = \mathbf{0}$, or to find the smallest dimensional symmetric cone $\mathcal{K}_{\min}$ satisfying $\mathrm{Ker}\mathcal{A} \cap \mathrm{int}\mathcal{K}_{\min} \neq \emptyset$ and $\mathcal{K}_{\min} \subsetneq \mathcal{K}$. It has been shown that the smallest dimensional symmetric cone $\mathcal{K}_{\min}$ can be detected by using a feasible solution of $\mathrm{D}(\mathcal{A})$ [15], and several algorithms have been proposed to find a feasible solution of $\mathrm{D}(\mathcal{A})$ in a finite number of iterations [5, 10].

It remains as future work to explore whether it is possible to extend Chuvanov's method to find $x \in \mathcal{K}$ close to the boundary and satisfying $\mathcal{A}(x) = \mathbf{0}$ directly or find a feasible solution of $\mathrm{D}(\mathcal{A})$ in a finite number of iterations.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers (B)19H02373 and JP 21J20875.

References

[1] Alizadeh F. (2012) . An introduction to formally real Jordan algebras and their applications in optimization. In: Anjos M., Lasserre J. (eds) Handbook on Semidefinite, Conic and Polynomial Optimization. International Series in Operations Research & Management Science, vol 166. Springer, Boston, MA.

[2] Lourenço, B. F., Kitahara, T., Muramatsu, M., & Tsuchiya, T. (2019). An extension of Chubanov ' s algorithm to symmetric cones. Mathematical Programming, 173(1-2), 117-149.

[3] Chubanov, S. (2012). A strongly polynomial algorithm for linear systems having a binary solution. Mathematical programming, 134(2), 533-570.

[4] Chubanov, S. (2015). A polynomial projection algorithm for linear feasibility problems. Mathematical Programming, 153(2), 687-713.

[5] Chubanov, S. (2017). A polynomial algorithm for linear feasibility problems given by separation oracles. Optimization Online, Jan.

[6] Faraut, J. & Korányi, A. (1994). Analysis on symmetric cones. Oxford University Press, Oxford, UK.

[7] Faybusovich, L. (1997). Euclidean Jordan algebras and interior-point algorithms. Positivity 1, 331–357.

[8] Kitahara, T., & Tsuchiya, T. (2018). An extension of Chubanov's polynomial-time linear programming algorithm to second-order cone programming. Optimization Methods and Software, 33(1), 1-25.

[9] Li, D., Roos, C., & Terlaky, T. (2015). A polynomial column-wise rescaling von Neumann algorithm. Optimization Onine, June.

[10] Muramatsu, M., Kitahara, T., Lourenço, B. F., Okuno, T., & Tsuchiya, T. (2018). An oracle-based projection and rescaling algorithm for linear semi-infinite feasibility problems and its application to SDP and SOCP. arXiv preprint arXiv:1809.10340.

[11] Pena, J., & Soheili, N. (2017). Solving conic systems via projection and rescaling. Mathematical Programming, 166(1-2), 87-111.

[12] Pena, J., & Soheili, N. (2019). Computational performance of a projection and rescaling algorithm. Optimization Methods and Software, 1-18.

[13] Roos, K. (2018). An improved version of Chubanov's method for solving a homogeneous feasibility problem. Optimization Methods and Software, 33(1), 26-44.

[14] Schmieta, S., Alizadeh, F. (2003). Extension of primal-dual interior point algorithms to symmetric cones. Mathematical Programming, Series. A 96, 409–438.

[15] Waki, H., & Muramatsu, M. (2013). Facial reduction algorithms for conic optimization problems. Journal of Optimization Theory and Applications, 158(1), 188-215.

[16] Wei, Z., & Roos, K. (2019). Using Nemirovski ' s Mirror-Prox method as Basic Procedure in Chubanov ' s method for solving homogeneous feasibility problems. Manuscript (http://www. optimization-online. org/DB HTML/2018/04/6559. html).

[17] Yoshise A. (2007) Interior point trajectories and a homogeneous model for nonlinear complementarity problems over symmetric cones. SIAM Journal on Optimization, 17(4), 1129-1153.

A Modified Basic procedure

Algorithm 7 Modified basic procedure

Input: $P_{\mathcal{A}}$, $y^1 \in \text{int}\mathcal{K}$ such that $\langle y^1, e\rangle = 1$ and ξ such that $0 < \xi < 1$
Output: a solution to P($\mathcal{A}$) or D($\mathcal{A}$) or certificates that, for any feasible solution x to $\text{P}_{S_\infty}(\mathcal{A})$, $\langle e, x\rangle < r$
initialization : $k \leftarrow 1, z^1 \leftarrow P_{\mathcal{A}}(y^1), v^1 \leftarrow y^1 - z^1, H_1, \dots, H_p = \emptyset$
while $k \leq \frac{p^2 r_{\max}^2}{\xi^2}$ **do**
  For every $\ell \in \{1, \dots, p\}$, spectral decomposition : $z_\ell^k = \sum_{i=1}^{r_\ell} \lambda(z_\ell^k)_i c(z_\ell^k)_i$ and $v_\ell^k = \sum_{i=1}^{r_\ell} \lambda(v_\ell^k)_i c(v_\ell^k)_i$
  if $z^k \in \text{int } \mathcal{K}$ **then**
    stop basic procedure and **return** z^k (z^k is a feasible solution of P($\mathcal{A}$))
  else if $z^k = 0$ or $v^k \in \mathcal{K} \setminus \{0\}$ **then**
    stop basic procedure and **return** y^k or v^k (y^k or v^k is a feasible solution of $D(\mathcal{A})$)
  end if
  if $\langle v^k, e\rangle > 0$ **then**
    for $\ell \in \{1, \dots, p\}$ **do**
      $I_\ell \leftarrow \left\{i \mid \lambda(v_\ell^k)_i > 0\right\}$ and then $H_\ell \leftarrow \left\{i \in I_\ell \middle| \left\langle e, \mathcal{P}_{\mathcal{K}}\left(-\frac{1}{\lambda(v_\ell^k)_i} v\right)\right\rangle \leq \xi\right\}$
    end for
  else
    for $\ell \in \{1, \dots, p\}$ **do**
      $I_\ell \leftarrow \left\{i \mid \lambda(v_\ell^k)_i < 0\right\}$ and then $H_\ell \leftarrow \left\{i \in I_\ell \middle| \left\langle e, \mathcal{P}_{\mathcal{K}}\left(-\frac{1}{\lambda(v_\ell^k)_i} v\right)\right\rangle \leq \xi\right\}$
    end for
  end if
  if $|H_1| + \cdots + |H_p| > 0$ **then**
    For every $\ell \in \{1, \dots, p\}$, let C_ℓ be $\{c(v_\ell^k)_1, \dots, c(v_\ell^k)_{r_\ell}\}$.
    stop basic procedure and **return** $H_1, \dots, H_p$ and $C_1, \dots, C_p$
  end if
  for $\ell \in \{1, \dots, p\}$ **do**
    $S_\ell \leftarrow \{i \mid \lambda(z_\ell^k)_i \leq 0\}$ and then $u_\ell \leftarrow \sum_{i \in S_\ell} c(z_\ell^k)_i$
  end for
  $u \leftarrow \frac{1}{\sum_{\ell=1}^p |S_\ell|} u$ and $p \leftarrow P_{\mathcal{A}}(u)$
  $y^{k+1} \leftarrow \alpha y^k + (1 - \alpha)u$, where $\alpha = \frac{\langle p, p - z^k\rangle}{\|z^k - p\|_J^2}$
  $k \leftarrow k + 1$, $z^k \leftarrow P_{\mathcal{A}}(y^k)$ and $v^k \leftarrow y^k - z^k$
end while

B Detailed numerical results

Tables 7-17 show the results for the strongly feasible, infeasible, and weakly infeasible cases. In these tables, the "BP" column shows the total number of iterations of the basic procedure, the "MA" column shows the number of iterations of the main algorithm. As described in section 6.1, the violation degree of the output (the columun "VDO") for output type A is the number of negative eigenvalues of the output solution scaled to the solution of P($\mathcal{A}$) (i.e., the solution of the problem before scaling), and the number in parenthesis shows the number of those nonnegative values less than ε.

Table 7: Results for ill-conditioned strongly feasible instances (μ = 1e-50)

Instance	Algorithms 1 and 2						Lourenço (2019)						Pena (2017)					
ν-#	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)
0.1-1	1	1	A	7.64e-15	0	0.9889971	1	1	A	7.64e-15	0	1.0030436	1	1	A	7.64e-15	0	1.1757979
0.1-2	1	1	A	5.70e-15	0	1.0735026	1	1	A	5.70e-15	0	1.0142510	1	1	A	5.70e-15	0	1.1203066
0.1-3	1	1	A	7.38e-15	0	1.1080386	1	1	A	7.38e-15	0	1.0165800	1	1	A	7.38e-15	0	1.1479178
0.1-4	1	1	A	7.13e-15	0	1.1061800	1	1	A	7.13e-15	0	1.0024174	1	1	A	7.13e-15	0	1.1846754
0.1-5	1	1	A	7.36e-15	0	1.0773073	1	1	A	7.36e-15	0	0.9880980	1	1	A	7.36e-15	0	1.1550579
0.3-1	1	1	A	6.93e-15	0	0.6436720	1	1	A	6.93e-15	0	0.4291594	1	1	A	6.93e-15	0	1.1550579
0.3-2	1	1	A	4.69e-15	0	0.6327618	1	1	A	4.69e-15	0	0.4268765	1	1	A	4.69e-15	0	0.6401682
0.3-3	1	1	A	3.13e-15	0	0.5959243	1	1	A	3.13e-15	0	0.4248649	1	1	A	3.13e-15	0	0.6190661
0.3-4	1	1	A	3.15e-15	0	0.5802657	1	1	A	3.15e-15	0	0.4332688	1	1	A	3.15e-15	0	0.6220018
0.3-5	1	1	A	4.96e-15	0	0.6938755	1	1	A	4.96e-15	0	0.4220729	1	1	A	4.96e-15	0	0.6871871
0.5-1	1	1	A	7.98e-15	0	0.8031498	1	1	A	7.98e-15	0	0.6179921	1	1	A	7.98e-15	0	0.8101831
0.5-2	1	1	A	7.69e-15	0	0.8036580	1	1	A	7.69e-15	0	0.6136709	1	1	A	7.69e-15	0	0.9044296
0.5-3	1	1	A	9.58e-15	0	0.8212432	1	1	A	9.58e-15	0	0.6253746	1	1	A	9.58e-15	0	0.8255751
0.5-4	1	1	A	8.40e-15	0	0.7694203	1	1	A	8.40e-15	0	0.6164253	1	1	A	8.40e-15	0	0.8641647
0.5-5	1	1	A	1.03e-14	0	0.7619901	1	1	A	1.03e-14	0	0.6288575	1	1	A	1.03e-14	0	0.8235627
0.7-1	1	1	A	4.15e-15	0	1.1379071	1	1	A	4.15e-15	0	0.8540908	1	1	A	4.15e-15	0	1.0651701
0.7-2	1	1	A	2.86e-15	0	1.0603544	1	1	A	2.86e-15	0	0.8720381	1	1	A	2.86e-15	0	1.0502731
0.7-3	1	1	A	3.87e-15	0	0.9640158	1	1	A	3.87e-15	0	0.8661242	1	1	A	3.87e-15	0	1.0364254
0.7-4	1	1	A	7.54e-15	0	1.0128730	1	1	A	7.54e-15	0	0.8589913	1	1	A	7.54e-15	0	1.0551719
0.7-5	1	1	A	5.81e-15	0	0.9355892	1	1	A	5.81e-15	0	0.8733455	1	1	A	5.81e-15	0	0.9912615
0.9-1	1	1	A	8.74e-15	0	1.3367960	1	1	A	8.74e-15	0	1.1815895	1	1	A	8.74e-15	0	1.3526518
0.9-2	1	1	A	1.26e-14	0	1.3277053	1	1	A	1.26e-14	0	1.2029873	1	1	A	1.26e-14	0	1.3977944
0.9-3	1	1	A	1.69e-14	0	1.3335071	1	1	A	1.69e-14	0	1.2064882	1	1	A	1.69e-14	0	1.4018636
0.9-4	1	1	A	1.43e-14	0	1.3638778	1	1	A	1.43e-14	0	1.2110710	1	1	A	1.43e-14	0	1.3553098
0.9-5	1	1	A	1.15e-14	0	1.3618632	1	1	A	1.15e-14	0	1.1765483	1	1	A	1.15e-14	0	1.4617772

Table 8: Results for ill-conditioned strongly feasible instances ($\mu = 1e - 100$)

Instance	Algorithms 1 and 2						Lourenço (2019)						Pena (2017)					
ν-#	BP	MA	output	$\Vert\mathcal{A}(X)\Vert_2$	VDO	time(s)	BP	MA	output	$\Vert\mathcal{A}(X)\Vert_2$	VDO	time(s)	BP	MA	output	$\Vert\mathcal{A}(X)\Vert_2$	VDO	time(s)
0.1-1	121	1	A	2.38e-14	0	1.6387744	121	1	A	2.38e-14	0	1.7150522	121	1	A	2.38e-14	0	2.2256335
0.1-2	124	1	A	9.05e-14	0	1.6551637	124	1	A	9.05e-14	0	1.6374681	124	1	A	9.05e-14	0	2.0414585
0.1-3	134	1	A	9.59e-15	0	1.7113225	134	1	A	9.59e-15	0	1.6677083	134	1	A	9.59e-15	0	2.1109679
0.1-4	125	1	A	1.35e-13	0	1.6607036	125	1	A	1.35e-13	0	1.6122614	125	1	A	1.35e-13	0	2.121956
0.1-5	154	1	A	8.03e-14	0	1.7984189	154	1	A	8.03e-14	0	1.7935053	154	1	A	8.03e-14	0	2.1867748
0.3-1	117	1	A	1.21e-14	0	1.0204448	117	1	A	1.21e-14	0	0.9954115	117	1	A	1.21e-14	0	1.408101
0.3-2	119	1	A	4.74e-14	0	1.0499242	119	1	A	4.74e-14	0	1.0047912	119	1	A	4.74e-14	0	1.385821
0.3-3	140	1	A	6.59e-14	0	1.2101179	140	1	A	6.59e-14	0	1.1254235	140	1	A	6.59e-14	0	1.5758034
0.3-4	111	1	A	1.44e-14	0	1.0070936	111	1	A	1.44e-14	0	0.9588138	111	1	A	1.44e-14	0	1.3578287
0.3-5	118	1	A	8.37e-14	0	1.0265649	118	1	A	8.37e-14	0	1.0042043	118	1	A	8.37e-14	0	1.3478176
0.5-1	96	1	A	6.12e-14	0	1.1216348	96	1	A	6.12e-14	0	1.1104324	96	1	A	6.12e-14	0	1.4705758
0.5-2	117	1	A	2.88e-14	0	1.2595878	117	1	A	2.88e-14	0	1.1944254	117	1	A	2.88e-14	0	1.7713674
0.5-3	103	1	A	6.72e-15	0	1.1668168	103	1	A	6.72e-15	0	1.1398436	103	1	A	6.72e-15	0	1.4755687
0.5-4	105	1	A	1.22e-14	0	1.1773877	105	1	A	1.22e-14	0	1.1496237	105	1	A	1.22e-14	0	1.511845
0.5-5	106	1	A	1.15e-13	0	1.1661042	106	1	A	1.15e-13	0	1.1533402	106	1	A	1.15e-13	0	1.4673021
0.7-1	1	1	A	6.90e-14	0	0.8757366	1	1	A	6.90e-14	0	0.852107	1	1	A	6.90e-14	0	0.9877629
0.7-2	1	1	A	2.03e-14	0	0.8557468	1	1	A	2.03e-14	0	0.8542043	1	1	A	2.03e-14	0	0.9953294
0.7-3	1	1	A	5.62e-15	0	0.8945929	1	1	A	5.62e-15	0	0.8639633	1	1	A	5.62e-15	0	0.9782953
0.7-4	1	1	A	4.89e-14	0	0.846341	1	1	A	4.89e-14	0	0.863654	1	1	A	4.89e-14	0	0.972185
0.7-5	82	1	A	7.21e-14	0	1.2761274	82	1	A	7.21e-14	0	1.2687806	82	1	A	7.21e-14	0	1.5554597
0.9-1	1	1	A	2.69e-14	0	1.2300883	1	1	A	2.69e-14	0	1.1975233	1	1	A	2.69e-14	0	1.4242997
0.9-2	1	1	A	2.37e-14	0	1.2137822	1	1	A	2.37e-14	0	1.2399116	1	1	A	2.37e-14	0	1.3557746
0.9-3	1	1	A	5.80e-15	0	1.2296263	1	1	A	5.80e-15	0	1.2058631	1	1	A	5.80e-15	0	1.3695546
0.9-4	1	1	A	2.72e-14	0	1.1819559	1	1	A	2.72e-14	0	1.2304005	1	1	A	2.72e-14	0	1.3675839
0.9-5	1	1	A	5.80e-14	0	1.207665	1	1	A	5.80e-14	0	1.1721569	1	1	A	5.80e-14	0	1.3076884

Table 9: Results for ill-conditioned strongly feasible instances (μ = 1e-150)

Instance	Algorithms 1 and 2						Lourenço (2019)						Pena (2017)					
ν-#	BP	MA	output	$\\|\mathcal{A}(X)\\|_2$	VDO	time(s)	BP	MA	output	$\\|\mathcal{A}(X)\\|_2$	VDO	time(s)	BP	MA	output	$\\|\mathcal{A}(X)\\|_2$	VDO	time(s)
0.1-1	5	3	A	5.03e-14	0	2.773468	21812	336	A	9.67e-14	0	426.1549301	11932	1	A	3.21e-15	0	
0.1-2	4	3	A	4.33e-14	0	2.806292	22531	373	A	4.63e-13	0	435.0133058	13085	1	A	1.40e-12	0	
0.1-3	4	3	A	1.26e-15	0	2.7670643	21192	359	A	3.09e-13	0	424.5882803	12585	1	A	4.77e-13	0	
0.1-4	4	3	A	2.42e-14	0	2.9246545	21483	360	A	7.84e-14	0	426.526301	12906	1	A	9.19e-13	0	
0.1-5	73	3	A	1.53e-14	0	3.0776831	21113	360	A	1.83e-13	0	425.4949696	13470	1	A	4.21e-13	0	
0.3-1	6	3	A	3.79e-14	0	1.0852375	20223	366	A	2.44e-13	0	216.7162588	11939	1	A	2.60e-13	0	
0.3-2	4	3	A	5.03e-14	0	1.0755992	20543	372	A	2.15e-13	0	219.8931125	10974	1	A	1.57e-15	0	
0.3-3	4	3	A	5.94e-16	0	1.0773654	20656	375	A	2.44e-13	0	222.8317563	11378	1	A	8.25e-13	0	
0.3-4	3	3	A	6.61e-15	0	1.0885237	19910	363	A	1.88e-14	0	213.7365411	12015	1	A	3.39e-13	0	
0.3-5	3	3	A	2.40e-14	0	1.0664575	19971	365	A	2.00e-14	0	215.9343883	11039	1	A	1.58e-13	0	
0.5-1	3	3	A	3.48e-14	0	1.7283577	17589	334	A	7.37e-15	0	258.8787464	9249	1	A	4.43e-13	0	
0.5-2	3	3	A	2.93e-14	0	1.6617702	17111	324	A	1.86e-13	0	250.9817415	9791	1	A	5.81e-13	0	
0.5-3	3	3	A	5.39e-14	0	1.6241163	18051	339	A	3.41e-13	0	272.7194712	9502	1	A	1.10e-14	0	
0.5-4	3	3	A	9.12e-14	0	1.6394842	16952	323	A	3.42e-13	0	254.8115518	9382	1	A	2.30e-13	0	
0.5-5	3	3	A	1.91e-14	0	1.636959	18310	343	A	2.55e-13	0	266.3118079	9926	1	A	1.74e-13	0	
0.7-1	3	3	A	4.77e-14	0	2.379381	13878	277	A	1.35e-13	0	278.8106301	7668	1	A	4.53e-13	0	
0.7-2	3	3	A	6.06e-14	0	2.4209785	14535	289	A	3.52e-14	0	292.6777175	7651	1	A	1.08e-12	0	
0.7-3	3	3	A	7.08e-14	0	2.3435376	12632	257	A	3.28e-13	0	256.8142889	7592	1	A	1.61e-13	0	
0.7-4	3	3	A	1.34e-14	0	2.3725624	13357	268	A	8.67e-14	0	268.8962612	7636	1	A	1.11e-13	0	
0.7-5	3	3	A	1.49e-14	0	2.374409	14719	291	A	1.24e-13	0	292.8987944	7664	1	A	8.09e-13	0	
0.9-1	3	3	A	3.21e-14	0	3.4017905	8520	195	A	2.57e-14	0	256.1041177	5805	1	A	8.28e-14	0	
0.9-2	3	3	A	3.07e-14	0	3.3751842	8145	188	A	1.22e-13	0	246.4843027	5440	1	A	5.43e-13	0	
0.9-3	3	3	A	2.96e-14	0	3.3996172	8757	201	A	6.89e-14	0	263.5397179	5992	1	A	9.06e-13	0	
0.9-4	3	3	A	2.84e-14	0	3.3365115	6685	156	A	4.45e-13	0	204.1381791	5949	1	A	4.00e-14	0	
0.9-5	3	3	A	2.23e-14	0	3.3542622	9346	213	A	1.09e-13	0	280.2315643	6289	1	A	1.37e-12	0	

Table 10: Results for ill-conditioned strongly feasible instances (μ = 1e-200)

Instance	Algorithms 1 and 2						Lourenço (2019)						Pena (2017)					
ν-#	BP	MA	output	$\Vert\mathcal{A}(X)\Vert_2$	VDO	time(s)	BP	MA	output	$\Vert\mathcal{A}(X)\Vert_2$	VDO	time(s)	BP	MA	output	$\Vert\mathcal{A}(X)\Vert_2$	VDO	time(s)
0.1-1	409	4	A	2.75e-14	0	5.7552823	40250	970	A	4.05e-13	0	1066.07629			E			
0.1-2	416	4	A	1.60e-14	0	5.7441049	42199	1003	A	2.18e-13	0	1102.459604			E			
0.1-3	383	4	A	6.22e-15	0	5.5618205	44511	1031	A	1.26e-12	0	1137.163597			E			
0.1-4	386	4	A	1.19e-13	0	5.6037385	34944	916	A	3.43e-13	0	986.3248812			E			
0.1-5	410	4	A	6.20e-14	0	5.657353	42330	993	A	2.34e-13	0	1093.694564			E			
0.3-1	331	4	A	1.20e-13	0	3.0299778	40960	964	A	5.08e-14	0	510.2614017			E			
0.3-2	371	4	A	2.86e-14	0	3.2289256	42704	1005	A	3.87e-13	0	532.8962177			E			
0.3-3	347	4	A	3.82e-14	0	3.1098924	42271	998	A	1.33e-13	0	527.6459355			E			
0.3-4	331	4	A	7.38e-14	0	3.0055526	42798	1007	A	7.94e-13	0	531.7280592			E			
0.3-5	346	4	A	5.60e-14	0	3.1212584	43396	1013	A	4.72e-13	0	536.7204093			E			
0.5-1	259	6	A	3.17e-14	0	4.4052456	32185	794	A	7.55e-13	0	567.6067278	970378	1	A	1.47e-11	0	6665.344358
0.5-2	294	6	A	4.97e-14	0	4.6318177	25855	650	A	5.88e-13	0	463.4696282	905575	1	A	3.17e-13	0	6243.26788
0.5-3	278	6	A	9.66e-15	0	4.650686	32839	805	A	1.61e-14	0	574.5412445	956494	1	A	5.20e-13	0	6514.193914
0.5-4	254	6	A	9.04e-14	0	4.4760995	35796	867	A	9,42e-14	0	623.5893255	893792	1	A	7.62e-12	0	6371.895568
0.5-5	308	6	A	6.19e-14	0	4.6709573	39005	932	A	3.06e-13	0	663.6567383	959852	1	A	3.48e-12	0	6520.961655
0.7-1	176	6	A	7.09e-14	0	5.4873865	22465	565	A	1.18e-12	0	537.975948	744325	1	A	1.11e-12	0	5286.233126
0.7-2	166	6	A	6.57e-14	0	5.4991973	28291	703	A	1.75e-13	0	670.5647452	706771	1	A	1.03e-12	0	5001.624059
0.7-3	155	6	A	1.46e-13	0	5.398204	22332	562	A	2.83e-13	0	534.1304925	695557	1	A	4.34e-12	0	4972.25653
0.7-4	85	6	A	9.39e-14	0	5.1058159	22362	562	A	8.46e-13	0	536.2010921	680455	1	A	5.88e-13	0	4613.547696
0.7-5	93	6	A	1.98e-14	0	5.0924505	22204	558	A	7.81e-13	0	532.033156	686677	1	A	1.28e-12	0	4888.655133
0.9-1	1036	34	A	1.10e-15	0	41.8078726	17035	474	A	5.11e-13	0	606.0034344	522477	1	A	9.76e-12	0	4115.587447
0.9-2	4	4	A	1.27e-13	0	4.5601347	15044	420	A	1.22e-12	0	535.5483217	601249	1	A	6.16e-12	0	4608.571072
0.9-3	1124	36	A	4.31e-14	0	44.801746	16775	466	A	1.66e-12	0	595.4614636	590494	1	A	3.52e-12	0	4501.457961
0.9-4	4	4	A	1.22e-13	0	4.4031596	16985	471	A	1.09e-12	0	600.6289894	550505	1	A	4.11e-12	0	4217.908702
0.9-5	1173	37	A	1.03e-14	0	45.8802531	17228	478	A	7.75e-13	0	608.5334611	573684	1	A	1.49e-12	0	4428.371154

Table 11: Results for ill-conditioned strongly feasible instances (μ = 1e-250)

Instance	Algorithms 1 and 2						Lourenço (2019)						Pena (2017)					
ν-#	BP	MA	output	$\\|\mathcal{A}(X)\\|_2$	VDO	time(s)	BP	MA	output	$\\|\mathcal{A}(X)\\|_2$	VDO	time(s)	BP	MA	output	$\\|\mathcal{A}(X)\\|_2$	VDO	time(s)
0.1-1	178	6	A	1.66e-14	0	6.2331712	72317	1766	A	4.82e-13	0	1908.04608			E			
0.1-2	180	6	A	4.51e-14	0	6.2784421	71383	1749	A	2.19e-12	0	1909.155933			E			
0.1-3	173	6	A	2.59e-13	0	6.3108265	72652	1764	A	1.39e-12	0	1931.952164			E			
0.1-4	193	6	A	2.54e-13	0	6.4164866	72913	1771	A	1.05e-11	0	1928.410829			E			
0.1-5	160	6	A	8.93e-14	0	6.1533087	71360	1748	A	9.21e-13	0	1951.197366			E			
0.3-1	155	6	A	1.78e-13	0	2.8176788	73792	1838	A	2.45e-12	0	942.2237684			E			
0.3-2	116	6	A	1.81e-13	0	2.6180193	68077	1681	A	1.71e-12	0	874.3937609			E			
0.3-3	161	6	A	6.75e-14	0	2.8536591	68757	1709	A	5.60e-12	0	918.3929694			E			
0.3-4	173	6	A	7.46e-14	0	2.8890576	67039	1645	A	5.69e-14	0	862.6867444			E			
0.3-5	162	6	A	5.68e-14	0	2.8313933	74794	1824	A	4.35e-12	0	949.8225914			E			
0.5-1	154	6	A	4.62e-14	0	3.9537311	72212	1748	A	1.39e-12	0	1252.636485			E			
0.5-2	136	6	A	1.46e-13	0	3.8504726	48984	1241	A	2.30e-12	0	876.8490339			E			
0.5-3	138	6	A	1.07e-13	0	3.8078101	48701	1233	A	5.54e-13	0	872.4318737			E			
0.5-4	128	6	A	2.15e-13	0	3.8257523	60386	1511	A	3.34e-13	0	1079.539503			E			
0.5-5	125	6	A	1.07e-14	0	3.7868361	49308	1249	A	1.24e-12	0	889.9471784			E			
0.7-1	117	6	A	1.72e-13	0	5.1786258	43761	1132	A	2.55e-13	0	1072.773566			E			
0.7-2	101	6	A	7.11e-15	0	5.2604768	43015	1124	A	3.59e-12	0	1058.43577			E			
0.7-3	95	6	A	4.53e-14	0	5.2255387	41187	1077	A	1.39e-12	0	1015.479041			E			
0.7-4	110	6	A	9.77e-14	0	5.2730147	41697	1096	A	2.53e-12	0	1030.979851			E			
0.7-5	110	6	A	9.24e-14	0	5.2742655	42413	1105	A	3.86e-12	0	1042.055228			E			
0.9-1	6	6	A	1.40e-13	0	6.5968646	26706	754	A	3.53e-12	0	955.3815094			E			
0.9-2	6	6	A	1.26e-13	0	6.5842219	27033	773	A	4.47e-12	0	980.415325			E			
0.9-3	6	6	A	2.27e-13	0	6.5801057	26503	751	A	9.46e-12	0	953.9931748			E			
0.9-4	6	6	A	7,11e-14	0	6.5534042	25493	728	A	2.04e-13	0	926.5546267			E			
0.9-5	6	6	A	1.29e-13	0	6.6335852	25124	720	A	1.42e-11	0	924.1553132			E			

Table 12: Results for infeasible instances (α =1e-1)

Instance	Algorithms 1 and 2						Lourenço (2019)						Pena (2017)					
ν-#	BP	MA	output	$\Vert\mathcal{A}(X)\Vert_2$	VDO	time(s)	BP	MA	output	$\Vert\mathcal{A}(X)\Vert_2$	VDO	time(s)	BP	MA	output	$\Vert\mathcal{A}(X)\Vert_2$	VDO	time(s)
0.1-1	4	4	B		0	3.7001904	27	1	B		0	1.0595699	27	1	B		0	1.1515893
0.1-2	3	3	B		0	2.8385329	453	24	B		0	23.768789	32	1	B		0	1.1615707
0.1-3	1	1	B		0	0.9686051	1	1	B		0	0.9407752	1	1	B		0	0.9491649
0.1-4	3	3	B		0	2.8412129	13	1	B		0	1.0085495	13	1	B		0	1.0408792
0.1-5	2	2	B		0	1.8741805	4	1	B		0	0.9518278	4	1	B		0	0.9615953
0.3-1	2	2	B		0	0.6123546	3	1	B		0	0.3048468	3	1	B		0	0.3020669
0.3-2	13	7	B		0	2.066111	26	1	B		0	0.4377363	26	1	B		0	0.4779242
0.3-3	1	1	B		0	0.4156639	1	1	B		0	0.3350198	1	1	B		0	0.3273617
0.3-4	2	2	B		0	0.6376384	4	1	B		0	0.3407447	4	1	B		0	0.3607341
0.3-5	15	6	B		0	1.8570432	1618	43	B		0	19.4427856	144	1	B		0	1.2316734
0.5-1	2	2	B		0	0.9850592	2	1	B		0	0.4748843	2	1	B		0	0.4993438
0.5-2	1	1	B		0	0.5057218	1	1	B		0	0.4859488	1	1	B		0	0.4804082
0.5-3	1	1	B		0	0.5114008	1	1	B		0	0.4802219	1	1	B		0	0.4990953
0.5-4	1	1	B		0	0.525315	1	1	B		0	0.4563445	1	1	B		0	0.4914326
0.5-5	5	3	B		0	1.6172137	9	1	B		0	0.5012577	9	1	B		0	0.5234442
0.7-1	1	1	B		0	0.7092575	1	1	B		0	0.674741	1	1	B		0	0.722887
0.7-2	1	1	B		0	0.7644295	1	1	B		0	0.7231888	1	1	B		0	0.7048646
0.7-3	1	1	B		0	0.7277629	1	1	B		0	0.6474515	1	1	B		0	0.6712546
0.7-4	1	1	B		0	0.7203443	1	1	B		0	0.6640843	1	1	B		0	0.824931
0.7-5	1	1	B		0	0.6990928	1	1	B		0	0.638089	1	1	B		0	0.7854615
0.9-1	1	1	B		0	1.0888668	1	1	B		0	1.0087134	1	1	B		0	1.0471631
0.9-2	1	1	B		0	1.1009438	1	1	B		0	0.9541349	1	1	B		0	1.1402338
0.9-3	1	1	B		0	1.0355239	1	1	B		0	0.9986441	1	1	B		0	1.0953168
0.9-4	1	1	B		0	1.0405158	1	1	B		0	0.9717854	1	1	B		0	1.1775349
0.9-5	1	1	B		0	1.1039531	1	1	B		0	0.9744629	1	1	B		0	1.1611032

Table 13: Results for infeasible instances (α =1e-2)

Instance	Algorithms 1 and 2						Lourenço (2019)						Pena (2017)					
ν-#	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)
0.1-1	9	9	B		0	8.2172375	3161	88	B		0	92.6441567	477	1	B		0	4.3251145
0.1-2	4	4	B		0	3.7334121	27	1	B		0	1.0762897	27	1	B		0	1.2653939
0.1-3	256	30	B		0	28.342651	21639	269	B		0	349.9697739	30818	1	B		0	213.8714908
0.1-4	347	46	B		0	43.2581381	31364	282	B		0	425.0839298	62360	1	D		0	415.4186114
0.1-5	5	5	B		0	4.5999158	1673	65	B		0	66.3646455	138	1	B		0	1.8497866
0.3-1	4	4	B		0	1.2244352	7	1	B		0	0.3717799	7	1	B		0	0.3701904
0.3-2	2	2	B		0	0.6204273	2	1	B		0	0.3604724	2	1	B		0	0.3399988
0.3-3	3	3	B		0	0.8760455	6	1	B		0	0.3352106	6	1	B		0	0.3727354
0.3-4	4	4	B		0	1.2260232	10	1	B		0	0.3674549	10	1	B		0	0.4156494
0.3-5	2	2	B		0	0.618568	2	1	B		0	0.3045045	2	1	B		0	0.3550636
0.5-1	2	2	B		0	0.9810526	2	1	B		0	0.4845246	2	1	B		0	0.4998802
0.5-2	2	2	B		0	0.9907018	4	1	B		0	0.4650267	4	1	B		0	0.5051942
0.5-3	1	1	B		0	0.5023657	1	1	B		0	0.4730836	1	1	B		0	0.4771757
0.5-4	12	6	B		0	2.8671091	50	2	B		0	1.1525649	26	1	B		0	0.6468266
0.5-5	1	1	B		0	0.4865508	1	1	B		0	0.4818792	1	1	B		0	0.489947
0.7-1	2	2	B		0	1.3609228	2	1	B		0	0.6824177	2	1	B		0	0.6819742
0.7-2	1	1	B		0	0.7392089	1	1	B		0	0.6779459	1	1	B		0	0.703029
0.7-3	1	1	B		0	0.7721353	1	1	B		0	0.6921287	1	1	B		0	0.687381
0.7-4	2	2	B		0	1.4430588	2	1	B		0	0.7022044	2	1	B		0	0.6740751
0.7-5	2	2	B		0	1.4214986	2	1	B		0	0.6700654	2	1	B		0	0.6925999
0.9-1	1	1	B		0	1.0526325	1	1	B		0	0.9169453	1	1	B		0	1.0618605
0.9-2	1	1	B		0	1.1189465	1	1	B		0	0.9909742	1	1	B		0	1.0122634
0.9-3	1	1	B		0	1.0413235	1	1	B		0	0.9694412	1	1	B		0	1.001551
0.9-4	1	1	B		0	1.0848454	1	1	B		0	1.0004292	1	1	B		0	0.9892756
0.9-5	1	1	B		0	1.1916999	1	1	B		0	1.065735	1	1	B		0	0.9376565

Table 14: Results for infeasible instances (α =1e-3)

Instance	Algorithms 1 and 2						Lourenço (2019)						Pena (2017)					
ν-#	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)
0.1-1	57	27	B		0	24.5245899	15548	277	B		0	329.2945936	29806	1	D		0	177.7538974
0.1-2	139	33	B		0	30.8580027	22358	289	B		0	397.2351272	49391	1	D		0	292.8056339
0.1-3	3	3	B		0	2.8171276	17	1	B		0	1.0260522	17	1	B		0	1.0592263
0.1-4	13	10	B		0	8.9379918	6304	186	B		0	197.8939123	1646	1	B		0	11.3808969
0.1-5	208	45	B		0	40.6143076	29194	356	B		0	469.5797717	307338	3	D		0	950.669117
0.3-1	18	8	B		0	2.6522652	31	1	B		0	0.5303537	31	1	B		0	0.5608975
0.3-2	76	14	B		0	4.8721578	11376	135	B		0	99.1449564	14996	1	D		0	90.3294462
0.3-3	16	7	B		0	2.3403141	647	18	B		0	8.9076151	62	1	B		0	0.7455794
0.3-4	3	3	B		0	1.0125133	6	1	B		0	0.3899123	6	1	B		0	0.366642
0.3-5	29	10	B		0	2.8931766	2124	46	B		0	25.3131169	237	1	B		0	1.835086
0.5-1	1	1	B		0	0.5000507	1	1	B		0	0.5034246	1	1	B		0	0.5006385
0.5-2	1	1	B		0	0.5308222	1	1	B		0	0.4604563	1	1	B		0	0.5083429
0.5-3	1	1	B		0	0.5010961	1	1	B		0	0.4827022	1	1	B		0	0.4887736
0.5-4	10	4	B		0	1.6994708	18	1	B		0	0.5679317	18	1	B		0	0.6041985
0.5-5	2	2	B		0	0.9296181	2	1	B		0	0.4886044	2	1	B		0	0.4694553
0.7-1	1	1	B		0	0.7564631	1	1	B		0	0.6753715	1	1	B		0	0.7045053
0.7-2	1	1	B		0	0.7507378	1	1	B		0	0.6527091	1	1	B		0	0.7542288
0.7-3	1	1	B		0	0.7029142	1	1	B		0	0.6982651	1	1	B		0	0.7119958
0.7-4	1	1	B		0	0.7223131	1	1	B		0	0.6471624	1	1	B		0	0.7071553
0.7-5	1	1	B		0	0.6626209	1	1	B		0	0.6661495	1	1	B		0	0.7098997
0.9-1	1	1	B		0	1.0320648	1	1	B		0	0.9662989	1	1	B		0	0.9613485
0.9-2	1	1	B		0	1.0086467	1	1	B		0	1.0025238	1	1	B		0	1.0264735
0.9-3	1	1	B		0	1.0075072	1	1	B		0	1.0397598	1	1	B		0	0.9873226
0.9-4	1	1	B		0	1.0552788	1	1	B		0	1.0529587	1	1	B		0	1.0559268
0.9-5	1	1	B		0	1.0192336	1	1	B		0	1.0608594	1	1	B		0	1.0507835

Table 15: Results for infeasible instances (α =1e-4)

Instance	Algorithms 1 and 2						Lourenço (2019)						Pena (2017)					
ν-#	BP	MA	output	$\Vert\mathcal{A}(X)\Vert_2$	VDO	time(s)	BP	MA	output	$\Vert\mathcal{A}(X)\Vert_2$	VDO	time(s)	BP	MA	output	$\Vert\mathcal{A}(X)\Vert_2$	VDO	time(s)
0.1-1	6	6	B		0	5.4223425	1633	66	B		0	67.5545693	84	1	B		0	1.5487362
0.1-2	4	4	B		0	3.6222482	620	27	B		0	26.7670382	45	1	B		0	1.2538611
0.1-3	72	27	B		0	24.2189419	19238	295	B		0	356.3288734	34709	1	D		0	205.75019
0.1-4	1329	128	B		0	119.4729866	85312	1012	B		0	1315.592211			E			
0.1-5	45	22	B		0	19.5964729	13559	279	B		0	322.4184792	30861	1	D		0	194.8114951
0.3-1	83	16	B		0	4.5086882	13920	128	B		0	106.5771431	13427	1	D		0	82.7106296
0.3-2	2	2	B		0	0.6062406	4	1	B		0	0.346117	4	1	B		0	0.3420163
0.3-3	2	2	B		0	0.5814123	4	1	B		0	0.3538336	4	1	B		0	0.3904092
0.3-4	5	4	B		0	1.141968	7	1	B		0	0.3561615	7	1	B		0	0.3913145
0.3-5	3	3	B		0	0.885372	7	1	B		0	0.3523754	7	1	B		0	0.3968585
0.5-1	2	2	B		0	0.9703464	2	1	B		0	0.5133042	2	1	B		0	0.5457557
0.5-2	2	2	B		0	0.9942528	2	1	B		0	0.4992089	2	1	B		0	0.5612568
0.5-3	1	1	B		0	0.5089972	1	1	B		0	0.4887569	1	1	B		0	0.5784672
0.5-4	2	2	B		0	1.0437497	2	1	B		0	0.5189815	2	1	B		0	0.5308671
0.5-5	1	1	B		0	0.4707021	1	1	B		0	0.5656425	1	1	B		0	0.5311213
0.7-1	1	1	B		0	0.665617	1	1	B		0	0.6942802	1	1	B		0	0.7281943
0.7-2	1	1	B		0	0.7016006	1	1	B		0	0.7355567	1	1	B		0	0.7441333
0.7-3	2	2	B		0	1.3582746	2	1	B		0	0.6984835	2	1	B		0	0.7545126
0.7-4	1	1	B		0	0.6844078	1	1	B		0	0.6583403	1	1	B		0	0.7588416
0.7-5	1	1	B		0	0.6723618	1	1	B		0	0.7176463	1	1	B		0	0.7079508
0.9-1	1	1	B		0	1.0482914	1	1	B		0	1.074521	1	1	B		0	1.0588227
0.9-2	1	1	B		0	1.0432198	1	1	B		0	0.9946941	1	1	B		0	1.0011707
0.9-3	1	1	B		0	1.037613	1	1	B		0	1.0480436	1	1	B		0	1.0414181
0.9-4	1	1	B		0	2.9471596	1	1	B		0	1.0071091	1	1	B		0	1.0435363
0.9-5	1	1	B		0	1.0542458	1	1	B		0	1.0520622	1	1	B		0	0.9933612

Table 16: Results for infeasible instances (α =1e-5)

Instance	Algorithms 1 and 2						Lourenço (2019)						Pena (2017)					
ν-#	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)
0.1-1	4	4	B		0	3.6335922	25	1	B		0	1.0918616	25	1	B		0	1.1357613
0.1-2	2574	154	B		0	155.3501023	80052	871	B		0	1195.042106			E			
0.1-3	92	28	B		0	25.2796406	17589	257	B		0	321.2619442	33091	1	D		0	196.7751412
0.1-4	12	11	B		0	9.8563917	4868	147	B		0	156.7754356	640	1	B		0	5.3167943
0.1-5	16	15	B		0	13.6102055	5896	188	B		0	198.9498894	1276	1	B		0	9.5919569
0.3-1	4	3	B		0	0.8766825	7	1	B		0	0.3792057	7	1	B		0	0.4061693
0.3-2	3	3	B		0	0.861668	7	1	B		0	0.3458187	7	1	B		0	0.3763403
0.3-3	18	7	B		0	1.9544264	1030	30	B		0	13.409027	89	1	B		0	0.9546305
0.3-4	1	1	B		0	0.3465197	1	1	B		0	0.3729548	1	1	B		0	0.3577409
0.3-5	11	6	B		0	1.7037819	354	12	B		0	5.2257418	39	1	B		0	0.6029442
0.5-1	2	2	B		0	1.1361508	2	1	B		0	0.491314	2	1	B		0	0.7091467
0.5-2	3	2	B		0	1.109544	4	1	B		0	0.5293216	4	1	B		0	0.511467
0.5-3	1	1	B		0	0.5957484	1	1	B		0	0.4901051	1	1	B		0	0.5948215
0.5-4	4	3	B		0	1.2676497	5	1	B		0	0.5091715	5	1	B		0	0.5231118
0.5-5	1	1	B		0	0.4795229	1	1	B		0	0.4987233	1	1	B		0	0.5568121
0.7-1	1	1	B		0	0.7572184	1	1	B		0	0.7382602	1	1	B		0	0.6956164
0.7-2	1	1	B		0	0.7238205	1	1	B		0	0.7672721	1	1	B		0	0.688156
0.7-3	1	1	B		0	0.7089488	1	1	B		0	0.7474796	1	1	B		0	0.7303869
0.7-4	1	1	B		0	0.6751619	1	1	B		0	0.727944	1	1	B		0	0.771833
0.7-5	1	1	B		0	0.732681	1	1	B		0	0.6710638	1	1	B		0	0.7180038
0.9-1	1	1	B		0	1.0082914	1	1	B		0	1.0987726	1	1	B		0	1.0828292
0.9-2	1	1	B		0	1.0967342	1	1	B		0	1.0248307	1	1	B		0	1.0574369
0.9-3	1	1	B		0	1.1164186	1	1	B		0	1.016096	1	1	B		0	1.1138199
0.9-4	1	1	B		0	1.0163322	1	1	B		0	1.0073382	1	1	B		0	1.0976848
0.9-5	1	1	B		0	1.049071	1	1	B		0	0.9883912	1	1	B		0	1.0200209

Table 17: Results for weakly feasible instances

Instance	Algorithms 1 and 3						Lourenço (2019)						Pena (2017)					
ν-#	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)	BP	MA	output	$\|\mathcal{A}(X)\|_2$	VDO	time(s)
0.1-1	9361	469	C			457.0683	298119	3060	A	2.04e-12	25	4224.9416			E			
0.1-2	8452	429	C			425.0041	308785	2867	A	2.80e-13	23	4109.3598			E			
0.1-3	9664	470	C			467.6987	299998	3039	A	3.59e-13	24	4212.9030			E			
0.1-4	9591	484	C			481.0258	277732	3268	A	8.45e-13	25	4315.1364			E			
0.1-5	9691	458	C			467.9452	308999	3010	A	1.85e-12	24	4219.8625			E			
0.3-1	6340	489	C			161.7357	305628	3363	B		0	2427.2107			E			
0.3-2	6372	485	C			160.9361	327035	3375	B		2	2512.3831			E			
0.3-3	6180	464	C			154.6871	327285	3382	B		3	2528.2178			E			
0.3-4	6410	480	C			159.1339	319265	3332	B		1	2463.6540			E			
0.3-5	6288	473	C			159.4782	317750	3360	B		0	2491.7035			E			
0.5-1	6680	436	C			216.0363	336318	3176	A	2.88e-13	23	2979.5541			E			
0.5-2	7067	449	C			222.2420	321216	3213	B		0	2920.6234			E			
0.5-3	7042	456	C			225.5087	340088	3321	A	4.22e-13	24	3024.0695			E			
0.5-4	7573	455	C			226.1386	331684	3305	A	4.40e-13	24	3007.0799			E			
0.5-5	7239	467	C			230.1067	336730	3478	A	4.06e-13	25	3108.1958			E			
0.7-1	6382	426	C			286.9537	319060	3289	A	3.81e-13	24	3548.4511			E			
0.7-2	6467	442	C			299.0165	323385	3433	B		0	3655.4779			E			
0.7-3	6541	449	C			300.7240	316243	3450	A	6.11e-13	25	3667.4578			E			
0.7-4	6533	438	C			296.2843	322159	3294	B		1	3602.3840			E			
0.7-5	6617	436	C			297.1496	315796	3111	B		1	3442.8030			E			
0.9-1	5652	375	C			362.1734	290804	3123	B		0	4217.0427			E			
0.9-2	5937	388	C			372.3538	301787	3184	B		0	4313.6604			E			
0.9-3	5896	386	C			370.3980	296108	3172	B		0	4252.3639			E			
0.9-4	5970	397	C			381.4179	297296	3311	B		0	4398.5395			E			
0.9-5	5840	387	C			368.1002	294705	3178	B		0	4174.3028			E			