Polyhedral approximations of the semidefinite cone and their applications

著者	吉瀬 章子
発行年	2019-05
シリーズ	Department of Policy and Planning Sciences Discussion Paper Series;no.1359
URL	http://hdl.handle.net/2241/00155286

Department of Policy and Planning Sciences

Discussion Paper Series

No.1359

Polyhedral approximations of the semidefinite cone and their applications

by

Yuzhu WANG, Akihiro TANAKA, and Akiko YOSHISE

May 2019

UNIVERSITY OF TSUKUBA
Tsukuba, Ibaraki 305-8573
JAPAN

Polyhedral approximations of the semidefinite cone and their applications*

Yuzhu Wang†, Akihiro Tanaka‡ and Akiko Yoshise§

May 2019

Abstract

We develop techniques to construct a series of sparse polyhedral approximations of the semidefinite cone. Motivated by the semidefinite (SD) bases proposed by Tanaka and Yoshise (2018), we propose a simple expansion of SD bases so as to keep the sparsity of the matrices composing it. We prove that the polyhedral approximation using our expanded SD bases contains the set of diagonally dominant matrices and is contained in the set of scaled diagonally dominant matrices. We also prove that the set of scaled diagonally dominant matrices can be expressed using an infinite number of expanded SD bases. Using our polyhedral approximations, we develop new methods for identifying elements in certain cones. We also use our approximations as the initial approximation in the cutting-plane methods for some conic optimization problems.

Key words: Semidefinite optimization problems; Conic optimization problems; Polyhedral approximation; Semidefinite bases; Expanded semidefinite bases.

AMS subject classifications: 90C05, 90C22, 90C25

1 Introduction

A semidefinite optimization problem (SDP) is an optimization problem of variables in the symmetric matrix space with a linear objective function and linear constraints over the semidefinite cone. We denote the space of symmetric matrices as $\mathbb{S}^n := \{X \in \mathbb{R}^{n\times n} \mid X_{i,j} = X_{j,i} \ (1 \le i < j \le n)\}$ and the semidefinite cone as $\mathcal{S}^n_+ := \{X \in \mathbb{S}^n \mid d^T X d \ge 0 \text{ for any } d \in \mathbb{R}^n\}$. Accordingly, we can ready define an

*This research was supported by the Japan Society for the Promotion of Science through a Grant-in-Aid for Challenging Exploratory Research (17K18946) and a Grant-in-Aid for Scientific Research ((B)19H02373) of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

†Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan. email: s1930138@s.tsukuba.ac.jp

‡Central Research Institute of Electric Power Industry, Yokosuka, Kanagawa 240-0196, Japan. email: a-tanaka@criepi.denken.or.jp

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

SDP in the standard form, as

$$\begin{array}{rl} \min & \langle C, X\rangle \\ \text{s.t.} & \langle A_j, X\rangle = b_j, j = 1, 2, \ldots, m, \\ & X \in \mathcal{S}^n_+, \end{array} \tag{1}$$

where $C \in \mathbb{S}^n$, $A_j \in \mathbb{S}^n$, $b_j \in \mathbb{R}$ $(j = 1, 2, \ldots, m)$, and $\langle A, B\rangle := \text{Trace}(A^T B) = \sum_{i,j=1}^n A_{i,j}B_{i,j}$ is the inner product over $\mathbb{S}^n$.

SDPs are powerful tools that provide convex relaxations for combinatorial and nonconvex optimizations, such as the max-cut problem (e.g., [18], [12]) and the k-equipartition problem (e.g., [45], [22]). Some of these relaxations can even attain the optimum, as shown in [30] and [23]. Interested readers may find details about SDPs and their relaxations in [45], [41] and [31].

A cone $\mathcal{K} \subset \mathbb{S}^n$ is called proper if it has a non-empty interior and is closed, pointed (i.e., $\mathcal{K} \cap -\mathcal{K} = \{O\}$), and convex. It is known that the SDP cone is a proper cone [9]. By replacing the semidefinite constraint $X \in \mathcal{S}^n_+$ with a general conic constraint $X \in \mathcal{K}$ in (1) (say, a proper cone $\mathcal{K} \subset \mathbb{S}^n$), one can obtain a general class of problems, namely, conic optimization problems. The class of conic optimization problems has been an active field because it contains many popular classes of problems, including linear optimization problems (LPs), second-order cone programs (SOCPs), SDPs, and copositive programs. Copositive programs have been shown capable of providing tight lower bounds for combinatorial and quadratic optimization problems, as described in the survey paper by Dür [16] and the recent work of Arima et al. [3], [24], [4], etc. It has been shown that a copositive relaxation sometimes gives a highly accurate approximate solution for some combinatorial problems under certain conditions [5], [11]. However, the copositive program and its dual problem are both NP-hard (see, e.g., [15] and [34]).

SDPs are also attractive because they can be solved in polynomial time to any desired precision. There are state-of-the-art solvers, such as SDPA [46], SeDuMi [39], and SDPT3 [42], but their computations become difficult when the size of the SDP becomes large. To overcome this deficiency, for example, one may use preprocessing to reduce the size of the SDPs, which leads to facial reduction methods [35], [36] and [43]. As another idea, one may generate relaxations of SDPs and solve them as easily handled optimization problems, e.g., LPs and SOCPs, which leads to cutting plane methods. We will focus on these latter methods.

The cutting plane method solves an SDP by transforming it into an optimization problem (e.g., an LP or an SOCP) adding cutting planes at each iteration to cut the current approximate solution out of the feasible region in the next iterations and to get close to the optimal value. The cutting plane method was first used on the traveling-salesman problem, by Dantzig, Fulkerson, and Johnson [13], [14] in 1954. It was used in 1958 by Gomory [19] to solve integer linear programming problems. As SDPs became popular, it came to be used on them; see, for instance, Krishnan and Mitchell [27], [29] and [28], and Konno et al. [26]. Kobayashi and Takano [25] applied it to a class of mixed-integer SDPs. In [1], Ahmadi, Dash, and Hall applied it to nonconvex polynomial optimization problems and copositive programs.

In the above mentioned cutting plane methods for SDPs, the semidefinite constraint $X \in \mathcal{S}^n_+$ in (1) is first relaxed to $X \in \mathcal{K}_{\text{out}}$, where $\mathcal{S}^n_+ \subseteq \mathcal{K}_{\text{out}} \subseteq \mathbb{S}^n$, and an initial relaxation of the SDP is obtained. If $\mathcal{K}_{\text{out}}$ is polyhedral, the initial relaxation may give an LP; if $\mathcal{K}_{\text{out}}$ consists of second-order constraints, the initial relaxation becomes an SOCP. To improve the performance of these cutting plane methods, we consider generating initial relaxations for SDPs that are both tight and computationally efficient and focus on the approximations of $\mathcal{S}^n_+$.

Many approximations of $\mathcal{S}^n_+$ have been proposed on the basis of its well-known properties. Kobayashi and

Takano [25] used the fact that the diagonal elements of semidefinite matrices are nonnegative. Konno et al. [26] imposed an assumption that all diagonal elements of the variable X in the SDPs appearing in their iterative algorithm are bounded by a constant. The sets of diagonally dominant matrices and scaled diagonally dominant matrices are known to be cones contained in $\mathcal{S}^n_+$, (see, e.g., [21] and [1] for details). The inclusive relation among them has been studied in, e.g., [7] and [8]. Ahmadi et al. [1] and [2] used these sets as initial approximations of their cutting plane method. Boman et al. [10] defined the *factor width* of a semidefinite matrix, and Permenter and Parrilo used it to generate approximations of $\mathcal{S}^n_+$, which they applied to facial reduction methods in [35].

Tanaka and Yoshise defined various bases of $\mathbb{S}^n$, wherein each basis consists of $\frac{n(n+1)}{2}$ semidefinite matrices, called semidefinite (SD) bases, and used them to devise approximations of $\mathcal{S}^n_+$ [40]. They showed that the conical hull of SD bases and its dual cone give inner and outer polyhedral approximations of $\mathcal{S}^n_+$, respectively. On the basis of the SD bases, they also developed techniques to identify the semidefinite plus nonnegative cone $\mathcal{S}^n_+ + \mathcal{N}^n$, which is the Minkowski sum of $\mathcal{S}^n_+$ and the nonnegative matrices cone $\mathcal{N}^n$. In this paper, we focus on the fact that SD bases are sometimes sparse, i.e., the number of nonzero elements in a matrix is relatively small, and hence, it is not so computationally expensive to solve polyhedrally approximated problems in such SD bases. We call such an approximation, a *sparse polyhedral approximation*, and propose efficient sparse approximations of $\mathcal{S}^n_+$.

The goal of this paper is to construct tight and sparse polyhedral approximations of $\mathcal{S}^n_+$ by using SD bases in order to solve hard conic optimization problems, e.g., doubly nonnegative (DNN, or $\mathcal{S}^n_+ \cap \mathcal{N}^n$) and semidefinite plus nonnegative ($\mathcal{S}^n_+ + \mathcal{N}^n$) optimization problems. The contributions of this paper are summarized as follows.

- This paper gives the relation between the conical hull of sparse SD bases and the set of diagonally dominant matrices. We propose a simple expansion of SD bases without losing the sparsity of the matrices and prove that one can generate a sparse polyhedral approximation of $\mathcal{S}^n_+$ that contains the set of diagonally dominant matrices and is contained in the set of scaled diagonally dominant matrices.
- The expanded SD bases are used to identify $A \in \mathcal{S}^n_+ + \mathcal{N}^n$ and cutting plane methods for some DNN problems. It is found that the proposed methods with expanded SD bases are computationally efficient and accurate for both identification and DNN problems.

The organization of this paper is as follows. Various approximations of $\mathcal{S}^n_+$ are introduced in section 2, including those based on the factor width by Boman et al. [10], diagonal dominance by Ahmadi et al. [1]; and SD bases by Tanaka and Yoshise [40]. The main results of this paper, i.e., an expansion of SD bases and an analysis of its theoretical properties, are provided in section 3. In section 4, we introduce the method of Tanaka and Yoshise [40] for identifying $\mathcal{S}^n_+ + \mathcal{N}^n$ and the new method with expanded SD bases. We also describe the results of numerical experiments. The approximations using expanded SD bases are also applied to cutting plane methods for DNN problems.

2 Some approximations of the semidefinite cone

2.1 Factor width approximation

In [10], Boman et al. defined a concept called factor width.

Definition 2.1. (Definition 1 in [10]) *The factor width of a real symmetric matrix $A \in \mathbb{S}^n$ is the smallest integer k such that there exists a real matrix $V \in \mathbb{R}^{n \times m}$ where $A = VV^T$ and each column of V contains at most k nonzero elements.*

For $k \in \{1, 2, \ldots, n\}$, we can also define

$$\mathcal{FW}(k) := \{X \in \mathbb{S}^n \mid \text{X has a factor width of at most } k\}.$$

It is obvious that the factor width is only defined for semidefinite matrices, because for every matrix A in Definition 2.1, the decomposition $A = VV^T$ implies that $A \in \mathcal{S}^n_+$. Therefore, for every $k \in \{1, 2, \ldots, n\}$, the set of matrices with a factor width of at most k gives an inner approximation of $\mathbb{S}^n_+$: $\mathcal{FW}(k) \subseteq \mathbb{S}^n_+$.

2.2 Diagonal dominance approximation

In [1] and [2], the authors approximated the cone $\mathcal{S}^n_+$ with the set of diagonally dominant matrices and the set of scaled diagonally dominant matrices.

Definition 2.2. *The set of diagonally dominant matrices $\mathcal{DD}_n$ and the set of scaled diagonally dominant matrices $\mathcal{SDD}_n$ are defined as follows:*

$$\mathcal{DD}_n := \{A \in \mathbb{S}^n \mid \ A_{i,i} \geq \sum_{j \neq i} |A_{i,j}| \quad (i = 1, 2, \ldots, n)\},$$
$$\mathcal{SDD}_n := \{A \in \mathbb{S}^n \mid DAD \in \mathcal{DD}_n \text{ for some positive diagonal matrix } D\}.$$

It is easy to see that $\mathcal{DD}_n$ is a convex cone and $\mathcal{SDD}_n$ is a cone in $\mathbb{S}^n$. As a consequence of the Gershgorin circle theorem [17], we have the relation $\mathcal{DD}_n \subseteq \mathcal{SDD}_n \subseteq \mathcal{S}^n_+$. Ahmadi et al. [1] defined $\mathcal{U}_{n,k}$ as the set of vectors in $\mathbb{R}^n$ with at most k nonzeros, each equal to 1 or -1. They also defined a set of matrices $U_{n,k} := \{uu^T \mid u \in \mathcal{U}_{n,k}\}$. Barker and Carlson [6] proved the following theorem.

Theorem 2.3. (Barker and Carlson [6]) $\mathcal{DD}_n = \text{cone}(U_{n,2})$.

The conical hull of a given set $\mathcal{K} \subseteq \mathbb{S}^n$ is defined as $\text{cone}(\mathcal{K}) := \{\sum_{i=1}^k \alpha_i X_i \mid X_i \in \mathcal{K}, \alpha_i \geq 0, k \in \mathbb{Z}_{\geq 0}\}$, where $\mathbb{Z}_{\geq 0}$ is the set of nonnegative integers. A cone generated in this way by a finite number of elements is called *finitely generated*. Theorem 2.3 implies that $\mathcal{DD}_n$ has n^2 extreme rays; thus, it is a finitely generated cone.

A cone $\mathcal{K} \in \mathbb{S}^n$ is *polyhedral* if $\mathcal{K} = \{X \in \mathbb{S}^n \mid \langle A_i, X \rangle \leq 0\}$ for some $A_i \in \mathbb{S}^n$. The following theorem follows from the results of Minkowski [33] and Weyl [44].

Theorem 2.4. (Minkowski-Weyl theorem, see Corollary 7.1a in [38]) *A convex cone is polyhedral if and only if it is finitely generated.*

The above theorem ensures that $\mathcal{DD}_n$ is a polyhedral cone. Using the expression in Theorem 2.3, Ahmadi et al. proved that optimization problems over $\mathcal{DD}_n$ can be solved as LPs. They also proved that optimization problems over $\mathcal{SDD}_n$ can be solved as SOCPs. They designed a column generation method using $\mathcal{DD}_n$ and $\mathcal{SDD}_n$ to obtain a series of inner approximations of $\mathcal{S}^+_n$. As for the relation between the factor width and diagonal dominance, useful results were presented in [10] and in [2], which gives a relation between $\mathcal{SDD}_n$ and the set of matrices with a factor width of at most 2.

Lemma 2.5. (See [10] and Theorem 8 in [2]) $\mathcal{FW}(2) = \mathcal{SDD}_n$

2.3 SD basis approximation

Tanaka and Yoshise defined semidefinite (SD) bases [40].

Definition 2.6. (Definitions 1 and 2 in [40]) *Let the set of orthogonal matrices in* $\mathbb{R}^{n\times n}$ *be denoted by* $\mathcal{O}^n$*. Given a matrix* $P = (p_1, .., p_n) \in \mathcal{O}^n$*,*

$$\mathcal{B}_+(P) := \{(p_i + p_j)(p_i + p_j)^T \mid 1 \le i \le j \le n\}$$

is called an SD basis of Type I.

$$\mathcal{B}_-(P) := \{(p_i + p_i)(p_i + p_i)^T \mid 1 \le i \le n\} \cup \{(p_i - p_j)(p_i - p_j)^T \mid 1 \le i < j \le n\}$$

is called an SD basis of Type II. Matrices in SD bases Type I and II are defined as

$$B^+_{i,j}(P) := (p_i + p_j)(p_i + p_j)^T, \ B^-_{i,j}(P) := (p_i - p_j)(p_i - p_j)^T.$$

As shown in [40], $\mathcal{B}_+(\mathcal{P})$ and $\mathcal{B}_-(\mathcal{P})$ are subsets of $\mathcal{S}^n_+$ and bases on $\mathbb{S}^n$. Given a set $\mathcal{K} \subseteq \mathbb{S}^n$, we define the dual cone of $\mathcal{K}$ as $(\mathcal{K})^* := \{A \in \mathbb{S}^n \mid \langle A, B\rangle \ge 0 \text{ for any } B \in \mathcal{K}\}$. The conical hull of $\mathcal{B}_+(P) \cup \mathcal{B}_-(P)$ and its dual give an inner and an outer polyhedral approximation of $\mathcal{S}^n_+$, as follows.

Definition 2.7. *Let* $\mathcal{O}^n$ *denote the set of orthogonal matrices in* $\mathbb{R}^{n\times n}$ *and let* $P = (p_1, .., p_n) \in \mathcal{O}^n$*. The inner and outer approximations of* $\mathcal{S}^n_+$ *by using SD bases are defined as*

$$\mathcal{S}_{\rm in} := \text{cone}(\mathcal{B}_+(P) \cup \mathcal{B}_-(P)), \quad \mathcal{S}_{\rm out} := (\mathcal{S}_{\rm in})^*.$$

By Definition 2.6, we know that $\mathcal{B}_+(P), \mathcal{B}_-(P) \subseteq \mathcal{S}^n_+$. Since $\mathcal{S}^n_+$ is a convex cone, we have $\mathcal{S}_{\rm in} \subseteq \text{cone}(\mathcal{S}^n_+) = \mathcal{S}^n_+$. By Lemma 1.7.3 in [31], we know that $\mathcal{S}^n_+$ is self-dual; that is, $\mathcal{S}^n_+ = (\mathcal{S}^n_+)^*$. Accordingly, the following simple calculation,

$$\begin{aligned}\mathcal{S}^n_+ = (\mathcal{S}^n_+)^* =& \{X \in \mathbb{S}^n \mid \forall Y \in \mathcal{S}^n_+, \langle X, Y\rangle \ge 0\} \\ \subseteq& \{X \in \mathbb{S}^n \mid \forall Y \in \mathcal{S}_{\rm in}, \langle X, Y\rangle \ge 0\} \text{ (since } \mathcal{S}_{\rm in} \subseteq \mathcal{S}^n_+) \\ =& (\mathcal{S}_{\rm in})^* = \mathcal{S}_{\rm out},\end{aligned}$$

enables us to conclude that $\mathcal{S}_{\rm in} \subseteq \mathcal{S}^n_+ \subseteq \mathcal{S}_{\rm out}$.

3 Expansion of SD bases

When we use the SD bases for approximating $\mathcal{S}^n_+$, the sparsity of the matrices in those bases is quite important in terms of computational efficiency. As we can see in Definition 2.6, the sparsity of the matrices in the SD bases depends on how we choose an orthogonal matrix P. If we choose the identity matrix I as P, then we obtain relatively sparse SD bases. However, the choice $P = I$ is rather limited. In this section, we try to extend the definition of the SD bases in order to obtain various sparse SD bases which will lead us to sparse polyhedral approximations of $\mathcal{S}^n_+$.

3.1 SD bases and their relations with $\mathcal{S}^n_+$ and $\mathcal{DD}_n$

First, we prove a lemma that provides an expression of $\mathcal{S}^n_+$ by using SD bases.

Lemma 3.1.

$$\mathcal{S}^n_+ = \text{cone}\left(\bigcup_{P\in\mathcal{O}^n} \mathcal{B}_+(P)\right) = \text{cone}\left(\bigcup_{P\in\mathcal{O}^n} \mathcal{B}_-(P)\right),$$

where $\mathcal{O}^n$ is the set of orthogonal matrices in $\mathbb{R}^{n\times n}$.

Proof. We only prove the result for $\mathcal{B}_+(P)$, since the proof for $\mathcal{B}_-(P)$ is similar.

First, we prove that $\text{cone}\left(\bigcup_{P\in\mathcal{O}^n} \mathcal{B}_+(P)\right) \subseteq \mathcal{S}^n_+$. By Definition 2.6 for $\mathcal{B}_+(P)$, we see that $\mathcal{B}_+(P) \subseteq \mathcal{S}^n_+$. This implies that $\bigcup_{P\in\mathcal{O}^n} \mathcal{B}_+(P) \subseteq \mathcal{S}^n_+$. Since $\mathcal{S}^n_+$ is a convex cone, we have

$$\text{cone}\left(\bigcup_{P\in\mathcal{O}^n} \mathcal{B}_+(P)\right) \subseteq \text{cone}\left(\mathcal{S}^n_+\right) = \mathcal{S}^n_+.$$

Next, we show that $\mathcal{S}^n_+ \subseteq \text{cone}\left(\bigcup_{P\in\mathcal{O}^n} \mathcal{B}_+(P)\right)$. For every $X \in \mathcal{S}^n_+$, there exists a $P = (p_1, .., p_n) \in \mathcal{O}^n$ and $\lambda_i \geq 0$ $(i = 1, 2, \dots, n)$ such that $X = \sum_{i=1}^n \lambda_i p_i p_i^T$. Since X is represented by

$$X = \sum_{i=1}^n \lambda_i p_i p_i^T = \sum_{i=1}^n \frac{\lambda_i}{4}(p_i + p_i)(p_i + p_i)^T,$$

we can conclude that $X \in \text{cone}\left(\bigcup_{P\in\mathcal{O}^n} \mathcal{B}_+(P)\right)$. □

Lemma 3.1 gives a way to approximate $\mathcal{S}^n_+$ by changing the matrix $P = (p_1, .., p_n) \in \mathcal{O}^n$ when creating SD bases. However, a dense matrix $P \in \mathcal{O}^n$ may lead to a dense SD basis, which is unattractive from the standpoint of computational efficiency. Convex cones generated by sparse SD bases $\mathcal{B}_+(I)$ and $\mathcal{B}_-(I)$ are considered in what follows.

Proposition 3.2. *When using the identity matrix I as the orthogonal matrix P, we have*

$$\text{cone}(\mathcal{B}_+(I) \cup \mathcal{B}_-(I)) = \mathcal{DD}_n.$$

Proof. First, we show that $\mathcal{DD}_n \subseteq \text{cone}(\mathcal{B}_+(I) \cup \mathcal{B}_-(I))$. For every $X \in \mathcal{DD}_n$, consider the following matrix:

$$X^1 := \sum_{i<j} |X_{i,j}|(e_i + \text{sgn}(X_{i,j})e_j)(e_i + \text{sgn}(X_{i,j})e_j)^T, \tag{2}$$

where $I = (e_1, ..., e_n)$, $\text{sgn}(a) \in \{-1, 1\}$ gives the sign of $a \in \mathbb{R}$. From the definitions of $\mathcal{B}_+(I)$ and $\mathcal{B}_-(I)$ in Definition 2.6, we see that

$$(e_i + \text{sgn}(X_{i,j})e_j)(e_i + \text{sgn}(X_{i,j})e_j)^T \in \mathcal{B}_+(I) \cup \mathcal{B}_-(I) \quad (1 \leq i < j \leq n). \tag{3}$$

A simple calculation ensures that

$$(X - X^1)_{i,j} = \begin{cases} 0 & (i \neq j), \\ X_{i,i} - \sum_{k \neq i} |X_{i,k}| & (i = j). \end{cases}$$

Since $X \in \mathcal{DD}_n$, by the Definition 2.2 of $\mathcal{DD}_n$, we find that for every $i = 1, 2, \ldots, n$:

$$(X - X^1)_{i,i} = X_{i,i} - \sum_{j \neq i} |X_{i,j}| \geq 0.$$

Thus, $X^2 := X - X^1$ is a nonnegative diagonal matrix, and

$$X^2 = \sum_{i=1}^{n} \frac{(X - X^1)_{i,i}}{4} (e_i + e_i)(e_i + e_i)^T \in \mathrm{cone}(\mathcal{B}_+(I) \cup \mathcal{B}_-(I)). \tag{4}$$

It follows from (2)-(4) that $X = X^1 + X^2 \in \mathrm{cone}(\mathcal{B}_+(I) \cup \mathcal{B}_-(I))$.

Next, we show that $\mathrm{cone}(\mathcal{B}_+(I) \cup \mathcal{B}_-(I)) \subseteq \mathcal{DD}_n$. It is easy to see that $(e_i + e_j)(e_i + e_j)^T$ and $(e_i - e_j)(e_i - e_j)^T$ are diagonally dominant. Thus, $\mathcal{B}_+(I) \cup \mathcal{B}_-(I) \subseteq \mathcal{DD}_n$. Since Definition 2.2 of $\mathcal{DD}_n$ implies that $\mathcal{DD}_n$ is a convex cone, we have

$$\mathrm{cone}(\mathcal{B}_+(I) \cup \mathcal{B}_-(I)) \subseteq \mathrm{cone}(\mathcal{DD}_n) = \mathcal{DD}_n.$$

□

3.2 Expansion of SD bases without losing sparsity

The previous section shows that we can obtain a sparse polyhedral approximation of $\mathcal{S}^n_+$ by choosing I as P in the SD bases. In this section, we try to extend the definition of the SD bases in order to obtain various sparse polyhedral approximations of $\mathcal{S}^n_+$.

Definition 3.3. *Let $P = (p_1, .., p_n) \in \mathcal{O}^n$. Define the expansion of the SD basis with P and one parameter $\alpha \in \mathbb{R}$ as*

$$\begin{aligned} \bar{B}_{i,j}(\alpha, P) &:= (p_i + \alpha p_j)(p_i + \alpha p_j)^T, \\ \bar{\mathcal{B}}(\alpha, P) &:= \{\bar{B}_{i,j}(\alpha, P) \mid 1 \leq i \leq j \leq n\}. \end{aligned}$$

The proposition below ensures that the expansion of the SD bases also gives bases of $\mathbb{S}^n$.

Proposition 3.4. *Let $P = (p_1, .., p_n) \in \mathcal{O}^n$. For any $\alpha \in \mathbb{R} \setminus \{0, -1\}$, $\bar{\mathcal{B}}(\alpha, P)$ is a set of $n(n+1)/2$ independent matrices and thus a basis of $\mathbb{S}^n$.*

Proof. Let $P \in \mathcal{O}^n$ and $\alpha \in \mathbb{R} \setminus \{0, -1\}$. For every $1 \leq i \leq j \leq n$, $\bar{B}_{i,j}(\alpha, P)$ in Definition 3.3 will be denoted as $\bar{B}_{i,j}$ and $B^+_{i,j}(P)$ in Definition 2.6 will be denoted as $B_{i,j}$. Accordingly, we have

$$\begin{aligned} \bar{B}_{i,j} :=& (p_i + \alpha p_j)(p_i + \alpha p_j)^T \\ =& p_i p_i^T + \alpha(p_i p_j^T + p_j p_i^T) + \alpha^2 p_j p_j^T \\ =& \alpha(p_i p_i^T + p_i p_j^T + p_j p_i^T + p_j p_j^T) + (1 - \alpha) p_i p_i^T + (\alpha^2 - \alpha) p_j p_j^T \\ =& \alpha B_{i,j} + \frac{1 - \alpha}{4} B_{i,i} + \frac{\alpha(\alpha - 1)}{4} B_{j,j}, \end{aligned} \tag{5}$$

and for every $1 \leq i \leq n$, we also have

$$\begin{aligned}\bar{B}_{i,i} :=& (p_i + \alpha p_i)(p_i + \alpha p_i)^T \\ =& (1+\alpha)^2 p_i p_i^T = \frac{(1+\alpha)^2}{4} B_{i,i}. \end{aligned} \tag{6}$$

Suppose that there exist $\gamma_{i,j} \geq 0$ $(1 \leq i \leq j \leq n)$ such that

$$\sum_{1\leq i\leq j\leq n} \gamma_{i,j} \bar{B}_{i,j} = O.$$

Then, by (5) and (6), we see that

$$\begin{aligned} O =& \sum_{i=1}^{n} \frac{\gamma_{i,i}(1+\alpha)^2}{4} B_{i,i} + \sum_{1\leq i<j\leq n} \gamma_{i,j} \left[\alpha B_{i,j} + \frac{1-\alpha}{4} B_{i,i} + \frac{\alpha(\alpha-1)}{4} B_{j,j}\right] \\ =& \sum_{i=1}^{n} \frac{(1+\alpha)^2}{4} \gamma_{i,i} B_{i,i} + \sum_{1\leq i<j\leq n} \alpha\gamma_{i,j} B_{i,j} + \sum_{i=1}^{n-1} \frac{1-\alpha}{4} (\sum_{j=i+1}^{n} \gamma_{i,j}) B_{i,i} \\ &+ \sum_{j=2}^{n} \frac{\alpha(\alpha-1)}{4} (\sum_{i=1}^{j-1} \gamma_{i,j}) B_{j,j} \\ =& \left[\frac{\gamma_{1,1}(1+\alpha)^2}{4} + \frac{1-\alpha}{4} (\sum_{j=2}^{n} \gamma_{1,j})\right] B_{1,1} \\ &+ \sum_{i=2}^{n-1} \left[\frac{(1+\alpha)^2}{4} \gamma_{i,i} + \frac{1-\alpha}{4} (\sum_{j=i+1}^{n} \gamma_{i,j}) + \frac{\alpha(\alpha-1)}{4} (\sum_{j=1}^{i-1} \gamma_{j,i})\right] B_{i,i} \\ &+ \left[\frac{\gamma_{n,n}(1+\alpha)^2}{4} + \frac{\alpha(\alpha-1)}{4} (\sum_{j=1}^{n-1} \gamma_{j,n})\right] B_{n,n} \\ &+ \sum_{1\leq i<j\leq n} \alpha\gamma_{i,j} B_{i,j}. \end{aligned} \tag{7}$$

Since $\{B_{i,j}\} = \mathcal{B}_+(P)$ is a set of linearly independent matrices, all the coefficients for $B_{i,j}$ in (7) should be 0. Thus, we have

$$0 = \frac{\gamma_{1,1}(1+\alpha)^2}{4} + \frac{1-\alpha}{4} (\sum_{j=2}^{n} \gamma_{1,j}), \tag{8}$$

$$0 = \frac{(1+\alpha)^2}{4} \gamma_{i,i} + \frac{1-\alpha}{4} (\sum_{j=i+1}^{n} \gamma_{i,j}) + \frac{\alpha(\alpha-1)}{4} (\sum_{j=1}^{i-1} \gamma_{j,i}) \ (2 \leq i \leq n-1), \tag{9}$$

$$0 = \frac{\gamma_{n,n}(1+\alpha)^2}{4} + \frac{\alpha(\alpha-1)}{4} (\sum_{j=1}^{n-1} \gamma_{j,n}), \tag{10}$$

$$0 = \alpha\gamma_{i,j} \ (1 \leq i < j \leq n). \tag{11}$$

Since $\alpha \neq 0$, by (11) we have

$$\gamma_{i,j} = 0 \ (1 \leq i < j \leq n). \tag{12}$$

Since $\alpha \neq -1$, (8)-(12) imply that

$$\gamma_{i,i} = 0 \ (i = 1, 2, \ldots, n). \tag{13}$$

By we can conclude that $\{\bar{B}_{i,j}\} = \bar{\mathcal{B}}_+(P)$ is a set of $n(n+1)/2$ linearly independent matrices. □

If we let $\alpha = 1$, then it is straightforward that $\bar{\mathcal{B}}(1, P) = \mathcal{B}_+(P)$. If we let α be other real numbers, we may obtain different SD bases. The following proposition gives the condition for generating different expanded SD bases.

Proposition 3.5. *Let $P = (p_1, .., p_n) \in \mathcal{O}^n$. Suppose that $\alpha_1 \in \mathbb{R} \setminus \{0, -1\}$ and $\alpha_2 \in \mathbb{R} \setminus \{0, \alpha_1\}$. Then, for every $1 \leq i < j \leq n$,*

$$(p_i + \alpha_2 p_j)(p_i + \alpha_2 p_j)^T \notin \text{cone}(\bar{\mathcal{B}}(\alpha_1, P)).$$

Proof. Let us define

$$\bar{B}^1_{i,j} := (p_i + \alpha_1 p_j)(p_i + \alpha_1 p_j)^T, \quad \bar{B}^2_{i,j} := (p_i + \alpha_2 p_j)(p_i + \alpha_2 p_j)^T \quad (1 \leq i \leq j \leq n).$$

Note that if $i = j$, then

$$\bar{B}^1_{i,i} := (1 + \alpha_1)^2 p_i p_i^T, \quad \bar{B}^2_{i,i} := (1 + \alpha_2)^2 p_i p_i^T. \tag{14}$$

For every $i < j$, we can write $\bar{B}^2_{i,j}$ as a linear combination of $\bar{B}^1_{i,j}$:

$$\begin{aligned}
\bar{B}^2_{i,j} =& p_i p_i^T + \alpha_2^2 p_j p_j^T + \alpha_2 (p_i p_j^T + p_j p_i^T) \\
=& p_i p_i^T + \alpha_2^2 p_j p_j^T + \frac{\alpha_2}{\alpha_1} \alpha_1 (p_i p_j^T + p_j p_i^T) \quad (\text{because } \alpha_1 \neq 0) \\
=& p_i p_i^T + \alpha_2^2 p_j p_j^T - \frac{\alpha_2}{\alpha_1} p_i p_i^T - \frac{\alpha_2 \alpha_1^2}{\alpha_1} p_j p_j^T \\
& + \frac{\alpha_2}{\alpha_1} \left[p_i p_i^T + \alpha_1 (p_i p_j^T + p_j p_i^T) + \alpha_1^2 p_j p_j^T \right] \\
=& \frac{\alpha_1 - \alpha_2}{\alpha_1} p_i p_i^T + \alpha_2 (\alpha_2 - \alpha_1) p_j p_j^T + \frac{\alpha_2}{\alpha_1} \bar{B}^1_{i,j} \\
=& \frac{\alpha_1 - \alpha_2}{\alpha_1 (1 + \alpha_1)^2} (1 + \alpha_1)^2 p_i p_i^T + \frac{\alpha_2 (\alpha_2 - \alpha_1)}{(1 + \alpha_1)^2} (1 + \alpha_1)^2 p_j p_j^T + \frac{\alpha_2}{\alpha_1} \bar{B}^1_{i,j} \\
& (\text{because } \alpha_1 \neq -1) \\
=& \frac{\alpha_1 - \alpha_2}{\alpha_1 (1 + \alpha_1)^2} \bar{B}^1_{i,i} + \frac{\alpha_2 (\alpha_2 - \alpha_1)}{(1 + \alpha_1)^2} \bar{B}^1_{j,j} + \frac{\alpha_2}{\alpha_1} \bar{B}^1_{i,j} \ (\text{by (14)}).
\end{aligned} \tag{15}$$

Since $\alpha_1 \notin \{0, -1\}$, Proposition 3.4 ensures that $\bar{\mathcal{B}}(\alpha_1, P)$ is linearly independent, and hence, the expression (15) for $\bar{B}^2_{i,j}$ is unique.

Suppose that $\bar{B}^2_{i,j} \in \text{cone}(\bar{\mathcal{B}}(\alpha_1, P))$. In this case, all the coefficients in (15) should be nonnegative, which implies that

$$\frac{\alpha_1 - \alpha_2}{\alpha_1 (1 + \alpha_1)^2} \geq 0, \ \frac{\alpha_2 (\alpha_2 - \alpha_1)}{(1 + \alpha_1)^2} \geq 0, \ \frac{\alpha_2}{\alpha_1} > 0. \tag{16}$$

From the last inequality in (16), we have either

$$\text{(i) } \alpha_1, \alpha_2 > 0 \quad \text{or} \quad \text{(ii) } \alpha_1, \alpha_2 < 0.$$

For case (i), from the first and second inequalities of (16), we have $\alpha_2 - \alpha_1 \geq 0$ and $\alpha_1 - \alpha_2 \geq 0$ which implies $\alpha_2 = \alpha_1$ and contradicts the assumption $\alpha_2 \neq \alpha_1$. A similar contradiction is obtained for case (ii). Thus, we have $\bar{B}^2_{i,j} \notin \text{cone}(\bar{\mathcal{B}}(\alpha_1, P))$. □

3.3 Expression of $\mathcal{SDD}_n$ with expanded SD bases

In this section, we show that the conical hull of the union of the extended SD bases $\bar{\mathcal{B}}(\alpha, P)$ on $\alpha \in \mathbb{R}$ coincides with the set of scaled diagonally dominant matrices $\mathcal{SDD}_n$.

Proposition 3.6. *Let $I = (e_1, .., e_n) \in \mathcal{O}^n$. Then we have*

$$\text{cone}\left(\bigcup_{\alpha \in \mathbb{R}} \bar{\mathcal{B}}(\alpha, I)\right) = \mathcal{SDD}_n.$$

Proof. In what follows, we show that

$$\text{cone}\left(\bigcup_{\alpha \in \mathbb{R}} \bar{\mathcal{B}}(\alpha, I)\right) = \mathcal{FW}(2),$$

where $\mathcal{FW}(k)$ is defined in Definition 2.1. Then, the assertion of the proposition follows from Lemma 2.5.

Let us show that $\mathcal{FW}(2) \subseteq \text{cone}\left(\bigcup_{\alpha \in \mathbb{R}} \bar{\mathcal{B}}(\alpha, I)\right)$. For any $X \in \mathcal{FW}(2)$, there exists a $V \in \mathbb{R}^{n \times m}$ where $X = VV^T$ and each column contains at most 2 nonzero elements. Denote the columns of V as $v_i (i = 1, 2, \ldots, m)$. Here, we can assume that $v_i (i = 1, 2, \ldots, m)$ are nonzero vectors without any loss of generality. Thus, we see that there exist $p(i), q(i) \in \{1, 2, \ldots, n\} (i = 1, 2, \ldots, m)$ and $\alpha_{p(i)}, \alpha_{q(i)} \in \mathbb{R}$, either of which is nonzero for every $i = 1, 2, \ldots, m$ satisfying

$$v_i = \alpha_{p(i)} e_{p(i)} + \alpha_{q(i)} e_{q(i)}.$$

For every $i = 1, 2, \ldots, m$, if $\alpha_{p(i)} = 0$, we have

$$v_i v_i^T = \alpha^2_{q(i)} e_{q(i)} e^T_{q(i)} \in \text{cone}\left(\bigcup_{\alpha \in \mathbb{R}} \bar{\mathcal{B}}(\alpha, I)\right), \tag{17}$$

and if $\alpha_{p(i)} \neq 0$, we have

$$\begin{aligned} v_i v_i^T =& \alpha^2_{p(i)} (e_{p(i)} + \frac{\alpha_{q(i)}}{\alpha_{p(i)}} e_{q(i)})(e_{p(i)} + \frac{\alpha_{q(i)}}{\alpha_{p(i)}} e_{q(i)})^T \\ &\in \text{cone}\left(\bigcup_{\alpha \in \mathbb{R}} \bar{\mathcal{B}}(\alpha, I)\right). \end{aligned} \tag{18}$$

By (17) and (18), we conclude that

$$X = VV^T = \sum_{i=1}^{m} v_i v_i^T \in \text{cone}\left(\bigcup_{\alpha\in\mathbb{R}} \bar{\mathcal{B}}(\alpha, I)\right).$$

Next, we show that cone $\left(\bigcup_{\alpha\in\mathbb{R}} \bar{\mathcal{B}}(\alpha, I)\right) \subseteq \mathcal{FW}(2)$.

Suppose that $X \in \text{cone}\left(\bigcup_{\alpha\in\mathbb{R}} \bar{\mathcal{B}}(\alpha, I)\right)$. Then there exist some positive integer k_1, and $\lambda_{ij}^k \geq 0, \alpha^k \in \mathbb{R}$ $(1 \leq i \leq j \leq n,\ 1 \leq k \leq k_1)$ such that

$$\begin{aligned} X &= \sum_{\substack{1\leq i\leq j\leq n,\\ k=1,..,k_1}} \lambda_{ij}^k (e_i + \alpha^k e_j)(e_i + \alpha^k e_j)^T. \\ &= \sum_{\substack{1\leq i\leq j\leq n,\\ k=1,..,k_1}} (\sqrt{\lambda_{ij}^k} e_i + \alpha^k \sqrt{\lambda_{ij}^k} e_j)(\sqrt{\lambda_{ij}^k} e_i + \alpha^k \sqrt{\lambda_{ij}^k} e_j)^T. \end{aligned}$$

Define a vector $v(i,j,k) := \sqrt{\lambda_{ij}^k} e_i + \alpha^k \sqrt{\lambda_{ij}^k} e_j$ for any $1 \leq i \leq j \leq n$ and $k \in \{1, .., k_1\}$. Then, $v(i,j,k)$ has at most two nonzero elements, so we can obtain a matrix $V \in \mathbb{R}^{n\times \frac{k_1 n(n+1)}{2}}$ whose columns are $v(i,j,k)$. Then,

$$X = \sum_{\substack{1\leq i\leq j\leq n,\\ k=1,..,k_1}} v(i,j,k)v(i,j,k)^T = VV^T$$

and by Definition 2.1, $X \in \mathcal{FW}(2)$. □

3.4 Notes on the parameter α

Here, we discuss the choice for the parameter α to increase the volume of the polyhedral approximation cone($\bar{\mathcal{B}}(\alpha, P)$) of the semidefinite cone $\mathcal{S}_+^n$. For any $P = (p_1, .., p_n) \in \mathcal{O}^n$, $\alpha \in \mathbb{R}$ and $1 \leq i < j \leq n$, by Definition 3.3, we can calculate the Frobenius norm of $\bar{B}_{i,j}(\alpha, P)$:

$$\begin{aligned} \|\bar{B}_{i,j}(\alpha, P)\| &= \|(p_i + \alpha p_j)(p_i + \alpha p_j)^T\| \\ &= \sqrt{\text{Trace}\left((p_i + \alpha p_j)(p_i + \alpha p_j)^T (p_i + \alpha p_j)(p_i + \alpha p_j)^T\right)} \\ &= \|p_i + \alpha p_j\|^2 \\ &= 1 + \alpha^2 \quad (\text{since } P \in \mathcal{O}^n). \end{aligned} \tag{19}$$

According to Proposition 3.5, by changing α, one can obtain different polyhedral approximations. However, we can see that

$$\begin{aligned}\lim_{|\alpha|\to\infty}\frac{\bar{B}_{i,j}(\alpha,P)}{\|\bar{B}_{i,j}(\alpha,P)\|} &= \lim_{|\alpha|\to\infty}\frac{1}{1+\alpha^2}(p_i+\alpha p_j)(p_i+\alpha p_j)^T \text{ (by (19))},\\ &= \lim_{|\alpha|\to\infty}\left[\frac{1}{1+\alpha^2}p_ip_i^T+\frac{\alpha}{1+\alpha^2}(p_ip_j^T+p_jp_i^T)+\frac{\alpha^2}{1+\alpha^2}p_jp_j^T\right]\\ &= p_jp_j^T=\frac{1}{4}B^+_{j,j}(P),\end{aligned}$$

and by Definitions 2.6 and 3.3, we have

$$\bar{B}_{i,j}(0,P)=\frac{1}{4}B^+_{i,i}(P),\ \bar{B}_{i,j}(1,P)=B^+_{i,j}(P),\ \bar{B}_{i,j}(-1,P)=B^-_{i,j}(P).$$

This shows that, if $|\alpha|\to\infty$ or $\alpha\in\{0,1,-1\}$, the new matrix $\bar{B}_{i,j}(\alpha,P)$ will become close to the existing matrices, e.g. $B^+_{i,i}(P)$, $B^+_{j,j}(P)$, $B^+_{i,j}(P)$ and $B^-_{i,j}(P)$, and the volume of the polyhedral approximation cone$(\bar{\mathcal{B}}(\alpha,P)\cup\mathcal{B}_+(P)\cup\mathcal{B}_-(P))$ of the semidefinite cone $\mathcal{S}^n_+$ will also be close to the volume of the existing inner approximation cone$(\mathcal{B}_+(P)\cup\mathcal{B}_-(P))$ of $\mathcal{S}^n_+$.

Given an $\alpha\in\mathbb{R}$ we can define the angles between matrices in the expanded SD bases and SD bases Type I and II for every $1\le i<j\le n$, as follows:

$$\begin{aligned}\theta_1(\alpha) &:= \arccos\frac{\langle\bar{B}_{i,j}(\alpha,P),B^+_{i,i}(P)\rangle}{\|\bar{B}_{i,j}(\alpha,P)\|\|B^+_{i,i}(P)\|},\ \theta_2(\alpha):=\arccos\frac{\langle\bar{B}_{i,j}(\alpha,P),B^+_{j,j}(P)\rangle}{\|\bar{B}_{i,j}(\alpha,P)\|\|B^+_{j,j}(P)\|},\\ \theta_3(\alpha) &:= \arccos\frac{\langle\bar{B}_{i,j}(\alpha,P),B^+_{i,j}(P)\rangle}{\|\bar{B}_{i,j}(\alpha,P)\|\|B^+_{i,j}(P)\|},\ \theta_4(\alpha):=\arccos\frac{\langle\bar{B}_{i,j}(\alpha,P),B^-_{i,j}(P)\rangle}{\|\bar{B}_{i,j}(\alpha,P)\|\|B^-_{i,j}(P)\|}.\end{aligned}$$

Thus, we have

$$\begin{aligned}\cos\theta_1(\alpha) =& \frac{\langle\bar{B}_{i,j}(\alpha,P),B^+_{i,i}(P)\rangle}{\|\bar{B}_{i,j}(\alpha,P)\|\|B^+_{i,i}(P)\|}\\ =& \frac{\langle(p_i+\alpha p_j)(p_i+\alpha p_j)^T,(p_i+p_i)(p_i+p_i)^T\rangle}{(1+\alpha^2)\|(p_i+p_i)(p_i+p_i)^T\|} \text{ (by (19))}\\ =& \frac{4\|p_i\|^4}{(1+\alpha^2)4\|p_i\|^2} \text{ (because } p_i^Tp_j=0)\\ =& \frac{1}{1+\alpha^2} \text{ (because } \|p_i\|=1).\end{aligned}$$

Similarly, we have

$$\cos\theta_2(\alpha)=\frac{\alpha^2}{1+\alpha^2},\ \cos\theta_3(\alpha)=\frac{(1+\alpha)^2}{2(1+\alpha^2)},\ \cos\theta_4(\alpha)=\frac{(1-\alpha)^2}{2(1+\alpha^2)}.$$

In general, to obtain a large enough inner approximation with limited parameters, we prefer an α that makes $\theta_1(\alpha)=\theta_3(\alpha)$, which means that the new matrix $\bar{B}_{i,j}(\alpha,P)$ will be in the middle of $B^+_{i,i}(P)$ and $B^+_{i,j}(P)$ on the boundary of $\mathcal{S}^n_+$. Similarly, we can obtain α by calculating $\theta_2(\alpha)=\theta_3(\alpha)$, $\theta_1(\alpha)=\theta_4(\alpha)$ and $\theta_2(\alpha)=\theta_4(\alpha)$. By solving these equalities, we find that

$$\alpha=\pm1\pm\sqrt{2}.$$

The expansions with these parameters are expected to provide generally large inner approximations for $\mathcal{S}^n_+$.

4 Numerical experiments

4.1 Identification of $A \in \mathcal{S}_+^n + \mathcal{N}^n$

Given a real symmetric matrix $A \in \mathbb{S}^n$, the problem of identifying whether $A \in \mathcal{S}_+^n + \mathcal{N}^n$ or not can be solved (to a specified accuracy ϵ) by using the following doubly nonnegative (DNN) optimization problem:

$$\begin{aligned} \min\ & \langle A, X\rangle \\ \text{s.t. } & \langle X, I\rangle = 1, \\ & X \in \mathcal{S}_+^n \cap \mathcal{N}^n, \end{aligned} \tag{20}$$

where $I \in \mathbb{S}^n$ is an $n \times n$ identity matrix. The DNN problem can be solved with an SDP solver, but the number of variables is $\frac{n(n+1)}{2}$ and the method might be inefficient when n is large. Instead of solving the DNN problem, Tanaka and Yoshise proposed an LP-based method using SD bases to identify $A \in \mathcal{S}_+^n + \mathcal{N}^n$ [40].

They first obtain a spectral decomposition of A with some $P = (p_1, .., p_n) \in \mathcal{O}^n,\ \lambda_i \in \mathbb{R}\ (i = 1, 2, \dots, n)$ such that $A = \sum_{i=1}^n \lambda_i p_i p_i^T$. By introducing parameters $w_i \in \mathbb{R}\ (i = 1, 2, \dots, n)$, A can be decomposed into

$$A = \sum_{i=1}^n (\lambda_i - w_i) p_i p_i^T + \sum_{i=1}^n w_i p_i p_i^T. \tag{21}$$

We know from decomposition that, if

$$\lambda_i - w_i \geq 0, \ \ \sum_{i=1}^n w_i p_i p_i^T \in \mathcal{N}^n\ (i = 1, 2, \dots, n), \tag{22}$$

then $\sum_{i=1}^n (\lambda_i - w_i) p_i p_i^T \in \mathcal{S}_+^n$, and $A \in \mathcal{S}_+^n + \mathcal{N}^n$. Whether 22) is satisfied or not can be determined by the following LP:

$$\begin{array}{llrl} (\mathrm{LP}_0) & \max & \beta & \\ & \text{s.t.} & \lambda_i \geq w_i, & i = 1, 2, \dots, n, \\ & & (\sum_{i=1}^n w_i p_i p_i^T)_{j,k} \geq \beta, & 1 \leq k \leq l \leq n. \end{array}$$

If the optimal value is nonnegative, then condition (22) is satisfied and $A \in \mathcal{S}_+^n + \mathcal{N}^n$.

With regard to (21), Tanaka and Yoshise [40] introduced additional parameters $w_{i,j}^+, w_{i,j}^- \in \mathbb{R}$ and the SD bases to generate the following decomposition of A:

$$\begin{aligned} A = & \sum_{i=1}^n (\lambda_i - w_i) p_i p_i^T + \sum_{i=1}^n w_i p_i p_i^T \\ & + \sum_{1 \leq i < j \leq n} (-w_{i,j}^+) B_{i,j}^+(P) + \sum_{1 \leq i < j \leq n} w_{i,j}^+ B_{i,j}^+(P) \\ & + \sum_{1 \leq i < j \leq n} (-w_{i,j}^-) B_{i,j}^-(P) + \sum_{1 \leq i < j \leq n} w_{i,j}^- B_{i,j}^-(P). \end{aligned} \tag{23}$$

From the decomposition above, if

$$\lambda_i - w_i \geq 0 \ (i = 1, 2, \dots, n), \tag{24}$$
$$w^+_{i,j} \leq 0, \ w^-_{i,j} \leq 0 \ (1 \leq i < j \leq n), \tag{25}$$
$$\sum_{i=1}^{n} w_i p_i p_i^T + \sum_{1 \leq i < j \leq n} (w^+_{i,j} B^+_{i,j}(P) + w^-_{i,j} B^-_{i,j}(P)) \in \mathcal{N}^n, \tag{26}$$

are satisfied, then $A \in \mathcal{S}^n_+ + \mathcal{N}^n$. This leads to the following LP:

$$\begin{array}{llcl}
(\mathrm{LP}_1) & \max & \beta & \\
 & \text{s.t.} & \lambda_i \geq w_i, & i = 1, 2, \dots, n \\
 & & w^+_{i,j} \leq 0, \ w^-_{i,j} \leq 0, & 1 \leq i < j \leq n \\
 & & \left(\sum_{i=1}^{n} w_i p_i p_i^T + \sum_{1 \leq i < j \leq n} (w^+_{i,j} B^+_{i,j}(P) + w^-_{i,j} B^-_{i,j}(P)) \right)_{k,l} \geq \beta, & 1 \leq k \leq l \leq n.
\end{array}$$

If the optimal value is nonnegative, then (24)-(26) are satisfied and $A \in \mathcal{S}^n_+ + \mathcal{N}^n$.

In problem LP_1, the SD bases are generated using the matrix P consisting of eigenvectors of A. Note that P is usually a dense matrix in practice, which is unattractive from the computational viewpoint. Aiming to create other sparse decompositions for A, this section first uses the SD basis generated by the identity matrix I and then the expansion introduced in section 3.3. The resulting LPs are relatively sparse.

We replace the dense matrix P with a sparse matrix I in decomposition (23) and obtain a new decomposition:

$$\begin{aligned}
A = & \sum_{i=1}^{n} (\lambda_i - w_i) p_i p_i^T + \sum_{i=1}^{n} w_i p_i p_i^T \\
& + \sum_{1 \leq i < j \leq n} (-w^+_{i,j}) B^+_{i,j}(I) + \sum_{1 \leq i < j \leq n} w^+_{i,j} B^+_{i,j}(I) \\
& + \sum_{1 \leq i < j \leq n} (-w^-_{i,j}) B^-_{i,j}(I) + \sum_{1 \leq i < j \leq n} w^-_{i,j} B^+_{i,j}(I).
\end{aligned} \tag{27}$$

Here, the SD bases are not related to P, and the decomposition has a relatively sparse structure. Using decomposition (27), we can introduce the following LP:

$$\begin{array}{llcl}
(\mathrm{LP}_2) & \max & \beta & \\
 & \text{s.t.} & \lambda_i \geq w_i, & i = 1, 2, \dots, n \\
 & & w^+_{i,j} \leq 0, \ w^-_{i,j} \leq 0, & 1 \leq i < j \leq n \\
 & & \left(\sum_{i=1}^{n} w_i p_i p_i^T + \sum_{1 \leq i < j \leq n} (w^+_{i,j} B^+_{i,j}(I) + w^-_{i,j} B^-_{i,j}(I)) \right)_{k,l} \geq \beta, & 1 \leq k \leq l \leq n.
\end{array}$$

If the optimal value is nonnegative, then $A \in \mathcal{S}^n_+ + \mathcal{N}^n$.

Let us denote the given set of parameters α in Definition 3.3 by $\mathcal{H} \subseteq \mathbb{R}$. We use the expansion of the

SD bases with parameter $\alpha \in \mathcal{H}$ (Definition 3.3) and obtain a general form for the decompositions of A:

$$\begin{aligned} A = & \sum_{i=1}^{n} (\lambda_i - w_i) p_i p_i^T + \sum_{i=1}^{n} w_i p_i p_i^T \\ & + \sum_{\alpha \in \mathcal{H}} \sum_{1 \le i < j \le n} \left((-w_{i,j}^{\alpha}) \bar{B}_{i,j}(\alpha, I) + w_{i,j}^{\alpha} \bar{B}_{i,j}(\alpha, I) \right), \end{aligned} \tag{28}$$

where $w_{i,j}^{\alpha} \in \mathbb{R}$ $(1 \le i < j \le n)$.

By taking different $\mathcal{H} \subseteq \mathbb{R}$, we may generate different decompositions. Suppose we choose $\mathcal{H} = \{1, -1\}$; decomposition (28) then becomes decomposition (27).

Now let us introduce a general form for LPs corresponding to decomposition (28):

$$\begin{array}{llll} (\mathrm{LP}(\mathcal{H})) & \max & \beta & \\ & \text{s.t.} & \lambda_i \ge w_i, & i = 1, 2, \ldots, n \\ & & w_{i,j}^{\alpha} \le 0, & 1 \le i < j \le n, \ \alpha \in \mathcal{H} \\ & & \left(\sum_{i=1}^{n} w_i p_i p_i^T + \sum_{\alpha \in \mathcal{H}} \sum_{1 \le i < j \le n} w_{i,j}^{\alpha} \bar{B}_{i,j}(\alpha, I) \right)_{k,l} \ge \beta, & 1 \le k \le l \le n. \end{array}$$

If the optimal value is nonnegative, then $A \in \mathcal{S}_+^n + \mathcal{N}^n$. In particular, when $\mathcal{H} = \{1, -1\}$, $\mathrm{LP}(\mathcal{H})$ becomes LP_2.

We conducted numerical experiments to evaluate three methods, namely, LP_0, LP_1, and $\mathrm{LP}(\mathcal{H})$ with several $\mathcal{H} \subseteq \mathbb{R}$, for identifying random $A \in \mathcal{S}_+^n + \mathcal{N}^n$. In reference to Section 4 in [40], 1000 random instances of $A \in \mathcal{S}_+^n + \mathcal{N}^n$ were generated for $n = 10, 20$, and 50. We also performed the identification $A \in \mathcal{S}_+^n + \mathcal{N}^n$ by solving DNN problem (20) with two popular SDP solvers: SeDuMi 1.3 [39] and SDPT3 4.0 [42]. All the experiments were performed with MATLAB 2018b on a PC with an Intel(R) Core(TM) i7-6700 CPU running at 3.4 GHz and 16 GB of RAM. The LPs were solved using Gurobi Optimizer 8.0.0 [20]. SeDuMi 1.3 was performed with the YALMIP [32] interface. We considered the following sets $\mathcal{H}$ of parameters α:

$$\begin{aligned} \mathcal{H}_1 := \{\pm 1\}, \ \mathcal{H}_2 := \{\pm 1, \pm 1 \pm \sqrt{2}\}, \ \mathcal{H}_3 := \{\pm 1, \pm 2, \pm 0.5\}, \\ \mathcal{H}_4 := \{\pm 1, \pm 5, \pm 0.2\}. \end{aligned} \tag{29}$$

Tables 1 and 2 shows the results, where $\#A$ denotes the number of matrices identified by each method, and AT denotes the average CPU time (in seconds) to handle 1000 instances for each n.

Table 1: Numerical results for identification of $A \in \mathcal{S}_+^n + \mathcal{N}^n$.

n	LP_0		LP_1		SeDuMi 1.3		SDPT3 4.0	
	#A	AT(s)	#A	AT(s)	#A	AT(s)	#A	AT(s)
10	262	0.003	1000	0.008	1000	0.079	1000	0.106
20	38	0.010	1000	0.137	1000	0.846	1000	0.327
50	0	0.074	1000	53.882	1000	134.116	1000	3.932

We can see from Tables 1 and 2 that:

Table 2: Numerical results for identification of $A \in \mathcal{S}^n_+ + \mathcal{N}^n$.

n	LP($\mathcal{H}_1$)		LP($\mathcal{H}_2$)		LP($\mathcal{H}_3$)		LP($\mathcal{H}_4$)	
	#A	AT(s)	#A	AT(s)	#A	AT(s)	#A	AT(s)
10	992	0.003	1000	0.004	1000	0.004	1000	0.004
20	848	0.014	1000	0.028	1000	0.029	998	0.029
50	3	0.600	408	1.137	461	1.112	136	1.317

- In all cases, SDPT3 4.0 obtained the best results in terms of the number of identified matrices and average CPU time. On the other hand, LP($\mathcal{H}_2$) and LP($\mathcal{H}_3$) attained more than 40% of SDPT3 4.0s identifications in less than 30% of its average CPU time.
- For any n, the number of matrices identified using LP($\mathcal{H}$) increased in the order $\mathcal{H}_1$, $\mathcal{H}_4$, $\mathcal{H}_2$ and $\mathcal{H}_3$, while the average time spent depended on the number of elements in $\mathcal{H}$. This shows that the proposed expansion of SD bases generates large polyhedral approximations with sparsity. When $n = 10$ and $n = 20$, solving LP($\mathcal{H}_2$) and LP($\mathcal{H}_3$) identified all the matrices in less than 0.05 seconds.
- For any n, the number of matrices identified using LP($\mathcal{H}$) was greater than the number identified using LP$_0$. For $n = 50$, LP($\mathcal{H}$) was around 50 times faster than LP$_1$ in terms of average CPU time, and LP($\mathcal{H}_2$) and LP($\mathcal{H}_3$) each identified around half the matrices.
- SDPT3 4.0 was much faster than SeDuMi 1.3 at solving the DNN problem (20) for the identification with $n = 50$.

4.2 Cutting plane method for DNN problems

In [1], Ahmadi et al. provided a column generation approach, called the cutting plane method, for DNN problems. We consider the standard DNN problem, called the standard cone-LP over the DNN cone in [45]:

$$
\begin{array}{lll}
(P) & \min & \langle C, X\rangle \\
 & \text{s.t.} & \langle A_j, X\rangle = b_j, j = 1, 2, \ldots, m, \\
 & & X \in \mathcal{S}^n_+ \cap \mathcal{N}^n,
\end{array}
$$

where $C, A_j \in \mathbb{S}^n$, $b_j \in \mathbb{R}$. Its dual is given as follows:

$$
\begin{array}{lll}
(D) & \max & b^T y \\
 & \text{s.t.} & C - \displaystyle\sum_{j=1}^{m} y_j A_j \in \mathcal{S}^n_+ + \mathcal{N}^n.
\end{array}
$$

The cutting plane method is based on the fact that $\mathcal{S}^n_+$ is represented by the set $\{X \in \mathbb{S}^n \mid d^T X d \geq 0 \text{ for any } d \in \mathbb{R}^n\}$. The method first selects a finite number of vectors $\{d_1, .., d_k\} \subseteq \mathbb{R}^n$ to obtain an initial outer approximation of the semidefinite cone:

$$
\mathcal{S}_0 := \{X \in \mathbb{S}^n \mid d_i^T X d_i \geq 0 \ (i = 1, .., k)\}.
$$

The following is an initial LP-relaxation problem where $\mathcal{S}^n_+$ in (P) is replaced by $\mathcal{S}_0$:

$$\begin{array}{ll} \min & \langle C, X\rangle \\ \text{s.t.} & \langle A_j, X\rangle = b_j,\ j = 1, 2, \ldots, m, \\ & d_i^T X d_i \geq 0,\ i = 1, .., k, \\ & X \in \mathcal{N}^n. \end{array}$$

By solving the LP-relaxation problem above, we can obtain a lower bound X_0^* for (P). If we add the eigenvector d_{k+1} corresponding to a negative eigenvalue λ^* of X_0^* to $\{d_1, .., d_k\}$, we can obtain a new outer approximation,

$$\mathcal{S}_1 := \{X \in \mathbb{S}^n \mid d_i^T X d_i \geq 0\ (i = 1, .., k+1)\}.$$

Because

$$d_{k+1}^T X d_{k+1} = \lambda^* < 0,$$

we can cut X_0^* out of the feasible region of the new relaxation problem by using $\mathcal{S}_1$. Since the eigenvectors are usually dense, we only have to add eigenvectors corresponding to up to the second smallest eigenvalues to $\{d_i\}$ at every iteration, which increases computational efficiency.

This approach generates a series of LP relaxations, and the lower bounds get close to the optimal value of the original DNN problem. In [1], the authors used the extreme vectors of the set $\mathcal{DD}_n$ of the diagonal dominant matrices as the vectors d_i in the initial outer approximation $\mathcal{S}_0$, and the resultant set $\mathcal{S}_0$ coincided with the set using the SD bases $\bar{\mathcal{B}}(1, I)$ and $\bar{\mathcal{B}}(-1, I)$ as the set of d_i in $\mathcal{S}_0$. In our approach, we use the expanded SD bases as the set of d_i, aiming to get higher computational efficiency.

We generated three random problems of different sizes as follows. First, we generated a primal strictly feasible solution, $X_0 := ee^T + I \in \text{int}(\mathcal{S}^n_+ \cap \mathcal{N}^n)$, and a slack variable $S_0 := ee^T + I$. Each element of the dual feasible solution $y \in \mathbb{R}^m$ and $A \in \mathbb{S}^n$ was chosen from a uniform distribution in $[-1, 1]$. Here, let $C = S_0 + \sum_{j=1}^m y_j A_j$ and $b_j = \langle A_j, X_0\rangle$ $(j = 1, 2, \ldots, m)$. Then (X_0, y) is a pair of strictly feasible primal-dual solutions of the generated DNN problem. By the strong duality theorem of conic optimization problems (see Theorem 3.2.6 in [37]), we know that the resulting DNN problem has no duality gap.

Let CLP($\mathcal{H}$) denote the cutting plane method using $\mathcal{H}$ in (29). Note that the inner approximation using the SD basis with $\mathcal{H}_1$ is equivalent to the set $\mathcal{DD}_n$. We used the value $\left|\frac{f^* - f_k}{f^*}\right| \times 100\%$ with the given lower bound f_k and the optimal value f^* to evaluate the accuracy of each iteration and called it *Interval*. Figure 1 shows the relation between CPU time and Interval for a DNN problem with $n = 100$, $m = 10$. The method with expanded SD bases not only had a better initial accuracy, but also got closer to the optimal value and was faster than the one with $\mathcal{DD}_n$.

Numerical experiments were performed on (P) with $n = 50, 100$ and $m = 10, 50$. We ran the experiment on (P) three times for each combination of n and m and took the average CPU time as the computational efficiency. The results are shown in Figures 3 to 6.

The results in Figures 3 to 6 indicate that:

- SDPT3 4.0 obtained the best results in all cases, whereas CLP($\mathcal{H}_2$) and CLP($\mathcal{H}_3$) attained an accuracy interval of less than 20% in less than 50% of the CPU time taken by SDPT3 4.0.

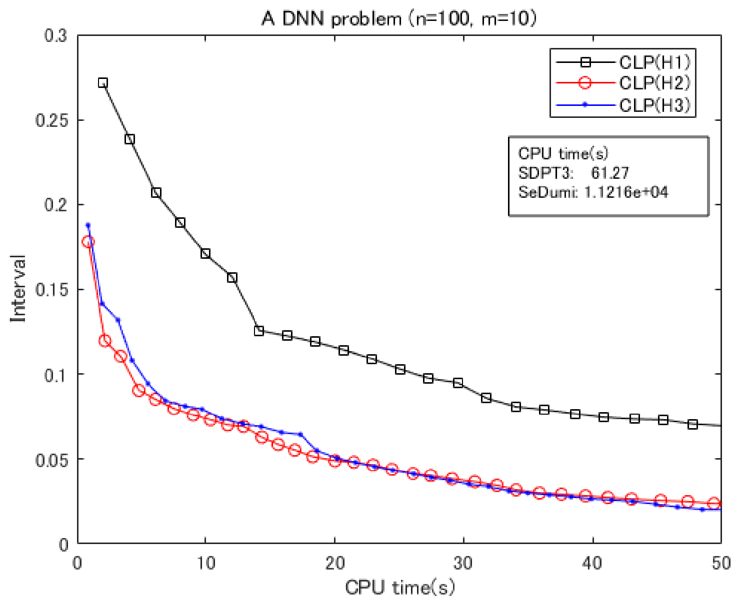

Figure 1: Relation between CPU time and Interval for a DNN problem with n=100,m=10.

Table 3: Numerical results for DNN problems with n=50, m=10.

No.	Ite	DNN Time		CLP($\mathcal{H}_1$)		CLP($\mathcal{H}_2$)		CLP($\mathcal{H}_3$)	
		SDPT3	SeDuMi	Interval	Time	Interval	Time	Interval	Time
1	1	6.81	105.58	21.43%	0.02	11.71%	0.10	13.49%	0.09
	10			8.70%	0.24	4.70%	1.10	5.51%	1.08
	20			6.11%	1.21	2.40%	3.44	2.94%	3.25
				3.68%	3.00	2.64%	3.00	3.03%	3.00
				2.38%	5.00	2.03%	5.00	2.16%	5.00
2	1	6.70	106.51	20.98%	0.02	14.86%	0.08	15.84%	0.08
	10			8.72%	0.26	5.59%	1.07	5.73%	0.96
	20			6.67%	1.04	3.60%	3.46	3.67%	3.19
				4.30%	3.00	4.03%	3.00	3.78%	3.00
				3.19%	5.00	3.09%	5.00	3.07%	5.00
3	1	7.03	111.04	11.92%	0.02	7.65%	0.07	8.19%	0.07
	10			3.99%	0.23	2.06%	0.93	2.30%	0.90
	20			2.32%	0.50	0.83%	2.41	1.02%	2.33
				0.38%	3.00	0.68%	3.00	0.83%	3.00
				0.14%	5.00	0.33%	5.00	0.40%	5.00

- For all random examples above, CLP($\mathcal{H}_2$) had better initial accuracy than CLP($\mathcal{H}_1$) and LP($\mathcal{H}_3$). CLP($\mathcal{H}_2$) was more accurate than CLP($\mathcal{H}_1$) and CLP($\mathcal{H}_3$) at almost every iteration.
- CLP($\mathcal{H}_2$) was faster than CLP($\mathcal{H}_1$) and CLP($\mathcal{H}_1$) on most of the problems with fewer constraints such as $m = 10$. As shown in Figure 1, for the DNN problem with $n = 100$ and $m = 10$, CLP($\mathcal{H}_2$) attained an interval of 6.92% within 13s, while CLP($\mathcal{H}_1$) spent more than 50s. For problems with more constraints, such as $m = 50$, the differences in efficiency among CLP($\mathcal{H}_1$), CLP($\mathcal{H}_2$), and CLP($\mathcal{H}_3$) were small.

Table 4: Numerical results for DNN problems with n=50, m=50.

No.	Ite	DNN Time		CLP($\mathcal{H}_1$)		CLP($\mathcal{H}_2$)		CLP($\mathcal{H}_3$)	
		SDPT3	SeDuMi	Interval	Time	Interval	Time	Interval	Time
1	1	10.95	117.30	24.03%	0.40	19.37%	1.12	19.56%	0.70
	10			13.10%	4.18	9.90%	11.26	10.84%	9.56
	20			10.15%	9.13	8.22%	25.40	8.79%	23.25
				12.32%	5.00	12.83%	5.00	12.99%	5.00
				10.02%	10.00	10.27%	10.00	10.84%	10.00
2	1	9.90	111.22	24.76%	0.38	20.76%	0.84	20.76%	0.68
	10			13.27%	4.58	9.18%	11.63	9.85%	11.22
	20			10.43%	9.85	7.49%	27.74	7.88%	26.19
				12.93%	5.00	11.83%	5.00	11.93%	5.00
				10.43%	10.00	10.18%	10.00	10.18%	10.00
3	1	10.28	112.32	15.09%	0.45	13.05%	0.64	13.12%	0.76
	10			8.49%	4.32	6.76%	11.05	7.03%	9.60
	20			6.62%	10.26	5.46%	26.50	5.69%	24.59
				8.35%	5.00	8.71%	5.00	8.11%	5.00
				6.72%	10.00	7.15%	10.00	7.03%	10.00

Table 5: Numerical results for DNN problems with n=100, m=10.

No.	Ite	DNN Time		CLP($\mathcal{H}_1$)		CLP($\mathcal{H}_2$)		CLP($\mathcal{H}_3$)	
		SDPT3	SeDuMi	Interval	Time	Interval	Time	Interval	Time
1	1	61.27	11215.84	27.13%	2.05	17.80%	0.86	18.78%	0.85
	10			11.44%	20.73	6.92%	12.91	7.07%	12.84
	20			7.36%	43.21	4.03%	27.45	3.92%	27.48
				9.49%	30.00	3.84%	30.00	3.53%	30.00
				7.07%	50.00	2.39%	50.00	2.04%	50.00
2	1	61.30	10531.35	26.92%	1.99	18.14%	1.84	19.16%	1.48
	10			15.01%	20.51	7.77%	19.81	8.69%	17.58
	20			11.39%	42.58	5.85%	46.14	5.94%	38.14
				12.90%	30.00	6.67%	30.00	6.61%	30.00
				9.93%	50.00	5.49%	50.00	5.52%	50.00
3	1	51.41	10158.72	38.34%	1.84	25.37%	1.28	27.57%	1.28
	10			18.21%	20.69	9.87%	16.49	11.47%	16.56
	20			13.31%	41.98	6.25%	36.70	7.92%	37.41
				16.02%	30.00	7.51%	30.00	8.77%	30.00
				11.60%	50.00	4.80%	50.00	6.05%	50.00

- The proposed cutting plane methods obtained relatively accurate results, e.g., an interval of 5%, and were much faster than the SDP solver SeDuMi 1.3.

5 Concluding remarks

We developed techniques to construct a series of sparse polyhedral approximations of the semidefinite cone. We provided a way to approximate the semidefinite cone by using SD bases and proved that the set of diagonally dominant matrices can be expressed with sparse SD bases. We proposed a simple expansion of SD bases that keeps the sparsity of the matrices that compose it. We gave the conditions

Table 6: Numerical results for DNN problems with n=100, m=50.

No.	Ite	DNN Time		CLP($\mathcal{H}_1$)		CLP($\mathcal{H}_2$)		CLP($\mathcal{H}_3$)	
		SDPT3	SeDuMi	Interval	Time	Interval	Time	Interval	Time
1	1	76.75	10932.57	14.90%	2.79	11.35%	8.72	11.67%	8.20
	10			9.37%	28.48	6.87%	105.39	7.34%	88.14
	20			7.88%	61.10	5.57%	250.64	6.18%	216.87
				9.09%	30.00	9.18%	30.00	8.94%	30.00
				8.17%	50.00	8.26%	50.00	8.50%	50.00
2	1	85.14	12123.57	32.27%	2.72	26.24%	6.27	26.70%	6.14
	10			20.25%	28.34	15.69%	98.73	15.90%	82.43
	20			16.96%	60.34	13.07%	239.16	13.30%	200.59
				19.90%	30.00	19.86%	30.00	20.08%	30.00
				18.41%	50.00	18.77%	50.00	17.75%	50.00
3	1	80.36	11343.73	22.58%	2.69	17.46%	5.78	18.14%	9.36
	10			13.54%	29.48	10.27%	98.47	10.35%	100.49
	20			11.83%	61.27	8.69%	246.40	8.64%	241.60
				13.54%	30.00	13.35%	30.00	14.12%	30.00
				12.23%	50.00	11.68%	50.00	12.13%	50.00

for generating linearly independent matrices in expanded SD bases as well as for generating an expansion different from the existing one. We showed that the polyhedral approximation using our expanded SD bases contains the set of diagonally dominant matrices and is contained in the set of scaled diagonally dominant matrices. We also proved that the set of scaled diagonally dominant matrices can be expressed using an infinite number of expanded SD bases.

The expanded SD bases were then used to identify $A \in \mathcal{S}^n_+ + \mathcal{N}^n$. The numerical results showed that the proposed method with the expanded SD bases can be computationally efficient with good accuracy for identification.

Polyhedral approximations were applied to the cutting plane method for DNN problems. The results of the numerical experiments showed that the method with expanded SD bases is computationally efficient for some instances.

One future direction of study is to increase the number of vectors in the definition of the SD bases. The current SD bases are defined as a set of matrices $(p_i + p_j)(p_i + p_j)^T$. If we use three vectors, as in $(p_i + p_j + p_k)(p_i + p_j + p_k)^T$, we might obtain another inner approximation that remains relatively sparse when the dimension n is large.

Another future direction is to focus on the factor width k of a matrix. The cone of matrices with factor width at most $k = 2$ was introduced in order to give another expression of the set $\mathcal{SDD}_n$ of scaled diagonally dominant matrices. By considering a larger width $k > 2$, we may obtain a larger inner approximation of the semidefinite cone $\mathcal{S}^n_+$, while it would not be polyhedral, or even characterized by using SOCP constraints. Finding ways to solve approximation problems over such cones might be an interesting challenge.

Acknowledgments

This research was supported by the Japan Society for the Promotion of Science through a Grant-in-Aid for Challenging Exploratory Research (17K18946) and a Grant-in-Aid for Scientific Research ((B)19H02373) of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] A. A. Ahmadi, S. Dash, and G. Hall, *Optimization over structured subsets of positive semidefinite matrices via column generation*, Discrete Optimization, 24 (2017), pp. 129–151.

[2] A. A. Ahmadi and A. Majumdar, *DSOS and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization*, arXiv preprint arXiv:1706.02586, (2017).

[3] N. Arima, S. Kim, and M. Kojima, *A quadratically constrained quadratic optimization model for completely positive cone programming*, SIAM Journal on Optimization, 23 (2013), pp. 2320–2340.

[4] N. Arima, S. Kim, M. Kojima, and K.-C. Toh, *A robust Lagrangian-DNN method for a class of quadratic optimization problems*, Computational Optimization and Applications, 66 (2017), pp. 453–479.

[5] N. Arima, S. Kim, M. Kojima, and K.-C. Toh, *Lagrangian-conic relaxations, part i: A unified framework and its applications to quadratic optimization problems*, Pacific Journal of Optimization, 14 (2018), pp. 161–192.

[6] G. Barker and D. Carlson, *Cones of diagonally dominant matrices*, Pacific Journal of Mathematics, 57 (1975), pp. 15–32.

[7] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, vol. 9, Siam, 1994.

[8] L. Bishan, L. Lei, M. Harada, H. Niki, and M. J. Tsatsomeros, *An iterative criterion for H-matrices*, Linear Algebra and Its Applications, 271 (1998), pp. 179–190.

[9] G. Blekherman, P. A. Parrilo, and R. R. Thomas, *Semidefinite Optimization and Convex Algebraic Geometry*, SIAM, 2012.

[10] E. G. Boman, D. Chen, O. Parekh, and S. Toledo, *On factor width and symmetric H-matrices*, Linear Algebra and Its Applications, 405 (2005), pp. 239–248.

[11] S. Burer, *On the copositive representation of binary and continuous nonconvex quadratic programs*, Mathematical Programming, 120 (2009), pp. 479–495.

[12] S. Burer and R. D. C. Monteiro, *A projected gradient algorithm for solving the maxcut SDP relaxation*, Optimization Methods and Software, 15 (2001), pp. 175–200.

[13] G. Dantzig, R. Fulkerson, and S. Johnson, *Solution of a large-scale traveling-salesman problem*, Journal of the Operations Research Society of America, 2 (1954), pp. 393–410.

[14] G. B. Dantzig, D. R. Fulkerson, and S. M. Johnson, *On a linear-programming, combinatorial approach to the traveling-salesman problem*, Operations Research, 7 (1959), pp. 58–66.

[15] P. J. Dickinson and L. Gijben, *On the computational complexity of membership problems for the completely positive cone and its dual*, Computational Optimization and Applications, 57 (2014), pp. 403–415.

[16] M. Dür, *Copositive programming–a survey*, in Recent Advances in Optimization and Its Applications in Engineering, Springer, 2010, pp. 3–20.

[17] S. A. Geršgorin, *Über die abgrenzung der eigenwerte einer matrix*, Bulletin de l'Académie des Sciences de l'URSS. Classe des Sciences Mathématiques et na, 6 (1931), pp. 749–754.

[18] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, Journal of the ACM, 42 (1995), pp. 1115–1145.

[19] R. E. Gomory, *Outline of an algorithm for integer solutions to linear programs*, Bulletin of the American Mathematical Society, 64 (1958), pp. 275–278.

[20] L. Gurobi Optimization, *Gurobi optimizer reference manual.* Available at `http://www.gurobi.com`, 2018, [Accessed: 2018-11-20].

[21] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge university press, 1990.

[22] S. E. Karisch and F. Rendl, *Semidefinite programming and graph equipartition*, Topics in Semidefinite and Interior-Point Methods, 18 (1998), pp. 77–95.

[23] S. Kim and M. Kojima, *Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations*, Computational Optimization and Applications, 26 (2003), pp. 143–154.

[24] S. Kim, M. Kojima, and K.-C. Toh, *A lagrangian–DNN relaxation: a fast method for computing tight lower bounds for a class of quadratic optimization problems*, Mathematical Programming, 156 (2016), pp. 161–187.

[25] K. Kobayashi and Y. Takano, *A branch-and-cut algorithm for solving mixed-integer semidefinite optimization problems.* Available at `http://www.optimization-online.org/DB_FILE/2018/09/6808.pdf`, 2018, [Accessed: 2018-11-21].

[26] H. Konno, J.-y. Gotoh, T. Uno, and A. Yuki, *A cutting plane algorithm for semi-definite programming problems with applications to failure discriminant analysis*, Journal of Computational and Applied Mathematics, 146 (2002), pp. 141–154.

[27] K. Krishnan, *Linear programming approach to semidefinite programming problems*, PhD thesis, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, 2002.

[28] K. Krishnan and J. E. Mitchell, *A semidefinite programming based polyhedral cut and price approach for the maxcut problem*, Computational Optimization and Applications, 33 (2006), pp. 51–71.

[29] K. Krishnan and J. E. Mitchell, *A unifying framework for several cutting plane methods for semidefinite programming*, Optimization Methods and Software, 21 (2006), pp. 57–74.

[30] J. B. Lasserre, *An explicit exact SDP relaxation for nonlinear 0-1 programs*, in Integer Programming and Combinatorial Optimization, K. Aardal and B. Gerards, eds., Berlin, Heidelberg, 2001, Springer Berlin Heidelberg, pp. 293–303.

[31] M. Laurent and F. Vallentin, *Semidefinite optimization.* Lecture Notes, available at `http://page. mi. fu-berlin.de/fmario/sdp/laurentv. pdf`, 2012, [Accessed: 2018-11-20].

[32] J. Lofberg, *YALMIP: A toolbox for modeling and optimization in MATLAB*, in Computer Aided Control Systems Design, 2004 IEEE International Symposium on, IEEE, 2004, pp. 284–289.

[33] H. Minkowski, *Geometrie der Zahlen*, Leipzig : Teubner, 1896.

[34] K. G. Murty and S. N. Kabadi, *Some NP-complete problems in quadratic and nonlinear programming*, Mathematical Programming, 39 (1987), pp. 117–129.

[35] F. Permenter and P. Parrilo, *Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone*, Mathematical Programming, (2014), pp. 1–54.

[36] F. Permenter and P. A. Parrilo, *Basis selection for SOS programs via facial reduction and polyhedral approximations*, in 53rd IEEE Conference on Decision and Control, Dec 2014, pp. 6615–6620.

[37] J. Renegar, *A Mathematical View of Interior-point Methods in Convex Optimization*, vol. 3, SIAM, 2001.

[38] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, 1998.

[39] J. F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11 (1999), pp. 625–653.

[40] A. Tanaka and A. Yoshise, *LP-based tractable subcones of the semidefinite plus nonnegative cone*, Annals of Operations Research, 265 (2018), pp. 155–182.

[41] M. J. Todd, *Semidefinite optimization*, Acta Numerica, 10 (2001), pp. 515–560.

[42] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, *SDPT3—a MATLAB software package for semidefinite programming, version 1.3*, Optimization Methods and Software, 11 (1999), pp. 545–581.

[43] H. Waki and M. Muramatsu, *Facial reduction algorithms for conic optimization problems*, Journal of Optimization Theory and Applications, 158 (2013), pp. 188–215.

[44] H. Weyl, *Elementare theorie der konvexen polyeder*, Commentarii Mathematici Helvetici, 7 (1935), pp. 290–306.

[45] H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, vol. 27, Springer Science & Business Media, 2012.

[46] M. Yamashita, K. Fujisawa, and M. Kojima, *Implementation and evaluation of SDPA 6.0 (Semidefinite Programming Algorithm 6.0)*, Optimization Methods and Software, 18 (2003), pp. 491–505.