



Polyhedral approximations of the semidefinite cone and their application

Yuzhu Wang¹ · Akihiro Tanaka² · Akiko Yoshise³

Received: 28 November 2019 / Accepted: 9 December 2020 / Published online: 2 January 2021
© The Author(s) 2021

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 (Ann Oper Res 265:155–182, 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 all diagonally dominant matrices and is contained in the set of all scaled diagonally dominant matrices. We also prove that the set of all scaled diagonally dominant matrices can be expressed using an infinite number of expanded SD bases. We use our approximations as the initial approximation in cutting plane methods for solving a semidefinite relaxation of the maximum stable set problem. It is found that the proposed methods with expanded SD bases are significantly more efficient than methods using other existing approximations or solving semidefinite relaxation problems directly.

Keywords Semidefinite optimization problem · Conic optimization problem · Polyhedral approximation · Semidefinite basis · Expanded semidefinite basis

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.

✉ Yuzhu Wang
s1930138@s.tsukuba.ac.jp

Akihiro Tanaka
a-tanaka@criepi.denken.or.jp

Akiko Yoshise
yoshise@sk.tsukuba.ac.jp

- ¹ Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
- ² Central Research Institute of Electric Power Industry, Yokosuka, Kanagawa 240-0196, Japan
- ³ Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan

1 Introduction

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

$$\begin{aligned} &\min \langle C, X \rangle \\ &\text{s.t. } \langle A_j, X \rangle = b_j, j = 1, 2, \dots, m, \\ &\quad X \in \mathcal{S}_+^n, \end{aligned} \tag{1}$$

where $C \in \mathbb{S}^n, A_j \in \mathbb{S}^n, b_j \in \mathbb{R} (j = 1, 2, \dots, m),$ 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., [12, 19]) and the k-equipartition problem (e.g., [23, 46]). Some of these relaxations can even attain the optimum, as shown in [24, 31]. Interested readers may find details about SDPs and their relaxations in [32, 42, 46].

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 of study 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 [17] and the recent work of Arima et al. [3, 4, 25], 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., [16, 36]).

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 [47], SeDuMi [40], SDPT3 [43], and Mosek [35], 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 [37, 38, 44]. 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 et al. [13, 14] in 1954. It was used in 1958 by Gomory [20] to solve integer linear programming problems. As SDPs became popular, it came to be used on them as well; see, for instance, Krishnan and Mitchell [28–30], and Konno et al. [27]. Kobayashi and Takano [26] applied it to a class of mixed-integer SDPs. In [2], Ahmadi et al. 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}_{out} is polyhedral, the initial relaxation may give an LP; if \mathcal{K}_{out} is given by 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 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 [26] used the fact that the diagonal elements of semidefinite matrices are nonnegative. Konno et al. [27] 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., [2, 22] for details). The inclusive relation among them has been studied in, e.g., [7, 8]. Ahmadi et al. [1, 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 [38].

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 [41]. 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 determine whether a given matrix is in 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 by cutting plane methods for a semidefinite relaxation of the maximum stable set problem. It is found that the proposed methods with expanded SD bases are significantly more efficient than methods using other approximations or solving semidefinite relaxation problems directly.

The organization of this paper is as follows. Various approximations of \mathcal{S}_+^n are introduced in Sect. 2, including those based on the factor width by Boman et al. [10], diagonal dominance by Ahmadi et al. [2], and SD bases by Tanaka and Yoshise [41]. The main results of this paper, i.e., an expansion of SD bases and an analysis of its theoretical properties, are provided in Sect. 3. In Sect. 4, we introduce the cutting plane method using different approximations of \mathcal{S}_+^n for calculating upper bounds of the maximum stable set problem. We also describe the results of numerical experiments and evaluate the efficiency of the proposed method with expanded SD bases.

2 Some approximations of the semidefinite cone

2.1 Factor width approximation

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

Definition 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 $k \in \{1, 2, \dots, n\}$, we can also define

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

It is obvious that the factor width is only defined for semidefinite matrices, because for every matrix A in Definition 1, the decomposition $A = VV^T$ implies that $A \in \mathcal{S}_+^n$. Therefore, for every $k \in \{1, 2, \dots, n\}$, 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, 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 The set of diagonally dominant matrices \mathcal{DD}_n and the set of scaled diagonally dominant matrices \mathcal{SDD}_n are defined as follows:

$$\mathcal{DD}_n := \{A \in \mathbb{S}^n \mid A_{i,i} \geq \sum_{j \neq i} |A_{i,j}| \quad (i = 1, 2, \dots, n)\},$$

$$\mathcal{SDD}_n := \{A \in \mathbb{S}^n \mid DAD \in \mathcal{DD}_n \text{ for some positive diagonal matrix } D \}.$$

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 [18], we have the relation $\mathcal{DD}_n \subseteq \mathcal{SDD}_n \subseteq \mathcal{S}_+^n$. Ahmadi et al. [2] 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 $\mathcal{U}_{n,k} := \{uu^T \mid u \in \mathcal{U}_{n,k}\}$. Barker and Carlson [6] proved the following theorem.

Theorem 1 (Barker and Carlson [6]) $\mathcal{DD}_n = \text{cone}(\mathcal{U}_{n,2})$.

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 non-negative integers. A cone generated in this way by a finite number of elements is called *finitely generated*. Theorem 1 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 [34] and Weyl [45].

Theorem 2 (Minkowski–Weyl theorem, see Corollary 7.1a in [39]) *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 1, 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 [1, 10], which gives a relation between \mathcal{SDD}_n and the set of matrices with a factor width of at most 2.

Lemma 1 (See [10] and Theorem 8 in [1]) $\mathcal{FW}(2) = \mathcal{SDD}_n$

Note that Definition 1 implies that the set $\mathcal{FW}(k)$ is convex for any $k \in \{1, 2, \dots, n\}$, and we obtain the following corollary of Lemma 1:

Corollary 1 *The set \mathcal{SDD}_n is a convex cone.*

2.3 SD basis approximation

Tanaka and Yoshise defined semidefinite (SD) bases [41].

Definition 3 (Definitions 1 and 2 in [41]) Let $e_i \in \mathbb{R}^n$ denotes the vector with a 1 at the i th coordinate and 0 elsewhere, and let $I = (e_1, \dots, e_n) \in \mathbb{S}^n$ be the identity matrix. Then

$$\mathcal{B}_+ := \{(e_i + e_j)(e_i + e_j)^T \mid 1 \leq i \leq j \leq n\}$$

is called an SD basis of Type I, and

$$\begin{aligned} \mathcal{B}_- := & \{(e_i + e_i)(e_i + e_i)^T \mid 1 \leq i \leq n\} \cup \\ & \{(e_i - e_j)(e_i - e_j)^T \mid 1 \leq i < j \leq n\} \end{aligned}$$

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

$$B_{i,j}^+ = (e_i + e_j)(e_i + e_j)^T, \quad B_{i,j}^- := (e_i - e_j)(e_i - e_j)^T.$$

As shown in [41], \mathcal{B}_+ and \mathcal{B}_- are subsets of \mathcal{S}_+^n and bases of \mathbb{S}^n . Given a set $\mathcal{K} \subseteq \mathbb{S}^n$, we define the dual cone of \mathcal{K} as $(\mathcal{K})^* := \{A \in \mathbb{S}^n \mid \langle A, B \rangle \geq 0 \text{ for any } B \in \mathcal{K}\}$. The conical hull of $\mathcal{B}_+ \cup \mathcal{B}_-$ and its dual give an inner and an outer polyhedral approximation of \mathcal{S}_+^n , as follows.

Definition 4 Let $I = (e_1, \dots, e_n) \in \mathbb{S}^n$ be the identity matrix. The inner and outer approximations of \mathcal{S}_+^n by using SD bases are defined as

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

By Definition 3, we know that $\mathcal{B}_+, \mathcal{B}_- \subseteq \mathcal{S}_+^n$. Since \mathcal{S}_+^n is a convex cone, we have $\mathcal{S}_{\text{in}} \subseteq \text{cone}(\mathcal{S}_+^n) = \mathcal{S}_+^n$. By Lemma 1.7.3 in [32], we know that \mathcal{S}_+^n is self-dual; that is, $\mathcal{S}_+^n = (\mathcal{S}_+^n)^*$. Accordingly, we can conclude that $\mathcal{S}_{\text{in}} \subseteq \mathcal{S}_+^n \subseteq \mathcal{S}_{\text{out}}$.

Remark 1 In [41], \mathcal{B}_+ and \mathcal{B}_- are defined as $\mathcal{B}_+(P)$ and $\mathcal{B}_-(P)$ using an orthogonal matrix P instead of the identity matrix I . In fact, for any orthogonal matrix P ,

$$P\mathcal{B}_+P^T := \{PB_{i,j}^+P^T \mid B_{i,j}^+ \in \mathcal{B}_+\} \text{ and } P\mathcal{B}_-P^T := \{PB_{i,j}^-P^T \mid B_{i,j}^- \in \mathcal{B}_-\}$$

also give other bases and generalizations of \mathcal{B}_+ and \mathcal{B}_- . However, as we will see in section 4, we use the matrices in the bases as in optimization problems of the form

$$\min \langle C, X \rangle \text{ s.t. } \langle A, X \rangle = b, \langle Y, X \rangle \geq 0 \ (Y \in \mathcal{B}_+),$$

which is equivalent to

$$\min \langle PCP^T, \bar{X} \rangle \text{ s.t. } \langle PAP^T, \bar{X} \rangle = b, \langle Y, \bar{X} \rangle \geq 0 \ (Y \in P\mathcal{B}_+P^T). \tag{2}$$

Therefore, we consider that the generalizations $P\mathcal{B}_+P^T$ and $P\mathcal{B}_-P^T$ are not essential throughout this paper and omit those descriptions from subsequent sections to simplify the presentation.

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 mentioned in Remark 1, for any orthogonal matrix P , $P\mathcal{B}_+P^T$ and $P\mathcal{B}_-P^T$ give generalizations of the SD bases. However, it is hard to choose an appropriate orthogonal matrix P (except for the identity matrix I) to keep the sparsity of the matrices PCP^T and PAP^T in (2). 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 give a lemma that provides an expression of \mathcal{S}_+^n by using SD bases. The lemma is a direct corollary of the fact that any $X \in \mathcal{S}_+^n$ has nonnegative eigenvalues and a corresponding orthogonal basis of eigenvectors.

Lemma 2

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

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

Lemma 2 gives a way to approximate \mathcal{S}_+^n by changing the matrix $P = (p_1, \dots, p_n) \in \mathcal{O}^n$ when creating SD bases. However, a dense matrix $P \in \mathcal{O}^n$ may lead to a dense formulation of the approximation using SD basis, which is unattractive from the standpoint of computational efficiency.

Note that we can easily see that the set $\text{cone}(\mathcal{B}_+ \cup \mathcal{B}_-)$, the conical hull of the sparse SD bases \mathcal{B}_+ and \mathcal{B}_- , is equivalent to $\text{cone}(U_{n,2})$. Thus, we obtain the following proposition as a corollary of Theorem 1.

Proposition 1

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

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 using 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 5 Let $I = (e_1, \dots, e_n) \in \mathcal{S}^n$ be the identity matrix. Define the expansion of the SD basis with one parameter $\alpha \in \mathbb{R}$ as

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

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

Proposition 2 *Let $I = (e_1, \dots, e_n) \in \mathbb{S}^n$ be the identity matrix. For any $\alpha \in \mathbb{R} \setminus \{0, -1\}$, $\bar{B}(\alpha)$ is a set of $n(n + 1)/2$ independent matrices and thus a basis of \mathbb{S}^n .*

Proof Let $\alpha \in \mathbb{R} \setminus \{0, -1\}$. Accordingly, for $1 \leq i < j \leq n$, we have

$$\begin{aligned} \bar{B}_{i,j}(\alpha) &:= (e_i + \alpha e_j)(e_i + \alpha e_j)^T \\ &= e_i e_i^T + \alpha(e_i e_j^T + e_j e_i^T) + \alpha^2 e_j e_j^T \\ &= \alpha(e_i e_i^T + e_i e_j^T + e_j e_i^T + e_j e_j^T) + (1 - \alpha)e_i e_i^T + (\alpha^2 - \alpha)e_j e_j^T \quad (3) \\ &= \alpha B_{ij}^+ + \frac{1 - \alpha}{4} B_{ii}^+ + \frac{\alpha(\alpha - 1)}{4} B_{jj}^+, \end{aligned}$$

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

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

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

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

Then, by (3) and (4), we see that

$$\begin{aligned}
 O &= \sum_{i=1}^n \frac{\gamma_{i,i}(1+\alpha)^2}{4} B_{i,i}^+ + \sum_{1 \leq i < j \leq n} \gamma_{i,j} \left[\alpha B_{i,j}^+ + \frac{1-\alpha}{4} B_{i,i}^+ + \frac{\alpha(\alpha-1)}{4} B_{j,j}^+ \right] \\
 &= \sum_{i=1}^n \frac{(1+\alpha)^2}{4} \gamma_{i,i} B_{i,i}^+ + \sum_{1 \leq i < j \leq n} \alpha \gamma_{i,j} B_{i,j}^+ + \sum_{i=1}^{n-1} \frac{1-\alpha}{4} \left(\sum_{j=i+1}^n \gamma_{i,j} \right) B_{i,i}^+ \\
 &\quad + \sum_{j=2}^n \frac{\alpha(\alpha-1)}{4} \left(\sum_{i=1}^{j-1} \gamma_{i,j} \right) B_{j,j}^+ \\
 &= \left[\frac{\gamma_{1,1}(1+\alpha)^2}{4} + \frac{1-\alpha}{4} \left(\sum_{j=2}^n \gamma_{1,j} \right) \right] B_{1,1}^+ \tag{5} \\
 &\quad + \sum_{i=2}^{n-1} \left[\frac{(1+\alpha)^2}{4} \gamma_{i,i} + \frac{1-\alpha}{4} \left(\sum_{j=i+1}^n \gamma_{i,j} \right) + \frac{\alpha(\alpha-1)}{4} \left(\sum_{j=1}^{i-1} \gamma_{j,i} \right) \right] B_{i,i}^+ \\
 &\quad + \left[\frac{\gamma_{n,n}(1+\alpha)^2}{4} + \frac{\alpha(\alpha-1)}{4} \left(\sum_{j=1}^{n-1} \gamma_{j,n} \right) \right] B_{n,n}^+ \\
 &\quad + \sum_{1 \leq i < j \leq n} \alpha \gamma_{i,j} B_{i,j}^+.
 \end{aligned}$$

Since $\{B_{i,j}^+\} = \mathcal{B}_+$ is a set of linearly independent matrices, all the coefficients for $B_{i,j}$ in (5) should be 0. Thus, we have

$$0 = \frac{\gamma_{1,1}(1+\alpha)^2}{4} + \frac{1-\alpha}{4} \left(\sum_{j=2}^n \gamma_{1,j} \right), \tag{6}$$

$$0 = \frac{(1+\alpha)^2}{4} \gamma_{i,i} + \frac{1-\alpha}{4} \left(\sum_{j=i+1}^n \gamma_{i,j} \right) + \frac{\alpha(\alpha-1)}{4} \left(\sum_{j=1}^{i-1} \gamma_{j,i} \right) \quad (2 \leq i \leq n-1), \tag{7}$$

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

$$0 = \alpha \gamma_{i,j} \quad (1 \leq i < j \leq n). \tag{9}$$

Since $\alpha \neq 0$, by (9) we have

$$\gamma_{i,j} = 0 \quad (1 \leq i < j \leq n). \tag{10}$$

Since $\alpha \neq -1$, (6)-(10) imply that

$$\gamma_{i,i} = 0 \quad (i = 1, 2, \dots, n). \tag{11}$$

The above leads us to conclude that $\{\bar{B}_{i,j}(\alpha)\} = \bar{B}(\alpha)$ is a set of $n(n + 1)/2$ linearly independent matrices. □

If we let $\alpha = 1$, then it is straightforward that $\bar{B}(1) = \mathcal{B}_+$. 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 *Let $I = (e_1, \dots, e_n) \in \mathbb{S}^n$ be the identity matrix. Suppose that $\alpha_1 \in \mathbb{R} \setminus \{0, -1\}$ and $\alpha_2 \in \mathbb{R} \setminus \{0, \alpha_1\}$. Then, for every $1 \leq i < j \leq n$,*

$$(e_i + \alpha_2 e_j)(e_i + \alpha_2 e_j)^T \notin \text{cone}(\bar{B}(\alpha_1)).$$

Proof For $1 \leq i \leq j \leq n$, let us define

$$\bar{B}_{i,j}^1 := (e_i + \alpha_1 e_j)(e_i + \alpha_1 e_j)^T, \quad \bar{B}_{i,j}^2 := (e_i + \alpha_2 e_j)(e_i + \alpha_2 e_j)^T.$$

Note that if $i = j$, then

$$\bar{B}_{i,i}^1 := (1 + \alpha_1)^2 e_i e_i^T, \quad \bar{B}_{i,i}^2 := (1 + \alpha_2)^2 e_i e_i^T. \tag{12}$$

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

$$\begin{aligned} \bar{B}_{i,j}^2 &= e_i e_i^T + \alpha_2^2 e_j e_j^T + \alpha_2(e_i e_j^T + e_j e_i^T) \\ &= e_i e_i^T + \alpha_2^2 e_j e_j^T + \frac{\alpha_2}{\alpha_1} \alpha_1 (e_i e_j^T + e_j e_i^T) \quad (\text{because } \alpha_1 \neq 0) \\ &= e_i e_i^T + \alpha_2^2 e_j e_j^T - \frac{\alpha_2}{\alpha_1} e_i e_i^T - \frac{\alpha_2 \alpha_1^2}{\alpha_1} e_j e_j^T \\ &\quad + \frac{\alpha_2}{\alpha_1} [e_i e_i^T + \alpha_1 (e_i e_j^T + e_j e_i^T) + \alpha_1^2 e_j e_j^T] \\ &= \frac{\alpha_1 - \alpha_2}{\alpha_1} e_i e_i^T + \alpha_2(\alpha_2 - \alpha_1) e_j e_j^T + \frac{\alpha_2}{\alpha_1} \bar{B}_{i,j}^1 \\ &= \frac{\alpha_1 - \alpha_2}{\alpha_1(1 + \alpha_1)^2} (1 + \alpha_1)^2 e_i e_i^T + \frac{\alpha_2(\alpha_2 - \alpha_1)}{(1 + \alpha_1)^2} (1 + \alpha_1)^2 e_j e_j^T + \frac{\alpha_2}{\alpha_1} \bar{B}_{i,j}^1 \\ &\quad (\text{because } \alpha_1 \neq -1) \\ &= \frac{\alpha_1 - \alpha_2}{\alpha_1(1 + \alpha_1)^2} \bar{B}_{i,i}^1 + \frac{\alpha_2(\alpha_2 - \alpha_1)}{(1 + \alpha_1)^2} \bar{B}_{j,j}^1 + \frac{\alpha_2}{\alpha_1} \bar{B}_{i,j}^1 \quad (\text{by (12)}). \end{aligned} \tag{13}$$

Since $\alpha_1 \notin \{0, -1\}$, Proposition 2 ensures that $\bar{B}(\alpha_1)$ is linearly independent, and hence, the expression (13) for $\bar{B}_{i,j}^2$ is unique.

Suppose that $\bar{B}_{i,j}^2 \in \text{cone}(\bar{B}(\alpha_1))$. In this case, all the coefficients in (13) should be nonnegative, which implies that

$$\frac{\alpha_1 - \alpha_2}{\alpha_1(1 + \alpha_1)^2} \geq 0, \quad \frac{\alpha_2(\alpha_2 - \alpha_1)}{(1 + \alpha_1)^2} \geq 0, \quad \frac{\alpha_2}{\alpha_1} > 0. \tag{14}$$

From the last inequality in (14), we have either

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

For case (i), from the first and second inequalities of (14), 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 $\bar{B}_{ij}^2 \notin \text{cone}(\bar{B}(\alpha_1))$. \square

3.3 Expression of SDD_n with expanded SD bases

As we have seen in Corollary 1, the set $SDD_n = \mathcal{FW}(2)$ is a convex cone. This fact ensures that as a corollary of Theorem 1, the conical hull of the union of the extended SD bases $\bar{B}(\alpha)$ on $\alpha \in \mathbb{R}$ coincides with $\mathcal{FW}(2)$ and hence, the set of scaled diagonally dominant matrices SDD_n :

Corollary 2

$$\text{cone}\left(\bigcup_{\alpha \in \mathbb{R}} \bar{B}(\alpha)\right) = SDD_n.$$

3.4 Notes on the parameter α

Here, we discuss the choice for the parameter α to increase the ‘‘volume’’ of the polyhedral approximation $\text{cone}(\bar{B}(\alpha))$ of the semidefinite cone S_+^n . For any $\alpha \in \mathbb{R}$ and $1 \leq i < j \leq n$, by Definition 5, we can calculate the Frobenius norm of $\bar{B}_{ij}(\alpha)$:

$$\begin{aligned} \|\bar{B}_{ij}(\alpha)\| &= \|(e_i + \alpha e_j)(e_i + \alpha e_j)^T\| \\ &= \sqrt{\text{Trace}((e_i + \alpha e_j)(e_i + \alpha e_j)^T(e_i + \alpha e_j)(e_i + \alpha e_j)^T)} \\ &= \|e_i + \alpha e_j\|^2 \\ &= 1 + \alpha^2. \end{aligned} \tag{15}$$

According to Proposition 3, by changing α , one can obtain different polyhedral approximations. However, we can see that

$$\begin{aligned} \lim_{|\alpha| \rightarrow \infty} \frac{\bar{B}_{ij}(\alpha)}{\|\bar{B}_{ij}(\alpha)\|} &= \lim_{|\alpha| \rightarrow \infty} \frac{1}{1 + \alpha^2} (e_i + \alpha e_j)(e_i + \alpha e_j)^T \text{ (by(15))}, \\ &= \lim_{|\alpha| \rightarrow \infty} \left[\frac{1}{1 + \alpha^2} e_i e_i^T + \frac{\alpha}{1 + \alpha^2} (e_i e_j^T + e_j e_i^T) + \frac{\alpha^2}{1 + \alpha^2} e_j e_j^T \right] \\ &= e_j e_j^T = \frac{1}{4} B_{jj}^+, \end{aligned}$$

and by Definitions 3 and 5, we have

$$\bar{B}_{ij}(0) = \frac{1}{4}B_{ii}^+, \bar{B}_{ij}(1) = B_{ij}^+, \bar{B}_{ij}(-1) = B_{ij}^-.$$

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

To give an illustrative explanation of the above discussion, here we consider the specific case

$$\mathcal{S}_+^2 = \left\{ \begin{pmatrix} a & c \\ c & b \end{pmatrix} \mid a, b, c \in \mathbb{R}, a, b \geq 0, ab - c^2 \geq 0 \right\}$$

and draw some figures in \mathbb{R}^3 with coordinate a, b and c . Figure 1a shows the set of \mathcal{S}_+^2 in \mathbb{R}^3 . The red arrow in Fig. 1b shows the extreme rays $\{\gamma\bar{B}_{ij}(\alpha) \mid \gamma \geq 0\}$ with $|\alpha| \rightarrow \infty$ and $\alpha \in \{0, 1, -1\}$. The conical hull of these extreme rays is cone($\mathcal{B}_+ \cup \mathcal{B}_-$) and its cross section with $\{X \in \mathcal{S}^2 \mid \langle X, I \rangle = 1\}$ is illustrated as the blue area. To avoid generating a new matrix $\bar{B}_{ij}(\alpha)$ that is close to the existing matrices, we should choose an α such that the angle between $\bar{B}_{ij}(\alpha)$ and existing matrices are equal, as illustrated in Fig. 1c.

We expand this idea to the case of generating a matrix $\bar{B}_{ij}(\alpha) \in \mathcal{S}^n$. 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 $1 \leq i < j \leq n$, as follows:

$$\theta_1(\alpha) := \arccos \frac{\langle \bar{B}_{ij}(\alpha), B_{ii}^+ \rangle}{\|\bar{B}_{ij}(\alpha)\| \|B_{ii}^+\|}, \theta_2(\alpha) := \arccos \frac{\langle \bar{B}_{ij}(\alpha), B_{jj}^+ \rangle}{\|\bar{B}_{ij}(\alpha)\| \|B_{jj}^+\|},$$

$$\theta_3(\alpha) := \arccos \frac{\langle \bar{B}_{ij}(\alpha), B_{ij}^+ \rangle}{\|\bar{B}_{ij}(\alpha)\| \|B_{ij}^+\|}, \theta_4(\alpha) := \arccos \frac{\langle \bar{B}_{ij}(\alpha), B_{ij}^- \rangle}{\|\bar{B}_{ij}(\alpha)\| \|B_{ij}^-\|}.$$

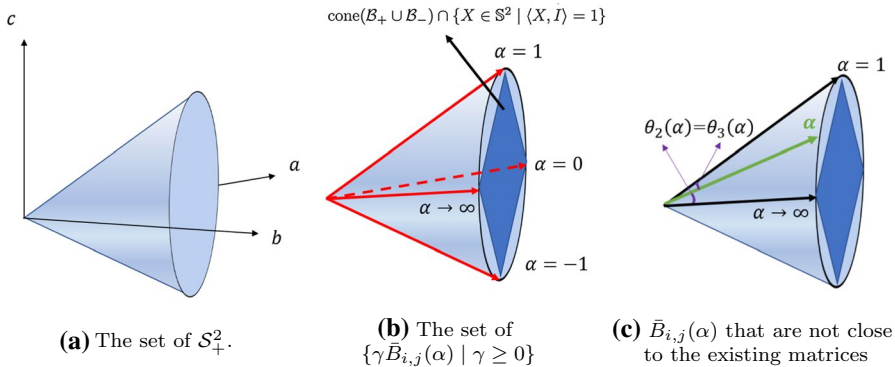


Fig. 1 Choice of α to generate $\bar{B}_{ij}(\alpha) \in \mathcal{S}^2$ in \mathbb{R}^3

Thus, we have

$$\begin{aligned} \cos\theta_1(\alpha) &= \frac{\langle \bar{B}_{i,j}(\alpha), B_{i,i}^+ \rangle}{\|\bar{B}_{i,j}(\alpha)\| \|B_{i,i}^+\|} \\ &= \frac{\langle (e_i + \alpha e_j)(e_i + \alpha e_j)^T, (e_i + e_i)(e_i + e_i)^T \rangle}{(1 + \alpha^2)\|(e_i + e_i)(e_i + e_i)^T\|} \quad (\text{by (15)}) \\ &= \frac{4\|e_i\|^4}{(1 + \alpha^2)4\|e_i\|^2} \quad (\text{because } e_i^T e_j = 0) \\ &= \frac{1}{1 + \alpha^2} \quad (\text{because } \|e_i\| = 1). \end{aligned}$$

Similarly, we have

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

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 $\bar{B}_{i,j}(\alpha)$ will be in the middle of $B_{i,i}^+$ and $B_{i,j}^+$ 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

$$\alpha = \pm 1 \pm \sqrt{2}.$$

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

4 Cutting plane methods for the maximum stable set problem

Conic optimization problems, including SDPs and copositive programs, have been shown to provide tight bounds for NP-hard combinatorial and nonconvex optimization problems. Here, we consider applying approximations of \mathcal{S}_+^n to one of those NP-hard problems, the maximum stable set problem. A stable set of a graph $G(V, E)$ is a set of vertices in V , such that there is no edge connecting any pair of vertices in the set. The maximum stable set problem aims to find the stability number, i.e. the number of vertices of the largest stable set of G , namely $\alpha(G)$.

De Klerk and Pasechnik [15] proposed a copositive programming formulation to obtain the exact stability number of a graph G with n vertices:

$$\begin{aligned} \alpha(G) &= \max \langle ee^T, X \rangle \\ \text{s.t. } &\langle A + I, X \rangle = 1, \\ &X \in \mathcal{C}_n^*, \end{aligned} \tag{16}$$

where e is the all-ones vector, A is the adjacency matrix of graph G , and C_n^* is the dual cone of the copositive cone $C_n := \{X \in \mathbb{S}^n \mid d^T X d \geq 0 \forall d \in \mathbb{R}^n, d \geq 0\}$.

Although problem (16) is a conic optimization problem, it is still difficult since determining whether $X \in C_n^*$ or not is NP-hard [16]. A natural approach is to relax this problem to a more tractable optimization problem. From the definition of each cone, we can see the validity of the following inclusions:

$$C_n^* \subseteq S_+^n \cap \mathcal{N}^n \subseteq S_+^n \subseteq S_+^n + \mathcal{N}^n \subseteq C_n.$$

By replacing C_n^* with $S_+^n \cap \mathcal{N}^n$, one can obtain an SDP relaxation of (16):

$$\begin{aligned} \max \quad & \langle ee^T, X \rangle \\ \text{s.t.} \quad & \langle A + I, X \rangle = 1, \\ & X \in S_+^n \cap \mathcal{N}^n. \end{aligned} \tag{17}$$

Solving this SDP is not as easy as it seems to be; in fact, we could not obtain a useful result of (17) after 6 hours of calculation using the state-of-the-art SDP solver Mosek for a random generalized problem when $n = 300$. Combining the expanded SD bases with the cutting plane method, we apply the approximations of S_+^n to (17) and solve it by calculating a series of more tractable problems.

Let \mathcal{P}^n satisfy $S_+^n \subseteq \mathcal{P}^n \subseteq \mathbb{S}^n$ and replace $X \in S_+^n$ by $X \in \mathcal{P}^n$ in (17). Then, we obtain a relaxation of (17):

$$\begin{aligned} \max \quad & \langle ee^T, X \rangle \\ \text{s.t.} \quad & \langle A + I, X \rangle = 1, \\ & X \in \mathcal{P}^n \cap \mathcal{N}^n. \end{aligned} \tag{18}$$

Usually, the relaxed problem (18) is expected to be easier to solve and to give us a better upper bound of problem (17) from its optimal solution X^* . To get a better upper bound, we select some eigenvectors with negative eigenvalues of an optimal solution X^* of problem (18), say d_1, \dots, d_k , by adding cutting planes

$$\langle d_i d_i^T, X \rangle \geq 0 \quad (i = 1, \dots, k)$$

to (18), and obtain a new optimization problem

$$\begin{aligned} \max \quad & \langle ee^T, X \rangle \\ \text{s.t.} \quad & \langle A + I, X \rangle = 1, \\ & \langle d_i d_i^T, X \rangle \geq 0 \quad (i = 1, \dots, k) \\ & X \in \mathcal{P}^n \cap \mathcal{N}^n. \end{aligned} \tag{19}$$

Notice that the optimal solution X^* of problem (18) is cut from the feasible region of problem (19) since $\langle d_i d_i^T, X^* \rangle < 0 \quad (i = 1, \dots, k)$. On the other hand, since $S_+^n = \{X \in \mathbb{S}^n \mid \forall d \in \mathbb{R}^n, \langle dd^T, X \rangle \geq 0\} \subseteq \mathcal{P}^n$, every feasible solution of (17) is feasible for (19), and hence problem (19) is a relaxation of problem (17). These facts ensure that problem (19) is a tighter relaxation of problem (17) than problem (18). By repeating this procedure, we are able to obtain a series of nonincreasing upper

bounds of (17). 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.

As for the selection of the initial relaxation \mathcal{P}^n , we are ready to use the approximations of \mathcal{S}_+^n based on the expanded SD bases. Let $\mathcal{H} := \{\pm 1, \pm 1 \pm \sqrt{2}\}$ be the set of parameters calculated in Sect. 3.4, and let SDB_n denote the conical hull of expanded SD bases using \mathcal{H} :

$$SDB_n := \text{cone}\left(\bigcup_{\alpha \in \mathcal{H}} \bar{B}(\alpha)\right).$$

Then, as has been described in the previous sections, we have

$$\mathcal{S}_+^n \subseteq SDD_n^* \subseteq SDB_n^* \subseteq DD_n^*. \tag{20}$$

If SDB_n^* or DD_n^* is selected to be \mathcal{P}_n , the corresponding relaxed problem in the cutting plane procedure becomes an LP, which allows us to use powerful state-of-the-art LP solvers, such as Gurobi [21]. Ahmadi et. al. [2] showed that when SDD_n^* is selected, the relaxations turn out to be SOCPs. Although SDD_n^* provides a tighter relaxation than either DD_n or SDB_n , the latter two relaxations are expected to have a lower computational cost. In addition, in [2], Ahmadi et al. also proposed an SOCP-based cutting plane approach, named SDSOS, which adds SOCP cuts at every iteration. We conducted experiments to compare the efficiencies of those cutting plane methods using different approximations and SDSOS. The specifications of the experimental methods are summarized in Table 1.

We tested these methods on the Erdős–Rényi graphs $ER(n, p)$, randomly generated by Ahmadi et al. in [2], where n is the number of vertices and every pair of vertices has an edge with probability p . All experiments were performed with MATLAB 2018b on a Windows 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 [21] and the SOCPs and SDPs are solved using Mosek Optimizer 9.0 [35].

Figure 2 shows the result for an instance with $n = 250$ and $p = 0.8$. The x-axis is the number of iterations, and the y-axis is the gap between the upper bounds of each method and the SDP bound obtained by (17); the gap is computed by

Table 1 Specifications of the experimental methods

| Method | \mathcal{P}^n | Number of cuts added at each iteration | | Solver |
|--------|-----------------|----------------------------------------|----------|--------|
| | | LP cut | SOCP cut | |
| CPDD | DD_n^* | 2 | 0 | Gurobi |
| CPSDB | SDB_n^* | 2 | 0 | Gurobi |
| CPSDD | SDD_n^* | 2 | 0 | Mosek |
| SDSOS | SDD_n^* | 2 | 1 | Mosek |

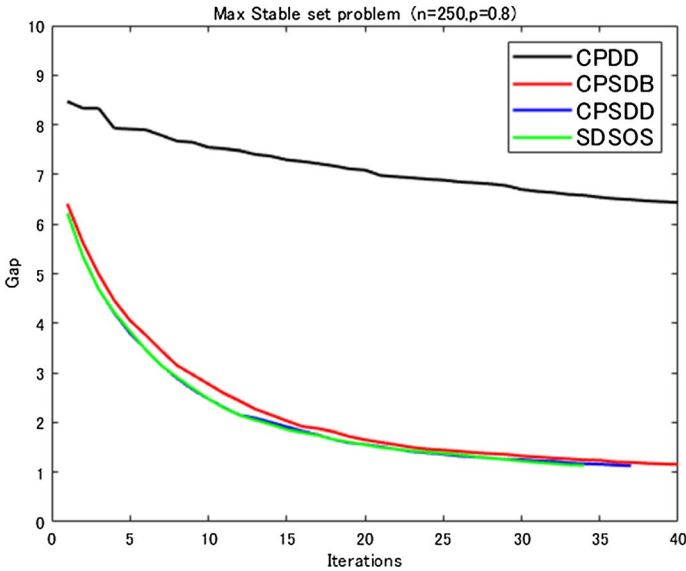


Fig. 2 Relation between the number of iterations and the gap

$\left| \frac{f^* - f_k}{f^*} \right| \times 100\%$ for the obtained upper bound f_k at k 's iteration and the SDP bound f^* obtained by solving (17) directly.

As can be seen in this figure, the accuracy of CPDD is the worst among the four methods at each iteration. CPSDB achieves almost the same upper bounds as CPSDD and SDSOS, which shows that the proposed polyhedral approximation SDB_n is promising for obtaining a solution close to the non-polyhedral approximation SDD_n of S_+^n . Although SDSOS adds an extra SOCP cut at every iteration and takes longer to solve, the accuracy of SDSOS does not seem to be affected and is not so different from the accuracy of CPSDD at each iteration.

Figure 3 shows the relation between the computation time and the gap of each method for the same instance. Although its accuracy is not necessarily the best at every iteration, it seems that CPSDB is the most efficient method. CPSDB attains an upper bound whose gap is 2 within 30 s, while CPSDD and SDSOS attain upper bounds whose gap is 4 after the same amount of time. The difference might come from that the subproblems of CPSDB are sparse LPs at earlier iterations and the computations are relatively cheaper than those of CPSDD and SDSOS whose subproblems are SOCPs.

Tables 2 and 3 give the bounds of iterative methods and the SDP bound for all the instances. In Table 2, the CPSDD₀/SDSOS₀ column shows the first upper bound obtained by CPSDD and SDSOS, i.e., the upper bound obtained by solving the same SOCP before adding any cutting plane. The (5 min) and (10 min) columns of CPSDD (SDSOS) show the upper bounds obtained after 5 min and after 10 min of the CPSDD (SDSOS) computation, respectively. The SDP column shows the SDP bound obtained by solving (17).

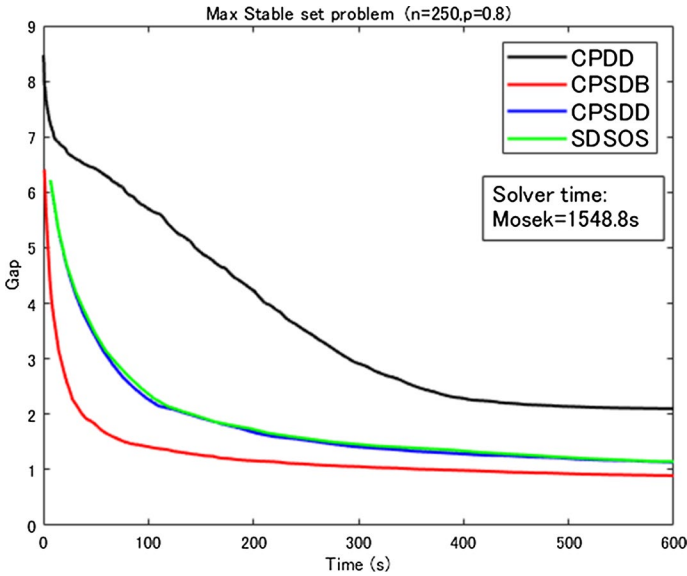


Fig. 3 Relation between the computational time (s) and the gap

Table 2 Upper bounds obtained by SDP and SOCP methods on $ER(n, p)$ graphs

| n | p | CPSDD ₀ /SDSOS ₀ | | CPSDD | | SDSOS | | SDP | |
|-----|-----|----------------------------------------|----------|---------|----------|---------|----------|---------|------------|
| | | Value | Time (s) | (5 min) | (10 min) | (5 min) | (10 min) | Value | Time (s) |
| 150 | 0.3 | 105.70 | 0.95 | 38.91 | 37.02 | 40.97 | 37.38 | 20.44 | 105.46 |
| 150 | 0.8 | 31.78 | 1.00 | 10.07 | 9.66 | 9.70 | 9.31 | 6.00 | 110.63 |
| 200 | 0.3 | 140.47 | 3.14 | 70.48 | 55.52 | 75.46 | 61.31 | 23.73 | 549.63 |
| 200 | 0.8 | 40.92 | 3.14 | 12.10 | 11.29 | 12.17 | 11.38 | 6.45 | 497.55 |
| 250 | 0.3 | 176.25 | 6.60 | 115.41 | 93.81 | 119.67 | 99.99 | 26.78 | 1562.52 |
| 250 | 0.8 | 51.87 | 6.79 | 17.36 | 15.30 | 17.43 | 15.39 | 7.18 | 1553.63 |
| 300 | 0.3 | 210.32 | 13.05 | 160.42 | 138.60 | 162.77 | 143.12 | (29.13) | (32300.60) |
| 300 | 0.8 | 60.97 | 13.31 | 21.71 | 17.77 | 22.66 | 18.50 | (7.65) | (20586.02) |

Similarly, in Table 3, the CPDD₀ and CPSDB₀ columns show the first upper bounds obtained by CPDD and CPSDB, respectively, before adding any cutting plane. The (5 min) and (10 min) columns of CPDD (CPSDB) show the upper bounds obtained after 5 min and after 10 min of the CPDD (CPSDB) computation, respectively.

Note that we failed to solve SDPs (17) for instances having $n = 300$ nodes within our time limit 20000s. In Table 2, the Value and Time (s) columns of SDP with $n = 300$ show the results obtained in [2] for these two instances, as a reference.

As can be seen in Tables 2 and 3, for all instances, the values of CPSDD₀/SDSOS₀ are better than the values of CPSDB₀ and CPDD₀. These results correspond to the

Table 3 Upper bounds obtained by LP methods on the same $ER(n, p)$ graphs

| n | p | CPDD ₀ | | CPDD | | CPSDB ₀ | | CPSDB | |
|-----|-----|-------------------|----------|---------|----------|--------------------|----------|---------|----------|
| | | Value | Time (s) | (5 min) | (10 min) | Value | Time (s) | (5 min) | (10 min) |
| 150 | 0.3 | 117 | 0.06 | 76.76 | 67.51 | 107.29 | 0.24 | 36.80 | 35.12 |
| 150 | 0.8 | 46 | 0.05 | 13.70 | 12.71 | 32.76 | 0.28 | 9.51 | 9.06 |
| 200 | 0.3 | 157 | 0.1 | 113.28 | 104.07 | 142.25 | 0.52 | 55.07 | 48.18 |
| 200 | 0.8 | 54 | 0.11 | 17.39 | 16.07 | 42.14 | 0.57 | 11.58 | 11.00 |
| 250 | 0.3 | 194 | 0.17 | 154.75 | 146.20 | 178.30 | 0.84 | 91.88 | 73.24 |
| 250 | 0.8 | 68 | 0.17 | 28.02 | 22.26 | 53.22 | 1.00 | 14.76 | 13.57 |
| 300 | 0.3 | 230 | 0.26 | 183.89 | 174.02 | 212.97 | 1.29 | 133.83 | 110.95 |
| 300 | 0.8 | 78 | 0.24 | 47.87 | 32.28 | 62.47 | 1.36 | 18.11 | 16.05 |

inclusion relationship of initial approximations (20). We can also see that the values of CPSDB₀ are almost the same as those of CPSDD₀/SDSOS₀ for all instances, while the values of CPDD₀ are much worse than others. For all instances, CPSDB seems to be significantly more efficient than all other methods. For example, for instance with $n = 250$ and $p = 0.3$, after 10 min of calculation, CPSDB obtained an upper bound of 73.24, while CPSDD and SDSOS got upper bounds greater than 90 and CPDD got a bound of more than 146.

At present, solving a large SDP, e.g., one with more than $n = 300$ nodes requires a significant amount of computational time. The cutting plane method CPSDB with our polyhedral approximation \mathcal{SDB}_n is a promising way of obtaining efficient upper bounds of such large SDPs in a moderate time.

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 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 polyhedral approximations were applied to the cutting plane method for solving a semidefinite relaxation of the maximum stable set problem. The results of the numerical experiments showed that the method with our expanded SD bases is more efficient than other methods (see Fig. 3); improving the efficiency of our method still remains an important study issue.

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 $(e_i + e_j)(e_i + e_j)^T$. If we use three vectors, as in $(e_i + e_j + e_k)(e_i + e_j + e_k)^T$, 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 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 S_+^n , although it would not be polyhedral, or even characterized by using SOCP constraints. Finding efficient ways to solve approximation problems over such cones might be an interesting challenge.

Also, our expanded SD bases can be applied to some other difficult problems. Mixed integer nonlinear programming has recently become popular in many practical applications. In [33], Lubin et al. proposed a cutting plane framework for mixed integer convex optimization problems. In [26], Kobayashi and Takano proposed a branch and bound cutting plane method for mixed integer SDPs. It would be interesting to see whether the approximations of S_+^n proposed in this paper could be used to improve the efficiency of those methods.

Acknowledgements The authors would like to sincerely thank the anonymous reviewers for their thoughtful and valuable comments which have significantly improved the paper. Among others, one of the reviewers pointed out Remark 1 which helped the authors to simplify the presentation of the paper. 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) from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Ahmadi, A.A., Majumdar, A.: DSOS and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization. *SIAM J. Appl. Algebra Geom.* **3**, 193–230 (2019)
2. Ahmadi, A.A., Dash, S., Hall, G.: Optimization over structured subsets of positive semidefinite matrices via column generation. *Discrete Optim.* **24**, 129–151 (2017)
3. Arima, N., Kim, S., Kojima, M.: A quadratically constrained quadratic optimization model for completely positive cone programming. *SIAM J. Optim.* **23**, 2320–2340 (2013)
4. Arima, N., Kim, S., Kojima, M., Toh, K.-C.: A robust Lagrangian-DNN method for a class of quadratic optimization problems. *Comput. Optim. Appl.* **66**, 453–479 (2017)
5. Arima, N., Kim, S., Kojima, M., Toh, K.-C.: Lagrangian-conic relaxations, part i: a unified framework and its applications to quadratic optimization problems. *Pac. J. Optim.* **14**, 161–192 (2018)
6. Barker, G., Carlson, D.: Cones of diagonally dominant matrices. *Pac. J. Math.* **57**, 15–32 (1975)

7. Berman, A., Plemmons, R.J.: *Nonnegative Matrices in the Mathematical Sciences*, vol. 9. SIAM, Philadelphia (1994)
8. Bishan, L., Lei, L., Harada, M., Niki, H., Tsatsomeros, M.J.: An iterative criterion for H-matrices. *Linear Algebra Appl.* **271**, 179–190 (1998)
9. Blekherman, G., Parrilo, P.A., Thomas, R.R.: *Semidefinite Optimization and Convex Algebraic Geometry*. SIAM, Philadelphia (2012)
10. Boman, E.G., Chen, D., Parekh, O., Toledo, S.: On factor width and symmetric H-matrices. *Linear Algebra Appl.* **405**, 239–248 (2005)
11. Burer, S.: On the copositive representation of binary and continuous nonconvex quadratic programs. *Math. Program.* **120**, 479–495 (2009)
12. Burer, S., Monteiro, R.D.C.: A projected gradient algorithm for solving the maxcut SDP relaxation. *Optim. Methods Softw.* **15**, 175–200 (2001)
13. Dantzig, G., Fulkerson, R., Johnson, S.: Solution of a large-scale traveling-salesman problem. *J. Oper. Res. Soc. Am.* **2**, 393–410 (1954)
14. Dantzig, G.B., Fulkerson, D.R., Johnson, S.M.: On a linear-programming, combinatorial approach to the traveling-salesman problem. *Oper. Res.* **7**, 58–66 (1959)
15. De Klerk, E., Pasechnik, D.V.: Approximation of the stability number of a graph via copositive programming. *SIAM J. Optim.* **12**, 875–892 (2002)
16. Dickinson, P.J., Gijben, L.: On the computational complexity of membership problems for the completely positive cone and its dual. *Comput. Optim. Appl.* **57**, 403–415 (2014)
17. Dür, M.: Copositive programming—a survey. In: *Recent Advances in Optimization and Its Applications in Engineering*. Springer, Berlin, pp. 3–20 (2010)
18. Geršgorin, S.A.: Über die abgrenzung der eigenwerte einer matrix. *Bull. l'Acad. Sci. l'URSS Classe Sci. Math.* **6**, 749–754 (1931)
19. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM* **42**, 1115–1145 (1995)
20. Gomory, R.E.: Outline of an algorithm for integer solutions to linear programs. *Bull. Am. Math. Soc.* **64**, 275–278 (1958)
21. Gurobi Optimization, L.: Gurobi optimizer reference manual. <http://www.gurobi.com> (2018). Accessed 20 Nov 2018
22. Horn, R.A., Johnson, C.R.: *Matrix Analysis*. Cambridge University Press, Cambridge (1990)
23. Karisch, S.E., Rendl, F.: Semidefinite programming and graph equipartition. In: *Topics in Semidefinite and Interior-point Method*. AMS, pp. 77–95 (1998)
24. Kim, S., Kojima, M.: Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations. *Comput. Optim. Appl.* **26**, 143–154 (2003)
25. Kim, S., Kojima, M., Toh, K.-C.: A lagrangian-DNN relaxation: a fast method for computing tight lower bounds for a class of quadratic optimization problems. *Math. Program.* **156**, 161–187 (2016)
26. Kobayashi, K., Takano, Y.: A branch-and-cut algorithm for solving mixed-integer semidefinite optimization problems. *Comput. Optim. Appl.* **75**, 493–513 (2020)
27. Konno, H., Gotoh, J.-Y., Uno, T., Yuki, A.: A cutting plane algorithm for semi-definite programming problems with applications to failure discriminant analysis. *J. Comput. Appl. Math.* **146**, 141–154 (2002)
28. Krishnan, K.: *Linear programming approach to semidefinite programming problems*. PhD thesis, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180 (2002)
29. Krishnan, K., Mitchell, J.E.: A semidefinite programming based polyhedral cut and price approach for the maxcut problem. *Comput. Optim. Appl.* **33**, 51–71 (2006)
30. Krishnan, K., Mitchell, J.E.: A unifying framework for several cutting plane methods for semidefinite programming. *Optim. Methods Softw.* **21**, 57–74 (2006)
31. Lasserre, J.B.: An explicit exact SDP relaxation for nonlinear 0–1 programs. In: Aardal, K., Gerards, B. (eds.) *Integer Programming and Combinatorial Optimization*. Springer, Berlin, pp. 293–303 (2001)
32. Laurent, M., Vallentin, F.: *Semidefinite optimization*. Lecture Notes. <http://page.mi.fu-berlin.de/fmario/sdp/laurentv.pdf> (2012). Accessed 20 Nov 2018
33. Lubin, M., Yamangil, E., Bent, R., Vielma, J.P.: Polyhedral approximation in mixed-integer convex optimization. *Math. Program.* **172**(1–2), 139–168 (2018)
34. Minkowski, H.: *Geometrie der Zahlen*. Teubner, Leipzig (1896)
35. MOSEK ApS: The MOSEK optimization toolbox for MATLAB manual. Version 8.1. Available at <http://docs.mosek.com/8.1/toolbox/index.html> (2017). Accessed 1 Oct 2019

36. Murty, K.G., Kabadi, S.N.: Some NP-complete problems in quadratic and nonlinear programming. *Math. Program.* **39**, 117–129 (1987)
37. Permenter, F., Parrilo, P. A.: Basis selection for SOS programs via facial reduction and polyhedral approximations. In: 53rd IEEE Conference on Decision and Control, CDC 2014, pp. 6615–6620 (2014)
38. Permenter, F., Parrilo, P.: Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone. *Math. Program.* 1–54 (2014)
39. Schrijver, A.: *Theory of Linear and Integer Programming*. Wiley, London (1998)
40. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.* **11**, 625–653 (1999)
41. Tanaka, A., Yoshise, A.: LP-based tractable subcones of the semidefinite plus nonnegative cone. *Ann. Oper. Res.* **265**, 155–182 (2018)
42. Todd, M.J.: Semidefinite optimization. *Acta Numer.* **10**, 515–560 (2001)
43. Toh, K.-C., Todd, M.J., Tütüncü, R.H.: SDPT3—a MATLAB software package for semidefinite programming, version 1.3. *Optim. Methods Softw.* **11**, 545–581 (1999)
44. Waki, H., Muramatsu, M.: Facial reduction algorithms for conic optimization problems. *J. Optim. Theory Appl.* **158**, 188–215 (2013)
45. Weyl, H.: Elementare theorie der konvexen polyeder. *Comment. Math. Helvet.* **7**, 290–306 (1935)
46. Wolkowicz, H., Saigal, R., Vandenberghe, L.: *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, vol. 27. Springer, Berlin (2012)
47. Yamashita, M., Fujisawa, K., Kojima, M.: Implementation and evaluation of SDPA 6.0 (Semidefinite Programming Algorithm 6.0). *Optim. Methods Softw.* **18**, 491–505 (2003)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.