An LP-based Algorithm to Test Copositivity

Akihiro Tanaka* and Akiko Yoshise†

April 2014
Revised October 2014

Abstract

A symmetric matrix is called copositive if it generates a quadratic form taking no negative values over the nonnegative orthant, and the linear optimization problem over the set of copositive matrices is called the copositive programming problem. Recently, many studies have been done on the copositive programming problem (see, for example, [14, 5]). Among others, several branch and bound type algorithms have been provided to test copositivity in the context of the fact that deciding whether a given matrix is copositive is co-NP-complete [23, 13]. In this paper, we propose a new branch and bound type algorithm for this testing problem based on Sponsel, Bundfuss and Dür's algorithm [27]. Two features of our algorithm are: (1) we introduce new classes of matrices $\mathcal{G}_n^s$ and $\widehat{\mathcal{G}_n^s}$ which are relatively large subsets of the set of copositive matrices and work well to check copositivity of a given $n \times n$ symmetric matrix, and (2) for incorporating the sets $\mathcal{G}_n^s$ or $\widehat{\mathcal{G}_n^s}$ in checking copositivity, we only have to solve a linear optimization problem with $n+1$ variables and $O(n^2)$ constraints after computing a singular value matrix decomposition, which implies that our algorithm is not so time-consuming. Our preliminary numerical experiments suggest that our algorithm is promising for determining upper bounds of the maximum clique problem.

Key words. Copositive programming, Matrix decomposition, Linear programming, Branch and bound algorithm, Maximum clique problem

1 Introduction

Recently, many studies have been done on the conic programming problem of the form

$$\text{(P)} \quad \begin{array}{ll} \text{Minimize} & \langle c, x\rangle \\ \text{subject to} & Ax = b,\ x \in \mathcal{K} \end{array}$$

where $\mathcal{K} \subset \mathbb{R}^n$ is a proper cone, i.e., $\mathcal{K}$ is a pointed closed convex cone with nonempty interior; $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is an inner product over $\mathbb{R}^n$; $A : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator and $b \in \mathbb{R}^m$. The dual problem (D) of (P) is given by

$$\text{(D)} \quad \begin{array}{ll} \text{Maximize} & \langle b, y\rangle \\ \text{subject to} & c - A^* y \in \mathcal{K}^* \end{array}$$

*Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan. e-mail: tanaka.akihiro@sk.tsukuba.ac.jp

†Corresponding author. Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan. e-mail: yoshise@sk.tsukuba.ac.jp Research supported in part by Grants-in-Aid for Scientific Research ((B)23310099) of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

where $\mathcal{K}^*$ is the dual cone of $\mathcal{K}$ defined by

$$\mathcal{K}^* := \{x \in \mathbb{R}^n \mid \langle x, y \rangle \geq 0 \text{ for all } y \in \mathcal{K}\}$$

and $\langle \cdot, \cdot \rangle : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is an inner product over $\mathbb{R}^m$ and A^* is the adjoint operator of A having the following relationship

$$\langle Ax, y \rangle = \langle x, A^* y \rangle.$$

Typical examples of the proper cone $\mathcal{K}$ are the n-dimensional nonnegative orthant

$$\mathbb{R}^n_+ := \{x \in \mathbb{R}^n \mid x \geq 0\}$$

in linear programming and the positive semidefinite cone

$$\mathcal{S}^+_n := \{X \in \mathcal{S}_n \mid d^T X d \geq 0 \text{ for all } x \in \mathbb{R}^n\},$$

with the set $\mathcal{S}_n$ of $n \times n$ symmetric matrices in semidefinite programming.

More recently, the following cones are attracting a lot of attention in a context of the relationship between combinatorial optimization and conic optimization (see, for example, [14, 5]).

- the nonnegative cone $\mathcal{N}_n := \{X \in \mathcal{S}_n \mid x_{ij} \geq 0 \text{ for all } i, j \in \{1, 2, \dots, n\}\}$,
- the copositive cone $\mathcal{COP}_n := \left\{X \in \mathcal{S}_n \mid d^T X d \geq 0 \ \text{ for all } d \in \mathbb{R}^n_+\right\}$,
- the Minkowski sum $\mathcal{S}^+_n + \mathcal{N}_n$ of $\mathcal{S}^+_n$ and $\mathcal{N}_n$,
- the doubly nonnegative cone $\mathcal{S}^+_n \cap \mathcal{N}_n$, i.e., the set of positive semidefinite and componentwise nonnegative matrices,
- the completely positive cone $\mathcal{CP}_n := \operatorname{conv}\left(\left\{xx^T \mid x \in \mathbb{R}^n_+\right\}\right)$ where $\operatorname{conv}(S)$ denotes the convex hull of the set S.

All of the above cones are proper (see Section 1.6 of [2] where the proper cone is called a *full cone*), and we can easily see from the definitions that the following inclusions hold:

$$\mathcal{COP}_n \supseteq \mathcal{S}^+_n \supseteq \mathcal{S}^+_n \cap \mathcal{N}_n \supseteq \mathcal{CP}_n. \tag{1}$$

It is known that the following proposition holds by defining an inner product between X and Y as

$$\langle X, Y \rangle := \operatorname{Tr}(Y^T X). \tag{2}$$

Proposition 1.1 (Properties of the copositive cone). **(i)** *The dual cone of the copositive cone $\mathcal{COP}_n$ with respect to the inner product (2) is the completely positive cone $\mathcal{CP}_n$ and vice versa (see p.57 of [1] and Theorem 2.3 of [2]).*

(ii) *If $n \leq 4$ then $\mathcal{COP}_n = \mathcal{S}^+_n + \mathcal{N}_n$ (see [11] and Proposition 1.23 of [2]).*

(iii) *The dual cone of the doubly nonnegative cone $\mathcal{S}^+_n \cap \mathcal{N}_n$ with respect to the inner product (2) is the Minkowski sum $\mathcal{S}^+_n + \mathcal{N}_n$ of the positive semidefinite cone $\mathcal{S}^+_n$ and the nonnegative cone $\mathcal{N}_n$ and vice versa (see Remark 1.2).*

Remark 1.2. *Proposition 1.1, (iii): The equality $(\mathcal{S}^+_n \cap \mathcal{N}_n)^* = \operatorname{cl}(\mathcal{S}^+_n + \mathcal{N}_n)$ follows from a well-known result that $(\mathcal{K}_1 \cap \mathcal{K}_2)^* = \operatorname{cl}(\mathcal{K}_1 + K_2)$ holds for any closed convex cones $\mathcal{K}_1$ and $\mathcal{K}_2$ (see, e.g., p.11 of [15] or Corollary 2.2 of [1]. The closedness of the set $\mathcal{S}^+_n + \mathcal{N}_n$ follows from a result in [26]. See also Proposition 4.1 of [29] where the authors showed the property in a little more general framework.*

The following inclusions follow from (1) and the above proposition

$$\mathcal{COP}_n \supseteq \mathcal{S}_n^+ + \mathcal{N}_n \supseteq \mathcal{S}_n^+ \supseteq \mathcal{S}_n^+ \cap \mathcal{N}_n \supseteq \mathcal{CP}_n \tag{3}$$

and specially, if $n \leq 4$ then we have

$$\mathcal{COP}_n = \mathcal{S}_n^+ + \mathcal{N}_n \supseteq \mathcal{S}_n^+ \supseteq \mathcal{S}_n^+ \cap \mathcal{N}_n = \mathcal{CP}_n. \tag{4}$$

Note that the four cones, $\mathcal{COP}_n$, $\mathcal{CP}_n$, $\mathcal{S}_n^+ \cap \mathcal{N}_n$ and $\mathcal{S}_n^+ + \mathcal{N}_n$ lack the self-duality and hence are not symmetric. Since about 2000, there have been many studies conducted on the above four cones as a new research direction in the field of conic optimization [3, 4, 10, 28, 24, 7, 25, 9, 20, 8, 29, 27, 21], and they are called studies on *copositive programming* [3].

A growing research interest in the field is to provide efficient algorithms to determine whether a given matrix belongs to $\mathcal{COP}_n$ (or $\mathcal{CP}_n$, or $\mathcal{S}_n^+ + \mathcal{N}_n$). It is known that the problem of testing copositivity, i.e., deciding $A \in \mathcal{COP}_n$ or not, is co-NP-complete [23, 13]. Bomze and Eichfelder [6] have pointed out what are desirable algorithms for copositivity detection as follows:

> However, there are but a few implemented numerical algorithms which (a) apply to general symmetric matrices without any structural assumptions or dimensional restrictions; (b) are not merely recursive, i.e., do not rely on information taken from all principal submatrices, but rather focus on generating subproblems in a somehow data-driven way.

After citing the paper [7] as an example satisfying criteria (a) and (b), they have presented their new tests based upon difference-of-convex (d.c.) decompositions, and have combined them to a branch-and-bound algorithm of ω-subdivision type employing LP or convex QP techniques.

In this paper, we propose a new branch and bound type algorithm based on Sponsel, Bundfuss and Dür's algorithm [27] which is a generalization of the algorithm in [7]. Two features of our algorithm are

1. we introduce new classes of matrices $\mathcal{G}_n^s$ and $\widehat{\mathcal{G}_n^s}$ which are relatively large subsets of the set of copositive matrices and work well to check copositivity of a given $n \times n$ symmetric matrix, and
2. for incorporating the sets $\mathcal{G}_n^s$ or $\widehat{\mathcal{G}_n^s}$ in checking copositivity, we only have to solve a linear optimization problem with $n+1$ variables and $O(n^2)$ constraints after computing a singular value matrix decomposition, which implies that our algorithm is not so time-consuming.

It should be noted that our algorithm is close to the one in [6] in the sense that its key elements are a decomposition of the quadratic term $d^T Xd$, simplicial partitions and LP techniques, while the number of constraints of the LP in [6] is smaller and $O(n)$. As we will describe in Section 4, the maximum clique problem can be solved by checking the copositivity of certain matrices. Our preliminary numerical experiments suggest that our algorithm and its improved versions are promising for determining upper bounds of the maximum clique problem while Bomze and Eichfelder reported effectiveness of their algorithm for determining its lower bounds[6].

This paper is organized as follows. In Section 2, we introduce Sponsel, Bundfuss and Dür's algorithm[27] for checking copositivity of a given matrix and summarize its theoretical results. Based on their algorithm, we propose our algorithm by employing new classes of matrices which are relatively large subsets of $\mathcal{COP}_n$. We also derive some properties of the classes in Section 3 and derive an improvement of our algorithm using them. Numerical results are shown in Section 4 and some refinement strategies are discussed in Section 5. Section 6 gives concluding remarks.

2 Sponsel, Bundfuss and Dür's algorithm to test copositivity

Our algorithm is based on Sponsel, Bundfuss and Dür's algorithm to test copositivity[27]. In what follows, we introduce their arguments.

Defining the standard simplex Δ^S by $\Delta^S = \{x \in \mathbb{R}^n_+ \mid \|x\|_1 = 1\}$, it can be seen that a given $n \times n$ symmetric matrix A is copositive if and only if

$$x^T A x \geq 0 \ \text{ for all } \ x \in \Delta^S$$

(see Lemma 1 of [7]). A family of simplices $\mathcal{P} = \{\Delta^1, \dots, \Delta^m\}$ is called a *simplicial partition* of Δ if it satisfies

$$\Delta = \bigcup_{i=1}^{m} \Delta^i \text{ and } \mathrm{int}(\Delta^i) \cap \mathrm{int}(\Delta^j) \neq \emptyset \text{ for all } i \neq j.$$

Such a partition can be generated by successively bisecting simplices in the partition. For a given simplex $\Delta = \mathrm{conv}\{v_1, \dots, v_n\}$, consider the midpoint $v_{n+1} = \frac{1}{2}(v_i + v_j)$ of the edge $[v_i, v_j]$. Then the subdivision $\Delta^1 = \{v_1, \dots, v_{i-1}, v_{n+1}, v_{i+1}, \dots, v_n\}$ and $\Delta^2 = \{v_1, \dots, v_{j-1}, v_{n+1}, v_{j+1}, \dots, v_n\}$ of Δ satisfies the above conditions for simplicial partitions. See [18] for a more detailed description of simplicial partitions.

Denote the set of vertices of partition $\mathcal{P}$ by

$$V(\mathcal{P}) = \{v \mid v \text{ is a vertex of some } \Delta \in \mathcal{P}\}.$$

Each simplex Δ is determined by its vertices and can be represented by a matrix V_Δ whose columns are these vertices. Note that V_Δ is nonsingular and unique up to a permutation of its columns which is irrelevant in the arguments [27]. Define the set of all matrices corresponding to simplices in partition $\mathcal{P}$ as

$$M(\mathcal{P}) = \{V_\Delta : \Delta \in \mathcal{P}\}.$$

The "fineness" of a partition $\mathcal{P}$ is quantified by the maximum diameter of a simplex in $\mathcal{P}$ denoted by

$$\delta(\mathcal{P}) = \max_{\Delta \in \mathcal{P}} \max_{u,v \in \Delta} ||u - v||. \tag{5}$$

Using the above notation, the following results on necessary and sufficient conditions for copositivity have been shown in [27]. The first theorem gives a sufficient condition for copositivity.

Theorem 2.1 (Theorem 2.1 of [27])**.** *If $A \in \mathcal{S}_n$ satisfies*

$$V^T A V \in \mathcal{COP}_n \text{ for all } V \in M(\mathcal{P})$$

then A is copositive. Hence, for any $\mathcal{M}_n \subseteq \mathcal{COP}_n$, if $A \in \mathcal{S}^n$ satisfies

$$V^T A V \in \mathcal{M}_n \text{ for all } V \in M(\mathcal{P})$$

then A is also copositive.

The above theorem implies that by choosing $\mathcal{M}_n = \mathcal{N}_n$ (see (3)), if $V_\Delta^T A V_\Delta \in \mathcal{N}_n$ holds for any $\Delta \in \mathcal{P}$ then we find that A is copositive.

Before describing further conditions for copositivity, we introduce the definition of *strict copositivity.* We say that $A \in \mathcal{S}_n$ is strictly copositive if it satisfies

$$x^T Ax > 0 \text{ for all } x \in \mathbb{R}^n_+ \setminus \{0\}.$$

It is well-known (and follows from Proposition 1.24 of [2]) that $A \in \mathcal{S}_n$ is strictly copositive if and only if $A \in \operatorname{int}(\mathcal{COP}_n)$. Combining this with Theorem 2.2 of [27], we obtain the following necessary condition for strict copositivity.

Theorem 2.2 (Theorem 2.2 of [27])**.** *Let $A \in \mathcal{S}_n$ be strictly copositive, i.e., $A \in \operatorname{int}(\mathcal{COP}_n)$. Then there exists $\varepsilon > 0$ such that for all partitions $\mathcal{P}$ of Δ^S with $\delta(\mathcal{P}) < \varepsilon$ we have*

$$V^T AV \in \mathcal{N}_n \text{ for all } V \in M(\mathcal{P}).$$

The above theorem ensures that if A is strictly copositive (i.e., $A \in \operatorname{int}(\mathcal{COP}_n)$) then the copositivity of A (i.e., $A \in \mathcal{COP}_n$) can be detected in finitely many steps by an algorithm employing a subdivision rule with $\delta(\mathcal{P}) \to 0$. A similar result can be obtained for the case $A \notin \mathcal{COP}_n$ by the following lemma.

Lemma 2.3 (Lemma 2.3 of [27])**.** *The following two statements are equivalent.*

1. *$A \notin \mathcal{COP}_n$*
2. *There exists an $\varepsilon > 0$ such that for any partition $\mathcal{P}$ with $\delta(\mathcal{P}) < \varepsilon$ there exists a vertex $v \in V(\mathcal{P})$ such that $v^T Av < 0$.*

Based on the above three results, the following algorithm has been provided by Sponsel, Bundfuss and Dür [27].

Algorithm 1 Sponsel, Bundfuss and Dür's algorithm to test copositivity

Input: $A \in \mathcal{S}_n, \mathcal{M}_n \subseteq \mathcal{COP}_n$
Output: "A is copositive" or "A is not copositive"

$\mathcal{P} \leftarrow \{\Delta^S\}$;
while $\mathcal{P} \neq \emptyset$ **do**
  Choose $\Delta \in \mathcal{P}$;
  if $v^T Av < 0$ for some $v \in V(\{\Delta\})$: **then**
    return "A is not copositive";
  end if
  if $V_\Delta^T AV_\Delta \in \mathcal{M}_n$ **then**
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \{\Delta\}$;
  else
    partition Δ into $\Delta = \Delta^1 \cup \Delta^2$;
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \{\Delta\} \cup \{\Delta^1, \Delta^2\}$;
  end if
end while
return "A is copositive";

As we have already observed, Theorem 2.2 and Lemma 2.3 imply the following corollary.

Corollary 2.4.
1. *If A is strictly copositive, i.e., $A \in \operatorname{int}(\mathcal{COP}_n)$ then Algorithm 1 terminates finitely returning "A is copositive."*
2. *If A is not copositive, i.e., $A \notin \mathcal{COP}_n$ then Algorithm 1 terminates finitely returning "A is not copositive."*

At Line 8 of Algorithm 1, the algorithm removes the simplex which is determined at Line 7 to be in no need of further exploration by Theorem 2.1. The accuracy and speed of the determination influence the total computational time and depend on the choice of the set $\mathcal{M}_n \subseteq \mathcal{COP}_n$. In the next section, we introduce three examples of $\mathcal{M}_n$, the set used in [27], our alternative suggestion and its generalization.

3 How to choose $\mathcal{M}_n$ to efficiently remove unnecessary simplices

In view of Algorithm 1, a desirable set $\mathcal{M}_n$ used at Line 7 would have the following properties.

P1 For any given $n \times n$ symmetric matrix $A \in \mathcal{S}_n$, we can easily check whether $A \in \mathcal{M}_n$, and

P2 $\mathcal{M}_n$ is a subset of the copositive cone $\mathcal{COP}_n$ as large as possible.

If we choose the nonnegative cone $\mathcal{N}_n$ as the set $\mathcal{M}_n$, we can easily check whether $A \in \mathcal{M}_n$ or not, but the set $\mathcal{N}_n$ is too small a subset of $\mathcal{COP}_n$ and it may take a long time to check the copositivity by Algorithm 1. In fact, $\mathcal{M}_n = \mathcal{N}_n$ was used in [7]. On the other hand, the set $\mathcal{S}_n^+ + \mathcal{N}_n$ is a rather large subset of $\mathcal{COP}_n$, but it is not so easy to check whether $A \in \mathcal{M}_n = \mathcal{S}_n^+ + \mathcal{N}_n$ or not; a well-known way is to solve the following doubly nonnegative program (which can be expressed as a semidefinite program)

$$\begin{array}{ll} \text{Minimize} & \langle A, X \rangle \\ \text{subject to} & \langle I, X \rangle = 1,\ X \in \mathcal{S}_n^+ \cap \mathcal{N}_n \end{array}$$

but solving the problem takes an awful lot of time [27, 29].

Observing these facts, a new alternate of $\mathcal{M}_n$ has been provided in [27]. Before stating its definition, we need to introduce some additional notation. For any given matrix $A \in \mathcal{S}_n$, we denote

$$N(A)_{ij} := \begin{cases} A_{ij} & A_{ij} > 0 \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad S(A) := A - N(A). \tag{6}$$

In [27], the authors defined the following set

$$\mathcal{H}_n := \{A \in \mathcal{S}_n \mid S(A) \in \mathcal{S}_n^+\}. \tag{7}$$

Note that $A = S(A) + N(A) \in \mathcal{S}_n^+ + \mathcal{N}_n$ if $A \in \mathcal{H}_n$. Also, for any $A \in \mathcal{N}_n$, $S(A)$ becomes a nonnegative diagonal matrix and hence $\mathcal{N}_n \subseteq \mathcal{H}_n$. The detection whether $A \in \mathcal{H}_n$ is easy and can be done by checking positivity of $A_{ij} (i \neq j)$ and by a Cholesky factorization of $S(A)$ (cf. Algorithm 4.2.4 in [17]). Thus, by the inclusion relation (3), we see that the set $\mathcal{H}_n$ satisfies the desirable properties **P1** and **P2** of $\mathcal{M}_n$. However, $S(A)$ is not necessarily positive semidefinite even if $A \in \mathcal{S}_n^+ + \mathcal{N}_n$ or $A \in \mathcal{S}_n^+$. The following theorem summarizes several properties of the set $\mathcal{H}_n$.

Theorem 3.1 ([16] and Theorem 4.2 of [27])**.** *$\mathcal{H}_n$ is a convex cone and $\mathcal{N}_n \subseteq \mathcal{H}_n \subseteq \mathcal{S}_n^+ + \mathcal{N}_n$. If $n \geq 3$, these inclusions are strict and $\mathcal{S}_n^+ \not\subseteq \mathcal{H}_n$. For $n = 2$, we have $\mathcal{H}_n = \mathcal{S}_n^+ \cup \mathcal{N}_n = \mathcal{S}_n^+ + \mathcal{N}_n = \mathcal{COP}_n$.*

The construction of the set $\mathcal{H}_n$ is based on the idea of "nonnegativity-checking first and positive semidefiniteness-checking second." Now, we provide an alternative choice of $\mathcal{M}_n$ based on the idea of "positive semidefiniteness-checking first and nonnegativity-checking second."

For a given symmetric matrix $A \in \mathcal{S}_n$, let P be an orthonormal matrix and $\Lambda = \text{Diag}\,(\lambda_1, \lambda_2, \ldots, \lambda_n)$ be a diagonal matrix satisfying

$$A = P \Lambda P^T. \tag{8}$$

We are interested in decomposing A into a semidefinite matrix and a nonnegative matrix according to the form $A = P\Lambda P^T$. By introducing another diagonal matrix $\Omega = \mathrm{Diag}(\omega_1, \omega_2, \dots, \omega_n)$, consider the following decomposition:

$$A = P(\Lambda - \Omega)P^T + P\Omega P^T \tag{9}$$

If $\Lambda - \Omega \in \mathcal{N}_n$, i.e., $\lambda_i \geq \omega_i$ $(i = 1, 2, \dots, n)$ hold, then the matrix $P(\Lambda - \Omega)P^T$ is positive semidefinite. Thus, if we can find a suitable diagonal matrix Ω satisfying

$$\lambda_i \geq \omega_i \ (i = 1, 2, \dots, n), \quad [P\Omega P^T]_{ij} \geq 0 \ (i, j = 1, 2, \dots, n, i \leq j) \tag{10}$$

then (9) and (3) imply

$$A = P(\Lambda - \Omega)P^T + P\Omega P^T \in \mathcal{S}_n^+ + \mathcal{N}_n \subseteq \mathcal{COP}_n. \tag{11}$$

We can determine whether such a matrix exists or not by solving the following linear optimization problem with variables ω_i $(i = 1, 2, \dots, n)$ and α:

$$(\mathrm{LP})_{P,\Lambda} \quad \begin{array}{lll} \text{Maximize} & \alpha & \\ \text{subject to} & \omega_i \leq \lambda_i & (i = 1, 2, \dots, n) \\ & [P\Omega P^T]_{i,j} = \sum_{k=1}^n \omega_k p_{ik} p_{jk} \geq \alpha & (i, j = 1, 2, \dots, n, i \leq j) \end{array} \tag{12}$$

Note that $(\mathrm{LP})_{P,\Lambda}$ has the feasible solution at which $\omega_i = \lambda_i$ $(i = 1, 2, \dots, n)$ and $\alpha = \min_{ij} \sum_{k=1}^n \lambda_k p_{ik} p_{jk}$ and hence has an optimal solution with optimal value $\alpha^*(P, \Lambda)$. If $\alpha^*(P, \Lambda) \geq 0$ then there exists a matrix Ω for which the decomposition (10) holds. Based on these observations, we provide another alternate $\mathcal{G}_n^s$ of $\mathcal{M}_n$ as follows:

$$\mathcal{G}_n^s := \{A \in \mathcal{S}_n \mid \alpha^*(P, \Lambda) \geq 0 \text{ for } \textit{some} \text{ orthonormal matrix } P \text{ satisfying (8) }\}. \tag{13}$$

As stated above, if $\alpha^*(P, \Lambda) \geq 0$ for a given decomposition $A = P\Lambda P^T$ then we can determine $A \in \mathcal{G}_n^s$. In this case, we just need to compute a matrix decomposition and to solve a linear optimization problem with $n + 1$ variables and $O(n^2)$ constraints which implies that it is rather practical to use the set $\mathcal{G}_n^s$ as an alternate of $\mathcal{M}_n$ Suppose that $A \in \mathcal{S}_n$ has n different eigenvalues. Then the possible orthonormal matrices $P = [p_1, p_2, \cdots, p_n]$ are identifiable except for permutation and sign inversion of $\{p_1, p_2, \cdots, p_n\}$ and by the representation

$$A = \sum_{i=1}^n \lambda_i p_i p_i^T$$

of (8), we see that the problem $(\mathrm{LP})_{P,\Lambda}$ is unique for any possible P. In this case, $\alpha^*(P, \Lambda) < 0$ with a specific P implies $A \notin \mathcal{G}_n^s$. However, otherwise (i.e., an eigenspace of A has at least dimension 2), $\alpha^*(P, \Lambda) < 0$ with a specific P does not necessarily guarantee that $A \notin \mathcal{G}_n^s$. So we cannot say that the set $\mathcal{G}_n^s$ satisfies the desirable property **P1** of $\mathcal{M}_n$. However, as we see in Theorem 3.2 below, $\mathcal{G}_n^s$ may satisfy the other desirable property **P2**.

Let us introduce other new sets $\mathcal{G}_n^a$ and $\widehat{\mathcal{G}_n^s}$ which are closely related to the set $\mathcal{G}_n^s$ and they might be useful to clarify some theoretical properties or to improve our algorithm:

$$\mathcal{G}_n^a := \{A \in \mathcal{S}_n \mid \alpha^*(P, \Lambda) \geq 0 \text{ for } \textit{any} \text{ orthonormal matrix } P \text{ satisfying (8) }\}, \tag{14}$$

$$\widehat{\mathcal{G}_n^s} := \{A \in \mathcal{S}_n \mid \alpha^*(P, \Lambda) \geq 0 \text{ for } \textit{some arbitrary} \text{ matrix } P \text{ satisfying (8) }\}. \tag{15}$$

Note that if (10) holds for any arbitrary (not necessarily orthonormal) matrix P then (11) also holds, which implies the following inclusions:

$$\mathcal{G}_n^a \subseteq \mathcal{G}_n^s \subseteq \widehat{\mathcal{G}_n^s} \subseteq \mathcal{S}_n^+ + \mathcal{N}_n. \tag{16}$$

More precisely, the sets $\mathcal{G}_n^s$, $\mathcal{G}_n^a$ and $\widehat{\mathcal{G}}_n^s$ have the following properties.

Theorem 3.2. *The sets $\mathcal{G}_n^s$, $\mathcal{G}_n^a$ and $\widehat{\mathcal{G}}_n^s$ are cones and*

$$\mathcal{S}_n^+ \cup \mathcal{N}_n \subseteq \mathcal{G}_n^a \subseteq \mathcal{G}_n^s = \mathrm{com}(\mathcal{S}_n^+ + \mathcal{N}_n) \subseteq \widehat{\mathcal{G}_n^s} \subseteq \mathcal{S}_n^+ + \mathcal{N}_n \subseteq \mathcal{COP}_n$$

where the set $\mathrm{com}(\mathcal{S}_n^+ + \mathcal{N}_n)$ *is defined by*

$$\mathrm{com}(\mathcal{S}_n^+ + \mathcal{N}_n) := \{S + N \mid S \in \mathcal{S}_n^+,\ N \in \mathcal{N}_n,\ S \text{ and } N \text{ commute}\}.$$

For $n = 2$, we have

$$\mathcal{S}_n^+ \cup \mathcal{N}_n = \mathcal{G}_n^a = \mathcal{G}_n^s = \mathrm{com}(\mathcal{S}_n^+ + \mathcal{N}_n) = \widehat{\mathcal{G}_n^s} = \mathcal{S}_n^+ + \mathcal{N}_n = \mathcal{COP}_n.$$

Proof. We assume that $A \in \mathcal{S}_n$ is diagonalized as in (8) throughout the proof.

Suppose that the associated linear optimization problem $(\mathrm{LP})_{P,\Lambda}$ has an optimal solution $(\omega^*, \alpha^*) := (\omega_1^*, \dots, \omega_n^*, \alpha^*)$. Then for any $\beta \geq 0$, βA is diagonalized as in $\beta A = P(\beta\Lambda)P^T$ and $(\beta\omega^*, \beta\alpha^*)$ is an optimal solution of the associated linear optimization problem $(\mathrm{LP})_{P,\beta\Lambda}$. This implies that $\beta A \in \mathcal{G}_n^s$ (respectively $\beta A \in \mathcal{G}_n^a$, respectively $\beta A \in \widehat{\mathcal{G}_n^s}$) if $A \in \mathcal{G}_n^s$ (respectively $A \in \mathcal{G}_n^a$, respectively $A \in \widehat{\mathcal{G}_n^s}$) and hence $\mathcal{G}_n^s$, $\mathcal{G}_n^a$ and $\widehat{\mathcal{G}_n^s}$ are cones.

We have already seen that (16) holds. So it is sufficient to show that (i) $\mathcal{S}_n^+ \cup \mathcal{N}_n \subseteq \mathcal{G}_n^a$ and (ii) $\mathcal{G}_n^s = \mathrm{com}(\mathcal{S}_n^+ + \mathcal{N}_n)$.

(i) $\mathcal{S}_n^+ \cup \mathcal{N}_n \subseteq \mathcal{G}_n^a$: Let us show that $\mathcal{N}_n \subseteq \mathcal{G}_n^a$ and $\mathcal{S}_n^+ \subseteq \mathcal{G}_n^a$, respectively. Suppose that $A \in \mathcal{N}_n$. Then for all P the problem $(\mathrm{LP})_{P,\Lambda}$ has a feasible solution where $(\omega, \alpha) = (\lambda_1, \dots, \lambda_n, 0)$ which implies that $A \in \mathcal{G}_n^a$. Suppose that $A \in \mathcal{S}_n^+$, i.e., $\lambda_i \geq 0$ $(i = 1, 2, \dots, n)$. Then for all P the problem $(\mathrm{LP})_{P,\Lambda}$ has a feasible solution where $(\omega, \alpha) = (0, \dots, 0, 0)$ which implies that $A \in \mathcal{G}_n^a$. Thus we have shown $\mathcal{S}_n^+ \cup \mathcal{N}_n \subseteq \mathcal{G}_n^a$.

(ii) $\mathcal{G}_n^s = \mathrm{com}(\mathcal{S}_n^+ + \mathcal{N}_n)$: The inclusion $\mathcal{G}_n^s \subseteq \mathrm{com}(\mathcal{S}_n^+ + \mathcal{N}_n)$ follows from the construction of the set $\mathcal{G}_n^s$ as in (13) and (12). The converse inclusion $\mathcal{G}_n^s \supseteq \mathrm{com}(\mathcal{S}_n^+ + \mathcal{N}_n)$ is also true since if $A \in \mathrm{com}(\mathcal{S}_n^+ + \mathcal{N}_n)$ then there exist an orthonormal matrix P and diagonal matrices $\Theta = \mathrm{Diag}\,(\theta_1, \theta_2, \dots, \theta_n)$ and $\Omega = \mathrm{Diag}\,(\omega_1, \omega_2, \dots, \omega_n)$ such that

$$A = P\Theta P^T + P\Omega P^T,\ P\Theta P^T \in \mathcal{S}_n^+,\ P\Omega P^T \in \mathcal{N}_n$$

(see Theorem 1.3.12 of [19]) which implies that $\theta_i \geq 0$ $(i = 1, 2, \dots, n)$ and that the problem $(\mathrm{LP})_{P,\Lambda}$ with $\Lambda = \Theta + \Omega$ has a nonnegative objective value at a solution (ω, α) where $\alpha = \min_{i,j}\{[P\Omega P^T]_{ij}\} \geq 0$.

The results for $n = 2$ follow from Theorem 3.1. □

As we have seen in Theorem 3.1, $\mathcal{N}_n \subseteq \mathcal{H}_n$ but $\mathcal{S}_n^+ \not\subseteq \mathcal{H}_n$ for $n \geq 3$. Theorem 3.2 suggests that the set $\mathcal{G}_n^s$ might be better than the set $\mathcal{H}_n$ in the sense of the desirable property **(P2)** of $\mathcal{M}_n$. The following examples show some contrasts between $\mathcal{H}_n$, $\mathcal{G}_n^s$ and $\mathcal{G}_n^a$.

Example 3.3. *Consider*

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix}.$$

Then, by the definition (6),

$$S(A) = A - N(A) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \in \mathcal{S}_3^+$$

which implies that $A \in \mathcal{H}_3$. *Moreover,*

$$N(A)S(A) = S(A)N(A) = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

which implies that $A = S(A) + N(A) \in \operatorname{com}(\mathcal{S}_3^+ + \mathcal{N}_3)$, *and by Theorem 3.2,* $A \in \mathcal{G}_3^s$ *holds. Thus* $\mathcal{H}_3 \cap \mathcal{G}_3^s \neq \emptyset$.

Example 3.4 (cf. Proof of Theorem 4.2 in [27])**.** *Consider*

$$A = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}.$$

Then $A \in \mathcal{S}_3^+$ *and by Theorem 3.2, we see that* $A \in \mathcal{G}_3^s$. *However,*

$$S(A) = A - N(A) = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \notin \mathcal{S}_3^+$$

which implies that $A \notin \mathcal{H}_3$. *Thus* $\mathcal{G}_3^s \setminus \mathcal{H}_3 \neq \emptyset$.

Example 3.5. *Consider*

$$A = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

and let

$$S = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } N = A - S = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Then $S \in \mathcal{S}_3^+, N \in \mathcal{N}_3$ *and*

$$SN = NS = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

holds which implies that $A \in \operatorname{com}(\mathcal{S}_3^+ + \mathcal{N}_3) \subseteq \mathcal{G}_3^s$. *Moreover, if we set*

$$P := \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{14}} & \frac{5}{\sqrt{42}} \\ \frac{1}{\sqrt{3}} & -\frac{3}{\sqrt{14}} & -\frac{1}{\sqrt{42}} \\ -\frac{1}{\sqrt{3}} & -\frac{2}{\sqrt{14}} & \frac{4}{\sqrt{42}} \end{bmatrix}, \Lambda := \begin{bmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

then P *and* Λ *satisfy (8) and the corresponding problem* $(\mathrm{LP})_{P,\Lambda}$ *is given as follows:*

$$\begin{array}{ll} \text{Maximize} & \alpha \\ \text{subject to} & \omega_1 \leq -1, \omega_2 \leq 2, \omega_3 \leq 2 \\ & \omega_1 \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{1}{3} \end{bmatrix} + \omega_2 \begin{bmatrix} \frac{1}{14} & -\frac{3}{14} & -\frac{1}{7} \\ -\frac{3}{14} & \frac{9}{14} & \frac{3}{7} \\ -\frac{1}{7} & \frac{3}{7} & \frac{2}{7} \end{bmatrix} + \omega_3 \begin{bmatrix} \frac{25}{42} & -\frac{5}{42} & \frac{10}{21} \\ -\frac{5}{42} & \frac{1}{42} & -\frac{2}{21} \\ \frac{10}{21} & -\frac{2}{21} & \frac{8}{21} \end{bmatrix} \geq \alpha E. \end{array}$$

By solving this problem, we know that $\alpha^*(P, \Lambda) < 0$. *Thus the matrix* A *lies on* $\mathcal{G}_3^s$ *but not on* $\mathcal{G}_3^a$. *Thus* $\mathcal{G}_3^s \setminus \mathcal{G}_3^a \neq \emptyset$.

In the next section, we will show numerical results of the following three algorithms:

Algorithm 1.1: The set $\mathcal{H}_n$ is used for $\mathcal{M}_n$ at Line 7 of Algorithm 1, i.e., the original algorithm proposed in [27].

Algorithm 1.2: The set $\mathcal{G}_n^s$ is used for $\mathcal{M}_n$ at Line 7 of Algorithm 1.

Algorithm 2: An improved version of Algorithm 1.2. The set $\mathcal{G}_n^s$ is used for $\mathcal{M}_n$ at Line 7 of Algorithm 1. Moreover, based on the fact that $\widehat{\mathcal{G}_n^s} \subseteq \mathcal{COP}_n$, some additional tests to remove simplices have been incorporated at Lines 7 and 10 of Algorithm 1.

Algorithm 2 An improved version of Algorithm 1.2

Input: $A \in \mathcal{S}_n, \mathcal{M}_n \subseteq \mathcal{COP}_n$
Output: "A is copositive" or "A is not copositive"
1: $\mathcal{P} \leftarrow \{\Delta^S\}$;
2: **while** $\mathcal{P} \neq \emptyset$ **do**
3: Choose $\Delta \in \mathcal{P}$;
4: **if** $v^T Av < 0$ for some $v \in V(\{\Delta\})$: **then**
5: **return** "A is not copositive";
6: **end if**
7: **if** $V_\Delta^T AV_\Delta \in \widehat{\mathcal{G}_n^s}$ **then**
8: $\mathcal{P} \leftarrow \mathcal{P} \setminus \{\Delta\}$;
9: **else**
10: **if** $V_\Delta^T AV_\Delta \in \mathcal{G}_n^s$ **then**
11: $\mathcal{P} \leftarrow \mathcal{P} \setminus \{\Delta\}$;
12: **else**
13: partition Δ into $\Delta = \Delta^1 \cup \Delta^2$ and set $\widehat{\Delta} \leftarrow \{\Delta^1, \Delta^2\}$;
14: **for** $p = 1, 2$ **do**
15: **if** $V_{\Delta^p}^T AV_{\Delta^p} \in \widehat{\mathcal{G}_n^s}$ **then**
16: $\widehat{\Delta} \leftarrow \widehat{\Delta} \setminus \{\Delta^p\}$;
17: **end if**
18: **end for**
19: $\mathcal{P} \leftarrow \mathcal{P} \setminus \{\Delta\} \cup \widehat{\Delta}$;
20: **end if**
21: **end if**
22: **end while**
23: **return** "A is copositive";

The details of the added steps in Algorithm 2 are as follows. Suppose that we have a diagonalization of the form (8) in advance.

At Line 7, we will solve an additional LP but need not diagonalize $V_\Delta^T AV_\Delta$. Let P and Λ be matrices satisfying (8). Then the matrix $V_\Delta^T P$ gives the diagonalization of $V_\Delta^T AV_\Delta$, i.e.,

$$V_\Delta^T AV_\Delta = V_\Delta^T (P\Lambda P^T)V_\Delta^T = (V_\Delta^T P)\Lambda(V_\Delta^T P)^T$$

while $V_\Delta^T P$ is not necessarily orthonormal. Thus we can test $V_\Delta^T AV_\Delta \in \widehat{\mathcal{G}_n^s}$ by solving $(\mathrm{LP})_{V_\Delta^T P,\Lambda}$.

If $V_\Delta^T AV_\Delta \in \widehat{\mathcal{G}_n^s}$ does not detected at Line 7, we will check whether $V_\Delta^T AV_\Delta \in \mathcal{G}_n^s$ at Line 10. Similarly to Algorithm 1.2 (where the set $\mathcal{G}_n^s$ is used for $\mathcal{M}_n$ at Line 7 of Algorithm 1), we will diagonalize $V_\Delta^T AV_\Delta$ as $V_\Delta^T AV_\Delta = P\Lambda P^T$ with an orthonomal matrix P and a diagonal matrix Λ, and solve $(\mathrm{LP})_{P,\Lambda}$.

At Line 15, we neither need diagonalize $V_{\Delta^p}^T A V_{\Delta^p}$ nor to solve any further LPs. Let $\omega^* \in \mathbb{R}^n$ be an optimal solution of $(\mathrm{LP})_{V_\Delta^T P,\Lambda}$ obtained at Line 7 and let $\Omega^* := \mathrm{Diag}\,(\omega^*)$. Then the feasibility of ω^* implies the positive semidefiniteness of the matrix $V_{\Delta^p}^T P(\Lambda - \Omega^*)P^T V_{\Delta^p}$. Thus, if $V_{\Delta^p}^T P\Omega^* P^T V_{\Delta^p} \in \mathcal{N}_n$ then we see that

$$V_{\Delta^p}^T A V_{\Delta^p} = V_{\Delta^p}^T P(\Lambda - \Omega^*)P^T V_{\Delta^p} + V_{\Delta^p}^T P\Omega^* P^T V_{\Delta^p} \in \mathcal{S}_n^+ + \mathcal{N}_n$$

and that $V_{\Delta^p}^T A V_{\Delta^p} \in \widehat{\mathcal{G}_n^s}$.

4 Numerical results

We implemented Algorithms 1.1, 1.2 and 2 in MATLAB to compare the performance of those algorithms.

As the test-instances, we used the following matrix

$$B_\gamma := \gamma(E - A_G) - E \tag{17}$$

where $\gamma \geq 1$, $E \in \mathcal{S}_n$ is the matrix whose elements are all one and the matrix $A_G \in \mathcal{S}_n$ is the adjacency matrix of a given undirected graph G with n nodes. The matrix B_γ comes from the maximum clique problem. The maximum clique problem is to find a clique (complete subgraph) of maximum cardinality in G. It has been shown in [10] that the maximum cardinality, the so-called clique number $\omega(G)$, is equal to the optimal value of

$$\omega(G) = \min\{\gamma \in \mathbb{N} \mid B_\gamma \in \mathcal{COP}_n\}.$$

Thus, the clique number can be found by checking the copositivity of B_γ at most for $\gamma = n, n-1, \ldots, 1$.

Note that in [27], the authors showed that $B_\gamma \notin \mathcal{COP}_n$ if $\gamma < \omega(G)$, $B_\gamma \in \mathcal{COP}_n \setminus \mathrm{int}\,(\mathcal{COP}_n)$ if $\gamma = \omega(G)$ and $B_\gamma \in \mathrm{int}\,(\mathcal{COP}_n)$ if $\gamma > \omega(G)$ (see Proposition 3.2 of [27]). The results and Corollary 2.4 imply that the algorithms may fail to terminate if $\gamma = \omega(G)$. In [27], the authors have provided a modified copositive program to avoid such difficulties as the following theorem.

Theorem 4.1 (Theorem 3.3 in [27])**.** *The clique number $\omega(G)$ of a given graph G can be obtained from the following modified copositive program:*

$$\omega(G) = \min\{\gamma \in \mathbb{N} \mid B_\gamma + \rho E \in \mathcal{COP}_n\}$$

if $0 \leq \rho < 1/\omega(G)$. Moreover, $B_\gamma + \rho E$ is strictly copositive for any $\gamma \geq \omega(G)$ and $\rho > 0$.

An aim of the implementation is to explore the difference of behaviors between the choices $\mathcal{M}_n = \mathcal{H}_n$ and $\mathcal{M}_n = \mathcal{G}_n^s$ rather than to compute the clique number efficiently. So we conducted our experiment to examine B_γ for various values of γ at intervals of 0.1 around the value $\omega(G)$ (Tables 1 and 2 on page 17). We also solved the modified problem in Theorem 4.1 to confirm the efficacy of the modification (Table 5 on page 19).

Figure 1 on page 17 shows our instances for G that have been used in [27]. We know the clique numbers of G_8 and G_{12} are $\omega(G_8) = 3$ and $\omega(G_{12}) = 4$, respectively.

For a given $A \in \mathcal{S}_n$, we used the MATLAB command "$[P, \Lambda] = \mathrm{eig}(A)$" to obtain the diagonalized form (8). As already mentioned above, $\alpha^*(P, \Lambda) < 0$ with a specific P does not necessarily guarantee that $A \notin \mathcal{G}_n^s$ ($A \notin \widehat{\mathcal{G}_n^s}$). Thus, it not strictly accurate to say that we have used $\mathcal{G}_n^s$ ($\widehat{\mathcal{G}_n^s}$) for $\mathcal{M}_n$ and the algorithms may miss some removable Δ's. Note that this may have some effect on speed but not on termination of the algorithm since the termination is guaranteed by the subdivision rule satisfying $\delta(\mathcal{P}) \to 0$ where $\delta(\mathcal{P})$ is defined by (5).

The performance of algorithms is also influenced by strategies to refine the simplex Δ used at Line 10 of Algorithm 1 or Line 13 of Algorithm 2. We employed the most classical longest-edge bisection rule as a common strategy among the experiments, i.e., we choose the longest edge of a given simplex Δ and bisect the edge with the fixed bisection ratio 1 : 1. It is well known that this type of longest-edge bisection rule generates a sequence such that $\delta(\mathcal{P}) \to 0$ ([18], see also [12]).

There have been different strategies on refinement of simplices [7, 8, 27]. Among others, an efficient strategy for the set $\mathcal{M}_n = \mathcal{H}_n$ has been provided in [27]. We will discuss our improvement on refinement for the set $\mathcal{M}_n = \mathcal{G}_n^s$ in the next section.

We tested our implementation on a 3.07GHz Core i7 machine with 12 GB of RAM. Tables 1 and 2 represent the numerical results for the graphs G_8 and G_{12}, respectively. In both tables, the symbol "−" means that the algorithm did not terminate within 6 hours. These results may come from the fact that for each graph G, the matrix B_γ lies on the boundary of the copositive cone $\mathcal{COP}_n$ when $\gamma = \omega(G)$ ($\omega(G_8) = 3$ and $\omega(G_{12}) = 4$).

We observe similar trends in Tables 1 and 2, and explain the implications of our results using Table 2 on page 18 for the larger graph G_{12}. The results in Table 2 imply that

- at any $\gamma \geq 5.2$, Algorithm 1.2 terminates in one iteration and its execution time is faster than the one of Algorithm 1.1. The reason may be that, as shown in Theorem 3.2, the set $\mathcal{G}_n^s$ is a relatively large subset of $\mathcal{COP}_n$ and useful to check copositivity of the matrix A if A is (strictly) copositive.
- at any $\gamma \in \{5.1, 5.0, \ldots, 4.7\}$, Algorithm 1.1 terminates within 4.2 hours while Algorithm 1.2 does not terminate within 6 hours. The results imply that the execution time of Algorithm 1.2 is slower than the one of Algorithm 1.1 since Algorithm 1.2 requires additional computation for solving an $n+1$ variables and $O(n^2)$ constraints linear optimization problem at each iteration. The execution time of the improved version, Algorithm 2, is substantially better than the one of Algorithm 1.2, but is not better than one of Algorithm 1.1.
- at each $\gamma < 4$, Algorithms 1.1, 1.2 and 2 have no significant differences in terms of the number of iterations since both of algorithms work to find a $v \in V(\{\Delta\})$ such that $v^T(\gamma(E - A_G) - E)v < 0$ while its computational time depends on our choice of simplex refinement strategy but not on the choice of $\mathcal{M}_n$.

In view of the above observations, we conclude that Algorithm 1.2 with the choice $\mathcal{M}_n = \mathcal{G}_n^s$ might be a promising way to check copositivity of a given matrix A when A is strictly copositive. In addition, the improved technique used in Algorithm 2 has a pronounced effect on the number of iterations and hence the execution time of Algorithm 1.2. This can be seen in Tables 3 and 4 where the columns "Line 8," "Line 11" and "Line 19" show the number of simplices removed at each line of Algorithm 2. The elimination of these simplices contributes to improve Algorithm 1.2.

Next we implemented the three algorithms to solve the modified programs provided in Theorem 4.1. The obtained results are shown in Table 5 on page 19. Note that Theorem 4.1 involves only integer values of γ. So we tested the matrices $B_4 + 0.199E$ ($0.199 < \frac{1}{5}$) and $B_3 + 0.249E$ ($0.249 < \frac{1}{4}$) of the graph G_8 with $\omega(G_8) = 3$, and the matrices $B_5 + 0.166E$ ($0.166 < \frac{1}{6}$) and $B_4 + 0.199E$ ($0.199 < \frac{1}{5}$) of the graph G_{12} with $\omega(G_{12}) = 4$. These results may be compared to the results for $\gamma = 3.9$ in Table 1 for the graph G_8 and the ones for $\gamma = 4.9$ in Table 2 for the graph G_{12} since the copositivity of $B_{3.9}$ implies $\omega(G_8) \leq 3$ for G_8 and the copositivity of $B_{4.9}$ implies $\omega(G_{12}) \leq 4$ for G_{12}, respectively. These comparisons suggest that for all algorithms, Algorithms 1.1, 1.2 and 2, the modified programs have positive effects especially for detecting $\omega(G_{12})$.

5 Improved strategies for refinement of simplices

In this section, we discuss our strategies for refinement of simplices for the improved version, Algorithm 2.

We first introduce the strategy for Algorithm 1.1 provided in [27]. Suppose that we choose $\mathcal{M}_n = \mathcal{N}_n$ at Line 10 of Algorithm 1. If $V_\Delta^T AV_\Delta \notin \mathcal{M}_n$ then there exist i and j such that $v_i^T Av_j < 0$. If $i = j$ then since $v_i \in \Delta$, we find that $A \notin \mathcal{COP}_n$ and the algorithm terminates and otherwise it would be natural to partition an edge $\{v_i, v_j\}$ which attains the optimal value of $\min_{i,j\in\{1,2,\ldots,n\},i\neq j} v_i^T Av_j < 0$.

Adopting the idea for the case $\mathcal{M}_n = \mathcal{H}_n$ defined in (7), Sponsel, Bundfuss and Dür [27] suggested a strategy to partition an edge $\{v_i, v_j\}$ which gives the optimal value of

$$\min_{i,j\in\{1,2,\ldots,n\},i\neq j} S(V_\Delta^T AV_\Delta)_{ij} x_i x_j$$

and have shown the numerical results using the strategy.

We adopt the same idea for the cases $\mathcal{M}_n = \mathcal{G}_n^s$ and $\mathcal{M}_n = \widehat{\mathcal{G}_n^s}$ where the definitions are given in (13) and (15), respectively. If $V_\Delta^T AV_\Delta \notin \mathcal{G}_n^s$ ($V_\Delta^T AV_\Delta \notin \widehat{\mathcal{G}_n^s}$) then for any orthonormal matrix (for any arbitrary) P which gives a diagonalization $V_\Delta^T AV_\Delta = P\Lambda P^T$, the feasible linear optimization $(\mathrm{LP})_{P,\Lambda}$ has the negative optimal value $\alpha^*(P,\Lambda) < 0$.

Our strategy, we call it the "negative-edge bisection rule," is to partition an edge $\{v_i, v_j\}$ which attains the optimal value

$$\sum_{k=1}^{n} \omega_k^* p_{ik} p_{jk} = \alpha^*(P,\Lambda) < 0$$

at the optimal solution (ω^*, α^*). Unfortunately, our negative-edge bisection rule does not guarantee that $\delta(\mathcal{P}) \to 0$, and we found that Algorithm 2 with the rule fails to terminate for some instances. To improve the termination behavior, we insert the longest-edge bisection steps periodically during performing the negative-edge bisection refinement. Note that recently Dickinson [12] showed that this strategy is not sufficient to ensure $\delta(\mathcal{P}) \to 0$.

We implemented Algorithm 2 with the negative-edge bisection steps and tested it for checking copositivity of the matrix $B_{3.4}$ of the graph G_8 and the matrix $B_{5.1}$ of the graph G_{12} for each of which the number of iteration of Algorithm 2 has a pronounced jump (see Tables 1 and 2).

Table 6 shows the performance of adding the negative-edge bisection steps for these two instances. The first row of the table represents the number of inserted longest-edge bisection steps (LEB steps) per twelve negative-edge bisection steps (NEB steps) where ∞ means that only longest-edge bisection steps and no negative-edge bisection step have been taken.

Note that at each iteration of Algorithm 2, we may have two optimal solutions of $(\mathrm{LP})_{P,\Lambda}$, i.e., the one obtained at Line 7 by detecting $V_\Delta^T AV_\Delta \in \widehat{\mathcal{G}_n^s}$ and the one obtained at Line 10 by detecting $V_\Delta^T AV_\Delta \in \mathcal{G}_n^s$. The third column of Table 6 shows which optimal solution is used for the negative-edge bisection steps.

For each case, the number of iterations shows the average number of iterations required after three-times execution since we have randomly chosen the negative-edge $\{v_i, v_j\}$ to be partitioned if there are multiple candidates.

We observe from Table 6 that the negative-edge bisection strategy using the solution obtained at Line 10 (by detecting $V_\Delta^T A V_\Delta \in \mathcal{G}_n^s$) has a positive effect to reduce the number of iterations for checking $B_{5.1}$ of G_{12} and that the effect is monotonically increasing with NEB frequency.

6 Concluding remarks

In this paper, we proposed a new branch and bound type algorithm for testing copositivity of a given symmetric matrix based on the algorithm proposed in [27]. Two features of our algorithm are

1. we have introduced new classes of matrices $\mathcal{G}_n^s$ and $\widehat{\mathcal{G}_n^s}$ which are relatively large subsets of $\mathcal{COP}_n$ and work well to check copositivity of a given matrix $A \in \mathcal{S}_n$ (see Theorem 3.2) , and
2. for incorporating the sets $\mathcal{G}_n^s$ or $\widehat{\mathcal{G}_n^s}$ in checking copositivity, we only have to solve a linear optimization problem with $n+1$ variables and $O(n^2)$ constraints after computing a singular value matrix decomposition, which implies that our algorithm is not so time-consuming.

Our algorithm determined the copositivity of some instances within a small number of iterations if they are strictly copositive. We also provided the negative-edge bisection strategy which aims to improve refinement of simplices for checking copositivity of the matrix.

Further research will include

- further improvement in checking $A \in \mathcal{G}_n^s$: We only solve the problem $(\mathrm{LP})_{P,\Lambda}$ for a specific P to check $A \in \mathcal{G}_n^s$. This is sufficient if A has n different eigenvalues. However, otherwise (i.e., an eigenspace of A has at least dimension 2), we may miss the fact $A \in \mathcal{G}_n^s$. Solving the problem $(\mathrm{LP})_{P,\Lambda}$ with other possible Ps might be effective for further improvement in checking $A \in \mathcal{G}_n^s$, or more specifically the results at around $\gamma = 5$ in Table 2.
- more observations on the sets $\mathcal{G}_n^s$, $\mathcal{G}_n^a$ and $\widehat{\mathcal{G}_n^s}$. Theorem 3.2 and Examples 3.3, 3.4 and 3.5 show the relationships among these sets and $\mathcal{S}_n^+ \cup \mathcal{N}_n$, $\mathrm{com}(\mathcal{S}_n^+ + \mathcal{N}_n)$, $\mathcal{S}_n^+ + \mathcal{N}_n$ and $\mathcal{COP}_n$. To observe how those sets are different and what properties they have will be of research interest in the future.

Note that Table 1 shows an interesting result concerning the second point. Let us see the result at $\gamma = 4.0$ of Algorithm 1.2. The multiple number of iterations at $\gamma = 4.0$ implies that we could not find $B_{4.0} \in \mathcal{G}_n^s$ at the first iteration for a certain orthonormal matrix P satisfying (8). Recall that the matrix B_γ is given by (17). It follows from the fact $E - A_G \in \mathcal{N}_n \subseteq \mathcal{G}_n^s$ and from the result at $\gamma = 3.5$ in Table 1 that

$$0.5(E - A_G) \in \mathcal{G}_n^s \text{ and } B_{3,5} = 3.5(E - A_G) - E \in \mathcal{G}_n^s.$$

Thus the fact that we could not find whether the matrix

$$B_{4.0} = 4.0(E - A_G) - E = 0.5(E - A_G) + B_{3.5}$$

lies on the set $\mathcal{G}_n^s$ might suggest that the set $\mathcal{G}_n^s = \mathrm{com}(\mathcal{S}_n + \mathcal{N}_n)$ is not convex.

Acknowledgment

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 brought us the idea of improving Algorithm 1.2 using the set $\widehat{\mathcal{G}}_n^s$. The authors greatly appreciate his/her constructive criticism and suggestions on improving this work. They are also grateful to Yasuaki Matsukawa for his careful reading of an earlier version of the manuscript.

References

[1] A. Berman, *Cones, Matrices and Mathematical Programming*, Lecture Notes in Economics and Mathematical Systems 79, Springer Verlag 1973.

[2] A. Berman and N. S. Monderer, *Completely Positive Matrices*, World Scientific Publishing, 2003.

[3] I. M. Bomze, M. Dür, E. De Klerk, C. Roos, A. J. Quist and T. Terlaky, On copositive programming and standard quadratic optimization problems, *Journal of Global Optimization* **18**(2000) 301-320.

[4] I. M. Bomze and E. De Klerk, Solving standard quadratic optimization problems via linear, semidefinite and copositive programming, *Journal of Global Optimization* **24**(2002) 163-185.

[5] I. M. Bomze, Copositive optimization - recent developments and applications, *European Journal of Operational Research* **216**(2012) 509-520.

[6] I. M. Bomze and G. Eichfelder Copositivity detection by difference-of-convex decomposition and ω-subdivision, *Mathematical Programming, Series A* **138**(2013) 365-400.

[7] S. Bundfuss and M. Dür, Algorithmic copositivity detection by simplicial partition, *Linear Algebra and its Applications* **428**(2008) 1511-1523.

[8] S. Bundfuss, Copositive matrices, copositive programming, and applications, Ph.D. Dissertation, TU Darmstadt, 2009.
Online at `http://www3.mathematik.tu-darmstadt.de/index.php?id=483`

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

[10] E. de Klerk, D. V. Pasechnik, Approximation of the stability number of a graph via copositive programming, *SIAM Journal on Optimization* **12**(2002) 875-892.

[11] P. H. Diananda, On non-negative forms in real variables some or all of which are non-negative, *Mathematical Proceedings of the Cambridge Philosophical Society* **58**(1962) 17-25.

[12] P. J. C. Dickinson, On the exhaustivity of simplicial partitioning, *Journal of Global Optimization* **58**(2014) 189-203.

[13] P. J. C. 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) 403-415.

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

[15] W. Fenchel, *Convex cones, sets and functions*, mimeographed notes by D. W. Blackett, Princeton Univ. Press, Princeton, N. J., 1953.

[16] M. Fiedler and V. Pták, On matrices with non-positive off-diagonal elements and positive principal minors, *Czechoslovak Mathematical Journal* **12**(1962) 382-400.

[17] G. H. Golub and C. F. Van Loan. *Matrix Computations*, Johns Hopkins University Press; Third edition, 1996.

[18] R. Horst, On generalized bisection of n-simplices, *Mathematics of Computation* **66**(1997) 691-698.

[19] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[20] F. Jarre and K. Schmallowsky, On the computation of $\mathcal{C}^*$ certificates, *Journal of Global Optimization* **45**(2009) 281-296.

[21] Y. Matsukawa and A. Yoshise, A primal barrier function Phase I algorithm for nonsymmetric conic optimization problems, *Japan Journal of Industrial and Applied Mathematics* **29**(2012) 499-517.

[22] T. S. Motzkin, E. G. Straus, Maxima for graphs and a new proof of a theorem of Turan, *Canadian Journal of Mathematics* **17**(1965) 533-540.

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

[24] J. Povh and F. Rendl, A copositive programming approach to graph partitioning, *SIAM Journal on Optimization* **18**(2007) 223-241.

[25] J. Povh and F. Rendl, Copositive and semidefinite relaxations of the quadratic assignment problem, *Discrete Optimization* **16**(2009) 231-241. .

[26] R.T. Rockafellar and R.J-B. Wets. *Variational Analysis*. Springer, 1998.

[27] J. Sponsel, S. Bundfuss and M. Dür, An improved algorithm to test copositivity, *Journal of Global Optimization* **52**(2012) 537-551.

[28] J. F. Sturm and S. Zhang. On cones of nonnegative quadratic functions. *Mathematics of Operations Research*, **28**(2003) 246-267.

[29] Y. Yoshise, Y. Matsukawa, On optimization over the doubly nonnegative cone, *Proceedings of 2010 IEEE Multi-conference on Systems and Control* (2010) 13-19.

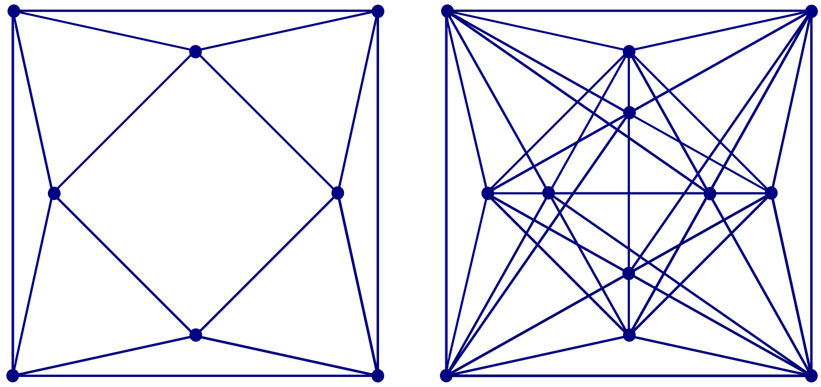

Figure 1: The graphs G_8 with $\omega(G_8) = 3$ (left) and G_{12} with $\omega(G_{12}) = 4$ (right).

Table 1: Results for the graph G_8

γ	Alg. 1.1. ($\mathcal{H}_n$)		Alg. 1.2. ($\mathcal{G}_n^s$)		Alg. 2. ($\mathcal{G}_n^s$ and $\widehat{\mathcal{G}_n^s}$)	
	# of iterations	CPU time (s)	# of iterations	CPU time (s)	# of iterations	CPU time (s)
2.8	2247	0.368	2599	35.741	2478	65.160
2.9	1609	0.243	2273	31.714	2190	55.401
3.0	-	-	-	-	-	-
3.1	3003	0.324	11227	137.453	6326	129.108
3.2	1509	0.152	6243	78.627	3273	69.279
3.3	469	0.047	4365	55.308	1585	33.204
3.4	395	0.039	3295	42.249	1491	33.043
3.5	369	0.037	1	0.012	1	0.013
3.6	209	0.021	1	0.011	1	0.012
3.7	115	0.012	1	0.012	1	0.012
3.8	79	0.009	1	0.011	1	0.012
3.9	63	0.007	1	0.011	1	0.011
4.0	39	0.021	385	4.769	339	7.366
4.1	23	0.002	1	0.011	1	0.014
4.2	17	0.002	1	0.011	1	0.011
4.3	17	0.002	1	0.011	1	0.011
4.4	7	0.001	1	0.011	1	0.011
4.5	7	0.001	1	0.012	1	0.011

Table 2: Results for G_{12}

γ	Alg. 1.1. ($\mathcal{H}_n$)		Alg. 1.2. ($\mathcal{G}_n^s$)		Alg. 2. ($\mathcal{G}_n^s$ and $\widehat{\mathcal{G}_n^s}$)	
	# of iterations	CPU time (s)	# of iterations	CPU time (s)	# of iterations	CPU time (s)
3.8	4084	1.753	4088	134.426	4088	253.307
3.9	4080	1.755	4088	136.621	4088	269.668
4.0	-	-	-	-	-	-
4.1	-	-	-	-	-	-
4.2	-	-	-	-	-	-
4.3	-	-	-	-	-	-
4.4	-	-	-	-	-	-
4.5	1125035	14789.042	-	-	-	-
4.6	762931	8166.234	-	-	-	-
4.7	610071	6121.059	-	-	-	-
4.8	569661	5375.867	-	-	-	-
4.9	407201	3225.592	-	-	219156	11135.302
5.0	305521	1693.459	-	-	141907	6882.747
5.1	206949	611.519	-	-	59063	2848.623
5.2	141383	262.289	1	0.052	1	0.053
5.3	110641	154.699	1	0.043	1	0.061
5.4	90877	102.006	1	0.056	1	0.052
5.5	44731	22.361	1	0.043	1	0.059
5.6	26171	8.353	1	0.052	1	0.056
5.7	15045	3.593	1	0.055	1	0.058
5.8	10239	2.167	1	0.058	1	0.061
5.9	6977	1.325	1	0.063	1	0.057
6	4717	0.839	1	0.053	1	0.064

Table 3: The number of simplices removed at each line of Algorithm 2 for G_8

γ	Line 8	Line 11	Line 19
2.8	169	175	5
2.9	179	220	3
3.0	-	-	-
3.1	2150	1000	27
3.2	1003	586	96
3.3	515	235	86
3.4	359	325	124

Table 4: The number of simplices removed at each line of Algorithm 2 for G_{12}

γ	Line 8	Line 11	Line 19
3.8	0	19	0
3.9	0	21	0
4.0	-	-	-
4.8	-	-	-
4.9	89561	18546	2943
5.0	56157	12744	4106
5.1	24983	4549	0

Table 5: Results for modified programs provided in Theorem 4.1

Graph	γ	ρ	Alg. 1.1. ($\mathcal{H}_n$)		Alg. 1.2. ($\mathcal{G}_n^s$)		Alg. 2. ($\mathcal{G}_n^s + \widehat{\mathcal{G}_n^s}$)	
			# of iterations	CPU time (s)	# of iterations	CPU time (s)	# of iterations	CPU time (s)
G_8	4	$0.199 < 1/5$	3	0.015	1	0.055	1	0.084
G_8	3	$0.249 < 1/4$	47	0.021	1	0.056	1	0.087
G_{12}	5	$0.166 < 1/6$	4771	0.166	1	0.068	1	0.109
G_{12}	4	$0.199 < 1/5$	337997	1829.384	997653	41301.295	168080	8580.701

Table 6: Results of Algorithm 2 with improved strategies for refinement of simplices

Graph	γ	$(\text{LP})_{P,\Lambda}$	$\infty/12$		6/12		4/12		3/12	
			# of iterations	CPU time (s)	# of iterations	CPU time (s)	# of iterations	CPU time (s)	# of iterations	CPU time (s)
G_8	3.4	Line 7	1491	33	3185	66	6850	137	7367	158
G_8	3.4	Line 10	1491	33	1545	33	1284	27	1595	34
G_{12}	5.1	Line 7	59063	2849	44583	2111	125443	6035	130456	6403
G_{12}	5.1	Line 10	59063	2849	43499	2063	36973	1755	33945	1612