A modified simplex partition algorithm to test copositivity

Mohammadreza Safi[1] · Seyed Saeed Nabavi[2] · Richard J. Caron[3]

Abstract

A real symmetric matrix A is copositive if $x^\top A x \geq 0$ for all $x \geq 0$. As A is copositive if and only if it is copositive on the standard simplex, algorithms to determine copositivity, such as those in Sponsel et al. (J Glob Optim 52:537–551, 2012) and Tanaka and Yoshise (Pac J Optim 11:101–120, 2015), are based upon the creation of increasingly fine simplicial partitions of simplices, testing for copositivity on each. We present a variant that decomposes a simplex $\triangle$, say with n vertices, into a simplex $\triangle_1$ and a polyhedron Ω_1; and then partitions Ω_1 into a set of at most $(n-1)$ simplices. We show that if A is copositive on Ω_1 then A is copositive on $\triangle_1$, allowing us to remove $\triangle_1$ from further consideration. Numerical results from examples that arise from the maximum clique problem show a significant reduction in the time needed to establish copositivity of matrices.

Keywords Copositive matrix · Maximum clique problem · Simplex · Semidefinite programming · Copositive programming

Mathematics Subject Classification MSC 65K05 · MSC 90C22 · MSC 90C99

1 Introduction

Let A be a real symmetric matrix of order n. If $x^\top A x \geq 0$ for all $x \in \mathbb{R}^n_+ = \{x \in \mathbb{R}^n \mid x \geq 0\}$ then A is called copositive [7]. If $x^\top A x > 0$ for all $x \in \mathbb{R}^n_+ \setminus \{0\}$ then A is strictly copositive. The set of all order n copositive matrices $\mathcal{COP}$ is a closed, convex, and full-dimensional pointed cone whose interior is the set of strictly copositive matrices.

✉ Richard J. Caron
  rcaron@uwindsor.ca

  Mohammadreza Safi
  safim@uwindsor.ca

  Seyed Saeed Nabavi
  saeednabavi86@gmail.com

[1] Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada

[2] Faculty of Mathematics, Statistics and Computer Science, Semnan University, Semnan, Iran

[3] Department of Mathematics and Statistics, University of Windsor, Windsor, ON, Canada

Methods to determine copositivity are important because of their role in algorithms for the solution of copositive optimization problems [2,5], a class of problems that include, for example, the maximum clique problem [1]. The challenge is that Murty and Kabadi [8] have shown that given a square integer matrix A the decision problem "is A not copositive?" is NP-complete.

This algorithm has the same branch-and-bound design as Bundfuss and Dür's [4] BD algorithm, which was improved by Sponsel et al. [9], and then generalized by Tanaka and Yoshise's [10] TY algorithm. All four methods depend on partitions of the standard simplex. Ours is uniquely different in that we decompose a simplex $\triangle$ into a simplex $\triangle_1$ and a polyhedron Ω_1, and then partition Ω_1 into a set of at most $(n-1)$ simplices, each of which is subsequently decomposed into a simplex and a polyhedron, with the process continuing until copositivity is determined. We will show that if A is copositive on Ω_1, then A is copositive on $\triangle_1$, allowing us to remove $\triangle_1$ from further consideration. As our selection of $\triangle_1$ is done in a maximal way, the number of required partitions is, hopefully, significantly reduced.

Similar to the algorithms in [4,9,10], our new algorithm determines copositivity of A by checking whether $M_\triangle^\top A M_\triangle \in \mathcal{M}$, where $M_\triangle$ is a matrix whose columns are the vertices of $\triangle$, and where $\mathcal{M} \subset \mathcal{COP}$. In [4], $\mathcal{M} = \mathcal{N}$, the set of all nonnegative $n \times n$ matrices. Sponsel et al. [9] comment that while checking $M_\triangle^\top A M_\triangle \in \mathcal{N}$ is relatively easy, the set $\mathcal{N}$ is too small and they suggest sets $\mathcal{M} \subset \mathcal{COP}$ with $\mathcal{M} \supset \mathcal{N}$. While this results in fewer iterations (simplices) the iterations are more time consuming as they require either the calculation of eigenvectors or the solution of semidefinite programs. They reported that in all instances, the BD algorithm with $\mathcal{M} = \mathcal{N}$ was fastest. It is to the BD algorithm that we compare the algorithm introduced in this paper where we also use $\mathcal{M} = \mathcal{N}$.

Tanaka and Yoshise's [10] generalization is based on more sophisticated selections for $\mathcal{M}$. They consider $\mathcal{M} = \mathcal{N} + \mathcal{S}^+$, where $\mathcal{S}^+$ is the set of all positive semidefinite matrices. The consequence is that a linear program, rather than a semidefinite program, is to be solved to determine whether $M_\triangle^\top A M_\triangle \in \mathcal{M}$. However, the formulation of the linear program requires a singular value decomposition, and the decomposition is not unique even though the result depends on that decomposition. Tanaka and Yoshise suggest that the TY algorithm is promising for the determination of upper bounds for the maximum clique problem.

Our limited numerical experiments show that for the instances examined there is a significant reduction in the time needed to establish copositivity of matrices that arise in the determination of the clique number of an undirected graph. Further, the experimental results show that for the instances examined there is an advantage to our algorithm when A is near the copositive cone boundary.

2 Simplicial partition algorithms

We begin with the standard simplex

$$\triangle^s = \left\{ x \in \mathbb{R}^n_+ \ | \ \|x\|_1 = 1 \right\},$$

where $\| \cdot \|_1$ is the 1-norm. The matrix A is copositive if and only if it is copositive on $\triangle^s$, that is, $x^\top A x \geq 0$ for all $x \in \triangle^s$. Let P be a simplicial partition of $\triangle^s$ as defined in [4]. That is, $\triangle^s$ is the union of all simplices in P; and the intersection of the interior of any two simplices in P is empty. For any $\triangle \in P$ denote by $V_\triangle$ the vertices $v_1, \ldots, v_n$ of $\triangle$. Thus, the columns of $M_\triangle$ are the elements of $V_\triangle$. Lemma 1, which follows from the work in [9], gives conditions for copositivity. In fact, it shows that if $M_\triangle^\top A M_\triangle \in \mathcal{N}$, then A is copositive on

$\triangle \in P$. Lemma 2, which follows from the work in [4], gives a condition for non-copositivity; and, if non-copositivity is not determined, it establishes the existence of a vertex $v_\ell \in V_\triangle$ with $v_\ell^\top A v_\ell > 0$, needed for the new partitioning process presented in Sect. 3.

Lemma 1 *If $M_\triangle^\top A M_\triangle \geq 0$, i.e., $M_\triangle^\top A M_\triangle \in \mathcal{N}$, then A is copositive on $\triangle \in P$. If A is copositive on all $\triangle \in P$ then A is copositive on $\triangle^s$, and is therefore copositive.*

Lemma 2 *For $\triangle \in P$ suppose that $M_\triangle^\top A M_\triangle \ngeq 0$ and that $v^\top A v \geq 0$ for all $v \in V_\triangle$. Either there exist v_i, $v_j \in V_\triangle$, $v_i \neq v_j$, with $v_i^\top A v_j < 0$ and $v_i^\top A v_i = v_j^\top A v_j = 0$ so that A is not copositive; or there exists a $v_\ell \in V_\triangle$ with $v_\ell^\top A v_\ell > 0$.*

Theorem 1, which is Lemma 2.3 in [9], gives a criterion for termination. Suppose that it has been determined that A is copositive on all $\triangle \in \tilde{P} \subset P$. Let $\hat{P} = P \setminus \tilde{P}$ and define the diameter of $\hat{P}$ by

$$\delta(\hat{P}) = \max_{\triangle \in \hat{P}} \left\{ \max_{u,v \in V_\triangle} \|u - v\|_2 \right\}.$$

Theorem 1 *If the real symmetric matrix A is not copositive, then there exists an $\epsilon > 0$ such that if $\delta(\hat{P}) < \epsilon$ there exists a $\triangle \in \hat{P}$ with $v^\top A v < 0$ for some $v \in V_\triangle$.*

The BD and TY simplicial partitioning algorithms are *branch and bound* algorithms that proceed as follows. Start with a partition P of $\triangle^s$. This is the first *branch* step. If Lemma 2 determines that A is not copositive then the algorithm terminates.

If Lemma 1 determines that A is copositive on $\triangle \in P$, then $\triangle$ can be removed from further consideration. That is, we do not need to create a partition of $\triangle$. This is the *bound* step.

The *branch* step is to find simplicial partitions for each of the simplices in the original partition that have not been removed in the bound step. That is, the partitions are further refined. The process continues until it is either determined that A is not copositive ($v^\top A v < 0$ for some simplex vertex v), that A is copositive on all remaining simplices, in which case A is copositive, or until the remaining simplices are sufficiently small so that the copositivity of A can be determined by the vertices of the remaining simplices.

3 A new method of partitions

Our new method differs in how the simplicial partitions are created. Consider a simplex $\hat{\triangle} \in P$. If $M_{\hat{\triangle}}^\top A M_{\hat{\triangle}} \geq 0$ then, by Lemma 1, A is copositive on $\hat{\triangle}$ and we can then consider copositivity on the next $\triangle \in P$. Let $D(M_\triangle)$ be the vector of diagonal elements of $M_\triangle^\top A M_\triangle$. If $D(M_\triangle)$ has at least one negative element, then A is not copositive and we are done. If $M_\triangle^\top A M_\triangle \ngeq 0$ and if $D(M_\triangle) \geq 0$, then $M_\triangle^\top A M_\triangle$ has a negative off-diagonal, say the entry in row i and column j. If both the i-th and j-th diagonal entries are zero, then, by Lemma 2, A is not copositive and we are done. Otherwise, there is at least one strictly positive element in $D(M_\triangle)$, say, $v_\ell^\top A v_\ell$, that we will use to create a simplex $\triangle_1$ which can be sliced off $\triangle$ and eliminated from further consideration.

Theorem 2 *If $D(M_\triangle) \geq 0$ and if there exists a $v_\ell \in V_\triangle$ with $v_\ell^\top A v_\ell > 0$, then for each $j \neq \ell$ there exists a scalar $\lambda_j \in (0, 1]$, with $\hat{v}_j^\top A v_\ell \geq 0$ where $\hat{v}_j = (1 - \lambda_j) v_\ell + \lambda_j v_j$.*
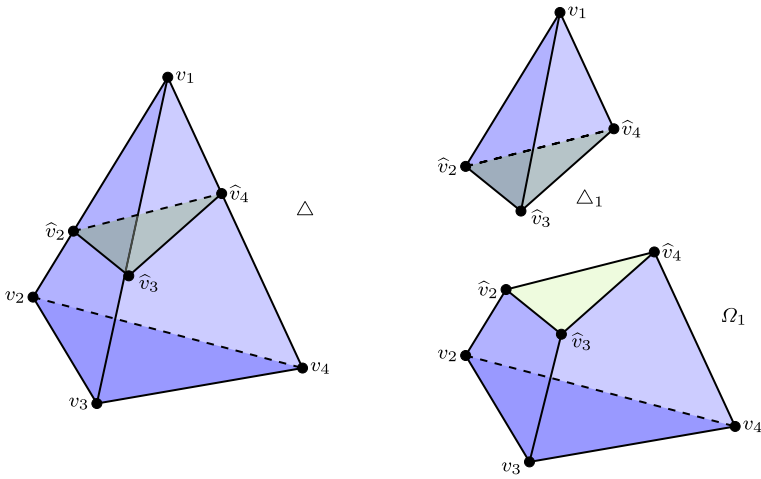
Fig. 1 The slicing and partitioning of $\triangle$ into $\triangle_1$ and Ω_1

Proof If $v_j^\top A v_\ell \geq 0$ then $\hat{v}_j^\top A v_\ell = (1 - \lambda_j) v_\ell^\top A v_\ell + \lambda_j v_j^\top A v_\ell \geq 0$ for all $\lambda_j \in (0, 1]$. Otherwise, $v_j^\top A v_\ell < 0$ and we have that

$$\hat{v}_j^\top A v_\ell \geq 0 \quad \Longleftrightarrow \quad \lambda_j \leq v_\ell^\top A v_\ell / (v_\ell^\top A v_\ell - v_j^\top A v_\ell). \tag{1}$$

$\square$

Let $\triangle_1$ be the simplex with vertices v_ℓ and $\hat{v}_j$, $1 \leq j \leq n$, $j \neq \ell$, where the $\hat{v}_j$ are as described in Theorem 2 using the largest value possible for λ_j. We have the partition

$$\triangle = \triangle_1 \cup \Omega_1, \tag{2}$$

where

$$\Omega_1 = \text{conv}((V_{\triangle_1} \cup V_\triangle) \setminus \{v_\ell\}), \tag{3}$$

the convex hull of the points $(V_{\triangle_1} \cup V_\triangle) \setminus \{v_\ell\}$. In Fig. 1 we see a slice of a simplex $\triangle$ in $\mathbb{R}^4$ and the partition into $\triangle_1$ and Ω_1 that it creates. In this example, $\ell = 1$.

Theorem 3 shows that we can remove $\triangle_1$ from further consideration as copositivity on Ω_1 implies copositivity on $\triangle_1$.
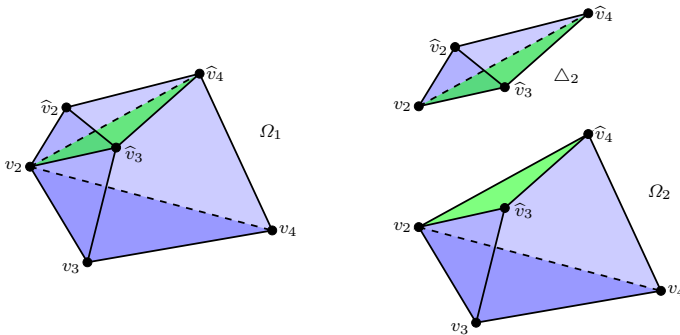
Theorem 3 *If A is copositive on Ω_1, then A is copositive on $\triangle_1$.*

Proof For $x = v_\ell \in \triangle_1$, where v_ℓ is as described in Theorem 2, we have $x^\top A x > 0$. For every $x \in \triangle_1 \setminus v_\ell$ there exist scalars $\alpha_j \geq 0$, $j = 1, \ldots, n$, summing to unity, such that

$$x = \alpha_\ell v_\ell + \sum_{\substack{j=1 \\ j \neq \ell}}^{n} \alpha_j \hat{v}_j = \alpha_\ell v_\ell + (1 - \alpha_\ell) \sum_{\substack{j=1 \\ j \neq \ell}}^{n} \frac{\alpha_j}{(1 - \alpha_\ell)} \hat{v}_j = \alpha_\ell v_\ell + (1 - \alpha_\ell) \hat{v},$$

where

$$\hat{v} = \sum_{\substack{j=1 \\ j \neq \ell}}^{n} \frac{\alpha_j}{(1 - \alpha_\ell)} \hat{v}_j.$$

Fig. 2 Decomposition of Ω_1 into $\triangle_2$ and Ω_2

In this case, $x \neq v_\ell$ so that $\alpha_\ell \neq 1$. Since the scalars $\alpha_j/(1 - \alpha_\ell)$ are nonnegative and sum to unity it follows that $\hat{v} \in \Omega_1$. By assumption, A is copositive on Ω_1 so that $\hat{v}^\top A \hat{v} \geq 0$. Since $v_\ell^\top A \hat{v}_j \geq 0$ for all $j \neq \ell$, we have $v_\ell^\top A \hat{v} \geq 0$. Thus,

$$x^\top A x = (\alpha_\ell v_\ell + (1 - \alpha_\ell)\hat{v})^\top A(\alpha_\ell v_\ell + (1 - \alpha_\ell)\hat{v}) \geq 0$$

and A is copositive on $\triangle_1$.   □

If Ω_1 has n vertices then it is a simplex from which we either determine copositivity or partition Ω_1 into a simplex and a polyhedron as above. If Ω_1 has more than n vertices we create the partition $\Omega_1 = \triangle_2 \cup \Omega_2$ as follows.

Let $J^\ell = \{1, 2, \ldots, n\}\backslash\{\ell\}$. Set $J_1 = \{j \in J^\ell \,|\, \lambda_j = 1\}$ and $J_2 = J^\ell \backslash J_1$, that is $J_2 = \{j \in J^\ell \,|\, 0 < \lambda_j < 1\}$. For $j \in J_1$ we have $\hat{v}_j = v_j$ and for $j \in J_2$ we have $\hat{v}_j = (1 - \lambda_j)v_\ell + \lambda_j v_j$. We then choose any index $\mu \in J_2$. In Fig. 2 we begin with the Ω_1 from Fig. 1. We have $J_1 = \emptyset$, $J_2 = \{2, 3, 4\}$ and we selected $\mu = 2$ giving the simplex $\triangle_2$ with vertices v_2, $\hat{v}_2$, $\hat{v}_3$ and $\hat{v}_4$. Lemma 3 shows that these vectors are linearly independent.

**Lemma 3** *The vectors $\hat{v}_j$, $1 \leq j \leq n$, $j \neq \ell$ together with v_μ are linearly independent.*

Proof Consider

$$\alpha_\mu v_\mu + \sum_{j \in J_1} \alpha_j \hat{v}_j + \sum_{j \in J_2} \alpha_j \hat{v}_j = 0$$

$$\iff \quad \alpha_\mu v_\mu + \sum_{j \in J_1} \alpha_j v_j + \sum_{j \in J_2} \alpha_j((1 - \lambda_j)v_\ell + \lambda_j v_j) = 0$$

$$\iff (\alpha_\mu + \alpha_\mu \lambda_\mu)v_\mu + \sum_{j \in J_1} \alpha_j v_j + (\sum_{j \in J_2} \alpha_j(1 - \lambda_j))v_\ell + \sum_{j \in J_2\backslash\mu} \alpha_j \lambda_j v_j = 0.$$

Since the vectors in $V_\triangle$ are linearly independent, and since $0 < \lambda_j < 1$, for all $j \in J_2$, we can conclude that $\alpha_j = 0$, for all $j \neq \ell$ and that $\alpha_\mu = 0$.   □

By Lemma 3 we conclude that the convex hull of the vectors $\hat{v}_j$, $1 \leq j \leq n$, $j \neq \ell$ together with v_μ is a simplex which we denote by $\triangle_2$. We denote by Ω_2 the convex hull of the set of vectors $V_{\Omega_1} \backslash \{\hat{v}_\mu\}$, which is a polyhedron with one vertex less than Ω_1. The proposed algorithm will then slice a simplex off Ω_2 creating $\triangle_3$ and Ω_3, with Ω_3 having one less vertex than Ω_2, continuing in this way until the Ω polyhedron is itself a simplex. The set of simplices created by this process is denoted by $SL(\triangle)$, where SL represents the slicing process. Since $\triangle_1$ need not be considered further, it is not included in $SL(\triangle)$. The

Fig. 3 The slicing procedure
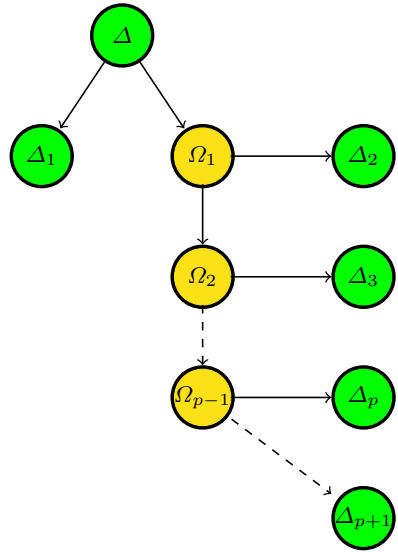
algorithm would continue by restarting the process for each simplex in $SL(\triangle)$. In Fig. 3 we see a graphic of this process with the $p \leq n$ simplices, labeled $\triangle_2$ to $\triangle_{p+1}$, in $SL(\triangle)$ down the right-hand side.

This, of course, depends on Ω_2 and $\triangle_2$ being a partition of Ω_1. This is shown in Lemma 4 by showing that the hyperplane passing through $v_\mu, \hat{v}_j, \forall j \neq \mu, \ell$ is a separating hyperplane for $\triangle_2$ and Ω_2. Denote the hyperplane by H and represent it by $w^\top x = b$.

Without loss of generality, assume that $w^\top \hat{v}_\mu < b$. Since $v_\mu, \hat{v}_j, \forall j \neq \mu, \ell$ are on H, and since $\hat{v}_\mu, v_\mu, \hat{v}_j, \forall j \neq \mu, \ell$ are linearly independent, it follows that $w^\top \hat{v}_\mu \neq b$. If $w^\top \hat{v}_\mu > b$, replace w with $-w$.

Lemma 4 *For all $x \in \triangle_2 \setminus H$, we have $w^\top x < b$ and for all $x \in \Omega_2 \setminus H$, we have $w^\top x > b$.*

Proof By assumption we have $w^\top x < b$ when $x = \hat{v}_\mu$. For $x \in \triangle_2 \setminus H$, $x \neq \hat{v}_\mu$, we can write x as the convex combination

$$x = t v_\mu + t_\mu \hat{v}_\mu + \sum_{\substack{j=1 \\ j \notin \{\ell, \mu\}}}^{n} t_j \hat{v}_j$$

with $t_\mu > 0$ (since $x \notin H$.) Thus

$$w^\top x - b = w^\top \left(t v_\mu + t_\mu \hat{v}_\mu + \sum_{\substack{j=1 \\ j \notin \{\ell, \mu\}}}^{n} t_j \hat{v}_j \right) - b = t(w^\top v_\mu) + t_\mu(w^\top \hat{v}_\mu)$$

$$+ \sum_{\substack{j=1 \\ j \notin \{\ell, \mu\}}}^{n} t_j (w^\top \hat{v}_j) - b.$$

Since $w^\top \hat{v}_j = b$ for all $j \neq \ell$, which includes $j = \mu$, we have

$$w^\top x - b = t(w^\top v_\mu) + \sum_{\substack{j=1 \\ j \neq \ell}}^n (t_j b) - b = t(w^\top v_\mu) + b\left(\sum_{\substack{j=1 \\ j \neq \ell}}^n t_j - 1\right).$$

As we have a convex combination it follows that

$$\sum_{\substack{j=1 \\ j \neq \ell}}^n t_j - 1 = -t$$

giving

$$w^\top x - b = t(w^\top v_\mu - b) < 0$$

since $t \geq 0$ and $w^\top v_\mu < b$ (by assumption). That $w^\top x > b$ for all $x \in \Omega_2 \setminus H$ is proved in a similar manner. □

4 The algorithm

We now present the algorithm. We use P to represent the set of simplices that have not yet been fathomed. For any $\triangle \in P$ the center and width of $\triangle$ are given by

$$\bar{v} = (1/n) \sum_{i=1}^n v_i \text{ and } \delta = \max_{i=1,\dots,n} ||v_i - \bar{v}||_1, \tag{4}$$

respectively. Let

$$\bar{\delta} = -1 + \sqrt{1 + \bar{v}^\top A \bar{v} / ||A||_1}, \tag{5}$$

where $||A||_1$ is the element-wise norm that is the sum of the absolute values of the entries in A.

Lemma 5 *Let $\triangle \in P$, and let $\bar{v}$, δ and $\bar{\delta}$ be as given in (4) and (5). If $\bar{v}^\top A \bar{v} > 0$ and $\delta \leq \bar{\delta}$, then A is copositive on $\triangle$.*

Proof Let $x \in \triangle$ and consider

$$\begin{aligned}
|x^\top A x - \bar{v}^\top A \bar{v}| &= |2\bar{v}^\top A(x - \bar{v}) + (x - \bar{v})^\top A(x - \bar{v})| \\
&\leq 2|\bar{v}^\top A(x - \bar{v})| + |(x - \bar{v})^\top A(x - \bar{v})| \\
&\leq 2||\bar{v}||_1 \, ||A||_1 \, ||x - \bar{v}||_1 + ||A||_1 \, ||x - \bar{v}||_1^2 \\
&\leq ||A||_1 \delta^2 + 2||A||_1 \delta,
\end{aligned} \tag{6}$$

where the last inequality follows since $\bar{v} \in \triangle$ implies that $||\bar{v}||_1 = 1$ and $x \in \triangle$ implies that $||x - \bar{v}||_1 \leq \delta$. The positive root of $||A||_1 \delta^2 + 2||A||_1 \delta - \bar{v}^\top A \bar{v}$ is $\bar{\delta}$ as given in (5). It follows that for $0 \leq \delta \leq \bar{\delta}$,

$$||A||_1 \delta^2 + 2||A||_1 \delta \leq \bar{v}^\top A \bar{v}. \tag{7}$$

Combining (6) and (7) we have that for $0 \leq \delta \leq \bar{\delta}$,

$$|x^\top A x - \bar{v}^\top A \bar{v}| \leq \bar{v}^\top A \bar{v}. \tag{8}$$

Since $\bar{v}^\top A \bar{v} > 0$, this implies that $x^\top A x \geq 0$ for all $x \in \triangle$. Thus, A is copositive on $\triangle$. $\quad\square$

Algorithm 1: A Modified Simplex Partition Algorithm

Data: An $n \times n$ symmetric matrix A.
Result: The message that "A is copositive" or that "A is not copositive".

1 initialization $P \longleftarrow \{\triangle^s\}$, Flag $\longleftarrow$ False;
2 **while** *Flag = False* **do**
3   **for** $\triangle \in P$ **do**
4   **if** $\bar{V} \geq 0$ *or* $(\bar{v}^\top A \bar{v} > 0$ *and* $\delta \leq \bar{\delta})$ **then**
5          $P \longleftarrow P \setminus \{\triangle\}$
6   **else**
7   **if** $\bar{v}^\top A \bar{v} < 0$ *or* $(\exists i, j \mid \bar{V}_{ij} < 0 \land (\bar{V}_{ii} = \bar{V}_{jj} = 0))$ *or* $(\exists i \mid \bar{V}_{ii} < 0)$
         then
8   Flag $\longleftarrow$ True and **Return** "A is not copositive"
9   **else**
10   **if** $P = \emptyset$ **then**
11   Flag $\longleftarrow$ True and **Return** "A is copositive"
12   **else**
13               $P \longleftarrow P \cup SL(\triangle)$

In the statement of the algorithm, we use the notation $\bar{V} = M_\triangle^\top A M_\triangle$ and $\bar{V}_{ij} = v_i^\top A v_j$. Line 6 of the algorithm follows from the definition of copositivity ($\bar{v}^\top A \bar{v} < 0$) and from Lemma 2; and Lemma 5 validates the line 5 of the algorithm.

Intuitively, we expect $\delta(P)$, the diameter of P, to converge to zero, that is, that the sequence of partitions is exhaustive. In [6], Dickinson tells us that whether a sequence of partitions is exhaustive can "*often defy our intuition*". Fortunately, he provides two methods, that can be adapted to our method, to ensure exhaustion.

Under the assumption that $\delta(P)$ converges to zero, if A is strictly copositive or A is not copositive, then the algorithm terminates, but if A is copositive but not strictly copositive, it may or may not terminate [4]. As in [4], we could modify the algorithm by choosing a small tolerance $\epsilon > 0$ and terminate when $M_\triangle^\top A M_\triangle > -\epsilon$ for all $\triangle$. In this case, we say that A is ϵ-copositive and Theorem 6 in [4] guarantees the termination of the algorithm.

5 Examples

The first three examples come from graph theory. Let E be the matrix of ones and let A_G be the adjacency matrix of a given undirected graph G and define $B_\gamma = \gamma (E - A_G) - E$. From [1] we know that the clique number of G is

$$w(G) = \min\{\gamma \in \mathbb{N} \mid B_\gamma \text{ is copositive}\}.$$

Consider the graphs G_8, G_{10} and G_{12} shown in Fig. 4 by way of definition. The figures for G_8 and G_{12} are from [9]. We have $w(G_8) = 3$, $w(G_{10}) = 3$ and $w(G_{12}) = 4$.

We will determine the copositivity of $A = B_\gamma$ for various values of γ.

For the remaining nine examples we check copositivity of A, where A is created as follows. We choose the order n and a $\gamma > 2$. Choose $\hat{A}$ to be symmetric with zeros on the diagonal and
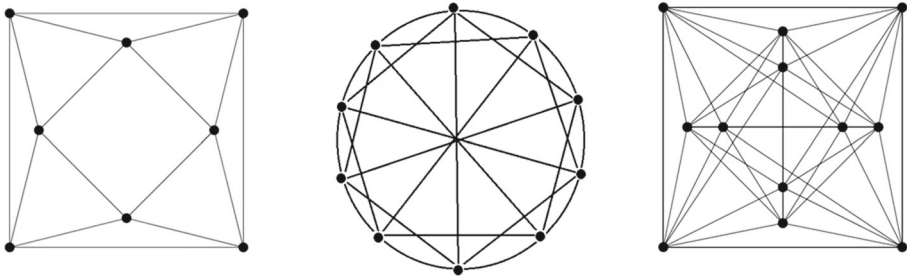
Fig. 4 The graphs G_8, G_{10} and G_{12}

with the upper off-diagonal elements to be zero or one with equal probability. We can think of A as being the adjacency matrix of a randomly generated graph. Then $A = \gamma(E - \hat{A}) - E$.

We test the proposed algorithm, called SNC, against the BD algorithm. Both algorithms were programmed by the second author and were run on a 2.70 GHz Dual Core Processor machine with 4 GB of RAM.

Table 1 shows the results for the graph G_8. We see that both algorithms correctly determine that $w(G_8) = 3$. For every value of γ, the time taken by SNC is at least an order of magnitude less than for BD. The number of simplices considered by SNC is at most half the number considered by BD. Also, when $\gamma = w(G_8)$, BD failed while SNC correctly determined copositivity. The time taken to determine non-copositivity is two orders of magnitude less for SNC and the number simplices considered an order of magnitude less.

Table 2 shows the results for the graph G_{10}. We see that both algorithms correctly determine that $w(G_{10}) = 3$. Both algorithms are able to determine copositivity when $\gamma = w(G_{10})$. The number of simplices considered by SNC is an order of magnitude less than for BD when $\gamma > 3.2$, and considerably less than BD for $\gamma \leq 3$.

Table 3 shows the results for the graph G_{12}. For $\gamma \geq 4 = w(G_{12})$ BD is unable to determine copositivity. While SNC is successful, the number of simplices considered is at least three orders of magnitude larger than for G_8 and G_{10}. For $\gamma < 4$, SNC outperforms BD.

For the three graph theory (clique number) examples, SNC outperforms BD, especially in the determination of non-copositivity.

In Table 4 we see the results for the remaining nine examples. In all cases SNC takes less time and considers fewer simplices than BD.

6 Conclusion

We have presented a novel method to decompose the standard simplex in order to determine copositivity of real symmetric matrices. Thus, every algorithm that is based on simplicial partitioning in order to test for copositivity can be modified using our method. This includes the algorithms in, for example, [3,4,9–11], and others.

At each major iteration of our algorithm we are able to remove a maximal simplex from future consideration. After removal of the simplex, we are left with a polyhedron, which is then decomposed into at most n simplices. This gives a significant reduction to the number of simplices that need to be considered, and possibly decomposed, making the algorithm run more quickly.

Table 1 Results for the graph G_8

γ	BD Algorithm		SNC Algorithm		Copositive
	time (seconds)	Number of simplices	time (seconds)	Number of simplices	
4.0	6.1560	48,517	0.7163	13,410	✓
3.4	6.2371	48,517	0.9503	22,250	✓
3.3	6.2348	48,517	0.2105	15,990	✓
3.2	44.0194	39,370	0.6187	8,946	✓
3.1	44.3307	348,281	1.2831	116,548	✓
3.0	–	–	0.0157	1,245,425	✓
2.9	6.0698	1,023	0.0591	144	✗
2.8	5.9572	1,023	0.0293	34	✗

Table 2 Results for the graph G_{10}

γ	BD Algorithm		SNC Algorithm		Copositive
	time (seconds)	Number of simplices	time (seconds)	Number of simplices	
4.0	19.2572	96,711	0.9524	3,398	✓
3.9	18.8949	96,711	0.7704	3,402	✓
3.8	18.9625	96,711	0.7255	3,402	✓
3.7	18.5922	96,711	0.9610	3,402	✓
3.7	18.5922	96,711	0.9610	3,402	✓
3.6	18.3455	96,711	0.9482	3,398	✓
3.5	19.1216	96,711	0.7047	3,402	✓
3.3	19.3064	96,711	0.6718	3,486	✓
3.2	18.7758	96,711	0.9673	3,482	✓
3.1	383.0125	1,909,691	0.7176	3,710	✓
3.0	385.8132	1,909,691	1.0867	4,323	✓
2.9	245.8222	1,230,531	0.3763	103	×
2.8	235.5141	1,232,747	0.3415	33	×

Table 3 Results for the graph G_{12}

γ	BD Algorithm		SNC Algorithm		Copositive
	time (seconds)	Number of simplices	time (seconds)	Number of simplices	
5.0	–	–	207.9534	2,772,465	✓
4.9	–	–	252.6406	3,712,683	✓
4.8	–	–	247.8836	3,683,443	✓
4.7	–	–	247.0464	3,574,883	✓
4.6	–	–	250.9389	3,712,627	✓
4.5	–	–	274.3945	41,41,250	✓
4.4	–	–	252.2970	3,835,235	✓
4.3	–	–	265.4872	4,034,400	✓
4.2	–	–	269.4826	4,053,507	✓
4.1	–	–	312.7002	3,963,596	✓
4.0	–	–	271.3328	4,099,051	✓
3.9	67.1859	262,143	1.3511	7,297	×
3.8	68.1118	262,143	0.4878	7,297	×
3.7	65.5907	249,891	0.4561	1,989	×
3.6	67.0729	249,891	0.4812	1,989	×

Table 4 Results for random graphs

n	γ	BD Algorithm		SNC Algorithm		Copositive
		time (seconds)	Number of simplices	time (seconds)	Number of simplices	
3	2.5	0.0140	7	0.0071	3	×
6	5.5	1.2569	14,681	0.0658	119	✓
7	5.5	0.18083	1,307	0.08593	288	✓
8	5.5	15.03577	116,027	10.5586	178,689	✓
9	5.5	2.849856	17,867	0.29017	2,451	✓
10	6.5	14.23175	79,469	0.348362	2,938	✓
11	5.5	94.285106	437,423	0.214739	1,262	×
12	5.5	432.3946	1,520,351	56.6735	535,746	×
15	7.5	–	–	231.38338	36,375,590	✓

Our algorithm works well with the set $\mathcal{N}$ so that we need not consider more sophisticated choices for $\mathcal{M} \subset \mathcal{COP}$ and hence avoid the need to solve a semidefinite programming problem or linear program. These claims, of fewer simplices and less time, are evidenced by the test results.

The purpose of our limited numerical testing is to validate the new method. As max-clique instances have a very particular structure they are perhaps not sufficient to fully establish a copositivity detection algorithm. Full numerical testing should include instances from the DIMACS challenge as in [4,9]. Rigorous numerical testing and application to other algorithms is the next step.

Data availability statement This paper has no associated data sets.

References

1. Bomze, I.M.: Evolution towards the maximum clique. J. Glob. Optim. **10**(2), 143–164 (1997). https://doi.org/10.1023/A:1008230200610
2. Bomze, I.M., Dür, M., de Klerk, E., Roos, C., Quist, A.J., Terlaky, T.: On copositive programming and standard quadratic optimization problems. J. Glob. Optim. **18**(4), 301–320 (2000). https://doi.org/10.1023/A:1026583532263
3. Bomze, I.M., Eichfelder, G.: Copositivity detection by difference-of-convex decomposition and ω-subdivision. Math. Program. **138**, 365–400 (2013)
4. Bundfuss, S., Dür, M.: Algorithmic copositivity detection by simplicial partition. Lin. Algebra Appl. **428**, 1511–1523 (2008)
5. Burer, S.: On the copositive representation of binary and continuous nonconvex quadratic programs. Math. Program. **120**(2), 479–495 (2009). https://doi.org/10.1007/s10107-008-0223-z
6. Dickinson, P.: On the exhaustivity of simplicial partitioning. J. Glob. Optim. **58**(1), 189–203 (2014). https://doi.org/10.1007/s10898-013-0040-7
7. Motzkin, T.: Copositive quadratic forms. National Bureau of Standards, Tech. rep. (1952)
8. Murty, K.G., Kabadi, S.N.: Some NP-complete problems in quadratic and nonlinear programming. Math. Program. **39**(2), 117–129 (1987). https://doi.org/10.1007/BF02592948
9. Sponsel, J., Bundfuss, S., Dür, M.: An improved algorithm to test copositivity. J. Glob. Optim. **52**(3), 537–551 (2012). https://doi.org/10.1007/s10898-011-9766-2
10. Tanaka, A., Yoshise, A.: An lp-based algorithm to test copositivity. Pac. J. Optim. **11**, 101–120 (2015)
11. Zilinskas, J., Dür, M.: Depth-first simplicial partition for copositivity detection, with an application to max-clique. Optim. Methods Softw. **26**(3), 499–510 (2011). https://doi.org/10.1080/10556788.2010.544310