J Glob Optim (2012) 52:537–551
DOI 10.1007/s10898-011-9766-2

An improved algorithm to test copositivity

Julia Sponsel · Stefan Bundfuss · Mirjam Dür

Received: 30 January 2011 / Accepted: 8 August 2011 / Published online: 25 August 2011

Abstract Copositivity plays a role in combinatorial and nonconvex quadratic optimization. However, testing copositivity of a given matrix is a co-NP-complete problem. We improve a previously given branch-and-bound type algorithm for testing copositivity and discuss its behavior in particular for the maximum clique problem. Numerical experiments indicate that the speedup is considerable.

Keywords Copositive matrices · Maximum clique problem · Standard quadratic program · Semidefinite programming

Mathematics Subject Classification (2000) 15A48 · 15A63 · 05C69 · 90C09 · 90C20

1 Introduction

In the last decade, copositivity of matrices has received a growing amount of interest in the optimization community. A symmetric matrix A is said to be copositive, if $x^T Ax \geq 0$ holds for all $x \geq 0$. A copositive matrix is strictly copositive if $x^T Ax = 0$ only holds for $x = 0$. The set of copositive matrices

$$\mathcal{C} = \{A \in \mathcal{S} : x^T Ax \geq 0 \text{ for all } x \geq 0\}$$

Dedicated to the memory of Reiner Horst.

J. Sponsel · M. Dür (✉)
Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen,
P.O. Box 407, 9700 AK Groningen, The Netherlands
e-mail: M.E.Dur@rug.nl

J. Sponsel
e-mail: J.K.Sponsel@rug.nl

S. Bundfuss
Uhlandstraße 14, 68542 Heddesheim, Germany
e-mail: stefan.bundfuss@gmx.de

(where $\mathcal{S}$ denotes the set of symmetric matrices) is known to be a closed, convex, full-dimensional and pointed cone whose interior is the set of strictly copositive matrices. Like the semidefinite cone it is nonpolyhedral, but its structure is more complex, as testing whether a given matrix A is in $\mathcal{C}$ is a co-NP-complete problem [23].

Numerous conditions for copositivity have been proposed, see [1,15,19,20] for recent surveys. Many of these conditions involve properties of principal submatrices, and it is hard to use those for optimization purposes. However, in [11] an algorithmic approach was proposed, which basically relies on investigating $x^T Ax$ on smaller and smaller parts of the standard simplex. The paper [7] follows somewhat related ideas. We will review and generalize the method of [11] in Sect. 2.2.

Although testing copositivity of a matrix is an interesting problem in its own, copositivity also plays an important role in optimization. A number of optimization problems can be solved through a sequence of copositivity tests. An example is the standard quadratic optimization problem (StQP):

$$\text{(StQP)} \qquad \begin{array}{l} \min\ x^T Qx \\ s.t.\ e^T x = 1 \\ \quad x \geq 0 \end{array}$$

with e denoting the all-ones vector and Q symmetric, but not necessarily positive semidefinite. This problem includes NP-hard problems like the maximum clique problem, see [5,6,13]. It is well-known ([6]) that (StQP) has the following copositive formulation

$$\max\{y : Q - yE \in \mathcal{C}\} \tag{1}$$

where $E = ee^T$ denotes the all-ones matrix. Since the variable y is one-dimensional and $\mathcal{C}$ is convex, we can approximate the optimal value by a bisection procedure: Assume we have a lower bound y_l with $B(y_l) := Q - y_l E \in \mathcal{C}$ and an upper bound y_u with $B(y_u) \notin \mathcal{C}$. Check whether $B(\frac{y_l+y_u}{2})$ is copositive. If it is, we can improve the lower bound, otherwise the upper bound. Iterating this procedure approximates the optimal value to arbitrary precision.

A special case of the standard quadratic problem is the maximum clique problem. Given an undirected graph G, it asks for a clique (i.e., a complete subgraph) of maximal cardinality. This maximal cardinality is called the *clique number* and is denoted by $\omega(G)$ or simply ω. Motzkin and Straus showed in [22] that

$$\tfrac{1}{\omega} = \min\{x^T(E - A_G)x : e^T x = 1,\ x \geq 0\} \tag{2}$$

where A_G is the adjacency matrix of G. Hence, the clique number can be determined by a standard quadratic optimization problem, which can be formulated as the copositive program

$$\tfrac{1}{\omega} = \max\{\lambda \in \mathbb{N} : (E - A_G) - \lambda E \in \mathcal{C}\}.$$

In the literature [13, Corollary 2.4], there is a second copositive formulation of the maximum clique problem:

$$\omega = \min\{\lambda \in \mathbb{N} : \lambda(E - A_G) - E \in \mathcal{C}\}. \tag{3}$$

These formulations are equivalent, see [9] for more details. As $\omega \in \{1, \ldots, n\}$, it can be determined by at most n copositivity tests using either formulation.

Whereas this paper will focus on the maximum clique problem, many other problems are also known to have a copositive formulation. The reader is referred to [12,15] and references therein for a more detailed exposition. Our paper is organized as follows: We first review and improve the copositivity conditions and the test algorithm introduced in [11]. In Sect. 3 we

discuss the behavior of our algorithm when applied to the maximum clique problem. This behavior depends crucially on the choice of a suitable set $\mathcal{M}$ in the algorithm. This choice is discussed in Sect. 4. Numerical experiments are described in Sect. 5.

2 Testing copositivity

2.1 Idea and notation

The starting point of our approach is the observation that A is copositive if and only if $x^T Ax \geq 0$ for all $x \geq 0$ with $\|x\|_1 = 1$. The set of all these points is called the standard simplex:

$$\Delta^S := \{x \in \mathbb{R}^n_+ : \|x\|_1 = 1\}.$$

The simplices Δ in $\mathbb{R}^n_+$ we consider are the convex hull of n affinely independent points (vertices) $v_1, \ldots, v_n$. The vertices of Δ^S are the unit vectors $e_1, \ldots, e_n$.

In [11], a sufficient copositivity condition based on the vertices of Δ^S was given, and it was shown how better conditions can be derived by looking at smaller and smaller parts of Δ^S. More formally, let Δ be some simplex in $\mathbb{R}^n$. A family $\mathcal{P} = \{\Delta^1, \ldots, \Delta^m\}$ of simplices satisfying

$$\Delta = \bigcup_{i=1}^{m} \Delta^i \qquad \text{and} \qquad \operatorname{int}(\Delta^i) \cap \operatorname{int}(\Delta^j) = \emptyset \quad \text{for} \quad i \neq j$$

is called a *simplicial partition* of Δ. Such a partition can be generated by successively bisecting simplices in the partition. For a more detailed description of simplicial partitions see [18]. It will be convenient to denote the set of vertices of partition $\mathcal{P}$ by

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

As a simplex Δ is determined by its vertices, it can be represented by a matrix V_Δ whose columns are these vertices. V_Δ is nonsingular and unique up to a permutation of its columns (which is irrelevant in our arguments). We shall refer to the set of all matrices corresponding to simplices in partition $\mathcal{P}$ as

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

We will quantify the "fineness" of a partition $\mathcal{P}$ by the maximum diameter of a simplex in $\mathcal{P}$, which we denote by

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

2.2 Copositivity conditions

Letting $\mathcal{N}$ resp. $\mathcal{S}^+$ denote the cones of entrywise nonnegative matrices resp. positive semidefinite matrices, it is easy to see from the definition that both $A \in \mathcal{N}$ and $A \in \mathcal{S}^+$ are sufficient conditions for copositivity of a matrix A. More generally, if $\mathcal{M} \subset \mathcal{C}$, then trivially $A \in \mathcal{M}$ is a sufficient condition for copositivity. Using simplicial partitions of Δ^S, this can be strengthened as follows:

Theorem 2.1 *Let $A \in \mathcal{S}$, let $\mathcal{M} \subset \mathcal{C}$, and let $\mathcal{P}$ be a simplicial partition of Δ^S. If*

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

then A is copositive.

Proof Let $x \in \Delta^S$. We have to show that $x^T A x \geq 0$. Let $\Delta \in \mathcal{P}$ be a simplex containing x, and let $v_1, \ldots, v_n$ denote its vertices. We represent x in barycentric coordinates with respect to Δ:

$$x = \sum_{i=1}^{n} \lambda_i v_i \quad \text{with} \quad \sum_{i=1}^{n} \lambda_i = 1.$$

Since $x \in \Delta$, we have $\lambda := (\lambda_1, \ldots, \lambda_n)^T \geq 0$, whence we get

$$x^T A x = (V\lambda)^T A (V\lambda) = \lambda^T \underbrace{V^T A V}_{\in \mathcal{M} \subset \mathcal{C}} \lambda \geq 0,$$

which shows that A is copositive. □

Theorem 2.1 is a generalization of the sufficient copositivity condition proposed in [11], where the case $\mathcal{M} = \mathcal{N}$ was discussed in detail. From the same paper it follows that, provided that the set $\mathcal{M}$ contains the nonnegative cone $\mathcal{N}$, we can formulate the following necessary criterion for strict copositivity:

Theorem 2.2 *Let $A \in \mathcal{S}$ be strictly copositive and let $\mathcal{M} \supseteq \mathcal{N}$. Then there exists $\varepsilon > 0$ such that for all partitions $\mathcal{P}$ of Δ^S with $\delta(\mathcal{P}) < \varepsilon$ we have*

$$V^T A V \in \mathcal{M} \quad \text{for all} \quad V \in M(\mathcal{P}). \tag{4}$$

Proof Since this statement was shown to be true for $\mathcal{M} = \mathcal{N}$ in [11, Theorem 2], this follows immediately. □

The results of this section can be interpreted as the construction of a new approximation hierarchy. We define

$$\mathcal{K}_{\mathcal{M},\mathcal{P}} = \{A \in \mathcal{S} : V^T A V \in \mathcal{M} \text{ for all } V \in M(\mathcal{P})\}.$$

Then Theorem 2.1 entails $\mathcal{K}_{\mathcal{M},\mathcal{P}} \subset \mathcal{C}$ for any $\mathcal{M} \subset \mathcal{C}$ and all partitions $\mathcal{P}$ of the standard simplex. And, assuming $\mathcal{M} \supseteq \mathcal{N}$, we get from Theorem 2.2 that

$$\operatorname{int}(\mathcal{C}) \subset \bigcup_{\varepsilon > 0} \bigcup_{\delta(\mathcal{P}) < \varepsilon} \mathcal{K}_{\mathcal{M},\mathcal{P}}.$$

Next, we prove a lemma which will be used to show that the algorithm described in the next section terminates finitely for $A \notin \mathcal{C}$.

Lemma 2.3 *Let $A \in \mathcal{S}$. The following two assertions are equivalent.*

1. *The matrix A is not copositive.*
2. *There exists $\varepsilon > 0$ such that for all partitions $\mathcal{P}$ of Δ^S with $\delta(\mathcal{P}) < \varepsilon$ there exists $v \in V(\mathcal{P})$ with $v^T A v < 0$.*

Proof Obviously, 2 implies 1. To show the converse, take $A \notin \mathcal{C}$. Then there exists $x \in \Delta^S$ with $x^T A x < 0$. Since the quadratic form $x^T A x$ is continuous, there exists $\varepsilon > 0$ such that $y^T A y < 0$ for all $y \in \mathbb{R}^n$ with $\|x - y\| < \varepsilon$. Let $\mathcal{P}$ be a partition of Δ^S with $\delta(\mathcal{P}) < \varepsilon$. Then there exists $v \in V(\mathcal{P})$ with $\|x - v\| < \varepsilon$ and consequently $v^T A v < 0$. Thus, 1 implies 2. □

2.3 The algorithm

The results of the preceding section naturally yield an algorithm to test whether a matrix is copositive or not: Starting with $\mathcal{P} = \{\Delta^S\}$, check whether there is a vertex v with $v^T Av < 0$, or whether the copositivity criterion of Theorem 2.1 is satisfied. If neither is the case, refine the partition and iterate the process. Formally, this procedure is stated in Algorithm 1.

Algorithm 1: Test whether a matrix A is copositive or not.

Input: $A \in \mathcal{S}, \mathcal{M} \subset \mathcal{C}$
Output: "A is copositive" or "A is not copositive"

```
1  P ← {Δ^S};
2  while P ≠ ∅ do
3      choose Δ ∈ P;
4      if ∃v ∈ V({Δ}) : v^T A v < 0 then
5          return "A is not copositive";
6      end
7      if V_Δ^T A V_Δ ∈ M then
8          P ← P \ {Δ};
9      else
10         partition Δ into Δ = Δ^1 ∪ Δ^2;
11         P ← P \ {Δ} ∪ {Δ^1, Δ^2};
12     end
13 end
14 return "A is copositive."
```

It is immediate from Theorem 2.1 and the definition of copositive matrices that if the algorithm terminates, the result is correct. Whether or not the algorithm does terminate depends on the input matrix A, the set $\mathcal{M}$, and the refinement strategy used in Step 10. For the latter, we will always assume that the refinement of the simplices is done in such a way that $\delta(\mathcal{P}) \to 0$.

- If $A \notin \mathcal{C}$, then the algorithm terminates, since then Lemma 2.3 applies. In this case, it does not matter which set $\mathcal{M}$ is used.
- If $\mathcal{M} \supseteq \mathcal{N}$ and A is strictly copositive, then Theorem 2.2 implies that the algorithm terminates. If A is copositive but not strictly copositive, then the algorithm may or may not terminate.
- If $\mathcal{M} = \mathcal{S}^+$ is used and $A \in \mathcal{S}^+$, then the algorithm terminates in one iteration because for any nonsingular matrix $V \in \mathbb{R}^{n\times n}$ we have

$$A \in \mathcal{S}^+ \quad \Leftrightarrow \quad V^T AV \in \mathcal{S}^+. \tag{5}$$

  If A is copositive but not positive semidefinite, then the algorithm does not terminate.

3 The algorithm and the maximum clique problem

3.1 Clique enumeration

We next describe an interesting feature that Algorithm 1 with $\mathcal{M} = \mathcal{N}$ exhibits when it is used on the maximum clique problem, namely, in this case it basically enumerates all cliques.

Of course this is a most undesirable behavior, and it will motivate different choices of $\mathcal{M}$ in Sect. 4.

Recall that in formulation (3) of the maximum clique problem we have to test copositivity of matrices $B_\lambda := \lambda(E - A_G) - E$ for $\lambda \in \{1, \ldots, n\}$. These matrices have the form

$$(B_\lambda)_{ij} = \begin{cases} -1 & \text{if } \{i, j\} \text{ is an edge in } G \\ \lambda - 1 & \text{otherwise.} \end{cases} \tag{6}$$

If we use Algorithm 1 with $\mathcal{M} = \mathcal{N}$ to test copositivity of B_λ for $\lambda \geq \omega$, it turns out that it enumerates every clique of G in form of the non-zero entries of a vertex v of a simplex Δ in the partition of Δ^S. We will use the following notation to describe the set of nodes corresponding to the non-zero entries of a vertex v:

$$C(v) = \{i : v_i > 0\}.$$

Theorem 3.1 *Consider a graph with clique number ω, and let C be a clique in G. Assume that Algorithm 1 with $\mathcal{M} = \mathcal{N}$ and $A = B_\lambda, \lambda \geq \omega$, terminates. Then it must terminate positively (i.e., in line 14). Further, if the branching rule in line 10 is such that a simplex Δ is split along an edge $\{u, v\}$, then the algorithm produces a partition $\mathcal{P}$ such that $C = C(v)$ for some $v \in V(\mathcal{P})$.*

Proof Since (3) implies $B_\lambda \in \mathcal{C}$, the algorithm cannot terminate with a negative certificate. Let C be a clique in G. If $|C| = 1$ then $C = \{i\}$ for some $i \in \{1, \ldots, n\}$, and $C(e_i) = C$. Consequently, the theorem is true for cliques of size 1.

So assume $|C| > 1$. We show that there is a sequence $(\Delta_i)_i$ of successive simplices generated by the algorithm ($\Delta_0 = \Delta^S$ and Δ_{i+1} is one of the two simplices that result of the bisection of Δ_i), such that

(*) there is a subset R^i of the vertices of Δ_i with

$$u^T B_\lambda v = -1, \text{ for all } u, v \in R^i \text{ with } u \neq v \quad \text{and} \quad \bigcup_{v \in R^i} C(v) = C.$$

We set $R^0 = \{e_i : i \in C\}$. Then for $e_i, e_j \in R^0$ with $i \neq j$ it holds $e_i^T B_\lambda e_j = -1$ and we have $\bigcup_{i \in C} C(e_i) = C$. It follows that Δ_0 fulfills (*). We now assume that Δ_i fulfills (*).

If $|R^i| = 1$ we have found a simplex such that $C = C(v)$ for a vertex v and the theorem is shown. If $|R^i| > 1$ there is at least one pair of vertices (v_1, v_2) with $v_1^T B_\lambda v_2 = -1$. This means that the algorithm does not yet terminate since $V_{\Delta_i}^T B_\lambda V_{\Delta_i} \notin \mathcal{N}$, and that the simplex is split at an edge $\{u, v\}$. Now we have to choose a successor simplex Δ_{i+1} satisfying (*). We distinguish two cases:

Case 1: $z = \mu u + (1 - \mu)v$ with $v \notin R^i$
We select

$$\Delta_{i+1} = \text{conv}(V(\{\Delta_i\}) \setminus \{v\} \cup \{z\})$$

and put $R^{i+1} = R^i$, so that property (*) holds for Δ_{i+1}.

Case 2: $z = \mu u + (1 - \mu)v$ with $u, v \in R^i$
We again put

$$\Delta_{i+1} = \text{conv}(V(\{\Delta_i\}) \setminus \{v\} \cup \{z\}).$$

We have $C(z) = C(u) \cup C(v)$. If $C(z) = C$ then Δ_{i+1} fulfills (*) and our statement is shown with $R^{i+1} = \{z\}$. Otherwise we set

$$R^{i+1} = R^i \setminus \{u, v\} \cup \{z\}.$$

Then

$$C = \bigcup_{w \in R^i \setminus \{u,v\}} C(w) \cup \underbrace{C(u) \cup C(v)}_{=C(z)} = \bigcup_{w \in R^{i+1}} C(w).$$

Since $R^{i+1} \setminus \{z\} = R^i \setminus \{u, v\}$ we have for all $w \in R^{i+1} \setminus \{z\}$ that

$$z^T B_\lambda w = \mu \underbrace{u^T B_\lambda w}_{=-1} + (1-\mu) \underbrace{v^T B_\lambda w}_{=-1} = -1$$

and for all $w_1, w_2 \in R^{i+1} \setminus \{z\}$, $w_1 \neq w_2$,

$$w_1^T B_\lambda w_2 = -1.$$

Hence Δ_{i+1} fulfills (*).

Now assume that the algorithm and hence our successor sequence terminates, say, at Δ_N. As argued above, this can only happen with a singleton $R^N = \{z\}$, and then $C = C(z)$. □

3.2 Non-strict copositivity of B_ω and an easy remedy

The formulation (3) implies that the matrices B_λ are not copositive for $\lambda < \omega$ and strictly copositive for $\lambda > \omega$. We prove in the following lemma that the matrix B_ω lies on the boundary of $\mathcal{C}$.

Proposition 3.2 *Let G be a graph with clique number ω. The matrix*

$$B_\omega = \omega(E - A_G) - E$$

is copositive but not strictly copositive.

Proof According to (3), the matrix $B_\omega = \omega(E - A_G) - E$ is copositive. To show that it is not strictly copositive, let C denote a maximum clique and x_C the incidence vector of C, i.e.

$$(x_C)_i = \begin{cases} 1 & \text{if } i \in C \\ 0 & \text{otherwise.} \end{cases}$$

Note that $x_C \neq 0$. From (6) we get $x_C^T B_\omega x_C = \sum_{i,j} (B_\omega)_{ij} (x_C)_i (x_C)_j = \omega(\omega - 1) + \omega(\omega - 1)(-1) = 0$, which says that B_ω is not strictly copositive. □

As noted in [11], Algorithm 1 with $\mathcal{M} = \mathcal{N}$ may fail to terminate if the input matrix lies on the boundary of $\mathcal{C}$. However, we next show that the matrices B_λ can be replaced by a modified matrix without changing the result. For $\lambda \geq \omega$, this modified matrix lies in the interior of $\mathcal{C}$, whereas for $\mathbb{N} \ni \lambda < \omega$ it is not in $\mathcal{C}$. This means that numerical problems can be avoided by solving the following problem instead of (3):

Theorem 3.3 *Let $0 \leq \rho < \frac{1}{\omega}$. Then the clique number ω can be obtained from the following modified copositive program:*

$$\omega = \min\{\lambda \in \mathbb{N} : B_\lambda + \rho \cdot E \in \mathcal{C}\}.$$

Moreover, $B_\lambda + \rho \cdot E$ is strictly copositive for any $\lambda \geq \omega$ and $\rho > 0$.

Proof Let C be a maximum clique and let $\hat{x} = \frac{1}{\omega} x_C$, where x_C denotes the incidence vector of C. Then it is easy to verify that $\hat{x}^T A_G \hat{x} = 1 - \frac{1}{\omega}$. So for $1 \leq \lambda \leq \omega - 1$ and $0 \leq \rho < \frac{1}{\omega}$ we have

$$\begin{aligned}\hat{x}^T (B_\lambda + \rho \cdot E)\hat{x} &= \hat{x}^T (\lambda(E - A_G) - E)\hat{x} + \rho \cdot \hat{x}^T E \hat{x} \\ &= (\lambda - 1 + \rho) \underbrace{\hat{x}^T E \hat{x}}_{=1} - \lambda \underbrace{\hat{x}^T A_G \hat{x}}_{=1-\frac{1}{\omega}} \\ &= \frac{\lambda - \omega}{\omega} + \rho \leq -\frac{1}{\omega} + \rho < 0.\end{aligned}$$

Consequently, $B_\lambda + \rho \cdot E \notin \mathcal{C}$ for all $\lambda \in \mathbb{N}$ with $\lambda < \omega$.

If $\rho > 0$ and $\lambda \geq \omega$, then for all $x \in \mathbb{R}^n_+ \setminus \{0\}$ we have

$$x^T (B_\lambda + \rho \cdot E)x = \underbrace{x^T (\lambda(E - A_G) - E)x}_{\geq 0} + \rho \cdot \underbrace{x^T E x}_{>0} > 0.$$

Therefore, $B_\lambda + \rho \cdot E$ is strictly copositive and lies in the interior of $\mathcal{C}$ for these values of ρ and λ. This holds in particular for $B_\omega + \rho \cdot E$ with $\rho > 0$. □

If the clique number of a graph is not known but an upper bound $u \geq \omega$ is given, we can choose any ρ such that $0 < \rho < \frac{1}{u} \leq \frac{1}{\omega}$. Consequently, we can avoid termination problems for Algorithm 1 by solving the modified version of the problem defined in Theorem 3.3.

The idea of the shift perturbation used in Theorem 3.3 is not new and has been used by several authors before. A systematic investigation of shift-equivariance of StQP-bounds which are closely related to testing copositivity is offered in [8]. Also, the shift perturbation is similar to the regularization approach where the identity matrix I replaces E. For details see [2–4].

4 How to choose $\mathcal{M}$

An important issue which influences the number of iterations and the runtime of Algorithm 1 is the choice of the set $\mathcal{M}$. To keep the number of iterations small the set $\mathcal{M}$ should be a good approximation of $\mathcal{C}$. On the other hand, checking membership of $\mathcal{M}$ should be cheap to keep the total runtime short. As we saw in Sect. 3.1, the choice $\mathcal{M} = \mathcal{N}$ is not always desirable. To check whether a matrix is in $\mathcal{N}$ does not take much effort but the nonnegative cone is a quite bad approximation of the copositive cone. So each iteration of the algorithm is cheap but the number of iterations will tend to be large. Hence, we discuss other possible choices of $\mathcal{M}$ here.

4.1 $\mathcal{M} = \mathcal{S}^+$

Using $\mathcal{M} = \mathcal{S}^+$ would be the next canonical choice, but unfortunately this is not appropriate, either. The problem is that for $\mathcal{M} = \mathcal{S}^+$ condition (4) from Theorem 2.2 is never met if the input matrix is not in $\mathcal{S}^+$. To see this, consider the strictly copositive but not positive semidefinite matrix

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix},$$

and consider the simplex $\Delta = \{e_1, v\}$ with $v = \lambda e_1 + (1-\lambda)e_2$ and $\lambda \in (0,1)$, which is present in any non-trivial partition $\mathcal{P}$ of Δ^S in $\mathbb{R}^2$. The matrix

$$V_\Delta^T A V_\Delta = \begin{pmatrix} e_1^T \\ v^T \end{pmatrix} A \begin{pmatrix} e_1 & v \end{pmatrix} = \begin{pmatrix} 1 & 2-\lambda \\ 2-\lambda & 1+2\lambda-2\lambda^2 \end{pmatrix}$$

has determinant $-3(\lambda-1)^2$ and is thus indefinite for all $\lambda \in (0,1)$. Consequently, there does not exist a partition $\mathcal{P}$ such that (4) is true for $\mathcal{M} = \mathcal{S}^+$.

Moreover, we already saw in (5) that for any nonsingular matrix $V \in \mathbb{R}^{n\times n}$ we have $A \in \mathcal{S}^+$ if and only if $V^TAV \in \mathcal{S}^+$. Hence the sufficient copositivity condition of Theorem 2.1 reduces to positive semidefiniteness, which is trivially sufficient for copositivity. For these reasons, the choice $\mathcal{M} = \mathcal{S}^+$ is not favorable.

4.2 $\mathcal{M} = \mathcal{S}^+ + \mathcal{N}$

By definition of copositivity, we have $\mathcal{S}^+ + \mathcal{N} \subseteq \mathcal{C}$, and for matrices up to order 4×4, it is known that $\mathcal{C} = \mathcal{S}^+ + \mathcal{N}$ (cf. [14]), so $\mathcal{S}^+ + \mathcal{N}$ is a good approximation to the copositive cone. This fact also implies that for graphs with at most 4 nodes the matrix B_λ lies in $\mathcal{S}^+ + \mathcal{N}$ for all $\lambda \geq \omega$. But there are more graphs for which this is true. Denoting by χ the chromatic number of a graph, the following was shown in [24, Corollary 15].

Lemma 4.1 *If G is a graph with $\chi = \omega$, then $B_\lambda \in \mathcal{S}^+ + \mathcal{N}$ for all $\lambda \geq \omega$.*

This means that for these graphs all copositive matrices B_λ, $\lambda \geq \omega$, can be written as the sum of a positive semidefinite and a nonnegative matrix. In particular Lemma 4.1 holds for perfect graphs. So if we use Algorithm 1 with $\mathcal{M} = \mathcal{S}^+ + \mathcal{N}$, we can determine copositivity of these matrices in the first iteration.

In fact, for any perfect graph (and any other graph for which the chromatic number equals the clique number), all SDP-bounds coincide. So even the Lovász number ϑ of the complementary graph of G equals ω, and to compute this number we only need $\mathcal{S}^+$ and not $\mathcal{S}^+ + \mathcal{N}$. But as we have seen in the previous subsection, using $\mathcal{M} = \mathcal{S}^+$ is not a good choice in general. And of course, there are also graphs for which $B_\omega \notin \mathcal{S}^+ + \mathcal{N}$ which implies $B_\omega \notin \mathcal{S}^+$. For these graphs, copositivity of B_ω cannot be verified using $\mathcal{M} = \mathcal{S}^+$, which motivates the choice of $\mathcal{M} = \mathcal{S}^+ + \mathcal{N}$. The smallest example of such a graph is the cycle with 5 nodes whose clique number is $\omega = 2$. Its corresponding matrix B_ω is the Horn-matrix H which can be shown to fulfill $H \notin \mathcal{S}^+ + \mathcal{N}$, see for example [17].

The cone $\mathcal{S}^+ + \mathcal{N}$ may be a good approximation of $\mathcal{C}$, but in order to check if a matrix A is in $\mathcal{S}^+ + \mathcal{N}$ we need to solve the semidefinite feasibility problem

$$\begin{aligned} A - N &\in \mathcal{S}^+ \\ N_{ij} &\geq 0 \quad \forall\, i,j \end{aligned}$$

which is possible in polynomial time but quite costly. So we may need only few iterations but still have a high runtime.

4.3 An alternative choice for $\mathcal{M}$

The above observations indicate that it may be favorable to choose as $\mathcal{M}$ some cone between $\mathcal{N}$ and $\mathcal{S}^+ + \mathcal{N}$. Theorem 2.2 suggests that is is desirable to impose $\mathcal{M} \supset \mathcal{N}$ to ensure termination of the algorithm.

To define such a set, observe that we can decompose any $A \in \mathcal{S}$ into $A = N(A) + S(A)$, where $N(A)$ denotes the matrix containing the positive non-diagonal entries of A:

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

The remaining part $S(A) := A - N(A)$ is not necessarily positive semidefinite (even if $A \in \mathcal{S}^+ + \mathcal{N}$), but if it is, then we have a decomposition which shows $A \in \mathcal{S}^+ + \mathcal{N}$. This motivates to define

$$\mathcal{H} := \{A \in \mathcal{S} : S(A) \in \mathcal{S}^+\},$$

and the next theorem shows that $\mathcal{H}$ has indeed the desired properties:

Theorem 4.2 *$\mathcal{H}$ is a convex cone, and $\mathcal{N} \subset \mathcal{H} \subset \mathcal{S}^+ + \mathcal{N}$. If $n \geq 3$, these inclusions are strict and $\mathcal{S}^+ \nsubseteq \mathcal{H}$. For $n = 2$ we have $\mathcal{H} = \mathcal{S}^+ \cup \mathcal{N} = \mathcal{S}^+ + \mathcal{N} = \mathcal{C}$.*

To show convexity of $\mathcal{H}$, we need the following auxiliary lemma:

Lemma 4.3 *Denote by $\mathcal{Z}$ the class of all real square matrices whose off-diagonal entries are nonpositive. Let $A, B \in \mathcal{Z}$ with $B \geq A$. If A is positive semidefinite, then B is also positive semidefinite.*

Proof Let $A, B \in \mathcal{Z}$ such that A is positive semidefinite and $B \geq A$. We make use of a similar result on positive definiteness taken from [16]: for $n \in \mathbb{N}$, consider the sequences $A_n = A + \frac{1}{n} I$ and $B_n = B + \frac{1}{n} I$. Then $A_n, B_n \in \mathcal{Z}$ and $B_n \geq A_n$ for all $n \in \mathbb{N}$. Furthermore, the matrices A_n are positive definite. From [16, (4.2)] it follows that the matrices B_n are positive definite. Since $B = \lim_{n\to\infty} B_n$, it follows that B is positive semidefinite. □

Proof (of Theorem 4.2) Both the property that $\mathcal{H}$ is a cone and the inclusions $\mathcal{N} \subset \mathcal{H} \subset \mathcal{S}^+ + \mathcal{N}$ are immediate from the definitions. For $n \geq 3$ both inclusions are strict, since

$$A := \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \in \mathcal{H} \quad \text{but} \quad A \notin \mathcal{N}$$

and

$$B := \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix} \in \mathcal{S}^+ \subset \mathcal{S}^+ + \mathcal{N} \quad \text{but} \quad B \notin \mathcal{H}.$$

For a matrix A of order 2×2 the following two cases can occur:

1. $a_{12} > 0 \Rightarrow S(A) = A - N(A) = \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix}$,
2. $a_{12} \leq 0 \Rightarrow S(A) = A - N(A) = A$.

In the first case, $A \in \mathcal{H}$ if and only if its diagonal entries are nonnegative which means that $A \in \mathcal{N}$. In the second case, we have $A \in \mathcal{H}$ if and only if $A \in \mathcal{S}^+$. It follows that $\mathcal{H} = \mathcal{S}^+ \cup \mathcal{N} = \mathcal{S}^+ + \mathcal{N} = \mathcal{C}$.

To show that $\mathcal{H}$ is convex, take $A, B \in \mathcal{H}$. We have to show that $A + B \in \mathcal{H}$. $A, B \in \mathcal{H}$ means that both $S(A) \in \mathcal{S}^+$ and $S(B) \in \mathcal{S}^+$, and hence $S(A) + S(B) \in \mathcal{S}^+$. By construction, we have $S(A + B) \geq S(A) + S(B)$. Consequently, $S(A + B) \in \mathcal{S}^+$ by Lemma 4.3, whence $A + B \in \mathcal{H}$. □

Table 1 Test instances

Instance	n	Problem	Value of y resp. λ
5_cycleCopos	5	(3)	2
5_cycleNotCopos	5	(3)	1.5
graph8_3Copos	8	(3)	3
graph8_3NotCopos	8	(3)	2.5
graph12_4Copos	12	(3)	4
graph12_4NotCopos	12	(3)	3.5
penaCopos	17	(3)	6
penaNotCopos	17	(3)	5.5
geneticCopos	5	(StQP)	$-16\frac{1}{3}$
geneticNotCopos	5	(StQP)	-16
icosahedronCopos	12	(StQP)	$\frac{1}{3}$
icosahedronNotCopos	12	(StQP)	0.5
pentagonCopos	5	(StQP)	0.5
pentagonNotCopos	5	(StQP)	1
portCopos	5	(StQP)	0.48
portNotCopos	5	(StQP)	0.5

The instances graph8_3 and graph12_4 are described in Fig. 1. The graph of the max clique related instances pena is the complementary graph of G_{17} from [24] (with ω=6). The instances (StQP) were taken from [5]

5 Numerical results

We implemented Algorithm 1 with various choices of $\mathcal{M}$ in Matlab and tested our implementation on a Pentium IV, 2.8 Gigahertz Linux machine. To solve the semidefinite programs resulting from the test $A \in \mathcal{S}^+ + \mathcal{N}$ we used yalmip [21] and sedumi [25].

As test-instances we used the matrices $B(y) = Q - yE$ originating from the standard quadratic problem (1), and B_λ from the max clique problem as described in the previous section. The test instances with their respective values of y resp. λ are listed in Table 1.

We solved these instances with our algorithm for different choices of $\mathcal{M}$. Note that the algorithm has freedom not only in the choice of $\mathcal{M}$, but also in the strategy according to which the simplex is refined (see Step 10 in Algorithm 1). Different refinement strategies have already been discussed in [11] and [10], so we will not go into too much detail here. The basic idea behind all strategies for the case $\mathcal{M} = \mathcal{N}$ is to partition one of the edges $\{v_i, v_j\}$ which give the most negative value $v_i^T A v_j$.

These strategies can be adapted for the choices $\mathcal{M} = \mathcal{H}$ or $\mathcal{M} = \mathcal{S}^+ + \mathcal{N}$ as follows:

Consider the case $\mathcal{M} = \mathcal{H}$. The partition is refined if $V^T AV \notin \mathcal{H}$. Then there exists an eigenvector $x \in \mathbb{R}^n$ with

$$0 > x^T S(V^T AV)x = \sum_{i,j=1}^{n} (S(V^T AV))_{ij}\, x_i x_j.$$

A promising strategy is to choose the eigenvector x corresponding to the smallest eigenvalue and then choose $i, j \in \{1, \ldots, n\}$ such that

$$(S(V^T AV))_{ij}\, x_i x_j = \min_{i,j\in\{1,\ldots,n\}} (S(V^T AV))_{ij}\, x_i x_j.$$

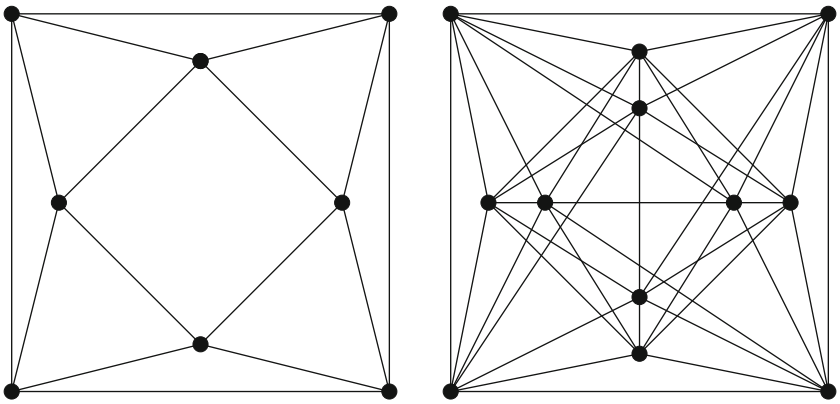

Fig. 1 The graphs corresponding to the instances `graph8_3` (*left*, with $\omega = 3$) and `graph12_4` (*right*, with $\omega = 4$)

Next, consider the case $\mathcal{M} = \mathcal{S}^+ + \mathcal{N}$. To test whether $V^T A V \in \mathcal{S}^+ + \mathcal{N}$ we solve the following semidefinite program

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

Its optimal value is negative if and only if $V^T A V \notin \mathcal{S}^+ + \mathcal{N}$. In this case, we refine the partition by splitting an edge of the simplex which corresponds to V. Consider the objective value for an optimal X:

$$0 > \langle V^T A V, X \rangle = \sum_{i,j=1}^{n} v_i^T A v_j X_{ij}$$

It seems promising to partition the edge $\{v_i, v_j\}$ which has the largest (negative) contribution to this sum, i.e., to choose i and j such that

$$v_i^T A v_j X_{ij} = \min_{i,j \in \{1,\ldots,n\}} v_i^T A v_j X_{ij}.$$

Unfortunately, this strategy turned out to be not very efficient. Therefore, we simply choose $i, j \in \{1, \ldots, n\}$ such that

$$X_{ij} = \min_{i,j \in \{1,\ldots,n\}} X_{ij}$$

and split the edge $\{v_i, v_j\}$. This turned out to give better results.

Generally, it is clear that the behavior of our algorithm highly depends on the partitioning strategy. There does not seem to be one strategy which outperforms the others for every instance, but the strategies described above seem to work reasonably well in our test instances.

Our results are stated in Tables 2 and 3. As expected, for $\mathcal{M} = \mathcal{S}^+ + \mathcal{N}$ we need the smallest number of iterations. In spite of that, all instances were solved faster with $\mathcal{M} = \mathcal{N}$ or $\mathcal{M} = \mathcal{H}$. Another observation of our experiments is that it seems to be much easier to verify non-copositivity of a matrix than copositivity.

These observations motivate the following idea to solve the maximum clique problem by our test procedure. We start with $\lambda = 1$ and test whether the matrices $B_1, B_2, \ldots$ are

Table 2 The number of iterations needed by the algorithm for different choices of $\mathcal{M}$

Instance	n	Iterations		
		$\mathcal{M} = \mathcal{N}$	$\mathcal{M} = \mathcal{H}$	$\mathcal{M} = \mathcal{S}^+ + \mathcal{N}$
5_cycleCopos	5	19	7	3
5_ cycleNotCopos	5	1	1	1
graph8_3Copos	8	1359	151	9
graph8_3NotCopos	8	2	2	3
graph12_4Copos	12	(128105)	21719	225
graph12_4NotCopos	12	6	7	4
penaCopos	17	(73058)	(84188)	(1054)
penaNotCopos	17	977	147	(829)
geneticCopos	5	29	7	1
geneticNotCopos	5	1	1	1
icosahedronCopos	12	71679	5183	703
icosahedronNotCopos	12	2	3	3
pentagonCopos	5	19	7	3
pentagonNotCopos	5	1	1	1
portCopos	5	25	5	1
portNotCopos	5	2	2	3

Brackets indicate that the algorithm did not terminate within half an hour

Table 3 The runtime for different choices of $\mathcal{M}$

Instance	Cpu-time (s)		
	$\mathcal{M} = \mathcal{N}$	$\mathcal{M} = \mathcal{H}$	$\mathcal{M} = \mathcal{S}^+ + \mathcal{N}$
5_cycleCopos	0.0055	0.0680	3.9753
5_cycleNotCopos	0.0008	0.0010	1.1225
graph8_3Copos	0.9785	0.2639	18.4915
graph8_3NotCopos	0.0092	0.0016	2.1444
graph12_4Copos	–	50.8841	335.5044
graph12_4NotCopos	0.0103	0.0045	5.7601
penaCopos	–	–	–
penaNotCopos	3.3632	0.3289	–
geneticCopos	0.0088	0.0691	0.7819
geneticNotCopos	0.0008	0.0010	0.6532
icosahedronCopos	218.0965	6.3096	1467.0274
icosahedronNotCopos	0.0013	0.0022	4.4716
pentagonCopos	0.0029	0.0712	5.9983
pentagonNotCopos	0.0042	0.0009	1.7977
portCopos	0.0190	0.0893	1.9345
portNotCopos	0.0013	0.0015	5.7740

If no value is given, this means that the algorithm did not terminate within half an hour

Table 4 Results for max clique instances from the second DIMACS challenge

Instance	Vertices	Edges	ω	Bounds for ω	# of victories			
					$\mathcal{N}$	$\mathcal{H}$	$\mathcal{S}^+ + \mathcal{N}$	None
`hamming6-2`	64	1824	32	21/32	12	0	1	19
`hamming6-4`	64	704	4	4/4	4	0	0	0
`johnson8-2-4`	28	210	4	4/4	2	1	1	0
`johnson8-4-4`	70	1855	14	12/14	11	0	0	3
`MANN_a9`	45	918	16	16/18	9	2	1	6

copositive. The first value of λ for which B_λ is copositive is the clique number. From the viewpoint of computation time it would be favorable to choose $\mathcal{M} \in \{\mathcal{N}, \mathcal{H}\}$ if the matrix is not copositive and $\mathcal{M} = \mathcal{S}^+ + \mathcal{N}$ otherwise. As we do not know this a priori, we started for each value of λ three parallel processes for the three considered choices of $\mathcal{M}$. When one process terminates we stop the others and increase λ. We also stop if the matrix is found to be copositive. We tested this procedure with some instances from the second DIMACS challenge. The results are listed in Table 4. The columns "# of victories" record the number of times the process with the respective choice of $\mathcal{M}$ was the first to terminate. We restricted the runtime for each value of λ to one hour and the amount of memory consumed by the partition to 500MB. For some instances the algorithm did not terminate with these restrictions. The last column indicates how often this happened. In this case we only got a lower and upper bound for the clique number which are shown in the column "bounds for ω". Consider for example the instance `hamming6-2`. The matrix B_{20} was found to be non-copositive which implies that $\omega \geq 21$. For the matrices $B_{21}, \ldots, B_{31}$ the algorithm did not terminate within the restrictions, and for B_{32} copositivity was verified. This means that we have a lower bound of 21 and an upper bound of 32.

6 Conclusion

We gave an improved algorithm which is built on a generalization of results of [11]. We saw that for the performance of this algorithm the choice of the set $\mathcal{M}$ is crucial. The total runtime is a trade-off between the number of iterations and the cost of each iteration. Choosing $\mathcal{M} = \mathcal{S}^+ + \mathcal{N}$ requires the least number of iterations but each iteration is so costly that the overall runtime is in most cases still higher than when choosing $\mathcal{M} = \mathcal{N}$ or $\mathcal{M} = \mathcal{H}$. We observe the well known numerical phenomenon that verifying copositivity is much harder than verifying non-copositivity.

Acknowledgments The authors wish to thank the two anonymous referees for careful reading and useful suggestions which helped to improve the presentation of the paper. M. Dür was partially supported by the Netherlands Organisation for Scientific Research (NWO) through Vici grant no.639.033.907.

References

1. Bomze, I.M.: Copositive optimization – recent developments and applications. Eur. J. Oper. Res. forthcoming (2011)
2. Bomze, I.M.: On standard quadratic optimization problems. J. Glob. Optim. **13**, 369–387 (1998)
3. Bomze, I.M.: Evolution towards the maximum clique. J. Glob. Optim. **10**, 143–164 (1997)
4. Bomze, I.M., Budinich, M., Pelillo, M., Rossi, C.: Annealed replication: a new heuristic for the maximum clique problem. Discrete Appl. Math. **121**, 27–49 (2002)
5. Bomze, I.M., de Klerk, E.: Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. J. Glob. Optim. **24**, 163–185 (2002)
6. 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**, 301–320 (2000)
7. Bomze, I.M., Eichfelder, G.: Copositivity detection by difference-of-convex decomposition and ω-subdivision. Preprint (2010), available online at http://www.optimization-online.org/DB_HTML/2010/01/2523.html
8. Bomze, I.M., Locatelli, M., Tardella, F.: New and old bounds for standard quadratic optimization: dominance, equivalence and incomparability. Math. Program. **115**, 31–64 (2008)
9. Bundfuss, S.: Copositive matrices, copositive programming, and applications. Ph.D. Dissertation, TU Darmstadt (2009). Online at http://www3.mathematik.tu-darmstadt.de/index.php?id=483
10. Bundfuss, S., Dür, M.: An adaptive linear approximation algorithm for copositive programs. SIAM J. Optim. **20**, 30–53 (2009)
11. Bundfuss, S., Dür, M.: Algorithmic copositivity detection by simplicial partition. Linear Algebra Appl. **428**, 1511–1523 (2008)
12. Burer, S.: On the copositive representation of binary and continuous nonconvex quadratic programs. Math. Program. **120**, 479–495 (2009)
13. de Klerk, E., Pasechnik, D.V.: Approximation of the stability number of a graph via copositive programming. SIAM J. Optim. **12**, 875–892 (2002)
14. Diananda, P.: On non-negative forms in real variables some or all of which are non-negative. Proc. Camb. Philol. Soc. **58**, 17–25 (1962)
15. Dür, M.: Copositive programming – a survey. In: Diehl, M., Glineur, F., Jarlebring, E., Michiels, W. (eds.) Recent Advances in Optimization and its Applications in Engineering, pp. 3–20. Springer, Berlin (2010)
16. Fiedler, M., Pták, V.: On matrices with non-positive off-diagonal elements and positive principal minors. Czechoslovak Math. J. **12**, 382–400 (1962)
17. Hall, M. Jr., Newman, M.: Copositive and completely positive quadratic forms. Proc. Camb. Philol. Soc. **59**, 329–339 (1963)
18. Horst, R.: On generalized bisection of n-simplices. Math. Comput. **218**, 691–698 (1997)
19. Hiriart-Urruty, J.-B., Seeger, A.: A variational approach to copositive matrices. SIAM Rev. **52**, 593–629 (2010)
20. Ikramov, K.D., Savel'eva, N.: Conditionally definite matrices. J. Math. Sci. **99**, 1–50 (2000)
21. Löfberg, J.: YALMIP: A toolbox for modeling and optimization in MATLAB. In: Proceedings of the CACSD Conference, Taipei, Taiwan (2004)
22. Motzkin, T.S., Straus, E.G.: Maxima for graphs and a new proof of a theorem of Turan. Canadian J. Math. **17**, 533–540 (1965)
23. Murty, K.G., Kabadi, S.N.: Some NP-complete problems in quadratic and nonlinear programming. Math. Program. **39**, 117–129 (1987)
24. Peña, J., Vera, J., Zuluaga, L.: Computing the stability number of a graph via linear and semidefinite programming. SIAM J. Optim. **18**, 87–105 (2007)
25. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw. **11/12**, 625–653 (1999)