Computational Optimization and Applications manuscript No.
(will be inserted by the editor)

Copositivity tests based on the Linear Complementarity Problem

Carmo Brás · **Gabriele Eichfelder** ·
Joaquim Júdice

Received: date / Accepted: date

Abstract We present copositivity tests based on new necessary and sufficient conditions which require the solution of linear complementarity problems (LCP). We propose methodologies involving Lemke's method, an enumerative algorithm and a linear mixed-integer programming formulation to solve the required LCPs. Moreover, we discuss a new necessary condition for (strict) copositivity based on solving a Linear Program (LP), which can be used as a preprocessing step. The algorithms with these three different variants are thoroughly applied to test matrices from the literature and to max-clique instances with matrices of order up to 496×496. We compare our procedures with three other copositivity tests from the literature as well as with a general global optimization solver. The numerical results are very promising and equally good and in many cases better than the results reported elsewhere.

Keywords Conic optimization · Copositivity · Complementarity problems · Linear programming

Mathematics Subject Classification (2000) 15A48 90C33 65F30 65K99 90C26

The work of the first author was partially supported by the Portuguese Foundation for Science and Technology through the project UID/MAT/00297/2013 (CMA).
The research of the third author was in the scope of R&D Unit 50008, financed by the applicable financial framework (FCT/MEC through national funds and when applicable co-funded by FEDER PT2020 partnership agreement.

Carmo Brás
CMA, Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa, Portugal
E-mail: mb@fct.unl.pt

Gabriele Eichfelder
Institute for Mathematics, Technische Universität Ilmenau
Po 10 05 65, D-98684 Ilmenau, Germany
E-mail: Gabriele.Eichfelder@tu-ilmenau.de

Joaquim Júdice
Instituto de Telecomunicações, Portugal
E-mail: joaquim.judice@co.it.pt

1 Introduction

The notion of (strict) copositivity of a matrix [11,26] is well known in the area of linear complementarity problems (LCP) in the context of existence results and results on the successful termination of Lemke's algorithm — a well known simplex-like vertex following algorithm for LCPs [11,17,26]. An $n \times n$ matrix is called $\mathbb{R}^n_+$-copositive (or $\mathbb{R}^n_+$-semidefinite or shortly copositive), if it generates a quadratic form which takes no negative values on the nonnegative orthant $\mathbb{R}^n_+$. A copositive matrix is called strictly copositive if it generates a quadratic form which takes only positive values on $\mathbb{R}^n_+ \setminus \{0\}$.

In the last decade there has been an increasing interest in this property of a matrix, and in linear optimization problems over the cone of copositive matrices. For recent surveys on copositive programming we refer to [2,5,10,15,19]. This interest is primarily based on the fact that some hard problems as the Maximum Clique problem (see [4,13]) were shown to have a reformulation as a copositive program. Burer showed in [9] that, under weak assumptions, every quadratic program with linear constraints can be formulated as a copositive program, even if some of the variables are binary. Hence efficient numerical copositivity tests are essential.

The problem of determining whether a matrix is not copositive is NP-complete [27]. As discussed in [3] various authors have proposed such a test. However, there are only a few implemented numerical algorithms which apply to general symmetric matrices without any structural assumptions or dimensional restrictions and which are not just recursive, i.e., do not rely on information taken from all principal submatrices. There are some quite recent implementations which satisfy both criteria to full extent: Bundfuss and Dür proposed in [7,8] the first algorithm, see also the modification and improvements by Žilinskas and Dür [33], by Sponsel, Bundfuss and Dür [29] and by Tanaka and Yoshise [30]. Later, Bomze and Eichfelder [3] presented another algorithm.

Both approaches combine necessary and sufficient criteria for copositivity with a branch-and-bound algorithm. The branching is done in a data driven way and consists in a partitioning of the standard simplex into subsimplices. For each of the subsimplices it is tested whether a sufficient criterion for copositivity is satisfied or whether a necessary condition is violated. The approaches use different necessary and sufficient criteria but both approaches obtain better results in verifying that a matrix is not copositive than in proving that a matrix is copositive. Moreover, the algorithms are also in most cases more successful in showing copositivity for matrices which are also strictly copositive. The first approach by Bundfuss and Dür is based on the evaluation of a set of inequalities on each subsimplex, while the second one by Bomze and Eichfelder requires to solve convex quadratic and linear optimization problems.

We present in this paper a new numerical approach for testing whether a matrix is copositive. This method does not rely on any assumptions on the matrix and does not use information from submatrices. We give new necessary and sufficient conditions. These are based on the relation between a (global) quadratic optimization problem and a mathematical program with linear complementarity constraints (MPLCC). The quadratic optimization problem is the minimization of the quadratic form of the matrix over the standard simplex, which is equivalent to testing whether the matrix is copositive. We derive conditions by studying some linear complementarity problems (LCP) which deliver feasible solutions for the MPLCC. These conditions require the determination of solutions of LCPs. We use an enumerative algorithm [22], Lemke's algorithm [11, Chapter 4.4] and a mixed integer formulation (MIP) [21] for this purpose.

Some of the new necessary conditions are easy and fast to verify as they require for instance the application of Lemke's algorithm only. Thus these conditions can also be evaluated as an additional criterion in each iteration of copositivity tests different from the ones mentioned above. Moreover, we use known preprocessing results and combine them with a new preprocessing step based on solving linear problems (LP).

We test the derived procedures on some famous matrices from the literature as well as on maximum-clique instances from the DIMACS challenge and generated smaller instances from the maximum clique problem. These matrices are also used as test instances for the above mentioned approaches. The considered matrices are up to order 496×496. The numerical results show that the procedures are quite efficient in showing that a matrix is not strictly copositive or even not copositive. A hybrid algorithm which combines some of the procedures discussed in this paper is successful for all instances. In particular, the algorithm is able to establish that the maximum clique is a lower-bound for each of the instances. More numerical effort is needed to verify that matrices are strictly copositive or copositive but not strictly copositive (which is a well known drawback also for the above mentioned approaches).

Instead of applying algorithms which are especially designed for testing whether a given matrix is copositive one could also directly apply a global optimization solver, as for instance BARON, to the quadratic optimization problem mentioned above. In this paper, for the first time, a numerical copositivity test is compared to a general global optimization solver. For the predefined allowed time BARON fails at five of the large instances while our proposed hybrid algorithm can solve all the instances.

The remainder of this article is structured as follows: in Section 2 we give necessary and sufficient conditions for copositivity based on a reformulation as a mathematical program with linear complementarity constraints (MPLCC). From that we derive conditions based on the solutions of LCPs which we again characterize by solutions of MIPs. We also give necessary conditions for copositivity based on LPs. In Section 3 we present the algorithms which consist of two basic steps 0 and 1 (preprocessing and applying Lemke's method) and a step 2 for which we present three different possible procedures. Numerical experiments with these techniques are reported in Section 4. In the last section we give some conclusions and an outlook on possible extensions of the proposed methods to be done in the future.

2 Sufficient and necessary conditions for copositivity

In this section, we first recall some basic definitions. Then we define a mathematical program with linear complementarity constraints (MPLCC). We study its relation to the task of testing whether a matrix is copositive. We derive necessary and sufficient conditions for (strict) copositivity based on linear complementarity problems (LCPs) and on a mixed integer formulation of one of these LCPs. Finally, we give some easy to verify necessary conditions for (strict) copositivity based on solving linear problems (LPs).

2.1 Copositivity and Global Optimization

We recall the definition of (strict) copositivity:

Definition 1 A real $n \times n$ matrix M is called copositive if $x^\top Mx \geq 0$ for all $x \in \mathbb{R}^n_+$, and strictly copositive if $x^\top Mx > 0$ for all $x \in \mathbb{R}^n_+ \setminus \{0\}$.

Any strictly copositive matrix is copositive. As a real $n \times n$ matrix M is (strictly) copositive if and only if the symmetric matrix $\frac{1}{2}(M + M^\top)$ is (strictly) copositive, we restrict our examinations to symmetric matrices. Let $\mathcal{S}$ denote the real linear space of real symmetric $n \times n$ matrices. The set of copositive symmetric $n \times n$ matrices is a convex cone and the interior of the cone of copositive matrices is the set of strictly copositive symmetric matrices (cf. [7,16]). We denote the cone of copositive matrices by $\mathcal{COP}$ and its interior, the set of strictly copositive matrices, by int$\mathcal{COP}$. The boundary of $\mathcal{COP}$ is denoted by bd$\mathcal{COP} = \mathcal{COP} \setminus \text{int}\mathcal{COP}$.

The task of testing whether a given matrix is copositive is related to the task of solving a quadratic program:

Lemma 1 *Let $M \in \mathcal{S}$ and let $\bar{x}$ be a (globally) minimal solution of the quadratic optimization problem*

$$\begin{aligned} QP\text{:}\ \ \min\ & f(x) := \tfrac{1}{2}x^\top Mx \\ \text{s.t. }\ & e^\top x = 1 \\ & x \geq 0 \end{aligned} \tag{1}$$

where $e \in \mathbb{R}^n$ denotes the vector with all components equal to one. Then

(a) $M \in \mathcal{COP}$ if and only if $f(\bar{x}) \geq 0$,
(b) $M \in \text{int}\mathcal{COP}$ if and only if $f(\bar{x}) > 0$.
(c) $M \notin \mathcal{COP}$ if and only if there exists a feasible x with $f(x) < 0$.
(d) $M \notin \text{int}\mathcal{COP}$ if there exists a feasible x with $f(x) = 0$.

Note that QP (1) is solvable as the feasible set is nonempty, compact and the objective function is continuous. As the feasible set is given by linear constraints, any minimal solution of QP (1) satisfies the KKT-conditions, i.e. there exists $\lambda \in \mathbb{R}$ and $w \in \mathbb{R}^n$ such that

$$\begin{aligned} Mx &= \lambda e + w \\ x &\geq 0,\ w \geq 0 \\ x^\top w &= 0,\ e^\top x = 1 \end{aligned}$$

We write $x \in K$ if there exists some $w \in \mathbb{R}^n$ and some $\lambda \in \mathbb{R}$ such that (x, λ, w) satisfies the above conditions. For any $x \in K$ it holds

$$f(x) = \frac{1}{2}x^\top Mx = \frac{1}{2}(\lambda e^\top x + w^\top x) = \frac{\lambda}{2}\ .$$

Therefore we have $\lambda = x^\top Mx$. Hence we can consider the following Mathematical Program with Linear Complementarity Constraints:

$$\begin{array}{lll} \text{MPLCC:} & \min & \frac{1}{2}\lambda \\ & \text{s.t.} & w = Mx - \lambda e \\ & & x \geq 0,\ w \geq 0 \\ & & x^\top w = 0,\ e^\top x = 1 \\ & & \lambda \in \mathbb{R} \end{array} \tag{2}$$

If $\bar{x}$ is a minimal solution of QP (1), then there exists some $\bar{\lambda} \in \mathbb{R}$ and some $\bar{w} \in \mathbb{R}^n$ such that $(\bar{x}, \bar{\lambda}, \bar{w})$ is a feasible solution of MPLCC (2) with the same objective function value. On the other hand, any feasible point $(\bar{x}, \bar{\lambda}, \bar{w})$ of MPLCC (2) gives a feasible solution of QP (1) with the same objective function value. Hence both problems are equivalent in the sense that they have the same objective function value and a minimal solution of one problem directly gives a minimal solution of the other problem. We conclude from Lemma 1:

Corollary 1 *(a) $M \in \mathcal{COP}$ if and only if MPLCC (2) has a (globally) minimal solution $(\bar{x}, \bar{\lambda}, \bar{w})$ with $\bar{\lambda} \geq 0$.*
(b) $M \in int\mathcal{COP}$ if and only if MPLCC (2) has a (globally) minimal solution $(\bar{x}, \bar{\lambda}, \bar{w})$ with $\bar{\lambda} > 0$.

2.2 Conditions for copositivity based on LCPs

For solving the MPLCC (2) we study its feasible set which contains a linear complementarity problem (LCP). The general form of an LCP is given as follows:

$$\begin{array}{l} \text{LCP: Find } x \in \mathbb{R}^n \text{ and } w \in \mathbb{R}^n \text{ such that} \\ \qquad w = q + Mx \\ \qquad x \geq 0,\ w \geq 0 \\ \qquad x^\top w = 0 \end{array} \tag{3}$$

where $M \in \mathcal{S}$ and $q \in \mathbb{R}^n$ are given. We also use the notation LCP(q, M) for representing a LCP with a given vector q and matrix M. We call a pair $(x, w) \in \mathbb{R}^n \times \mathbb{R}^n$ *feasible* for LCP(q, M) if

$$w = q + Mx,\ x \geq 0 \text{ and } w \geq 0.$$

We say that a pair $(x, w) \in \mathbb{R}^n \times \mathbb{R}^n$ *satisfies* LCP(q, M) if it is *feasible* for LCP(q, M) and if additionally $x^\top w = 0$ holds. In the latter case x is called a *solution* of the LCP(q, M).

A point $(x, \lambda, w) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n$ is feasible for the MPLCC (2) if and only if $e^\top x = 1$ and (x, w) satisfies LCP($-\lambda e, M$). Hence, a feasible solution of MPLCC (2) can be found by first determining a solution $\bar{x} \neq 0$ of LCP($-\lambda e, M$) and then by setting

$$\tilde{x} := \frac{1}{e^\top \bar{x}} \bar{x}\ .$$

Therefore, it is enough to study the problem LCP($-\lambda e, M$) for $\lambda = 0$, a positive and a negative λ to cover all cases of $\lambda \in \mathbb{R}$ due to the following result:

Lemma 2 *Let $\lambda_1 \in \mathbb{R}$ and $\lambda_2 \in \mathbb{R}$ be given. If $\lambda_1 \cdot \lambda_2 > 0$, then the following holds: LCP($-\lambda_1 e, M$) has a solution $\bar{x} \neq 0$ if and only if LCP($-\lambda_2 e, M$) has a solution*

$$\frac{\lambda_2}{\lambda_1}\bar{x} \neq 0 \, .$$

Proof If LCP($-\lambda_1 e, M$) has a solution $\bar{x} \neq 0$, then let $J := \{i \in \{1,\ldots,n\} \mid \bar{x}_i > 0\}$ and $L := \{1,\ldots,n\} \setminus J$. Due to complementarity we have

$$\begin{array}{ll} 0 = -\lambda_1 e_J + M_{JJ}\bar{x}_J, & 0 \leq -\lambda_1 e_L + M_{LJ}\bar{x}_J, \\ \bar{x}_J > 0, & \bar{x}_L = 0. \end{array}$$

Multiplying by λ_2/λ_1 yields that $\frac{\lambda_2}{\lambda_1}\bar{x}$ is a solution of LCP($-\lambda_2 e, M$). □

Next we study LCP($-\lambda e, M$) for $\lambda = 0$ and $\lambda = -1$ and its solutions. With the help of Corollary 1 we derive sufficient conditions for the (strict) copositivity of M as well as for M not being (strictly) copositive. The existence of nonzero solutions of LCP($-\lambda e, M$) for $\lambda > 0$ does not imply the strict copositivity of the matrix. It has to be guaranteed that there are no nonzero solutions for $\lambda \leq 0$.

Corollary 2 *(a) If LCP($0, M$) has a solution $\bar{x} \neq 0$, then $M \notin \text{int}\mathcal{COP}$.*
(b) If LCP(e, M) has a solution $\bar{x} \neq 0$, then $M \notin \mathcal{COP}$ and thus also $M \notin \text{int}\mathcal{COP}$.
(c) If no nonzero solution of LCP(e, M) and of LCP($0, M$) exists, then $M \in \text{int}\mathcal{COP}$ and thus $M \in \mathcal{COP}$.
(d) If no nonzero solution of LCP(e, M) exists, then $M \in \mathcal{COP}$.

Example 1 For the matrix

$$M = \begin{pmatrix} 1 & -2 \\ -2 & 1 \end{pmatrix}$$

the point $\bar{x} = (1\ 1)^\top$ with $\bar{w} = (0\ 0)$ is a nonzero solution of LCP(e, M) and hence $M \notin \mathcal{COP}$ (and $M \notin \text{int}\mathcal{COP}$) by Corollary 2.

The solvability of the LCP for any q gives another necessary condition for strict copositivity.

Lemma 3 *[11, Theorem 3.8.5] If $M \in \text{int}\mathcal{COP}$, then for each $q \in \mathbb{R}^n$ the problem LCP(q, M) has a solution.*

A solution can be found by Lemke's method, see for instance [11, Chapter 4.4], which is a simplex-like vertex following algorithm that uses basic feasible solutions of a system of the form

$$w = q + \xi d + Mx,\ x \geq 0,\ \xi \geq 0,\ w \geq 0 \tag{4}$$

where d is a positive vector (note that $x^\top w = 0$ in each iteration of the method). This method is guaranteed to terminate in a finite number of iterations if all the basic feasible solutions of the system (4) are nondegenerate. We suggest [11] for a discussion of the steps, convergence and complexity of Lemke's method. The procedure either finds a solution of the LCP or it terminates in an unbounded ray. For some classes of matrices, including the strictly copositive matrices, this latter termination can not occur and Lemke's method always terminates with a solution of the LCP, cf. [11, Theorem 4.4.9]. We derive the following necessary condition for a matrix to be strictly copositive which we use in Step 1 of our algorithm:

Corollary 3 *If Lemke's method applied to LCP(q,M) for some $q \in \mathbb{R}^n$ terminates in an unbounded ray, then $M \notin$ int$\mathcal{COP}$.*

If M is only a copositive matrix which is not strictly copositive, then LCP(q,M) does not need to have a solution. For instance, this is the case for M the zero matrix and q negative.

Example 2 For the matrix M of Example 1 Lemke's method applied to LCP($-e,M$) (i.e. $\lambda = 1$ in LCP($-\lambda e, M$)) terminates in an unbounded ray. Hence, $M \notin$ int$\mathcal{COP}$.

The drawback of the sufficient conditions for (strict) copositivity of Corollary 2 is that it has to be guaranteed that there is no nonzero solution of these LCPs. This is much more difficult than computing a solution for these problems. This indicates why in practice establishing (strict) copositivity of a matrix is more difficult than showing that a matrix is not (strictly) copositive.

Finally, we study a different LCP which gives a certificate for $M \notin$ int$\mathcal{COP}$ and $M \notin \mathcal{COP}$. For that, let

$$p := \begin{pmatrix} 0 \\ -1 \end{pmatrix} \in \mathbb{R}^{n+1} \text{ and } Q := \begin{pmatrix} M & e \\ e^\top & 0 \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)} \tag{5}$$

where 0 in the vector p is a n-vector and 0 in the matrix Q is the real number zero. Then LCP(p,Q) is equivalent to finding a pair $(x,\mu) \in \mathbb{R}^{n+1}$ such that there exists some $(w,\eta) \in \mathbb{R}^{n+1}$ with

$$\begin{aligned} \begin{pmatrix} w \\ \eta \end{pmatrix} &= \begin{pmatrix} 0 \\ -1 \end{pmatrix} + \begin{pmatrix} M & e \\ e^\top & 0 \end{pmatrix} \begin{pmatrix} x \\ \mu \end{pmatrix} \\ \begin{pmatrix} x \\ \mu \end{pmatrix} &\geq 0, \begin{pmatrix} w \\ \eta \end{pmatrix} \geq 0 \\ \begin{pmatrix} x \\ \mu \end{pmatrix}^\top \begin{pmatrix} w \\ \eta \end{pmatrix} &= 0\,. \end{aligned} \tag{6}$$

The following result holds:

Lemma 4 *(a) LCP(p,Q) has a solution (x,μ) if and only if $M \notin$ int$\mathcal{COP}$.*
(b) LCP(p,Q) has a solution (x,μ) with $\mu > 0$ if and only if $M \notin \mathcal{COP}$.

Proof (a) If LCP(p,Q) has a solution, then either $\eta = 0$ and $\mu \geq 0$ or $\eta > 0$ and $\mu = 0$. In the first case we immediately obtain that there is a feasible point $(x,\lambda,w) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n$ for MPLCC (2) with $\lambda = -\mu \leq 0$ and hence, by Corollary 1, $M \notin$ int$\mathcal{COP}$. In the second case, there exists some $x \geq 0$ with $x \neq 0$ and $0 = x^\top w = x^\top Mx$ which implies $M \notin$ int$\mathcal{COP}$. On the other hand, if $M \notin$ int$\mathcal{COP}$, then there exists some feasible $(x,\lambda,w) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n$ with $\lambda \leq 0$ for MPLCC (2) which by setting $\eta = 0$ and $\mu = -\lambda \geq 0$ gives a feasible solution for LCP(p,Q).

(b) First let LCP(p,Q) have a solution with $\mu > 0$ and thus $\eta = 0$. Then there is a feasible point $(x,\lambda,w) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n$ for MPLCC (2) with $\lambda = -\mu < 0$ and hence, by Corollary 1, $M \notin \mathcal{COP}$. On the other hand, if $M \notin \mathcal{COP}$, then there exists some feasible $(x,\lambda,w) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n$ with $\lambda < 0$ for MPLCC (2) which by setting $\eta = 0$ and $\mu = -\lambda > 0$ gives a feasible solution for LCP(p,Q). □

Then we can state the following copositivity tests:

Corollary 4 *(a) If LCP(p,Q) has a solution (x,μ) with $\mu = 0$, then $M \notin int\mathcal{COP}$.*
(b) If LCP(p,Q) has a solution (x,μ) with $\mu > 0$, then $M \notin \mathcal{COP}$ and thus also $M \notin int\mathcal{COP}$.
(c) If LCP(p,Q) has no solution, then $M \in int\mathcal{COP}$ and thus $M \in \mathcal{COP}$.
(d) If LCP(p,Q) has no solution (x,μ) with $\mu > 0$, then $M \in \mathcal{COP}$.

Example 3 For the matrix M of Example 1 LCP(p,Q) has a solution (x,μ) with $x = (1/2,1/2)^\top$ and $\mu = 1/2 > 0$ and thus, by Corollary 4, $M \notin \mathcal{COP}$.

In order to use Corollary 4(d) one needs to know all solutions of LCP(p,Q). For obtaining the complete solution set we will apply the enumerative algorithm described in [22]. We also exploit the following reformulation of LCP(p,Q) as a mixed integer linear program (MIP) [21]:

Lemma 5 *The mixed integer linear program*

$$\begin{aligned} MIP_1 : \max\ & \alpha \\ s.t.\ & 0 \le Qy + \alpha p \le z \\ & 0 \le y \le e - z \\ & z \in \{0,1\}^{n+1} \\ & 0 \le \alpha \le 1. \end{aligned}$$

with p and Q as in (5) has a solution, and $(x^*,\mu^*) = \dfrac{y^*}{\alpha^*}$ is a solution of LCP(p,Q), if and only if MIP_1 has a feasible solution (α^*,y^*,z^*) with $\alpha^* > 0$.

Proof This result follows from Proposition 2.5 in [21] by noting that $p \neq 0$ and that it is enough that (α^*,y^*,z^*) is feasible (and not additionally optimal) in the proof given there. □

Note that this MIP is feasible and y_{n+1} corresponds to the μ variable of LCP(p,Q) (with $\mu = \dfrac{1}{\alpha^*}y^*_{n+1}$). Therefore, by Corollary 4, the following result holds.

Corollary 5 *(a) If MIP_1 has a feasible solution (α,y,z) with $\alpha > 0$, then $M \notin int\mathcal{COP}$.*
(b) If MIP_1 has a feasible solution (α,y,z) with $\alpha > 0$ and $y_{n+1} > 0$, then $M \notin \mathcal{COP}$.
(c) MIP_1 has a globally optimal value equal to zero if and only if $M \in int\mathcal{COP}$.
(d) If MIP_1 has no feasible solution (α,y,z) with $\alpha > 0$ and $y_{n+1} > 0$, then $M \in \mathcal{COP}$.

Note that for part (a), in case it holds $\alpha > 0$ and $y_{n+1} > 0$, then (x^*,μ^*) with $\mu^* = \dfrac{1}{\alpha^*}y^*_{n+1} > 0$ is a solution of LCP(p,Q), and by Corollary 4(b) we have $M \notin \text{int}\mathcal{COP}$.

The case of $\alpha > 0$ and $y_{n+1} = 0$ in the computed solution of MIP_1 is the most difficult, as we cannot conclude whether M is copositive or not from the solution of MIP_1. In this case, we may consider a MIP of the form:

$$\begin{aligned} \text{MIP}_2 : \max\ & y_{n+1} \\ \text{s.t. } & 0 \le Qy + \alpha p \le z \\ & 0 \le y \le e - z \\ & z \in \{0,1\}^{n+1} \\ & 0 \le \alpha \le 1 \\ & \alpha \ge \varepsilon, \end{aligned}$$

where ε is a fixed positive tolerance (usually $\varepsilon = 10^{-4}$). Then, by Corollary 5, the following result holds:

Corollary 6 *Let $\varepsilon > 0$. If MIP_2 has a feasible solution (α, y, z) with $y_{n+1} > 0$, then $M \notin \mathcal{COP}$.*

As one can see from Corollary 5 it is easier to show that a matrix $M \notin \text{int}\mathcal{COP}$ than $M \notin \mathcal{COP}$. There is a further possibility for establishing that $M \notin \mathcal{COP}$ by making use of a related matrix that is not strictly copositive:

Lemma 6 *Let $M, H \in \mathcal{S}$ and $H \in int\mathcal{COP}$. Then $M \in \mathcal{COP}$ if and only if $M + \beta H \in int\mathcal{COP}$ for all $\beta > 0$.*

Proof The implication $\Rightarrow$ is obvious from the definitions of a copositive and a strictly copositive matrix. Now, if $M \notin \mathcal{COP}$, then there exists a point $0 \neq \bar{x} \geq 0$ such that $\bar{x}^\top M \bar{x} < 0$. Since $H \in \text{int}\mathcal{COP}$ it holds that

$$\beta = -\frac{\bar{x}^\top M \bar{x}}{\bar{x}^\top H \bar{x}}$$

is positive and satisfies

$$\bar{x}^\top (M + \beta H)\bar{x} = 0.$$

Hence $M + \beta H \notin \text{int}\mathcal{COP}$. □

2.3 Conditions for copositivity based on LPs

In this section, we present additional conditions which are based on the feasibility of LPs. Therefore, we need the following class of matrices:

Definition 2 A real $n \times n$ matrix M is said to be a S_0 matrix ($M \in S_0$) if there exists a point $0 \neq x \geq 0$ such that $Mx \geq 0$.

Thus for a matrix $M \in \mathcal{S}$ it holds

$$-M \in S_0 \;\Leftrightarrow\; \exists x \geq 0,\ x \neq 0 : \; Mx \leq 0.$$

As a consequence we have that $-M \in S_0$ implies $M \notin \text{int}\mathcal{COP}$ and that $-M \in S_0$ holds if and only if the system

$$Mx \leq 0,\ x \geq 0,\ e^\top x = 1$$

has a solution. Hence we can state the following result:

Lemma 7 *Let $c \in \mathbb{R}^n_+$. If*

$$LP_1 : \ \inf\{c^\top x \mid Mx \le 0,\ x \ge 0,\ e^\top x = 1\}$$

is feasible, then $M \notin int\mathcal{COP}$.

The problem LP_1 is bounded from below as $c \in \mathbb{R}^n_+$. Thus it has a minimal solution if and only if it is feasible.

For our next result we use the implication

$$M \notin S_0 \Rightarrow M \notin \mathcal{COP}$$

given in [11]. Note that $M \notin S_0$ if and only if the system

$$Mx \ge 0,\ x \ge 0,\ e^\top x = 1$$

has no solution. Therefore the following result holds:

Lemma 8 *Let $c \in \mathbb{R}^n_+$. If*

$$LP_2 : \inf\{c^\top x \mid Mx \ge 0,\ x \ge 0,\ e^\top x = 1\}$$

has no feasible solution, then $M \notin \mathcal{COP}$ and thus also $M \notin int\mathcal{COP}$.

Lemmas 7 and 8 give easy to verify sufficient conditions for a matrix not to be (strictly) copositive.

Example 4 For the matrix M of Example 1 the system

$$Mx \le 0, \qquad x \ge 0, \qquad e^\top x = 1$$

has the solution $(1/2, 1/2)$. Thus, by Lemma 7, $M \notin \mathrm{int}\mathcal{COP}$.

The system

$$\begin{array}{c} Mx \ge 0 \\ x \ge 0 \\ e^\top x = 1 \end{array} \Leftrightarrow \begin{cases} x_1 \ge 2x_2 \\ x_2 \ge 2x_1 \\ x_1 + x_2 = 1 \\ x_i \ge 0,\ i = 1,2 \end{cases}$$

has no feasible solution. By Lemma 8 we have $M \notin \mathcal{COP}$.

3 Algorithm

Below we collect the results of the previous sections and give an algorithm for testing whether a given matrix is (strictly) copositive. In the algorithm we also combine some preprocessing steps based on the conditions given in Section 2.3 with well known results from the literature.

3.1 Preprocessing

In addition to the results of Lemma 7 and 8 we will make use of the following preprocessing steps which are based on the collection in [3], see also [31].

Lemma 9 *Let $M = [m_{ij}] \in \mathcal{S}$ and choose an arbitrary $i \in \{1,\ldots,n\}$.*

(a) If $m_{ii} < 0$, then $M \notin \mathcal{COP}$.
(b) if $m_{ii} = 0$, then $M \notin \text{int}\mathcal{COP}$.
(c) if $m_{ii} = 0 > m_{ij}$ for some $j \in \{1,\ldots,n\}$, then $M \notin \mathcal{COP}$.

The preprocessing steps used later in our main algorithm are summarized in Algorithm 1. We used $c = e$ in the LPs.

Algorithm 1 Preprocessing

Input: matrix $M \in \mathcal{S}$
  (Part (1):)
  if $m_{ii} < 0$ for any $i \in \{1,\ldots,n\}$ **then**
    $M \notin \mathcal{COP}$ and stop.
  end if
  if $m_{ii} = 0$ for any $i \in \{1,\ldots,n\}$ **then**
    $M \notin \text{int}\mathcal{COP}$.
  end if
  if $m_{ii} = 0 > m_{ij}$ for any $i \neq j$, $i, j \in \{1,\ldots,n\}$ **then**
    $M \notin \mathcal{COP}$ and stop.
  end if
  Let $c = e$.
  (Part (2):)
  if LP_1 has a feasible solution **then**
    $M \notin \text{int}\mathcal{COP}$.
  end if
  (Part (3):)
  if LP_2 has no feasible solution **then**
    $M \notin \mathcal{COP}$ and stop.
  end if
Output: $M \notin C$ or $M \notin \text{int}\mathcal{COP}$ or preprocessing not conclusive.

3.2 Outline of the algorithm to test (strict) copositivity

Algorithm 2 gives the structure of our main algorithm. In Step 1 we make use of Corollary 3. For Step 2 we propose three different procedures in this section. Note that it depends on the choice of the procedure in Step 2 whether Algorithm 2 is guaranteed to find $M \notin \mathcal{COP}$, $M \in \mathcal{COP}$ or $M \in \text{int}\mathcal{COP}$.

3.3 Procedures for Step 2

As discussed in Section 2.2, we can solve LCP(p,Q) to derive proofs for the matrix to be in $\mathcal{COP}$, in int$\mathcal{COP}$ or not copositive. We may apply an enumerative algorithm,

Algorithm 2 Test on (strict) copositivity

Input: Matrix $M \in \mathcal{S}$
STEP 0: Apply Algorithm 1 (Preprocessing).
if STEP 0 was not conclusive **then**
  STEP 1: Use Lemke's method with different initial complementary basic solutions to solve LCP$(-e, M)$.
  if Lemke's method terminates in an unbounded ray **then**
    $M \notin \text{int}\mathcal{COP}$.
  end if
  STEP 2: Apply Procedures 1-3 to be discussed in Section 3.3.
end if
Output: $M \notin C$ or $M \in C$ or $M \in \text{int}\mathcal{COP}$ (or not conclusive depending on the choice in Step 2).

Lemke's method and the mixed integer linear formulation for solving this LCP. Each of the following three procedures are discussed next and can be used in Step 2 of Algorithm 2.

Procedure 1: Solve LCP(p, Q) by an enumerative algorithm and apply Corollary 4.

An efficient enumerative method for the Linear Complementarity Problem (LCP) has been proposed by Júdice, Faustino and Ribeiro [22]. This method finds a solution of the LCP by exploring a binary tree generated by the dichotomy $x_i = 0$ or $w_i = 0$ associated with the complementary condition, see Fig. 1.

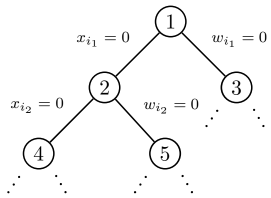

Fig. 1 Branching procedure of the enumerative method.

In each node of the tree, the algorithm finds a stationary point of a nonconvex quadratic program of the form

$$\begin{array}{ll} \min & f(x,w,\mu,\eta) = x^\top w + \mu\,\eta \\ \text{s.t.} & w = Mx + \mu e \\ & \eta = e^\top x - 1 \\ & x_i = 0,\ i \in I \\ & w_j = 0,\ j \in J \\ & x,\, w,\, \mu,\, \eta \geq 0, \end{array} \tag{7}$$

where I and J are the index sets defined by the fixed variables, i.e. $I := \{i \in \{1,\ldots,n\}\ :\ x_i = 0\ \text{ fixed}\}$ (and also additionally $\mu = 0$ may be fixed) and $J := \{i \in \{1,\ldots,n\}\ :\ w_i = 0\ \text{fixed}\}$ (and also additionally $\eta = 0$ may be fixed), respectively, in the path of the tree from this node to the root. Furthermore the algorithm contains some heuristic rules for choosing the node and the pair of complementary variables for branching. For details on the algorithm we refer to [22].

It is important to note that it is much easier to compute a solution for an LCP (when it exists) than showing that an LCP has no solution. So, as expected it is much easier to show that a matrix is not copositive than proving that it has this property. However, the algorithm can at least in theory test copositivity of any given matrix.

Procedure 2: Solve LCP(p,Q) with Lemke's algorithm and $n+1$ initial complementary basic solutions and apply Corollary 4.

Lemke's method is a pivotal algorithm that aims at solving LCP(p,Q) by using basic feasible solutions (BFS) of the following general LCP (GLCP)

$$\begin{array}{l} v = p + \xi\, d + Qu \\ u \geq 0,\, v \geq 0,\, \xi \geq 0 \\ u^\top v = 0 \end{array} \tag{8}$$

where in our setting $v = [w\ \eta]^\top$, $u = [x\ \mu]^\top$ and $d \in \mathbb{R}^{n+1}$ is a positive vector (usually $d = e$). Moreover, p and Q are given by (5). Given a basic solution with basic variables $v_i = w_i$, $i \in \{1,\ldots,n\}$ and $v_{n+1} = \eta$ and nonbasic variables $x_i = 0$, $i \in \{1,\ldots,n\}$ and $\mu = 0$ an initial BFS of GLCP (8) is obtained as follows:

$$\bar{u} = 0,\ \bar{\xi} = -\frac{p_r}{d_r},\ v = p + \bar{\xi} d, \tag{9}$$

where:

$$-\frac{p_r}{d_r} := \max\{-\frac{p_i}{d_i} : i = 1,\ldots,n+1\} > 0. \tag{10}$$

All the variables u_i and the variable v_r are nonbasic and the remaining variables are basic. Hence there exists exactly one complementary pair (v_r, u_r) of nonbasic variables. In the next iteration the algorithm chooses u_r (complementary to the previous leaving variable) as the entering variable which interchanges with a leaving basic variable (found by the common minimum quotient rule) [11]. If such a leaving variable does not exist, the algorithm stops in an unbounded ray. Otherwise, a new BFS of GLCP (8) is obtained and either $\xi = 0$ and a solution of the LCP is at hand or the procedure is repeated.

Lemke's method can start with any basic feasible solution of GLCP (8). A possible choice is to consider the vector x equal to one of the canonical basis vectors e^i instead of the null vector in the initial basic solution of GLCP (8). Therefore the basic variables of this initial basic solution are the variable x_i, the variable μ and the variables w_j, $j \in \{1,\ldots,n\}\backslash\{i\}$ of LCP(p,Q). In this way we can construct $(n+1)$ initial basic solutions (including the trivial one with $x = 0$) for initializing Lemke's method. Note that for computing initial BFS of GLCP (8) from each one of these basic solutions, the vector d should be given by $d = Be$, where B is the associated basis matrix.

An obvious drawback of this approach is that the process may not be conclusive.

Procedure 3: Solve MIP$_1$ and MIP$_2$ and apply Corollaries 5 and 6.

In procedure 3 we aim again at solving the problem LCP(p,Q) but this time by using its reformulation as a MIP. We start by trying to find a feasible solution with $\alpha > 0$ of MIP$_1$ to be able to apply Corollary 5. If MIP$_1$ has a feasible solution with $\alpha > 0$ and $y_{n+1} = 0$, then we want to find a feasible solution of MIP$_2$ to be able to apply Corollary 6. If the globally optimal value of MIP$_1$ is equal to zero, then $M \in \mathcal{COP}$ (and even $M \in \mathrm{int}\mathcal{COP}$). This discussion confirms that it is much easier to show that a matrix is not copositive (or not strictly copositive) than establishing that it is copositive.

4 Numerical results

We test the basic steps (Step 0 and 1) of our algorithm as well as all three procedures on several test instances from the literature — like the famous Horn matrix. We also consider generated Max-clique instances and instances from the DIMACS collection [14] with matrices up to order 496×496. We compare the results with those from the literature on copositivity tests by Bundfuss and Dür [8], Žilinskas and Dür [33], and Bomze and Eichfelder [3]. Moreover, as testing copositivity is equivalent to determining a globally optimal solution of the quadratic optimization problem (1), see Lemma 1, we apply the global optimization solver BARON (with default parameters settings) and compare with our results.

All experiments have been performed on a Pentium IV (Intel) with 3.0 GHz and 2 GBytes of RAM memory, using the operating system Linux. The algorithm was implemented in the General Algebraic Modeling System (GAMS) language (Rev 118 Linux/Intel) [6] and the solvers CPLEX [12] (version 9.1), MINOS [25] (version 5.51) and BARON [28] (version 22.7.2) were used to solve the MIP and the nonlinear optimization problems, and the LP was solved with CPLEX. Lemke's method has been implemented in MATLAB [24] environment (version 7.11, R2010b). The running times which we present in these sections are always given in CPU seconds (a value of zero for CPU means that the CPU time is less than one second). The maximum CPU time allowed for all procedures is 7200 seconds.

4.1 Test matrices

The following non-copositive matrices $M_1 \notin \mathcal{COP}$ and $M_2 \notin \mathcal{COP}$ are taken from Bomze and Eichfelder [3] and Kaplan [23], respectively:

$$M_1 = \begin{pmatrix} 1 & -0.72 & -0.59 & 1 \\ -0.72 & 1 & -0.6 & -0.46 \\ -0.59 & -0.6 & 1 & -0.6 \\ 1 & -0.46 & -0.6 & 1 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 1 & -0.72 & -0.59 & -0.6 \\ -0.72 & 1 & 0.21 & -0.46 \\ -0.59 & 0.21 & 1 & -0.6 \\ -0.6 & -0.46 & -0.6 & 1 \end{pmatrix}.$$

The matrix $M_3 \in \mathrm{int}\mathcal{COP}$ is from Kaplan [23] and $M_4 \in \mathrm{int}\mathcal{COP}$ is a principal submatrix of a matrix from Kaplan [23, Ex. 1]:

$$M_3 = \begin{pmatrix} 1 & 0.9 & -0.54 & 0.21 \\ 0.9 & 1 & -0.03 & 0.78 \\ -0.54 & -0.03 & 1 & 0.52 \\ 0.21 & 0.78 & 0.52 & 1 \end{pmatrix}, \qquad M_4 = \begin{pmatrix} 1 & 0.9 & -0.54 \\ 0.9 & 1 & -0.03 \\ -0.54 & -0.03 & 1 \end{pmatrix}.$$

The matrix $M_5 \in \mathrm{bd}\mathcal{COP}$ is from Väliaho [32] and the famous Horn matrix M_6 is also an example with $M_6 \in \mathrm{bd}\mathcal{COP}$ [18]:

$$M_5 = \begin{pmatrix} 1 & -1 & 1 & 2 & -3 \\ -1 & 2 & -3 & -3 & 4 \\ 1 & -3 & 5 & 6 & -4 \\ 2 & -3 & 6 & 5 & -8 \\ -3 & 4 & -4 & -8 & 16 \end{pmatrix}, \qquad M_6 = \begin{pmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{pmatrix}$$

The Hoffman-Pereira matrix $M_7 \in \mathrm{bd}\mathcal{COP}$ [20] is another example of this type:

$$M_7 = \begin{pmatrix} 1 & -1 & 1 & 0 & 0 & 1 & -1 \\ -1 & 1 & -1 & 1 & 0 & 0 & 1 \\ 1 & -1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 1 & -1 & 1 & -1 \\ -1 & 1 & 0 & 0 & 1 & -1 & 1 \end{pmatrix}.$$

Thus $M_1, M_2 \notin \mathcal{COP}$, $M_3, M_4 \in \mathrm{int}\mathcal{COP}$ and $M_5, M_6, M_7 \in \mathrm{bd}\mathcal{COP}$.

4.2 Maximum clique instances

Next to the above test matrices we also used test instances derived from the maximum clique problem (based on a reformulation as a linear optimization problem over the cone of copositive matrices) from the DIMACS collection [14], cf. [3,8,33], and generated instances (cf. [33]). For a simple, i.e. loopless and undirected, graph $G = (V,E)$ with node set $V = \{1,\dots,n\}$ and edge set E, a clique C is a subset of V such that every pair of nodes in C is connected by an edge in E. A clique C is said to be a maximum clique if it is a clique of maximum cardinality and its size $\omega(G)$ is called the (maximum) clique number. Finding the clique number can be reformulated as a copositive optimization problem

$$\omega(G) = \min\{\lambda \in \mathbb{N} \mid \lambda(E_n - A_G) - E_n \text{ is copositive}\} \tag{11}$$

with E_n the $n \times n$ all-ones matrix and $A_G = [a_{ij}]_{i,j}$ the adjacency matrix of the graph G, i.e. $a_{ij} = 1$ if $\{i,j\} \in E$, and $a_{ij} = 0$ else, $i,j \in \{1,\dots,n\}$. According to [29, Prop. 3.2] it holds

$$\lambda(E_n - A_G) - E_n \begin{cases} \in \mathrm{int}\mathcal{COP} & \text{if } \lambda > \omega(G) \\ \in \mathrm{bd}\mathcal{COP} & \text{if } \lambda = \omega(G) \\ \notin \mathcal{COP} & \text{if } \lambda < \omega(G). \end{cases}$$

Thus if $\lambda(E_n - A_G) - E_n \notin \mathcal{COP}$, we can conclude that $\omega(G) \geq \lambda + 1$. In Table 1 we list the characteristics of the graphs from the DIMACS [14] collection and the generated graphs (cf. [33]). The number n of nodes gives the order $n \times n$ of the examined matrices. The small instances (with $n \in [14, 22]$) as well as eight large instances of Table 1 were also tested in [3]. In all test problems we used $\lambda = \omega(G) - 1$.

Table 1 Generated small instances [33] and large instances from DIMACS collection [14].

Matrix	n	$\|E\|$	$\omega(G)$
c-fat14-1	14	52	6
Brock14	14	51	5
Brock16	16	59	5
Brock18	18	78	5
Brock20	20	95	5
Morgen14	14	50	5
Morgen16	16	59	5
Morgen18	18	60	5
Morgen20	20	67	5
Morgen22	22	68	5
Johnson6-2-4	15	45	3
Johnson6-4-4	15	45	3
Johnson7-2-4	21	105	3
Jagota14	14	31	6
Jagota16	16	57	8
Jagota18	18	84	10
sanchis14	14	50	5
sanchis16	16	50	5
sanchis18	18	50	5
sanchis20	20	50	5
sanchis22	22	50	5

Matrix	n	$\|E\|$	$\omega(G)$
Brock200-1	200	14834	21
Brock200-2	200	9876	12
Brock200-3	200	12048	15
Brock200-4	200	13089	17
c-fat200-1	200	1534	12
c-fat200-2	200	3235	24
c-fat200-5	200	8473	58
Hamming6-2	64	1824	32
Hamming6-4	64	704	4
Hamming8-2	256	31616	128
Hamming8-4	256	20864	16
Johnson8-2-4	28	210	4
Johnson8-4-4	70	1855	14
Johnson16-2-4	120	5460	8
Johnson32-2-4	496	107880	16
Keller4	171	9435	11
Mann-a9	45	918	16
Mann-a27	378	70551	126

4.3 Numerical Experiments: Steps 0 and 1

In Tables 2, 3 and 4 we report the performance of Steps 0 and 1 of the algorithm where It and Time denote respectively, the number of LP iterations and the time of execution required by Lemke's algorithm.
We have run all steps of the algorithm even in cases where one of the previous steps was conclusive. The numbers (1), (2) and (3) in Step 0 refer to the Parts as marked in Algorithm 1.

Furthermore in Step 1, for instances where Lemke's method was inconclusive for the initial trivial basis ($x_i = 0$, for all $i = 1, \ldots, n$), i.e., Lemke's method found a solution, we applied the method with different initial basic solutions as described in Procedure 2 for Step 2. The process is repeated until the method terminates in an unbounded ray (Case: unb. ray) or finds a solution of the LCP in each one of the $(n+1)$ applications of the method (Case: sol.).

The numerical results indicate that the preprocessing phase does not help much for these instances. Actually there are only two cases were the preprocessing was effective. Contrary to this, the use of Lemke's method in Step 1 seems quite promising for establishing that a matrix is not strictly copositive. Furthermore, in many cases

Table 2 Performance of the algorithm for the matrices of Section 4.1 (Steps 0 and 1).

Matrix	Step 0			Step 1					
	(1)	(2)	(3)	Trivial Basis		Other Basis		Unb. Ray	Sol.
	$M \notin \mathcal{COP}$	$M \notin \text{int}\mathcal{COP}$	$M \notin \mathcal{COP}$	It	Time	It	Time	$M \notin \text{int}\mathcal{COP}$	
M_1	-	✓	✓	3	0			✓	
M_2	-	✓	✓	4	0			✓	
M_3	-	-	-	2	0	8	0		✓
M_4	-	-	-	2	0	5	0		✓
M_5	-	-	-	3	0			✓	
M_6	-	-	-	2	0			✓	
M_7	-	-	-	2	0			✓	

Table 3 Performance of the algorithm for small matrices $\lambda(E_n - A_G) - E_n$, with $\lambda = \omega(G) - 1$ (Steps 0 and 1).

Matrix	Step 0			Step 1					
	(1)	(2)	(3)	Trivial Basis		Other Basis		Unb. Ray	Sol.
	$M \notin \mathcal{COP}$	$M \notin \text{int}\mathcal{COP}$	$M \notin \mathcal{COP}$	It	Time	It	Time	$M \notin \text{int}\mathcal{COP}$	
c-fat14-1	-	-	-	6	0			✓	
Brock14	-	-	-	4	0			✓	
Brock16	-	-	-	3	0	9	0	✓	
Brock18	-	-	-	4	0			✓	
Brock20	-	-	-	4	0			✓	
Morgen14	-	-	-	5	0			✓	
Morgen16	-	-	-	4	0			✓	
Morgen18	-	-	-	2	0	15	0	✓	
Morgen20	-	-	-	4	0			✓	
Morgen22	-	-	-	4	0			✓	
Johnson6-2-4	-	-	-	3	0			✓	
Johnson6-4-4	-	-	-	3	0			✓	
Johnson7-2-4	-	-	-	3	0			✓	
Jagota14	-	-	-	4	0	11	0	✓	
Jagota16	-	-	-	5	0	15	0	✓	
Jagota18	-	-	-	6	0	19	0	✓	
sanchis14	-	-	-	5	0			✓	
sanchis16	-	-	-	2	0	26	0	✓	
sanchis18	-	-	-	5	0			✓	
sanchis20	-	-	-	5	0			✓	
sanchis22	-	-	-	3	0	4	0	✓	

it was enough to apply Lemke's algorithm only once with the trivial basic feasible solution of GLCP (8). The computational effort of Lemke's algorithm in this last case is quite small. So, as a final conclusion of this first study, Steps 0 and 1 should be included in a more elaborated algorithm to verify whether a given matrix is strictly copositive or copositive or not copositive.

4.4 Numerical Experiments: Step 2

4.5 Procedure 1

In Tables 5, 6 and 7 we report the performance of the enumerative algorithm to solve LCP(p,Q) in order to detect copositivity or non-copositivity of the matrices. If the algorithm finds a solution, then it gives an indication of just $M \notin \text{int}\mathcal{COP}$ or $M \notin \mathcal{COP}$ depending on the value of μ ($\mu = 0$ or $\mu > 0$, respectively). If a complementary feasible solution is found with $\mu > 0$, the algorithm terminates with a certificate that M

Table 4 Performance of the algorithm for large matrices $\lambda(E_n - A_G) - E_n$, with $\lambda = \omega(G) - 1$ (Steps 0 and 1).

MATRIX	STEP 0			STEP 1					
	(1)	(2)	(3)	TRIVIAL BASIS		OTHER BASIS		UNB. RAY	SOL.
	$M \notin \mathcal{COP}$	$M \notin \text{int}\mathcal{COP}$	$M \notin \mathcal{COP}$	IT	TIME	IT	TIME	$M \notin \text{int}\mathcal{COP}$	
Brock200-1	-	-	-	131	0	2589	5.72E+00		✓
Brock200-2	-	-	-	7	0	1246	3.05E+00		✓
Brock200-3	-	-	-	11	0	1881	4.23E+00		✓
Brock200-4	-	-	-	12	0	2062	5.14E+00		✓
c-fat200-1	-	-	-	12	0			✓	
c-fat200-2	-	-	-	24	0			✓	
c-fat200-5	-	-	-	58	0			✓	
Hamming6-2	-	-	-	32	0			✓	
Hamming6-4	-	-	-	4	0			✓	
Hamming8-2	-	-	-	128	0			✓	
Hamming8-4	-	-	-	16	0			✓	
Johnson8-2-4	-	-	-	3	0			✓	
Johnson8-4-4	-	-	-	14	0			✓	
Johnson16-2-4	-	-	-	8	0			✓	
Johnson32-2-4	-	-	-	16	0			✓	
Keller4	-	-	-	7	0	1131	2.39E+00		✓
Mann-a9	-	-	-	9	0	468	0		✓
Mann-a27	-	-	-	27	0	14040	1.27E+02		✓

is not copositive. If it finds a complementary feasible solution with $\mu = 0$, then M is not strictly copositive. In this latter case the algorithm continues until showing that LCP(p,Q) has no solution with $\mu > 0$ (and $M \in \mathcal{COP}$) or finding a complementary feasible solution with $\mu > 0$ (and $M \notin \mathcal{COP}$). If LCP(p,Q) has no solution, then $M \in \text{int}\mathcal{COP}$ (and $M \in \mathcal{COP}$). A maximum CPU time of 7200 seconds is allowed for the algorithm. In the following tables the notation NODES, IT and TIME stands respectively, for the total number of nodes, iterations and time used by the enumerative algorithm. We marked with (-) and (*) respectively, the instances for which a complementary solution does not exist ($M \in \text{int}\mathcal{COP}$) and those for which the algorithm was not able to find such a solution within the CPU time allowed.

Table 5 (Procedure 1) Performance of the enumerative algorithm to solve LCP(p,Q) for matrices of Section 4.1.

MATRIX	NODES	IT	TIME	μ	$M \in \text{int}\mathcal{COP}$	$M \notin \text{int}\mathcal{COP}$	$M \notin \mathcal{COP}$
M_1	1	4	0	9.19E-02			✓
M_2	1	5	0	1.16E-01			✓
M_3	14	19	1.74E+00	-	✓		
M_4	12	16	1.42E+00	-	✓		
M_5	1	8	0	0.00E+00		✓	
M_6	1	2	0	0.00E+00		✓	
M_7	1	2	0	0.00E+00		✓	

The numerical results indicate that the enumerative method was in general efficient to show that the matrix is at least not strictly copositive. However, there were three instances where the method was unable to terminate within the maximum time allowed. Furthermore, the algorithm required a small amount of effort for the smallest instances (Tables 5 and 6) but this effort increases very much when the order of the matrices increases.

Table 6 (Procedure 1) Performance of the enumerative algorithm to solve LCP(p,Q) for small matrices $\lambda(E_n - A_G) - E_n$, with $\lambda = \omega(G) - 1$.

MATRIX	NODES	IT	TIME	μ	$M \notin \text{int}\mathcal{COP}$	$M \notin \mathcal{COP}$
c-fat14-1	3	15	0	1.67E-01		✓
Brock14	1	18	0	0.00E+00	✓	
Brock16	10	52	1.02E+00	0.00E+00	✓	
Brock18	1	8	0	0.00E+00	✓	
Brock20	1	8	0	0.00E+00	✓	
Morgen14	1	12	0	2.00E-01		✓
Morgen16	1	13	0	0.00E+00	✓	
Morgen18	32	98	7.96E+00	0.00E+00	✓	
Morgen20	1	16	0	2.00E-01		✓
Morgen22	1	15	0	0.00E+00	✓	
Johnson6-2-4	1	19	0	0.00E+00	✓	
Johnson6-4-4	1	19	0	0.00E+00	✓	
Johnson7-2-4	1	19	0	2.86E-02		✓
Jagota14	3	33	0	1.67E-01		✓
Jagota16	13	103	3.05E+00	1.25E-01		✓
Jagota18	23	236	8.36E+00	1.00E-01		✓
sanchis14	12	69	1.40E+00	0.00E+00	✓	
sanchis16	10	28	1.02E+00	0.00E+00	✓	
sanchis18	1	8	0	2.00E-01		✓
sanchis20	1	9	0	2.00E-01		✓
sanchis22	80	226	7.35E+01	2.00E-01		✓

Table 7 (Procedure 1) Performance of the enumerative algorithm to solve LCP(p,Q) for large matrices $\lambda(E_n - A_G) - E_n$, with $\lambda = \omega(G) - 1$.

MATRIX	NODES	IT	TIME	μ	$M \notin \text{int}\mathcal{COP}$	$M \notin \mathcal{COP}$
Brock200-1	730	5.35E+04	7.20E+03	0.00E+00	✓	
Brock200-2	760	5.43E+04	7.20E+03	*		
Brock200-3	734	6.19E+04	7.20E+03	*		
Brock200-4	718	4.14E+04	7.20E+03	*		
c-fat200-1	1	1.60E+01	0	8.33E-02		✓
c-fat200-2	1	2.80E+01	0	4.17E-02		✓
c-fat200-5	1	6.10E+01	0	1.72E-02		✓
Hamming6-2	1	4.53E+02	0	3.13E-02		✓
Hamming6-4	1	4.20E+01	0	0.00E+00	✓	
Hamming8-2	1	7.11E+03	2.25E+00	7.81E-03		✓
Hamming8-4	7	3.99E+03	2.61E+00	6.25E-02		✓
Johnson8-2-4	1	1.50E+01	0	0.00E+00	✓	
Johnson8-4-4	1	5.71E+02	0	7.14E-02		✓
Johnson16-2-4	1	1.43E+02	0	0.00E+00	✓	
Johnson32-2-4	1	1.26E+03	2.71E+00	0.00E+00	✓	
Keller4	641	7.85E+04	2.93E+03	9.09E-02		✓
Mann-a9	4	6.50E+01	0	0.00E+00	✓	
Mann-a27	840	3.80E+04	7.20E+03	0.00E+00	✓	

4.6 Procedure 2

Tables 8, 9 and 10 include the performance of Lemke's algorithm for solving LCP(p,Q) by changing the initial basic solution for each application k of the method ($k = 1,\ldots,n+1$). The process is repeated until the method finds a solution with $\mu > 0$ ($M \notin \mathcal{COP}$) or it computes a solution with $\mu = 0$ ($M \notin \text{int}\mathcal{COP}$) or until it terminates in an unbounded ray for all $n+1$ initial basic solutions (no conclusion is given then). The notation IT stands for the total number of pivotal iterations required by Lemke's algorithm for the visited basis.

Table 8 (Procedure 2) Performance of Lemke's algorithm for the matrices of Section 4.1.

	Trivial Basis		Other Basis			
Matrix	It	Time	It	Time	$M \notin \text{int}\mathcal{COP}$	$M \notin \mathcal{COP}$
M_1	1	0	3	0		✓
M_2	1	0	4	0		✓
M_3	1	0	7	0		
M_4	1	0	6	0		
M_5	1	0	19	0	✓	
M_6	1	0	15	0	✓	
M_7	1	0	22	0	✓	

Table 9 (Procedure 2) Performance of Lemke's algorithm for small matrices $\lambda(E_n - A_G) - E_n$, with $\lambda = \omega(G) - 1$.

	Trivial Basis		Other Basis			
Matrix	It	Time	It	Time	$M \notin \text{int}\mathcal{COP}$	$M \notin \mathcal{COP}$
c-fat14-1	1	0	6	0		✓
Brock14	1	0	15	0		✓
Brock16	1	0	59	0	✓	
Brock18	1	0	37	0		✓
Brock20	1	0	24	0		✓
Morgen14	1	0	16	0		✓
Morgen16	1	0	19	0		✓
Morgen18	1	0	7	0		✓
Morgen20	1	0	13	0		✓
Morgen22	1	0	28	0		✓
Johnson6-2-4	1	0	3	0		✓
Johnson6-4-4	1	0	3	0		✓
Johnson7-2-4	1	0	89	0		✓
Jagota14	1	0	14	0		✓
Jagota16	1	0	18	0		✓
Jagota18	1	0	22	0		✓
sanchis14	1	0	5	0		✓
sanchis16	1	0	45	0		✓
sanchis18	1	0	5	0		✓
sanchis20	1	0	5	0		✓
sanchis22	1	0	22	0		✓

The numerical results indicate that Lemke's method was able to verify the matrices to be not strictly copositive or not copositive for all instances but four. Actually, three of these instances were the ones for which the enumerative method was not conclusive. The computational effort also increases much with the dimension of the problems.

4.7 Procedure 3

In the next tables 11-13 we report the performance of the solver CPLEX to find a feasible solution for the problems MIP_1 and MIP_2. The symbol (*) stands for problems where the solver CPLEX was unable to find a solution with $\alpha > 0$ within 3600 seconds of CPU time. In this case, CPLEX gives the feasible solution $\alpha = 0$ for MIP_1. We used the notations Nodes, It and Time respectively, for the number of nodes, iterations and CPU seconds required by the solver.

Table 10 (Procedure 2) Performance of Lemke's algorithm for large matrices $\lambda(E_n - A_G) - E_n$, with $\lambda = \omega(G) - 1$.

Matrix	Trivial Basis It	Trivial Basis Time	Other Basis It	Other Basis Time	$M \notin \text{int}\mathcal{COP}$	$M \notin \mathcal{COP}$
Brock200-1	1	0	2.86E+03	6.41E+00		
Brock200-2	1	0	1.43E+03	3.27E+00		
Brock200-3	1	0	1.89E+03	5.82E+00		
Brock200-4	1	0	2.12E+03	6.77E+00		
c-fat200-1	1	0	1.20E+01	0		✓
c-fat200-2	1	0	2.40E+01	0		✓
c-fat200-5	1	0	5.80E+01	0		✓
Hamming6-2	1	0	3.20E+01	0		✓
Hamming6-4	1	0	4.00E+00	0		✓
Hamming8-2	1	0	1.28E+02	0		✓
Hamming8-4	1	0	1.60E+01	0		✓
Johnson8-2-4	1	0	2.20E+01	0		✓
Johnson8-4-4	1	0	2.31E+02	0		✓
Johnson16-2-4	1	0	8.00E+00	0		✓
Johnson32-2-4	1	0	1.60E+01	1.00E+00		✓
Keller4	1	0	1.33E+03	2.47E+00	✓	
Mann-a9	1	0	1.80E+01	0		✓
Mann-a27	1	0	4.73E+04	4.19E+02	✓	

Table 11 (Procedure 3) Performance of applying CPLEX to MIP_1 and MIP_2 for the matrices of Section 4.1.

Matrix	MIP_1						MIP_2				
	$\alpha = 0$ $M \in \text{int}\mathcal{COP}$	$(\alpha \neq 0,$ $y_{n+1} = 0)$ $M \notin \text{int}\mathcal{COP}$	$(\alpha \neq 0,$ $y_{n+1} > 0)$ $M \notin \mathcal{COP}$	It	Nodes	Time	$y_{n+1} = 0$ $M \in \mathcal{COP}$	$y_{n+1} > 0$ $M \notin \mathcal{COP}$	It	Nodes	Time
M_1			✓	10	0	0					
M_2			✓	9	0	0					
M_3	✓			20	5	0					
M_4	✓			20	5	0					
M_5		✓		22	3	0	✓		68	26	0
M_6		✓		21	5	0	✓		32	10	0
M_7		✓		56	12	0	✓		59	18	0

The numerical results indicate that using the MIP formulation of LCP(p, Q) seems to be an interesting approach for showing that a matrix is not strictly copositive or not copositive. Like the remaining procedures, the computational effort is small for the smallest problems but tends to increase with the dimension of the problems. The procedure could not give an indication of non-copositivity for four instances. However, Lemke's method has shown non-copositivity for these four matrices.

4.8 Summary of numerical results

As a conclusion of this numerical study, we suggest to use in Step 2 Procedure 2 (Lemke's method) first and then Procedure 3 (mixed integer formulation) when the Procedure 2 is not conclusive. It is important to add that such a hybrid method was able to show that all matrices but one (Mann-a27) of the maximum clique collection are not copositive.

The numerical experiments also show that in general it is easier to show that a matrix M is not strictly copositive than showing that M is not copositive. For that

Table 12 (Procedure 3) Performance of applying CPLEX to MIP_1 and MIP_2 for small matrices $\lambda(E_n - A_G) - E_n$, with $\lambda = \omega(G) - 1$.

	MIP_1						MIP_2				
MATRIX	$\alpha = 0$ $M \in \text{int}\mathcal{COP}$	$(\alpha \neq 0,$ $y_{n+1} = 0)$ $M \notin \text{int}\mathcal{COP}$	$(\alpha \neq 0,$ $y_{n+1} > 0)$ $M \notin \mathcal{COP}$	IT	NODES	TIME	$y_{n+1} = 0$ $M \in \mathcal{COP}$	$y_{n+1} > 0$ $M \notin \mathcal{COP}$	IT	NODES	TIME
c-fat14-1		✓		546	68	0		✓	561	9.00E+01	0
Brock14		✓		281	34	0		✓	932	2.12E+02	0
Brock16			✓	205	25	0					
Brock18		✓		14	0	0		✓	1474	3.71E+02	0
Brock20		✓		374	54	0		✓	2957	4.19E+02	0
Morgen14			✓	69	1	0					
Morgen16		✓		40	4	0		✓	1038	1.99E+02	0
Morgen18		✓		304	29	0		✓	1406	2.40E+02	0
Morgen20		✓		478	66	0		✓	1786	2.89E+02	0
Morgen22		✓		199	23	0		✓	1414	2.78E+02	0
Johnson6-2-4		✓		179	19	0		✓	239	4.90E+01	0
Johnson6-4-4		✓		122	8	0		✓	160	2.80E+01	0
Johnson7-2-4		✓		340	38	0		✓	1026	1.21E+02	0
Jagota14			✓	266	53	0					
Jagota16			✓	719	137	0					
Jagota18			✓	1934	297	0					
sanchis14		✓		169	27	0		✓	696	1.75E+02	0
sanchis16			✓	182	20	0					
sanchis18			✓	37	1	0					
sanchis20			✓	42	2	0					
sanchis22			✓	62	1	0					

reason we make use of Lemma 6 to establish that the matrix Mann-a27 of the clique collection is not copositive. Let $M := (\omega(G)-1)(E_n - A_G) - E_n$ be this matrix. The matrix $H := E_n - A_G$ is a nonnegative matrix with positive diagonal elements and $H \in \text{int}\mathcal{COP}$ [11, Chapter 3]. When we apply the hybrid method to the matrix $P := M + 0.1H$ (i.e., $\beta = 0.1$) then the algorithm terminates in Step 1 with the indication that $P \notin \text{int}\mathcal{COP}$. The algorithm used more than one initial basic solution, and required 2.56E+04 iterations and 2.67E+02 CPU time. So, matrix Mann-a27 for $\lambda = \omega(G) - 1$ is not copositive.

As a final conclusion of this numerical study, the hybrid algorithm with Steps 0, 1 and 2 as discussed above was able to establish non-copositivity for all maximum clique matrices. Note that for one matrix this certificate has been given based on the application of the algorithm to a carefully chosen related matrix. This means that such a procedure was able to give a lower bound of $\omega(G)$ for all these problems. Table 14 demonstrates this behavior of the hybrid algorithm and shows the superiority of this algorithm over the approaches discussed by Bomze and Eichfelder [3], Bundfuss and Dür [8], and Žilinskas and Dür [33].

4.9 Global solution with Baron

Tables 15, 16 and 17 report the performance of the solver BARON (with default parameters settings) for finding the global minimum of problem (1). We marked with (*) the problems for which the solver was not able to prove the optimality of the solution within the allowed 7200 CPU seconds and we report the best upper bound obtained by the solver for the limited time of execution. For all small instances $\lambda(E_n - A_G) - E_n$

Table 13 (Procedure 3) Performance of applying CPLEX to MIP_1 and MIP_2 for large matrices $\lambda(E_n - A_G) - E_n$, with $\lambda = \omega(G) - 1$.

MATRIX	MIP_1					
	$\alpha = 0$ $M \in \mathrm{int}\mathcal{COP}$	($\alpha \neq 0$, $y_{n+1} = 0$) $M \notin \mathrm{int}\mathcal{COP}$	($\alpha \neq 0$, $y_{n+1} > 0$) $M \notin \mathcal{COP}$	IT	NODES	TIME
Brock200-1			✓	3.46E+06	8.59E+04	9.13E+02
Brock200-2			✓	2.47E+06	5.48E+04	5.29E+02
Brock200-3			✓	7.21E+06	1.79E+05	1.55E+03
Brock200-4			✓	8.83E+04	1.96E+03	4.86E+01
c-fat200-1		✓		1.31E+05	2.34E+03	5.84E+01
c-fat200-2		✓		5.31E+05	5.95E+03	1.54E+02
c-fat200-5	*			1.61E+07	3.57E+05	3.60E+03
Hamming6-2			✓	4.52E+07	3.20E+06	3.13E+03
Hamming6-4		✓		2.90E+05	1.62E+04	1.13E+01
Hamming8-2	*			5.88E+06	1.48E+05	3.60E+03
Hamming8-4	*			1.13E+07	1.33E+05	3.60E+03
Johnson8-2-4		✓		4.60E+01	0.00E+00	0
Johnson8-4-4			✓	1.40E+05	5.07E+03	1.06E+01
Johnson16-2-4			✓	3.47E+07	5.68E+05	3.60E+03
Johnson32-2-4			✓	1.59E+06	7.75E+03	3.60E+03
Keller4		✓		2.42E+07	5.67E+05	3.60E+03
Mann-a9		✓		4.43E+06	3.78E+05	4.00E+02
Mann-a27	*			1.53E+06	6.31E+04	3.60E+03

MATRIX	MIP_2				
	$y_{n+1} = 0$ $M \in \mathcal{COP}$	$y_{n+1} > 0$ $M \notin \mathcal{COP}$	IT	NODES	TIME
Brock200-1					
Brock200-2					
Brock200-3					
Brock200-4					
c-fat200-1		✓	9.92E+04	4.63E+03	3.72E+01
c-fat200-2		✓	2.67E+05	2.40E+04	8.11E+01
c-fat200-5					
Hamming6-2					
Hamming6-4		✓	4.23E+04	2.55E+03	1.94E+00
Hamming8-2					
Hamming8-4					
Johnson8-2-4		✓	2.22E+03	3.49E+02	0
Johnson8-4-4					
Johnson16-2-4					
Johnson32-2-4					
Keller4		✓	2.22E+07	1.84E+06	3.60E+03
Mann-a9		✓	3.28E+07	1.80E+07	3.60E+03
Mann-a27					

this upper bound allows to conclude that $M \notin \mathcal{COP}$ but it is inconclusive for five of the bigger matrices. Note that M is considered to be copositive if the globally optimal value is greater than or equal to minus the square root of macheps (10^{-8}). These results clearly indicate that it is better to employ our new hybrid method to establish non-copositivity than an efficient global optimizer for studying this property by exploiting the definition of a copositive matrix.

5 Conclusions

In this paper we introduce a number of procedures based on the linear complementarity problem and on linear programming. These procedures proved to be useful

Table 14 Comparison of lower bounds for DIMACS collection with the results in [3], [8] and [33].

		Lower Bounds			
Matrix	$\omega(G)$	Hybrid Algorithm	in [3]	in [8]	in [33]
Brock200-1	21	21			13
Brock200-2	12	12		9	10
Brock200-3	15	15		11	11
Brock200-4	17	17	7		13
c-fat200-1	12	12			
c-fat200-2	24	24			
c-fat200-5	58	58			
Hamming6-2	32	32	32	28	32
Hamming6-4	4	4		4	4
Hamming8-2	128	128	128		128
Hamming8-4	16	16	16	12	16
Johnson8-2-4	4	4	4	4	4
Johnson8-4-4	14	14	14	14	14
Johnson16-2-4	8	8	8	8	8
Johnson32-2-4	16	16			16
Keller4	11	11	6	9	8
Mann-a9	16	16		16	16
Mann-a27	126	126			121

Table 15 Performance of the solver Baron for the matrices of Section 4.1.

Matrix	Upper Bound	Nodes	Time	$M \in \text{int}\mathcal{COP}$	$M \in \mathcal{COP}$	$M \notin \mathcal{COP}$
M_1	-9.19E-02	129	0			✓
M_2	-1.16E-01	365	0			✓
M_3	2.30E-01	21	0	✓		
M_4	2.30E-01	7	0	✓		
M_5	-7.40E-17	1.42E+03	0		✓	
M_6	0.00E+00	6.98E+04	4.68E+01		✓	
M_7	0.00E+00	1.73E+05	1.53E+02		✓	

Table 16 Performance of the solver Baron for small matrices $\lambda(E_n - A_G) - E_n$, with $\lambda = \omega(G) - 1$.

Matrix	Upper Bound	Nodes	Time	$M \in \mathcal{COP}$	$M \notin \mathcal{COP}$
c-fat14-1	-1.67E-01	3.57E+05	5.76E+02		✓
Brock14	-2.00E-01*	4.60E+05	7.20E+03		✓
Brock16	-2.00E-01*	8.96E+05	1.61E+03		✓
Brock18	-2.00E-01*	4.60E+05	7.20E+03		✓
Brock20	-2.00E-01*	2.51E+06	7.20E+03		✓
Morgen14	-2.00E-01*	1.51E+05	7.20E+03		✓
Morgen16	-2.00E-01*	4.73E+05	7.20E+03		✓
Morgen18	-2.00E-01*	2.86E+06	7.20E+03		✓
Morgen20	-2.00E-01*	2.34E+06	7.20E+03		✓
Morgen22	-2.00E-01*	1.85E+06	5.83E+03		✓
Johnson6-2-4	-3.33E-01*	1.80E+06	4.30E+03		✓
Johnson6-4-4	-3.33E-01*	1.72E+06	4.09E+03		✓
Johnson7-2-4	-3.33E-01*	2.92E+04	7.20E+03		✓
Jagota14	-1.67E-01	3.06E+04	5.27E+01		✓
Jagota16	-1.25E-01	6.62E+05	1.42E+03		✓
Jagota18	-1.00E-01*	2.11E+06	7.20E+03		✓
sanchis14	-2.00E-01*	3.83E+05	5.15E+02		✓
sanchis16	-2.00E-01*	3.93E+05	7.13E+02		✓
sanchis18	-2.00E-01*	3.68E+05	6.37E+02		✓
sanchis20	-2.00E-01*	4.02E+06	7.20E+03		✓
sanchis22	-2.00E-01*	3.29E+06	7.20E+03		✓

Table 17 Performance of the solver Baron for matrices $\lambda(E_n - A_G) - E_n$, with $\lambda = \omega(G) - 1$.

MATRIX	UPPER BOUND	NODES	TIME	$M \in \mathcal{COP}$	$M \notin \mathcal{COP}$
Brock200-1	5.26E-02*	7.49E+02	7.20E+03		
Brock200-2	1.00E-01*	7.75E+02	7.20E+03		
Brock200-3	7.69E-02*	8.92E+02	7.20E+03		
Brock200-4	6.67E-02*	8.62E+02	7.20E+03		
c-fat200-1	-8.33E-02*	4.20E+03	7.20E+03		✓
c-fat200-2	-4.17E-02*	3.68E+03	7.20E+03		✓
c-fat200-5	-1.72E-02*	3.06E+03	7.20E+03		✓
Hamming6-2	-3.13E-02*	6.68E+04	7.20E+03		✓
Hamming6-4	-2.50E-01*	3.80E+04	7.20E+03		✓
Hamming8-2	-7.81E-03*	1.19E+03	7.20E+03		✓
Hamming8-4	-6.25E-02*	9.15E+02	7.20E+03		✓
Johnson8-2-4	-2.50E-01*	1.35E+06	7.20E+03		✓
Johnson8-4-4	-7.14E-02*	4.04E+04	7.20E+03		✓
Johnson16-2-4	-1.25E-01*	5.31E+03	7.20E+03		✓
Johnson32-2-4	-6.25E-02*	3.31E+02	7.20E+03		✓
Keller4	-9.09E-02*	1.21E+03	7.20E+03		✓
Mann-a9	-6.25E-02*	3.91E+05	7.20E+03		✓
Mann-a27	6.84E-02*	1.02E+03	7.20E+03		

for studying the copositivity or non-copositivity of a matrix. A hybrid algorithm has been constructed based on these procedures and has shown to perform well to establish non-copositivity of matrices of the so-called maximum clique collection that are usually used as test instances for similar procedures.

Numerical results with these instances indicate that the hybrid algorithm is more efficient to detect that a matrix is not strictly copositive than showing that it is not copositive. This conclusion has been exploited to establish the non-copositivity of one of the matrices of the maximum clique set by showing that a carefully chosen related matrix is not strictly copositive. In our opinion, such type of approach should deserve more attention in the future. Recently, a similar strategy was suggested by Sponsel et al. in [29, Theorem 3.3].

It is also interesting to investigate the performance of the algorithm discussed in this paper for instances with copositive matrices. Finally, the use of these techniques to provide lower and upper bounds of copositive programming formulations of some structured global optimization problems (such as the maximum clique problem) should deserve attention in the near future.

References

1. Adler, I. and Verma, S.: The Linear Complementarity Problem, Lemke Algorithm, Perturbation, and the Complexity Class PPAD. Manuscript, Department of IEOR, University of California, Berkeley, CA 94720, February (2011).
2. Bomze, I.M.: Copositive optimization – recent developments and applications. *Eur. J. Oper. Res.* **216**, 509–520 (2012).
3. Bomze, I.M. and Eichfelder, G.: Copositivity detection by difference-of-convex decomposition and ω-subdivision. *Mathematical Programming Ser. A* **138**, 365–400 (2013).
4. 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.* **13**, 369–387 (1998).
5. Bomze, I.M., Schachinger, W. and Uchida, G.: Think co(mpletely)positive ! Matrix properties, examples and a clustered bibliography on copositive optimization. *J. Global Optim.* **52**, 425–445 (2012).
6. Brooke, A., Kendrick, D., Meeraus, A. and Raman, R.: GAMS *a User's Guide*. GAMS Development Corporation, Washington (1998).

7. Bundfuss, S.: Copositive Matrices,Copositive Programming, and Applications. Dissertation, Technischen Universität Darmstadt (2009).
8. Bundfuss, S., Dür, M.: Algorithmic copositivity detection by simplicial partition. *Linear Algebra Appl.* **428**, 1511–1523 (2008).
9. Burer, S.: On the copositive representation of binary and continuous nonconvex quadratic programs. *Math. Program.* **120**, 479–495 (2009).
10. Burer, S.: Copositive programming. In: M.F. Anjos and J.-B. Lasserre (eds.), *Handbook of Semidefinite, Cone and Polynomial Optimization: Theory, Algorithms, Software and Applications*, Springer Series in Operations Research and Management Science (2011).
11. Cottle, R.W., Pang, J.-S. and Stone, R.E. *The Linear Complementarity Problem*. SIAM, New York (2009).
12. CPLEX, I.: 11.0 Users Manual. ILOG SA, Gentilly, France (2008).
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. DIMACS: Second DIMACS Challenge. Test instances available at http://dimacs.rutgers.edu/challenges, last accessed 13 Jan. 2010.
15. Dür, M.: Copositive Programming – a survey. In: M. Diehl, F. Glineur, E. Jarlebring, W. Michiels (eds.), *Recent Advances in Optimization and its Applications in Engineering*, pp. 3-20. Springer, New York (2010).
16. Eichfelder, G. and Jahn, J.: Set-Semidefinite Optimization. *Journal of Convex Analysis* **15**, 767–801 (2008).
17. Facchinei, F. and Pang, J.-S., *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer-Verlag, New York (2003).
18. Hall, M. Jr., Newman, M.: Copositive and completely positive quadratic forms. *Proc. Camb. Phil. Soc.* **59**, 329–339 (1963).
19. Hiriart-Urruty, J.-B., Seeger, A.: A Variational Approach to Copositive Matrices. *SIAM Rev.* **52** 593–629 (2010).
20. Hoffman, A.J. and Pereira, F.: On copositive matrices with -1,0,1 entries. *Journal of Combinatorial Theory* (A) **14**, 302–309 (1973).
21. Horst, R., Pardalos, P.M. and Thoai, N. *Introduction to Global Optimization*. Kluwer Academic Publishers, Dordrecht (2000).
22. Júdice, J., Faustino, A. and Ribeiro, I.: On the solution of NP-hard linear complementarity problems. *Top* **10**, 125–145 (2002).
23. Kaplan, W.: A copositivity probe. *Linear Algebra Appl.* **337**, 237–251 (2001).
24. Moler, C., Little, J., and Bangert, S. The MathWorks, Sherborn, Mass. Matlab User's Guide - The Language of Technical Computing (2001).
25. Murtagh, B., Saunders, M., Murray, W., Gill, P., Raman, R. and Kalvelagen, E.: MINOS-NLP. Systems Optimization Laboratory, Stanford University, Palo Alto, CA.
26. Murty, K.G., *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann Verlag, Berlin, (1988).
27. Murty, K.G. and Kabadi, S.N.: Some NP-complete problems in quadratic and linear programming. *Math. Programming* **39**, 117–129 (1987).
28. Sahinidis, N. and Tawarmalani, M.: BARON 7.2.5: Global Optimization of Mixed-Integer Nonlinear Programs. GAMS Development Corporation, Washington (2005).
29. Sponsel, J., Bundfuss, S., Dür, M.: An improved algorithm to test copositivity *J. Glob. Optim.* **52**, 537–551 (2012).
30. Tanaka, A., Yoshise, A.: An LP-based Algorithm to Test Copositivity. *Pacific Journal of Optimization* **11(1)**, 101–120 (2015).
31. Väliaho, H.: Criteria for copositive matrices. *Linear Algebra Appl.* **81**, 19–34 (1986).
32. Väliaho, H.: Quadratic-programming criteria for copositive matrices. *Linear Algebra Appl.* **119**, 163–182 (1989).
33. Žilinskas, J., Dür, M.: Depth-first simplicial partition for copositivity detection, with an application to Maxclique. *Optim. Methods. Softw.* **26**, 499–510 (2011).