An Adaptive Linear Approximation Algorithm for Copositive Programs

Bundfuss, Stefan; Dur, Mirjam

Published in:
SIAM Journal on Optimization

DOI:
10.1137/070711815

Publication date:
2009

Citation for published version (APA):
Bundfuss, S., & Dur, M. (2009). An Adaptive Linear Approximation Algorithm for Copositive Programs. *SIAM Journal on Optimization*, *20*(1), 30-53. https://doi.org/10.1137/070711815

SIAM J. Optim.
Vol. 20, No. 1, pp. 30–53

AN ADAPTIVE LINEAR APPROXIMATION ALGORITHM FOR COPOSITIVE PROGRAMS[*]

STEFAN BUNDFUSS[†] AND MIRJAM DÜR[‡]

Abstract. We study linear optimization problems over the cone of copositive matrices. These problems appear in nonconvex quadratic and binary optimization; for instance, the maximum clique problem and other combinatorial problems can be reformulated as such problems. We present new polyhedral inner and outer approximations of the copositive cone which we show to be exact in the limit. In contrast to previous approximation schemes, our approximation is not necessarily uniform for the whole cone but can be guided adaptively through the objective function, yielding a good approximation in those parts of the cone that are relevant for the optimization and only a coarse approximation in those parts that are not. Using these approximations, we derive an adaptive linear approximation algorithm for copositive programs. Numerical experiments show that our algorithm gives very good results for certain nonconvex quadratic problems.

Key words. copositive cone, copositive programming, quadratic programming, approximation algorithms

AMS subject classifications. 90C05, 90C20, 15A48, 15A63, 05C69

DOI. 10.1137/070711815

1. Introduction.

In this paper we are concerned with the topic of conic formulations and relaxations for binary and quadratic problems. Semidefinite relaxations have been proposed as a strong method to obtain good bounds for many combinatorial optimization problems. Quist et al. [21] suggested that one might get tighter relaxations by looking at cones other than the semidefinite one. Bomze et al. [3] were the first to observe that certain combinatorial problems like the maximum clique problem can equivalently be reformulated as a linear optimization problem over the cone of so-called completely positive matrices. A matrix A is called completely positive if it can be decomposed as $A = BB^T$ with an entrywise nonnegative matrix B. There is a large amount of papers on complete positivity in the linear algebra literature (a good survey is [1]), but the optimization community has only recently become aware of the connections between the fields.

The completely positive cone $\mathcal{C}^*$ is the dual cone of the cone $\mathcal{C}$ of copositive matrices. Formally, these cones are defined as

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

(where $\mathcal{S}$ is the set of symmetric $n \times n$ matrices), and

$$\mathcal{C}^* = \left\{ \sum_{i=1}^{k} v_i v_i^T : v_i \in \mathbb{R}^n_+ \text{ for all } i = 1, \dots, k \right\}.$$

Both $\mathcal{C}$ and $\mathcal{C}^*$ are closed, convex, pointed, full dimensional, nonpolyhedral cones. It can be shown that the interior of $\mathcal{C}$ is the set of strictly copositive matrices: $\text{int}(\mathcal{C}) =$

[*] Received by the editors December 25, 2007; accepted for publication (in revised form) October 21, 2008; published electronically March 13, 2009.
http://www.siam.org/journals/siopt/20-1/71181.html

[†] Department of Mathematics, TU Darmstadt, Schloßgartenstr. 7, D–64289 Darmstadt, Germany (bundfuss@mathematik.tu-darmstadt.de).

[‡] Institute of Mathematics and Computer Science, University of Groningen, P.O. Box 407, 9700 AK Groningen, The Netherlands (M.E.Dur@rug.nl).

$\{A \in \mathcal{S} : x^T Ax > 0 \text{ for all } x \in \mathbb{R}^n_+ \setminus \{0\}\}$. The interior of $\mathcal{C}^*$ has recently been characterized in [10]. The extremal rays of $\mathcal{C}^*$ are known to be the rank one matrices vv^T with $v \geq 0$, while characterizing the extremal copositive matrices is an open problem. Both cones are related to the cones $\mathcal{N}$ of nonnegative symmetric matrices and $\mathcal{S}^+$ of symmetric positive semidefinite matrices, since

$$\mathcal{C} \supseteq \mathcal{S}^+ + \mathcal{N} \quad \text{and} \quad \mathcal{C}^* \subseteq \mathcal{S}^+ \cap \mathcal{N}.$$

Interestingly, for $n \times n$-matrices of order $n \leq 4$, equality holds in the above relations, whereas for $n \geq 5$, both inclusions are strict; see [1]. In contrast to $\mathcal{N}$ and $\mathcal{S}^+$, the cones $\mathcal{C}$ and $\mathcal{C}^*$ are not tractable: It is known that testing whether a given matrix is in $\mathcal{C}$ is co-NP-complete (cf. [16]). Consequently, restating a problem as an optimization problem over one of these cones does not resolve the difficulty of that problem. However, we believe that getting a good understanding of the conic formulations will help to improve the solution strategies for both binary and nonconvex quadratic problems. Moreover, in some cases copositive formulations motivate stronger semidefinite relaxations.

Up to now, the list of problems known to have representations as completely positive programs has grown to include standard quadratic problems [3], the stable set problem [15, 9], the quadratic assignment problem [20], and certain graph-partitioning problems [19]. Burer [6] showed the very general result that every quadratic problem with linear and binary constraints can be rewritten as such a problem. More precisely, he showed that a quadratic binary problem of the form

$$\begin{array}{ll} \min & x^T Qx + 2c^T x \\ \text{s. t.} & a_i^T x = b_i, \quad i = 1, \dots, m, \\ & x \geq 0, \\ & x_j \in \{0, 1\}, \quad j \in B, \end{array}$$

(with Q not necessarily positive semidefinite) can equivalently be written as the following linear problem over the cone of completely positive matrices:

$$\begin{array}{ll} \min & \langle Q, X \rangle + 2c^T x \\ \text{s. t.} & a_i^T x = b_i, \quad i = 1, \dots, m, \\ & \langle a_i a_i^T, X \rangle = b_i^2, \quad i = 1, \dots, m, \\ & x_j = X_{jj}, \quad j \in B, \\ & \begin{pmatrix} 1 & x \\ x & X \end{pmatrix} \in \mathcal{C}^*. \end{array}$$

This means that any nonconvex quadratic integer problem can equivalently be written as a linear problem over a convex cone, i.e., a convex optimization problem which has no nonglobal local optima. It is an open question whether problems with general quadratic constraints can similarly be restated as completely positive problems.

In this paper we develop an algorithm to solve the dual problem, i.e., the optimization problem over the copositive cone which can be stated in the form

$$\text{(CP)} \qquad \begin{array}{ll} \max & \langle C, X \rangle \\ \text{s. t.} & \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m \\ & X \in \mathcal{C} \end{array}$$

with $C, A_i \in \mathbb{R}^{n \times n}, b_i \in \mathbb{R}$.

Our approach is based on new polyhedral inner and outer approximations of the copositive cone which we show to be exact in the limit. In contrast to previous approximation schemes, our approximation is not necessarily uniform for the whole cone but can be guided adaptively through the objective function, yielding a good approximation in those parts of the cone that are relevant for the optimization and only a coarse approximation in those parts that are not. Using these approximations, we derive an adaptive linear approximation algorithm for copositive programs. We show that our algorithm gives very good results for certain nonconvex quadratic problems.

Note that (CP) is related to the problem of testing whether a given matrix is in $\mathcal{C}^*$: From the fact that the cone $\mathcal{C}$ is the dual of $\mathcal{C}^*$ we have

$$\begin{aligned} A \notin \mathcal{C}^* &\Leftrightarrow \exists X \in \mathcal{C} : \langle A, X \rangle < 0 \\ &\Leftrightarrow \exists X \in \mathcal{C} : \langle I + E, X \rangle = 1, \langle A, X \rangle < 0 \\ &\Leftrightarrow \min\{\langle A, X \rangle : \langle I + E, X \rangle = 1, X \in \mathcal{C}\} < 0. \end{aligned}$$

(Here I denotes the identity and E the all ones matrix, and $\langle I + E, X \rangle = 1$ serves as a normalization constraint.) This minimization problem is of the form (CP), so an algorithm to solve (CP) can be used to decide whether or not $A \in \mathcal{C}^*$. It is an open question how a matrix A known to be in $\mathcal{C}^*$ can be factorized into $A = BB^T$, cf. [2] and [14] for attempts to answer this question.

1.1. Notation. Throughout the paper we use the following notation: The nonnegative orthant is denoted by $\mathbb{R}^n_+$, and the unit vectors are denoted by e_i. For a given vector v or matrix M, the relations $v \geq 0$ and $M \geq 0$ will be understood entrywise. We write $\mathcal{S}$ to denote the cone of symmetric matrices, $\mathcal{N} = \{A \in \mathcal{S} : A \geq 0\}$ to denote the cone of (entrywise) nonnegative matrices, and $\mathcal{S}^+ = \{A \in \mathcal{S} : A \succeq 0\}$ to denote the cone of positive semidefinite matrices. Dimensions of the cones will always be obvious from the context and therefore not stated explicitly. As usual, the inner product in $\mathcal{S}$ is defined as $\langle A, B \rangle := \operatorname{trace}(AB)$.

1.2. Relations to previous work. Since we will compare our algorithm to existing approaches, we briefly summarize previous work on copositive programming. Copositivity of a matrix is defined by positivity of a quadratic form, whence previous approaches have used various conditions which ensure positivity of polynomials.

For a given matrix $M \in \mathcal{S}$, consider the polynomial

$$P_M(x) := \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} x_i^2 x_j^2.$$

Clearly, $M \in \mathcal{C}$ if and only if $P_M(x) \geq 0$ for all $x \in \mathbb{R}^n$. A sufficient condition for this is that $P_M(x)$ has a representation as a sum of squares (sos) of polynomials. Parrilo [17] showed that $P_M(x)$ allows a sum of squares decomposition if and only if $M \in \mathcal{S}^+ + \mathcal{N}$, yielding again the relation $\mathcal{S}^+ + \mathcal{N} \subseteq \mathcal{C}$. Using similar reasoning, Parrilo [17] defined the following hierarchy of cones (cf. also [15] and [4]) for $r \in \mathbb{N}$:

$$\mathcal{K}^r := \left\{ M \in \mathcal{S} : P_M(x) \left(\sum_{i=1}^{n} x_i^2 \right)^r \text{ has an sos decomposition} \right\}.$$

Parrilo showed $\mathcal{S}^+ + \mathcal{N} = \mathcal{K}^0 \subset \mathcal{K}^1 \subset \dots$, and $\operatorname{int}(\mathcal{C}) \subseteq \bigcup_{r \in \mathbb{N}} \mathcal{K}^r$, so the cones $\mathcal{K}^r$ approximate $\mathcal{C}$ from the interior. Since the sos condition can be written as a system of linear matrix inequalities (LMIs), optimizing over $\mathcal{K}^r$ amounts to solving a semidefinite program (SDP).

Exploiting a different sufficient condition for nonnegativity of a polynomial, de Klerk and Pasechnik [15], cf. also Bomze and de Klerk [4], define

$$\mathcal{C}^r := \left\{ M \in \mathcal{S} : P_M(x) \left(\sum_{i=1}^n x_i^2 \right)^r \text{ has nonnegative coefficients} \right\}.$$

de Klerk and Pasechnik showed that $\mathcal{N} = \mathcal{C}^0 \subset \mathcal{C}^1 \subset \dots$, and $\operatorname{int}(\mathcal{C}) \subseteq \bigcup_{r \in \mathbb{N}} \mathcal{C}^r$. Each of these cones is polyhedral, so optimizing over one of them is solving an LP.

Refining these approaches, Peña et al. [18] derive yet another hierarchy of cones approximating $\mathcal{C}$. Adopting standard multiindex notation, where for a given multiindex $\beta \in \mathbb{N}^n$ we have $|\beta| := \beta_1 + \dots + \beta_n$ and $x^\beta := x_1^{\beta_1} \cdots x_n^{\beta_n}$, they define the following set of polynomials

$$\mathcal{E}^r := \left\{ \sum_{\beta \in \mathbb{N}^n, |\beta| = r} x^\beta x^T (P_\beta + N_\beta) x : P_\beta \in \mathcal{S}^+, N_\beta \in \mathcal{N} \right\}.$$

With this, they define the cones

$$\mathcal{Q}^r := \left\{ M \in \mathcal{S} : x^T M x \left(\sum_{i=1}^n x_i^2 \right)^r \in \mathcal{E}^r \right\}.$$

They show that $\mathcal{C}^r \subseteq \mathcal{Q}^r \subseteq \mathcal{K}^r$ for all $r \in \mathbb{N}$, with $\mathcal{Q}^r = \mathcal{K}^r$ for $r = 0, 1$. Similar to $\mathcal{K}^r$, the condition $M \in \mathcal{Q}^r$ can be rewritten as a system of LMIs. Optimizing over $\mathcal{Q}^r$ is therefore again an SDP.

It is a common feature of all these approximation hierarchies that they approximate $\mathcal{C}$ uniformly and do not take into account any information provided by the objective function of the optimization problem. Moreover, in all these approaches the system of LMIs (resp. linear inequalities) gets large quickly as r increases, meaning that the dimension of the SDPs increases so quickly that current SDP-solvers can only solve problems over those cones for small values of r, i.e., $r \leq 3$ at most.

In contrast to this, in our approach the approximation of $\mathcal{C}$ can be guided through the objective function in such a way that a fine approximation is reached in those regions of $\mathcal{C}$ which are relevant for the optimization, and little computational effort goes to approximating those regions of $\mathcal{C}$ which are not. The dimension (i.e., the number of variables) of the linear subproblems in our algorithm is constant, though the number of constraints grows. Moreover, solving a relaxation of a copositive program over one of the cones introduced above provides in general just a relaxation and no information on the quality of the corresponding bound (an exception is [4]). Our approach works not only with inner approximations of $\mathcal{C}$, but simultaneously with outer approximations. Therefore, it provides exact information on the approximation error and the accuracy of the solution.

We are not aware of comparable approximation schemes for the (dual) cone $\mathcal{C}^*$. A recent attempt to solve optimization problems over $\mathcal{C}^*$ is a descent algorithm by Jarre et al. [13]. We remark that another recent contribution to the field of copositive programming is a unified theory of KKT type optimality conditions and duality by Eichfelder and Jahn [11].

1.3. Outline of the paper. We start in section 2 by reviewing criteria for copositivity of a matrix. Based on these criteria, we develop inner and outer polyhedral approximations of $\mathcal{C}$ in section 3. With these cones, we state our algorithm for copositive programs and prove convergence (section 4). In section 5 we discuss how the algorithm can be fine-tuned and which details make an implementation efficient. Finally, we present numerical results in section 6.

2. Criteria for copositivity. In this section, we review some conditions for copositivity that we developed in [5]. These conditions will be the basis for approximations of the copositive cone $\mathcal{C}$ which we introduce in the next section. We start with the following:

OBSERVATION. *Let $\|\cdot\|$ denote any norm on $\mathbb{R}^n$. We have*

(a) *A is copositive $\Leftrightarrow$ $x^TAx \geq 0$ for all $x \in \mathbb{R}^n_+$ with $\|x\| = 1$,*

(b) *A is strictly copositive $\Leftrightarrow$ $x^TAx > 0$ for all $x \in \mathbb{R}^n_+$ with $\|x\| = 1$.*

If we choose the 1-norm $\|\cdot\|_1$, then the set $\Delta^S := \{x \in \mathbb{R}^n_+ : \|x\|_1 = 1\}$ is the so-called *standard simplex.* The copositivity property then translates to

$$x^TAx \geq 0 \quad \text{for all} \quad x \in \Delta^S,$$

i.e., we search for conditions which ensure that the quadratic polynomial x^TAx is nonnegative over a simplex. A convenient way to describe polynomials with respect to a simplex is to use barycentric coordinates: Let $\Delta = \operatorname{conv}\{v_1, \ldots, v_n\}$ be a simplex and

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

Then $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$ are called the *barycentric coordinates* of x with respect to Δ. The representation of the quadratic form in these coordinates reads

$$x^TAx = \left(\sum_{i=1}^{n} \lambda_i v_i\right)^T A \left(\sum_{j=1}^{n} \lambda_j v_j\right) = \sum_{i,j=1}^{n} v_i^T A v_j \lambda_i \lambda_j.$$

The polynomials $\lambda_1^2, \ldots, \lambda_n^2$ and $2\lambda_i\lambda_j$ $(i \neq j)$ appearing in this representation are called *Bézier–Bernstein polynomials*, and the coefficients $v_i^TAv_j$ are the corresponding *Bézier–Bernstein coefficients.* Since all λ_i are nonnegative on Δ, the next lemma is immediate:

LEMMA 2.1. *Let $\Delta = \operatorname{conv}\{v_1, \ldots, v_n\}$ be a simplex. If $v_i^TAv_j \geq 0$ for all $i, j \in \{1, \ldots, n\}$, then $x^TAx \geq 0$ for all $x \in \Delta$.*

If Δ is the standard simplex $\Delta^S = \operatorname{conv}\{e_1, \ldots, e_n\}$, then this lemma shows that A is copositive if $0 \leq e_i^TAe_j = a_{ij}$ for all i, j. This is the well-known property that any (entrywise) nonnegative matrix is copositive. This condition can be refined by looking at so-called simplicial partitions of Δ^S:

DEFINITION 2.2. *Let Δ be a simplex in $\mathbb{R}^n$. A family $\mathcal{P} = \{\Delta^1, \ldots, \Delta^m\}$ of simplices satisfying*

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

is called a simplicial partition *of Δ. For convenience, we denote by $V_{\mathcal{P}}$ the set of all vertices of simplices in $\mathcal{P}$, and by $E_{\mathcal{P}}$ the set of all edges of simplices in $\mathcal{P}$.*

Simplicial partitions are a useful tool in many branches of applied mathematics. A good survey on this topic including convergence results is [12].

It is easy to see that a simplicial partition can be generated through the following "radial" subdivision of $\Delta = \operatorname{conv}\{v_1, \dots, v_n\}$: let $w \in \Delta \setminus \{v_1, \dots, v_n\}$, which is uniquely represented by

$$w = \sum_{i=1}^{n} \lambda_i v_i, \quad \text{with} \quad \lambda_i \geq 0, \ \sum_{i=1}^{n} \lambda_i = 1.$$

For each index i with $\lambda_i > 0$, form the simplex Δ^i obtained from Δ by replacing the vertex v_i by w, i.e., $\Delta^i = \operatorname{conv}\{v_1, \dots, v_{i-1}, w, v_{i+1}, \dots, v_n\}$. The collection of all those Δ^i is a simplicial partition of Δ. If w is a point on one of the longest edges of Δ, the above procedure is called *bisection* of the simplex along the longest edge. Generating a nested sequence of subsimplices of Δ through midpoint bisection along the longest edge has the nice property that this sequence converges to a singleton. This property is sometimes referred to as "exhaustiveness". It can be generalized from midpoint bisection to settings where the bisection point is an almost arbitrary point on one of the longest edges; see [12] for a detailed discussion.

Using this concept, the following theorem gives sufficient conditions for copositivity which generalize the aforementioned relation that A is copositive if all $a_{ij} \geq 0$:

THEOREM 2.3. *Let $A \in \mathcal{S}$, let $\mathcal{P}$ be a simplicial partition of Δ^S.*

(a) *If $u^T Av \geq 0$ for all $\{u, v\} \in E_{\mathcal{P}}$ and $v^T Av \geq 0$ for all $v \in V_{\mathcal{P}}$, then A is copositive.*

(b) *If $u^T Av > 0$ for all $\{u, v\} \in E_{\mathcal{P}}$ and $v^T Av > 0$ for all $v \in V_{\mathcal{P}}$, then A is strictly copositive.*

Proof. To show (a), it is sufficient to prove nonnegativity of $x^T Ax$ for $x \in \Delta^S$. So choose an arbitrary $x \in \Delta^S$. Then $x \in \Delta$ for some $\Delta \in \mathcal{P}$. By assumption, $u^T Av \geq 0$ for all combinations of vertices of this simplex Δ which, by Lemma 2.1, implies $x^T Ax \geq 0$. Part (b) is shown analogously. □

We define the diameter $\delta(\mathcal{P})$ of a partition $\mathcal{P}$ to be

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

Once a partition gets finer and finer, one will eventually capture more and more strictly copositive matrices. In the limit we get a necessary condition for strict copositivity:

THEOREM 2.4. *Let $A \in \mathcal{S}$ be strictly copositive. Then there exists $\varepsilon = \varepsilon(A) > 0$ such that for all finite simplicial partitions $\mathcal{P}$ of Δ^S with $\delta(\mathcal{P}) \leq \varepsilon$ we have*

$$u^T Av > 0 \textit{ for all } \{u, v\} \in E_{\mathcal{P}} \quad \textit{and} \quad v^T Av > 0 \textit{ for all } v \in V_{\mathcal{P}}.$$

Proof. The detailed proof can be found in [5]. It relies on strict positivity of the bilinear form $x^T Ay$ on the diagonal of the compact set $\Delta^S \times \Delta^S$, followed by a continuity argument. □

Observe that the ε in Theorem 2.4 certainly depends on the matrix A, i.e., there is not a single ε that works uniformly for all strictly copositive A. Indeed, the ε relates to how "ill-conditioned" A is.

3. Polyhedral approximations.

In this section, we present polyhedral inner and outer approximations of the cone $\mathcal{C}$.

3.1. Inner approximation of $\mathcal{C}$. We use the sufficient condition of Theorem 2.3 to define inner approximations of $\mathcal{C}$. As before, consider a simplicial partition $\mathcal{P} = \{\Delta^1, \dots, \Delta^m\}$ of Δ^S, and let $V_\mathcal{P}$ denote the set of all vertices of simplices in $\mathcal{P}$, and $E_\mathcal{P}$ the set of all edges of simplices in $\mathcal{P}$. For a given partition $\mathcal{P}$, define

$$\begin{aligned} \mathcal{I}_\mathcal{P} := \{A \in \mathcal{S} : \; & v^T A v \geq 0 \text{ for all } v \in V_\mathcal{P}, \\ & u^T A v \geq 0 \text{ for all } \{u, v\} \in E_\mathcal{P}\}. \end{aligned}$$

Note that given the vertices u,v, an expression of the form $u^T A v \geq 0$ is a linear inequality for the entries of A. Therefore, $\mathcal{I}_\mathcal{P}$ is a polyhedral cone.

Obviously, $\mathcal{I}_\mathcal{P}$ depends on the partition $\mathcal{P}$. If $\mathcal{P}_1$ and $\mathcal{P}_2$ are two simplicial partitions of the same simplex, we call $\mathcal{P}_2$ a *refinement* of $\mathcal{P}_1$ if for all $\Delta \in \mathcal{P}_1$ there exists a subset $\mathcal{P}_\Delta \subseteq \mathcal{P}_2$ which is a simplicial partition of Δ.

We have the following properties:

LEMMA 3.1. *Let $\mathcal{P}, \mathcal{P}_1, \mathcal{P}_2$ denote simplicial partitions of Δ^S. Then*

(a) *$\mathcal{I}_\mathcal{P}$ is a closed convex polyhedral cone,*

(b) *$\mathcal{I}_\mathcal{P} \subseteq \mathcal{C}$, i.e., $\mathcal{I}_\mathcal{P}$ is an inner approximation of $\mathcal{C}$,*

(c) *if $\mathcal{P}_2$ is a refinement of $\mathcal{P}_1$, then $\mathcal{I}_{\mathcal{P}_1} \subseteq \mathcal{I}_{\mathcal{P}_2}$.*

Proof. (a) is obvious from the definition. (b) follows from Theorem 2.3. To prove (c), let $A \in \mathcal{I}_{\mathcal{P}_1}$, let $\Delta^2 \in \mathcal{P}_2$, and let u, v be two arbitrary vertices of Δ^2 (possibly equal). We have to show $u^T A v \geq 0$. Since $\mathcal{P}_2$ is a refinement of $\mathcal{P}_1$, there exists a simplex $\Delta^1 \in \mathcal{P}_1$ with $\Delta^2 \subseteq \Delta^1$. Therefore, u and v are convex combinations of the vertices $v_1, \dots, v_n$ of Δ^1, i.e., $u = \sum_{i=1}^n \lambda_i v_i$ and $v = \sum_{i=1}^n \mu_i v_i$ with $\lambda_i, \mu_i \geq 0$ for all $i \in \{1, \dots, n\}$ and $\sum_{i=1}^n \lambda_i = 1 = \sum_{i=1}^n \mu_i$. Since $v_i^T A v_j \geq 0$ for all i, j due to $A \in \mathcal{I}_{\mathcal{P}_1}$, we have

$$u^T A v = \sum_{i,j=1}^n \lambda_i \mu_j v_i^T A v_j \geq 0.$$

Therefore, $A \in \mathcal{I}_{\mathcal{P}_2}$. □

Example 3.2. If $\mathcal{P} = \{\Delta^S\}$, i.e., the partition consists only of the standard simplex, then

$$\mathcal{I}_\mathcal{P} = \{A \in \mathcal{S} : a_{ij} \geq 0 \text{ for all } i, j = 1, \dots, n\} = \mathcal{N},$$

i.e., $\mathcal{I}_{\{\Delta^S\}}$ equals the cone $\mathcal{N}$ of nonnegative matrices.

Consider instead the partition $\mathcal{P}_2 = \{\Delta^1, \Delta^2\}$ which is derived from $\mathcal{P}$ by bisecting the edge $\{e_1, e_2\}$ at the midpoint $w := \frac{1}{2}(e_1 + e_2)$. We get $\Delta^1 = \operatorname{conv}\{w, e_2, \dots, e_n\}$ and $\Delta^2 = \operatorname{conv}\{e_1, w, e_3, \dots, e_n\}$. For the definition of $\mathcal{I}_{\mathcal{P}_2}$ this means that the inequality $e_1^T A e_2 \geq 0$ (i.e., $a_{12} \geq 0$) corresponding to the bisected edge is removed and replaced by a number of new inequalities. More precisely,

$$\begin{aligned} \mathcal{I}_{\mathcal{P}_2} = \{A \in \mathcal{S} : \; & a_{ij} \geq 0 \text{ for all } \{i, j\} \neq \{1, 2\}, \\ & a_{i1} + a_{i2} \geq 0 \text{ for all } i = 1, \dots, n, \\ & a_{11} + 2a_{12} + a_{22} \geq 0\}. \end{aligned}$$

This defines a larger cone, i.e., a better approximation to $\mathcal{C}$. Observe that the system defining $\mathcal{I}_{\mathcal{P}_2}$ is redundant. This property will cause some difficulty later in the paper, cf. section 5.2. A reduced representation is

$$\begin{aligned} \mathcal{I}_{\mathcal{P}_2} = \{A \in \mathcal{S} : \; & a_{ij} \geq 0 \text{ for all } \{i, j\} \neq \{1, 2\}, \\ & a_{11} + a_{12} \geq 0, \\ & a_{22} + a_{12} \geq 0\}. \end{aligned}$$

This is a reduction from n^2+n-1 to n^2 inequalities; i.e., already $O(n)$ inequalities are redundant after a single bisection step.

The next theorem shows that a sequence of simplicial partitions $\{\mathcal{P}_\ell\}$ yields a sequence of polyhedral inner approximations $\{\mathcal{I}_{\mathcal{P}_\ell}\}$ that will eventually approximate $\mathcal{C}$ with arbitrary precision, provided that the diameter $\delta(\mathcal{P})$ of the simplicial partition goes to zero.

THEOREM 3.3. *Let $\{\mathcal{P}_\ell\}$ be a sequence of simplicial partitions of Δ^S with $\delta(\mathcal{P}_\ell) \to 0$. Then we have*

$$\operatorname{int}\mathcal{C} \subseteq \bigcup_{\ell\in\mathbb{N}} \mathcal{I}_{\mathcal{P}_\ell} \subseteq \mathcal{C}, \quad \textit{and consequently} \quad \mathcal{C} = \overline{\bigcup_{\ell\in\mathbb{N}} \mathcal{I}_{\mathcal{P}_\ell}}.$$

Proof. Take $A \in \operatorname{int}\mathcal{C}$, i.e., A strictly copositive. Then Theorem 2.4 implies that there exists $\ell_0 \in \mathbb{N}$, such that $A \in \mathcal{I}_{\mathcal{P}_{\ell_0}}$. Therefore $A \in \bigcup_{\ell\in\mathbb{N}} \mathcal{I}_{\mathcal{P}_\ell}$, and hence $\operatorname{int}\mathcal{C} \subseteq \bigcup_{\ell\in\mathbb{N}} \mathcal{I}_{\mathcal{P}_\ell}$. From Lemma 3.1, we have $\mathcal{I}_{\mathcal{P}_\ell} \subseteq \mathcal{C}$ for all $\ell \in \mathbb{N}$, so $\bigcup_{\ell\in\mathbb{N}} \mathcal{I}_{\mathcal{P}_\ell} \subseteq \mathcal{C}$. Finally, $\mathcal{C} = \overline{\bigcup_{\ell\in\mathbb{N}} \mathcal{I}_{\mathcal{P}_\ell}}$ since $\mathcal{C} = \overline{\operatorname{int}\mathcal{C}}$. □

This shows that our approach generates a sequence of approximating polyhedral cones $\mathcal{N} = \mathcal{I}_{\mathcal{P}_0} \subset \mathcal{I}_{\mathcal{P}_1} \subset \ldots \subset \mathcal{C}$ in a similar way as the approaches described in section 1.2. To compare our approximations to the hierarchy of polyhedral cones $\mathcal{C}^r$ by Bomze and de Klerk [4], observe that $\mathcal{C}^0 = \mathcal{N} = \mathcal{I}_{\{\Delta^S\}}$. For $\mathcal{C}^1$, it is shown in [4] that $A \in \mathcal{C}^1$ if and only if

$$\begin{aligned} a_{ii} &\geq 0, & & i \in \{1,\ldots,n\}, \\ a_{ii} + 2a_{ij} &\geq 0, & & i \neq j, \\ a_{ij} + a_{jk} + a_{ki} &\geq 0, & & i < j < k. \end{aligned}$$

To see the difference between the approaches, consider dimension $n = 2$ for simplicity, in which case the above system describing $\mathcal{C}^1$ reduces to

$$a_{11} \geq 0, \quad a_{22} \geq 0, \quad a_{11} + 2a_{12} \geq 0, \quad a_{22} + 2a_{12} \geq 0. \tag{3.1}$$

Consider the partition $\mathcal{P}_1 = \{\operatorname{conv}\{e_1, v\}, \operatorname{conv}\{v, e_2\}\}$ with $v = \frac{1}{2}(e_1 + e_2)$. The corresponding system of inequalities for $\mathcal{I}_{\mathcal{P}_1}$ is then

$$a_{ii} \geq 0, \qquad i \in \{1,2\}, \tag{3.2}$$

$$a_{11} + a_{12} \geq 0, \tag{3.3}$$

$$a_{22} + a_{12} \geq 0, \tag{3.4}$$

plus the redundant inequality $v^T A v \geq 0$. Obviously, system (3.2)–(3.4) is implied by (3.1), and therefore $\mathcal{I}_{\mathcal{P}_1} \supseteq \mathcal{C}^1$. As the matrix $A = \left(\begin{smallmatrix} 1 & -1 \\ -1 & 1 \end{smallmatrix}\right)$ fullfills (3.2)–(3.4) but not (3.1), we have $\mathcal{I}_{\mathcal{P}_1} \neq \mathcal{C}^1$. It is easy to see that for $v = \lambda e_1 + (1-\lambda)e_2$ with $\lambda \in [\frac{1}{3}, \frac{2}{3}]$ we have $\mathcal{I}_{\mathcal{P}_1} \supsetneq \mathcal{C}^1$, whereas for the other values of λ we get $\mathcal{I}_{\mathcal{P}_1} \not\supset \mathcal{C}^1$. These arguments extend to higher dimensions, but get much more technical there.

Comparing our approximation with $\mathcal{S}^+ + \mathcal{N}$, it is clear that there is no partition $\mathcal{P}$ such that $\mathcal{I}_\mathcal{P} \supset \mathcal{S}^+ + \mathcal{N}$ because in dimension $n = 2$ we have $\mathcal{S}^+ + \mathcal{N} = \mathcal{C}$, while $\mathcal{I}_\mathcal{P}$ is a polyhedral subset of $\mathcal{C}$. However, depending on the subdivision strategy it is possible to construct partitions with $\mathcal{I}_\mathcal{P} \not\subset \mathcal{S}^+ + \mathcal{N}$.

3.2. Outer approximation of $\mathcal{C}$.

As before, consider a simplicial partition $\mathcal{P}$ of Δ^S, let $V_\mathcal{P}$ denote the set of all vertices in $\mathcal{P}$, and define

$$\mathcal{O}_\mathcal{P} := \{A \in \mathcal{S} : v^T A v \geq 0 \text{ for all } v \in V_\mathcal{P}\}.$$

It is easy to see that, similar to $\mathcal{I}_{\mathcal{P}}$, the set $\mathcal{O}_{\mathcal{P}}$ is polyhedral, as well. In analogy to Lemma 3.1, we have the following properties:

LEMMA 3.4. *Let $\mathcal{P}, \mathcal{P}_1, \mathcal{P}_2$ denote simplicial partitions of Δ^S. Then*

(a) *$\mathcal{O}_{\mathcal{P}}$ is a closed convex polyhedral cone,*

(b) *$\mathcal{O}_{\mathcal{P}} \supseteq \mathcal{C}$, i.e., $\mathcal{O}_{\mathcal{P}}$ is an outer approximation of $\mathcal{C}$,*

(c) *if $\mathcal{P}_2$ is a refinement of $\mathcal{P}_1$, then $\mathcal{O}_{\mathcal{P}_2} \subseteq \mathcal{O}_{\mathcal{P}_1}$.*

Proof. (a) is obvious from the definition. (b) If $A \in \mathcal{C}$, then $x^T A x \geq 0$ for all $x \in \Delta^S$. Since $V_{\mathcal{P}} \subset \Delta^S$, the statement follows. (c) We have $V_{\mathcal{P}_1} \subseteq V_{\mathcal{P}_2}$. Therefore, the set of inequalities describing $\mathcal{O}_{\mathcal{P}_1}$ is a subset of the set of inequalities describing $\mathcal{O}_{\mathcal{P}_2}$, and hence $\mathcal{O}_{\mathcal{P}_2} \subseteq \mathcal{O}_{\mathcal{P}_1}$. □

Example 3.5. If $\mathcal{P}$ consists only of the standard simplex, i.e., $\mathcal{P} = \{\Delta^S\}$, then

$$\mathcal{O}_{\mathcal{P}} = \{A \in \mathcal{S} : a_{ii} \geq 0 \text{ for all } i\}.$$

This corresponds to the well-known fact that a copositive matrix necessarily has nonnegative entries on the diagonal. Observe that $\mathcal{O}_{\{\Delta^S\}}$ is not pointed.

Performing a midpoint bisection of the edge $\{e_1, e_2\}$ gives the new vertex $w := \frac{1}{2}(e_1 + e_2)$ and the resulting partition $\mathcal{P}_2$ yields the set

$$\mathcal{O}_{\mathcal{P}_2} = \{A \in \mathcal{S} : a_{ii} \geq 0 \text{ for all } i, a_{11} + 2a_{21} + a_{22} \geq 0\},$$

a smaller set and better approximation to $\mathcal{C}$.

The sequence of outer approximations $\{\mathcal{O}_{\mathcal{P}_\ell}\}$ converges to the copositive cone as the partitions $\mathcal{P}_\ell$ get finer.

THEOREM 3.6. *Let $\{\mathcal{P}_\ell\}$ be a sequence of simplicial partitions of Δ^S with $\delta(\mathcal{P}_\ell) \to 0$. Then we have*

$$\mathcal{C} = \bigcap_{\ell \in \mathbb{N}} \mathcal{O}_{\mathcal{P}_\ell}.$$

Proof. Lemma 3.4(b) implies $\mathcal{C} \subseteq \bigcap_{\ell \in \mathbb{N}} \mathcal{O}_{\mathcal{P}_\ell}$. To see the reverse, take $A \notin \mathcal{C}$. Then $\bar{x}^T A \bar{x} < 0$ for some $\bar{x} \in \Delta^S$. From continuity it follows that there is an ε-neighborhood $N_\varepsilon(\bar{x})$ of $\bar{x}$ such that

$$x^T A x < 0 \quad \text{for all } x \in N_\varepsilon(\bar{x}). \tag{3.5}$$

Let $\mathcal{P} \in \{\mathcal{P}_\ell\}$ be some partition with $\delta(\mathcal{P}) < \varepsilon$. Then there is a simplex $\Delta \in \mathcal{P}$ with $\bar{x} \in \Delta$, and hence a vertex v of Δ with $\|\bar{x} - v\| < \varepsilon$, so $v \in N_\varepsilon(\bar{x})$. From (3.5), we see that $v^T A v < 0$, whence $A \notin \mathcal{O}_{\mathcal{P}}$. Therefore, $A \notin \bigcap_{\ell \in \mathbb{N}} \mathcal{O}_{\mathcal{P}_\ell}$. □

3.3. Approximations of the dual cone $\mathcal{C}^*$. Recall that the dual cone of $\mathcal{C}$ is the cone $\mathcal{C}^*$ of completely positive matrices. By duality, the dual cone of an inner (resp. outer) approximation of $\mathcal{C}$ is an outer (resp. inner) approximation of $\mathcal{C}^*$. Indeed, it is not difficult to see that for any partition $\mathcal{P}$ of Δ^S

$$\mathcal{I}_{\mathcal{P}}^* = \left\{ \sum_{\{u,v\} \in E_{\mathcal{P}}} \lambda_{uv}(uv^T + vu^T) + \sum_{v \in V_{\mathcal{P}}} \lambda_v vv^T : \lambda_{uv}, \lambda_v \in \mathbb{R}_+ \right\} \supseteq \mathcal{C}^*$$

is an outer approximation of $\mathcal{C}^*$, and

$$\mathcal{O}_{\mathcal{P}}^* = \left\{ \sum_{v \in V_{\mathcal{P}}} \lambda_v vv^T : \lambda_v \in \mathbb{R}_+ \right\} \subseteq \mathcal{C}^*$$

is an inner approximation of $\mathcal{C}^*$. From Theorems 3.3 and 3.6 we immediately get that if $\{\mathcal{P}_\ell\}$ is a sequence of simplicial partitions of Δ^S with $\delta(\mathcal{P}_\ell) \to 0$, then the approximations converge, i.e.,

$$\mathcal{C}^* = \bigcap_{\ell \in \mathbb{N}} \mathcal{I}^*_{\mathcal{P}_\ell} \qquad \text{and} \qquad \mathcal{C}^* = \overline{\bigcup_{\ell \in \mathbb{N}} \mathcal{O}^*_{\mathcal{P}_\ell}}.$$

4. An adaptive approximation algorithm for copositive programs.

We now turn to the problem of solving an optimization problem over the copositive cone. The difficulty of such a problem lies in the cone condition. If the copositive cone is replaced by a linear inner or outer approximation, we get a linear program whose optimal value is a lower, respectively upper, bound of the optimal value of the original problem. We first state our algorithm and illustrate its behavior with a small example. After that, we study convergence of the algorithm.

4.1. Algorithm framework.

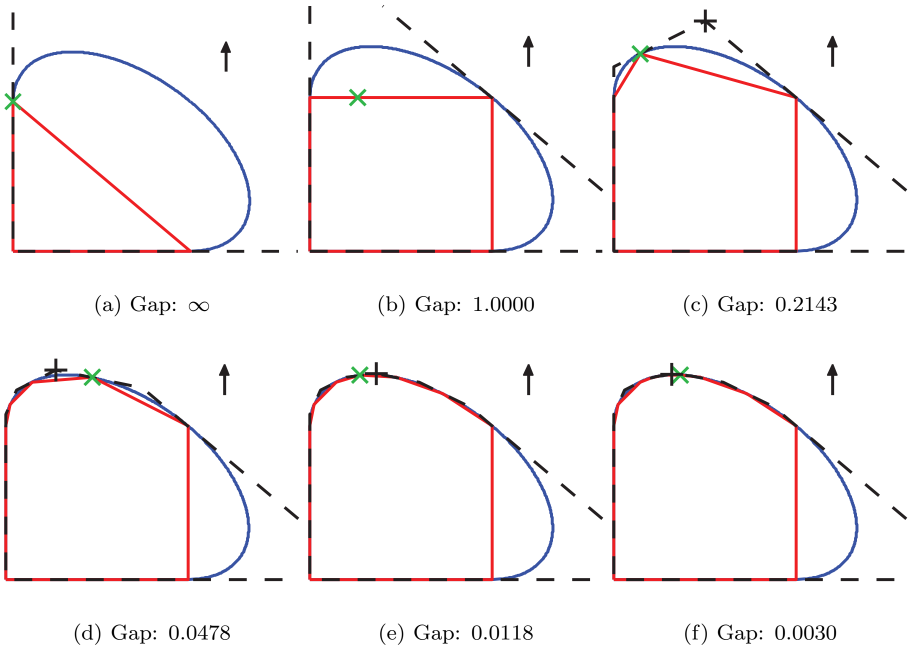
We state the algorithm for copositive programs of the form

$$\text{(CP)} \qquad \begin{array}{ll} \max & \langle C, X\rangle \\ \text{s. t.} & \langle A_i, X\rangle = b_i, \quad i = 1, \dots, m \\ & X \in \mathcal{C}. \end{array}$$

Given a solution accuracy $\varepsilon > 0$, Algorithm 1 computes an ε-optimal solution of (CP), i.e., a feasible solution X with $\frac{\langle C, X^{\text{opt}}\rangle - \langle C, X\rangle}{1 + |\langle C, X^{\text{opt}}\rangle| + |\langle C, X\rangle|} < \varepsilon$. Note that the algorithm also provides the valid lower (resp. upper) bounds $\langle C, X^{\mathcal{I}}\rangle$ (resp. $\langle C, X^{\mathcal{O}}\rangle$).

Algorithm 1 ε-approximation algorithm for (CP).

```
 1: set $\mathcal{P} = \{\Delta^S\}$
 2: solve the inner LP
        (ILP)    $\max\ \langle C, X\rangle$
                 $\text{s. t.}\ \langle A_i, X\rangle = b_i, \quad i = 1, \dots, m$
                 $X \in \mathcal{I}_{\mathcal{P}}$
    let $X^{\mathcal{I}}$ denote the solution of this problem
 3: solve the outer LP
        (OLP)    $\max\ \langle C, X\rangle$
                 $\text{s. t.}\ \langle A_i, X\rangle = b_i, \quad i = 1, \dots, m$
                 $X \in \mathcal{O}_{\mathcal{P}}$
    let $X^{\mathcal{O}}$ denote the solution of this problem
 4: if $\frac{\langle C, X^{\mathcal{O}}\rangle - \langle C, X^{\mathcal{I}}\rangle}{1 + |\langle C, X^{\mathcal{O}}\rangle| + |\langle C, X^{\mathcal{I}}\rangle|} < \varepsilon$, then
 5:     STOP: $X^{\mathcal{I}}$ is an $\varepsilon$-optimal solution of (CP)
 6: end if
 7: choose $\Delta \in \mathcal{P}$
 8: bisect $\Delta = \Delta^1 \cup \Delta^2$
 9: set $\mathcal{P} \leftarrow \mathcal{P} \setminus \{\Delta\} \cup \{\Delta^1, \Delta^2\}$
10: go to 2.
```


(a) Gap: ∞

(b) Gap: 1.0000

(c) Gap: 0.2143

(d) Gap: 0.0478

(e) Gap: 0.0118

(f) Gap: 0.0030

FIG. 4.1. *Iterations for Example* 4.1.

In this prototype algorithm it is not specified how a simplex is selected in Step 7 or how the bisection is performed in Step 8. Here lies some freedom which allows us to guide the partitioning procedure adaptively in a way that is advantageous for the optimization. The choice of the partitions also influences the convergence behavior and finiteness of the algorithm.

We will discuss these points later in more detail in section 5. First, we illustrate the behavior of this algorithm with a small example:

Example 4.1. Consider the problem

$$\begin{aligned} \max \quad & \left\langle \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, X \right\rangle \\ \text{s. t.} \quad & \left\langle \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, X \right\rangle = 2 \\ & X \in \mathcal{C}. \end{aligned}$$

The sequence of iterations is displayed in Figure 4.1. In this simple example, the cone $\mathcal{C}$ of symmetric copositive matrices is a cone in $\mathbb{R}^3$. The feasible set is therefore a two-dimensional set which is displayed with the curved line in the figure. The upward arrow indicates the direction of the objective function. The solid line represents the inner approximating cones $\mathcal{I}_{\mathcal{P}}$, whereas the dashed lines represent the outer approximating cones $\mathcal{O}_{\mathcal{P}}$. The symbols × and + indicate the subproblem optimal solutions (computed by an interior point solver in this example). For the starting partition, the outer approximation is unbounded, a consequence of the fact that $\mathcal{O}_{\{\Delta^S\}}$ is not pointed. "Gap" denotes the difference $\langle C, X^{\mathcal{O}}\rangle - \langle C, X^{\mathcal{I}}\rangle$.

Observe that the feasible set is approximated with high accuracy in those parts which are important for the optimization, whereas the irrelevant parts are not refined.

4.2. Convergence. We proceed to investigate convergence of Algorithm 1. Convergence of this algorithm relies on convergence of the approximating cones $\mathcal{I}_{\mathcal{P}_\ell}$ and $\mathcal{O}_{\mathcal{P}_\ell}$ as described in section 3. Therefore, we need the assumption that $\delta(\mathcal{P}_\ell) \to 0$ as $\ell \to \infty$ for the partitions generated in the algorithm. Note that by construction of Algorithm 1 we have the monotonicity $\mathcal{I}_{\mathcal{P}_\ell} \subset \mathcal{I}_{\mathcal{P}_{\ell+1}}$ and $\mathcal{O}_{\mathcal{P}_{\ell+1}} \subset \mathcal{O}_{\mathcal{P}_\ell}$.

Further, observe that feasibility of (CP) implies feasibility of (OLP), but not necessarily feasibility of (ILP). Therefore, we need assumptions which imply that (ILP) will eventually become feasible in the course of the iterations. This is done by assuming that there exists a strictly feasible point by which we mean a solution $\widehat{X}$ of the linear system $\langle A_i, \widehat{X}\rangle = b_i$ for all $i = 1, \ldots, m$ with $\widehat{X} \in \operatorname{int}(\mathcal{C})$.

Moreover, the feasible set of the outer approximation problem (OLP) may be unbounded even if the feasible set of (CP) is compact; cf. Example 4.1. The next theorem shows, however, that in this case the feasible set of the outer approximation eventually becomes bounded as the algorithm progresses.

THEOREM 4.2. *Assume the feasible set of* (CP) *is bounded and contains a strictly feasible point. Assume further that in every iteration of Algorithm* 1 *the selection of* Δ *and the bisection into* $\Delta = \Delta^1 \cup \Delta^2$ *is performed in such a way that the generated sequence* $\{\mathcal{P}_\ell\}$ *of partitions fulfills* $\delta(\mathcal{P}_\ell) \to 0$ *as* $\ell \to \infty$. *Let* (ILP$_{\mathcal{P}_\ell}$) *(resp.* (OLP$_{\mathcal{P}_\ell}$)*) denote the inner (resp. outer) approximation LPs corresponding to partition* $\{\mathcal{P}_\ell\}$ *in Steps* 2 *and* 3 *of Algorithm* 1, *and let* $X^{\mathcal{I}_\ell}$ *(resp.* $X^{\mathcal{O}_\ell}$*) denote the optimal solutions of* (ILP$_{\mathcal{P}_\ell}$) *(resp.* (OLP$_{\mathcal{P}_\ell}$)*). Then*

(a) *there exists* $\ell_0 \in \mathbb{N}$ *such that the feasible set of* (ILP$_{\mathcal{P}_\ell}$) *is nonempty and bounded for any* $\ell \geq \ell_0$*; the corresponding optimal solution* $X^{\mathcal{I}_\ell}$ *is then feasible for* (CP)*;*

(b) *there exists* $\ell_1 \in \mathbb{N}$ *such that the feasible set of* (OLP$_{\mathcal{P}_\ell}$) *is nonempty and bounded for any* $\ell \geq \ell_1$*;*

(c) *both sequences* $\{X^{\mathcal{I}_\ell}\}$ *and* $\{X^{\mathcal{O}_\ell}\}$ *have accumulation points, and any accumulation point of either sequence is optimal for* (CP).

Proof. Let X^* denote an optimal solution of (CP), and let $\widehat{X}$ be a strictly feasible solution of (CP). Let $\mathcal{A} := \{X \in \mathcal{S} : \langle A_i, X\rangle = b_i \text{ for } i = 1, \ldots, m\}$ denote the subspace of points satisfying the linear constraints. We use the notation $\max(P)$ to denote the optimal value of a maximization problem (P).

(a) Since $\mathcal{I}_{\mathcal{P}_\ell} \subseteq \mathcal{C}$ for any $\ell \in \mathbb{N}$, the feasible sets of (ILP$_{\mathcal{P}_\ell}$) are all bounded. As $\widehat{X}$ is a strictly copositive matrix, it follows from Theorem 3.3 that there exists $\ell_0 \in \mathbb{N}$ such that $\widehat{X} \in \mathcal{I}_{\mathcal{P}_{\ell_0}}$. Since also $\widehat{X} \in \mathcal{A}$, the feasible set $\mathcal{A} \cap \mathcal{I}_{\mathcal{P}_{\ell_0}}$ of (ILP$_{\mathcal{P}_{\ell_0}}$) is nonempty, and so are the feasible sets of (ILP$_{\mathcal{P}_\ell}$) for all $\ell \geq \ell_0$. Therefore, any such (ILP$_{\mathcal{P}_\ell}$) has an optimal solution $X^{\mathcal{I}_\ell}$ which, by $\mathcal{I}_{\mathcal{P}_\ell} \subseteq \mathcal{C}$, is feasible for (CP).

(b) Since (CP) is feasible, the feasible set $\mathcal{A} \cap \mathcal{O}_{\mathcal{P}_\ell}$ of any (OLP$_{\mathcal{P}_\ell}$) is nonempty, as well. To show boundedness, assume by contradiction that $\mathcal{A} \cap \mathcal{O}_{\mathcal{P}_\ell}$ is unbounded for all $\ell \in \mathbb{N}$. Take an arbitrary $X \in \mathcal{A} \cap \mathcal{C}$. Then by polyhedrality of $\mathcal{A} \cap \mathcal{O}_{\mathcal{P}_\ell}$, the set

$$\mathcal{D}_\ell := \{D \in \mathcal{S} : \|D\| = 1, X + \alpha D \in \mathcal{A} \cap \mathcal{O}_{\mathcal{P}_\ell} \text{ for all } \alpha \geq 0\}$$

is nonempty for any ℓ. The monotonicity $\mathcal{O}_{\mathcal{P}_\ell} \supseteq \mathcal{O}_{\mathcal{P}_{\ell+1}}$ implies $\mathcal{D}_\ell \supseteq \mathcal{D}_{\ell+1}$ for all ℓ. Moreover, closedness of $\mathcal{A} \cap \mathcal{O}_{\mathcal{P}_\ell}$ implies closedness of $\mathcal{D}_\ell$, whence all $\mathcal{D}_\ell$ are compact. Using a theorem of Cantor, we infer that the intersection of all $\mathcal{D}_\ell$ is nonempty, i.e., there exists $\widehat{D} \in \bigcap_{\ell \in \mathbb{N}} \mathcal{D}_\ell$. But then $\{X + \alpha\widehat{D} :$

$\alpha \geq 0\} \subset \mathcal{A} \cap \mathcal{O}_{\mathcal{P}_\ell}$ for all ℓ, and therefore

$$\{X + \alpha\widehat{D} : \alpha \geq 0\} \subset \bigcap_{\ell\in\mathbb{N}} (\mathcal{A} \cap \mathcal{O}_{\mathcal{P}_\ell}) = \mathcal{A} \cap \bigcap_{\ell\in\mathbb{N}} \mathcal{O}_{\mathcal{P}_\ell} = \mathcal{A} \cap \mathcal{C}.$$

This contradicts the assumption that $\mathcal{A} \cap \mathcal{C}$ is bounded, and consequently there exists $\ell_1 \in \mathbb{N}$ such that $\mathcal{A} \cap \mathcal{O}_{\mathcal{P}_\ell}$ is bounded for all $\ell \geq \ell_1$.

(c) We first show the statement for the sequence $\{X^{\mathcal{I}_\ell}\}$. An accumulation point exists because the sequence $\{X^{\mathcal{I}_\ell}\}_{\ell\geq\ell_0}$ is contained in the compact feasible set of (CP). Let X^a denote an accumulation point of $\{X^{\mathcal{I}_\ell}\}$. From the inner approximation property, we have $\langle C, X^{\mathcal{I}_\ell}\rangle \leq \max(\text{CP})$ for all $\ell \geq \ell_0$, so the same must hold for the accumulation point, i.e.,

$$\langle C, X^a\rangle \leq \max(\text{CP}). \tag{4.1}$$

To see the converse, consider points $Z_\lambda := \lambda X^* + (1-\lambda)\widehat{X}$ for $\lambda \in (0,1)$. By construction, Z_λ is strictly feasible for (CP), i.e., strictly copositive. By Theorem 3.3, for each such λ there exists $\ell_\lambda \in \mathbb{N}$ such that $Z_\lambda \in \mathcal{I}_{\mathcal{P}_\ell}$ for all $\ell \geq \ell_\lambda$. Therefore,

$$\langle C, X^a\rangle = \sup_{\ell\in\mathbb{N}} \max(\text{ILP}_{\mathcal{P}_\ell}) \geq \lim_{\lambda\nearrow 1} \langle C, Z_\lambda\rangle = \langle C, X^*\rangle = \max(\text{CP}).$$

Combined with (4.1), this proves that X^a is optimal for (CP).
Next, we show the statement for the sequence $\{X^{\mathcal{O}_\ell}\}$. An accumulation point exists because the sequence $\{X^{\mathcal{O}_\ell}\}_{\ell\geq\ell_1}$ is contained in the compact feasible set of $(\text{OLP}_{\mathcal{P}_{\ell_1}})$. Let X^A denote an accumulation point of $\{X^{\mathcal{O}_\ell}\}$. From the outer approximation property we have $\langle C, X^{\mathcal{O}_\ell}\rangle \geq \max(\text{CP})$ for all ℓ, so the same must hold for the accumulation point, i.e.,

$$\langle C, X^A\rangle \geq \max(\text{CP}).$$

The reverse inequality follows from $X^A \in \bigcap_{\ell\in\mathbb{N}} \mathcal{O}_{\mathcal{P}_\ell} = \mathcal{C}$, which shows that X^A is an optimal solution of (CP). □

If the assumptions of Theorem 4.2 are not fulfilled, the situation gets more involved:

If the feasible set of (CP) is empty because $\mathcal{A} = \emptyset$, then (OLP) is infeasible in the very first iteration. If the feasible set of (CP) is empty because $\mathcal{A}$ does not intersect $\mathcal{C}$, then obviously all inner approximations $(\text{ILP}_{\mathcal{P}_\ell})$ are infeasible, as well, but unfortunately infeasibility of (ILP) is no certificate of infeasibility of (CP). Detection of infeasibility of (CP) is only possible if (OLP) is infeasible. We observed that in numerical examples, infeasibility of (CP) was detected through infeasibility of an outer approximation $(\text{OLP}_{\mathcal{P}_\ell})$ in the course of the iterations. In exceptional cases, however, this may fail: If the set $\mathcal{A}$ is parallel to an face of $\mathcal{C}$ induced by the hyperplane $\mathcal{H} := \{X \in \mathcal{C} : v^T X v = 0\}$, then the outer approximations remain feasible unless the partitioning process eventually generates v as a vertex in $V_{\mathcal{P}}$ by pure chance.

If (CP) is feasible but has no strictly feasible point, i.e., the feasible set is contained in the boundary of $\mathcal{C}$, then clearly all outer approximations are feasible, but the inner approximations are most likely all infeasible, unless the inner approximation happens to touch the boundary of $\mathcal{C}$ in the right portion.

If (CP) is unbounded, then in most practical cases the inner approximation will also be unbounded in some finite iteration. In any case, we have the following:

Theorem 4.3. *Assume that* (CP) *has a strictly feasible solution. If* (CP) *is unbounded, then*

$$\lim_{\ell\to\infty} \max(\mathrm{ILP}_{\mathcal{P}_\ell}) = \infty.$$

Proof. If (CP) is unbounded, then there exists a sequence $\{\tilde{X}_n\}$ of feasible solutions such that $\lim_{n\to\infty}\langle C, \tilde{X}_n\rangle = \infty$. Let $\widehat{X}$ be a strictly feasible solution of (CP). Then $X_n := \frac{1}{2}\widehat{X} + \frac{1}{2}\tilde{X}_n$ is strictly feasible for all $n \in \mathbb{N}$, and

$$\lim_{n\to\infty} \langle C, X_n\rangle = \infty.$$

By Theorem 2.4, for each $n \in \mathbb{N}$ there exists an index ℓ_n such that $X_n \in \mathcal{I}_{\mathcal{P}_{\ell_n}}$. Now the assertion follows. □

5. Fine-tuning the algorithm. As mentioned, Algorithm 1 contains freedom in Steps 7 and 8 where the partitioning process of Δ^S is guided. In this section, we discuss how the partitioning is performed in each iteration. Moreover, we consider the problem of redundancies appearing in the subproblems, and we show how the starting partition can be tuned given a known heuristic solution.

5.1. Selecting and subdividing $\boldsymbol{\Delta}$. Generating a sequence of partitions $\{\mathcal{P}_\ell\}$ of Δ^S with $\delta(\mathcal{P}_\ell) \to 0$ results in a sequence of cones $\{\mathcal{I}_{\mathcal{P}_\ell}\}$ and $\{\mathcal{O}_{\mathcal{P}_\ell}\}$ that approximate $\mathcal{C}$ uniformly arbitrarily well. For optimization purposes, however, this is not efficient. We would rather like to obtain a high approximation accuracy in those parts of the feasible set which are relevant for the optimization, and we would like to invest as little computational effort as possible into uninteresting parts. Therefore, we use information gained through the objective function.

First note that, once an edge $\{u, v\}$ is chosen for bisection, it makes sense to partition all simplices containing this edge at the same time. Otherwise, $\{u, v\}$ would remain an edge in $E_\mathcal{P}$, and the corresponding cone $\mathcal{I}_\mathcal{P}$ would not change. We bisect all simplices at the new vertex $w := \lambda u + (1 - \lambda)v$. Experiments with various values of λ showed no big effects, whence we simply use $\lambda = \frac{1}{2}$, i.e., we perform midpoint bisection throughout.

Furthermore, observe that, when an edge $\{u, v\} \in E_\mathcal{P}$ is splitted, the corresponding inequality $u^T X v \geq 0$ is removed from the system describing $\mathcal{I}_\mathcal{P}$ and replaced by several new inequalities (cf. Example 3.2). All other inequalities present before the bisection step are also present after bisection. As the optimal value of an LP does not change if an inactive constraint is removed, it makes sense to consider for splitting only edges $\{u, v\} \in E_\mathcal{P}$ corresponding to active constraints, i.e., edges with $u^T X^\mathcal{I} v = 0$ (where $X^\mathcal{I}$ is the solution of (ILP) in Step 2 of the algorithm). Only in this way can we hope to improve the solution of the inner approximation.

We call an edge $\{u, v\} \in E_\mathcal{P}$ with $u^T X^\mathcal{I} v = 0$ an *active edge* and choose in Step 7 of Algorithm 1, the longest of the edges active in $X^\mathcal{I}$ for bisection. The next lemma states that such an edge always exists:

Lemma 5.1. *In Step* 7 *of Algorithm* 1*, there always exists* $\{u, v\} \in E_\mathcal{P}$ *with* $u^T X^\mathcal{I} v = 0$.

Proof. The proof relies on the fact that the optimal value of an LP does not change if constraints which are inactive at the solution are omitted. The solution $X^\mathcal{I}$ of problem (ILP) clearly fulfills $X^\mathcal{I} \in \mathcal{I}_\mathcal{P}$, i.e., $u^T X^\mathcal{I} v \geq 0$ for all $\{u, v\} \in E_\mathcal{P}$ and $v^T X^\mathcal{I} v \geq 0$ for all $v \in V_\mathcal{P}$. Assume by contradiction that all constraints $u^T X v \geq 0$ with $\{u, v\} \in E_\mathcal{P}$ are inactive. Then the solution of (ILP) does not change if those constraints are omitted. But this means that $X^{\mathcal{I}}$ also solves (OLP), whence the algorithm stops in Step 4 with a zero gap. $\square$

Selecting in Step 7 of Algorithm 1 one of the longest active edges may not result in a sequence of partitions $\{\mathcal{P}_\ell\}$ with $\delta(\mathcal{P}_\ell) \to 0$. Instead of $\delta(\mathcal{P}_\ell)$, we now have to monitor the length $\alpha(\mathcal{P}_\ell)$ of the longest active edge in $\mathcal{P}_\ell$. If this quantity goes to zero, then the algorithm converges:

THEOREM 5.2. *Assume that* (CP) *has a strictly feasible point and a bounded feasible set. Let $\{\mathcal{P}_\ell\}$ be a sequence of simplicial partitions generated from $\mathcal{P}_0 = \{\Delta^S\}$ by bisecting one of the respective longest active edges $\{u_\ell, v_\ell\}$. Assume further that the length $\alpha(\mathcal{P}_\ell)$ of the respective longest edge in $\mathcal{P}_\ell$ goes to zero as $\ell \to \infty$. Then*

$$\lim_{\ell\to\infty} \max(\mathrm{ILP}_{\mathcal{P}_\ell}) = \max\,(\mathrm{CP}).$$

Proof. Convergence Theorem 4.2 cannot be directly applied since we do not necessarily have $\delta(\mathcal{P}_\ell) \to 0$ as $\ell \to \infty$. However, we show that there exists a sequence $\{\mathcal{R}_\ell\}$ of partitions which fulfills $\max(\mathrm{ILP}_{\mathcal{P}_\ell}) = \max(\mathrm{ILP}_{\mathcal{R}_\ell})$ for all $\ell \in \mathbb{N}$, and $\delta(\mathcal{R}_\ell) \to 0$ as $\ell \to \infty$.

Consider $\mathcal{P}_\ell$ for some $\ell \in \mathbb{N}$, and let X^ℓ be the solution of the inner approximation problem $(\mathrm{ILP}_{\mathcal{P}_\ell})$. Since $\alpha(\mathcal{P}_\ell)$ denotes the length of the longest active edge in $\mathcal{P}_\ell$, edges in $\mathcal{P}_\ell$ with length greater than $\alpha(\mathcal{P}_\ell)$ are necessarily inactive.

We construct $\mathcal{R}_\ell$ from $\mathcal{P}_\ell$ by splitting all edges in $\mathcal{P}_\ell$ which are longer than $\alpha(\mathcal{P}_\ell)$. If necessary, we repeat this process until no edge of length greater than $\alpha(\mathcal{P}_\ell)$ remains. All edges which are in $\mathcal{P}_\ell$ but not in $\mathcal{R}_\ell$ were splitted in the process of constructing $\mathcal{R}_\ell$. Therefore, they had length greater than $\alpha(\mathcal{P}_\ell)$ and thus were inactive with respect to the optimal solution of $(\mathrm{ILP}_{\mathcal{P}_\ell})$. Let (AUX) be the linear program which has the same constraints as $(\mathrm{ILP}_{\mathcal{P}_\ell})$ except for those induced by an edge from $E_{\mathcal{P}_\ell} \setminus E_{\mathcal{R}_\ell}$. Removing inactive constraints from an LP does not change the optimal value, so $\max(\mathrm{ILP}_{\mathcal{P}_\ell}) = \max(\mathrm{AUX})$. Adding constraints cannot increase the optimal value, so $\max(\mathrm{ILP}_{\mathcal{R}_\ell}) \le \max(\mathrm{AUX}) = \max(\mathrm{ILP}_{\mathcal{P}_\ell})$. On the other hand, $\mathcal{R}_\ell$ is by construction a refinement of $\mathcal{P}_\ell$, so we immediately get $\max(\mathrm{ILP}_{\mathcal{P}_\ell}) \le \max(\mathrm{ILP}_{\mathcal{R}_\ell})$ from Lemma 3.1. Consequently, the two values are equal.

Observe that $\delta(\mathcal{R}_\ell) \le \alpha(\mathcal{P}_\ell)$. Now the assumption $\alpha(\mathcal{P}_\ell) \to 0$ implies $\delta(\mathcal{R}_\ell) \to 0$ as $\ell \to \infty$, so $\{\mathcal{R}_\ell\}$ fulfills the prerequisites of Theorem 4.2, and hence

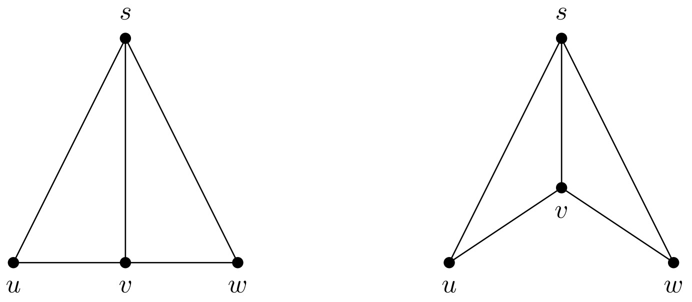
$$\lim_{\ell\to\infty} \max(\mathrm{ILP}_{\mathcal{P}_\ell}) = \lim_{\ell\to\infty} \max(\mathrm{ILP}_{\mathcal{R}_\ell}) = \max\,(\mathrm{CP}),$$

and the proof is complete. $\square$

In practical implementations of our algorithm, it may happen that $\alpha(\mathcal{P}_\ell) \not\to 0$ such that convergence is not guaranteed. However, we never observed nonconvergence in our test instances (cf. section 6). If convergence does not occur, it may be necessary to alternate between bisection of the longest edge and bisection of the longest active edge to maintain convergence.

Observe that Theorem 5.2 ensures convergence of the inner approximations but not of the outer approximations. Therefore, the adaptive algorithm which splits along the longest active edges might have a positive gap. In our experiments, this seemed unproblematic. However, if the outer approximations fail to converge, a remedy is to use additional points for the outer approximation in such a way that these points eventually become dense in Δ^S.

5.2. Handling redundancies. Given a partition $\mathcal{P}$, the description $\mathcal{O}_{\mathcal{P}} := \{A \in \mathcal{S} : v^T A v \ge 0 \text{ for all } v \in V_{\mathcal{P}}\}$ does not contain any redundant inequalities.



(a) The inequalities induced by vertex v and edge $\{s, v\}$ are redundant.

(b) The inequality induced by edge $\{s, v\}$ is redundant.

FIG. 5.1. *Two situations where redundancies occur.*

This follows from the fact that for any $v \in V_{\mathcal{P}}$ and $\mathcal{H} := \{X \in \mathcal{S} : v^T X v = 0\}$ the set $\mathcal{H} \cap \mathcal{O}_{\mathcal{P}}$ is a facet of $\mathcal{O}_{\mathcal{P}}$. Indeed, assume $v^T X v = 0$ does not define a facet of $\mathcal{O}_{\mathcal{P}}$. Then there exist vertices $v_1, \dots, v_s \in V_{\mathcal{P}}$ different from v, and $\alpha \in \mathbb{R}^s_+$ such that

$$vv^T = \sum_{i=1}^{s} \alpha_i v_i v_i^T.$$

But this contradicts the fact that vv^T is an extremal ray of $\mathcal{C}^*$.

Consequently, the description of $\mathcal{O}_{\mathcal{P}}$ contains no redundancies. Note that every bisection step generates precisely one additional vertex. Therefore, a partition $\mathcal{P}$ with m simplices has $|V_{\mathcal{P}}| = n + m$ vertices. This means that the size of the linear systems describing $\mathcal{O}_{\mathcal{P}}$ grows moderately during the iterations of our algorithm.

In contrast to this, the representation

$$\begin{aligned} \mathcal{I}_{\mathcal{P}} := \{A \in \mathcal{S} : & \ v^T A v \geq 0 \ \text{ for all } v \in V_{\mathcal{P}}, \\ & \ u^T A v \geq 0 \ \text{ for all } \{u, v\} \in E_{\mathcal{P}}\} \end{aligned}$$

contains a lot of redundancy, as has already been shown in Example 3.2. Redundancies are generated in situations as the following:

Example 5.3.

(a) For some partition $\mathcal{P}$, let $s, u, v, w \in V_{\mathcal{P}}$, and let $v = \lambda u + (1 - \lambda) w$ with some $\lambda \in (0, 1)$. Assume that $\{s, u\}$, $\{s, v\}$, $\{s, w\}$, $\{u, v\}$, $\{v, w\} \in E_{\mathcal{P}}$. See Figure 5.1(a) for a picture of this setting.
Then the inequalities $u^T A v \geq 0$ and $w^T A v \geq 0$ imply

$$(\lambda u + (1 - \lambda) w)^T A v \geq 0 \qquad \Leftrightarrow \quad v^T A v \geq 0,$$

whence the latter inequality is redundant. Likewise, $u^T A s \geq 0$ and $w^T A s \geq 0$ imply $v^T A s \geq 0$, showing that this is a redundant inequality, too.

(b) The situation is similar if we have $v = \lambda s + \mu u + (1 - \lambda - \mu) w$ with $\lambda, \mu, (1 - \lambda - \mu) > 0$ (see Figure 5.1(b)).
As before, $s^T A v \geq 0$ is a convex combination of the inequalities $s^T A s \geq 0$, $s^T A u \geq 0$, and $s^T A w \geq 0$, and is therefore redundant.
More complicated examples can be constructed analogously.

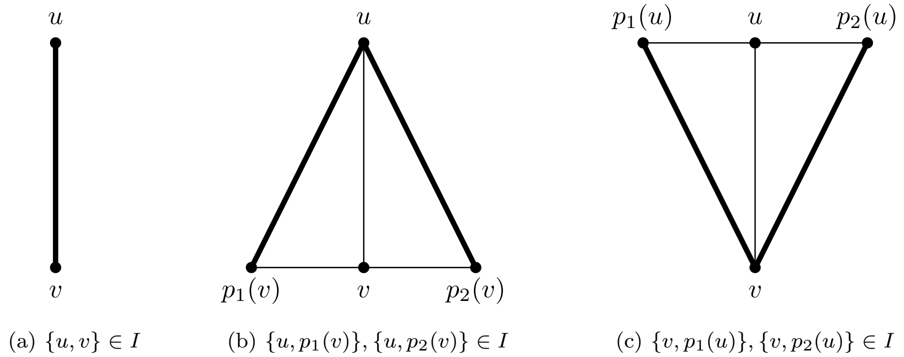

u
v
u
$p_1(v)$
v
$p_2(v)$
$p_1(u)$
u
$p_2(u)$
v
(a) $\{u, v\} \in I$
(b) $\{u, p_1(v)\}, \{u, p_2(v)\} \in I$
(c) $\{v, p_1(u)\}, \{v, p_2(u)\} \in I$

FIG. 5.2. *The three cases where $e\{u, v\}$ is true. Edges which belong to the set I are drawn bold.*

Observe that situations like in the example occur in abundance by construction of the partition. In order to speed up our algorithm, it is therefore essential to find a way to deal with these redundancies. Note that an n-dimensional simplex has $n+1$ vertices and $\binom{n+1}{2}$ edges. Hence, for a partition $\mathcal{P}$ with m simplices, we have $|V_{\mathcal{P}}| + |E_{\mathcal{P}}| = \frac{1}{2}m(n+1)(n+2)$, so the full system describing $\mathcal{I}_{\mathcal{P}}$ would have that many constraints. This number grows too quickly, so we have to be careful to keep the system irredundant.

Unfortunately, we can not simply eliminate redundant inequalities and forget about them, since a redundant constraint may become irredundant in later iterations. This happens if a redundant inequality is a convex combination of others, and an edge corresponding to one of the "parent inequalities" is bisected in a later iteration. This phenomenon makes it necessary to keep track of the history of all vertices and edges in the partition. We do this by introducing suitable maps.

DEFINITION 5.4. *Assume that for all ℓ, partition $\mathcal{P}_{\ell+1}$ is generated from $\mathcal{P}_\ell$ through bisection of an edge in $E_{\mathcal{P}_\ell}$. We call two vertices $u, w \in V_{\mathcal{P}}$* parents *of v, if the edges $\{u, v\}$ and $\{v, w\}$ are edges of the partition $\mathcal{P}$ and there exists $\lambda \in (0, 1)$ such that $v = \lambda u + (1 - \lambda)w$. We call a map*

$$p : V_{\mathcal{P}} \to V_{\mathcal{P}} \times V_{\mathcal{P}}$$

with the property that $p(v)$ are parents of v a parent map.

For a given set $I \subset E_{\mathcal{P}}$ and for $\{u, v\} \in E_{\mathcal{P}}$, we define the boolean function e as

$$e : \{u, v\} \mapsto \begin{cases} \text{true} & \textit{if } \{u, v\} \in I \\ & \textit{or } \{u, p_1(v)\}, \{u, p_2(v)\} \in I \\ & \textit{or } \{v, p_1(u)\}, \{v, p_2(u)\} \in I \\ \text{false} & \textit{else.} \end{cases}$$

(See Figure 5.2 for an illustration). We write e_I if it is necessary to emphasize that e depends on the set I.

Note that for a partition $\mathcal{P}$ there may exist several parent maps. In what follows it does not matter which one is used. The most natural one (which we use in our implementation) is the "historical" parent map; i.e., if edge $\{u, v\}$ is splitted at the point w, we define $p(w) = (u, v)$.

The next lemma states that if $e\{u, v\}$ is true, then $\{u, v\}$ is a redundant edge.

LEMMA 5.5. *Let p be a parent map and let $u, v \in V_{\mathcal{P}}$. If $e\{u, v\} = \text{true}$, then there exist $\{u_1, v_1\}, \{u_2, v_2\} \in I$ and $\lambda \in (0, 1)$ such that*

$$u^T Av = \lambda u_1^T Av_1 + (1 - \lambda)u_2^T Av_2 \quad \textit{for all } A \in \mathcal{S}.$$

Proof. This follows immediately from the definitions. □

DEFINITION 5.6. *Let $\{u, v\} \in I$. An edge $\{s, t\} \in E_{\mathcal{P}}$ is said to* depend on $\{u, v\}$, *if $e_I(s, t) = \text{true}$ and $e_{I\setminus\{\{u,v\}\}}(s, t) = \text{false}$.*

We use the set I to generate a less redundant description of $\mathcal{I}_{\mathcal{P}}$. We start with $I = \{\{e_i, e_j\} : i, j = 1, \ldots, n\}$. If the partition is refined by splitting the edge $\{u, w\} \in I$ at the point $v = \lambda u + (1 - \lambda)w$ with some $\lambda \in (0, 1)$, then the set I is updated as follows:

- remove the edge $\{u, w\}$ from I,
- insert the edges $\{u, v\}$ and $\{v, w\}$ into I,
- for all $\{s, t\} \in E_{\mathcal{P}}$: if $\{s, t\}$ depends on $\{u, w\}$, then insert $\{s, t\}$ into I.

The next lemma shows that the set I is indeed sufficient to describe the cone $\mathcal{I}_{\mathcal{P}}$:

LEMMA 5.7. *If $\mathcal{P}$ is generated from the standard simplex by bisections and the updating procedure for I described above is used, then*

$$\mathcal{I}_{\mathcal{P}} = \mathcal{I}_I := \{X \in \mathcal{S} : u^T Xv \geq 0 \textit{ for all } \{u, v\} \in I\}.$$

Proof. We have $I \subseteq E_{\mathcal{P}}$ because the only edge leaving $E_{\mathcal{P}}$ also leaves I, and every edge inserted into I is an element of $E_{\mathcal{P}}$. Thus, $\mathcal{I}_I \supseteq \mathcal{I}_{\mathcal{P}}$.

Let $\{u, v\} \in E_{\mathcal{P}}$. Then $\{u, v\} \in I$ or $e\{u, v\} = \text{true}$. Obviously the update procedure maintains this property. Using Lemma 5.5, it follows that $\mathcal{I}_I \subseteq \mathcal{I}_{\mathcal{P}}$. □

The third point of the update procedure requires knowledge of $E_{\mathcal{P}}$, whence we have to store also this information. The set $E_{\mathcal{P}}$ can also be updated efficiently:

Set $E = \{\{e_i, e_j\} : i, j = 1, \ldots, n;\ i \neq j\}$. Then obviously $E = E_{\{\Delta^S\}}$. If an edge $\{u, v\}$ is bisected at a point w, the set E is updated as follows:

- remove the edge $\{u, v\}$ from E,
- insert $\{u, w\}$ and $\{w, v\}$ into E,
- if $\{u, s\} \in E$ and $\{v, s\} \in E$, then insert $\{w, s\}$ into E.

The next lemma implies that this update procedure works, i.e., $E = E_{\mathcal{P}}$.

LEMMA 5.8. *Let $\{u, v\}, \{v, w\}, \{w, u\} \in E_{\mathcal{P}}$. Then there is a simplex $\Delta \in \mathcal{P}$ such that u, v, and w are vertices of Δ.*

Proof. If $u, v, w \in \Delta^S$, then $\operatorname{conv}\{u, v, w\} \subset \Delta^S$. Since $\mathcal{P}$ is a partition of Δ^S, there exist $\Delta^1, \ldots, \Delta^m \in \mathcal{P}$ such that $\operatorname{conv}\{u, v, w\} \subseteq \bigcup_{i=1}^m \Delta^i$. Let m be minimal in the sense that $\operatorname{conv}\{u, v, w\}$ is not covered by any subset of $\{\Delta^1, \ldots, \Delta^m\}$ and assume $m > 1$. Then there exists a vertex $s \in V_{\{\Delta^1,\ldots,\Delta^m\}} \setminus \{u, v, w\}$ with $s \in \operatorname{conv}\{u, v, w\}$. Since $\mathcal{P}$ is constructed through bisections, there must be a vertex on one of the edges $\{u, v\}, \{v, w\}, \{w, u\}$. This contradicts $\{u, v\}, \{v, w\}, \{w, u\} \in E_{\mathcal{P}}$. □

5.3. Tuning the starting partition.

Many interesting copositive programs arise from dualization of a completely positive program of the form

$$(\text{CP}^*) \qquad \begin{array}{ll} \min & \langle C, X\rangle \\ \text{s.t.} & \langle A_i, X\rangle = b_i, \quad i \in \{1, \ldots, m\} \\ & X \in \mathcal{C}^*. \end{array}$$

This holds in particular for many combinatorial problems. For example, the stability number $\alpha(G)$ of a graph $G = (V_G, E_G)$ fulfills (cf. [15])

$$\frac{1}{\alpha(G)} = \min\{\langle Q, X\rangle : \langle E, X\rangle = 1, X \in \mathcal{C}^*\},$$

where $Q = (A_G + I)$ and A_G is the adjacency matrix of G. Often a good feasible solution X of (CP*) can be obtained through some heuristic procedure. For instance, for any stable set $S \subset V_G$ take the vector x to be a suitably scaled version of the characteristic vector of S, and take X to be xx^T.

The dual of (CP*) is a copositive program of the form

$$\begin{aligned} \max \quad & \sum_{i=1}^m b_i y_i \\ \text{s. t.} \quad & Z = C - \sum_{i=1}^m y_i A_i, \\ & Z \in \mathcal{C}, y \in \mathbb{R}^m. \end{aligned}$$

This form is equivalent to (CP); i.e., each copositive program can be transformed from one form to the other.

By weak duality, for any feasible solution X of (CP*) the value $\langle C, X\rangle$ is an upper bound for the copositive problem, so it is desirable to initialize Algorithm 1 with an outer approximation $\mathcal{O}_{\mathcal{P}_0}$ yielding a bound not worse than $\langle C, X\rangle$. The next lemma states that this is always possible.

LEMMA 5.9. *Let X be feasible for* (CP*). *Then there exists a simplicial partition $\mathcal{P}$ such that the optimal value of the outer approximation* (OLP$_{\mathcal{P}}$) *is at most $\langle C, X\rangle$.*

Proof. Since $X \in \mathcal{C}^*$, it can be decomposed as $X = \sum_{k=1}^r v_k v_k^T$ with $v_1, \dots, v_r \in \mathbb{R}^n_+$. Set $w_k := \frac{v_k}{\|v_k\|_1}$. Then $w_k \in \Delta^S$, and therefore there exists a simplicial partition $\mathcal{P}$ such that $w_1, \dots, w_r \in V_{\mathcal{P}}$. Let (Z, y) be an optimal (dual) solution of the outer approximation, i.e., $(C - \sum_{i=1}^m y_i A_i) = Z \in \mathcal{O}_{\mathcal{P}}$. This implies

$$\begin{aligned} & w_k^T \left(C - \sum_{i=1}^m y_i A_i \right) w_k \geq 0 && \text{for all } k \in \{1, \dots, r\} \\ \Leftrightarrow \quad & \|v_k\|_1^2 w_k^T \left(C - \sum_{i=1}^m y_i A_i \right) w_k \geq 0 && \text{for all } k \in \{1, \dots, r\} \\ \Rightarrow \quad & \sum_{k=1}^r v_k^T \left(C - \sum_{i=1}^m y_i A_i \right) v_k \geq 0 \\ \Leftrightarrow \quad & \langle C, X\rangle - \sum_{i=1}^m y_i \langle A_i, X\rangle \geq 0 \\ \Leftrightarrow \quad & \langle C, X\rangle \geq \sum_{i=1}^m y_i b_i, \end{aligned}$$

which was to be shown. ▯

The partition $\mathcal{P}$ with $w_1, \dots, w_r \in V_{\mathcal{P}}$ can be generated iteratively by performing a radial subdivision as described in section 2 for each of the w_i at a time. Observe that in order to construct $\mathcal{P}$ it is necessary to have the decomposition $X = \sum_{k=1}^r v_k v_k^T$ (with $v_k \geq 0$ for all k) of the completely positive X. Determining this decomposition for general $X \in \mathcal{C}^*$ is a nontrivial task. However, in the combinatorial applications we have in mind (max clique, QAP, 0/1-quadratic programming), every feasible solution corresponds to a rank-one completely positive matrix X which can be utilized as described above.

6. Numerical results.

We implemented our algorithm in C++ and tested our implementation on a Pentium IV, 2.8GHz Linux machine with 1GB RAM. As a solver for the linear subproblems we used COIN-OR Linear Program Solver (CLP, Version 1.3.3).

We first report results obtained for some instances of the standard quadratic optimization problem, i.e., the problem of minimizing a nonconvex quadratic form over the standard simplex:

$$\min_{x \in \Delta^S} x^T Q x. \tag{6.1}$$

This is a well-studied problem which can be restated as the copositive program

$$\begin{array}{ll} \max & \lambda \\ \text{s.t.} & Q - \lambda E \in \mathcal{C}, \\ & \lambda \in \mathbb{R}. \end{array}$$

We first discuss the behavior of our algorithm on four examples taken from [4]. These authors solve the problems by using the LP-based approximations $\mathcal{C}^r$ and the SDP-based approximations $\mathcal{K}^r$ discussed in section 1.2. As mentioned there, these approaches provide only one-sided bounds on the optimum, without any information on the solution quality. An exception is [4], where approximation estimates are given. Those bounds, however, require knowledge or a good estimate of the range (maximum minus minimum) of the quadratic form over Δ^S.

The instances

$$Q_1 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix} \quad \text{and} \quad Q_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

are Examples 5.1 and 5.2 from [4] and correspond to the problem of determining the clique number in a pentagon and an icosahedron, respectively. The optimal values are $\frac{1}{2}$ for Q_1 and $\frac{1}{3}$ for Q_2, respectively. Our algorithm solves instance Q_1 to optimality (i.e., the gap between upper and lower bound is closed) in six iterations (0.01 sec) and instance Q_2 in 158 iterations (0.54 sec). In the latter instance we used the fact that the reciprocal of the optimal value is integer such that both lower and upper bounds could be rounded accordingly.

Using the approximating cones $\mathcal{C}^r$ and $\mathcal{K}^r$, Bomze and de Klerk [4] obtain the following results: for instance Q_1, they get the bound 0 when using $\mathcal{C}^0$ and $\frac{1}{3}$ when using $\mathcal{C}^1$. The cones $\mathcal{K}^0$ and $\mathcal{K}^1$ yield the bounds $\frac{1}{\sqrt{5}}$ and $\frac{1}{2}$, respectively. Hence, for this instance $\mathcal{K}^1$ yields the exact solution.

For instance Q_2, the respective numbers are 0 for the cone $\mathcal{C}^1$ and 0.309 for the cone $\mathcal{K}^1$. In this case, the bound obtained by using $\mathcal{K}^1$ is not exact. To use higher order approximations $\mathcal{K}^r$ with $r > 1$ is difficult, since the dimension of those problems is rapidly increasing. Bomze and de Klerk do not report a bound obtained by using $\mathcal{K}^2$.

The next instance

$$Q_3 = \begin{pmatrix} -14 & -15 & -16 & 0 & 0 \\ -15 & -14 & -12.5 & -22.5 & -15 \\ -16 & -12.5 & -10 & -26.5 & -16 \\ 0 & -22.5 & -26.5 & 0 & 0 \\ 0 & -15 & -16 & 0 & -14 \end{pmatrix}$$

is Example 5.3 from [4] and arises in a model in population genetics. Its optimal value is $-16\frac{1}{3}$. Bomze and de Klerk report the bounds -21 for $\mathcal{C}^1$, while cone $\mathcal{K}^1$ gives the exact result. Our algorithm takes 44 iterations (0.03 sec) to solve the problem with an accuracy of 10^{-6}.

Finally,

$$Q_4 = \begin{pmatrix} 0.9044 & 0.1054 & 0.5140 & 0.3322 & 0 \\ 0.1054 & 0.8715 & 0.7385 & 0.5866 & 0.9751 \\ 0.5140 & 0.7385 & 0.6936 & 0.5368 & 0.8086 \\ 0.3322 & 0.5866 & 0.5368 & 0.5633 & 0.7478 \\ 0 & 0.9751 & 0.8086 & 0.7478 & 1.2932 \end{pmatrix}$$

corresponds to Example 5.3 from [4] after homogenization. This is an example coming from portfolio optimization. The results reported in [4] are 0.3015 for $\mathcal{C}^1$ and 0.4839 for $\mathcal{K}^1$, which is optimal. Our algorithm takes 27 iterations (0.01 sec) to obtain an accuracy of 10^{-6}.

Next, we consider an example taken from Peña et al. [18]. For a graph with 17 vertices, they propose bounds on the clique number obtained by solving SDPs over the cones $\mathcal{Q}^r$ (cf. section 1.2). They state that using $\mathcal{Q}^4$ is beyond current computational capabilities. This is indeed a hard instance: our algorithm solves this problem to optimality in 14,411 iterations (20 hours, 18 min, and 5 sec). In this instance we again used the fact that the reciprocal is integer to round the bounds appropriately.

We also tested some of the max-clique instances from the Second DIMACS Challenge ([8]). The smallest instance, Johnson 8-2-4, a graph with 28 vertices, was solved to optimality in 946 iterations (1 min and 33 sec). We also solved the Hamming 6-4 instance, a graph with 64 vertices. This instance took 2,385 iterations (57 min and 52 sec). For all other instances from this library, our algorithm produced only poor bounds within two hours of computation time (the best lower bound was usually 3, and the upper bound stayed at $+\infty$).

We also tried to solve other combinatorial problems like the quadratic assignment problem using a formulation of Povh and Rendl [20]. However, for most of these instances, our algorithm gave only trivial or weak bounds.

Finally, we generated random instances of the standard quadratic optimization problem (6.1), where the entries of the symmetric matrix $Q \in \mathbb{R}^{n\times n}$ were uniformly distributed in $[-n, n]$. For each size, 100 instances where generated. The algorithm was stopped when the relative gap between upper and lower bound was smaller than 10^{-6}. The results are listed in Tables 6.1 and 6.2.

The first column in the tables denotes the problem dimension, i.e., the number of variables. The 2nd and 3rd columns describe the average and maximal number of iterations. Finally, the last four columns give information about the cpu-time.

Table 6.1

Numerical results for randomly generated instances of the standard quadratic optimization problem, obtained on a Pentium IV, 2.8*GHz Linux machine with* 1*GB RAM. All problems were solved up to a relative tolerance of* 10^{-6}.

	Iterations		cpu-time (sec.)			
n	avg	max	init	avg	min	max
10	4.25	38	0.0001	0.0034	0	0.04
30	3.26	26	0.0019	0.0056	0	0.05
50	3.78	40	0.0046	0.0124	0	0.11
100	3.32	34	0.0269	0.0557	0.01	0.68
200	2.97	35	0.1154	0.2202	0.04	3.02
500	3.17	27	0.5451	1.5483	0.38	14.26
750	2.92	23	1.9535	3.4373	0.88	30.24
1,000	3.14	29	2.5706	5.9362	1.48	59.89
1,500	4.33	75	5.9710	19.4610	3.51	366.35
2,000	2.85	24	11.4993	23.9875	6.26	225.21

Table 6.2

Numerical results for randomly generated instances of the standard quadratic optimization problem, obtained on a 16 *Dual-Core AMD Opteron*$^{\text{TM}}$ 8220 *machine with* 2.8*GHz frequency and* 130*GB RAM. Only one core was used in our computations. All problems were solved up to a relative tolerance of* 10^{-6}.

	Iterations		cpu-time (sec.)			
n	avg	max	init	avg	min	max
2,500	3.13	53	8.9037	30.3367	7.22	571.38
3,000	2.56	22	14.3911	34.5022	10.23	338.79
4,000	2.85	25	26.6361	70.8114	18.48	698.08
5,000	2.45	18	44.2364	101.155	31.18	872.96
7,000	2.45	23	91.2996	203.620	59.89	2,187.65
10,000	2.97	27	192.3010	477.258	116.08	5,184.74

The cpu-time was measured in two parts: The first part is the initialization time, which is the time needed to set up the starting LP and feed it to the solver. The initialization time is the same for all instances of the same size and is listed in the column init. The second part is the actual solution time, which is the elapsed time from solving the starting LP to termination of the algorithm. This time differs not only with the size but also with the data of the instance. Therefore, the average, minimum, and maximum solution times are stated in the respective columns.

As can be seen from Table 6.1, the solution times for these problems are not bad. However, our algorithm requires a lot of memory, and for this reason higher dimensional problems took more time on this computer due to memory swapping. Therefore, we did some further experiments on a computer with larger memory: We used a 16 Dual-Core AMD Opteron$^{\text{TM}}$ 8220 machine with 2.8GHz frequency and 130GB RAM. Our algorithm used only one of the CPUs. On this machine, we were able to solve even higher dimensional problems in very reasonable time, as can be seen in Table 6.2.

Observe that in all instances the number of iterations of our algorithm is very low and comparable to interior point methods.

To provide some intuition on how difficult the problems in Tables 6.1 and 6.2 are to solve to global optimality, we tried solving these problems with BARON [22] which is available via the NEOS server. Obviously, it was impossible to solve 100 instances for each dimension through the NEOS server. Therefore, we were only able to try a few random instances, which admittedly only give a rough picture. Nonetheless, we believe that running 100 instances per dimension would not give an entirely different pattern. Observe that on the NEOS server, each job is allotted a run time of 1,000 seconds only. NEOS currently uses version BARON 8.1.4.

We observed that for instances of size 10 the solution times of BARON were similar to ours. With instances of size 30 and bigger, BARON did not succeed to solve the problems to optimality within the given 1,000 seconds. For instances of size 250 we observed that BARON ran into memory problems and accordingly returned an error message. So it seems that BARON cannot compete with our method for this specific type of problems in large dimensions.

Bomze and de Klerk [4] also state some numerical results for randomly generated instances of the standard quadratic optimization problem. They did calculations for the linear and semidefinite approximations resulting from the cones $\mathcal{C}^1$ and $\mathcal{K}^1$, respectively. Compared with these results, our algorithm is much faster even in consideration of the faster hardware, and is able to solve much bigger instances. Moreover, we get a guaranteed solution accuracy of at least 10^{-6} for all instances.

7. Conclusions. We introduced new polyhedral inner and outer approximations of the copositive cone and presented a solution algorithm for copositive programs which uses this approximation scheme. The advantage of our algorithm is that it does not approximate the copositive cone uniformly, but can be guided by the objective function. Numerical experiments show that the algorithm works very well for quadratic programs over the simplex.

Open points of interest are:

- Can we use our method to solve other types of quadratic optimization problems? We tried to solve some box-constrained problems, but were unable to solve even medium size instances.
- Can we tailor our method towards combinatorial problems like the quadratic assignment problem, or can we find better copositive formulations of those problems?

Our approach can easily be extended to optimization problems involving more general notions of copositivity in the sense of [11]. Here one is concerned with matrices which are copositive with respect to some general cone $\mathcal{D}$, i.e., matrices that induce a quadratic form nonnegative not over $\mathbb{R}^n_+$ but over $\mathcal{D}$. If $\mathcal{D}$ is polyhedral and pointed, then it is easy to find a base $\mathcal{B}$ such that $\mathbb{R}_+\mathcal{B} = \mathcal{D}$. Instead of working with simplicial partition of Δ^S, one then has to work with partitions of $\mathcal{B}$. The computational effort of course increases if the structure of $\mathcal{B}$ is more complex. Also, we are not aware of applications that necessitate optimization over $\mathcal{D}$-copositive matrices, so we believe the canonical setting is the most interesting, but see [7] for an application of the problem of deciding $\mathcal{D}$-copositivity of a matrix.

Acknowledgments. We are grateful to the referees for taking the time to provide detailed and highly valuable comments on this paper.

REFERENCES

[1] A. Berman and N. Shaked-Monderer, *Completely Positive Matrices*, World Scientific, River Edge, NJ, 2003.

[2] A. Berman and U. Rothblum, *A Note on the Computation of the CP-Rank*, Linear Algebra Appl., 419 (2006), pp. 1–7.

[3] I.M. Bomze, M. Dür, E. de Klerk, C. Roos, A.J. Quist, and T. Terlaky, *On copositive programming and standard quadratic optimization problems*, J. Global Optim., 18 (2000), pp. 301–320.

[4] I.M. Bomze and E. de Klerk, *Solving standard quadratic optimization problems via linear, semidefinite and copositive programming*, J. Global Optim., 24 (2002), pp. 163–185.

[5] S. Bundfuss and M. Dür, *Algorithmic copositivity detection by simplicial partition*, Linear Algebra Appl., 428 (2008), pp. 1511–1523.

[6] S. Burer, *On the copositive representation of binary and continuous nonconvex quadratic programs*, Math. Program., in print. Available at http://dx.doi.org/10.1007/s10107-008-0223-z

[7] G. Danninger, *A recursive algorithm for determining (strict) copositivity of a symmteric matrix*, Methods Oper. Res., 62 (1990), pp. 45–52.

[8] DIMACS, *Second DIMACS Challenge* (1992–1993). Test instances available at: http://dimacs.rutgers.edu/Challenges/.

[9] I. Dukanovic and F. Rendl, *Copositive programming motivated bounds on the stability and the chromatic numbers*, Math. Program., in print. Available at http://dx.doi.org/10.1007/s10107-008-0233-x

[10] M. Dür and G. Still, *Interior points of the completely positive cone*, Electronic J. Linear Algebra, 17 (2008), pp. 48–53.

[11] G. Eichfelder and J. Jahn, *Set-semidefinite optimization*, J. Convex Analysis, 15 (2008), pp. 767–801.

[12] R. Horst, *On generalized bisection of n-simplices*, Math. Comput., 218 (1997), pp. 691–698.

[13] F. Jarre, I.M. Bomze, and F. Rendl, *Quadratic Factorization Heuristics for Copositive Programming*, working paper presented at the Czech-French-German Conference on Optimization, September 2007, University of Heidelberg.

[14] F. Jarre and K. Schmallowsky, *On the computation of C^* certificates*, Report, Mathematisches Institut, Universität Düsseldorf, 2008. Available at: http://www.optimization-online.org/DB_HTML/2008/05/1969.html.

[15] E. de Klerk and D.V. Pasechnik, *Approximation of the stability number of a graph via copositive programming*, SIAM J. Optim., 12 (2002), pp. 875–892.

[16] K. G. Murty and S. N. Kabadi, *Some NP-complete problems in quadratic and nonlinear programming*, Math. Program., 39 (1987), pp. 117–129.

[17] P. Parrilo, *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, Ph.D. Dissertation, California Institute of Technology, 2000. Available at: http://etd.caltech.edu/etd/available/etd-05062004-055516/.

[18] J. Peña, J. Vera, and L. Zuluaga, *Computing the stability number of a graph via linear and semidefinite programming*, SIAM J. Optim., 18 (2007), pp. 87–105.

[19] J. Povh and F. Rendl, *A copositive programming approach to graph partitioning*, SIAM J. Optim., 18 (2007), pp. 223–241.

[20] J. Povh and F. Rendl, *Copositive and semidefinite relaxations of the quadratic assignment problem*, to appear. Preprint available at: http://www.optimization-online.org/DB_HTML/2006/10/1502.html.

[21] A.J. Quist, E. de Klerk, C. Roos, and T. Terlaky, *Copositive relaxation for general quadratic programming*, Optim. Methods Software, 9 (1998), pp. 185–208.

[22] M. Tawarmalani and N. V. Sahinidis, *Global optimization of mixed-integer nonlinear programs: A theoretical and computational study*, Math. Program., 99 (2004), pp. 563–591.