Central European Journal of Operations Research
https://doi.org/10.1007/s10100-021-00737-6

On monotonicity and search strategies in face-based copositivity detection algorithms

B. G.-Tóth[1] · E. M. T. Hendrix[2,3] · L. G. Casado[4]

Accepted: 22 January 2021

Abstract

Over the last decades, algorithms have been developed for checking copositivity of a matrix. Methods are based on several principles, such as spatial branch and bound, transformation to Mixed Integer Programming, implicit enumeration of KKT points or face-based search. Our research question focuses on exploiting the mathematical properties of the relative interior minima of the standard quadratic program (StQP) and monotonicity. We derive several theoretical properties related to convexity and monotonicity of the standard quadratic function over faces of the standard simplex. We illustrate with numerical instances up to 28 dimensions the use of monotonicity in face-based algorithms. The question is what traversal through the face graph of the standard simplex is more appropriate for which matrix instance; top down or bottom up approaches. This depends on the level of the face graph where the minimum of StQP can be found, which is related to the density of the so-called convexity graph.

Keywords Copositive matrix · Standard simplex · Monotone · Face

This work has been funded by grants from the Spanish Ministry (RTI2018-095993-B-I00), in part financed by the European Regional Development Fund (ERDF). We are grateful to Fabio Tardella for the discussion of his findings on the StQP and to the anonymous referees that brought up a discussion on the convexity graph.

✉ B. G.-Tóth
boglarka@inf.szte.hu

E. M. T. Hendrix
eligius@uma.es

L. G. Casado
leo@ual.es

1 University of Szeged, Szeged, Hungary

2 Universidad de Málaga, Málaga, Spain

3 Wageningen University, Wageningen, The Netherlands

4 Universidad de Almería, (ceiA3), Almería, Spain

Published online: 06 March 2021

1 Introduction

Copositivity of a matrix is an important concept in combinatorial and quadratic optimization (Burer 2009; Kaplan 2000; Povh and Rendl 2007; Väliaho 1986). Given the standard simplex with the n coordinate unit vectors e_i, $i = 1, \dots, n$ as vertices:

$$\Delta_n := \left\{ x \in \mathbb{R}^n_+ \;\middle|\; \sum_{i=1}^{n} x_i = 1 \right\}. \tag{1}$$

A symmetric $n \times n$ matrix A is called copositive if

$$\forall x \in \Delta_n,\ x^T A x \geq 0 \tag{2}$$

and noncopositive if

$$\exists x \in \Delta_n,\ x^T A x < 0. \tag{3}$$

Copositivity is a weaker condition than positive semidefiniteness (PSD), i.e. PSD implies copositivity

$$\forall x \in \mathbb{R}^n,\ x^T A x \geq 0. \tag{4}$$

One can determine if a matrix is PSD via Cholesky decomposition in polynomial time ($O(n^3)$). However, determination of copositivity of a matrix has been shown to be a co-NP complete problem (Murty and Kabadi 1987). The certification of copositivity is related to the standard quadratic program (StQP)

$$f^* := \left(\min_{x \in \Delta_n} f(x) := x^T A x \right). \tag{5}$$

The StQP is generic in the sense that optimizing a quadratic function $f(x) := x^T Qx + b^T x + c$ over Δ_n is equivalent to (5), taking $A := Q + \frac{1}{2}(\mathbf{1}b^T + b\mathbf{1}^T) + c\mathbf{1}\mathbf{1}^T$, where $\mathbf{1}$ is the all-ones vector. Clearly, if $f^* < 0$, then A is not copositive and if $f^* \geq 0$, then A is copositive.

There have been suggestions in literature to create procedures for copositivity testing based on properties of the matrix (Nie et al. 2018; Yang and Li 2009). The spatial branch and bound (B&B) algorithm introduced by Bundfuss and Dür (2008) can either certify that a matrix is not copositive, or prove it is so-called ϵ-copositive. Following such a procedure, Žilinskas and Dür (2011) claim that certifying ϵ-copositivity of a copositive matrix is limited to a size up to $n = 22$ in a reasonable time. Certification of ϵ-copositivity by simplicial refinement requires much more computation than verifying non-copositivity of a matrix, which can be done for a dimension n up to several thousands.

Basically, a spatial B&B approach samples and evaluates points that do not necessarily coincide with candidates of optima of (5). In contrast, a recent work which also compares with B&B approaches by Liuzzi et al. (2019), focusing on the first order conditions, i.e. Karush–Kuhn–Tucker (KKT) conditions of the optima of (5). They apply convex and linear bounds in a B&B context implicitly enumerating KKT points, which proved to be very efficient in low-density graphs. The consequence of introducing monotonicity in the original B&B in Bundfuss and Dür (2008) has been investigated in Hendrix et al. (2019) and Salmerón (2019).

Also recent is the work of Gondzio and Yildirim (2018) which translates the KKT conditions to a MIP type of approach making use of fast integer programming implementations in order to solve the StQP (5) for instances of hundreds of variables. The mentioned procedures do not focus on the second order considerations to find an optimum.

An older work, Scozzari and Tardella (2008) derives algorithms based on the observation that a minimum point of (5) can only be on the so-called relative interior of a face of the standard simplex, if f is convex on that face. Their focus is on the convexity of f on the edges of that face. In this way, they look for what they call a clique in the convexity graph, consisting of the vertices of the standard simplex and edges where f is strictly convex.

Our focus is also face-based, where we look for negative points of f for noncopositivity detection, where only local minimum points on the relative interior of faces are evaluated. The search procedure traverses what we call the face graph of the standard simplex, as depicted in Fig. 1. Each node is a face of the standard simplex, described by a bit string b_k; if the ith position is 1, then vertex e_i is included in the face. We introduce the ordered index set $\mathcal{I}_k$ as the ordered set of indices of the variables in face F_k. Nodes on level ℓ are ℓ-faces, so the standard simplex is the root, and vertices e_i are on the lowest level. Level ℓ has $\binom{n}{\ell}$ faces. Node set $\mathcal{F} := \{1, \ldots, 2^n - 1\}$ contains all face numbers. An edge (m, k) implies that either F_m is a facet of F_k or vice versa. This means that bit string b_k differs in one bit from b_m and consequently $\mathcal{I}_k$ has one index more than $\mathcal{I}_m$ or the other way around.

In this context, the procedure of Scozzari and Tardella (2008) first marks nodes on level $\ell = 2$ (edges of the standard simplex) as strictly convex and goes up in the graph to identify faces on a level as high as possible with all edges convex to find interior minima, with a value as low as possible. For faces F_m on a lower level, we have that $F_m \subset F_k$ implies $\min_{x \in F_m} f(x) \geq \min_{y \in F_k} f(y)$. Now, having f convex on all edges is a necessary but not sufficient condition for f to be convex on the face. Therefore, Scozzari and Tardella (2008) test whether the matrix A_k related to the elements of face F_k is positive definite (PD). We will show, that this is a sufficient, but not necessary condition for f to be convex on F_k.

Our main research question is how we can traverse the face graph identifying those faces where f is strictly convex on the relative interior and evaluate the corresponding minima in order to find points with a negative objective function value or to prove that A is copositive. To report on the findings of our investigation of the research questions, our paper is organized as follows. Section 2 discusses the mathematical properties relevant for the algorithm development about monotonicity and first and second order conditions. Section 3 sketches the traversal variants of a face-based algorithm. Section 4 uses several benchmark instances to investigate numerically for what type of matrices which graph traversal is more effective. The main conclusions and future research questions are described in Sect. 5.

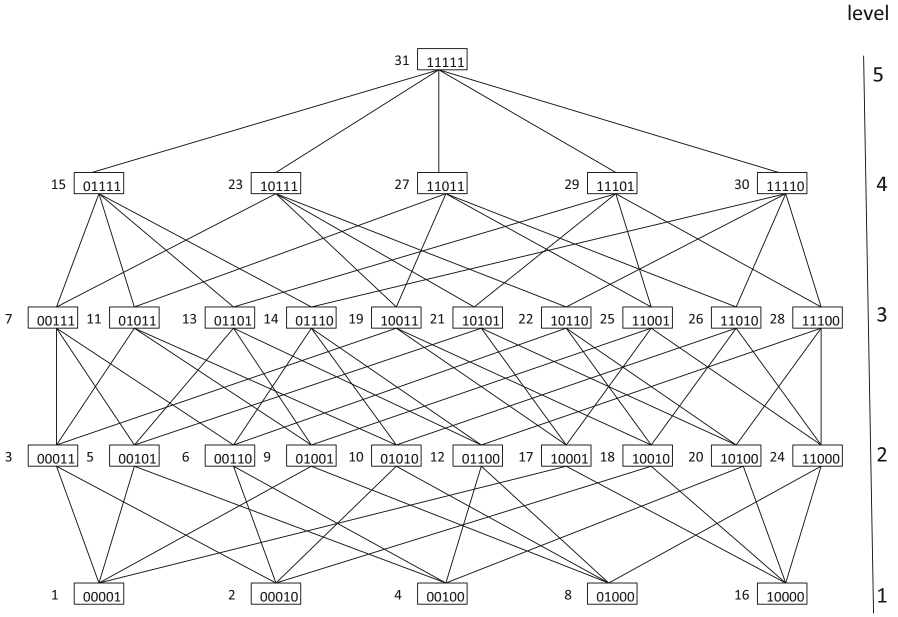

Fig. 1 Face graph of the standard simplex for $n = 5$, where faces F_k are numbered by index k and bit string b_k indicates which elements are considered positive and which zero

2 Properties of a StQP

In our notation, we use as identity matrix in dimension n the symbol $I_n := (e_1, \ldots, e_n)$ and $\mathbf{1}$ represents the all-ones vector in appropriate dimension. Moreover, we use $D_n := I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T = (d_1, \ldots, d_n)$ as the projection matrix on the zero sum plane $\mathcal{P} := \{x \in \mathbb{R}^n | \mathbf{1}^T x = 0\}$. It is useful to consider matrix A_k in order to evaluate $f := x^T Ax$ on face F_k.

Definition 1 Given a symmetric $n \times n$ matrix A and a binary vector b_k with corresponding index set $\mathcal{I}_k$, A_k is the sub-matrix of A with rows and columns that correspond to indices in $\mathcal{I}_k$. For face F_k at level ℓ, A_k is an $\ell \times \ell$ matrix.

Note that $\min_{x \in F_k} x^T Ax$ is equivalent to $\min_{x \in \Delta_\ell} x^T A_k x$.

2.1 Optimality conditions

The first order conditions (KKT conditions) for a local minimum of StQP (5) are used in the studies of Gondzio and Yildirim (2018), Liuzzi et al. (2019), Salmerón et al. (2018) and Scozzari and Tardella (2008). For a local minimum point $x \geq 0$ of (5) there exist values of the dual variables μ and $\lambda_i \geq 0$ such that

$$Ax = \mu \mathbf{1} + I_n \lambda, \; x^T \lambda = 0. \tag{6}$$

The expression $x_i \lambda_i = 0$ is called complementarity and is closely related to the question on which face the minimum point can be found. Gondzio and Yildirim (2018) shows that the StQP (5) can be solved for hundreds of variables using Mixed Integer Programming (MIP) applying binary variables to capture the complementarity. Liuzzi et al. (2019) derives a B&B algorithm which implicitly enumerates the KKT points obtaining new linear and convex expressions for the bounds.

It is known that the minimum of an indefinite quadratic function can be found at the boundary of the feasible set. Basically, the feasible set Δ_n of (5) does not have an interior. Consider the relative interior $\text{rint}(\Delta_n)$ of Δ_n as

$$\text{rint}(\Delta_n) := \left\{ x \in \mathbb{R}^n \;\middle|\; \sum_{j=1}^{n} x_j = 1;\; x_j > 0, \quad j = 1, \ldots, n \right\}. \tag{7}$$

When we know that a face F_k has a relative interior minimum point y^*, it is given by

$$A_k y^* - \mu \mathbf{1} = 0, \;\; \mathbf{1}^T y^* = 1, \tag{8}$$

where y^* is mostly in a space of dimension lower than n. Translation of the solution to n-dimensional space requires adding zeros on the positions $i \notin \mathcal{I}_k$. So, either the global minimum point of StQP can be found in one of the vertices of the standard simplex (unit vector e_i) or at the relative interior of one of the other faces. To have a relative interior optimum on F_k, f should at least be convex on F_k. Scozzari and Tardella (2008) characterize this by looking for faces where A_k is positive definite (PD). However, this is not a necessary condition. For instance, matrix

$$A := \begin{pmatrix} 0 & -1 & -2 \\ -1 & 1 & -3 \\ -2 & -3 & 2 \end{pmatrix} \tag{9}$$

defines a function f which is strictly convex on Δ_3, but is not PD. Consider the matrix $H := D_n A D_n$. This matrix defines the convexity on the standard simplex.

Proposition 1 *If the matrix $H := D_n A D_n$ is positive semidefinite (PSD), then the function $f := x^T A x$ is convex on Δ_n.*

Proof Let $x, y \in \Delta_n$. Notice that $(y - x) \in \mathcal{P}$ such that $\exists r \in \mathbb{R}^n$, $y - x = D_n r$. Then for $0 \leq \lambda \leq 1$

$$\begin{aligned}
&(1-\lambda) f(x) + \lambda f(y) - f((1-\lambda)x + \lambda y) \\
&\quad = (1-\lambda) f(x) + \lambda f(y) - (1-\lambda)^2 f(x) - \lambda^2 f(y) - 2(1-\lambda)\lambda x^T A y \\
&\quad = \lambda(1-\lambda)[f(x) + f(y) - 2x^T A y] \\
&\quad = \lambda(1-\lambda)[x^T A x + y^T A y - 2x^T A y] \\
&\quad = \lambda(1-\lambda)(y-x)^T A (y-x) = \lambda(1-\lambda) r^T H r \geq 0
\end{aligned}$$

This means that for any $x, y \in \Delta_n$ and $0 \le \lambda \le 1$, $(1-\lambda)f(x) + \lambda f(y) \ge f((1-\lambda)x + \lambda y)$. □

The other way around, to have a relative interior optimum, H should be PSD.

Proposition 2 *If* $\exists\ x^* \in \operatorname{argmin}_{\Delta_n} f(x) \cap \operatorname{rint}(\Delta_n)$*, then* $D_n A x^* = 0$ *and* $H := D_n A D_n$ *is positive semidefinite.*

Proof Following the KKT conditions (6), only the constraint $\mathbf{1}^T x = 1$ is binding, i.e. there exists a value μ such that $Ax^* = \mu \mathbf{1}$. As $D_n = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$,

$$D_n A x^* = D_n \mu \mathbf{1} = \mu[I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T]\mathbf{1} = \mu[\mathbf{1} - \frac{n}{n}\mathbf{1}] = 0. \quad (10)$$

Considering $f(x)$ in a δ−ball $B(x^*, \delta)$ around x in $\operatorname{rint}(\Delta_n)$ can be described as considering $x = (x^* + h) \in \operatorname{rint}(\Delta_n)$ with $h = D_n r \in \mathcal{P}$. As x^* is a relative interior minimum point, there exists $\delta > 0$, such that $\forall r \in B(0, \delta)$, $f(x^* + D_n r) \ge f(x^*)$. So $\forall r \in B(0, \delta)$

$$f(x^* + D_n r) = f(x^*) + 2(D_n A x^*)^T r + r^T H r \ge f(x^*). \quad (11)$$

As $D_n A x^* = 0$, we have that $\forall r \in B(0, \delta)$, $r^T H r \ge 0$, so H is a positive semidefinite matrix. □

With respect to strict convexity, one of the eigenvalues of H with respect to direction $\mathbf{1}$ is zero, as $D_n \mathbf{1} = 0$. Basically, this means that the other eigenvalues of H should be positive. Our implementations and Scozzari and Tardella (2008) use Cholesky decomposition routines to test whether A_k or H_k are PD or PSD.

Proposition 2 can be extended to any face F_k if we consider the standard simplex Δ_ℓ in dimension ℓ equal to the number of positive elements in face F_k.

Corollary 1 *If* $\exists\ x^* \in \operatorname{argmin}_{F_k} f(x) \cap \operatorname{rint}(F_k)$*, then also* $\exists\ y^* \in \operatorname{rint}(\Delta_\ell)$*, such that* $D_\ell A_k y^* = 0$ *and* $H_k := D_\ell A_k D_\ell$ *is positive semidefinite.*

2.2 The convexity graph

The analysis of Scozzari and Tardella (2008) focuses on convexity of the edges of a face. Function f over edge (e_i, e_j) can be written as

$$\varphi(\rho) := f(e_i + \rho(e_j - e_i)) = A_{ii} + 2\rho(A_{ij} - A_{ii}) + \rho^2(A_{ii} + A_{jj} - 2A_{ij}). \quad (12)$$

The function f is strictly convex over edge (e_i, e_j) of Δ_n if

$$A_{ii} + A_{jj} - 2A_{ij} > 0. \quad (13)$$

In that case, a relative minimum point of f over the line $e_i + \rho(e_j - e_i)$ is given by

$$\rho^* := \frac{A_{ii} - A_{ij}}{A_{ii} + A_{jj} - 2A_{ij}}. \quad (14)$$

If $0 < \rho^* < 1$, then we have an interior minimum over edge (e_i, e_j) with function value

$$f_{ij}^* := A_{ii} - \frac{(A_{ii} - A_{ij})^2}{A_{ii} + A_{jj} - 2A_{ij}}. \tag{15}$$

Definition 2 The convexity graph $G := (N, C)$ of a matrix A is a graph with node set $N := \{1, \dots, n\}$ having an edge $(i, j) \in C$ if (13) holds, i.e. f is strictly convex over the edge between e_i and e_j (Scozzari and Tardella 2008).

Actually, a node can be removed from G if it is not incident to any edge on which f is convex. Graph G_k is defined similarly to matrix A_k. A necessary but not sufficient condition for f to be convex on F_k is that graph G_k is complete. Due to the focus of Scozzari and Tardella (2008), we illustrate that the condition is not sufficient for matrix

$$A_k := \begin{pmatrix} 8 & 5 & 3 \\ 5 & 3 & 3 \\ 3 & 3 & 4 \end{pmatrix}. \tag{16}$$

The corresponding convexity graph G_k is complete, but f is not convex on F_k, neither A_k nor H_k are PSD.

Looking for a face F_k with an interior optimum of the StQP means looking for faces that correspond to a complete G_k on a level as high as possible. This consideration brings up some typical properties. We did not focus on sparse matrices in our study. A counter-intuitive property is that $A_{ij} = 0$ corresponds to a convex edge. Hence, when A is sparse, the corresponding convexity graph will be quite dense.

The necessary condition of convex edges for an interior optimum, also implies that an edge, where f is strictly concave, cannot belong to a face with a relative interior optimum.

Proposition 3 *Let $x^* \in \operatorname{argmin}_{\Delta_n} x^T Ax$. If $A_{ii} + A_{jj} - 2A_{ij} < 0$, i.e. $f(x) := x^T Ax$ is strictly concave over edge (e_i, e_j), then $x_i^* x_j^* = 0$.*

Proof A minimum point x^* is either a vertex or a relative interior point of a face F_k. The edge (e_i, e_j) apparently cannot be subset of F_k, so either $x_i^* = 0$ or $x_j^* = 0$. □

An anonymous referee drew our attention to the following consequence, which may be used in algorithm development. Consider matrix $\tilde{A}$ where in A, we replace the corresponding concave edge entrance A_{ij} by $\tilde{A}_{ij} := \frac{1}{2}(A_{ii} + A_{jj})$ if $A_{ii} + A_{jj} - 2A_{ij} < 0$. The corresponding function $\tilde{f} := x^T \tilde{A} x$ enhances a linearization of f over the concave edges. Following the same reasoning that a relative interior optimum on face F_k requires f to be strictly convex on its edges, we have the following property.

Corollary 2 *Let $\tilde{f}(x) := x^T \tilde{A} x$, where $\tilde{A}$ is defined as A replacing $\tilde{A}_{ij} = \frac{1}{2}(A_{ii} + A_{jj})$ if $A_{ii} + A_{jj} - 2A_{ij} < 0$. The StQP (5) is equivalent for A and $\tilde{A}$, i.e. $x^* \in \operatorname{argmin}_{\Delta_n} x^T Ax$ implies $x^* \in \operatorname{argmin}_{\Delta_n} x^T \tilde{A} x$ and $f^* = \tilde{f}^*$.*

Due to the equivalence, this property does not seem relevant when checking faces on convexity going level up in the face graph. However, this might be very relevant for search procedures going downward in the face graph that use monotonicity considerations; this is outlined in the following section.

2.3 Monotonicity considerations

In simplicial B&B methods like that of Bundfuss and Dür (2008) that use simplicial partition sets $S := \text{conv}(\mathcal{V})$ based on vertex set $\mathcal{V} := \{v_1, \ldots, v_n\}$, Hendrix et al. (2019) elaborated theoretically how monotonicity can be used. We extend the results here and investigate what this enhances for face-based algorithms specifically. For a general simplicial set $\text{conv}(\mathcal{V})$, we can consider the facet $\Phi_i := \text{conv}(\mathcal{V} \setminus \{v_i\})$.

Proposition 4 *Consider simplex* $S := \text{conv}(\mathcal{V})$ *and facet* Φ_i*. If* $\exists y \in \Phi_i$ *such that* $(v_i - y)^T AV \geq 0$*, then* $\min_S f(x)$ *is attained at* Φ_i*.*

Proof Consider the matrix V with columns that correspond to the vertices of S. Then point $y \in \Phi_i$ can be written as $y = V\lambda$ for a vector $\lambda \in \Delta$ with $\lambda_i = 0$. Assume the minimum is attained at $x^* := V\lambda^*$ with $\lambda_i^* > 0$. Now consider the point $x := x^* - \lambda_i^*(v_i - y) = V\gamma$ with $\gamma = \lambda^* - \lambda_i^*(e_i - \lambda)$. Then we have that $\gamma \in \Delta$ with $\gamma_i = 0$, so $x \in \Phi_i$. Moreover,

$$f(x) = x^{*T} Ax^* + (x - x^*)^T A(x + x^*) = f(x^*) - \lambda_i^*(v_i - y)^T AV(\gamma + \lambda^*).$$

As $(v_i - y)^T AV \geq 0$ and $\gamma + \lambda^* \geq 0$, we have that $f(x) \leq f(x^*)$, so the minimum is attained at $x \in \Phi_i$. □

The importance of this theoretical result is that, for the StQP (5) and the proof on copositivity, one can reduce the search in a B&B algorithm like Bundfuss and Dür (2008) to a facet of simplicial subset S.

Corollary 3 *Consider simplex* $S := \text{conv}(\mathcal{V})$ *and facet* Φ_i*. If* $\exists p \neq i \forall j (v_i - v_p)^T A v_j \geq 0$*, the minimum of* f *is attained at facet* Φ_i*.*

Proof Follows directly from Proposition 4, by taking y as vertex $v_p \in \Phi_i$. □

For a face-based algorithm, the analysis is easier, as it says we can drop out vertex e_i from the face F_k we are investigating. The condition of Corollary 3 implies

$$\exists p \neq i, \quad a_{ki} - a_{kp} \geq 0, \tag{17}$$

where a_{ki} and a_{kp} are columns i and p of A_k, respectively.

Actually, monotonicity considerations in face-based algorithms are only relevant when searching the face graph in Fig. 1 top-down, as it specifies which facets on the lower level can contain a minimum point, eliminating in a B&B way the search on other faces.

3 Algorithms

The authors of Salmerón (2019) report the findings of the consequence of including monotonicity in the B&B algorithm of Bundfuss and Dür (2008). In this paper, we develop three traversal variants of the face graph of Δ_n. The algorithms only evaluate the vertices, i.e. the diagonal elements of A, the centroid and proven interior minimum points of faces F_k.

1. TDk (Alg. 1), **T**raverses the faces in **D**ecreasing order of ***k*** and checks whether they should be investigated. This requires at least for each face to store a marker indicating if a face and sometimes all its sub-faces need not be checked.
2. TDown (Alg. 2), tries to avoid storage by checking the faces in a list for each level and **T**raverses the face graph **Down**wards level-wise.
3. TUp (Alg. 3), follows the line of the algorithm of Scozzari and Tardella (2008) **T**raversing **Up**wards the face graph. However, in contrast it works also level-wise, as negative points may be found in local minima on lower levels.

In each algorithm, we first check matrix A to be copositive due to

- $\forall i, j, \ A_{ij} \geq 0$, i.e. A is entry-wise nonnegative
- A is PSD (Cholesky decomposition).

Then the following actions are taken:

- Check diagonal elements $A_{ii} \geq 0$, i.e f is nonnegative in vertices of Δ_n ($\ell = 1$).
- Evaluate the centroid; $f(\frac{1}{n}\mathbf{1}) \geq 0$.
- Create the convexity graph G checking strict convexity over edges via (13), meanwhile calculating their minima given by (15) (this corresponds to $\ell = 2$).

3.1 TDk

The TDk version in face-based Algorithm 1 goes over the faces of the graph from higher index value k downward until the list has been checked or a negative point has been found. It has the similarity with B&B approaches, that on a higher level, we hope to exclude evaluation on lower levels in the graph. Therefore, it uses a global indicator list Tag_k

- $\text{Tag}_k = 0$: face F_k has not been checked,
- $\text{Tag}_k = 1$: face F_k need not be checked and
- $\text{Tag}_k = 2$: no need to check face F_k nor any of its sub-faces $F_m \subset F_k$.

When f is shown to be monotonous, we can tag some of the facets as checked. As all sub-matrices A_m of a copositive matrix A_k are also copositive, we do not have to evaluate the sub-faces F_m in the face graph. This means that like in B&B the efficiency of the algorithm depends on which level ℓ of the graph we detect a face F_k where A_k is copositive. The higher in the graph, the more nodes do not have to be evaluated.

Computationally, Algorithm 1 has the advantage that we only have to store one byte Tag_k for each face. However, from a complexity point of view, the number of faces increases exponentially in the dimension n. Moreover, one face has usually several parents in the face graph, such that it may be visited or marked several times during the algorithm.

3.2 TDown

The idea of Algorithm 2 is to inspect only the interesting faces of each level of the face graph. Only faces are evaluated at a level that may still contain negative points of

Algorithm 1 Copositivity detection, traversing in decreasing k order (TDk)

Require: A: symmetric $n \times n$ matrix, $n \geq 3$.
if A is entry-wise nonnegative or A is PSD **then**
  return A is copositive
Evaluate A_{ii}, centroid and edge minima where f is convex via (15), generating G
if negative point found **then**
  return A is not copositive
$\text{Tag}_k := 0$ for faces k not checked
$k := 2^n - 1$
while $k > 2$ **do**
  if $\text{Tag}_k = 0$ **then**
    if $A_k \geq 0$ **then**
      Set Tag_m:=2 for all sub-facets $F_m \subset F_k$
      break
    if $\exists i$ for which (17) holds, the minimum is on the facet without $e_{\mathcal{I}_k(i)}$ **then**
      Set Tag_m:=1 for facets $F_m \subset F_k$ which include $e_{\mathcal{I}_k(i)}$
    else
      if G_k is complete **then**
        if H_k is positive semidefinite **then**
          Determine $x_k^* \in F_k$ solving (8)
          if $x_k^{*T} A_k x_k^* < 0$ **then**
            return A is not copositive
          else
            Tag_m:=2 for $F_m \subset F_k$
  $k := k - 1$
return A is copositive

f. Two lists of candidate faces are active $\mathcal{L}_\ell$ and $\mathcal{L}_{\ell-1}$, where only the numbers of the faces are stored. In the pseudo-code, removal or insertion of a face F_k in a list means the removal or insertion of number k.

Moreover, it maintains one global list $\mathcal{R}$, which stores the values k of faces with a nonnegative interior minimum or with a completely positive matrix A_k. On each level, it keeps a list $\mathcal{N}$ of facets of the faces that cannot have a minimum due to monotonicity. Let us note that sub-faces of the faces in $\mathcal{N}$ are still considered, as the optimum of f on a monotonous face is on the boundary. This is different for faces in $\mathcal{R}$; none of the sub-faces of $F_k, k \in \mathcal{R}$ can contain negative minima, so all its sub-faces can be dropped. In this way, no sub-face of $F_k, k \in \mathcal{R}$ nor facet $F_k, k \in \mathcal{N}$ has to be included in the next level list $\mathcal{L}_{\ell-1}$. In the worst case, list $\mathcal{L}_\ell$ may still be huge with increasing dimension n. Most troublesome, however, is the list management overhead.

3.3 TUp

Algorithm 3, TUp, follows the upward search of the algorithm of Scozzari and Tardella (2008) in their search for the minimum of StQP(5) in the levels of the face graph. Their terminology is to look for maximum cliques in the convexity graph, i.e. G_k is complete on a level ℓ as high as possible. Finding such a face F_k still requires checking whether H_k is PSD in order to have a possible interior optimum on face F_k. The algorithm stops if it finds a negative interior optimum or alternatively has found the highest

Algorithm 2 Copositivity detection top-down in the face graph (TDown)

Require: A: symmetric $n \times n$ matrix.
if A is entry-wise nonnegative or A is PSD **then**
  return A is copositive
Evaluate A_{ii}, centroid and edge minima where f is convex via (15), generating G
if negative point found **then**
  return A is not copositive
$\mathcal{L}_\ell := \{2^n - 1\}, \mathcal{R} := \emptyset, \ell := n$
while $\ell > 2$ **do**
  $\mathcal{L}_{\ell-1} := \emptyset, \mathcal{N} := \emptyset$
  while $\mathcal{L}_\ell \neq \emptyset$ **do**
    Retrieve k from $\mathcal{L}_\ell$
    if $A_k \geq 0$ **then**
      Save k in $\mathcal{R}$
    else if $\exists i$ for which (17) holds, the minimum is on the facet without $e_{\mathcal{I}_k(i)}$ **then**
      Save m of facet $F_m \subset F_k$ without $e_{\mathcal{I}_k(i)}$ in $\mathcal{L}_{\ell-1}$
      Save m of facets $F_m \subset F_k$ including $e_{\mathcal{I}_k(i)}$ in $\mathcal{N}$
    else
      if G_k is complete **then**
        if H_k is PSD **then**
          Solve (8) to find $x_k^* \in F_k$
          if $x_k^{*T} A_k x_k^* < 0$ **then**
            return A is not copositive
          else
            Save k in $\mathcal{R}$
      else
        Save all facets F_m: $F_m \subset F_k$ in $\mathcal{L}_{\ell-1}$
  $\mathcal{L}_{\ell-1} := \mathcal{L}_{\ell-1} \setminus \mathcal{N}$
  Eliminate from $\mathcal{L}_{\ell-1}$ all sub-faces $F_m : F_m \subseteq F_k, k \in \mathcal{R}$
  $\ell := \ell - 1$
return A is copositive

level (maximum clique) corresponding to the convexity graph where the minimum is nonnegative.

4 Numerical investigation

The most appropriate way to traverse the face graph for copositivity testing, depends on the convexity graph of the instance under consideration. On one hand there is the density of the convexity graph, i.e. on how many edges f is strictly convex, and on the other hand the level ℓ on which the minimum of StQP can be found in case the matrix is copositive. These two number are related. We first illustrate the behavior with cases from literature based on the maximum clique problem. Then we vary more systematically the level ℓ on which the minimum can be found and the density of the convexity graph. The instances can be found in the appendices.

Computer time is relative to the computational platform used. The algorithms were implemented in Matlab 2016b using routines to run standard Cholesky decomposition and solving the linear set of equations (8). They were run on an i5 CPU on a desktop computer.

Algorithm 3 Copositivity detection searching bottom-up (TUp)

Require: A: symmetric $n \times n$ matrix.
if A is entry-wise nonnegative or A is PSD **then**
    return A is copositive
Evaluate A_{ii}, centroid and edge minima where f is convex via (15), generating G
if negative point found **then**
    return A is not copositive
$\ell := 2$ and $\mathcal{L}_2$ includes all edges where f is strictly convex
while $\ell < n - 1$ **do**
    $\mathcal{L}_{\ell+1} := \emptyset$
    while $\mathcal{L}_\ell \neq \emptyset$ **do**
        Retrieve m from $\mathcal{L}_\ell$
        for each vertex $e_j \notin F_m$ **do**
            Add vertex e_j to F_m generating F_k
            if $k \notin \mathcal{L}_{\ell+1}$ **and** G_k is complete **then**
                if not $A_k \geq 0$ **then**
                    if H_k is PSD **then**
                        Solve (8) to find $x_k^* \in F_k$
                        if $x_k^{*T} A_k x_k^* < 0$ **then**
                            return A is not copositive
                Add k to $\mathcal{L}_{\ell+1}$
    $\ell := \ell + 1$
return A is copositive

A first example is due to Hall and Newman (1963) and called the Horn matrix. It is copositive in dimension $n = 5$; the corresponding face graph is depicted in Fig. 1. The traversal variants show a varying behavior. TDk visits all faces in 0.01s. detecting monotonicity and does not require the PSD check of Cholesky. TDown visits all faces by level in 0.02s. TUp detects that none of the faces on level $\ell = 3$ corresponds to a complete graph G_k (clique) in 0.05s, such that the edge minima of 0 determine the global minimum. No face on a higher level is investigated.

4.1 Measuring performance of face graph traversal on max-clique instances

Part of benchmark cases in literature are based on the maximum clique problem according to the following relation: Let ω be the clique number of a graph defined by its adjacency matrix A_G. One way to find it is to determine the minimum integer value t such that $(t-1)\mathbf{1}\mathbf{1}^T - tA_G$ is copositive, i.e. the clique number is

$$\omega := \min\{t : [(t-1)\mathbf{1}\mathbf{1}^T - tA_G] \text{ is copositive}\}. \tag{18}$$

Despite that there are better ways to determine the clique number, we used instances from maximum clique DIMACS challenge (http://archive.dimacs.rutgers.edu/Challenges/) of dimensions n =14, 16 and 28. Instance 1tc.16.clique ($n = 16$) was converted to a clique from Challenge Problems (https://oeis.org/A265032/a265032.html). Notice that this type of instances do no exhibit edges where f is strictly concave, as for (13), we have that $A_{ii} + A_{jj} - 2A_{ij} = 2t - t(A_G)_{ij} - 2 \geq 0$ for $t \geq 2$. This also implies that the matrix $\tilde{A}$ discussed in Corollary 2 is $\tilde{A} = A$.

Table 1 Results of Algorithm 1 (TDk) on DIMACS max-clique instances

n	t	d%	ℓ^*	#Eval	#Mon	#An	#PSD	# f	T	Cpos
14	3	68	5	1467	1427	0	2	2	0.31 s	✗
14	4	68	5	1467	1427	0	2	2	0.25 s	✗
14	5	68	5	7495	7169	0	22	22	1.05 s	✓
16	6	88	8	4920	4918	0	2	2	0.66 s	✗
16	7	88	8	4920	4918	0	2	2	0.65 s	✗
16	8	88	8	26,764	16,617	0	147	147	3.44 s	✓
28	3	59	4	93,341,409	39,975,812	68	2	2	11,451 s	✗
28	4	59	4	180,630,459	85,812,632	445	106	106	21,672 s	✓

For the instances, we have the following characteristics:

– n: dimension
– t: parameter for the clique number ω
– $d\%$: density of the convexity graph measured as the percentage of edges (excluding diagonal elements of C) on which f is strictly convex
– ℓ^*: level on which the minimum point of StQP can be found, or a negative point is found

Running the algorithms, We measure the following indicators:

– #Eval: number of evaluated faces
– #Mon: number of times f is monotonous on a face
– #An: number of times the evaluated A_k was completely nonnegative
– #PSD: number of times PSD evaluation of H_k was performed using Cholesky decomposition
– # f: number of function evaluations of an interior optimum
– T: running time of the algorithm
– Cpos: copositivity has been proven.

Algorithm TDk runs over the list of $2^n - 1$ faces and marks them with respect to monotonicity detection and the existence of higher level faces that may be all nonnegative or have a nonnegative relative interior optimum. In Table 1, we can observe that each time the PSD status has been checked with Cholesky, H_k appeared PSD, the solution of (8) resulted in an interior point that has been evaluated. Therefore, we leave out this column for the same instances in the other face-based algorithm results.

The largest instance to solve is Johnson8-2-4 (http://archive.dimacs.rutgers.edu/Challenges/), with $n = 28$ and max clique number $\omega = 4$. Computationally, if each marker Tag_k only requires one byte, the algorithm requires 2^{28} bytes, i.e. 0.5 GB, just to store the marker. For $t = 3$, A is not copositive and the TDk algorithm requires 3 h 11 min to find an interior negative minimum at level $\ell = 4$. For $t = 4$, the matrix is copositive and the matlab implementation of the algorithm requires 6 h to run over the complete list.

The idea of Algorithm TDown is to use sets $\mathcal{R}$ and $\mathcal{N}$ not to store information on all faces. Theoretically, this is an elegant idea and the monotonicity is passed on to

Table 2 Results of Algorithm 2 (TDown) on DIMACS max-clique instances

n	t	d%	ℓ^*	#Eval	$\|\mathcal{R}\|$	#Mon	#An	$\#f$	T	Cpos
14	3	68	5	289	0	126	0	2	**0.06 s**	✗
14	4	68	5	289	0	126	0	2	**0.05 s**	✗
14	5	68	5	319	5	152	0	6	**0.09 s**	✓
16	6	88	8	145	0	8	0	2	**0.03 s**	✗
16	7	88	8	145	0	8	0	2	**0.01 s**	✗
16	8	88	8	145	1	8	0	2	**0.01 s**	✓
28	3	59	4	–	–	–	–	–	> 24 h	✗
28	4	59	4	–	–	–	–	–	> 24 h	✓

We also measure $|\mathcal{R}|$: number of faces in global list $\mathcal{R}$

Table 3 Results of Algorithm 3, TUp on DIMACS max-clique instances

n	t	d%	ℓ^*	#Eval	$\#f$	#An	T	Cpos
14	3	68	4	177	73	0	0.09 s	✗
14	4	68	5	207	103	0	0.10 s	✗
14	5	68	5	209	105	0	0.11 s	✓
16	6	88	7	1423	1287	0	0.69 s	✗
16	7	88	8	1459	1324	0	0.66 s	✗
16	8	88	8	1461	1326	0	0.67 s	✓
28	3	59	4	827	422	0	**1.21 s**	✗
28	4	59	4	931	526	0	**1.44 s**	✓

next levels in a more systematic way. However, computationally the algorithm may get stuck if the list $\mathcal{L}_\ell$ gets larger. Table 2 shows this effect for the largest instance, where suddenly the algorithm is not successful anymore; it looses a lot of time in managing the lists. One should also take into account that level $\ell = 14$ alone contains $\binom{28}{14} = 40{,}116{,}600$ faces. Therefore, the time required by TDown is practically larger than that of algorithm TDk for the largest instance, whereas for the instances up to $n = 16$ it is the fastest of the three traversal variants due to the efficient use of the monotonicity information.

Algorithm TUp works upwards. As all instances have relative interior optima on a relatively low level, the number of faces to be checked is very low. Table 3 shows that for all measured instances, the algorithm requires less than 2 seconds. It surprised the authors that the TDown implementation walking down from $n = 14, 16$ could be faster than the TUp implementation. The latter requires more PSD tests, but this is due to a compiled Cholesky routine, which is faster than the monotonicity test which is based on a matlab script. Apparently, one can reduce the number of facets to be evaluated either by proving convexity over the facet or by monotonicity from top-down. The implementation of TDown gets stuck in handling the large list $\mathcal{L}_\ell$ on each level.

Table 4 Results of the three algorithms varying the density of the convexity graph. Instances can be found in "Appendix B". We also measured #KKT: number of times KKT point determined and measured the effect of replacing A by $\tilde{A}$ according to Corollary 2

n	d%	ℓ^*	Alg	#Eval	#PSD	KKT	# f	#Mon	#An	T
11	75	6	TDk	915	14	9	3	649	164	0.13 s
			TDk $\tilde{A}$	732	13	8	3	633	136	0.12 s
			TDown	466	13	8	3	122	6	0.07 s
			TDown $\tilde{A}$	181	13	8	3	36	8	**0.03 s**
			TUp	237	74	27	2	–	97	0.06 s
11	85	8	TDk	954	49	22	3	774	188	0.13 s
			TDk $\tilde{A}$	792	48	22	3	679	136	0.12 s
			TDown	364	53	25	3	118	4	0.06 s
			TDown $\tilde{A}$	190	44	19	3	35	4	**0.03 s**
			TUp	455	222	94	9	–	167	0.10 s
11	93	10	TDk	825	153	60	6	672	132	0.12 s
			TDk $\tilde{A}$	825	153	60	6	672	132	0.12 s
			TDown	206	111	48	6	26	4	**0.03 s**
			TDown $\tilde{A}$	206	111	48	6	26	4	**0.03 s**
			TUp	1091	761	234	14	–	264	0.18 s
16	70	6	TDk	22,039	33	19	4	17,546	1030	4.28 s
			TDk $\tilde{A}$	19,140	27	17	4	18,385	816	4.13 s
			TDown	8693	33	19	4	4132	14	2.07 s
			TDown $\tilde{A}$	529	22	14	4	125	6	**0.08 s**
			TUp	630	132	47	4	–	362	0.32 s
16	84	9	TDk	26,104	265	67	5	21,427	1139	4.25 s
			TDk $\tilde{A}$	25,514	265	67	5	22,224	1235	4.20 s
			TDown	8670	273	69	6	3467	22	2.32 s
			TDown $\tilde{A}$	6862	288	70	6	2,985	16	1.80 s
			TUp	3122	1485	209	8	–	1528	**1.35 s**
16	94	15	TDk	22,416	1532	112	5	20,862	710	4.12 s
			TDk $\tilde{A}$	22,416	1532	112	5	20,862	710	4.12 s
			TDown	644	379	74	5	123	7	**0.12 s**
			TDown $\tilde{A}$	644	379	74	5	123	7	**0.12 s**
			TUp	33,030	23,546	543	9	–	9438	9.19 s

4.2 Face graph traversal on instances with a varying density convexity graph

Scozzari and Tardella (2008) show us that random matrices for n=10, 30, 50, 100, 200, 500, 1000 and 1500, with a control on the density of the convexity graph of 0.25, 0.5 and 0.75 can be generated according to a description in Bomze and De Klerk (2002) and Nowak (1999). The findings report that one can solve problems with a density of 0.25 up to dimension $n = 500$, a density of 0.5 up to $n = 200$ and a density of 0.75 up to $n = 100$. Intuitively, this provides the idea that the bottom-up approach is more appropriate for low density convexity graphs and the top-down approach for cases where the density graph is dense.

To measure this effect, we generated the instances following Nowak (1999) varying in dimension and convexity graph density that can be found in "Appendix B". First of all, they are all copositive, so the algorithms solve the StQP. As we can directly observe from the instances, is that the matrices contain more variation in the numbers, i.e. implying a higher condition number providing the phenomenon, that now the matrix H_k is not necessarily PSD when the graph G_k is complete and the computed KKT point is not necessarily interior of a face, so evaluation is not necessary. The occurrence of monotonicity is far bigger for these random instances than for the max-clique instances, leading to less computation time (fewer faces are evaluated) for the same dimension. For those instances, we also evaluated the effect of using Corollary 2 by changing A_{ij} into $\tilde{A}_{ij}$ when f is concave over edge e_i, e_j. This appeared not to make any difference for the matrices with the highest density convexity graph, as concavity hardly occurs. However, for lower density, using $\tilde{A}$ instead of A appears to increase the effectiveness of the monotonicity check drastically.

The top-down traversal algorithms profit from an increasing density. They have to check more the PSD of H_k with higher density as can be observed from Table 4. However, they make use of monotonicity in order not to check the complete face graph. What is typical for those instances is that the level-wise traversal of TDown, which looked hopeless for the largest DIMACS instance due to list management, is now faster because it does a more systematic elimination of monotone faces. Basically, the number of checked faces decreases drastically. Checking the faces in index order TDk from top to down leads to less efficiency in concluding on the monotonicity than the TDown traversal. Moreover, as can be observed, this is helped by using matrix $\tilde{A}$ instead of A.

For the bottom-up traversal TUp, the work starts to be harder with increasing density, as more faces are kept on the list as their convexity graph G_k is complete. In computing time, the bottom-up approach gets slower than the top-down traversal variants for the denser instances. For the instances where the optimum is at a relatively low level, the TDown traversal is harder and costs more time. Moreover, the computing time depends a lot on how well the management of the lists is organized. We can observe, that TDown and also TUp require far more time if the number of faces to be evaluated increases.

5 Conclusions

This paper derives several properties on the monotonicity and convexity of the standard quadratic function f over faces and subsets of the standard simplex. It illustrates that monotonicity can be applied in face-based copositivity detection algorithms which traverse the face graph of the standard simplex top-down. We found that randomly generated instances provide a different characteristic in terms of monotonicity and convexity than maximum clique based instances. The latter have a lot of symmetry and no edges on which f is concave.

We show that the success of a top-down or bottom-up traversal of the face graph depends not only on the density of the convexity graph, but also on the level on which strictly convex faces can be discovered and on how well the list management overhead can be reduced by efficient implementations. A level-wise implementation looked hopeless for the larger symmetric maximum clique based instances due to list management overhead, but appeared a very systematic and efficient approach for the randomly generated instances. A transformation of the matrix of the StQP towards a linearization over edges on which f is concave, helped a lot in the monotonicity tests and reduces the number of faces to be investigated and the total computational time.

The implementations of the algorithms used for the illustration are based on easily available matrix subroutines such as the Cholesky decomposition, but do not exploit a lot the management of lists. The monotonicity considerations reduce the number of Cholesky calls drastically. From a computer science perspective there is still ample opportunities to improve the implementation of the algorithms to obtain a computing time comparable to earlier published results. From this perspective, we are looking into the parallelization of the face-based algorithms.

A Matrix instances from literature

Example 1 Horn matrix, $n = 5$ from Hall and Newman (1963).

$$A := \begin{pmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{pmatrix}$$

Example 2 Brock14, $n = 14$ from The Second DIMACS Implementation Challenge. Generated with *grahgen -g14 -c5 -p0.5 -d0 -s0*.

$$A_G := \begin{pmatrix} 0&1&1&0&1&1&0&1&0&1&0&1&0&1\\ 1&0&1&1&1&0&0&1&0&1&0&1&0&0\\ 1&1&0&0&1&1&0&0&1&0&1&1&1&1\\ 0&1&0&0&1&0&1&0&0&1&1&1&0&1\\ 1&1&1&1&0&1&1&1&1&0&0&0&1&0\\ 1&0&1&0&1&0&1&1&0&1&0&1&1&1\\ 0&0&0&1&1&1&0&0&1&1&0&0&1&1\\ 1&1&0&0&1&1&0&0&0&0&1&0&1&1\\ 0&0&1&0&1&0&1&0&0&1&1&0&1&1\\ 1&1&0&1&0&1&1&0&1&0&1&0&1&1\\ 0&0&1&1&0&0&0&1&1&1&0&1&0&0\\ 1&1&1&1&0&1&0&0&0&0&1&0&0&1\\ 0&0&1&0&1&1&1&1&1&1&0&0&0&1\\ 1&0&1&1&0&1&1&1&1&1&0&1&1&0 \end{pmatrix},$$

$A := (t-1)\mathbf{1}\mathbf{1}^T - tA_G$ is copositive for $t = 5$.

Example 3 1tc.16.clique, $n = 16$ from oeis.org/A265032/a265032.html.

$$A_G := \begin{pmatrix} 0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 1&0&0&1&0&1&1&1&1&1&1&1&1&1&1&1\\ 1&0&0&1&0&1&1&1&0&1&1&1&1&1&1&1\\ 1&1&1&0&1&0&0&1&1&0&1&1&1&1&1&1\\ 1&0&0&1&0&1&1&1&0&1&1&1&1&1&1&1\\ 1&1&1&0&1&0&0&1&1&0&0&1&1&1&1&1\\ 1&1&1&0&1&0&0&1&1&0&0&1&0&1&1&1\\ 1&1&1&1&1&1&1&0&1&1&1&0&1&0&1&1\\ 1&1&0&1&0&1&1&1&0&1&1&1&1&1&1&1\\ 1&1&1&0&1&0&0&1&1&0&0&1&0&1&1&1\\ 1&1&1&1&1&0&0&1&1&0&0&1&0&1&1&1\\ 1&1&1&1&1&1&1&0&1&1&1&0&1&0&0&1\\ 1&1&1&1&1&1&0&1&1&0&0&1&0&1&1&1\\ 1&1&1&1&1&1&1&0&1&1&1&0&1&0&0&1\\ 1&1&1&1&1&1&1&1&1&1&1&1&0&1&0&0&1\\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0 \end{pmatrix},$$

$A = (t-1)\mathbf{1}\mathbf{1}^T - tA_G$ is copositive for $t = 8$.

B Variable density matrices

Example 4 Ivo-n10-dens0.75-dvert20, $n = 11$ from Nowak (1999).

$$A := \begin{pmatrix} 3.49 & 9.56 & 8.31 & 13.41 & 3.06 & 1.32 & 7.44 & 9.93 & 3.35 & -2.75 & 11.72 \\ 9.56 & 4.51 & 4.76 & -2.24 & 5.72 & 6.46 & 14.40 & -3.11 & 5.64 & 2.84 & 12.22 \\ 8.31 & 4.76 & 12.18 & 11.51 & 10.16 & 10.82 & 17.88 & 3.92 & 11.68 & 0.32 & 16.06 \\ 13.41 & -2.24 & 11.51 & 9.61 & 7.32 & 7.97 & 8.34 & -0.43 & 6.84 & 2.31 & 14.77 \\ 3.06 & 5.72 & 10.16 & 7.32 & 18.02 & 11.87 & 12.42 & 15.59 & 5.29 & 5.12 & 18.98 \\ 1.32 & 6.46 & 10.82 & 7.97 & 11.87 & 14.83 & 7.11 & 6.85 & 9.68 & 10.04 & 17.38 \\ 7.44 & 14.40 & 17.88 & 8.34 & 12.42 & 7.11 & 17.42 & 15.64 & 4.16 & 9.59 & 18.68 \\ 9.93 & -3.11 & 3.92 & -0.43 & 15.59 & 6.85 & 15.64 & 8.59 & 6.11 & 1.47 & 14.26 \\ 3.35 & 5.64 & 11.68 & 6.84 & 5.29 & 9.68 & 4.16 & 6.11 & 8.53 & 4.97 & 14.23 \\ -2.75 & 2.84 & 0.32 & 2.31 & 5.12 & 10.04 & 9.59 & 1.47 & 4.97 & 7.52 & 13.73 \\ 11.72 & 12.22 & 16.06 & 14.77 & 18.98 & 17.38 & 18.68 & 14.26 & 14.23 & 13.73 & 19.94 \end{pmatrix}$$

Density $= 0.746$, max-clique $= 6$ of convexity graph, minimum at level $\ell = 3$ of $f^* = 0.85$

Example 5 Ivo-n10-dens0.95-dvert20, $n = 11$ from Nowak (1999).

$$A := \begin{pmatrix} 3.49 & -0.44 & 8.31 & 3.41 & 3.06 & 1.32 & 7.44 & -0.07 & 3.35 & -2.75 & 11.72 \\ -0.44 & 4.51 & 4.76 & -2.24 & 5.72 & 6.46 & 14.40 & -3.11 & 5.64 & 2.84 & 12.22 \\ 8.31 & 4.76 & 12.18 & 11.51 & 10.16 & 10.82 & 7.88 & 3.92 & 1.68 & 0.32 & 16.06 \\ 3.41 & -2.24 & 11.51 & 9.61 & 7.32 & 7.97 & 8.34 & -0.43 & 6.84 & 2.31 & 14.77 \\ 3.06 & 5.72 & 10.16 & 7.32 & 18.02 & 11.87 & 12.42 & 5.59 & 5.29 & 5.12 & 18.98 \\ 1.32 & 6.46 & 10.82 & 7.97 & 11.87 & 14.83 & 7.11 & 6.85 & 9.68 & 10.04 & 17.38 \\ 7.44 & 14.40 & 7.88 & 8.34 & 12.42 & 7.11 & 17.42 & 15.64 & 4.16 & 9.59 & 18.68 \\ -0.07 & -3.11 & 3.92 & -0.43 & 5.59 & 6.85 & 15.64 & 8.59 & 6.11 & 1.47 & 14.26 \\ 3.35 & 5.64 & 1.68 & 6.84 & 5.29 & 9.68 & 4.16 & 6.11 & 8.53 & 4.97 & 14.23 \\ -2.75 & 2.84 & 0.32 & 2.31 & 5.12 & 10.04 & 9.59 & 1.47 & 4.97 & 7.52 & 13.73 \\ 11.72 & 12.22 & 16.06 & 14.77 & 18.98 & 17.38 & 18.68 & 14.26 & 14.23 & 13.73 & 19.94 \end{pmatrix}$$

Density $= 0.855$, max-clique $= 8$ of convexity graph, minimum at level $\ell = 4$ of $f^* = 0.80$

Example 6 Ivo-n10-dens1-dvert20, $n = 11$ from Nowak (1999).

$$A := \begin{pmatrix} 3.49 & -0.44 & -1.69 & 3.41 & 3.06 & 1.32 & 7.44 & -0.07 & 3.35 & -2.75 & 11.72 \\ -0.44 & 4.51 & 4.76 & -2.24 & 5.72 & 6.46 & 4.40 & -3.11 & 5.64 & 2.84 & 12.22 \\ -1.69 & 4.76 & 12.18 & 1.51 & 10.16 & 10.82 & 7.88 & 3.92 & 1.68 & 0.32 & 16.06 \\ 3.41 & -2.24 & 1.51 & 9.61 & 7.32 & 7.97 & 8.34 & -0.43 & 6.84 & 2.31 & 14.77 \\ 3.06 & 5.72 & 10.16 & 7.32 & 18.02 & 11.87 & 12.42 & 5.59 & 5.29 & 5.12 & 18.98 \\ 1.32 & 6.46 & 10.82 & 7.97 & 11.87 & 14.83 & 7.11 & 6.85 & 9.68 & 10.04 & 17.38 \\ 7.44 & 4.40 & 7.88 & 8.34 & 12.42 & 7.11 & 17.42 & 5.64 & 4.16 & 9.59 & 18.68 \\ -0.07 & -3.11 & 3.92 & -0.43 & 5.59 & 6.85 & 5.64 & 8.59 & 6.11 & 1.47 & 14.26 \\ 3.35 & 5.64 & 1.68 & 6.84 & 5.29 & 9.68 & 4.16 & 6.11 & 8.53 & 4.97 & 14.23 \\ -2.75 & 2.84 & 0.32 & 2.31 & 5.12 & 10.04 & 9.59 & 1.47 & 4.97 & 7.52 & 13.73 \\ 11.72 & 12.22 & 16.06 & 14.77 & 18.98 & 17.38 & 18.68 & 14.26 & 14.23 & 13.73 & 19.94 \end{pmatrix}$$

Density $= 0.927$, max-clique $= 10$ of convexity graph, minimum at level $\ell = 4$ of $f^* = 0.8$

Example 7 Ivo-n15-dens0.75-dvert20, $n = 16$ from Nowak (1999).

$$A := \begin{pmatrix} 12.12 & 17.40 & 10.93 & 16.98 & 7.06 & 2.57 & 10.71 & 13.14 & 11.83 & 6.03 & 7.51 & 4.99 & 9.58 & 4.35 & 15.00 & 15.86 \\ 17.40 & 11.55 & 0.53 & 8.96 & 11.30 & 10.74 & 8.52 & 6.29 & 17.28 & 7.54 & 12.13 & 4.47 & 8.35 & 3.02 & 6.11 & 15.57 \\ 10.93 & 0.53 & 8.81 & -1.06 & 10.88 & 2.50 & 7.54 & 2.30 & 15.12 & 4.65 & 1.79 & 3.61 & 8.62 & 3.90 & 8.78 & 14.20 \\ 16.98 & 8.96 & -1.06 & 8.12 & 15.39 & -0.41 & 8.85 & 4.80 & 5.90 & 9.24 & 6.83 & 7.48 & 15.71 & 11.26 & 5.81 & 13.86 \\ 7.06 & 11.30 & 10.88 & 15.39 & 17.41 & 20.84 & 16.23 & 14.34 & 9.05 & 12.50 & 4.12 & 25.79 & 19.86 & 8.17 & 14.06 & 18.50 \\ 2.57 & 10.74 & 2.50 & -0.41 & 20.84 & 8.69 & 10.40 & 4.52 & 11.41 & 11.92 & 0.52 & 7.42 & 3.86 & 6.32 & 8.88 & 14.14 \\ 10.71 & 8.52 & 7.54 & 8.85 & 16.23 & 10.40 & 15.36 & 17.74 & 16.62 & 16.20 & 13.28 & 8.12 & 21.88 & 15.49 & 9.31 & 17.48 \\ 13.14 & 6.29 & 2.30 & 4.80 & 14.34 & 4.52 & 17.74 & 6.38 & 4.64 & 6.26 & 9.76 & 4.08 & 11.65 & 4.07 & 18.14 & 12.99 \\ 11.83 & 17.28 & 15.12 & 5.90 & 9.05 & 11.41 & 16.62 & 4.64 & 16.85 & 15.60 & 7.85 & 9.39 & 16.41 & 10.34 & 13.01 & 18.22 \\ 6.03 & 7.54 & 4.65 & 9.24 & 12.50 & 11.92 & 16.20 & 6.26 & 15.60 & 16.45 & 11.35 & 15.85 & 11.54 & 7.16 & 6.60 & 18.02 \\ 7.51 & 12.13 & 1.79 & 6.83 & 4.12 & 0.52 & 13.28 & 9.76 & 7.85 & 11.35 & 10.05 & 8.99 & 18.42 & -2.02 & 4.20 & 14.82 \\ 4.99 & 4.47 & 3.61 & 7.48 & 25.79 & 7.42 & 8.12 & 4.08 & 9.39 & 15.85 & 8.99 & 16.45 & 20.58 & 6.00 & 19.81 & 18.02 \\ 9.58 & 8.35 & 8.62 & 15.71 & 19.86 & 3.86 & 21.88 & 11.65 & 16.41 & 11.54 & 18.42 & 20.58 & 18.13 & 1.89 & 7.41 & 18.87 \\ 4.35 & 3.02 & 3.90 & 11.26 & 8.17 & 6.32 & 15.49 & 4.07 & 10.34 & 7.16 & -2.02 & 6.00 & 1.89 & 2.99 & -0.57 & 11.30 \\ 15.00 & 6.11 & 8.78 & 5.81 & 14.06 & 8.88 & 9.31 & 18.14 & 13.01 & 6.60 & 4.20 & 19.81 & 7.41 & -0.57 & 11.02 & 15.31 \\ 15.86 & 15.57 & 14.20 & 13.86 & 18.50 & 14.14 & 17.48 & 12.99 & 18.22 & 18.02 & 14.82 & 18.02 & 18.87 & 11.30 & 15.31 & 19.60 \end{pmatrix}$$

Density $= 0.7$, max-clique $= 6$ of convexity graph, minimum at level $\ell = 4$ of $f^* = 1.47$

Example 8 Ivo-n15-dens0.95-dvert20, $n = 16$ from Nowak (1999).

$$A := \begin{pmatrix} 12.12 & 7.40 & 10.93 & 6.98 & 7.06 & 2.57 & 10.71 & 3.14 & 11.83 & 6.03 & 7.51 & 4.99 & 9.58 & 4.35 & 15.00 & 15.86 \\ 7.40 & 11.55 & 0.53 & 8.96 & 11.30 & 10.74 & 8.52 & 6.29 & 7.28 & 7.54 & 2.13 & 4.47 & 8.35 & 3.02 & 6.11 & 15.57 \\ 10.93 & 0.53 & 8.81 & -1.06 & 10.88 & 2.50 & 7.54 & 2.30 & 5.12 & 4.65 & 1.79 & 3.61 & 8.62 & 3.90 & 8.78 & 14.20 \\ 6.98 & 8.96 & -1.06 & 8.12 & 15.39 & -0.41 & 8.85 & 4.80 & 5.90 & 9.24 & 6.83 & 7.48 & 5.71 & 1.26 & 5.81 & 13.86 \\ 7.06 & 11.30 & 10.88 & 15.39 & 17.41 & 20.84 & 16.23 & 14.34 & 9.05 & 12.50 & 4.12 & 15.79 & 9.86 & 8.17 & 14.06 & 18.50 \\ 2.57 & 10.74 & 2.50 & -0.41 & 20.84 & 8.69 & 10.40 & 4.52 & 11.41 & 11.92 & 0.52 & 7.42 & 3.86 & -3.68 & 8.88 & 14.14 \\ 10.71 & 8.52 & 7.54 & 8.85 & 16.23 & 10.40 & 15.36 & 17.74 & 6.62 & 6.20 & 13.28 & 8.12 & 21.88 & 15.49 & 9.31 & 17.48 \\ 3.14 & 6.29 & 2.30 & 4.80 & 14.34 & 4.52 & 17.74 & 6.38 & 4.64 & 6.26 & -0.24 & 4.08 & 11.65 & 4.07 & 8.14 & 12.99 \\ 11.83 & 7.28 & 5.12 & 5.90 & 9.05 & 11.41 & 6.62 & 4.64 & 16.85 & 15.60 & 7.85 & 9.39 & 16.41 & 10.34 & 13.01 & 18.22 \\ 6.03 & 7.54 & 4.65 & 9.24 & 12.50 & 11.92 & 6.20 & 6.26 & 15.60 & 16.45 & 11.35 & 15.85 & 11.54 & 7.16 & 6.60 & 18.02 \\ 7.51 & 2.13 & 1.79 & 6.83 & 4.12 & 0.52 & 13.28 & -0.24 & 7.85 & 11.35 & 10.05 & 8.99 & 8.42 & -2.02 & 4.20 & 14.82 \\ 4.99 & 4.47 & 3.61 & 7.48 & 15.79 & 7.42 & 8.12 & 4.08 & 9.39 & 15.85 & 8.99 & 16.45 & 10.58 & 6.00 & 19.81 & 18.02 \\ 9.58 & 8.35 & 8.62 & 5.71 & 9.86 & 3.86 & 21.88 & 11.65 & 16.41 & 11.54 & 8.42 & 10.58 & 18.13 & 1.89 & 7.41 & 18.87 \\ 4.35 & 3.02 & 3.90 & 1.26 & 8.17 & -3.68 & 15.49 & 4.07 & 10.34 & 7.16 & -2.02 & 6.00 & 1.89 & 2.99 & -0.57 & 11.30 \\ 15.00 & 6.11 & 8.78 & 5.81 & 14.06 & 8.88 & 9.31 & 8.14 & 13.01 & 6.60 & 4.20 & 19.81 & 7.41 & -0.57 & 11.02 & 15.31 \\ 15.86 & 15.57 & 14.20 & 13.86 & 18.50 & 14.14 & 17.48 & 12.99 & 18.22 & 18.02 & 14.82 & 18.02 & 18.87 & 11.30 & 15.31 & 19.60 \end{pmatrix}$$

Density $= 0.842$, max-clique $= 9$ of convexity graph, minimum at level $\ell = 4$ of $f^* = 0.401$

Example 9 Ivo-n15-dens1-dvert20, $n = 16$ from Nowak (1999).

$$A := \begin{pmatrix} 12.12 & 7.40 & 0.93 & 6.98 & 7.06 & 2.57 & 10.71 & 3.14 & 11.83 & 6.03 & 7.51 & 4.99 & 9.58 & 4.35 & 5.00 & 15.86 \\ 7.40 & 11.55 & 0.53 & 8.96 & 11.30 & 0.74 & 8.52 & 6.29 & 7.28 & 7.54 & 2.13 & 4.47 & 8.35 & 3.02 & 6.11 & 15.57 \\ 0.93 & 0.53 & 8.81 & -1.06 & 10.88 & 2.50 & 7.54 & 2.30 & 5.12 & 4.65 & 1.79 & 3.61 & 8.62 & 3.90 & 8.78 & 14.20 \\ 6.98 & 8.96 & -1.06 & 8.12 & 5.39 & -0.41 & 8.85 & 4.80 & 5.90 & 9.24 & 6.83 & 7.48 & 5.71 & 1.26 & 5.81 & 13.86 \\ 7.06 & 11.30 & 10.88 & 5.39 & 17.41 & 10.84 & 16.23 & 4.34 & 9.05 & 12.50 & 4.12 & 15.79 & 9.86 & 8.17 & 14.06 & 18.50 \\ 2.57 & 0.74 & 2.50 & -0.41 & 10.84 & 8.69 & 10.40 & 4.52 & 11.41 & 11.92 & 0.52 & 7.42 & 3.86 & -3.68 & 8.88 & 14.14 \\ 10.71 & 8.52 & 7.54 & 8.85 & 16.23 & 10.40 & 15.36 & 7.74 & 6.62 & 6.20 & 3.28 & 8.12 & 11.88 & 5.49 & 9.31 & 17.48 \\ 3.14 & 6.29 & 2.30 & 4.80 & 4.34 & 4.52 & 7.74 & 6.38 & 4.64 & 6.26 & -0.24 & 4.08 & 11.65 & 4.07 & 8.14 & 12.99 \\ 11.83 & 7.28 & 5.12 & 5.90 & 9.05 & 11.41 & 6.62 & 4.64 & 16.85 & 15.60 & 7.85 & 9.39 & 16.41 & 0.34 & 13.01 & 18.22 \\ 6.03 & 7.54 & 4.65 & 9.24 & 12.50 & 11.92 & 6.20 & 6.26 & 15.60 & 16.45 & 11.35 & 15.85 & 11.54 & 7.16 & 6.60 & 18.02 \\ 7.51 & 2.13 & 1.79 & 6.83 & 4.12 & 0.52 & 3.28 & -0.24 & 7.85 & 11.35 & 10.05 & 8.99 & 8.42 & -2.02 & 4.20 & 14.82 \\ 4.99 & 4.47 & 3.61 & 7.48 & 15.79 & 7.42 & 8.12 & 4.08 & 9.39 & 15.85 & 8.99 & 16.45 & 10.58 & 6.00 & 9.81 & 18.02 \\ 9.58 & 8.35 & 8.62 & 5.71 & 9.86 & 3.86 & 11.88 & 11.65 & 16.41 & 11.54 & 8.42 & 10.58 & 18.13 & 1.89 & 7.41 & 18.87 \\ 4.35 & 3.02 & 3.90 & 1.26 & 8.17 & -3.68 & 5.49 & 4.07 & 0.34 & 7.16 & -2.02 & 6.00 & 1.89 & 2.99 & -0.57 & 11.30 \\ 5.00 & 6.11 & 8.78 & 5.81 & 14.06 & 8.88 & 9.31 & 8.14 & 13.01 & 6.60 & 4.20 & 9.81 & 7.41 & -0.57 & 11.02 & 15.31 \\ 15.86 & 15.57 & 14.20 & 13.86 & 18.50 & 14.14 & 17.48 & 12.99 & 18.22 & 18.02 & 14.82 & 18.02 & 18.87 & 11.30 & 15.31 & 19.60 \end{pmatrix}$$

Density $= 0.943$, max-clique $= 15$ of convexity graph, minimum at level $\ell = 5$ of $f^* = 0.401$

References

Bomze IM, De Klerk E (2002) Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. J Glob Optim 24(2):163–185

Bundfuss S, Dür M (2008) Algorithmic copositivity detection by simplicial partition. Linear Algebra Appl 428(7):1511–1523

Burer S (2009) On the copositive representation of binary and continuous nonconvex quadratic programs. Math Program 120(2):479–495

Challenge problems: independent sets in graphs. https://oeis.org/A265032/a265032.html

Gondzio J, Yildirim EA (2018) Global solutions of nonconvex standard quadratic programs via mixed integer linear programming reformulations. Technical Report ERGO-18-022, School of Mathematics, The University of Edinburgh

Hall M, Newman M (1963) Copositive and completely positive quadratic forms. Math Proc Camb Philos Soc 59(2):329–339

Hendrix EMT, Salmerón JMG, Casado LG (2019) On function monotonicity in simplicial branch and bound. AIP Conf Proc 2070(1):020007

Kaplan W (2000) A test for copositive matrices. Linear Algebra Appl 313(1–3):203–206

Liuzzi G, Locatelli M, Piccialli V (2019) A new branch-and-bound algorithm for standard quadratic programming problems. Optim Methods Softw 34(1):79–97

Murty KG, Kabadi SN (1987) Some NP-complete problems in quadratic and nonlinear programming. Math Program 39(2):117–129

Nie J, Yang Z, Zhang X (2018) A complete semidefinite algorithm for detecting copositive matrices and tensors. SIAM J Optim 28(4):2902–2921

Nowak I (1999) A new semidefinite programming bound for indefinite quadratic forms over a simplex. J Glob Optim 14(4):357–364

Povh J, Rendl F (2007) A copositive programming approach to graph partitioning. SIAM J Optim 18(1):223–241

Salmerón JMG (2019) High performance computing to solve global optimization problems. Ph.D. thesis, Universidad de Almería

Salmerón JMG, Casado LG, Hendrix EMT (2018) Paralelización de la detección de una matriz copositiva mediante la evaluación de las facetas de un simplex unidad. In: Libro de Abstracts de las Jornadas SARTECO 2018. Universidad de Zaragoza, pp 77–82

Scozzari A, Tardella F (2008) A clique algorithm for standard quadratic programming. Discrete Appl Math 156(13):2439–2448

Second DIMACS implementation challenge. http://archive.dimacs.rutgers.edu/Challenges/

Väliaho H (1986) Criteria for copositive matrices. Linear Algebra Appl 81:19–34

Yang S, Li X (2009) Algorithms for determining the copositivity of a given symmetric matrix. Linear Algebra Appl 430(2–3):609–618

Žilinskas J, Dür M (2011) Depth-first simplicial partition for copositivity detection, with an application to maxclique. Optim Methods Softw 26(3):499–510