On monotonicity and search traversal in copositivity detection algorithms*

Eligius M.T. Hendrix[a], Leocadio G. Casado[b] and Boglarka G.-Toth[c]

[a] Computer Architecture, Universidad de Málaga, eligius@uma.es
[b] Informatics, Universidad de Almería, leo@ual.es
[c] Informatics, Szeged University, boglarka@inf.szte.hu

Abstract

Matrix copositivity has an important theoretical background. Over the last decades, the use of algorithms to check copositivity has made a big progress. Methods are based on 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 unit simplex and illustrate with numerical instances that exploiting monotonicity in algorithms may speed up the search. For face-based algorithms the question is what traversal through the face graph of the unit simplex is more appropriate for which matrix instance.
keywords Copositive matrix, Unit simplex, monotone, face.

1 Introduction

Copositivity provides an important theoretical foundation in combinatorial and quadratic optimization [5, 9, 13, 17]. Given the unit simplex with n vertices

$$\Delta_n = \{x \in \mathbb{R}^n_+ \mid \sum_{i=1}^{n} x_i = 1\} \tag{1}$$

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

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

and noncopositive if

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

One can determine if a matrix is positive semidefinite (PSD) via its eigenvalues. To determine whether a matrix is copositive has been shown to be an NP complete problem [11].

The certification of copositivity is directly related to the standard quadratic program (StQP)

$$f^* := \min_{x \in \Delta_n} f(x) := x^T Ax. \tag{4}$$

*This work has been funded by grants from the Spanish Ministry (RTI2018-095993-B-100), 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.

The StQP is generic in the sense that optimizing a quadratic function $f(x) = x^T Qx + b^T x + c$ is equivalent to (4), due to the transformation $A = Q + \frac{1}{2}(\mathbf{1}b^T + b\mathbf{1}^T)$, were $\mathbf{1}$ 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 mainly related to low dimensional cases [18]. The spatial branch and bound (B&B) algorithm introduced by [4] can either certify that a matrix is not copositive, or prove it is so-called ϵ-copositive. Following such a procedure, [19] 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 (4). A recent work, that also compares with B&B approaches is [10] focusing on the first order conditions, i.e. Karush-Kuhn-Tucker (KKT) conditions of the optima of (4) applying convex and linear bounds in a B&B context implicitly enumerating KKT points. One of our sub-questions is what would happen with the original B&B in [4] if it would take monotonicity considerations [8] into account in the branching procedure.

Also recent is the work of [6] which translates the KKT conditions to a MIP type of approach making use of fast integer programming implementations in order to solve the StQP (4) for instances of hundreds of variables. The mentioned procedures do not focus on the second order considerations to find an optimum.

An older work, [16] derives algorithms based on the observation that a minimum point of (4) can only be on the so-called relative interior of a face of the unit 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 unit simplex and edges where 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 goes over what we call the face graph of the unit simplex as depicted in Figure 1. The nodes of the face graph consist of set $\mathcal{F} := \{1, \ldots, 2^n - 1\}$ and an edge (m, k) is given when the bitstring $k_{(2}$ differs in one bit from $m_{(2}$. So the top node corresponds to the unit simplex itself, and faces are connected with an edge if one is the facet of the other. The bitstring $k_{(2}$ corresponds to face F_k in Figure 1 and indicates which elements are considered positive, $x_i > 0$. The number of elements $\ell = \sum_{i=1}^{n} (k_{(2)})_i$ gives the level in the graph, i.e. the dimension of the face. Level ℓ has $\binom{n}{\ell}$ faces. In this context, the procedure of [16] first marks nodes on level $\ell = 2$ (edges of the unit simplex) as strictly convex and then tries to identify faces as high-level as possible with all edges convex to find interior minima, as low-value 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, [16] test whether the matrix A_k related to the elements of face 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 former algorithms and modifications and traversal variants of a face-based algorithm. Section 4 uses several benchmark instances to illustrate numerically the effectiveness of monotonicity and face graph traversal. The

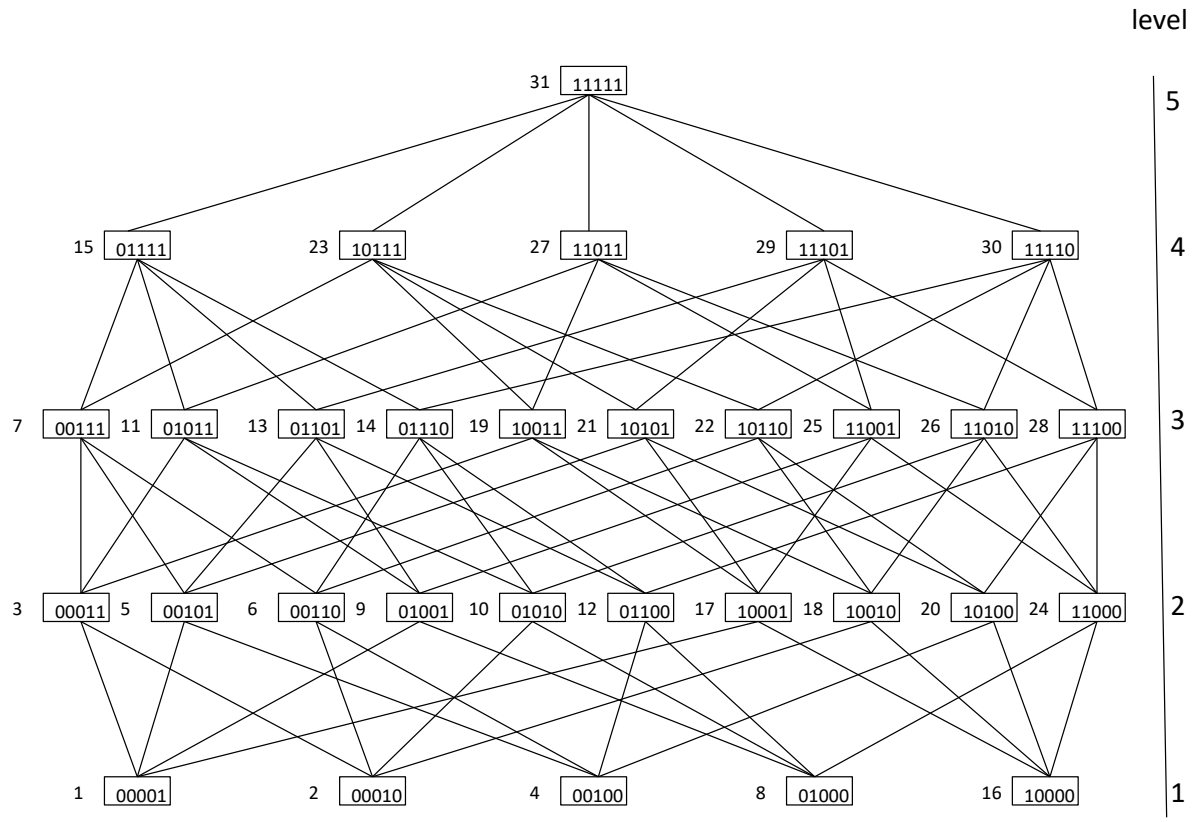

Figure 1: Face graph of the unit simplex for $n = 5$, where faces F_k are numbered by index k and bitstring $k_{(2}$ indicates which elements are considered positive and which zero.

main conclusions and future research questions are described in Section 5.

2 Properties of a StQP

In our notation, we use as identity matrix in dimension n the symbol $E_n = (e_1, \ldots, e_n)$ and $\mathbf{1}$ represents the all-ones vector in appropriate dimension. Moreover, we use $D_n = E_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T = (d_1, \ldots, d_n)$ as the projection matrix on the zero sum plane $\mathcal{P} := \{x \in \mathbb{R}^n | \mathbf{1}^T x = 0\} =< D_n >$. 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 $k_{(2}$, A_k is the $\ell \times \ell$ matrix with $\ell \leq n$ and elements of the rows and columns i of A, selected when $(k_{(2})_i = 1$.*

This means that the optimization over a face $\min_{x \in F_k} x^T Ax$ 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 (4) are used in the studies of [6, 10, 15, 16]. For a local minimum point $x \geq 0$ of (4) there exist values of the dual variables μ and $\lambda_i \geq 0$ such that

$$Ax = \mu\mathbf{1} + E_n\lambda, \; x^T\lambda = 0. \tag{5}$$

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. [6] shows that the StQP (4) can be solved for hundreds

of variables using Mixed Integer Programming (MIP) applying binary variables to capture the complementarity. [10] 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 of (4) Δ_n does not have an interior. Consider the relative interior $\text{rint}(\Delta_n)$ of Δ_n as

$$\text{rint}(\Delta_n) = \{x \in \mathbb{R}^n \mid \sum_{j=1}^{n} x_j = 1;\ x_j > 0, \quad j = 1, \ldots, n\}. \tag{6}$$

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

$$A_k y^* - \mu \mathbf{1} = 0,\ \mathbf{1}^T y^* = 1, \tag{7}$$

where y^* is mostly in a lower dimensional space than n. Translation of the solution to n-dimensional space requires adding zeros on the positions where $(k_{(2)})_i = 0$. So, either the global minimum point of StQP can be found in one of the vertices of the unit 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. [16] characterize this by looking for faces where A_k is positive definite (PD). However, this is not a necessary condition. For instance, matrix

$$A = \begin{pmatrix} 0 & -1 & -2 \\ -1 & 1 & -3 \\ -2 & -3 & 2 \end{pmatrix} \tag{8}$$

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

Proposition 1. If matrix $H = D_n A D_n$ is positive semidefinite (PSD), then 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 $0 \leq \lambda \leq 1$

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

This means that for any $x, y \in \Delta_n$ and $0 \leq \lambda \leq 1$, $(1-\lambda)f(x) + \lambda f(y) \geq f((1-\lambda)x + \lambda y)$. □
The other way around, to have a relative interior optimum, H should be PSD.

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

Proof. Following the KKT conditions (5), only the constraint $\mathbf{1}^T x = 1$ is binding, i.e. there exists a multiplier μ such that $Ax^* = \mu \mathbf{1}$. As $D_n = E_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, we have

$$D_n A x^* = D_n \mu \mathbf{1} = \mu[E_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T]\mathbf{1} = \mu[\mathbf{1} - \frac{n}{n}\mathbf{1}] = 0. \tag{9}$$

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

$$f(x^* + D_n r) = f(x^*) + 2(D_n A x^*)^T r + r^T H r \geq f(x^*). \tag{10}$$

As $D_n A x^* = 0$, we have $\forall r \in B(0, \delta), r^T H r \geq 0$, 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. In algorithmic context, [16] and our implementations test whether A_k or H_k are PD or PSD applying routines on Cholesky decomposition.

The analysis of [16] focuses on convexity of the edges of a face. Function f is strictly convex over edge (i, j) in Δ_n if

$$A_{ii} + A_{jj} - 2A_{ij} > 0. \tag{11}$$

Definition 2. *The convexity graph $G = (N, C)$ of matrix A with nodes $N = \{1, \ldots, n\}$ and edge $(i, j) \in C$ if (11) is valid, i.e. f is strictly convex over the edge between e_i and e_j [16].*

Actually, a node can be removed from G if it has no 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. Consider matrix

$$A_k = \begin{pmatrix} 8 & 5 & 3 \\ 5 & 3 & 3 \\ 3 & 3 & 4 \end{pmatrix} \tag{12}$$

The corresponding convexity graph G_k is complete, but f is not convex on F_k; nor A_k nor H_k are PSD.

Proposition 2 can be extended to any face F_k if we consider the unit simplex Δ_ℓ in dimension ℓ (number of positive elements in face k).

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

The result can be extended for simplicial B&B to any simplex $S = \text{conv}(V)$ where V is the vertex matrix:

$$S := \{x = V\lambda, \lambda \in \Delta_n\} \tag{13}$$

and $Q = V^T A V$, such that $f(x) = x^T A x$ corresponds to $\lambda^T V^T A V \lambda = \lambda^T Q \lambda$.

Corollary 2. *If $\exists x^* \in \text{argmin}_S f(x) \cap \text{rint}(S)$, then $\exists \lambda^* \in \text{rint}(\Delta_n), D_n Q \lambda^* = D_n V A x^* = 0$ and $D_n Q D_n$ is positive semidefinite.*

Although more complicated, this analysis also extends to lower dimensional faces. However, the question is at which face the minimum is attained. For that, we focus on directional derivatives $D_n A x$ and $D_n Q \lambda$.

2.2 Monotonicity considerations

One of our questions is whether monotonicity can be used to speed up the search in simplicial B&B methods like that of [4] that use simplicial partition sets S based on vertex matrix $V := (v_1, \ldots, v_n)$. We elaborated this theoretically in more general terms in [8]. Here we investigate what this enhances for the StQP specifically.

Proposition 3. Let V be a vertex matrix, simplex S from (13), matrix D_n with columns d_i and $Q = V^T AV$. If $\exists i\ d_i^T Q \geq 0$, then $\exists x^* \in \operatorname{argmin}_S f(x)$ at the facet of S that corresponds to $\lambda_i = 0$.

Proof. Assume the minimum is attained at $x^* = V\lambda^*$ with $\lambda_i^* > 0$. Now consider the point $x = V\lambda$ with $\lambda = \lambda^* - \frac{n\lambda_i^*}{n-1} d_i$ which is on the facet corresponding to $\lambda_i = 0$. One can write

$$f(x) = x^{*T} Ax^* + (x - x^*)^T A(x + x^*) = f(x^*) - \frac{n\lambda_i^*}{n-1} d_i^T Q(2\lambda^* - \frac{n\lambda_i^*}{n-1} d_i).$$

Analysing the right hand term, one can observe that $\lambda_i^* > 0$, $d_i^T Q \geq 0$ and component wise $2\lambda^* - \frac{n\lambda_i^*}{n-1} d_i > 0$. This means that only in the specific case that $d_i^T Q = 0$ we have $f(x) = f(x^*)$, else $f(x) < f(x^*)$ which shows that x is a minimum point on the facet which corresponds to $\lambda_i = 0$. □

The importance of this theoretical result is that, for the StQP (4) and the proof on copositivity, one can reduce the search in a B&B algorithm like [4] to a facet of subset S. For a face-based algorithm, the analysis is easier, as it says we can drop out vertex e_i from the face we are investigating. Actually, this is only relevant when searching the face graph in Figure 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.

Specifically, for a spatial B&B approach, similar to interval arithmetic, the monotonicity observation can lead to a rejection of the sub-simplex S, if the facet we would like to reduce the search to, is a bit off the relative boundary of the unit simplex Δ_n. The conditions are a bit cumbersome, but the following proposition provides us a test to reject a sub-simplex in a spatial B&B. Consider again the vertex matrix $V = (v_1, v_2, \ldots, v_n)$ and a strictly positive directional derivative. Then it can be shown for some conditions on V that the minimum over Δ_n is not in S.

Proposition 4. Let V be a vertex matrix, simplex S from (13) with centroid $c = \frac{1}{n} V\mathbf{1}$ and $Q = V^T AV$ and $\exists i, d_i^T Q > 0$. If for all elements m with $v_{im} > 0$ also $v_{jm} > 0$ for all other vertices $j \neq i$, then the minimum point of (4) (in Δ_n) cannot be found in S.

Proof. Consider $\epsilon = \min_{m,j}\{\frac{v_{jm}}{v_{im}} | v_{im} > 0\}$ and direction $r = -Vd_i = c - v_i$. A minimum point $x^* = V\lambda^*$ can be found according to Proposition 3 on a facet where $\lambda_i^* = 0$. Due to the condition for all elements m with $v_{im} > 0$ also $v_{jm} > 0$ we have that the corresponding component value $x_m^* > 0$. Consider stepping into direction r according to $x = x^* + \rho r = x^* + \rho(c - v_i)$, $0 \leq \rho \leq \epsilon$. Notice that $x^T\mathbf{1} = 1$. Componentwise we observe that if $v_{im} = 0$ we have $x_{im} \geq 0$ and if $v_{im} > 0$, we have $x_{im} \geq 0$ due to the condition and definition of step size ϵ. So $x = x^* + \rho r \in \Delta_n$ is feasible and r a feasible direction from x^*.

The directional derivative $r^T \nabla f(x^*) = -2d_i^T V^T Ax^* = -2d_i^T Q\lambda^* < 0$, so r is a feasible descent direction from x^* on Δ_n implying that x^* cannot be minimum point of (4). □

The conditions of Proposition 4 do not exclude the possibility that f has a negative value in S, but it says there exist lower regions of f on Δ_n, because the minimum of (4) is not attained in S.

2.3 Bundfuss-Dür property

The basic exploited poperty in [4] qualifies simplex S based on vertex matrix V as copositive (i.e. $x^T Ax \geq 0, x \in S$) if $Q = V^T AV$ is a nonnegative matrix. This is relatively easy to see, as for $x \in S$, $f(x) = x^T Ax = \lambda^T V^T AV\lambda = \lambda^T Q\lambda$. So having Q nonnegative and λ too, $f(x) \geq 0$.

The B&B procedure focuses on the edge (v_i, v_j) with the most negative value $\min_{ij} v_i^T Av_j = \min_{ij} Q_{ij}$ as lower bound and uses that edge for further bisecting along its minimum point.

Given that $f(v_i), f(v_j) \geq 0$ and $v_i^T A v_j < 0$, the minimum is located at $x_{ij}^* = v_i + \xi^*(v_j - v_i)$ with

$$\xi^* = \frac{f(v_i) - v_i^T A v_j}{f(v_i) + f(v_j) - 2v_i^T A v_j} \tag{14}$$

and

$$f(x_{ij}^*) = f(v_i) - \frac{(f(v_i) - v_i^T A v_j)^2}{f(v_i) + f(v_j) - 2v_i^T A v_j}. \tag{15}$$

Basically, the algorithm in [4], chooses as a bisection point x_{ij}^* with some safeguards. Besides candidates for the minimum on the edges of Δ_n, this procedure also samples points in the relative interior $\mathrm{rint}(\Delta_n)$ and other faces. However, these are not necessarily local minimum points of StQP.

3 Algorithms

Algorithm 1 Spatial B&B ϵ-copositive detection (A, ϵ)

Require: A: matrix, ϵ: accuracy.
if diagonal elements A are nonnegative **then**
  if $\min_{ij} Q_{ij} > -\epsilon$ **then**
    return A is ϵ-copositive
else
  return A is not copositive
If $\exists i,\ d_i^T A \geq 0$, Δ_n is reduced to a facet F_k
$\Lambda := \{\Delta_n\}$ {List of sub-simplices to be evaluated}
while $\Lambda \neq \{\emptyset\}$ **do**
  Retrieve a simplex S from Λ $\{\Lambda := \Lambda \backslash S\}$
  Select an edge $(i, j) \in \mathrm{argmin}\, Q_{ij}$
  Determine x_{ij} via (16) and $f(x_{ij})$
  if $f(x_{ij}) < 0$ **then**
    return A is not copositive
  Bisect S into S_1 and S_2 over x_{ij}
  for $m = 1, 2$ **do**
    Determine $Q = V^T A V$ for S_m
    if $\min_{ij} Q_{ij} < -\epsilon$ **then**
      check monotonicity via $d_i^T Q$ to reduce to a facet
      if S_m does not fulfill Proposition 4 **then**
        Store S_m in Λ
return A is ϵ-copositive

First, we implemented the findings on monotonicity in the B&B algorithm of [4] in order to observe the consequences numerically. Second, we developed three traversal variants of the face graph of Δ_n which evaluate the vertices, i.e. the diagonal elements of A, the centroid and proven interior minimum points of faces F_k.

1. CDK, follows in decreasing order of k the faces and checks whether they should be investigated. This requires at least for each face to store a marker.

2. Cdown, tries to avoid storage by checking the faces in a list for each level and walks down the levels of the face graph.

3. Cup, follows the line of the algorithm of [16] in working upwards into the face graph. However, in contrast it works also level wise, as negative points may be found in local minima on lower levels.

3.1 Bundfuss-Dür algorithm

The variant of the spatial B&B algorithm in [4] depicted here applies a depth-first search to reduce memory requirement, as suggested in [19]. We add the monotonicity test of f on simplex S in order to decrease its dimension. In fact, the algorithm determines either that A is not copositive, or A is ϵ-copositive.

The minimum point x_{ij}^* on an edge where f is convex may be out of bounds, such that ξ^* in (14) is safeguarded and the bisection point $x_{ij} = v_i + \xi(v_j - v_i)$ is given by

$$\xi = \max\{\frac{v_i^T A v_j}{v_i^T A v_j - f(v_i)}, \min\{\xi^*, \frac{f(v_j)}{f(v_j) - v_i^T A v_j}\}\}. \tag{16}$$

The bounds in (16) ensure $v_i^T A x_{ij} \geq 0$ and $x_{ij}^T A v_j \geq 0$ (see [4]).

Before storing the new simplices in the list Λ, the properties 3 and 4 can be used iteratively to either reduce the dimension to a facet or to discard the new sub-simplex. In case S is reduced to just one vertex, it can be discarded because $Q_{ii} \geq 0$.

3.2 Traversal variants of a face-based algorithm

We first check the matrix A to see whether it is copositive due to

- $\forall i, j,\ A_{ij} \geq 0$, i.e. it is all positive
- A is PSD (Cholesky decomposition).

Then the following actions are taken:

- Check diagonal elements $A_{ii} \geq 0$, i.e f is positive 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 (11), meanwhile calculating their minima given by (16) ($\ell = 2$).

The CDK version in face-based Algoritm 2 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

- Tag_k=0: face k has not been checked,
- Tag_k=1: face k has been checked and
- Tag_k=2: no need to check face 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-faces m of a copositive face k are also copositive, we do not have to evaluate the sub-faces 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 k where A_k is copositive. The higher in the graph, the more nodes we do not have to evaluate.

Computationally, Algorithm 2 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

Algorithm 2 Copositivity detection, in decreasing k order (CDK)

Require: A: symmetric $n \times n$ matrix, $n \geq 3$.

if A is positive or A is PSD **then**
  return A is copositive
Evaluate A_{ii}, centroid and edge minima where f is convex via (16), 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**
      Tag_m:=2 for $F_m \subset F_k$
      break
    if $\exists i$, $D_{\ell i}^T A_k > 0$, i.e. f is monotone in direction d_i **then**
      Tag_m:=1 for facets $F_m \subset F_k$ including e_i
      if there is more than one monotone direction **then**
        Mark all facets $F_m \subset F_k$ as Tag_m:=1
    else
      if G_k is complete **then**
        if H_k is positive semidefinite **then**
          Determine $x_k^* \in F_k$ solving (7)
          if ${x^*}^T A_k x^* < 0$ **then**
            return A is not copositive
          else
            Tag_m:=2 for all faces $F_m \subset F_k$
  $k := k$-1
return A is copositive

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.

The idea of Algorithm 3 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 f. Two lists of candidate faces are active $\mathcal{L}_\ell$ and $\mathcal{L}_{\ell-1}$.

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 that cannot have a minimum due to monotonicity. Let us note that sub-faces of $\mathcal{N}$ are still considered, as the optimum of a monotonous face is on the boundary, while none of the sub-faces of $k \in \mathcal{R}$ can contain negative minima, so all the sub-faces can be dropped. In this way, no sub-face of $k \in \mathcal{R}$ nor facet $k \in \mathcal{N}$ has to be included in the next level list $\mathcal{L}_{\ell-1}$. In the worst case, the list $\mathcal{L}_\ell$ may still be huge with increasing dimension n. Most troublesome, however, is the management overhead of the lists.

Algorithm 4, CUp, follows the upward search of the algorithm of [16] in their search for the minimum of (4) 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 to check whether H_k is PSD in order to have a possible interior optimum on face k. The algorithm stops if it finds a negative interior optimum or alternatively has found the highest level (maximum clique) corresponding to the convexity graph where the minimum is positive.

Algorithm 3 Copositivity detection top-down in the face graph (CDown)

Require: A: symmetric $n \times n$ matrix.

if A is positive or A is PSD **then**
  return A is copositive
Evaluate A_{ii}, centroid and edge minima where f is convex via (16), 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 $\exists i$, $D_{\ell i}^T A_k > 0$, i.e. f monotonous in direction d_i **then**
      Save m of the facet $F_m \subset F_k$ without e_i in $\mathcal{L}_{\ell-1}$
      Save m of facets $F_m \subset F_k$ with $e_i = 1$ in $\mathcal{N}$
    else
      if G_k is complete **then**
        if H_k is PSD **then**
          Solve (7) to find $x_k^* \in F_k$
          if $x^{*T} A_k x^* < 0$ **then**
            return A is not copositive
          else
            Save k in $\mathcal{R}$
      else
        Save all facets 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 $m : F_m \subseteq F_k, k \in \mathcal{R}$
  $\ell := \ell - 1$
return A is copositive

4 Numerical illustration

For the investigation on the effect of including monotonicity tests in spatial B&B and on the question of what is the most appropiate way to traverse the face graph for copositivity testing, we first selected smaller dimensional cases from literature. We first illustrate the effect on including the monotonicity in Section 4.1. More results and a discussion can be found in the PhD thesis [14]. This is followed by studying the behavior of face-based algorithms on practical instances.

Computer time is relative to the computational platform used. Due to its time consuming behavior, the B&B was coded in gcc version 8.2.1 using g++ and run on a PC with Fedora 29 Linux with Kernel 4.18.17-300, having four Intel® Core(TM) i7-4770K CPU @ 3.50GHz with 16 GB RAM. The accuracy was taken as $\epsilon = 10^{-6}$. The face-based algorithms were implemented in Matlab 2016b exploiting the facility to run the standard Cholesky decomposition and solving the linear set of equations (7) and run it on an i5 CPU on a desktop computer.

4.1 Effect of monotonicity in spatial B&B

In the two first tables with respect to the B&B, we measure the number or times that all matrix $Q = V^T AV > -\epsilon$ ($\#Q > -\epsilon$), the number of bisections (#Div), number of simplex

Algorithm 4 Copositivity detection searching bottom-up (CUp)

Require: A: symmetric $n \times n$ matrix.

if A is positive or A is PSD **then**
  return A is copositive
Evaluate A_{ii}, centroid and edge minima where f is convex via (16), 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 (7) to find $x_k^* \in F_k$
            if $x^{*T} A_k x^* < 0$ **then**
              return A is not copositive
        Add k to $\mathcal{L}_{\ell+1}$
  $\ell = \ell + 1$
return A is copositive

evaluations (#Eval), number of function evaluations ($\#f$) and computing time (T). We also measure in Table 2 the number of times the function was found to be monotonous on a subsimplex (#Mon). Depth-first search was used in order to reduce the memory demand of the algorithms.

The very **first example**, $n = 3$, is the graphical illustrative case in [4] on how the subdivision works in spatial B&B. In this very small case, one can observe an effect of the monotonicity test reducing the number of necessary evaluations comparing the first row of both tables. The face-based algorithms are less interesting here, as they directly detect that A is PSD and therefore also copositive. The algorithm of [16] first determines that all edges are strictly convex and then finds that the minimum is 0 in the centroid.

The **second illustrative example** is due to [7] and called the Horn matrix. It is copositive in dimension $n = 5$, such that its faces correspond to Figure 1. Comparing the second line in the two tables, monotonicity does not occur; it is a very symmetric instance. The face-based algorithms now show a difference in behavior. Algorithm 2 (CDK) visits all faces in 0.01s. detecting monotonicity and does not require the PSD check of Cholesky. Algorithm 3 (CDown) visits all faces by level in 0.02s. Algorithm 4 (CUp) 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 and no face on a higher level has to be investigated.

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

$$\omega = \min\{t : [(t-1)\mathbf{1}\mathbf{1}^T - tAdj] \text{ is copositive}\}. \tag{17}$$

We used some instances from literature. despite that there are much better ways to determine

Table 1: Results for the original algorithm of [4] in Algorithm 1 without monotonicity test (lines 17 to 20), number of times that matrix $Q = V^T AV > -\epsilon$ ($\#Q > -\epsilon$), bisections (#Div), simplex evaluations (#Eval), function evaluations ($\#f$) and running time (T).

n	t	$\#Q > -\epsilon$	#Div	#Eval	$\#f$	T	Copos
3	.	8	7	15	90	0.02s	✓
5	.	10	9	19	285	0.02s	✓
8	2	0	1	2	72	0.02s	✗
8	3	268	275	551	19,836	0.02s	✗
8	4	7,015	7,014	14,029	505,044	0.5s	✓
14	3	392	407	815	85,575	0.12s	✗
14	4	16,661	16,667	33,354	3,502,170	3.93s	✗
14	5	1,959,335	1,959,334	3,918,669	411,460,245	463.03s	✓
15	1	0	1	1	120	0.02s	✗
15	2	0	1	2	240	0.02s	✗
15	3	74,116	74,115	148,231	17,787,720	22.46s	✓
16	3	19	63	127	17,272	0.06s	✗
16	4	1	19	39	5,304	0.3s	✗
16	5	.	.	.	.	>10m	.
16	6	.	.	.	.	>10m	.
16	7	.	.	.	.	>10m	.
16	8	.	.	.	.	>10m	.

the clique number. The **third example** is based on such a case and can be found in the appendix. It is of dimension $n = 8$, and the corresponding marix A is copositive for $t = 4$. For that value, we can now observe that indeed the monotonicity inclusion reduces the number of function evaluations drastically from about 505 thousand to 359 thousand in the spatial B&B. The algorithm of [16] would find a minimum at level $\ell = 5$.

The face graph traversal Algorithms 2, 3 and 4 detect that A is PSD and therefore copositive. For the values of $t = 2, 3$, the algorithms detect that the centroid has a negative value and therefore do not start a search over the faces. To come back to our research question, we try to observe what happens with the monotonicity effectiveness for spatial B&B when we increase dimension of the cases of (17). In the appendix several instances can be found for $n = 14, 15$ and $n = 16$. Brock14 (n=14), Johnson6-2-4 ($n = 15$) originate from [2]. The instance 1tc.16.clique (n=16) was converted to a clique from [1].

It feels difficult to detect a fixed advantage of the monotonicity test for these clique based cases, although with long searches to prove ϵ-copositivity, the test seems more profitable. We can observe that when the parameter value t increases and we are getting closer to copositivity, more sub-simplices are found to have a monotonous objective function and the simplex can be shrunk to a lower dimensional face. Some of the larger cases that could not be solved by the original algorithm of [4] within a time limit of 10 minutes can be solved when including the monotonicity test. Our experiments confirmed the findings of [19], that when n reaches the value of 20, copositivity detection by B&B becomes difficult in a reasonable time.

The max-clique based instances provide a certain symmetry in the used numbers and condition number of the matrices. Following the custom of using random matrices, we also generated 6 instances, using the code described in [12] with *-dvert*=20 and different *-dens* values. Appendix B gives six matrix instances of dimension $n = 11, 16$ with varying density. The minimum of the StQP (4) is positive and can be found around level $\ell = 4$, whereas due to the variation in the density of the convexity graph, the maximum level (max-clique) where G_k is complete increases with the density. The matrices are thus copositive, creating hard instances for copositivity de-

Table 2: Results of Algorithm 1 (with monotonicity test in lines 17 to 20), number of times that all matrix $Q = V^T AV > -\epsilon$ ($\#Q > -\epsilon$), bisections (#Div), simplex evaluations (#Eval), function evaluations ($\#f$), times the monotonicity was found on a sub-simplex and computing time (T).

n	t	$\#Q > -\epsilon$	#Div	#Eval	$\#f$	#Mon	T	Copos
3	.	6	5	13	66	2	0.02s	✓
5	.	10	9	19	285	0	0.02s	✓
8	2	0	1	2	72	0	0.02s	✗
8	3	10	17	44	1,062	9	0.02s	✗
8	4	5,139	5,138	11,851	359,047	1,574	0.4s	✓
14	3	392	407	815	85,575	0	0.02s	✗
14	4	13,993	14,009	28,068	2,925,419	50	3.5s	✗
14	5	1,925,338	1,925,337	3,856,382	403,866,696	5,707	468.5s	✓
15	1	0	1	1	120	0	0.02s	✗
15	2	0	1	2	240	0	0.02s	✗
15	3	74,116	74,115	148,297	17,791,094	66	20.85s	✓
16	3	0	28	60	7,089	4	0.03s	✗
16	4	1	19	39	5,304	0	0.03s	✗
16	5	91,010	91,038	184,684	24,973,605	2,607	29.31s	✗
16	6	140,437	140,458	293,369	36,831,666	12,452	43.33s	✗
16	7	140,094	140,112	290,675	37,180,458	10,450	44.12s	✗
16	8	-	-	-	-	-	>10m	-

Table 3: Results for the original Algorithm 1 in [4] without monotonicity test (lines 17 to 20), number of times that all matrix $Q = V^T AV > -\epsilon$ ($\#Q > -\epsilon$), bisections (#Div), simplex evaluations (#Eval), function evaluations ($\#f$) and computing time (T) in seconds.

n	ℓ	$\#Q > -\epsilon$	#Div	#Eval	$\#f$	T
11	6	12	11	23	1,518	0.04s
11	8	18	17	35	2,310	0.04s
11	10	36	35	71	4,686	0.08s
16	6	9	8	17	2,312	0.02s
16	9	19	18	37	5,032	0.02s
16	15	19	18	37	5,032	0.02s

tection algorithms. On the other hand, the minimum f is positive, leaving space for the B&B algorithm to reject larger parts based on the lower bound of sub-simplices.

It is typical to observe that for those instances, the monotonicity is apparently reached in a high point of the search tree and further not occuring anymore. This means that the instances have far more global monotonicity than the max-clique cases cutting away large part of the search space from the beginning.

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

The two types of approaches, spatial B&B and face-based search are not really comparable. Earlier research of [16] has already shown that face-based approaches may be far more efficient for larger dimension when solving the StQP and more recent publications [6] and [10] have shown that other approaches may do equally good.

To focus on the second research question with respect to face graph traversal variants, we first extend the experiments to the maximum clique DIMACS challenge with dimensions $n = 14, 16$

Table 4: Results of Algorithm 1 (with monotonicity test in lines 17 to 20), number of times that all matrix $Q = V^T AV > -\epsilon$ ($\#Q > -\epsilon$), bisections (#Div), simplex evaluations (#Eval), function evaluations ($\#f$), times sub-simplex monotonicity was found and computing time in seconds (T).

n	t	$\#Q > -\epsilon$	#Div	#Eval	$\#f$	#Mon	T
11	6	12	11	24	1,331	1	0.02s
11	8	18	17	36	1,991	1	0.02s
11	10	36	35	72	3,971	1	0.02s
16	6	9	8	18	2,176	1	0.02s
16	9	19	18	38	4,576	1	0.02s
16	15	19	18	38	4,576	1	0.02s

Table 5: Results of Algorithm 2 (CDK) on DIMACS max-clique instances. #: number of times #Eval: check face; #Mon: f is monotonous on a face; #Ap: A_k completely nonnegative; #Chol: PSD check of H_k via Cholesky; $\#f$: function evaluations; level: where the best function value found; T: running time; Copos: the matrix is copositive.

n	t	#Eval	#Mon	#Ap	Chol	$\#f$	level	T	Copos
14	3	3,243	16	0	2	2	5	0.16s	✗
14	4	3,243	16	0	2	2	5	0.16s	✗
14	5	15,824	202	3	28	28	.	0.68s	✓
16	6	11,870	0	0	2	2	8	0.52s	✗
16	7	11,870	0	0	2	2	8	0.55s	✗
16	8	64,086	14	3	46	46	.	2.98s	✓
28	3	134,177,458	13,441	52	2	2	4	5,459s	✗
28	4	268,338,5668	50,142	370	106	106	.	10,810s	✓

and $n = 28$. We measure the following indicators: #Eval: number of evaluated faces; #Mon: number of times f is monotonous on a face; #Ap: A_k appeared completely nonnegative; Chol: number of times PSD evaluation of H_k was performed with Cholesky decomposition; level: on which the best function evaluation (negative in case of non-copositivity) has been performed; T: running time of the algorithm; Copos: copositivity has been proven.

The algorithms have been implemented in Matlab R2016 in order to make use of standard procedures and they were run on a desktop PC with i5 CPU. This also means that computational time is not comparable with the B&B implementation, but rather in between the traversal variants.

Algorithm CDK 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 positive (nonnegative) or have a positive relative interior optimum. For the instances, we can observe that each time the PSD status has been checked with Cholesky, H_k appeared PSD, the solution of (7) 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 [2], with n=28 and max clique of t=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 CDK algorithm requires 1.5 hours to find an interior negative minimum at level $\ell = 4$. For t=4, the matrix is copositive and the algorithm requires 3 hours to run over the complete list.

The idea of Algorithm CDown is to use sets $\mathcal{R}$ and $\mathcal{N}$ not to store info on all faces. Theo-

Table 6: Results of Algorithm 3 (CDown) on DIMACS max-clique instances. Number of #Eval: evaluated faces; $|\mathcal{R}|$: faces in $\mathcal{R}$;#Mon: times f is monotonous on a face; #Ap: A_k completely nonnegative; $\#f$: funcion evaluations; level: where the best function value found; T: running time; Copos: the matrix is copositive.

n	t	#Eval	$\|\mathcal{R}\|$	#Mon	#Ap	$\#f$	level	T	Copos
14	3	14,567	0	85	0	2	5	2.34s	✗
14	4	14,567	0	111	0	2	5	2.67s	✗
14	5	16,053	33	203	3	31	.	3.14s	✓
16	6	36,639	0	0	0	2	8	9.71s	✗
16	7	36,639	0	0	0	2	8	9.71s	✗
16	8	64,226	48	17	3	46	.	14.1s	✓
28	3	.	.	.	.	.	.	>24h	✗
28	4	.	.	.	.	.	.	>24h	✓

retically, this might be an elegant idea, but computationally one can observe that the algorithm 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 CDown is practically larger than that of algorithm CDK for these instances.

Table 7: Results of Algorithm 4, CUp on DIMACS max-clique instances. #Eval: number of evaulated faces; $\#f$: funcion evaluations; level: where the best function value found;#Ap: A_k completely nonnegative; T: running time; Copos: the matrix is copositive.

n	t	#Eval	$\#f$	level	#Ap	T	Copos
14	3	177	73	4	0	0.10s	✗
14	4	207	103	5	0	0.13s	✗
14	5	203	105	6	0	0.16s	✓
16	6	1423	1287	7	0	0.73s	✗
16	7	1459	1324	8	0	0.74s	✗
16	8	1459	1326	8	0	0.76s	✓
28	3	827	422	4	0	1.18s	✗
28	4	931	526	4	0	1.42s	✓

Algorithm CUp works upwards. As all instances have relative interior optima on a relatively low level, the number of faces to be checked is very low. For all measured instances, the algorithm requires less than 2 seconds.

4.3 Face graph traversal on varying density matrices

Literature [16], shows 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 y 0.75 can be generated according to a description in [3, 12]. 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 earlier described instances varying in dimension and convexity graph density that can be found in Appendix B.

Table 8: Results of top-down traversal algorithms on varying density instances 7,..,12, ℓ: max-clique of G, number of..Chol: calls to Cholesky to check H_k is PSD; KKT: H_k is PSD, KKT point determined; $\#f$: interior optimum evaluated; #Mon: f is monotonous on F_k; #Ap: matrix A_k is completely nonnegative

		CDK						CDown					
n	ℓ	Chol	KKT	$\#f$	#Mon	#Ap	T	Chol	KKT	$\#f$	#Mon	#Ap	T
11	6	9	4	3	413	118	0.17s	15	5	2	85	6	0.14s
11	8	39	15	8	497	135	0.15s	54	13	3	99	4	0.14s
11	10	177	31	12	560	146	0.15s	305	47	6	131	4	0.16s
16	6	29	7	1	11,784	967	3.74s	44	13	4	1,612	4	3.32s
16	9	373	31	2	14,110	712	3.56s	570	52	5	1,969	6	4.03s
16	15	9,537	113	2	14,010	713	3.62s	17,326	175	5	1,833	115	4.06s

As we can directly observe from the tables, the matrices contain less symmetry in 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 (less faces are evaluated) for the same dimension.

The top-down traversal algorithms do not profit a lot from the increasing density. They have to check more the PSD of H_k with higher density as can be observed from Table 8. Moreover, they make use of monotonicity in order not to check the complete face graph. Checking the faces in index order CDK from top to down, leads to less efficiency in concluding on the same monotonicity than the CDown traversal. However, as we can observe, that comes with the cost of the list management, such that in the end for the two variants the total computing time is basically the same.

Table 9: Results of face traversal algorithm CUp on varying density instances 7,..,12. Number of..Chol: calls to Cholesky to check H_k is PSD; KKT: H_k is PSD, finding KKT points; $\#f$: interior optimum evaluated; #Ap: matrix A_k is completely positive

n	ℓ	Chol	KKT	$\#f$	#Ap	T
11	6	74	27	2	97	0.14s
11	8	222	94	9	167	0.17s
11	10	761	234	14	146	0.32s
16	6	132	47	4	362	0.39s
16	9	1,458	209	8	1,528	1.51s
16	15	23,546	543	9	9,438	9.00s

For the bottom-up traversal CUp, 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. We can observe this in Table 9. In computing time, the bottom-up approach only gets slower than the top-down traversal variants for the most dense and largest instance. However, we have to admit that for these instances the optimum is at a relatively low level, which is bad for the CDown traversal. Moreover, the computing time depends a lot on how well the management of the lists is organized.

5 Conclusions

This paper derives several properties on the convexity of the standard quadratic problem over faces and subsets of the unit simplex and on monotonicity. It illustrates that monotonicity considerations help in copositivity detection algorithms as well for B&B type algorithms and face-based algorithms which traverse the face graph of the unit simplex top-down. We found that randomly generated instances provide far more monotonicity than maximum clique based instances with a lot of symmetry.

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.

The implementations of the algorithms used for the illustration are based on easily available matrix subroutines, but do not exploit a lot the management of lists. From a computer science perspective this can be improved a lot in order to come to efficient algorithms which can be compared in computing time to earlier published results. Moreover, we are looking into the parallelization of the search algorithm.

References

[1] Challenge Problems: Independent Sets in Graphs. `https://oeis.org/A265032/a265032.html`https://oeis.org/A265032/a265032.html.

[2] Second DIMACS implementation challenge. `http://archive.dimacs.rutgers.edu/Challenges/`.

[3] I. M. Bomze and E. De Klerk. Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *Journal of Global Optimization*, 24(2):163–185, 2002.

[4] S. Bundfuss and M. Dür. Algorithmic copositivity detection by simplicial partition. *Linear Algebra and its Applications*, 428(7):1511–1523, 2008.

[5] S. Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming*, 120(2):479–495, 2009.

[6] J. Gondzio and E. A. Yildirim. 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, 2018.

[7] M. Hall and M. Newman. Copositive and completely positive quadratic forms. *Mathematical Proceedings of the Cambridge Philosophical Society*, 59(2):329–339, 1963.

[8] E.M.T. Hendrix, J M.G. Salmerón, and L.G. Casado. On function monotonicity in simplicial branch and bound. *AIP Conference Proceedings*, 2070(1):020007, 2019.

[9] W. Kaplan. A test for copositive matrices. *Linear Algebra and its Applications*, 313(1-3):203–206, 2000.

[10] G. Liuzzi, M. Locatelli, and V. Piccialli. A new branch-and-bound algorithm for standard quadratic programming problems. *Optimization Methods and Software*, 34(1):79–97, 2019.

[11] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39(2):117–129, 1987.

[12] I. Nowak. A new semidefinite programming bound for indefinite quadratic forms over a simplex. *Journal of Global Optimization*, 14(4):357–364, Jun 1999.

[13] J. Povh and F. Rendl. A copositive programming approach to graph partitioning. *SIAM Journal on Optimization*, 18(1):223–241, 2007.

[14] J. M. G. Salmerón. *High Performance Computing to solve Global Optimization Problems.* PhD thesis, Universidad de Almería.

[15] J.M.G. Salmerón, L.G. Casado, and E.M.T. Hendrix. 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*, pages 77–82. Universidad de Zaragoza, July 2018.

[16] A. Scozzari and F. Tardella. A clique algorithm for standard quadratic programming. *Discrete Applied Mathematics*, 156(13):2439 – 2448, 2008.

[17] H. Väliaho. Criteria for copositive matrices. *Linear Algebra and its Applications*, 81:19–34, 1986.

[18] S. Yang and X. Li. Algorithms for determining the copositivity of a given symmetric matrix. *Linear Algebra and its Applications*, 430(2-3):609–618, 2009.

[19] J. Žilinskas and M. Dür. Depth-first simplicial partition for copositivity detection, with an application to maxclique. *Optimization Methods and Software*, 26(3):499–510, 2011.

A Matrix instances from literature

Example A.1. $n = 3$ from [4]:

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

Example A.2. Horn matrix, $n = 5$ from [7].

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

Example A.3. 1tc.8.clique, $n = 8$ from `oeis.org/A265032/a265032.html`,`oeis.org/A265032/a265032.html`

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

$A = (t-1)\mathbf{1}\mathbf{1}^T - tAdj$ is copositive for $t = 4$.

Example A.4. Brock14, $n = 14$ from
`http://archive.dimacs.rutgers.edu/pub/challenge/graph/The Second DIMACS Implementation Challenge`. Generated with *grahgen -g14 -c5 -p0.5 -d0 -s0*.

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

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

Example A.5. Johnson6-2-4, $n = 15$ from
`http://archive.dimacs.rutgers.edu/pub/challenge/graph/The Second DIMACS Implementation Challenge`.

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

$A = (t-1)\mathbf{1}\mathbf{1}^T - tAdj$ is copositive for $t = 3$.

Example A.6. 1tc.16.clique, $n = 16$ from `https://oeis.org/A265032/a265032.html`.

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

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

B Variable density matrices

Example B.1. Ivo-n10-dens0.75-dvert20, $n = 11$ from [12].

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

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

Example B.2. Ivo-n10-dens0.95-dvert20, $n = 11$ from [12].

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

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

Example B.3. Ivo-n10-dens1-dvert20, $n = 11$ from [12].

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

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

Example B.4. Ivo-n15-dens0.75-dvert20, $n = 16$ from [12]. $A=$

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

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

Example B.5. Ivo-n15-dens0.95-dvert20, $n = 16$ from [12]. $A=$

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

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

Example B.6. Ivo-n15-dens1-dvert20, $n = 16$ from [12]. $A=$

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

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