

ON COMPUTING THE COPOSITIVE MINIMUM AND ITS REPRESENTATIVES

ALEXANDER OERTEL AND ACHILL SCHÜRMANN

ABSTRACT. Computing the copositive minimum of a strictly copositive quadratic form is a natural generalization of computing the arithmetical minimum of a positive definite one. In this paper we show that this generalized problem is NP-complete. Moreover, we describe a practical method to calculate all shortest vectors using the LDLT-decomposition in a big class of special cases. Our numerical tests show that our method performs significantly better than previous approaches.

1. INTRODUCTION

A classical problem in the geometry of numbers with multiple applications in numerous contexts is the calculation of shortest vectors of positive definite quadratic forms (or, equivalently, shortest vectors of a lattice). More precisely, using matrix notation, the problem is for a positive definite matrix Q to compute the *arithmetical minimum*

$$\min Q = \min_{z \in \mathbb{Z}^n \setminus \{0\}} Q[z]$$

and the (integral) vectors

$$\text{Min } Q = \{z \in \mathbb{Z}^n : Q[z] = \min Q\}$$

attaining it. Here we use the notation $Q[z] = z^T Q z$ for column vectors $z \in \mathbb{R}^n$.

We denote by $\mathcal{S}^n$ the space of real symmetric $n \times n$ matrices. A matrix $Q \in \mathcal{S}^n$ is said to be *copositive*, if $Q[x] \geq 0$ for all $x \in \mathbb{R}_{\geq 0}^n$, and *strictly copositive*, if it is copositive and $Q[x] = 0$ for $x \in \mathbb{R}_{\geq 0}^n$ only if $x = 0$. For a matrix $Q \in \mathcal{S}^n$ the *copositive minimum* is a natural generalization of the arithmetical minimum and defined as

$$\min_{\mathcal{COP}} Q := \min \{Q[z] : z \in \mathbb{Z}_{\geq 0}^n \setminus \{0\}\}.$$

The vectors attaining it are denoted by

$$\text{Min}_{\mathcal{COP}} Q := \{z \in \mathbb{Z}_{\geq 0}^n : Q[z] = \min_{\mathcal{COP}} Q\}.$$

We call the task of computing $\text{Min}_{\mathcal{COP}} Q$, i.e. enumerating all nonnegative shortest vectors, the *MinCOP problem*. Note that every calculation of a classical arithmetical minimum and the corresponding shortest vectors can be realized as a copositive minimum and representatives calculation [12, Theorem 6.1].

These notations have been introduced in [18], in the context of computing certificates for complete copositivity of a given matrix, which we summarize briefly in Section 3.2. The main bottleneck there is the computation of the copositive minimum. The latter is deeply connected to the copositivity of Q .

Lemma 1.1. *For a matrix $Q \in \mathcal{S}^n$ we have*

Date: September 27, 2025.

2020 Mathematics Subject Classification. 11H50, 11Y16, 15A23, 90C20.

Key words and phrases. Copositive Minimum, Complexity, LDLT-decomposition.

- (i) $\min_{\text{COP}} Q > 0$ if and only if Q is strictly copositive,
- (ii) $\min_{\text{COP}} Q = 0$ if and only if Q is copositive, but not strictly so, or
- (iii) $\min_{\text{COP}} Q = -\infty$ if and only if Q is not copositive.

Proof. Part (i) follows from [17, Lemma 2.3] and [18, Lemma 2.2]. If $z \in \mathbb{Z}_{\geq 0}^n$ with $Q[z] < 0$, then $Q[\alpha z] = \alpha^2 Q[z] \rightarrow -\infty$ for $\alpha \rightarrow \infty$ with $\alpha \in \mathbb{Z}$, thus (iii). The remaining part (ii) follows from exhaustion of all cases. $\square$

Lemma 2.2 of [18] also yields that the copositive minimum of a strictly copositive matrix is attained, but attained only by finitely many vectors. Strictly copositive matrices are thus a natural domain for the copositive minimum, though for mostly technical reasons we allow general symmetric matrices as an input.

We give a short example which we pick up repeatedly.

Example 1.2. The matrix

$$Q = \begin{pmatrix} 3 & -1 & 3 \\ -1 & 2 & -1 \\ 3 & -1 & 2 \end{pmatrix}$$

is strictly copositive, since

$$Q[x] = 2x_1^2 + (x_1 - x_2)^2 + 6x_1x_3 + (x_2 - x_3)^2 + x_3^2.$$

A quick mod 2 argument shows that $Q[z] > 1$ for $z \in \mathbb{Z}_{\geq 0}^n \setminus \{0\}$. Then we have $\min_{\text{COP}} Q = 2$ and

$$\text{Min}_{\text{COP}} Q = \left\{ \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

are the vectors attaining the copositive minimum.

The computational ansatz for MinCOP utilized in [18] is based on a simplicial cone partition of the nonnegative orthant such that $v^T Q w > 0$ for each pair v, w of extreme ray generators of a common simplicial cone in $\mathbb{R}_{\geq 0}^n$. This simplicial subdivision of $\mathbb{R}_{\geq 0}^n$ was first introduced by Dür and Bundfuss [10] in the context of testing for copositivity. Intuitively, the condition translates to the matrices being nonnegative with respect to each cone separately and in each cone a Fincke-Pohst-type algorithm (see Section 3.3) is performed. This elegant approach has the drawback that some matrices require a seemingly large number of cones. For this reason there is much performance improvement to be gained by finding alternative ways to calculate the copositive minimum.

Copositivity itself has numerous applications and connections to other parts of mathematics, cf. [32, Section 2.1]. Arguably the most important one is copositive optimization. Many NP-complete problems have reformulations in this context and the already mentioned certificates are a central building block here for verifiable computer proofs, which are becoming more and more important. For details on copositive optimization we refer to the surveys [16] and [6].

Our paper is structured as follows. We start in Section 2 by establishing that (a decision variant of) the copositive minimum is NP-complete. In Section 3 the necessary background information about symmetric matrix cones, the origins of the copositive minimum, the Fincke-Pohst algorithm for the shortest vector problem, and the LDLT-decomposition is then reviewed. Next, in Section 4 the LDLT-decomposition is combined with the Fincke-Pohst algorithm to attack MinCOP. In particular, we discuss how the decomposition should be used and introduce the concept of *difficult coordinates*, which is central for our new algorithm. In Section 5 we describe how our method may solve multiple special cases, such as the case of one

difficult coordinate, semidefinite matrices, and when the matrix is decomposable into a positive semidefinite and nonnegative part. Our method is then analyzed numerically in Section 6. Finally, several open questions and possible directions for further research are discussed in Section 7.

2. COMPLEXITY OF THE COPOSITIVE MINIMUM

In this section we prove that a decision variant of the copositive minimum is NP-complete by reduction of the *Subset Sum Problem* (see [20]) to it. The analogue *shortest vector problem* (SVP) is known to be NP-hard under randomized reductions [2], while NP-hardness under deterministic reductions is still an open problem. Its complexity in general is still an active research topic (cf. e.g. the recent survey [4]). We remark that the rather small symmetry group of the copositive cone (see Section 3.1) induces a lack of (a comparable) reduction theory for copositive matrices. This hints that calculating the copositive minimum has a much harder practical difficulty, which we also observe in practice.

Definition 2.1 (Copositive Minimum – Decision Variant). Given a strictly copositive matrix Q with integer entries and a positive number λ , decide if there is $z \in \mathbb{Z}_{\geq 0}^n \setminus \{0\}$ with $Q[z] \leq \lambda$.

Definition 2.2 (Subset Sum Problem). Given a vector a of dimension n , consisting of positive integers and a target positive integer s , decide if there is $x \in \{0, 1\}^n$ such that $a^T x = s$.

The proof of the NP-completeness of the Subset Sum Problem given in [25, Thm. 8.23] yields that this variant of the subset sum problem (with only positive integers) is NP-complete.

Theorem 2.3. *The decision variant of the copositive minimum is NP-complete.*

Proof. Note that a certificate for $\min_{\mathcal{COP}} Q \leq \lambda$ would be a suitable vector $z \in \mathbb{Z}_{\geq 0}^n \setminus \{0\}$: We can calculate $Q[z]$ in polynomial time and check the inequality in polynomial time as well.

Let $(a; s)$ be an instance of the subset sum problem. Define

$$Q := n \begin{pmatrix} aa^T & -sa \\ -sa^T & s^2 \end{pmatrix} + \begin{pmatrix} 2I_n & -\mathbf{1}_n \\ -\mathbf{1}_n^T & n \end{pmatrix},$$

with I_n being the identity matrix of dimension n and $\mathbf{1}_n$ the all-ones column vector of dimension n . The matrix Q can be calculated in polynomial time in the input length.

A later part of the proof will yield that $Q[y] \geq n$ for all $y \in \mathbb{Z}_{\geq 0}^n \setminus \{0\}$ from which we deduce $Q \in \text{int } \mathcal{COP}^{n+1}$ by Theorem 1.1. Alternatively, this follows from

$$Q = H^T H + \begin{pmatrix} I_n & \\ & 0 \end{pmatrix}$$

for

$$H = \begin{pmatrix} \sqrt{n}a^T & -\sqrt{n}s \\ -I_n & \mathbf{1}_n \end{pmatrix}.$$

We write $y = \begin{pmatrix} x \\ t \end{pmatrix}$ with $x \in \mathbb{R}^n$ and $t \in \mathbb{R}$. Then we have

$$Q[y] = n((a^T x)^2 - 2sta^T x + t^2 s^2) + 2x^T x - 2t\mathbf{1}_n^T x + t^2 n.$$

We also abbreviate

$$h_t(x) := (a^T x)^2 - 2sta^T x + t^2 s^2 = (a^T x - ts)^2 \geq 0.$$

Now we show that the subset sum problem has a solution $x \in \{0, 1\}^n$ if and only if the copositive minimum of Q is less than or equal to n .

If $x \in \{0, 1\}^n$ is a solution to the subset sum problem, i.e. $a^T x = s$, then

$$\begin{aligned} Q \left[\begin{pmatrix} x \\ 1 \end{pmatrix} \right] &= n(s^2 - 2s^2 + s^2) + 2x^T x - 2\mathbf{1}_n^T x + n \\ &= 2 \sum_{i=1}^n (x_i^2 - x_i) + n = n, \end{aligned}$$

since $x_i^2 - x_i = 0$ for $x_i \in \{0, 1\}$.

In the other direction, we show that if the subset problem has no solution, then the copositive minimum of Q is strictly greater than n (and is in general greater than n). To that matter, we consider three cases for the last coordinate t of $y \in \mathbb{Z}_{\geq 0}^{n+1} \setminus \{0\}$.

Case 1: $t = y_{n+1} \geq 2$. Since $x_i^2 - tx_i \geq -\frac{t^2}{4}$, we can bound

$$\begin{aligned} Q[y] &= nh_t(x) + 2x^T x - 2t\mathbf{1}_n^T x + t^2 n \\ &= nh_t(x) + 2 \sum_{i=1}^n (x_i^2 - tx_i) + t^2 n \\ &\geq nh_t(x) + 2n \frac{-t^2}{4} + t^2 n \\ &= nh_t(x) + \frac{t^2}{2} n > n. \end{aligned}$$

Case 2: $t = y_{n+1} = 1$. We have

$$Q[y] = nh_1(x) + 2 \sum_{i=1}^n (x_i^2 - x_i) + n.$$

If $x \in \{0, 1\}^n$, then $h_1(x) > 0$, because there is no solution for the subset sum problem. If $x \notin \{0, 1\}^n$, then $\sum_{i=1}^n (x_i^2 - x_i) > 0$. So in total, $Q[y] > n$.

Case 3: $t = y_{n+1} = 0$. Then $x \neq 0$ and thus $a^T x \geq 1$. In particular,

$$Q[y] = n(a^T x)^2 + 2x^T x > n. \quad \square$$

3. PRELIMINARIES AND NOTATIONS FOR OUR ALGORITHMS

3.1. Symmetric Matrix Cones. Throughout the paper we equip the space $\mathcal{S}^n$ with the standard trace scalar product $\langle \cdot, \cdot \rangle$. In this section we introduce different convex cones contained in $\mathcal{S}^n$ and refer to [32] for a thorough treatment. Arguably the most important convex cone contained in $\mathcal{S}^n$ is the *positive semidefinite cone* $\mathcal{S}_{\succcurlyeq 0}^n$ consisting of the positive semidefinite matrices of dimension n . The cone is generated by the rank 1 matrices, i.e.

$$\mathcal{S}_{\succcurlyeq 0}^n = \text{cone} \{xx^T : x \in \mathbb{R}^n\}$$

and is selfdual, meaning that

$$(\mathcal{S}_{\succcurlyeq 0}^n)^D = \{Q \in \mathcal{S}^n : \langle Q, X \rangle \geq 0 \text{ for all } X \in \mathcal{S}_{\succcurlyeq 0}^n\} = \mathcal{S}_{\succcurlyeq 0}^n.$$

Another cone that has been investigated for a long time is the *completely positive cone*

$$\mathcal{CP}^n = \text{cone} \{xx^T : x \in \mathbb{R}_{\geq 0}^n\}.$$

See [32, Section 3.1] for a short survey on the numerous applications of the cone. Its dual cone is the *copositive cone*

$$\begin{aligned} \mathcal{COP}^n &= \{Q \in \mathcal{S}^n : Q[x] \geq 0 \text{ for all } x \in \mathbb{R}_{\geq 0}^n\} \\ &= \{Q \in \mathcal{S}^n : \langle Q, X \rangle \geq 0 \text{ for all } X \in \mathcal{CP}^n\}. \end{aligned}$$

Its interior $\text{int } \mathcal{COP}^n$ is precisely made up of the strictly copositive matrices [32, Theorem 2.25]. As seen in Theorem 1.1, a matrix $Q \in \mathcal{COP}^n$ is strictly copositive if its copositive minimum is strictly greater than zero. The automorphisms Φ preserving $\mathcal{COP}^n$ are of the form

$$\Phi(Q) = (PD)^T QPD$$

with a permutation matrix P and a diagonal matrix D with only positive entries [22, 33]. Matrices of the form PD are called *monomial*.

The copositive cone contains $\mathcal{S}_{\geq 0}^n$ and also the nonnegative cone $\mathcal{N}^n$ of all matrices with nonnegative entries. Therefore, we also have

$$\mathcal{SPN}^n = \mathcal{S}_{\geq 0}^n + \mathcal{N}^n \subseteq \mathcal{COP}^n.$$

Up to $n = 4$ the cones $\mathcal{SPN}^n$ and $\mathcal{COP}^n$ coincide [14]. For $n \geq 5$ the matrices in $\mathcal{COP}^n \setminus \mathcal{SPN}^n$ are called *exceptional* and a standard example is the so called *Horn matrix* (see [14] for its first appearance). Some information regarding the relative size of $\mathcal{SPN}^n$ in $\mathcal{COP}^n$ can be found in [26, 37].

The dual cone of $\mathcal{SPN}^n$ is the cone of the doubly nonnegative matrices

$$\mathcal{DNN} = \mathcal{S}_{\geq 0}^n \cap \mathcal{N}^n,$$

which we list here for completeness only.

3.2. Certificates for Complete Positivity. In this section we sketch how the copositive minimum originated in [18] from the study of $\mathcal{CP}$ -factorizations. In doing so we also introduce the *perfect copositive matrices*, which are an important subclass of copositive matrices to test our algorithms against. This section is not strictly necessary to understand the algorithms themselves.

The usual definition for a matrix to be completely positive, i.e. $Q \in \mathcal{CP}^n$, is that there exists a nonnegative matrix B (that is, B has nonnegative entries) such that $Q = B^T B$. An equivalent condition is the existence of a $\mathcal{CP}$ -factorization

$$Q = \sum_{i=1}^m \alpha_i x_i x_i^T \text{ with } m \in \mathbb{N}, \alpha_i \geq 0, x_i \in \mathbb{R}_{\geq 0}^n \text{ for } i = 1, \dots, m.$$

Such a $\mathcal{CP}$ -factorization is called *rational*, if the coefficients α_i and the (components of the) vectors x_i are rational for all i . It has been shown that deciding whether $Q \in \mathcal{CP}^n$ is NP-hard [15].

Nonetheless, in some important contexts, such as computer proofs based on results from copositive optimization, it is essential to find certificates for (non-) complete positivity (cf. [5]).

In [18] a simplex type algorithm for rational $\mathcal{CP}$ -factorizations is described and in this context the notion of the copositive minimum is introduced. It is used to define the *copositive Ryshkov polyhedron*

$$\mathcal{R} := \{Q \in \mathcal{S}^n : \min_{\mathcal{COP}} Q \geq 1\},$$

which is a so called locally finite polyhedron (see [31] for details about locally finite polyhedra). It is on the edge graph of this set where the simplex type algorithm is performed.

In analogy to the classical theory of perfect quadratic forms, we call copositive matrices *perfect copositive* if they are uniquely determined by their copositive minimum and its representatives. In particular, the vertices of $\mathcal{R}$ are precisely the perfect copositive matrices with copositive minimum equal to 1.

The algorithm now walks along a path of edges in $\mathcal{R}$, visiting neighboring perfect copositive matrices $B^{(k)}$ (vertices of $\mathcal{R}$), until Q is either contained in the (inner) normal cone

$$\mathcal{V}(B^{(k)}) = \text{cone} \left\{ vv^T : v \in \text{Min}_{\mathcal{COP}} B^{(k)} \right\}$$

(dubbed *Voronoi cone*) or a perfect copositive matrix $B^{(k)}$ acts as a separating witness, i.e. is a certificate for non-complete positivity via $\langle Q, B^{(k)} \rangle < 0$.

Finding the next contiguous perfect copositive matrix in $\mathcal{R}$ requires a large number of copositive minimum calculations: For a given perfect copositive matrix Q one drops (at least) one minimal vector, obtains a ray on which the next perfect copositive matrix has to be and performs a search on the ray. The copositive minimum for each matrix on the ray then yields the information whether one is already beyond the next perfect copositive matrix or not.

Note that there are some other algorithms for checking for complete positivity. Some recent ones are by Lai and Yoshise [27] and Badenbroek and de Klerk [3]. In these references one can also find further references to other approaches. See also [18] for a division into numerical and theoretical algorithms. For more information on perfect copositive matrices we refer to [12], where first theoretical investigations into these matrices are given.

3.3. Shortest Vector Problem and Fincke-Pohst-Algorithm. Given a lattice Λ (in the sense of an additive and discrete subgroup of an Euclidean vector space) with a fixed basis we denote its positive definite Gram matrix by Q . The shortest vector problem now asks for the minimal length of a non-zero lattice vector. We refer to the survey [40] for information about the problem, in particular how it is solved. See also [29]. Since we are interested in all shortest vectors for MinCOP, we describe in this section the core idea behind enumeration algorithms.

The usual approach to enumerate all shortest vectors is to shift the viewpoint towards the positive definite quadratic form Q , where the lattice minimum translates to the (square root of the) arithmetical minimum and the minimal vectors of Q correspond to coefficients of linear combinations giving short lattice vectors.

Next we describe the rough idea of the Fincke-Pohst algorithm [19], which is the usual way to compute $\text{Min } Q$ and also the fundamental idea we build upon in this paper.

The goal of the Fincke-Pohst algorithm for a given positive definite matrix Q and a constant $\lambda > 0$ is to enumerate all integral points inside the ellipsoid

$$\{x \in \mathbb{R}^n : Q[x] \leq \lambda\}.$$

The tool to do so is the *Lagrange expansion*

$$Q[x] = \sum_{i=1}^n A_i \left(x_i + \sum_{j=i+1}^n u_{ij} x_j \right)^2$$

of $Q[x]$ with the *outer coefficients* $A_i > 0$ and the *inner coefficients* $u_{ij} \in \mathbb{R}$.

This expansion is equivalent to the Cholesky-decomposition of Q . If Q is given as the Gram matrix of some basis $B = (b_1, \dots, b_n) \in \mathbb{R}^{n \times n}$, then there is also a simple geometric interpretation involving the Gram-Schmidt-orthogonalization $\bar{B} = (\bar{b}_1, \dots, \bar{b}_n)$. If B is given as $B = \bar{B}U$, then

$$Q[x] = x^T B^T B x = (Ux)^T \bar{B}^T \bar{B} (Ux).$$

In particular, the outer coefficients A_i are the squared lengths of the $\bar{b}_i$. The inner coefficients are given as

$$u_{ij} = \frac{\langle b_i, \bar{b}_j \rangle}{\langle \bar{b}_j, \bar{b}_j \rangle}$$

and describe the “relative length” of the projection of b_i onto the line spanned by $\bar{b}_j$.

This special structure allows for a recursive algorithm, in which each coordinate is subsequently bounded. More precisely, if $Q[x] \leq \lambda$ is wanted, then the inequality

$Q[x] \geq A_n x_n^2$ implies

$$-\sqrt{\frac{\lambda}{A_n}} \leq x_n \leq \sqrt{\frac{\lambda}{A_n}}.$$

Since x is to be integral, there are only finitely many values x_n can take. If we fix x_n to any such value, the inequality

$$\lambda - A_n x_n^2 \geq A_{n-1} (x_{n-1} + u_{n-1,n} x_n)^2$$

yields a range in which x_{n-1} has to be in.

In the same way one can obtain bounds for each coordinate x_i if $x_{i+1}, \dots, x_n$ are fixed. This yields a search tree overall, whose leaves (at the lowest level) correspond to the desired integral vectors. When only the shortest vectors are of interest, one can lower the constant λ with each new leaf corresponding to a shorter vector, effectively pruning the search tree.

3.4. LDLT-Decomposition. In this subsection we describe for a given symmetric matrix its *LDLT-decomposition*, which is a generalization of the Cholesky-decomposition and also the central tool in the remainder of this paper. For self-containment we also describe how to calculate the decomposition as far as we need it.

Definition 3.1. Let $Q \in \mathcal{S}^n$. An *LDLT-decomposition* of Q is a decomposition

$$Q = LDL^T,$$

with a lower triangular matrix L with unit diagonal and a block diagonal matrix D , where the blocks are at most of size 2.

A classical application of the decomposition is the calculation of the inertia of a symmetric matrix and solving linear symmetric systems [8, 9]. For a treatment other than the just mentioned references, see e.g. [21, Chapter 4]. It is a fact that for a symmetric matrix $Q \in \mathcal{S}^n$ there exists a permutation matrix P , such that the symmetric permutation $P^T Q P$ of Q has an LDLT-decomposition. The use of 2×2 blocks is necessary both for existence and numerical stability, with the standard example for a matrix without an LDLT-decomposition with a proper diagonal matrix being

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

See also Section 4.3. Positive semidefinite matrices, however, always admit an LDLT-decomposition with no permutation needed (see e.g. [35, Theorem 2.2.6]).

We omit the details here, because for our application only LDLT-decompositions with a proper diagonal matrix D are (currently) interesting to us. With such decompositions, we obtain a *Lagrange-type expansion*

$$(1) \quad Q[x] = \sum_{i=1}^n D_{ii} \left(x_i + \sum_{j=i+1}^n u_{ij} x_j \right)^2,$$

with $(u_{ij}) := U := L^T$, with the important difference that the *outer* coefficients D_{ii} may be zero or negative. We can tolerate the drawbacks associated with disallowing 2×2 blocks in the diagonal matrix, because in Section 4.3 we describe a strategy that deals with nonexistence. All (relevant) calculations can be made in exact (rational) arithmetic so that numerical instability is not an issue either.

Next we describe next how to perform an LDLT-step (with proper diagonal matrix). We partition the matrix Q into

$$Q = \begin{pmatrix} E & C^T \\ C & B \end{pmatrix},$$

with $E = (A_{11})$, $C \in \mathbb{R}^{(n-1) \times 1}$, and $B \in \mathbb{R}^{(n-1) \times (n-1)}$. We then have $Q = LAL^T$ with

$$L = \begin{pmatrix} 1 & 0 \\ CE^{-1} & I_{n-1} \end{pmatrix} \text{ and } A = \begin{pmatrix} E & 0 \\ 0 & B - CE^{-1}C^T \end{pmatrix},$$

provided that $E \neq 0$. We note that this step is compatible with permutations, which fix the first coordinate: If we have

$$P = \begin{pmatrix} 1 & \\ & \tilde{P} \end{pmatrix}$$

with $\tilde{P}$ being a $(n-1) \times (n-1)$ -permutation matrix, we have that applying the step to P^TQP yields $\tilde{P}^T(B - CE^{-1}C^T)\tilde{P}$ as the remainder.

This one step is equivalent to *completing the square* for the first coordinate, for example

$$(2) \quad ax_1^2 + 2bx_1x_2 + cx_2^2 = a \left(x_1 + \frac{b}{a}x_2 \right)^2 + \left(c - \frac{b^2}{a} \right) x_2^2.$$

One can then iterate the step with the remaining matrix $B - CE^{-1}C^T$ to obtain an LDLT-decomposition of Q .

Example 3.2. The matrix Q from Theorem 1.2 has the LDLT-decomposition

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 & 0 & 0 \\ 0 & \frac{5}{3} & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}^T.$$

4. APPLYING LDLT TO THE MINCOP PROBLEM

4.1. Difficult Coordinates. Let in this section $Q \in \mathcal{S}^n$ and let $Q^{(k)} \in \mathcal{S}^k$ be the leading principal submatrix of size k . Knowing an LDLT-decomposition of Q means knowing it for $Q^{(k)}$ as well.

Lemma 4.1. *Suppose $Q = LDL^T$ is an LDLT-decomposition of Q . Then for $1 \leq k \leq n$ and $x = (x_1 \dots x_k)^T$ the Lagrange expansion*

$$Q^{(k)}[x] = \sum_{i=1}^k D_{ii} \left(x_i + \sum_{j=i+1}^k u_{ij}x_j \right)^2,$$

with $U = (u_{ij}) = L^T$, holds.

Proof. By definition we have

$$Q[(x_1 \dots x_k 0 \dots 0)^T] = \sum_{i,j=0}^k q_{ij}x_ix_j = Q^{(k)}[x]$$

and by the Lagrange expansion Eq. (1) of Q we have

$$Q[(x_1 \dots x_k 0 \dots 0)^T] = \sum_{i=1}^k D_{ii} \left(x_i + \sum_{j=i+1}^k u_{ij}x_j \right)^2. \quad \square$$

Corollary 4.2. *Let $1 \leq k \leq n$ and assume $D_{ii} > 0$ for $1 \leq i \leq k$. Then $Q^{(k)}$ is positive definite.*

Our idea is now to replace Q by P^TQP for a permutation matrix P , such that

$$k := \operatorname{argmin}_i \{D_{ii} \leq 0\}$$

is as large as possible. This is useful, since if $x_k, \dots, x_n$ are assumed to be fixed, we can obtain bounds on $x_1, \dots, x_{k-1}$ implied by $Q[x] \leq \lambda$ from the expansion

in the same way as in the positive definite case. We provide an example, to make clear that the “easy” coordinates $x_1, \dots, x_{k-1}$ can be handled as usual.

Example 4.3. We continue with the matrix Q from Theorem 1.2 and Theorem 3.2. Let us search for vectors $x \in \mathbb{Z}_{\geq 0}^n$ satisfying $Q[x] \leq 2$. Theorem 1.2 showed that 2 is the copositive minimum of Q . Setting $x_3 = 0$ to start, we find

$$2 \geq 3 \left(x_1 - \frac{1}{3}x_2 \right)^2 + \frac{5}{3}x_2^2.$$

This is the exact setting discussed in Section 3.3, so we may continue as usual, finding the vector $(0 \ 1 \ 0)^T$. Setting $x_3 = 1$ on the other hand, we find

$$3 \geq 3 \left(x_1 - \frac{1}{3}x_2 + 1 \right)^2 + \frac{5}{3}x_2^2,$$

which allows us to continue in the usual way as well, leading to the vectors $(0 \ 0 \ 1)^T$ and $(0 \ 1 \ 1)^T$. In Theorem 5.2 we find that these are all shortest vectors by showing that x_3 may only take the values 0 and 1.

Definition 4.4. Let $Q = LDL^T$ be an LDLT-decomposition of $Q \in \mathcal{S}^n$ and let $k = \operatorname{argmin}_i \{D_{ii} \leq 0\}$. Then the coordinates $x_k, \dots, x_n$ are called *difficult coordinates*.

A reformulation of the previously stated main idea is to minimize the number of difficult coordinates. By Sylvester’s law of inertia the number of negative (zero) diagonal entries of D is exactly the number of negative (zero) eigenvalues. This implies that the number of negative and zero eigenvalues is a lower bound for the number of difficult coordinates. Theorem 4.1 implies now the following corollary.

Corollary 4.5. *The minimal number of difficult coordinates (of any symmetric permutation of Q) is the same as the minimal number of rows and columns that are necessary to delete to obtain a positive definite matrix.*

To conclude, only bounds for the difficult coordinates are necessary. These bounds do not need to be particularly good, since the bounds of the “easy” coordinates can mitigate the bad bounds. A rather simple idea is to “box” the difficult coordinates in: We calculate for a difficult coordinate x_k the value of (or find a bound for)

$$(3) \quad \max\{x_k : x \in \mathbb{R}_{\geq 0}^n, Q[x] \leq \lambda\}.$$

This bound ignores the other coordinates, so it might not be very good. Even worse, the optimization problem is not convex and very difficult to solve in general. In Section 5 we give some cases in which this is, however, easy to solve.

Geometrically speaking, we intersect the set $\{x \in \mathbb{R}^n : Q[x] \leq \lambda\}$ consecutively with suitable affine subspaces such that the intersection is a (lower-dimensional) ellipsoid, on which we can apply the Fincke-Pohst algorithm again. See Fig. 1.

4.2. Pivot Strategy. In this subsection we explain how to accomplish the goal formulated in the last subsection heuristically. Finding the permutation of a suitable symmetric permutation $P^T Q P$ can be done while performing the LDLT-algorithm by choosing a *pivot element* among the diagonal elements. With respect to this pivot element the LDLT-step is performed. In practice, this means applying a transposition so that the chosen diagonal element becomes the new first diagonal element. Then a regular LDLT-step can be performed as described in Section 3.4. With the viewpoint of completing the square, as in Eq. (2), the pivot element decides towards which variable we complete the square.

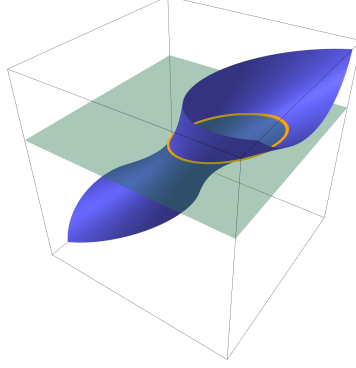


FIGURE 1. A surface plot of $Q[x] = 2$ with Q from Theorem 1.2 with the plane $x_3 = 1$.

Our heuristic to obtain the permutation is split into two phases. In the first, we aim to, figuratively speaking, push negative outer coefficients as far right as possible. Towards this, we always choose the largest positive diagonal element as the pivot.

In the second phase we reorder the earlier “easy” coordinates to improve the performance of the algorithm. This procedure has also already been proposed by Fincke and Pohst (cf. also [11, Sec. 2.7.3]): A reordering is made, with the idea being that a segment $(x_k, \dots, x_n)$ should be extendable to a vector $(x_1, \dots, x_n)$, since we have a certain amount of vectors in the set

$$\{z \in \mathbb{Z}_{\geq 0}^n : Q[z] \leq \lambda\},$$

which need to be traversed in the search tree. Better extendability means traversing fewer “useless” branches of the tree. In our case the reordering translates to having the (positive) diagonal elements of D be increasing in value.

One way to accomplish this is described next. We take the LDLT-decomposition

$$P_1^T Q P_1 = L D L^T \text{ and } (P_1^T Q P_1)[x] = \sum_{i=1}^n D_{ii} \left(x_i + \sum_{j=i+1}^n u_{ij} x_j \right)^2$$

of (the symmetric permutation of) Q . Then we subtract the summands corresponding to the difficult coordinates. More precisely, define

$$\begin{aligned} R[x] &:= (P_1^T Q P_1)[x] - \sum_{i=k+1}^n D_{ii} \left(x_i + \sum_{j=i+1}^n u_{ij} x_j \right)^2 \\ &= \sum_{i=1}^k D_{ii} \left(x_i + \sum_{j=i+1}^n u_{ij} x_j \right)^2, \end{aligned}$$

where $k+1$ is the smallest difficult coordinate. The matrix R is given by $L \tilde{D} L^T$, where $\tilde{D}$ arises from D by setting

$$\tilde{D}_{k+1 \ k+1} = \dots = \tilde{D}_{nn} = 0.$$

Notice that by its LDLT-decomposition the matrix R is positive semidefinite (and has a kernel of dimension $n - k$). Furthermore, R is of the form

$$\begin{pmatrix} \tilde{R} & \\ & 0 \end{pmatrix}$$

with $\tilde{R} \in \mathcal{S}^k$. Now we apply the LDLT-algorithm to R , where we use the smallest non-zero diagonal element as the pivot element, while we fix the last k coordinates. This means that we skip the last $n - k$ diagonal elements in the search for the pivot element. Or, in other terms, we calculate an LDLT-decomposition of $\tilde{R} \in \mathcal{S}^k$ taking the smallest non-zero diagonal as the pivot element. This leads to

$$A := P_2^T R P_2 = L_2 D_2 L_2^T.$$

Note that the last $n - k$ diagonal elements of D_2 are 0 by the choice of P_2 and R .

Overall, it follows that

$$Q[P_1 P_2 x] = A[x] + \sum_{i=k+1}^n D_{ii} \left(x_i + \sum_{j=i+1}^n u_{ij} x_j \right)^2$$

is a Lagrange expansion, where the outer coefficients are heuristically positive as long as possible as well as small for the first few coefficients.

4.3. Avoiding the Need for Block Diagonal LDLT. Sometimes an LDLT-decomposition with a proper diagonal matrix D is not possible if at some point no non-zero diagonal entries remain. The standard example for this case is $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. This matrix (and its multiples) are the only two dimensional examples. In three dimensions, for $x, y, z \neq 0$ the matrices

$$\begin{pmatrix} 0 & x & y \\ x & 0 & z \\ y & z & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & x \\ 0 & y & 0 \\ x & 0 & 0 \end{pmatrix}, \begin{pmatrix} \pm \frac{xy}{z} & x & y \\ x & \pm \frac{xz}{y} & z \\ y & z & \pm \frac{yz}{x} \end{pmatrix},$$

where at least one $\pm$ is $-$, and their symmetric permutations are the only examples. This is apparent by performing an LDLT-step with every possible pivot element and comparing the resulting 2×2 matrix with the two dimensional counterparts. Note that none of these matrices are strictly copositive.

In dimension four we can do a similar procedure, comparing the matrices obtained by doing an LDLT-step to the three dimensional matrices, which do not admit an LDLT-decomposition with a proper diagonal matrix. This leads to several systems of polynomial equations. In particular, we find matrices which are both strictly copositive and need blocks in their LDLT-decomposition, for example

$$\begin{pmatrix} 8 & -2 & -8 & 0 \\ -2 & 1 & 0 & 8 \\ -8 & 0 & 24 & 16 \\ 0 & 8 & 16 & 32 \end{pmatrix}.$$

This indicates that we have to take special care of such matrices.

A simple idea to deal with such matrices is to split off a small nonnegative part, i.e. writing

$$Q = \tilde{Q} + N,$$

with $N \in \mathcal{N}^n$ such that $\tilde{Q}$ can be handled as usual. Note that this can be always accomplished if Q is strictly copositive. The examples in two and three dimensions as well as our calculations for the four dimensional case (which is omitted here) indicate that such matrices are very specific. Thus by splitting something off, one should expect to obtain a matrix which can be handled without block matrices.

Since Q is strictly copositive the diagonal elements of Q are a natural choice to take from. In Section 5.2 we describe how to find the necessary bounds in the case of “split” matrices $\tilde{Q} + N$.

To close this section, we list some existence results for LDLT-decompositions without 2×2 blocks. From the Cholesky-decomposition it follows that all positive (and negative) semidefinite matrices possess such decompositions. More generally, every symmetric permutation of a *quasidefinite matrix* possesses one as well [39], i.e. every matrix of the form

$$\begin{pmatrix} -E & A^T \\ A & F \end{pmatrix},$$

where E and F are positive definite matrices. The existence of the decomposition $Q = LDL^T$ (without permutations) is known to be equivalent to the existence of an LU-factorization (cf. [21, Sec. 4.1.2]). Necessary and sufficient conditions for that were given by Okunev and Johnson [30].

5. SPECIAL CASES

5.1. One Difficult Coordinate. This special case can be readily solved by additionally solving just one convex quadratic optimization problem to obtain bounds for the difficult coordinate. To this end we partition

$$Q = \begin{pmatrix} Q^{(n-1)} & q \\ q^T & Q_{nn} \end{pmatrix} \text{ and } x = \begin{pmatrix} \hat{x} \\ x_n \end{pmatrix}$$

with $\hat{x} \in \mathbb{R}^{n-1}$ and $Q^{(n-1)} \in \mathcal{S}^{n-1}$. If we fix x_n , the function

$$Q[x] = Q^{(n)}[\hat{x}] + 2x_n q^T \hat{x} + Q_{nn} x_n^2$$

is a convex quadratic function in $\hat{x}$. Indeed, $Q^{(n-1)}$ is positive definite, since x_n is the only difficult coordinate.

We can obtain bounds then as follows. Define

$$f(s) := \min \{Q[x] : x \in \mathbb{R}_{\geq 0}^n, x_n = s\}.$$

Note that for $z \in \mathbb{Z}_{\geq 0}^n$, if $f(z_n) > \lambda$, then $Q[z] > \lambda$ as well. Thus for $Q[z] \leq \lambda$ to hold, $f(z_n) \leq \lambda$ is a necessary condition. This observation allows us to bound the difficult coordinate, as the map f is very well-behaved.

Lemma 5.1. *For $s \geq 0$ we have $f(s) = s^2 f(1)$.*

Proof. Clearly, $f(0) = 0$, so assume $s > 0$. Then

$$\begin{aligned} f(s) &= \min \{Q[(x_1 \dots x_{n-1} s)^T] : x_1, \dots, x_{n-1} \geq 0\} \\ &= s^2 \min \left\{ Q \left[\begin{pmatrix} x_1 & \dots & x_{n-1} \\ s & \dots & s \end{pmatrix} \right] : x_1, \dots, x_{n-1} \geq 0 \right\} = s^2 f(1). \quad \square \end{aligned}$$

The value $f(1)$ can be calculated by solving a single convex quadratic minimization problem, which can be done efficiently using different methods, see e.g. [7]. The difficult coordinate x_n may then be bounded as $0 \leq x_n \leq \sqrt{\frac{\lambda}{f(1)}}$.

Note that this technique may also be applied to positive semidefinite matrices Q : We fix a difficult coordinate $x_k = 1$, calculate the minimum (which is easy since the corresponding submatrix is positive semidefinite as well!), and finally obtain the bound analogously.

To close this section, we continue with the Examples 1.2, 3.2, and 4.3.

Example 5.2. To find bounds on x_3 we minimize

$$Q[(x_1 \ x_2 \ 1)^T] = 3x_1^2 - 2x_1x_2 + 6x_1 + 2x_2^2 - 2x_2 + 2$$

over $\mathbb{R}_{\geq 0}^2$. The optimal solution is $(0, \frac{1}{2})$ with the objective value $f(1) = \frac{3}{2}$. Taking $\lambda = 2$ then yields $0 \leq x_3 \leq \sqrt{\frac{4}{3}}$, so only the cases $x_3 = 0$ and $x_3 = 1$ have to be considered.

5.2. Matrices in $\mathcal{SPN}$. For this section we assume $Q \in \mathcal{SPN}^n$, e.g. $Q = S + N$ with $S \in \mathcal{S}_{\succcurlyeq 0}^n$ and $N \in \mathcal{N}^n$. Suppose now that we have fixed $x_{k+1}, \dots, x_n$ already and that we want to bound x_k next. A simple way to find bounds for the difficult coordinates is to lower bound $Q[x]$ by $S[x]$ and apply the ideas of Section 5.1 to the positive semidefinite matrix S . This is what we currently do.

Another idea is as follows. We can compute an LDLT-decomposition $S = LDL^T$ for S as well (even without permutations since it can be derived from the Cholesky-decomposition of S), so that from the quadratic inequality

$$\lambda \stackrel{!}{\geq} Q[x] \geq \sum_{i=k}^n D_{ii} \left(x_i + \sum_{j=i+1}^n u_{ij} x_j \right)^2 + \sum_{i=k}^n \sum_{j=k}^n N_{ij} x_i x_j$$

(in x_k) we can find bounds for the difficult coordinate x_k . However, if the rank of S is low and (some of) the coefficients of N are also small, then the inequality may produce inacceptably large bounds. This seems to be not uncommon, which is why we prefer the first approach. A combination, depending on the given matrix might be advantageous.

It is usually not clear whether a given strictly copositive matrix is in $\mathcal{SPN}^n$. This can be tested, however, efficiently by a semidefinite program: The matrix Q is in $\mathcal{SPN}^n$ if and only if there is nonnegative matrix N such that $Q - N \succcurlyeq 0$. Note that $N \in \mathcal{N}^n$ is a linear condition in $\mathcal{S}^n$. Thus by solving the problem

$$\begin{aligned} & \sup \langle O, N \rangle \\ & \text{s.t. } N \in \mathcal{N}^n \\ & \quad Q - N \succcurlyeq 0 \end{aligned}$$

where $O \in \mathcal{S}^n$ is a suitable objective, we can find such a decomposition if it exists. For the objective O we propose the matrix made up entirely of -1 entries. The effect we wish to achieve with this objective is that the nonnegative part N has only a small influence, so that bounding only via S is not too bad.

Since the copositive cone and $\mathcal{SPN}^n$ coincide for $n \leq 4$, our new method is always applicable for $n \leq 4$. In general, this method works for all non-exceptional copositive matrices.

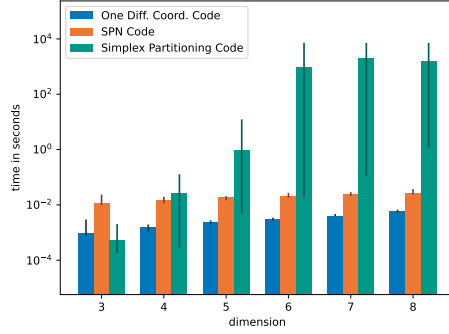
6. NUMERICAL RESULTS

We implemented the algorithm with the two special cases to bound the difficult coordinates. It is a preliminary implementation using Python [38], numpy [23], Gurobi [41], cvxpy [13, 1], CVXOPT [42], and OSQP [36]. The solution described in Section 4.3 is not yet implemented. Note that our implementation (the bound calculation specifically) relies currently on non-rigorous floating point arithmetic. The code and the matrices used for testing are publicly available [43].

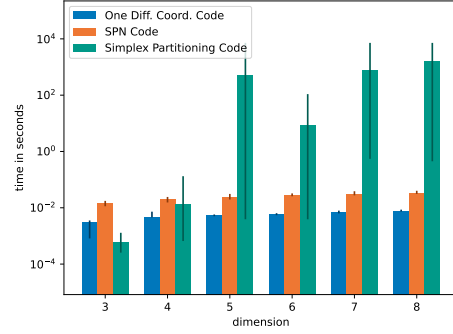
We have also tested our algorithms on multiple classes of matrices in dimensions three to eight, with 15 matrices for each class and dimension. The classes are positive semidefinite matrices, $\mathcal{SPN}$ -matrices, $\mathcal{SPN}$ -matrices with at least two negative eigenvalues, and, lastly, perfect copositive matrices. The matrices of the last class have been manually sampled in the early stages of the certificate algorithm of [18]. The other matrices have been randomly sampled as BB^T from Sage's [44] `random_matrix`. We recorded the runtime and applicability of our algorithms. For comparison we also tested the implementation of the algorithm using a simplicial

partition, introduced in [18]. In particular, we used [45], compiled on July 25, 2025. If the execution time, however, exceeded two hours for a single matrix, we aborted the calculation and used two hours as the recorded time. We also recorded the execution time of our algorithms from the moment the necessary libraries (e.g. numpy, gurobipy, cvxpy) were loaded, since in many examples, these loading times are significantly longer than the actual code execution time and would not be present in a more thorough implementation. The tests were done on a local machine with an Intel Core i7-1185G7 with 32GiB of RAM running Ubuntu 24.04.2. In Fig. 2 the results are displayed.

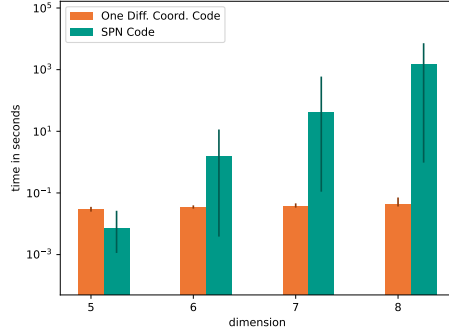
It is clear that our new methods offer a tremendous speedup in dimensions five and above, if they are applicable. They also appear to be a lot more consistent and predictable in their runtime in comparison to the method on simplicial subdivisions. Furthermore, they appear to scale better to higher dimensions.



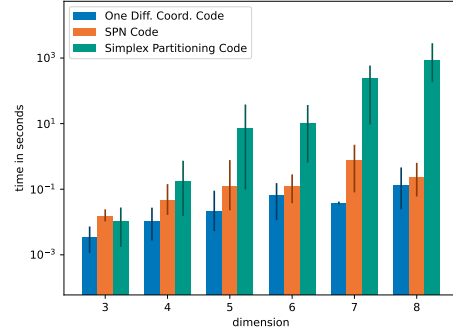
(A) Test on positive semidefinite matrices. All examples could be solved by our algorithms.



(B) Test on SPN matrices. The one difficult coordinate code was not applicable for 2 of the 90 examples.



(C) Test on SPN matrices with at least two negative eigenvalues. The one difficult coordinate code was not applicable here.



(D) Test on perfect copositive matrices. The one difficult coordinate code was not applicable for 22 of the 90 examples.

FIGURE 2. Runtime of the algorithms on a logarithmic timescale. The broad bar indicates the average runtime of the algorithm on the 15 examples of the dimension, the lower end of the small bar indicates the minimum runtime of the algorithm on the 15 examples of that dimension, and the upper end of the small bar indicates the maximum runtime of the algorithm on the 15 examples of that dimension. Overall, the code for SPN -matrices was always applicable, while the code for one difficult coordinate matrices was not applicable for 84 out of 359 examples.

7. OPEN QUESTIONS AND FURTHER DIRECTIONS

In this final section we discuss some open questions and possible further research directions.

7.1. Extensions to All Copositive Matrices. Section 5 was largely concerned with finding good approximations for Eq. (3). If one had a decent and reliable way to find such upper bounds, the approach using an LDLT-decomposition would work for all input matrices in general. However, due to the non-convex nature of the problem, this has turned out to be very difficult. For example, the standard Shor (SDP) relaxation turns out to be unbounded.

A promising approach seems to be to relax the problem

$$\max \{x_k^2 : x \geq 0, Q[x] \leq 1\}$$

to

$$\begin{aligned} & \max \{ \langle O, xx^T \rangle : x \geq 0, \langle Q, xx^T \rangle \leq 1 \} \\ & \leq \sup \{ \langle O, Y \rangle : Y \in \mathcal{CP}^n, \langle Q, Y \rangle \leq 1 \}, \end{aligned}$$

where O is a matrix everywhere zero except at $O_{kk} = 1$. This is a in general NP-hard problem and solving it exactly becomes quickly intractable. A sufficient relaxation might, however, be easier to find here.

A second approach is to combine our $\mathcal{SPN}$ special case with the simplicial partition used in [18]. Some preliminary tests indicate that the number of simplicial cones necessary for our method to become applicable is generally quite low. Further, one could replace the semidefinite problems we are solving with the approach of [24] to calculate the $\mathcal{SPN}$ decomposition.

7.2. Reduction Theory. One of the highlights of the classical theory of quadratic forms is its reduction theory. Likewise, the much celebrated *LLL-algorithm* [28] is a lattice algorithm, which allows, for instance, for a 2^n approximation to SVP in polynomial time. This algorithm is indispensable in the computational aspects of lattice theory and a significant question is if something similar can be done in the copositive situation as well. This would allow for efficient preprocessing, similar to what is done for the shortest vector problem.

The major hurdle is that by replacing $Q \in \text{int } \mathcal{COP}^n$ with $U^T Q U$ for an unimodular matrix U , we also change the cone in which we search for the shortest vectors. In particular, we have to search for integer vectors in the polyhedral cone

$$C := \{x \in \mathbb{R}^n : Ux \geq 0\}.$$

The problem of having a different cone already appears in [18] and is solved there by an application of the Hermite normal form. By the unimodularity of U this is in our case the same as reversing the reduction.

Another drawback is that it is not clear how to apply the important special case of Section 5.2.

Special care has also be put into how the reduction is done. The idea to split the quadratic form Q into a positive and negative part, so that $Q = Q^+ - Q^-$, and applying the LLL-algorithm to $Q^+ + Q^-$ seems to increase the number of difficult coordinates. A more promising approach is the use of an adapted *indefinite-LLL-algorithm* [34], which works for indefinite quadratic forms as well by adjusting the so called Lovász condition to disallow negative numbers.

7.3. Special Algorithms for Perfect Copositive Matrices. With a view towards the application of finding $\mathcal{CP}$ -certificates, it makes sense to consider special algorithms for perfect copositive matrices. We recall that we search systematically for perfect copositive matrices by traversing the neighborhood graph of the copositive Ryshkov polyhedron and that two contiguous perfect copositive matrices share a very large portion of their minimal vectors. This very strong property has so far been unused and might offer significant speedups in the certificate algorithm of [18], if leveraged correctly.

Moreover, these speedups may also be beneficial for other applications. One quite promising idea is that we can find polyhedral approximations for the complicated cone $\mathcal{CP}^n$ using perfect copositive matrices: Taking a finite set $\mathcal{P}$ of perfect copositive matrices, we have an outer polyhedral approximation

$$\mathcal{CP}^n \subset \{Q \in \mathcal{N}^n : \langle Q, P \rangle \geq 0 \text{ for all } P \in \mathcal{P}\}.$$

Using the Voronoi cone for each perfect copositive matrix of $\mathcal{P}$, we also have an inner polyhedral approximation

$$\text{cone} \{xx^T : x \in \text{Min}_{\mathcal{COP}} P \text{ for } P \in \mathcal{P}\} \subset \mathcal{CP}^n.$$

Both of these polyhedral approximations can be arbitrary close to $\mathcal{CP}^n$ (cf. [12]). Note that by duality we also get arbitrarily close linear approximations to $\mathcal{COP}^n$. Since the certificate algorithm produces perfect copositive matrices as a byproduct, we may also use it to systematically generate such polyhedral approximations. These approximations have not been investigated yet. If they can be calculated sufficiently fast, they may prove very useful for solving or bounding copositive optimization problems.

ACKNOWLEDGEMENT

We like to thank Wessel van Woerden for helpful discussions on the complexity of the problem. We also like to thank Thomas Kalinowski and Matthias Schymura for helpful comments on the manuscript. Both authors gratefully acknowledge support by the German Research Foundation (DFG) under grant SCHU 1503/8-1.

REFERENCES

- [1] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. “A rewriting system for convex optimization problems”. In: *Journal of Control and Decision* 5.1 (2018), pp. 42–60. DOI: 10.1080/23307706.2017.1397554.
- [2] Miklós Ajtai. “The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract)”. In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. STOC ’98. Dallas, Texas, USA: Association for Computing Machinery, 1998, pp. 10–19. DOI: 10.1145/276698.276705.
- [3] Riley Badenbroek and Etienne de Klerk. “An analytic center cutting plane method to determine complete positivity of a matrix”. English. In: *INFORMS J. Comput.* 34.2 (2022), pp. 1115–1125. DOI: 10.1287/ijoc.2021.1108.
- [4] Huck Bennett. “The Complexity of the Shortest Vector Problem”. In: *SIGACT News* 54.1 (Mar. 2023), pp. 37–61. DOI: 10.1145/3586165.3586172.
- [5] Avi Berman, Mirjam Dür, and Naomi Shaked-Monderer. “Open problems in the theory of completely positive and copositive matrices”. English. In: *Electron. J. Linear Algebra* 29 (2015), pp. 46–58. DOI: 10.13001/1081-3810.2943.

- [6] Immanuel Bomze, Werner Schachinger, and Gabriele Uchida. “Think co-(mpletely) positive! Matrix properties, examples and a clustered bibliography on copositive optimization”. English. In: *Journal of Global Optimization* 52.3 (2012), pp. 423–445. DOI: 10.1007/s10898-011-9749-3.
- [7] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [8] James R. Bunch and Linda Kaufman. “Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems”. In: *Mathematics of Computation* 31.137 (1977), pp. 163–179. DOI: 10.2307/2005787.
- [9] James. R. Bunch and Beresford. N. Parlett. “Direct Methods for Solving Symmetric Indefinite Systems of Linear Equations”. In: *SIAM Journal on Numerical Analysis* 8.4 (1971), pp. 639–655. DOI: 10.1137/0708060.
- [10] Stefan Bundfuss and Mirjam Dür. “Algorithmic copositivity detection by simplicial partition”. In: *Linear Algebra and its Applications* 428.7 (2008), pp. 1511–1523. DOI: <https://doi.org/10.1016/j.laa.2007.09.035>.
- [11] Henri Cohen. *A course in computational algebraic number theory*. English. Vol. 138. Grad. Texts Math. Berlin: Springer-Verlag, 1993.
- [12] Valentin Dannenberg and Achill Schürmann. “Perfect Copositive Matrices”. In: *Communications in Mathematics* Volume 31 (2023), Issue 2 (Special issue: Euclidean lattices: theory and applications), 7 (July 2023). DOI: 10.46298/cm.11141.
- [13] Steven Diamond and Stephen Boyd. “CVXPY: A Python-embedded modeling language for convex optimization”. In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5. URL: <http://jmlr.org/papers/v17/15-408.html>.
- [14] Palahenedi H. Diananda. “On non-negative forms in real variables some or all of which are non-negative”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 58.1 (Jan. 1962), pp. 17–25. DOI: 10.1017/S0305004100036185.
- [15] Peter J. C. Dickinson and Luuk Gijben. “On the computational complexity of membership problems for the completely positive cone and its dual”. In: *Computational Optimization and Applications* 57.2 (Mar. 2014), pp. 403–415. DOI: 10.1007/s10589-013-9594-z.
- [16] Mirjam Dür. “Copositive Programming – a Survey”. In: *Recent Advances in Optimization and its Applications in Engineering*. Ed. by Moritz Diehl, Francois Glineur, Elias Jarlebring, and Wim Michiels. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 3–20. DOI: 10.1007/978-3-642-12598-0_1.
- [17] Mathieu Dutour Sikirić, Achill Schürmann, and Frank Vallentin. “Rational factorizations of completely positive matrices”. In: *Linear Algebra and its Applications* 523 (2017), pp. 46–51. DOI: <https://doi.org/10.1016/j.laa.2017.02.017>.
- [18] Mathieu Dutour Sikirić, Achill Schürmann, and Frank Vallentin. “A simplex algorithm for rational cp-factorization”. In: *Mathematical Programming* 187.1 (May 2021), pp. 25–45. DOI: 10.1007/s10107-020-01467-4.
- [19] Ulrich Fincke and Michael Pohst. “Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis”. In: *Mathematics of Computation* 44.170 (1985), pp. 463–471. DOI: 10.2307/2007966.
- [20] Michael R. Garey and David S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness*. English. A Series of Books in the mathematical Sciences. San Francisco: W. H. Freeman and Company. X, 338 p. 1979.

- [21] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. English. 4th ed. Baltimore, MD: The Johns Hopkins University Press, 2013. DOI: 10.1137/1.9781421407944.
- [22] M. Seetharama Gowda, Roman Sznajder, and Jiyuan Tao. “The automorphism group of a completely positive cone and its Lie algebra”. In: *Linear Algebra and its Applications* 438.10 (2013). Special issue in honor of Abraham Berman, Moshe Goldberg, and Raphael Loewy, pp. 3862–3871. DOI: <https://doi.org/10.1016/j.laa.2011.10.006>.
- [23] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.
- [24] Wilfred Kaplan. “A copositivity probe”. In: *Linear Algebra and its Applications* 337.1 (2001), pp. 237–251. DOI: 10.1016/S0024-3795(01)00351-2.
- [25] Jon Kleinberg and Éva Tardos. *Algorithm design*. 1. ed. Boston, Mass.: Pearson / Addison-Wesley, 2006.
- [26] Igor Klep, Tea Štrekelj, and Aljaž Zalar. “A Random Copositive Matrix Is Completely Positive with Positive Probability”. In: *SIAM Journal on Applied Algebra and Geometry* 8.3 (2024), pp. 583–611. DOI: 10.1137/23M1583491.
- [27] Zhijian Lai and Akiko Yoshise. “Completely positive factorization by a Riemannian smoothing method”. In: *Computational Optimization and Applications* 83.3 (Dec. 2022), pp. 933–966. DOI: 10.1007/s10589-022-00417-4.
- [28] Arjen K. Lenstra, Hendrik W. Lenstra, and László Lovász. “Factoring polynomials with rational coefficients”. In: *Mathematische Annalen* 261.4 (Dec. 1982), pp. 515–534. DOI: 10.1007/BF01457454.
- [29] Daniele Micciancio and Michael Walter. “Fast Lattice Point Enumeration with Minimal Overhead”. In: *Proceedings of the 2015 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 276–294. DOI: 10.1137/1.9781611973730.21.
- [30] Pavel Okunev and Charles R. Johnson. *Necessary And Sufficient Conditions For Existence of the LU Factorization of an Arbitrary Matrix*. 2005. arXiv: [math/0506382](https://arxiv.org/abs/math/0506382) [math.NA].
- [31] Achill Schürmann. *Computational geometry of positive definite quadratic forms. Polyhedral reduction theories, algorithms, and applications*. English. Vol. 48. Univ. Lect. Ser. Providence, RI: American Mathematical Society (AMS), 2009. DOI: 10.1090/ulect/048.
- [32] Naomi Shaked-Monderer and Abraham Berman. *Copositive and completely positive matrices*. English. 2nd updated and extended edition. Hackensack, NJ: World Scientific, 2021. DOI: 10.1142/11386.
- [33] Yaroslav Shitov. “Linear mappings preserving the copositive cone”. English. In: *Proc. Am. Math. Soc.* 149.8 (2021), pp. 3173–3176. DOI: 10.1090/proc/15432.
- [34] Denis Simon. “Solving Quadratic Equations Using Reduced Unimodular Quadratic Forms”. In: *Mathematics of Computation* 74.251 (2005), pp. 1531–1543. DOI: 10.1090/S0025-5718-05-01729-1.

- [35] Anthony Man–Cho So. “A semidefinite programming approach to the graph realization problem: Theory, applications and extensions”. PhD thesis. Stanford, California: Stanford University, June 2007. URL: <https://www1.se.cuhk.edu.hk/~manchoso/papers/thesis.pdf>.
- [36] Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. “OSQP: an operator splitting solver for quadratic programs”. In: *Mathematical Programming Computation* 12.4 (2020), pp. 637–672. DOI: 10.1007/s12532-020-00179-2.
- [37] Tea Štrekelj and Aljaž Zalar. *Construction of exceptional copositive matrices*. 2025. arXiv: 2502.20133 [math.FA].
- [38] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.
- [39] Robert J. Vanderbei. “Symmetric quasidefinite matrices”. English. In: *SIAM J. Optim.* 5.1 (1995), pp. 100–113. DOI: 10.1137/0805005.
- [40] Masaya Yasuda. “A Survey of Solving SVP Algorithms and Recent Strategies for Solving the SVP Challenge”. In: *International Symposium on Mathematics, Quantum Theory, and Cryptography*. Ed. by Tsuyoshi Takagi, Masato Wakayama, Keisuke Tanaka, Noboru Kunihiro, Kazufumi Kimoto, and Yasuhiko Ikematsu. Singapore: Springer Singapore, 2021, pp. 189–207. DOI: 10.1007/978-981-15-5191-8_15.

SOFTWARE

- [41] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2024. URL: <https://www.gurobi.com>.
- [42] Joachim Dahl Martin S. Andersen and Lieven Vandenbergh. *CVXOPT: A Python package for convex optimization. Version 1.3.2*. 2023. URL: <https://cvxopt.org>.
- [43] Alexander Oertel. *MinCOP_LDLT. A Python package for computing the copositive minimum*. 2025. URL: https://github.com/AlexOertel/MinCOP_LDLT.
- [44] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 10.4)*. 2024. URL: <https://www.sagemath.org>.
- [45] Mathieu Dutour Sikiric. *Copositivity*. 2018. URL: https://github.com/MathieuDutSik/polyhedral_common/.

A. OERTEL, UNIVERSITÄT ROSTOCK, INSTITUTE OF MATHEMATICS, 18051 ROSTOCK, GERMANY

Email address: alexander.oertel@uni-rostock.de

A. SCHÜRMANN, UNIVERSITÄT ROSTOCK, INSTITUTE OF MATHEMATICS, 18051 ROSTOCK, GERMANY

Email address: achill.schuermann@uni-rostock.de