

BEYOND BINARITY: SEMIDEFINITE PROGRAMMING FOR TERNARY QUADRATIC PROBLEMS

FRANK DE MEIJER*, VERONICA PICCIALLI†, RENATA SOTIROV‡, AND ANTONIO M. SUDOSO§

Abstract. We study the ternary quadratic problem (TQP), a quadratic optimization problem with linear constraints where the variables take values in $\{0, \pm 1\}$. While semidefinite programming (SDP) techniques are well established for $\{0, 1\}$ - and $\{\pm 1\}$ -valued quadratic problems, no dedicated integer semidefinite programming framework exists for the ternary case. In this paper, we introduce a ternary SDP formulation for the TQP that forms the basis of an exact solution approach.

We derive new theoretical insights in rank-one ternary positive semidefinite matrices, which lead to a basic SDP relaxation that is further strengthened by valid triangle, RLT, split and k -gonal inequalities. These are embedded in a tailored branch-and-bound algorithm that iteratively solves strengthened SDPs, separates violated inequalities, applies a ternary branching strategy and computes high-quality feasible solutions. We test our algorithm on TQP variations motivated by practice, including unconstrained, linearly constrained and quadratic ratio problems. Computational results on these instances demonstrate the effectiveness of the proposed algorithm.

Key words. ternary optimization problem, integer semidefinite programming, branch-and-bound, quadratic ratio problems

MSC codes. 90C10, 90C20, 90C22, 90C26

1. Introduction. Integer semidefinite programming (ISDP) may be seen as a generalization of integer linear programming, where the vector of variables is replaced by a positive semidefinite (PSD) matrix variable whose elements are constrained to be integers. Recent results show that integer semidefinite programs allow for more compact and substantially different formulations for several problem classes compared to traditional modeling approaches, and also enable the development of new algorithmic techniques. These programs do not include \mathcal{NP} -hard rank constraints, but utilize an alternative notion of exactness in terms of integrality of PSD matrices. Although it is straightforward to embed a linear integer program (IP) into the space of diagonal integer PSD matrices, it is generally not clear how to construct an ISDP program for a quadratic IP. However, it is well known that quadratic optimization problems with variables in $\{\pm 1\}$, e.g., the max-cut problem, can be formulated as ISDP problems, see e.g., [18, 22]. It is also known how to reformulate binary quadratically constrained quadratic programs and binary quadratic matrix programs as binary semidefinite programs [16]. In this work, we introduce an ISDP formulation for the ternary quadratic problem (TQP), that is an optimization problem with quadratic objective and linear constraints, where the decision variables take values in $\{0, \pm 1\}$. To the best of our knowledge, there are no up to date results on ternary semidefinite programs.

The TQP belongs to the class of nonlinear IPs, which have received increased attention in recent years. Ternary optimization problems are frequently studied in the literature as special cases of quadratic integer programs. In 2010, Letchford [32]

*Delft Institute of Applied Mathematics, Delft University of Technology, The Netherlands. f.j.j.demeijer@tudelft.nl

†Department of Computer, Control and Management Engineering “Antonio Ruberti”, Sapienza University of Rome, Italy. veronica.piccialli@uniroma1.it

‡CentER, Department of Econometrics and OR, Tilburg University, The Netherlands. r.sotirov@uvt.nl

§Department of Computer, Control, and Management Engineering “Antonio Ruberti”, Sapienza University of Rome, Italy. antoniomaria.sudoso@uniroma1.it

studied integer quadratic quasi-polyhedra, i.e., polyhedra with a countably infinite number of facets. Such polyhedra arise as feasible sets of unconstrained integer quadratic programming problems. Letchford’s work was extended to convex sets for nonconvex mixed-integer quadratic programming in [10]. Buchheim et al. [9, 7] propose semidefinite programming (SDP) relaxations for nonconvex mixed-integer quadratic optimization problems, including an SDP relaxation for the TQP. These relaxations are derived from exact problem reformulations that involve a nonconvex rank-one constraint.

Approaches for solving optimization problems with ternary variables also appear in the broader literature on general quadratic integer programming, where the TQP is often considered in computational experiments. In the following, we provide a list of these works. In 2012, Buchheim et al., [5] present a branch-and-bound (B&B) algorithm for solving convex quadratic unconstrained IPs for instances of up to 120 variables. A B&B algorithm for nonconvex quadratic IPs with box constraints, presented in [6], solves instances with up to 50 variables. An SDP-based B&B framework for nonconvex quadratic mixed-integer programming, introduced in [9] and improved in [7], solves instances of the nonconvex quadratic unconstrained ternary optimization problem (QUTO) up to 80 variables and TQP instances with one linear inequality up to 50 variables. For the closely related max-cut problem, several SDP-based solvers exist, see e.g., [30, 31, 25]. The parallel version of the solver MADAM [30] solves instances on graphs with up to 180 vertices. We note that general nonconvex mixed-integer programming solvers can also be used to solve nonconvex quadratic IPs, such as COUENNE [3], BARON [43], and Gurobi [24]. However, these solvers are often outperformed on nonconvex TQP instances by the specialized solvers mentioned earlier, while they perform well on TQP instances with a convex objective function.

An SDP-based approximation algorithm for quadratic programs with ratio objective and ternary variables is derived in [4], and an SDP-based randomized algorithm for the sparse integer least squares problem (SILS) was studied in [39]. A heuristic approach for the TQP with circulant matrix is presented in [11].

We highlight several applications of optimization problems with ternary variables. Problems arising in engineering applications may be formulated as the TQP. For instance, a variant of TQP with a single linear constraint appears in the design of turbomachines [38]. In electronics, the filtered approximation problem can be modeled as a convex QUTO [11]. Quadratic programs with ratio objectives and ternary variables arise when modeling density clustering problems with two voting options, including the possibility of abstention [2], in PageRank for folksonomies in social media [29], in discovering conflicting groups in signed networks [48], and when computing max-cut gain using eigenvalue techniques [47]. The SILS problem finds applications in many fields, including cybersecurity, sensor networks, digital fingerprints, array signal processing, and multiuser detection algorithms, see e.g., [39] and references therein.

Main results and outline. This paper studies theoretical and algorithmic approaches for the TQP, resulting in an SDP-based B&B algorithm for general ternary quadratic programs. Our algorithm is tested on various special cases of the TQP inspired by practical applications, showing its effectiveness compared to general nonconvex mixed-integer programming solvers.

Our approach starts with a theoretical analysis of the set of ternary PSD matrices. We recall some known results on integer PSD matrices and extend these with several new characterizations of the set of ternary PSD matrices. One of these characterizations induces a ternary SDP formulation for the TQP. Relaxation of the integrality

constraints yields an SDP relaxation, which serves as the foundation of our approach. We show that the relaxation can be strengthened by a large number of linear inequalities, including the triangle, k -gonal, RLT (i.e., reformulation linearization technique), standard split and non-standard split inequalities.

The basic SDP relaxation and the derived inequalities are combined in a specialized B&B algorithm for solving nonconvex TQP instances. The strengthened SDP bounds are incorporated in a ternary branching framework, where a Variable Neighborhood Search (VNS) heuristic is embedded to obtain high-quality feasible solutions throughout the algorithm. Our B&B algorithm is tested on three special cases of the TQP. The first two cases correspond to the unconstrained version of the problem (i.e., QUTO) and the TQP with one linear constraint (i.e., TQP-Linear) [38]. The third type of TQP involves the minimization of the ratio of two quadratic functions over the set $\{0, \pm 1\}^n$, which we denote by TQP-Ratio [2, 29, 47]. These problems can be solved using a parametrized approach inspired by [19], where each subproblem boils down to solving a QUTO. Alternatively, we also introduce an SDP relaxation for TQP-Ratio problems based on a reformulation of the objective function inspired by the convexification approach from [27].

In our computational analysis, we compare our B&B algorithm with state-of-the-art commercial solvers, such as Gurobi [24]. Our results show that we significantly outperform Gurobi on all three TQP variations. For the instances with up to 90 variables, our approach is able to solve all instances within a time limit of 3,600 seconds, whereas Gurobi fails on a substantial number of them. Also for larger instances, our B&B algorithm either solves instances to optimality or is able to produce high-quality solutions, with remaining gaps typically below about 2%. This study shows that our algorithm is currently the best known algorithm for solving nonconvex TQP instances.

This paper is organized as follows. Theoretical results on ternary PSD matrices and their characterizations are discussed in Section 2. In Section 3, we introduce the TQP and reformulate it as an SDP with integrality constraints. Its SDP relaxation and various families of inequalities are derived in Section 4. In Section 5, we present three special cases of the TQP. Our B&B algorithm is introduced in Section 6. Finally, in Section 7 numerical results are presented.

Notation. For $n \in \mathbb{Z}_+$, we define the set $[n] := \{1, \dots, n\}$. We denote by $\mathbf{0}_n$ and $\mathbf{1}_n$ the vector of zeros and ones in \mathbb{R}^n , respectively. The $n \times n$ matrix of ones is denoted by \mathbf{J}_n . The $n \times n$ identity matrix is denoted by \mathbf{I}_n . We omit the subscripts of these matrices when their sizes are clear from the context. For fixed n , matrices of size $(n+1) \times (n+1)$ and vectors of size $(n+1) \times 1$ are indexed from zero. We denote by \mathbf{e}_i the i^{th} unit vector. For any $S \subseteq [n]$, $\mathbb{1}_S$ denotes the indicator vector induced by S . We denote the set of all $n \times n$ real symmetric matrices by \mathcal{S}^n , and by X_J the $|J| \times |J|$ principal submatrix of $X \in \mathcal{S}^n$ with rows and columns indexed by $J \subseteq [n]$. The cone of symmetric PSD matrices of order n is defined as $\mathcal{S}_+^n := \{X \in \mathcal{S}^n : X \succeq \mathbf{0}\}$. The trace of a square matrix $X = (X_{ij})$ is given by $\text{tr}(X) = \sum_i X_{ii}$. For any $X, Y \in \mathcal{S}^n$ the trace inner product is defined as $\langle X, Y \rangle := \text{tr}(XY) = \sum_{i,j=1}^n X_{ij}Y_{ij}$. For $x \in \mathbb{R}^n$ we denote by $|x|$ the vector whose elements are the absolute values of the corresponding elements of x , and by $\text{supp}(x) \subseteq [n]$ its support. The operator $\text{diag} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^n$ maps a square matrix to a vector consisting of its diagonal elements. We denote by $\text{Diag} : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ its adjoint operator. We use $\text{conv}\{S\}$ to denote the convex hull of the set S .

2. Characterizations of PSD $\{0, \pm 1\}$ -matrices and related polyhedra.

In this section, we present three characterizations of ternary PSD matrices, which are

then used to derive novel reformulations of polytopes from the literature [32, 10]. We also relate these polytopes with the well-known Boolean quadric polytope and the set-completely positive matrix cone.

The following characterization of PSD $\{0, \pm 1\}$ -matrices is derived in [16].

PROPOSITION 2.1 ([16]). *Let $X \in \{0, \pm 1\}^{n \times n}$ be symmetric. Then, $X \succeq \mathbf{0}$ if and only if $X = \sum_{i=1}^r x_i x_i^\top$ for some $x_i \in \{0, \pm 1\}^n$, $i \in [r]$.*

Observe that the vectors x_i in the above proposition have disjoint supports. There exists a similar decomposition of PSD $\{0, 1\}$ -matrices [33]. The following characterization of ternary PSD matrices follows directly from the previous proposition.

COROLLARY 2.2. *A symmetric ternary matrix is positive semidefinite if and only if it equals TT^\top for some ternary matrix T .*

The following result from [16] relates a PSD $\{0, \pm 1\}$ -matrix of rank one with an extended linear matrix inequality.

PROPOSITION 2.3 ([16]). *Let $Y = \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \in \mathcal{S}^{n+1}$ with $\text{diag}(X) = |x|$. Then, $Y \in \{0, \pm 1\}^{(n+1) \times (n+1)}$, $Y \succeq \mathbf{0}$ if and only if $X = xx^\top$.*

One can also replace $\text{diag}(X) = |x|$ by $\text{supp}(\text{diag}(X)) = \text{supp}(x)$ in Proposition 2.3.

In the sequel, we first present known results that characterize PSD matrices with entries in $\{\pm 1\}$ and $\{0, 1\}$ by using inequalities, and then we derive a similar result for PSD $\{0, \pm 1\}$ -matrices. The following well-known result characterizes PSD $\{\pm 1\}$ -matrices using the triangle inequalities and a diagonal constraint.

PROPOSITION 2.4 ([18]). *Let $X \in \{\pm 1\}^{n \times n}$ be symmetric. Then, $X = xx^\top$ for some $x \in \{\pm 1\}^n$ if and only if X satisfies*

$$\begin{aligned} X_{ij} + X_{ik} + X_{jk} &\geq -1, & -X_{ij} + X_{ik} - X_{jk} &\geq -1 & \forall 1 \leq i < j < k \leq n \\ X_{ij} - X_{ik} - X_{jk} &\geq -1, & -X_{ij} - X_{ik} + X_{jk} &\geq -1 & \forall 1 \leq i < j < k \leq n \\ X_{ii} &= 1 & & & \forall i \in [n]. \end{aligned}$$

One can replace the rank-one condition from Proposition 2.4 by the positive semidefiniteness constraint. Namely, for a symmetric matrix X , we have that $X \succeq \mathbf{0}$, $X \in \{\pm 1\}^{n \times n}$ if and only if $X = xx^\top$ for some $x \in \{\pm 1\}^n$ [22].

A characterization of PSD $\{0, 1\}$ -matrices in terms of inequalities is given below.

PROPOSITION 2.5 ([33]). *Let $X \in \{0, 1\}^{n \times n}$ be symmetric with $n \geq 3$. Then, $X \succeq \mathbf{0}$ if and only if X satisfies*

$$\begin{aligned} X_{ik} + X_{jk} &\leq X_{kk} + X_{ij} & \forall 1 \leq i < j \leq n, k \neq i, j \\ X_{ij} &\leq X_{ii} & \forall 1 \leq i < j \leq n. \end{aligned}$$

For the case of ternary PSD matrices, we derive the following result.

THEOREM 2.6. *Let $X \in \{0, \pm 1\}^{n \times n}$ be symmetric with $n \geq 3$. Then, $X \succeq \mathbf{0}$ if and only if X satisfies*

$$(2.1) \quad X_{ij} + X_{ik} + X_{jk} \geq -1, \quad -X_{ij} + X_{ik} - X_{jk} \geq -1 \quad \forall 1 \leq i < j < k \leq n$$

$$(2.2) \quad X_{ij} - X_{ik} - X_{jk} \geq -1, \quad -X_{ij} - X_{ik} + X_{jk} \geq -1 \quad \forall 1 \leq i < j < k \leq n$$

$$(2.3) \quad X_{ij} \leq X_{ii}, \quad X_{ij} \geq -X_{ii} \quad \forall i, j \in [n], i \neq j.$$

Proof. (\Rightarrow): Let $X \succeq \mathbf{0}$. Then, by Proposition 2.1 it follows that there exist $x_i \in \{0, \pm 1\}^n$ for $i \in [r]$ such that $X = \sum_{i=1}^r x_i x_i^\top$. Let $K_i = \text{supp}(x_i)$ for all $i \in [r]$ and define $K_0 = [n] \setminus (\bigcup_{i \in [r]} K_i)$. Since the vectors x_i have disjoint support, $\{K_0, K_1, \dots, K_r\}$ forms a partition of $[n]$.

It follows from Proposition 2.4 that for each i, j, k within the same set K_ℓ of the partition, the inequalities (2.1)–(2.2) are satisfied. Now, let us assume that i and j are in the same subset of the partition, while k is in a different subset. Then, $X_{ik} = X_{jk} = 0$, while X_{ij} can be either 1 or -1 . One can easily check that (2.1)–(2.2) are satisfied. Finally, if i, j and k are all in separate subsets of the partition, then $X_{ij} = X_{ik} = X_{jk} = 0$, which also satisfy (2.1)–(2.2).

Let us now consider the inequalities (2.3). Since $X \succeq \mathbf{0}$, each diagonal element X_{ii} is either 0 or 1. If $X_{ii} = 1$, then (2.3) is clearly satisfied for all $j \in [n]$. If $X_{ii} = 0$, then $X_{ij} = 0$ for all $j \in [n]$ due to the fact that X is PSD and thus, (2.3) are also satisfied. We conclude that X satisfies the inequalities (2.1)–(2.3).

(\Leftarrow): Now, assume that X satisfies (2.1)–(2.3). From (2.3) it follows that $X_{ii} \in \{0, 1\}$ for all $i \in [n]$. Let $K_0 := \{i \in [n] : X_{ii} = 0\}$. It follows from (2.3) that $X_{ij} = X_{ji} = 0$ for $i \in K_0$ and $j \in [n]$. Now, take some $i^* \in [n] \setminus K_0$ and define $K_1 := \{j \in [n] : X_{i^*j} \neq 0\}$. We show that the submatrix of X induced by K_1 has entries in $\{\pm 1\}$. Clearly, this submatrix cannot have zeros on the diagonal. Now, suppose $X_{jk} = 0$ for some $j, k \in K_1$. Since $X_{i^*j}, X_{i^*k} \in \{\pm 1\}$, one can verify that at least one of the triangle inequalities is violated in case $X_{jk} = 0$. Thus, the submatrix of X induced by K_1 contains only entries in $\{\pm 1\}$.

Next, we show that $X_{jk} = 0$ for all $j \in K_1, k \notin K_1$. It follows from the definition of K_1 that $X_{i^*k} = 0$. Moreover, as argued above, we must have $X_{i^*j} \in \{\pm 1\}$. Suppose $X_{i^*j} = 1$. Then, X_{jk} cannot be 1, since then $-X_{i^*j} + X_{i^*k} - X_{jk} = -1 + 0 - 1 < -1$, which violates one of the triangle inequalities (2.1)–(2.2). Similarly, X_{jk} cannot be -1 , since then $-X_{i^*j} - X_{i^*k} + X_{jk} = -1 - 0 + (-1) < -1$, also violating the triangle inequalities. Thus, $X_{jk} = 0$. The proof for the case $X_{i^*j} = -1$ is similar. We conclude that $X_{jk} = 0$ for all $j \in K_1, k \notin K_1$.

Now, we take $j^* \in [n] \setminus (K_0 \cup K_1)$ and define $K_2 := \{j \in [n] : X_{j^*j} \neq 0\}$. By the same arguments as above, it follows that the submatrix of X induced by K_2 has entries in $\{\pm 1\}$ and that $X_{ij} = 0$ for all $i \in K_2, j \notin K_2$. By iteratively repeating this argument, we obtain a partition $\{K_0, K_1, \dots, K_r\}$ of $[n]$ that defines a non-overlapping block structure in X , where K_0 corresponds to a block of zeros and each $K_\ell, \ell \neq 0$, defines a block with entries in $\{\pm 1\}$. It follows from Proposition 2.4 that there exist $x_i \in \{0, \pm 1\}^n$ for $i \in [r]$ such that $X = \sum_{i=1}^r x_i x_i^\top$, hence $X \succeq \mathbf{0}$. \square

The inequalities from Theorem 2.6 can be used to strengthen SDP relaxations of TQPs.

REMARK 1. *Similar to the binary PSD matrices [33], the ternary PSD matrices have a graphical representation. Let X be a $n \times n$ rank-one ternary PSD and let $S = \{i \in [n] : X_{ii} = 1\}$. Then, the submatrix of X induced by S corresponds to the edge cut-set of the subgraph of the complete graph of order n induced by S . If $\text{rank}(X) > 1$, then X is the direct sum of r non-overlapping edge cut-set matrices.*

In what follows we exploit characterizations of PSD $\{0, \pm 1\}$ -matrices to derive equivalent formulations of polytopes from the literature. Letchford [32] introduced and studied the sets

$$F_n := \left\{ (x, y) \in \mathbb{Z}^{n + \binom{n+1}{2}} : y_{ij} = x_i x_j \text{ for } 1 \leq i \leq j \leq n \right\},$$

and $\text{IQ}_n := \text{clconv}\{F_n\}$ for $n \in \mathbb{N}$. Here $\text{clconv}\{\cdot\}$ denotes the closure of the convex hull of F_n . The set IQ_n is not a polyhedron since it has a countably infinite number of faces. Such sets are known as quasi-polyhedra. It was proven in [32] that every point in F_n is an extreme point of IQ_n . Moreover, it is not difficult to verify that

$$F_n = \left\{ X^+ = \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} : X^+ \succeq \mathbf{0}, \text{rank}(X^+) = 1, x \in \mathbb{Z}^n \right\}.$$

Burer and Letchford [10] proved that minimizing a linear function over IQ_n is \mathcal{NP} -hard in the strong sense. If there exists $u \in \mathbb{Z}_+$ such that $-u\mathbf{1}_n \leq x \leq u\mathbf{1}_n$ for all $(x, y) \in F_n$, then one can define the following bounded sets:

$$F_n^u := \left\{ (x, y) \in \mathbb{Z}^{n+\binom{n+1}{2}} : y_{ij} = x_i x_j \text{ for } 1 \leq i \leq j \leq n, x_i \in \{-u, \dots, u\} \ i \in [n] \right\}$$

and $\text{IQ}_n^u := \text{conv}\{F_n^u\}$. The set IQ_n^u is a polytope. The sets IQ_n and IQ_n^u are studied in [8, 10, 21, 32]. For the case $u = 1$, we define

$$(2.4) \quad \mathcal{F}_n^1 := \left\{ X \in \{0, \pm 1\}^{n \times n} : \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq \mathbf{0}, \text{diag}(X) = |x| \right\}.$$

Note that, apart from the embedding, \mathcal{F}_n^1 coincides with F_n^1 . Elements of the set \mathcal{F}_n^1 are neither described by a nonconvex rank constraint nor by a product of variables. The cardinality of \mathcal{F}_n^1 is $1 + \sum_{k=1}^n \binom{n}{k} 2^{k-1}$, see [17] for the enumeration of its vertices. Let us consider the following two polytopes:

$$(2.5) \quad \begin{aligned} \mathcal{IQ}_n^1 &:= \text{conv}\{\mathcal{F}_n^1\}, \\ \mathcal{M}_n &:= \{X \in \mathcal{S}^n : (2.1) - (2.3), X_{ii} \leq 1 \text{ for } i \in [n]\}. \end{aligned}$$

It follows from Proposition 2.3 that \mathcal{IQ}_n^1 is the convex hull of all rank-one ternary PSD matrices. One can easily show that ternary PSD matrices of arbitrary rank are also contained in \mathcal{IQ}_n^1 . For example, any ternary PSD matrix X of rank 2 can be decomposed as follows: Since $X = x_1 x_1^\top + x_2 x_2^\top$ for some $x_1, x_2 \in \{0, \pm 1\}^n$, we define $S_i := \{j : (x_i)_j = 1\}$ and $T_i := \{i : (x_i)_j = -1\}$ for $i \in \{1, 2\}$. Then, we have

$$X = \frac{1}{2}(\mathbb{1}_{S_1 \cup S_2} - \mathbb{1}_{T_1 \cup T_2})(\mathbb{1}_{S_1 \cup S_2} - \mathbb{1}_{T_1 \cup T_2})^\top + \frac{1}{2}(\mathbb{1}_{S_1 \cup T_2} - \mathbb{1}_{S_2 \cup T_1})(\mathbb{1}_{S_1 \cup T_2} - \mathbb{1}_{S_2 \cup T_1})^\top$$

showing that $X \in \mathcal{IQ}_n^1$. A similar construction holds for higher rank matrices.

The set \mathcal{M}_n includes constraints (2.1)–(2.3), and additionally upper bounds on the diagonal elements. Note that without those upper bounds, the corresponding set would be unbounded. Intrigued by the well-known result that relates the metric polytope and the cut polytope, we study the relation between \mathcal{IQ}_n^1 and \mathcal{M}_n . Recall that the metric polytope is the set of all symmetric matrices with ones on the diagonal that satisfy the triangle inequalities (2.1)–(2.2), and the cut polytope is $\text{conv}\{X \in \mathcal{S}^n : X = yy^\top, y \in \{\pm 1\}^n\}$. The well-known result that relates these two polytopes is that the metric polytope coincides with the cut polytope for $n \in \{3, 4\}$, while the cut polytope is strictly contained in the metric polytope for $n \geq 5$, see e.g., [18].

Next, we show that $\mathcal{IQ}_n^1 \subseteq \mathcal{M}_n$ for $n = 3$. It is not difficult to verify that $\mathcal{IQ}_3^1 \subseteq \mathcal{M}_3$. One can, for example enumerate all 14 extreme points of \mathcal{F}_3^1 and verify that points from $\text{conv}\{\mathcal{F}_3^1\}$ are also in \mathcal{M}_3 . To show that the inclusion is strict, we consider the matrix $\frac{1}{3} \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & 1 \\ -1 & 1 & 3 \end{pmatrix} \in \mathcal{M}_3$, that is one of the vertices of \mathcal{M}_3 . This matrix is not positive semidefinite and it is therefore not in \mathcal{IQ}_3^1 .

Next, we consider an extended formulation of the polytope $\mathcal{I}Q_n^1$, i.e., we write $\mathcal{I}Q_n^1$ as a linear projection of a lifted polytope in a higher-dimensional space. To do so, we start from the observation that any ternary vector $x \in \{0, \pm 1\}^n$ can be decomposed as $x = x' - x''$, where $x', x'' \in \{0, 1\}^n$. This decomposition is unique if we additionally require that $x' + x'' \leq \mathbf{1}_n$. Exploiting the decomposition, the ternary rank-one PSD matrix xx^\top can be written as

$$(2.6) \quad xx^\top = (x' - x'')(x' - x'')^\top = x'(x')^\top + x''(x'')^\top - x'(x'')^\top - x''(x')^\top.$$

Now, consider the extended rank-one matrix Z that is defined as

$$(2.7) \quad Z = \begin{pmatrix} x' \\ x'' \end{pmatrix} \begin{pmatrix} x' \\ x'' \end{pmatrix}^\top = \begin{pmatrix} x'x'^\top & x'(x'')^\top \\ x''(x')^\top & x''(x'')^\top \end{pmatrix} =: \begin{pmatrix} Z^{11} & Z^{12} \\ Z^{21} & Z^{22} \end{pmatrix}.$$

Combining (2.6) and (2.7), it follows that $xx^\top = Z^{11} + Z^{22} - Z^{12} - Z^{21}$. The discrete set \mathcal{F}_n^1 can therefore be written as the linear projection of a discrete set in $(2n + 1)$ -dimensional space. Let the lifting \mathcal{L} be defined as

$$\mathcal{L}(z_1, z_2, Z^{11}, Z^{12}, Z^{21}, Z^{22}) := \begin{pmatrix} 1 & z_1^\top & z_2^\top \\ z_1 & Z^{11} & Z^{12} \\ z_2 & Z^{21} & Z^{22} \end{pmatrix}.$$

Then,

$$(2.8) \quad \mathcal{F}_n^1 = \left\{ Z^{11} + Z^{22} - Z^{12} - Z^{21} : \begin{array}{l} \mathcal{L}(z_1, z_2, Z^{11}, Z^{12}, Z^{21}, Z^{22}) \succeq \mathbf{0}, \\ \text{diag}(Z^{11}) = z_1, \text{diag}(Z^{22}) = z_2, \\ z_1 + z_2 \leq \mathbf{1}_n, z_1, z_2 \in \{0, 1\}^n \end{array} \right\}.$$

Observe that it is sufficient to enforce integrality on z_1, z_2 , as $\text{diag}(Z^{11}) = z_1$ and $\text{diag}(Z^{22}) = z_2$ together with the linear matrix inequality enforce integrality of the entire $(2n + 1)$ -dimensional matrix [28]. Moreover, as shown in [16], each extended matrix in the set on the right-hand side of (2.8) has rank-one. The description of \mathcal{F}_n^1 in (2.8) shows that any ternary rank-one PSD matrix is the projection of a binary rank-one PSD matrix in a higher-dimensional space. The following proposition connects their induced polytopes in a similar way.

PROPOSITION 2.7. *Let $\mathcal{P}_n^1 := \text{conv}\{X \in \{0, 1\}^{n \times n} : X \succeq \mathbf{0}, \text{rank}(X) = 1\}$, i.e., the convex hull of all binary rank-one PSD matrices. Then,*

$$(2.9) \quad \mathcal{I}Q_n^1 = \left\{ Z^{11} + Z^{22} - Z^{12} - Z^{21} : \begin{array}{l} \mathcal{L}(z_1, z_2, Z^{11}, Z^{12}, Z^{21}, Z^{22}) \in \mathcal{P}_{2n+1}^1 \\ \text{diag}(Z^{11}) = z_1, \text{diag}(Z^{22}) = z_2, \\ z_1 + z_2 \leq \mathbf{1}_n \end{array} \right\}.$$

Proof. Let $\mathcal{D}_n^1 := \{X \in \{0, 1\}^{n \times n} : X \succeq \mathbf{0}, \text{rank}(X) = 1\}$, i.e., the set of binary rank-one PSD matrices. Then, the set on the right-hand side of (2.8) can be written as

$$(2.10) \quad \mathcal{F}_n^1 = \left\{ Z^{11} + Z^{22} - Z^{12} - Z^{21} : \begin{array}{l} \mathcal{L}(z_1, z_2, Z^{11}, Z^{12}, Z^{21}, Z^{22}) \in \mathcal{D}_{2n+1}^1 \\ \text{diag}(Z^{11}) = z_1, \text{diag}(Z^{22}) = z_2, \\ z_1 + z_2 \leq \mathbf{1}_n \end{array} \right\},$$

Taking the convex hull on both sides of (2.10) yields

$$\mathcal{I}Q_n^1 = \text{conv} \left\{ Z^{11} + Z^{22} - Z^{12} - Z^{21} : \begin{array}{l} \mathcal{L}(z_1, z_2, Z^{11}, Z^{12}, Z^{21}, Z^{22}) \in \mathcal{D}_{2n+1}^1 \\ \text{diag}(Z^{11}) = z_1, \text{diag}(Z^{22}) = z_2, \\ z_1 + z_2 \leq \mathbf{1}_n \end{array} \right\}.$$

The latter set reduces to (2.9) due to the linearity of $\text{diag}(\cdot)$ and the fact that $\mathcal{P}_{2n+1}^1 = \text{conv}\{\mathcal{D}_{2n+1}^1\}$. \square

Proposition 2.7 extends any known representation of the convex hull of binary rank-one PSD matrices into a representation of $\mathcal{I}Q_n^1$. It is not difficult to verify that the convex hull of binary rank-one PSD matrices relates to the well-known Boolean quadric polytope [37], which is defined as $\mathcal{BQP}_n = \text{conv}\{(x, y) \in \{0, 1\}^{n+\binom{n}{2}} : y_{ij} = x_i x_j \text{ for } 1 \leq i < j \leq n\}$. Apart from the embedding, the set \mathcal{P}_n^1 coincides with \mathcal{BQP}_n , see also [16]. More specifically, if we define the symmetric vectorization operator $\text{svec}^+ : \mathcal{S}^{n+1} \rightarrow \mathbb{R}^{n+\binom{n}{2}}$, which maps a symmetric matrix to the vector containing its strictly upper triangular entries, i.e.,

$$\text{svec}^+ \left(\begin{pmatrix} 1 & z^\top \\ z & Z \end{pmatrix} \right) := (z, y) \text{ where } y_{ij} = Z_{ij} \text{ for } 1 \leq i < j \leq n,$$

it follows that $\text{svec}^+(\mathcal{P}_n^1) = \mathcal{BQP}_n$. Now, it follows from Proposition 2.7 that

$$\mathcal{I}Q_n^1 = \left\{ Z^{11} + Z^{22} - Z^{12} - Z^{21} : \begin{array}{l} \text{svec}^+(\mathcal{L}(z_1, z_2, Z^{11}, Z^{12}, Z^{21}, Z^{22})) \in \mathcal{BQP}_{2n+1}^1 \\ \text{diag}(Z^{11}) = z_1, \text{diag}(Z^{22}) = z_2, z_1 + z_2 \leq \mathbf{1}_n \end{array} \right\}.$$

Alternatively, it is also known that \mathcal{P}_n^1 as defined in Proposition 2.7 possesses a set-completely positive reformulation [34] in terms of the cone of set-completely positive matrices $\mathcal{SCP}_n = \text{conv}\{xx^\top : x \in \mathbb{R}_+^n, x_1 \geq x_i \text{ for all } i = 2, \dots, n\}$. Combining a result from [34], see also [16, Theorem 5], with Proposition 2.7, it follows that

$$\mathcal{I}Q_n^1 = \left\{ Z^{11} + Z^{22} - Z^{12} - Z^{21} : \begin{array}{l} \mathcal{L}(z_1, z_2, Z^{11}, Z^{12}, Z^{21}, Z^{22}) \in \mathcal{SCP}_{2n+1}^1 \\ \text{diag}(Z^{11}) = z_1, \text{diag}(Z^{22}) = z_2, z_1 + z_2 \leq \mathbf{1}_n \end{array} \right\}.$$

3. ISDP formulation for the ternary quadratic problem. We first introduce the quadratic optimization problem of interest and then reformulate it as an integer semidefinite program (ISDP), using the results from Section 2.

Let $Q = (q_{ij}) \in \mathcal{S}^n$, $c \in \mathbb{R}^n$, and $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$ for all $i \in [p]$, where $p \in \mathbb{N}$. We consider the ternary quadratic problem (TQP):

$$(3.1) \quad \begin{array}{ll} \min_{x \in \{0, \pm 1\}^n} & x^\top Q x + c^\top x \\ \text{s.t.} & a_i^\top x = b_i \quad i \in [p]. \end{array}$$

Observe that if Q is indefinite, the resulting TQP is nonconvex. Although the TQP (3.1) has no quadratic constraints, the approach described here applies also if we add quadratic constraints. The quadratic objective in (3.1) can be written as $\langle Q, X \rangle + c^\top x$, where we substitute X for xx^\top . This yields the following exact reformulation of (3.1):

$$(3.2) \quad \begin{array}{ll} \min_{x \in \{0, \pm 1\}^n} & \langle Q, X \rangle + c^\top x \\ \text{s.t.} & a_i^\top x = b_i \quad i \in [p] \\ & Y = \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq \mathbf{0}, \quad \text{rank}(Y) = 1. \end{array}$$

The above problem contains a nonconvex rank-one constraint. However, one can utilize an alternative notion of exactness in terms of integrality by exploiting Proposition 2.3, and obtain the following ternary semidefinite program:

$$\begin{aligned}
(3.3) \quad & \min \langle Q, X \rangle + c^\top x \\
& \text{s.t. } a_i^\top x = b_i \quad i \in [p] \\
& Y = \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq \mathbf{0}, \quad \text{diag}(X) = |x| \\
& Y \in \{0, \pm 1\}^{(n+1) \times (n+1)}.
\end{aligned}$$

One can also reformulate (3.2) as a binary semidefinite program by exploiting the transformation of ternary matrices into binary matrices, see (2.8). However, such decomposition increases the size of the problem and makes it more difficult to solve. The next result follows from the previous discussion.

THEOREM 3.1. *Optimization problems (3.1) and (3.3) are equivalent.*

One can relax some of the integrality conditions on Y in (3.3) and maintain exactness. Namely, we can request only $x \in \{0, \pm 1\}$ instead of $Y \in \{0, \pm 1\}^{(n+1) \times (n+1)}$ in (3.3). To see this, we consider the three-by-three principal submatrices of Y indexed by $\mathcal{I} \times \mathcal{I}$, where $\mathcal{I} = [0, i, j]^\top$, for some $i, j \in [n], i < j$. Here, 0 represents the first row/column of Y . Then, $\det(Y_{\mathcal{I}}) = -(x_i x_j - X_{ij})^2 \geq 0$, from where it follows $X_{ij} = x_i x_j$. Here, we exploit the fact that $|x_i| = x_i^2$ for $x_i \in \{0, \pm 1\}$. A more compact formulation of (3.3) can be obtained by using the following result.

LEMMA 3.2. *Let $Y = \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq \mathbf{0}$ such that $\text{diag}(X) = |x|$ and $x \in \{0, \pm 1\}$, and let $S = \sum_{i=1}^p \begin{pmatrix} -b_i \\ a_i \end{pmatrix} \begin{pmatrix} -b_i \\ a_i \end{pmatrix}^\top$. Then, $a_i^\top x = b_i \forall i \in [p]$ if and only if $\langle S, Y \rangle = 0$.*

Proof. It follows from Proposition 2.3 and the previous discussion, that $Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^\top$. Therefore, $\sum_{i=1}^p \left\langle \begin{pmatrix} -b_i \\ a_i \end{pmatrix} \begin{pmatrix} -b_i \\ a_i \end{pmatrix}^\top, \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^\top \right\rangle = \sum_{i=1}^p (a_i^\top x - b_i)^2$, from where the result follows. \square

4. SDP relaxation and valid inequalities. In this section, we first present a basic SDP relaxation for the TQP (3.1), and then introduce several classes of valid inequalities that can be imposed to strengthen it. We derive those inequalities by using various techniques, including the RLT and Chvátal-Gomory procedure.

We derive a semidefinite relaxation for the TQP (3.1) from the ISDP reformulation (3.3) by relaxing integrality constraints and the constraint $\text{diag}(X) = |x|$. The resulting basic SDP relaxation is given below.

$$\begin{aligned}
(4.1) \quad & \min \langle Q, X \rangle + c^\top x \\
& \text{s.t. } a_i^\top x = b_i \quad i \in [p] \\
& \text{diag}(X) \geq x, \quad \text{diag}(X) \geq -x \\
& \text{diag}(X) \leq \mathbf{1}_n, \quad \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq \mathbf{0}.
\end{aligned}$$

We add constraints that impose upper bounds on the diagonal of the matrix variable to preserve the boundedness of the relaxation. In particular, one can add to (4.1) the following constraints

$$(4.2) \quad \langle a_i a_i^\top, X \rangle = b_i^2 \quad \text{for all } i \in [p].$$

Note that by adding those constraints the corresponding linear constraints are, in general, not redundant. Further, one can easily verify that for (x, X) such that $X - xx^\top \succeq \mathbf{0}$, $\text{diag}(X) \geq x$, $\text{diag}(X) \geq -x$ and $\text{diag}(X) \leq \mathbf{1}$ it follows that $-1 \leq X_{ij} \leq 1$ and $-1 \leq x_i \leq 1$ for all $i, j \in [n]$.

REMARK 2. *Buchheim and Wiegele [9] consider a nonconvex quadratic mixed-integer optimization problem and derive its exact reformulation in matrix form by using a rank-one constraint. Their exact matrix reformulation, when restricted to the TQP (3.1), differs from (3.2) in the constraint $x \in \{0, \pm 1\}^n$, which is replaced by $(x_i, X_{ii}) \in P(\{0, \pm 1\}) = \text{conv}\{(x, x^2) : x \in \{0, \pm 1\}\}$ for all $i \in [n]$. The polytope $P(\{0, \pm 1\})$ is completely described by the following lower bounding facets*

$$(4.3) \quad X_{ii} \geq x_i \quad \text{and} \quad X_{ii} \geq -x_i \quad i \in [n],$$

and the upper bounding facet $X_{ii} \leq 1$ for $i \in [n]$. Therefore, the SDP relaxation of the TQP (3.1) from [9], obtained by omitting the rank-one constraint, corresponds to (4.1).

In what follows, we introduce valid inequalities, which together with (4.2), can be added to the SDP relaxation (4.1). The most natural choice is to add to (4.1) the inequalities (2.1)–(2.3) from Theorem 2.6. The inequalities (2.3) are known in the literature as the *pair inequalities*, see e.g., [40]. These inequalities are non-standard split inequalities, and we show how to derive them later.

The (generalized) triangle inequalities. The inequalities (2.1)–(2.2) are known in the literature as the triangle inequalities. These inequalities are also Chvátal-Gomory cuts for the ISDP (3.3), as we will show below. The Chvátal-Gomory (CG) cuts form a well-known family of cutting planes for integer programs, initially designed for integer linear programs by Chvátal [13] and Gomory [23]. Cuts for integer conic problems and ISDP have been investigated in [12] and in [15, 14], respectively.

One can show that the first set of inequalities in (2.1) are Chvátal-Gomory cuts by taking the vector $v \in \mathbb{Z}^{n+1}$ (indexed from zero) which has all zero entries except at positions i, j, k where $v_i = v_j = v_k = 1$. Then, for any Y feasible for (3.3) we have $v^\top Y v \geq 0$ since $Y, vv^\top \succeq \mathbf{0}$. Thus, $v^\top Y v = 2Y_{ij} + 2Y_{ik} + 2Y_{jk} \geq -Y_{ii} - Y_{jj} - Y_{kk} \geq -3$, from where it follows that $Y_{ij} + Y_{ik} + Y_{jk} \geq \lceil -1.5 \rceil = -1$ where we exploited the fact that the left-hand side of the inequality is an integer number. Similarly, one can derive other triangle inequalities from (2.1)–(2.2) as Chvátal-Gomory cuts.

Interestingly, triangle inequalities (2.1)–(2.2) can be generalized for any $S \subseteq [n]$ such that $|S|$ is an *odd* number. In particular, let $S \subseteq [n]$ and $|S|$ odd, then we introduce the following valid inequalities

$$(4.4) \quad \sum_{i, j \in S, i < j} v_i v_j X_{ij} \geq \left\lceil -\frac{1}{2}|S| \right\rceil,$$

where $v_i, v_j \in \{\pm 1\}$. There are $2^{|S|-1}$ constraints of this type for a given S . To prove that (4.4) are Chvátal-Gomory cuts for the ISDP (3.3), one can take the vector $v \in \mathbb{Z}^{n+1}$ (indexed from zero) with all entries equal to zero except for 1 or -1 at positions corresponding to the elements of S . Then, one can proceed similarly as for deriving the triangle inequalities. Thus, for $|S| = 5$ (resp. $|S| = 7$) we obtain pentagonal (resp. heptagonal) inequalities for the TQP, and so on. Note that for subsets with an even number of elements, the corresponding inequalities are not binding.

Although the triangle, pentagonal and heptagonal inequalities for the max-cut problem can be obtained as hypermetric inequalities [18], the inequalities (4.4) for the TQP cannot be derived in that way.

The RLT inequalities. The SDP relaxation (4.1) can be strengthened by adding valid inequalities obtained by using the reformulation linearization technique [45]. In particular, we derive the following RLT inequalities considering the bounds on the original variables:

$$(4.5) \quad \text{From } x_i \geq -1 \text{ and } x_j \geq -1 \text{ for } i, j \in [n], i < j : \quad X_{ij} + x_i + x_j \geq -1.$$

$$(4.6) \quad \text{From } x_i \leq 1 \text{ and } x_j \leq 1 \text{ for } i, j \in [n], i < j : \quad X_{ij} - x_i - x_j \geq -1.$$

$$(4.7) \quad \text{From } x_i \geq -1 \text{ and } x_j \leq 1 \text{ for } i, j \in [n], i < j : \quad -X_{ij} + x_i - x_j \geq -1.$$

$$(4.8) \quad \text{From } x_i \leq 1 \text{ and } x_j \geq -1 \text{ for } i, j \in [n], i < j : \quad -X_{ij} - x_i + x_j \geq -1.$$

Inequalities of this kind were originally introduced by McCormick [35]. It is interesting to note that the RLT inequalities (4.5)–(4.8) may be derived by applying the Chvátal-Gomory procedure with appropriate dual multipliers. For example, to derive (4.5) one can take $v \in \mathbb{Z}^{n+1}$ (indexed from zero) with all entries equal to zero except at positions $0, i, j$ where $v_0 = v_i = v_j = 1$ for $i < j$. Then, from $\langle v v^\top, Y \rangle \geq 0$, where Y is feasible for (3.3), and with appropriate rounding, (4.5) follows. Similarly, one can derive (4.6)–(4.8) as Chvátal-Gomory cuts. Thus, the RLT-inequalities (4.5)–(4.8) are also Chvátal-Gomory cuts. Note that inequalities (4.5)–(4.8) can be generalized to any $S \subseteq [n]$ such that $|S|$ is even.

Split inequalities. Split inequalities are introduced by Letchford [32], and further studied in [10, 8]. A split inequality corresponds to split disjunction of the form

$$(4.9) \quad (w^\top x \leq s) \vee (w^\top x \geq s + 1),$$

where $w \in \mathbb{Z}^n$ and $s \in \mathbb{Z}$. The above disjunction imposes that points in the feasible region lie either in the halfspace $w^\top x \leq s$ or in the halfspace $w^\top x \geq s + 1$. The split disjunction (4.9) can be reformulated as the quadratic inequality $(w^\top x - s)(w^\top x - s - 1) \geq 0$. In the lifted space, that quadratic inequality is linearized and takes the form:

$$(4.10) \quad \left\langle v(v + e_0)^\top, \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \right\rangle \geq 0 \quad \forall v \in \mathbb{Z}^{n+1},$$

where $v = (-s - 1, w^\top)^\top \in \mathbb{Z}^{n+1}$, e_0 is the unit vector indexed from zero. Buchheim and Traversi [8] tested the quality of split inequalities on problems with up to 50 variables and concluded that these cuts are strong. The main obstacle in applying these cuts is identifying the violated ones. Moreover, it is unclear whether the separation of split inequalities is an \mathcal{NP} -hard problem or not.

In this work, we exploit two types of split inequalities, which we list below. It is not difficult to verify that the inequalities (4.3) are split inequalities. To verify this claim for $X_{ii} \geq x_i$ (resp. $X_{ii} \geq -x_i$), take $v \in \mathbb{Z}^{n+1}$ (indexed from zero) that has all zero entries except $v_0 = -1$ and $v_i = 1$ (resp. $v_i = -1$) for some $i \in [n]$. Then, the inequalities (4.3) follow by using the vectors in (4.10).

A generalization of inequalities (4.3) to split inequalities with two indices $i, j \in [n]$, $1 \leq i < j \leq n$ is given below:

$$(4.11) \quad X_{ii} + X_{jj} + 2X_{ij} + x_i + x_j \geq 0, \quad X_{ii} + X_{jj} + 2X_{ij} - x_i - x_j \geq 0$$

$$(4.12) \quad X_{ii} + X_{jj} - 2X_{ij} + x_i - x_j \geq 0, \quad X_{ii} + X_{jj} - 2X_{ij} - x_i + x_j \geq 0.$$

We derive the first split inequality in (4.11) by taking a binary vector $v \in \{0, 1\}^{n+1}$ (indexed from zero) with all entries equal to zero except at positions i and j ($i < j$), and applying it in (4.10). The remaining inequalities can be derived in a similar way. Clearly, this approach can be extended to split inequalities with three or more indices. However, the number of associated valid inequalities increases exponentially. That is, for a given $k \leq n$ indices, there are 2^k split inequalities of the above type. We note that we tested additional types of split inequalities, but the two mentioned above were the most effective in our numerical experiments.

Non-standard split inequalities. Inequalities of the form $\langle vw^\top, Y \rangle \geq 0$ where w, v are vectors of appropriate size are introduced in [32]. Non-standard split inequalities are further studied in [10, 8]. The following result, which holds for the polytope \mathcal{IQ}_n^1 , see (2.5), was established in [8].

LEMMA 4.1 ([8]). *For any vector $v \in \mathbb{Z}^{n+1}$ with $v_0 = 0$, the non-standard split inequalities $\langle v(v + e_j)^\top, \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \rangle \geq 0$, for $j \in [n]$, are valid for $X \in \mathcal{IQ}_n^1$.*

One can prove that the inequalities in (2.3) are non-standard split inequalities. Namely, for the first set of inequalities in (2.3) (resp. the second set in (2.3)), take $v \in \mathbb{Z}^{n+1}$ (indexed from zero) with all entries zero except at position $i \in [n]$, where $v_i = -1$ (resp. $v_i = 1$), and the j^{th} unit vector $e_j \in \mathbb{Z}^{n+1}$ (indexed from zero, $i \neq j$), and apply Lemma 4.1.

We used the software SageMath [46] to find the complete description of \mathcal{IQ}_3^1 . The polytope \mathcal{IQ}_3^1 is described by the triangle inequalities (2.1)–(2.2), bound constraints $X_{ii} \leq 1$ for $i \in [3]$ (see also Remark 2), non-standard split inequalities of the form (2.3), and 12 additional non-standard split inequalities. In total, there are 31 facet-defining inequalities of \mathcal{IQ}_3^1 . All these inequalities can be imposed on any principal submatrix X_J of X from (4.1), where $J \subseteq [n]$, $|J| = 3$, to strengthen the SDP relaxation. Preliminary numerical results show that it is most beneficial adding the inequalities (2.3).

5. Special cases of the TQP.

5.1. The quadratic unconstrained ternary optimization problem. In this section, we study the QUTO and its relation to the max-cut problem.

The QUTO is given as follows:

$$(5.1) \quad \begin{aligned} \min \quad & \langle Q, X \rangle + c^\top x \\ \text{s.t.} \quad & \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq \mathbf{0}, \quad \text{diag}(X) = |x| \\ & X \in \{0, \pm 1\}^{(n+1) \times (n+1)}. \end{aligned}$$

The well-known max-cut problem is a special case of the QUTO (5.1), in which the variables are restricted to take values in $\{\pm 1\}$. In the sequel, we show that the QUTO also reduces to the max-cut problem for a particular quadratic objective.

THEOREM 5.1. *Consider the QUTO (5.1). If $q_{ii} \leq 0$ for some $i \in [n]$, then there exists an optimal solution to this problem with $x_i \in \{\pm 1\}$.*

Proof. Let i be such that $q_{ii} \leq 0$. Now, suppose x^* is a feasible solution to the unconstrained problem (3.1) with $x_i^* = 0$. We define the sets $S := \{j : x_j^* = 1\}$

and $T := \{j : x_j^* = -1\}$ and let $f(x^*) = (x^*)^\top Q x^* + c^\top x^*$. Moreover, we define $\beta = 2 \sum_{j \in S} q_{ij} - 2 \sum_{j \in T} q_{ij} + c_i$.

Now, we construct another solution x' from x^* such that $x'_i = 1$ and $x'_j = x_j^*$ for all $j \neq i$. The objective value of this solution equals $f(x') = f(x^*) + q_{ii} + \beta$. Alternatively, we construct x'' from x^* such that $x''_i = -1$ and $x''_j = x_j^*$ for all $j \neq i$. Then, $f(x'') = f(x^*) + q_{ii} - \beta$. Since $q_{ii} \leq 0$, it follows that either $f(x') \leq f(x^*)$ or $f(x'') \leq f(x^*)$. Restricting the i th element of x to an element in $\{\pm 1\}$ will therefore never increase the objective value. This holds in particular for the optimal solution. \square

The result of Theorem 5.1 is exploited in the following corollary, which states that the QUTO is equivalent to the max-cut problem if $\text{diag}(Q) \leq \mathbf{0}_n$.

COROLLARY 5.2. *Suppose $q_{ii} \leq 0$ for all $i \in [n]$. Then, the QUTO (5.1) is equivalent to the max-cut problem.*

Proof. It follows from Theorem 5.1 that under the given conditions we can always find an optimal solution x in the set $\{\pm 1\}^n$. Therefore, the problem (5.1) is equivalent to the minimization version of the quadratic unconstrained binary optimization problem (QUBO) with variables in $\{\pm 1\}$, also known as the Ising QUBO, see e.g., [41]. The Ising QUBO, both in its minimization and maximization version, is equivalent to the max-cut problem, see e.g., [41, Section 1.4.2]. \square

Our basic SDP relaxation for the QUTO is as follows:

$$(5.2) \quad \begin{aligned} \min \quad & \langle Q, X \rangle + c^\top x \\ \text{s.t.} \quad & X_{ii} = 1, \quad i \in N \\ & \text{diag}(X) \geq x, \quad \text{diag}(X) \geq -x \\ & \text{diag}(X) \leq \mathbf{1}_n, \quad \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq \mathbf{0}, \end{aligned}$$

where $N = \{i : q_{ii} \leq 0\}$. One can strengthen the SDP relaxation (5.2) by adding valid inequalities introduced in Section 4.

5.2. Ternary quadratic problem with one linear constraint. We study a variant of the TQP (3.1) with a single linear constraint, which we refer to as TQP-Linear. In particular, we present our basic SDP relaxation for the TQP-Linear and derive its strictly feasible formulation.

The TQP in which a linear constraint enforces that the sum of the variables equals zero, is used in the design of turbomachines, see e.g., [38]. Inspired by this application, we consider the TQP (3.1) in which $a_1 = \mathbf{1}_n$, $b_1 = 0$ and $p = 1$, thereby resulting in the following TQP-Linear:

$$(5.3) \quad \begin{aligned} \min \quad & \langle Q, X \rangle + c^\top x \\ \text{s.t.} \quad & \mathbf{1}_n^\top x = 0, \quad \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq \mathbf{0}, \quad \text{diag}(X) = |x| \\ & X \in \{0, \pm 1\}^{(n+1) \times (n+1)}. \end{aligned}$$

Our basic SDP relaxation for the TQP-Linear is as follows:

$$(5.4) \quad \begin{aligned} \min \quad & \langle Q, X \rangle + c^\top x \\ \text{s.t.} \quad & \langle \mathbf{J}, X \rangle = 0, \quad \text{diag}(X) \geq x, \quad \text{diag}(X) \geq -x \\ & \text{diag}(X) \leq \mathbf{1}_n, \quad \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq \mathbf{0}. \end{aligned}$$

Here we include the squared representation of the linear constraint, see also (4.2). Note that the SDP relaxation (5.4) does not include the constraint $x^\top \mathbf{1}_n = 0$, as it is redundant in the presence of the constraint $\langle \mathbf{J}, X \rangle = 0$.

LEMMA 5.3. *Let (X, x) satisfy $X - xx^\top \succeq \mathbf{0}$ and $\langle \mathbf{J}, X \rangle = 0$. Then, $x^\top \mathbf{1}_n = 0$.*

Proof. Let $Z := X - xx^\top$. Then, $0 = \mathbf{1}_n^\top X \mathbf{1}_n = \mathbf{1}_n^\top (Z + xx^\top) \mathbf{1}_n \Rightarrow (x^\top \mathbf{1}_n)^2 = -\mathbf{1}_n^\top Z \mathbf{1}_n \leq 0$, since $Z \succeq \mathbf{0}$. Therefore, $x^\top \mathbf{1}_n = 0$. \square

One can verify that $v = (0, \mathbf{1}_n^\top)^\top$ is an eigenvector corresponding to the zero eigenvalue of any matrix that is feasible for (5.4). Therefore, the SDP relaxation (5.4) has no Slater feasible points. We apply facial reduction to obtain an equivalent SDP relaxation that is Slater feasible. For an introduction to facial reduction, we refer the interested reader to [20] and the references therein.

Let $\mathcal{R} := \{x \in \mathbb{R}^{n+1} : v^\top x = 0\}$ denote the orthogonal complement of v and let

$$(5.5) \quad F_{\mathcal{R}} := \{Y \in \mathcal{S}_+^{n+1} : \text{Col}(Y) \subseteq \mathcal{R}\}.$$

It follows from Lemma 5.3 that the feasible set of (5.4) is contained in $F_{\mathcal{R}}$. It is well-known that sets of the form $F_{\mathcal{R}}$ are faces of \mathcal{S}_+^{n+1} [20]. Now, $F_{\mathcal{R}}$ can be expressed as $F_{\mathcal{R}} = W \mathcal{S}_+^n W^\top$, where $W \in \mathbb{R}^{(n+1) \times n}$ is such that its columns form a basis of \mathcal{R} . A possible and sparse choice for W is the matrix

$$(5.6) \quad W = \begin{pmatrix} 1 & 0 & \mathbf{0}_{n-1}^\top \\ \mathbf{0}_{n-1} & \mathbf{1}_{n-1} & -\mathbf{I}_{n-1} \end{pmatrix}^\top.$$

We rewrite (5.4) by replacing $\begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix}$ by WZW^\top , where $Z \in \mathcal{S}_+^n$. This yields:

$$(5.7) \quad \begin{aligned} \min \quad & \left\langle \begin{pmatrix} 0 & \frac{1}{2}c^\top \\ \frac{1}{2}c & Q \end{pmatrix}, WZW^\top \right\rangle \\ \text{s.t.} \quad & \text{diag}(WZW^\top) \geq WZW^\top \mathbf{e}_1, \quad \text{diag}(WZW^\top) \geq -WZW^\top \mathbf{e}_1 \\ & \text{diag}(WZW^\top) \leq \mathbf{1}_n, \quad \mathbf{e}_1^\top WZW^\top \mathbf{e}_1 = 1, \quad Z \succeq \mathbf{0}, \end{aligned}$$

which is equivalent to (5.4). Observe that all constraints that are present in (5.4) are explicitly present in (5.7), except for the constraint $\langle \mathbf{J}_n, X \rangle = 0$. This constraint, which after substitution is equivalent to $\langle \mathbf{J}_n, (\mathbf{0}_n \ \mathbf{I}_n) WZW^\top (\mathbf{0}_n \ \mathbf{I}_n)^\top \rangle = 0$, becomes redundant in (5.7) due to the construction of W . We now state the following result regarding (5.7).

LEMMA 5.4. *The SDP (5.7) is strictly feasible and equivalent to (5.4).*

Proof. The equivalence between the two programs follows from the construction of (5.7) and the discussion above. To show that (5.7) is strictly feasible, let

$$\hat{Z} := \frac{1}{n} \mathbf{e}_1 \mathbf{e}_1^\top + \frac{1}{n} \sum_{k=2}^n (\mathbf{e}_1 + \mathbf{e}_k)(\mathbf{e}_1 + \mathbf{e}_k)^\top = \begin{pmatrix} 1 & \frac{1}{n} \mathbf{1}_{n-1}^\top \\ \frac{1}{n} \mathbf{1}_{n-1} & \frac{1}{n} \mathbf{I}_{n-1} \end{pmatrix}.$$

Using W as in (5.6), we define $\hat{Y} := W \hat{Z} W^\top \in \mathcal{S}^{n+1}$. By direct verification, it follows that $\text{diag}(\hat{Y}) \geq \hat{Y} \mathbf{e}_1$, $\text{diag}(\hat{Y}) \geq -\hat{Y} \mathbf{e}_1$, $\text{diag}(\hat{Y}) \leq \mathbf{1}_{n+1}$ and $\mathbf{e}_1^\top \hat{Y} \mathbf{e}_1 = 1$. Moreover, since \hat{Z} is defined as a convex combination of rank-one PSD matrices, we know $\hat{Z} \succeq \mathbf{0}$. Thus, \hat{Z} is feasible for (5.7). To show that \hat{Z} is contained in the interior of \mathcal{S}_+^n , observe that $M := (\mathbf{e}_1 \ \mathbf{e}_1 + \mathbf{e}_2 \ \dots \ \mathbf{e}_1 + \mathbf{e}_n)$ has rank n . Since $\hat{Z} = \frac{1}{n} M M^\top$, it follows that \hat{Z} has full rank, implying that $\hat{Z} \succ \mathbf{0}$. \square

5.3. Ternary quadratic problem with a ratio objective. We study the quadratic program with a ratio objective and ternary variables, which we refer to as TQP-Ratio. We propose two approaches for solving the TQP-Ratio. The first approach is based on a version of Dinkelbach’s algorithm for fractional programming [19] that relies on a B&B algorithm for the QUTO, see Section 5.1. The second approach exploits the convexification technique from [27] to derive an SDP relaxation for the TQP-Ratio, that is used within our B&B algorithm for solving instances of the TQP-Ratio to optimality.

We consider the following fractional quadratic optimization problem:

$$(5.8) \quad \min_{x \in \{0, \pm 1\}^n} \frac{f(x)}{g(x)},$$

where $f(x) := x^\top Ax + a^\top x + a_0$, and $g(x) := x^\top Bx + b^\top x + b_0 > 0$ for all $x \in \{0, \pm 1\}^n$. Ternary quadratic problems with a ratio objective appear in many applications, see e.g., [29, 47]. In the sequel, we provide two approaches for solving (5.8) to optimality.

Parametric optimization. Dinkelbach [19] introduced a framework for solving convex fractional programs. Dinkelbach’s approach was later extended to solve general fractional programs [42]. To the best of our knowledge, we are the first to exploit Dinkelbach’s approach for ternary quadratic problems. To this end, we introduce the following parameterized optimization problem w.r.t. $\lambda \in \mathbb{R}$ associated with (5.8):

$$(5.9) \quad \Phi(\lambda) := \min_{x \in \{0, \pm 1\}^n} (f(x) - \lambda g(x)).$$

It is straightforward to verify that the function $\Phi(\cdot)$ is concave and strictly monotonically decreasing. Moreover, for any $\lambda = f(x)/g(x)$ with $x \in \{0, \pm 1\}^n$, it holds that $\Phi(\lambda) \leq 0$. As shown by Dinkelbach [19], we have

$$\lambda^* = \min_{x \in \{0, \pm 1\}^n} \frac{f(x)}{g(x)} \iff \Phi(\lambda^*) = 0,$$

where the minimum in $\Phi(\lambda^*)$ is attained at $x^* \in \arg \min_{x \in \{0, \pm 1\}^n} f(x)/g(x)$. In other words, to solve the problem (5.8), it suffices to find the unique root of the parametrized function $\Phi(\cdot)$. Due to the properties of the function $\Phi(\cdot)$, this root can be found by iteratively evaluating the function $\Phi(\lambda)$ at an incumbent point $\lambda = f(x)/g(x)$ for some $x \in \{0, \pm 1\}^n$, which is updated accordingly if $\Phi(\lambda) < 0$.

For a given λ , each main iteration of Dinkelbach’s algorithm [42] solves the parametrized ternary optimization problem (5.9), that can be equivalently formulated as follows:

$$(5.10) \quad \begin{aligned} \Phi(\lambda) = \min \quad & \left\langle \begin{pmatrix} a_0 - \lambda b_0 & \frac{1}{2}(a^\top - \lambda b^\top) \\ \frac{1}{2}(a - \lambda b) & A - \lambda B \end{pmatrix}, \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \right\rangle \\ \text{s.t.} \quad & \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq \mathbf{0}, \quad \text{diag}(X) = |x|, \quad X \in \{0, \pm 1\}^{n \times n}. \end{aligned}$$

The optimization problem (5.10) is the QUTO whose basic SDP relaxation is (5.2). Schaible [44] proved superlinear convergence of Dinkelbach’s algorithm. A similar convergence result can be established for a version of Dinkelbach’s algorithm applied to the TQP-Ratio problem. Our version of Dinkelbach’s algorithm iteratively updates the parameter λ by solving a sequence of parametric subproblems (5.10), see

Algorithm 5.1. Observe that the starting solution x is computed using a Variable Neighborhood Search heuristic, which is introduced in Section 6.1.

Algorithm 5.1 Dinkelbach's algorithm for the TQP-Ratio

```

1: input  $(A, a, a_0), (B, b, b_0), \epsilon > 0$ 
2:  $x \leftarrow$  VNS-Heuristic (Section 6.1),  $\lambda \leftarrow f(x)/g(x)$ 
3: while true do
4:    $x \in \arg \min_{x \in \{0, \pm 1\}^n} (f(x) - \lambda g(x))$ 
5:   if  $|f(x) - \lambda g(x)| < \epsilon$  then
6:     return  $x^* \leftarrow x, \lambda^* \leftarrow f(x^*)/g(x^*)$ 
7:   end if
8:    $\lambda \leftarrow f(x)/g(x)$ 
9: end while
10: return  $x^*, \lambda^*$ 

```

An SDP relaxation for TQP-Ratio with no fractional terms. We follow the approach of He et al., [27] to reformulate (5.8) as an equivalent optimization problem without fractional terms. Define the rescaling

$$(5.11) \quad \rho := g(x)^{-1}, \quad y := \rho x, \quad Y := \rho x x^\top.$$

Then for any $x \in \{0, \pm 1\}^n$ we have the identities $f(x)/g(x) = \langle A, Y \rangle + a^\top y + a_0 \rho$ and $\langle B, Y \rangle + b^\top y + b_0 \rho = 1$. It follows from [27] that (5.8) can be equivalently reformulated as the following optimization problem

$$(5.12) \quad \begin{aligned} \min_{\rho, y, Y} \quad & \langle A, Y \rangle + a^\top y + a_0 \rho \\ \text{s.t.} \quad & \langle B, Y \rangle + b^\top y + b_0 \rho = 1, \\ & \rho \geq 0, |y| = \text{diag}(Y), Y \in \rho \text{conv} \{ \mathcal{F}_n^1 \}, \end{aligned}$$

where the set \mathcal{F}_n^1 is defined in (2.4), see also (2.5). To derive an SDP relaxation for (5.12), we introduce the positive semidefinite matrix variable $Z := \begin{pmatrix} \rho & y^\top \\ y & Y \end{pmatrix} \succeq \mathbf{0}$. Moreover, for $x_i \in \{0, \pm 1\}$ we have the implications $Y_{ii} = \rho x_i^2 \in \{0, \rho\}$ and $|y_i| = \rho x_i^2 = Y_{ii}$. This motivates the linear inequalities $0 \leq Y_{ii} \leq \rho$ and $-Y_{ii} \leq y_i \leq Y_{ii}$ for all $i \in [n]$. The previous discussion yields the following SDP relaxation of (5.12):

$$(5.13) \quad \begin{aligned} \min \quad & \langle A, Y \rangle + a^\top y + a_0 \rho \\ \text{s.t.} \quad & \langle B, Y \rangle + b^\top y + b_0 \rho = 1 \\ & \text{diag}(Y) \geq y, \quad \text{diag}(Y) \geq -y \\ & \rho \geq 0, \quad \text{diag}(Y) \leq \rho \mathbf{1}_n, \quad \begin{pmatrix} \rho & y^\top \\ y & Y \end{pmatrix} \succeq \mathbf{0}. \end{aligned}$$

In order to strengthen the above SDP relaxation, one can add scaled versions of the cuts discussed in Section 4. Rescaling is performed with respect to ρ , see (5.11). Our preliminary results suggest that adding the following constraints to the SDP relaxation (5.13) is most beneficial for strengthening it: Scaled triangle inequalities

$$(5.14) \quad \begin{aligned} Y_{ij} + Y_{ik} + Y_{jk} &\geq -\rho, & -Y_{ij} + Y_{ik} - Y_{jk} &\geq -\rho, \\ Y_{ij} - Y_{ik} - Y_{jk} &\geq -\rho, & -Y_{ij} - Y_{ik} + Y_{jk} &\geq -\rho, \end{aligned}$$

for $1 \leq i < j < k \leq n$, see also (2.1)–(2.2); scaled RLT inequalities

$$(5.15) \quad \begin{aligned} Y_{ij} + y_i + y_j &\geq -\rho, & Y_{ij} - y_i - y_j &\geq -\rho, \\ -Y_{ij} + y_i - y_j &\geq -\rho, & -Y_{ij} - y_i + y_j &\geq -\rho, \end{aligned}$$

for $i, j \in [n]$, $i < j$, see also (4.5)–(4.8); split inequalities of the form (4.11)–(4.12), and non-standard split inequalities of the form (2.3). Split inequalities for the SDP relaxation (5.13) are obtained from the inequalities (4.11)–(4.12) by replacing X by Y , and x by y , and similar for non-standard split inequalities.

6. SDP-based branch-and-bound algorithm for the TQP. In this section we incorporate our basic SDP relaxation and the derived cutting planes from Section 4 in a specialized branch-and-bound algorithm to solve TQP instances. We tailor this algorithm to the three special cases discussed in Section 5. We refer to the resulting method as **SDP-B&B**.

SDP-B&B systematically explores a tree of subproblems, starting from the root node. At each node of the tree, a lower bound is obtained by strengthening the basic SDP relaxation (4.1) by a cutting-plane procedure. For the special cases of the QUTO and the TQP-Linear, this relaxation boils down to solving (5.2) and (5.4), respectively. A primal heuristic is exploited to find valid upper bounds for the TQP. This heuristic takes a solution of the strengthened SDP relaxation as an input, hence ideally resulting in better solutions when the lower bounds improve. Our branching strategy takes a fractional variable and fixes this variable to one of the ternary values in $\{0, \pm 1\}$, so that the parent node (i.e., subproblem) is split into three child nodes. Details on the ingredients of our algorithm are provided in the next subsections.

6.1. Primal heuristic. We design a Variable Neighborhood Search (VNS) heuristic to compute high-quality feasible solutions. The method follows the classical VNS framework [36], alternating between (i) a randomized shaking phase to diversify the search and escape local minima and (ii) a deterministic best-improvement local search phase to intensify around promising solutions [26]. We first present the heuristic for the QUTO, then describe the adaptation for TQP-Linear, and finally detail the specialization for the TQP-Ratio.

VNS for QUTO. For a current solution x , we maintain the auxiliary vector $\alpha(x) = Qx$. The local neighborhood is defined by all single-coordinate ternary changes $x_i \leftarrow v$, with $v \in \{-1, 0, 1\} \setminus \{x_i\}$. Let $\Delta = v - x_i \in \{-2, -1, 1, 2\}$, \tilde{x} denote the resulting solution, and $f(x) = x^\top Qx + c^\top x$ denote the objective of the QUTO, then the objective variation is

$$f(\tilde{x}) = f(x) + 2\Delta \alpha_i + \Delta^2 Q_{ii} + \Delta c_i.$$

Hence, any trial move can be evaluated in $O(1)$ time. If a move is accepted, the auxiliary vector is updated as $\alpha \leftarrow \alpha + \Delta Q_{:i}$, where $Q_{:i}$ is the i th column of Q , which requires $O(n)$ time. Algorithm 6.1 summarizes the VNS scheme. The local search phase evaluates all $2n$ candidate moves in $O(n)$ time, selects the best improving one, and iterates until no improving move exists, yielding a 1-move local optimum. The shaking phase randomly modifies s coordinates with $s \in [s_{\min}, s_{\max}]$.

Whenever an improvement over the incumbent solution is found, the neighborhood radius is reset to its minimum value; otherwise, it is increased to promote diversification.

Algorithm 6.1 Variable Neighborhood Search for QUTO

```

1: input  $s_{\min}, s_{\max}, s_{\text{step}}, iter_{\max}$ 
2: Choose  $x \in \{0, \pm 1\}^n$  and compute  $(f, \alpha)$ 
3:  $(x_{\text{best}}, f_{\text{best}}) \leftarrow (x, f)$ 
4: for  $it = 1$  to  $iter_{\max}$  do
5:    $s \leftarrow s_{\min}$ 
6:   while  $s \leq s_{\max}$  do
7:      $x \leftarrow \text{Shake}(x, s)$ 
8:      $x \leftarrow \text{LocalSearch}(x)$ 
9:     Recompute  $f$ 
10:    if  $f < f_{\text{best}}$  then
11:       $(x_{\text{best}}, f_{\text{best}}) \leftarrow (x, f)$ 
12:       $s \leftarrow s_{\min}$ 
13:    else
14:       $x \leftarrow x_{\text{best}}$  and recompute  $(f, \alpha)$ 
15:       $s \leftarrow s + s_{\text{step}}$ 
16:    end if
17:  end while
18: end for
19: return  $x_{\text{best}}$ 

```

VNS for TQP-Linear. Here, feasibility is restricted by the linear constraint $\mathbf{1}_n^\top x = 0$. The local neighborhood is therefore defined by paired ternary changes on two distinct indices $i \neq j$, $x_i \leftarrow v_i$, $x_j \leftarrow v_j$, with $v_i, v_j \in \{-1, 0, 1\}$ such that $(v_i - x_i) + (v_j - x_j) = 0$. Let $\Delta_i = v_i - x_i$ and $\Delta_j = v_j - x_j$. Using again $\alpha(x) = Qx$, the objective variation is

$$f(\tilde{x}) = f(x) + 2\Delta_i \alpha_i + 2\Delta_j \alpha_j + \Delta_i^2 Q_{ii} + \Delta_j^2 Q_{jj} + 2\Delta_i \Delta_j Q_{ij} + \Delta_i c_i + \Delta_j c_j.$$

Each feasible paired move can be evaluated in $O(1)$ time. When such a move is accepted, the auxiliary vector is updated as $\alpha \leftarrow \alpha + \Delta_i Q_{:i} + \Delta_j Q_{:j}$. The local search phase explores this 2-move neighborhood and applies the best improving feasible move until convergence. The shaking phase is adapted accordingly by generating random feasible paired moves. Apart from the use of paired moves, the overall VNS structure is identical to the QUTO case. Since the balanced neighborhood contains $O(n^2)$ candidate moves, one complete neighborhood scan requires $O(n^2)$ time.

VNS for TQP-Ratio. For the fractional case, the local neighborhood again consists of single-coordinate ternary moves. The objective is the ratio denoted by $R(x) = f(x)/g(x)$. For a current solution x , we maintain the auxiliary vectors $\alpha^A(x) = Ax$, $\alpha^B(x) = Bx$. For a move $x_i \leftarrow v$ with $\Delta = v - x_i$, the numerator and denominator variations are

$$f(\tilde{x}) = f(x) + 2\Delta \alpha_i^A + \Delta^2 A_{ii} + \Delta a_i, \quad g(\tilde{x}) = g(x) + 2\Delta \alpha_i^B + \Delta^2 B_{ii} + \Delta b_i.$$

Thus, each trial move and the corresponding ratio value can be evaluated in $O(1)$ time. If the move is accepted, the auxiliary vectors are updated in $O(n)$ time as $\alpha^A \leftarrow \alpha^A + \Delta A_{:i}$, $\alpha^B \leftarrow \alpha^B + \Delta B_{:i}$.

The local search phase selects the best feasible move that strictly improves the ratio and terminates when no improving move exists, yielding a solution that is locally optimal with respect to all single-coordinate ternary changes. The VNS structure (shaking, intensification, and diversification) is identical to the QUTO case.

6.2. Cut separation. At each node in the B&B tree, violated triangle, split, RLT, and k -gonal inequalities are iteratively added to the relaxation. The separation

problems for triangle, (non-standard) split, and RLT inequalities can be solved exactly by exhaustive enumeration. In contrast, the number of k -gonal inequalities grows rapidly with the problem dimension, making exhaustive enumeration computationally prohibitive. Consequently, we employ heuristic separation strategies for these classes. Following the approach proposed in [25, 30] for the max-cut problem, the separation of k -gonal inequalities is formulated as the quadratic assignment problem (QAP) $\min_{P \in \Pi} \langle H, PX^*P^\top \rangle$, where Π denotes the set of permutation matrices, X^* is the current SDP solution, and $H = vv^\top$ encodes the structure of the corresponding inequality defined by a vector v ; see (4.4). The vector v has nonzero entries in its first k components, with $k \in \{5, 7, 9\}$ determining the inequality type. Since the QAP is \mathcal{NP} -hard, we solve it approximately using simulated annealing combined with a multistart strategy based on random initial permutations.

6.3. Branching strategy. We employ a ternary branching scheme reflecting the discrete domain $x_i \in \{0, \pm 1\}$. Whenever the SDP solution at a node is not integral, branching is performed by fixing one selected variable to each of its three possible values. In particular, branching on variable x_i creates three child nodes by imposing $(x_i, X_{ii}) \in \{(-1, 1), (0, 0), (1, 1)\}$. The branching variable is selected using a most-fractional rule adapted to the ternary domain. For each variable x_i , its fractionality is defined as the distance to the nearest feasible value, $\phi(x_i) = \min\{|x_i + 1|, |x_i|, |x_i - 1|\}$, and we choose $i^* \in \arg \max_i \phi(x_i)$. For the TQP-Ratio formulation, we instead branch on the variable exhibiting the largest violation of the scaled rank-one constraint $\rho Y_{jk} \approx y_j y_k$. Given the SDP solution (ρ^*, y^*, Y^*) , we select

$$i^* \in \arg \max_{j \in [n]} \sum_{k=1}^n Q_{kj} (y_j^* y_k^* - \rho^* Y_{jk}^*).$$

The scaled rank-one constraints follow from $|y| = \text{diag}(Y)$, $Y \in \rho \text{conv}\{\mathcal{F}_n^1\}$. Empirically, this branching rule outperforms the most-fractional strategy for solving (5.12), although detailed computational results are omitted for brevity.

6.4. Implementation details. Before starting the B&B procedure, we perform 100 local searches using the VNS algorithm initialized from randomly generated points, and the best solution found is used to initialize the global upper bound. In the VNS heuristic, we set $s_{\min} = 2$, $s_{\max} = n$, $s_{\text{step}} = 2$, and $iter_{\max} = 3$. Nodes of the B&B tree are explored using a best-first search strategy. At each cutting-plane iteration, inequalities are tested for violation with tolerance 10^{-3} , and at most 5,000 of the most violated inequalities are added. For the k -gonal inequalities, the simulated annealing procedure is executed 500 times for pentagonal ($k = 5$) inequalities and 1,000 times for heptagonal ($k = 7$) and enneagonal ($k = 9$) inequalities. The heuristic is run for all the distinct vectors v in (4.4). The cutting-plane procedure is terminated when fewer than n violated inequalities are identified. The algorithm terminates when the relative optimality gap satisfies $(UB - LB)/UB \leq 10^{-4}$, where UB and LB denote the best upper and lower bounds, respectively. **SDP-B&B** is implemented in **MATLAB** (version 2025a). The SDP relaxation at each node is solved with the interior-point solver **MOSEK** (version 11.0). All experiments are performed on a macOS system equipped with an Apple M4 Max processor (14 cores) and 36 GB of RAM, running macOS version 15.5.

7. Numerical results. In this section, we report the computational results for the QUTO, TQP-Linear, and TQP-Ratio formulations of the proposed **SDP-B&B** algorithm and compare it with state-of-the-art solvers.

We consider three types of instances for the QUTO and TQP-Linear called respectively Type-1, Type-2 and Type-3. Furthermore, we generate a set of random instances for the TQP-Ratio. Details on instance generation and parameter configuration are described in Appendix A. For all problem classes, we consider problem sizes $n \in \{60, 70, 80, 90, 100, 110, 120\}$ and three random seeds for each parameter configuration, resulting in 189 instances for each of the QUTO and TQP-Linear, and 63 instances for the TQP-Ratio.

We conducted experiments at the root node to evaluate the effectiveness of different classes of valid inequalities introduced in Section 4. Based on these experiments that we do not report for the sake of brevity, we implement within SDP-B&B a cutting-plane strategy that first enumerates all computationally inexpensive classes of inequalities jointly, namely triangle, RLT, and split inequalities, and subsequently generates violated k -gonal inequalities with $k = 5$ and $k = 7$. Observe that the enneagonal (i.e., k -gonal with $k = 9$) inequalities are no longer considered, due to their high computational cost.

7.1. Branch-and-bound results. In this section, we evaluate the computational performance of SDP-B&B and compare it with the commercial solver GUROBI [24] (version 12.0.2). We set the optimality gap tolerance to 10^{-4} , as for SDP-B&B. We also conducted experiments using BARON (version 25.3.19) [43] and SCIP [1] (version 10.0.0); however, both solvers consistently exhibited inferior performance compared to GUROBI. For this reason, their results are not reported in the following analysis. The comparison focuses on the QUTO, TQP-Linear, and TQP-Ratio instances, with problem sizes $n \in \{60, 70, 80, 90\}$. For these instances, we set a time limit of 3,600 seconds. In addition, to assess the scalability of the proposed approach, we also consider large-scale instances with $n \in \{100, 110, 120\}$. For these cases, we run only SDP-B&B with an increased time limit of 10,800 seconds. This choice is motivated by the observation that GUROBI already reached the time limit of 3,600 seconds on a substantial number of instances in the smaller size range (n between 60 and 90).

In Figure 1, we summarize the comparison between SDP-B&B and GUROBI on the three classes of problems. The plot reports the percentage of instances solved by each method over time, with time displayed on a logarithmic scale on the x-axis.

In the upper-left panel, results for the QUTO instances are shown. SDP-B&B clearly outperforms GUROBI: it solves all instances within the one-hour time limit, whereas GUROBI cannot. Indeed, GUROBI fails to solve 6 out of 36 instances of Type-1 of sizes 80 and 90, leaving residual gaps between 1.8% and 6%. For Type-2 instances, GUROBI fails on 4 out of 36 instances of size 90, with residual gaps ranging from 2.2% to 5.5%. Type-3 instances are the most challenging: GUROBI cannot solve 19 instances with sizes 70, 80, and 90, with residual gaps between 4.5% and 25%.

The upper-right panel reports results for the TQP-Linear instances. Again, SDP-B&B solves all instances within the time limit, whereas GUROBI does not. For Type-1 instances, GUROBI fails on 5 instances of sizes 80 and 90, with residual gaps between 0.02% and 0.07%. For Type-2 instances, only one instance remains unsolved, with a residual gap of 1.4%. For Type-3 instances, GUROBI fails on 19 instances of sizes between 70 and 90, with residual gaps ranging from 0.03% to 0.1%.

In the second row of Figure 1, we compare three approaches: (i) the Dinkelbach algorithm where the QUTO subproblem is solved by SDP-B&B (Parametrized QUTO); (ii) the Dinkelbach algorithm where the QUTO subproblem is solved by GUROBI; and (iii) the direct solution of the TQP-Ratio via SDP-B&B, exploiting relaxation (5.13). The Dinkelbach algorithm is initialized with the solution obtained by the VNS-Ratio

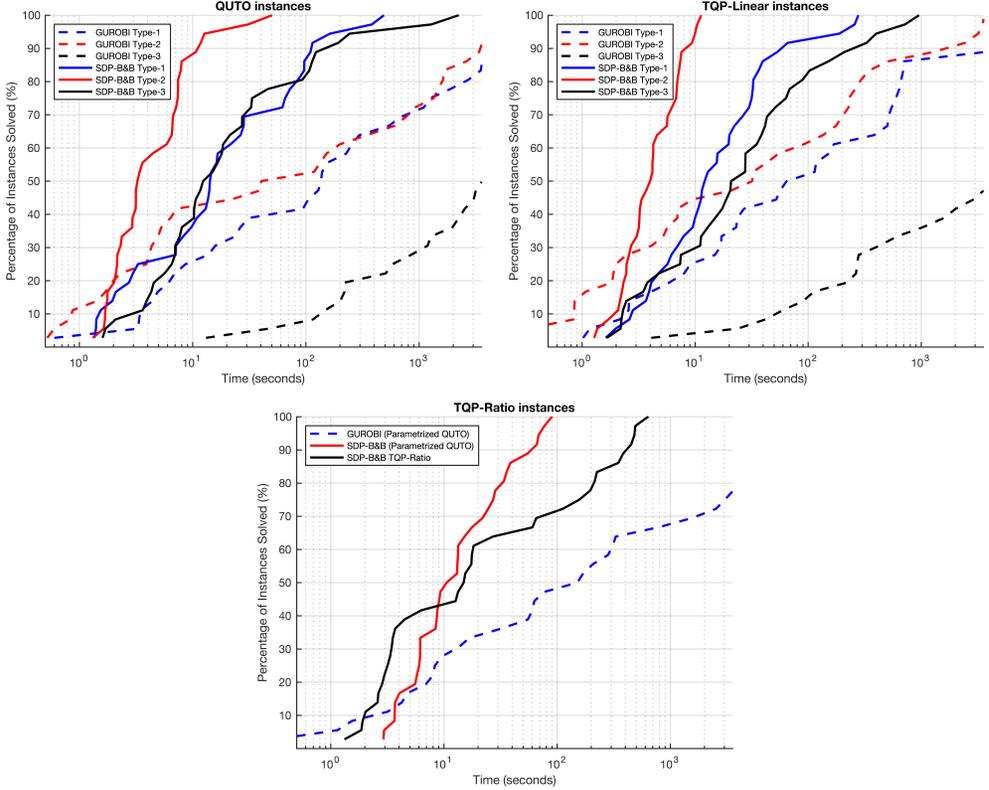


Figure 1: Survival plots comparing SDP-B&B with GUROBI on QUTO, TQP-Linear and TQP-Ratio instances of size $n \in \{60, 70, 80, 90\}$. The plot reports the percentage of problems solved over time, where the time is represented on a logarithmic scale.

heuristic, which turned out to be optimal on all tested instances; hence, only one QUTO subproblem needs to be solved for each instance. The plot again shows the percentage of instances solved over time. Here, GUROBI is clearly outperformed by both of our approaches, which solve all instances within the time limit, whereas GUROBI fails on 18 out of 36 instances. Moreover, our implementation of the parametrized QUTO is generally faster than solving the TQP-Ratio directly.

Table 1 reports the average performance of SDP-B&B on large-scale instances with $n \in \{100, 110, 120\}$ for the QUTO, TQP-Linear, and TQP-Ratio. Results are averaged over the three random seeds per parameter configuration. Overall, the method scales well up to $n = 120$, with performance strongly influenced by the instance type.

For both the QUTO and TQP-Linear, Type-2 instances are the easiest, typically solved at the root node or with very few nodes due to the favorable low-rank structure, which leads to tight relaxations. Type-1 instances show moderate growth in difficulty with n , but most cases are still solved to optimality, with only small residual gaps at the largest size (at most 0.94% for the QUTO and 0.61% for the TQP-Linear). Type-3 instances are the most challenging, as the structure weakens the bounds and results in larger search trees and longer running times. Nevertheless, even in these cases SDP-B&B produces high-quality solutions, with remaining gaps typically below about 2% (maximum 2.05% for the QUTO and 1.59% for the TQP-Linear).

Table 1: Average results of SDP-B&B for the QUTO, TQP-Linear, and TQP-Ratio instances ($n = 100$ – 120). All metrics are averaged over three random seeds. Column “sol” reports the number of instances successfully solved.

(a) QUTO						(b) TQP-Linear					
n	p	gap (%)	nodes	time (s)	sol	n	p	gap (%)	nodes	time (s)	sol
Type-1						Type-1					
100	25	0.00	6.00	101.40	3	100	25	0.00	8.00	154.34	3
100	50	0.00	4.00	96.27	3	100	50	0.00	7.00	172.36	3
100	75	0.00	33.00	366.03	3	100	75	0.00	4.00	135.71	3
110	25	0.00	20.00	541.19	3	110	25	0.00	16.00	369.44	3
110	50	0.00	114.00	3752.82	3	110	50	0.00	81.00	2811.27	3
110	75	0.00	112.00	1899.71	3	110	75	0.00	24.00	1146.84	3
120	25	0.00	79.00	5026.98	3	120	25	0.00	47.00	2097.98	3
120	50	0.94	53.00	7807.28	3	120	50	0.61	105.33	8044.35	1
120	75	0.00	184.00	8342.39	3	120	75	0.00	67.00	4059.90	3
Type-2						Type-2					
100	25	0.00	1.00	8.81	3	100	25	0.00	1.00	7.34	3
100	50	0.00	1.00	15.86	3	100	50	0.00	1.00	12.50	3
100	75	0.00	4.00	49.50	3	100	75	0.00	1.00	17.83	3
110	25	0.00	1.00	17.70	3	110	25	0.00	1.00	15.33	3
110	50	0.00	1.00	16.17	3	110	50	0.00	1.00	12.90	3
110	75	0.00	6.00	163.61	3	110	75	0.00	1.00	24.96	3
120	25	0.00	1.00	23.02	3	120	25	0.00	7.00	87.87	3
120	50	0.00	2.00	66.19	3	120	50	0.00	1.00	36.95	3
120	75	0.00	8.00	217.43	3	120	75	0.00	2.00	145.60	3
Type-3						Type-3					
100	25	0.00	57.00	1338.23	3	100	25	0.00	175.00	3747.39	3
100	50	0.00	80.00	1893.25	3	100	50	0.44	101.33	3808.51	2
100	75	0.62	101.33	4554.90	2	100	75	0.45	121.33	4455.57	2
110	25	0.00	91.00	3249.31	3	110	25	0.00	93.00	3280.33	3
110	50	0.31	96.33	5544.80	2	110	50	0.20	115.67	4904.71	2
110	75	2.05	160.33	10567.30	1	110	75	1.59	189.00	9918.83	1
120	25	0.96	118.33	8424.84	1	120	25	1.35	81.33	7816.62	1
120	50	0.54	141.33	8283.22	2	120	50	0.39	114.67	7203.70	2
120	75	1.29	151.33	10782.52	1	120	75	1.29	133.33	9628.57	1
(c) TQP-Ratio (Parametrized QUTO)						(d) TQP-Ratio					
n	d	iter	nodes	time (s)	sol	n	d	gap (%)	nodes	time (s)	sol
100	25	1.00	10.00	335.98	3	100	25	0.00	92.00	1090.39	3
100	50	1.00	138.00	4640.18	3	100	50	0.25	441.33	8362.24	1
100	75	1.00	92.00	2632.86	3	100	75	0.23	580.67	8525.15	1
110	25	1.00	45.00	2064.74	3	110	25	0.25	147.67	5192.44	2
110	50	1.00	83.00	5179.87	2	110	50	0.53	393.67	10800.00	0
110	75	1.00	104.67	7915.80	1	110	75	0.60	631.00	10800.00	0
120	25	1.00	68.00	3715.54	3	120	25	0.29	168.33	5917.93	2
120	50	1.00	93.33	10800.00	0	120	50	1.37	370.33	10800.00	0
120	75	1.00	82.33	9571.01	1	120	75	0.14	309.67	6206.39	2

The TQP-Ratio results highlight some differences between the parametrized QUTO (subtable (c)) and the direct solution (subtable (d)) approaches. The parametric method is often more efficient, but when the iterative procedure does not terminate within the time limit, it cannot provide a valid optimality gap for the original ratio problem. In contrast, the direct solution approach always produces a lower bound and therefore a certified gap, even when prematurely terminated. The reported gaps for the direct solution approach remain small overall, ranging between 0.14% and 1.37% on the unsolved instances. Moreover, there are configurations in which the direct approach outperforms the parametric one, as for example for $n = 120$ and $d = \{50, 75\}$. Hence, the direct solution approach remains valuable whenever solution certification is required, or when the Dinkelbach scheme is not initialized with a high-quality solution, in which case multiple outer iterations may be needed. In our experiments, the VNS heuristic consistently identifies a solution that is already optimal, so that no further Dinkelbach updates are necessary. Consequently, the computational effort

of the parametric method essentially corresponds to solving a single QUTO instance, explaining its good practical performance.

8. Conclusions. We study the TQP from the perspective of ISDP. Our novel approach yields a theoretical analysis of the set of ternary PSD matrices (Theorem 2.6) and the associated polyhedra (see (2.5) and Proposition 2.7). These results enable us to derive an ISDP formulation for the TQP (3.3), from which we obtain several classes of cutting planes, including (generalized) triangle inequalities and (non-standard) split inequalities, that improve our basic SDP relaxation for the TQP (4.1).

We consider three variants of the TQP; QUTO (5.1), TQP-Linear (5.3), and TQP-Ratio (5.8). We prove in Theorem 5.1 that the QUTO reduces to the max-cut problem for a particular quadratic objective. We also derive a reformulation of the TQP-Ratio (5.12) that does not involve rational terms.

Finally, we design a B&B algorithm that solves, at each node of the tree, the basic SDP relaxation strengthened by a number of valid inequalities. The B&B algorithm integrates a ternary branching scheme with sophisticated primal heuristics, yielding a method that outperforms significantly competing approaches. Overall, these results position ISDP as a competitive new approach for ternary quadratic programming.

REFERENCES

- [1] T. ACHTERBERG, *SCIP: solving constraint integer programs*, Math. Program. Comput., 1 (2009), pp. 1–41.
- [2] P. A. AMARAL AND I. M. BOMZE, *Copositivity-based approximations for mixed-integer fractional quadratic optimization*, Pac. J. Optim., 11 (2015), pp. 225–238.
- [3] P. BELOTTI, J. LEE, L. LIBERTI, F. MARGOT, AND A. WÄCHTER, *Branching and bounds tightening techniques for non-convex MINLP*, Optim. Methods Softw., 24 (2009), pp. 597–634.
- [4] A. BHASKARA, M. CHARIKAR, R. MANOKARAN, AND A. VIJAYARAGHAVAN, *On quadratic programming with a ratio objective*, in International Colloquium on Automata, Languages, and Programming, Springer, 2012, pp. 109–120.
- [5] C. BUCHHEIM, A. CAPRARA, AND A. LODI, *An effective branch-and-bound algorithm for convex quadratic integer programming*, Math. Program., 135 (2011), pp. 369–395.
- [6] C. BUCHHEIM, M. DE SANTIS, L. PALAGI, AND M. PIACENTINI, *An exact algorithm for non-convex quadratic integer minimization using ellipsoidal relaxations*, SIAM J. Optim., 23 (2013), pp. 1867–1889.
- [7] C. BUCHHEIM, M. MONTENEGRO, AND A. WIEGELE, *SDP-based branch-and-bound for non-convex quadratic integer optimization*, J. Glob. Optim., 73 (2018), pp. 485 – 514.
- [8] C. BUCHHEIM AND E. TRAVERSI, *On the separation of split inequalities for non-convex quadratic integer programming*, Discret. Optim., 15 (2015), pp. 1–14.
- [9] C. BUCHHEIM AND A. WIEGELE, *Semidefinite relaxations for non-convex quadratic mixed-integer programming*, Math. Program., 141 (2012), pp. 435 – 452.
- [10] S. BURER AND A. N. LETCHFORD, *Unbounded convex sets for non-convex mixed-integer quadratic programming*, Math. Program., 143 (2014), pp. 231–256.
- [11] S. CALLEGARI, F. BIZZARRI, R. ROVATTI, AND G. SETTI, *On the approximate solution of a class of large discrete quadratic programming problems by $\Delta\Sigma$ modulation: the case of circulant quadratic forms*, Trans. Sig. Proc., 58 (2010), p. 6126–6139.
- [12] M. T. ÇEZİK AND G. N. IYENGAR, *Cuts for mixed 0-1 conic programming*, Math. Program., 104 (2005), pp. 179–202.
- [13] V. CHVÁTAL, *Edmonds polytopes and a hierarchy of combinatorial problems*, Discrete Math., 4 (1973), pp. 305–337.
- [14] F. DE MELJER, M. SIEBENHOFER, R. SOTIROV, AND A. WIEGELE, *Spanning and splitting: Integer semidefinite programming for the quadratic minimum spanning tree problem*, Eur. J. Oper. Res., (2025).
- [15] F. DE MELJER AND R. SOTIROV, *The Chvátal-Gomory procedure for integer SDPs with applications in combinatorial optimization*, Math. Program., 209 (2024), p. 323–395.
- [16] F. DE MELJER AND R. SOTIROV, *On integrality in semidefinite programming for discrete optimization*, SIAM J. Optim., 34 (2024), pp. 1071–1096.
- [17] F. DE MELJER AND R. SOTIROV, *Lagrangian duality for mixed-integer semidefinite programming:*

- Theory and algorithms*. Preprint, <https://arxiv.org/pdf/2501.11397>, 2025.
- [18] M. M. DEZA AND M. LAURENT, *Geometry of Cuts and Metrics*, vol. 15 of Algorithms and Combinatorics, Springer, Berlin, 1997.
- [19] W. DINKELBACH, *On nonlinear fractional programming*, *Manag. Sci.*, 13 (1967), pp. 492–498.
- [20] D. DRUSVYATSKIY AND H. WOLKOWICZ, *The many faces of degeneracy in conic optimization*, *Foundations and Trends in Optimization*, 3 (2017), pp. 77–170.
- [21] L. GALLI AND A. N. LETCHFORD, *Valid inequalities for quadratic optimisation with domain constraints*, *Discrete Optim.*, 41 (2021), p. 100661.
- [22] M. X. GOEMANS AND D. P. WILLIAMSON, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, *J. ACM*, 42 (1995), pp. 1115–1145.
- [23] R. E. GOMORY, *Outline of an algorithm for integer solutions to linear programs*, *Bull. Am. Math. Soc.*, 64 (1958), pp. 275–278.
- [24] GUROBI OPTIMIZATION, LLC, *Gurobi Optimizer Reference Manual*, 2025, <https://www.gurobi.com>.
- [25] N. GUSMEROLI, T. HRGA, B. LUŽAR, J. POVH, M. SIEBENHOFER, AND A. WIEGELE, *Biqbin: A parallel branch-and-bound solver for binary quadratic problems with linear constraints*, *ACM Trans. Math. Softw.*, 48 (2022).
- [26] P. HANSEN, N. MLADENOVIĆ, AND J. A. MORENO PEREZ, *Variable neighbourhood search: methods and applications*, *Ann. Oper. Res.*, 175 (2010), pp. 367–407.
- [27] T. HE, S. LIU, AND M. TAWARMALANI, *Convexification techniques for fractional programs*, *Math. Program.*, 213 (2025), pp. 107–149.
- [28] C. HELMBERG, *Semidefinite programming for combinatorial optimization*, Habilitationsschrift ZR-00-34, Technische Universität Berlin, January 2000.
- [29] A. HOTHO, R. JÄSCHKE, C. SCHMITZ, AND G. STUMME, *Information retrieval in folksonomies: Search and ranking*, in *The Semantic Web: Research and Applications*, Y. Sure and J. Domingue, eds., vol. 4011 of Lecture Notes in Computer Science, Berlin/Heidelberg, 2006, Springer, pp. 411–426.
- [30] T. HRGA AND J. POVH, *MADAM: a parallel exact solver for max-cut based on semidefinite programming and admm*, *Comput. Optim. Appl.*, 80 (2021), pp. 347–375.
- [31] N. KRISLOCK, J. MALICK, AND F. ROUPIN, *BiqCrunch: A semidefinite branch-and-bound method for solving binary quadratic problems*, *ACM Trans. Math. Softw.*, 43 (2017), pp. 1–23.
- [32] A. N. LETCHFORD, *Integer quadratic quasi-polyhedra*, in *Integer Programming and Combinatorial Optimization*, F. Eisenbrand and F. B. Shepherd, eds., Springer Berlin Heidelberg, 2010, pp. 258–270.
- [33] A. N. LETCHFORD AND M. M. SØRENSEN, *Binary positive semidefinite matrices and associated integer polytopes*, *Math. Program.*, 131 (2008), pp. 253–271.
- [34] F. LIEDER, F. B. A. RAD, AND F. JARRE, *Unifying semidefinite and set-copositive relaxations of binary problems and randomization techniques*, *Comput. Optim. Appl.*, 61 (2015), pp. 669–688.
- [35] G. P. MCCORMICK, *Computability of global solutions to factorable nonconvex programs: Part i — convex underestimating problems*, *Math. Program.*, 10 (1976), pp. 147–175.
- [36] N. MLADENOVIĆ AND P. HANSEN, *Variable neighborhood search*, *Comput. Oper. Res.*, 24 (1997), pp. 1097–1100.
- [37] M. W. PADBERG, *The boolean quadric polytope: Some characteristics, facets and relatives*, *Math. Program.*, 45 (1989), pp. 139–172.
- [38] N. T. PHUONG, H. TUY, AND F. AL-KHAYYAL, *Optimization of a quadratic function with a circulant matrix*, *Comput. Optim. Applic.*, 35 (2006), p. 135–159.
- [39] A. D. PIA AND D. ZHOU, *An SDP relaxation for the sparse integer least square problem*, *INFORMS J. Optim.*, accepted, (2025), arXiv:2203.02607.
- [40] V. PICCIALI, A. M. SUDOSO, AND A. WIEGELE, *SOS-SDP: an exact solver for minimum sum-of-squares clustering*, *INFORMS J. Comput.*, 34 (2022), pp. 2144–2162.
- [41] A. P. PUNNEN, ed., *The Quadratic Unconstrained Binary Optimization Problem: Theory, Algorithms, and Applications*, Springer Cham, 2022.
- [42] R. G. RÓDENAS, M. L. LÓPEZ, AND D. VERASTEGUI, *Extensions of Dinkelbach’s algorithm for solving non-linear fractional programming problems*, *TOP*, 7 (1999), pp. 33–70.
- [43] N. V. SAHINIDIS, *BARON 16.3.4: Global Optimization of Mixed-Integer Nonlinear Programs, User’s Manual*, 2016. User’s Manual.
- [44] S. SCHAIBLE, *Fractional programming. II, on Dinkelbach’s algorithm*, *Manag. Sci.*, 22 (1976), pp. 868–873, <http://www.jstor.org/stable/2630018> (accessed 2026-01-13).
- [45] H. D. SHERALI AND W. P. ADAMS, *A Reformulation-Linearization Technique for Solving Dis-*

crete and Continuous Nonconvex Problems, Springer, 1998.

- [46] THE SAGE DEVELOPERS, *SageMath, the Sage Mathematics Software System (Version 10.6)*, 2025, <https://www.sagemath.org>.
- [47] L. TREVISAN, *Max cut and the smallest eigenvalue*, in Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09), New York, NY, USA, 2009, ACM, pp. 263–272.
- [48] R.-C. TZENG, B. ORDOZGOITI, AND A. GIONIS, *Discovering conflicting groups in signed networks*, in NeurIPS, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., vol. 33, Curran Associates, Inc., 2020, pp. 10974–10985.

Appendix A. Instance generation. We use the following three types of data generators to obtain instance for the TQP-Linear (5.3).

- (i) Type-1 instances are generated as follows, see also [9]. For given $n \in \mathbb{N}$ and $p \in \{25, 50, 75\}$, we sample $\lfloor pn/100 \rfloor$ values uniformly from the interval $[-1, 0]$, and the remaining $n - \lfloor pn/100 \rfloor$ values uniformly from $[0, 1]$. These values form the vector μ . Then, we generate an $n \times n$ matrix whose elements are drawn uniformly at random from $[-1, 1]$ and orthonormalize its columns to obtain vectors v_i . Finally, we set $Q = \sum_{i=1}^n \mu_i v_i v_i^\top$.
- (ii) Type-2 instances provide matrices Q whose rank is $\lceil n/2 \rceil$, see also [7] for a similar generator. For given $n \in \mathbb{N}$, we generate an $n \times n$ matrix whose elements are drawn uniformly at random from $[-1, 1]$ and orthonormalize its columns to obtain vectors v_i . Then, the first $\lfloor n/2 \rfloor$ entries of μ are set to zero, and each remaining entry is nonnegative with probability $p/100$ and zero otherwise. We consider $p \in \{25, 50, 75\}$.
- (iii) Type-3 instances provide random sparse matrices Q , see also [7]. For given $n \in \mathbb{N}$ and $p \in \{25, 50, 75\}$, each entry of the matrix Q is zero with probability $1 - p/100$ and the remaining entries are drawn uniformly at random from the interval $[-1, 1]$.

In all above types of instance generators, elements of c are drawn uniformly at random from $[-1, 1]$.

We generate QUTO instances that do not reduce to max-cut instances. In particular, we use the same three types of generators as described above and take the absolute values of the diagonal elements, ensuring the existence of positive diagonal elements, see Corollary 5.2.

We generate instances for the TQP-Ratio (5.8) as follows. For each test instance, the numerator coefficients (A, a, a_0) and denominator coefficients (B, b, b_0) are generated independently at random. The quadratic matrices $A, B \in \mathcal{S}^n$ are sparse: given a target density $d \in [0, 100]$, each entry in the lower triangular part is set to a non-zero value with probability $d/100$ and to zero otherwise, and symmetry is enforced by mirroring to the upper triangular part. Whenever an entry is nonzero, it is sampled uniformly from the integer interval $\{-50, \dots, 50\}$. The linear vectors $a, b \in \mathbb{R}^n$ and the constant term a_0 are generated independently with i.i.d. entries uniformly distributed over $\{-50, \dots, 50\}$. We set $b_0 = 1 + \sum_{i=1}^n |B_{ii}| + 2 \sum_{1 \leq i < j \leq n} |B_{ij}| + \sum_{i=1}^n |b_i|$, which ensures $g(x) \geq 1$ for all $x \in \{0, \pm 1\}^n$.