Decompositions of Semidefinite Matrices and the Perspective Reformulation of Nonseparable Quadratic Programs

Antonio Frangioni [*] Claudio Gentile [†] James Hungerford [‡]

November 1, 2019

Abstract

We study the problem of decomposing the Hessian matrix of a Mixed-Integer Convex Quadratic Program into the sum of positive semidefinite 2×2 matrices. Solving this problem enables the use of Perspective Reformulation techniques for obtaining strong lower bounds for MICQPs with semi-continuous variables but a non-separable objective function. An explicit formula is derived for constructing 2×2 decompositions when the underlying matrix is Weakly Scaled Diagonally Dominant, and necessary and sufficient conditions are given for the decomposition to be unique. For matrices lying outside this class, two exact SDP approaches and an efficient heuristic are developed for finding approximate decompositions. We present preliminary results on the bound strength of a 2×2 Perspective Reformulation for the Portfolio Optimization Problem, showing that for some classes of instances the use of 2×2 matrices can significantly improve the quality of the bound w.r.t. the best previously known approach, although at a possibly high computational cost.

Keywords: *Mixed-Integer Quadratic Programming, Semicontinuous variables, Portfolio Optimization, Scaled Diagonal Dominance, Matrix Decomposition.*

1 Introduction

We are interested in the solution of Mixed-Integer NonLinear Programs (MINLP) with semicontinuous variables, but where the objective function is *not* separable among the semicontinuous variables. To simplify the discussion we will mainly refer to Mixed-Integer Quadratic Programs (MIQP) of the form

$$\min x^T Q x + q^T x + c^T y \tag{1a}$$

$$\text{s.t. } Ax + By \leq b \tag{1b}$$

$$l_i y_i \leq x_i \leq u_i y_i \qquad i \in N \tag{1c}$$

$$y_i \in \{0, 1\} \qquad i \in N \tag{1d}$$

where $x = [x_i]_{i \in N} \in \mathbb{R}^n$ ($N = \{1, 2, \dots, n\}$), $Q \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite (PSD), $A, B \in \mathbb{R}^{m \times n}$, and q, c, b, l and u are real-valued column vectors of appropriate dimensions. Constraints (1c) and (1d) imply that each x_i is a semicontinuous variable governed by the

[*] Dipartimento di Informatica, Università di Pisa, e-mail: frangio@di.unipi.it

[†] Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti" del CNR, e-mail: gentile@iasi.rm.cnr.it

[‡] RaceTrac, Atlanta, Georgia, e-mail: jamesthungerford@gmail.com

binary variable y_i; that is, $x_i = 0$ if $y_i = 0$ and $x_i \in \mathcal{P}_i = [l_i, u_i]$ if $y_i = 1$. We require each $\mathcal{P}_i$ to be a compact interval (i.e., $-\infty < l_i \leq u_i < \infty$). The formulation can be made more general in several ways, e.g., by allowing the constraints $A(x)$ and the objective function $q(x)$ to be nonlinear (but, hopefully, convex for the approach to provide significant benefits), or having "other" variables and (possibly, convex) constraints, but we will stick to (1) for notational simplicity.

When Q is diagonal, we have

$$x^T Q x = \textstyle\sum_{i \in N} Q_{ii} x_i^2 \ , \tag{2}$$

and the problem is well-suited for application of the *Perspective Reformulation* (PR) technique. This amounts to replacing the objective function with its *convex envelope* with respect to the semi-continuous constraints (1c)–(1d), which is obtained by substituting (1a) with

$$\min \textstyle\sum_{i \in N} Q_{ii} x_i^2 / y_i + q^T x + c^T y \ , \tag{3}$$

where we assume that $x_i^2/y_i = 0$ if $y_i = 0$. Note that $\sum_{i \in N} Q_{ii} x_i^2 / y_i$ is the *perspective function* of (2). Despite its appearance, (3) is convex and therefore its continuous relaxation

$$\begin{array}{lll} \min & \sum_{i \in N} Q_{ii} x_i^2 / y_i + q^T x + c^T y & \\ \text{s.t.} & Ax + By \leq b & \\ & l_i y_i \leq x_i \leq u_i y_i & i \in N \\ & y_i \in [0, 1] & i \in N \end{array}$$

can be efficiently solved. (Henceforth, we will use the notation $\underline{\text{P}}$ to denote the continuous relaxation of a Mixed-Integer Nonlinear Programming Problem P, and $\underline{\text{PR}}$ as shorthand for the continuous relaxation of a perspective reformulation, known as a *Perspective Relaxation.*) For instance, one can either iteratively approximate it by using linear approximations (*Perspective Cuts* [10]), or reformulate it as a Second-Order Cone Program and solve it in one blow with existing approaches [12], or even consider using specific reformulations [8].

In the present paper, we are interested in the case where the objective function (1a) is *not* separable (ie. Q is not a diagonal matrix). A simple but effective extension of the above approach to the non-separable case was proposed in [10, 11], and refined in [21]. The main idea is to reformulate (1) as

$$\min \{ \ \textstyle\sum_{i \in N} \delta_i x_i^2 + x^T (Q - diag(\delta)) x + q^T x + c^T y \ : \ \text{(1b)–(1d)} \ \} \ , \tag{4}$$

where $\delta \geq 0$ is a vector *chosen* such that $Q - diag(\delta) \succeq 0$. One can then apply the PR to the separable part of the objective function in (4), which leads to

$$\min \{ \ \textstyle\sum_{i \in N} \delta_i x_i^2 / y_i + x^T (Q - diag(\delta)) x + q^T x + c^T y \ : \ \text{(1b)} - \text{(1d)} \ \} \ . \tag{5}$$

The advantage of (5) is that its continuous relaxation ($\underline{5}$), the $\underline{\text{PR}}$, often provides a strictly better bound than the continuous relaxation of (1). Clearly, the quality of the bound depends on δ, and, intuitively, "the larger δ, the better the bound".

In this paper, we generalize the above approach by attempting to extract $k \times k$ principal submatrices from Q, where k is small. This leads to either an *approximate* or an *exact* decomposition of Q into $k \times k$ matrices. We show how the Perspective Reformulation technique may then be extended to each of the extracted $k \times k$ matrices to obtain a potentially tighter convex relaxation for (1), as a result of incorporating more of the structure of Q into the PR. For general k, the problem of finding a "large" or even an "optimal" collection of $k \times k$ matrices to extract is shown to amount to the solution of a semidefinite program. For the case $k = 2$, we give a characterization of the matrices Q which have an exact decomposition (ie. the remainder after the extraction is zero), and we derive a closed form expression for the matrices in the decomposition whenever such a decomposition exists. Moreover, we show how for a general matrix Q the characterization may be exploited to devise an efficient heuristic for finding approximate 2×2 decompositions. We remark that in [4] a similar idea (philosophically) has been used to develop convex MIQP reformulations of non-convex MIQPs. However, whereas in [4] the authors add a large non-diagonal matrix to the Hessian of the objective function, here we extract $k \times k$ principal submatrices, where k is small.

The structure of the paper is as follows. In Section 2 we review the relevant literature on the diagonal extraction technique (5). In Section 3 we show how to compute the PR of a general non-separable k-dimensional MIQP with semicontinuous variables for arbitrary (but, in fact, necessarily "small") k. In Section 4 we present two semidefinite programming approaches for extracting *approximate* $k \times k$ decompositions of Q. In Section 5 we give a characterization of the class of matrices having an exact 2×2 decomposition and we show how to construct an exact 2×2 decomposition when one exists. In Section 6, we exploit the characterization to devise alternative, faster (possibly heuristic) approaches to compute approximate 2×2 decompositions. In addition, we discuss the relationships between our approach and a method of extending the PR to non-separable quadratic functions independently developed in the recent [1]. Finally, in Section 7 computational results showing the efficiency and effectiveness of the proposed approaches are reported, and conclusions are drawn.

Throughout the paper, the following notation is used. $\mathbb{R}_+ = \{x \in \mathbb{R} \; : \; x \geq 0\}$. For a given $n \times n$ matrix A, $diag(A)$ is the diagonal matrix whose i-th diagonal element is A_{ii}. Given an n-vector d, $diag(d)$ denotes the $n \times n$ diagonal matrix whose i-th diagonal element is d_i. $\langle A \, , \, B \rangle = trace(A^T B)$ whenever $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n}$, and $||A|| = \sqrt{\langle A \, , \, A \rangle}$. I denotes the $n \times n$ identity matrix. $\Lambda(A)$ is the set of eigenvalues of A and $\rho(A) = \max\{\, |\lambda| \; : \; \lambda \in \Lambda(A) \,\}$ is its spectral radius. $P = \{\, (i,j) \in N \times N \; : \; i < j \,\}$. If $t \in \mathbb{R}$, then $sgn(t)$ is the sign of t. The remaining notation is fairly standard.

2 Related work on diagonal extractions

In this section, we give an overview of the methods proposed in the literature for extracting 1×1 principal submatrices from Q; that is, extracting a diagonal matrix $diag(\delta)$ and reformulating (1) as (5). As already pointed out, the quality of the bound provided by the continuous relaxation of (5) depends on δ. In [10] a simple and inexpensive way of choosing δ, based on an eigenvalue computation, was used. In [11] an SDP approach was proposed. In particular, given a vector of weights $\alpha = [\alpha_i]_{i \in N} \geq 0$ for the individual components of δ, finding the "largest" possible δ can be cast as the following dual pair of SemiDefinite Programs (SDP):

$$\begin{array}{c} \max_\delta \; \left\{ \; \sum_{i \in N} \alpha_i \delta_i \; : \; Q - \sum_{i \in N} D^i \delta_i \succeq 0 \;\; , \;\; \delta \geq 0 \; \right\} \\ \min_F \; \left\{ \; \langle \, Q \, , \, F \, \rangle \; : \; diag(F) \geq \alpha \;\; , \;\; F \succeq 0 \; \right\} \end{array} \quad , \tag{6}$$

where $D^i = e_i e_i^T$, e_i being the i-th vector of the canonical basis of $\mathbb{R}^n$. The idea is that the weights α_i should be chosen in order to reflect the different relevance of having a large quadratic coefficient for each x_i in (4). In [11], unitary weights were used for simplicity, while an approach for finding the "best possible" δ based on solving the following program, was proposed in [21]:

$$\max_\delta \qquad \min_{x,y} \;\; q^T x + c^T y + \textstyle\sum_{i \in N} \delta_i x_i^2 / y_i + x^T (Q - diag(\delta)) x \tag{7a}$$

$$\text{s.t.} \qquad (1b) - (1c) \;\; , \;\; y \in [0,1]^n \tag{7b}$$

$$\text{s.t. } \delta \geq 0 \;\; , \;\; Q - diag(\delta) \succeq 0 \tag{7c}$$

It is plain to see that the above program indeed produces the best possible lower bound. It is also easy to see that it is "easy", as the function $\phi(\delta) = \min_{x,y} \{ \text{(7a)} \;:\; \text{(7b)} \}$ is clearly concave in δ, being the pointwise infimum of (infinitely many) linear functions in δ, indexed over x and y. Indeed, (7) can be formulated as an SDP. In [21] this is done using Lagrangian duality arguments, but an alternative—and perhaps simpler—approach based on SDP duality is as follows. Since the innermost objective function of (7) is convex in x, y and concave in δ, and the feasible region is bounded in the (x, y) component, we may interchange the maximization and minimization in (7) [18, Corollary 37.3.2], to obtain

$$\min_{x,y} \left\{ x^T Q x + q^T x + c^T y + \psi(x,y) \;:\; \text{(1b)} \;,\; \text{(1c)} \;,\; y \in [0,1]^n \right\} \;, \tag{8}$$

$$\text{where} \qquad \psi(x,y) = \max_\delta \left\{ \textstyle\sum_{i \in N} (x_i^2/y_i - x_i^2)\delta_i \;:\; Q - diag(\delta) \succeq 0 \;,\; \delta \geq 0 \right\} \;. \tag{9}$$

Clearly, (8) is a convex program: the feasible set is convex, and the objective function is convex, $\psi(x, y)$ being the pointwise maximum of infinitely many linear functions in δ (and the rest being convex from the start). To cast (8) as a single SDP it suffices to perform the variable change $\Phi = Q - diag(\delta)$ ($\succeq 0$), which yields $\delta_i = Q_{ii} - \Phi_{ii}$; then, one can write

$$\psi(x,y) = \sum_{i \in N} Q_{ii} x_i^2 / y_i + \left\{ \begin{array}{lll} \max_\Phi & \langle\, xx^T - V(x,y) \,,\, \Phi \,\rangle & \\ \text{s.t.} & \langle\, O^{ij} \,,\, \Phi \,\rangle = 2Q_{ij} & (i,j) \in P \\ & \langle\, D^i \,,\, \Phi \,\rangle \leq Q_{ii} & i \in N \\ & \Phi \succeq 0 & \end{array} \right. \tag{10}$$

where $O^{ij} = e_i e_j^T + e_j e_i^T$ (the symmetric matrix having 1 only in the elements (i, j) and (j, i) and zero elsewhere) and $V(x, y) = \sum_{i \in N} D^i x_i^2 / y_i$. The dual of the maximization problem in (10) is

$$\min_F \left\{ \langle\, Q \,,\, F \,\rangle \;:\; F \succeq xx^T - V(x,y) \;,\; diag(F) \geq 0 \right\} \;.$$

Now, a well-known application of the Lemma on the Schur complement gives

$$F \succeq xx^T - V(x,y) \qquad \equiv \qquad \begin{bmatrix} 1 & x^T \\ x & F + V(x,y) \end{bmatrix} \succeq 0 \;.$$

Analogously, using the well-known

$$y_i \geq 0 \;,\; w_i \geq 0 \;,\; w_i \geq x_i^2/y_i \qquad \equiv \qquad \begin{bmatrix} w_i & x_i \\ x_i & y_i \end{bmatrix} \succeq 0 \;,$$

one ends up with the SDP form of (8)

$$\begin{array}{lll} \min_{x,y,F,w} & q^T x + c^T y + \sum_{i \in N} Q_{ii} w_i + \langle\, Q \,,\, F \,\rangle & \\ \text{s.t.} & \text{(1b)} \;,\; \text{(1c)} \;,\; y \in [0,1]^n \;,\; diag(F) \geq 0 & \\ & \begin{bmatrix} 1 & x^T \\ x & F + diag(w) \end{bmatrix} \succeq 0 & \\ & \begin{bmatrix} w_i & x_i \\ x_i & y_i \end{bmatrix} \succeq 0 & i \in N \end{array} \;. \tag{11}$$

If the strong duality property holds for (10) (e.g., if $Q \succ 0$), then solving (11) provides the best possible lower bound and the corresponding optimal solution (x, y). This could be used to compute the optimal δ as in (9), but in fact that is already provided by the dual variables of the $diag(F) \geq 0$ constraint. This is important as typically one does not want to solve (11) at all iterations of an enumerative approach to the original (1), for the large-scale SDP (11) is rather costly to solve. Instead, this is only done once at the root node to compute the "best possible" diagonal, denoted by δ_l, which is then kept fixed throughout the branch and cut (B&C) algorithm. This has been shown [9, 21] to be significantly better than using the diagonal, denoted by δ_s, obtained from the (much cheaper) SDP problem (6); that is, the extra time spent in the SDP is largely compensated by the reduction in B&C time, at least for "hard" instances. Note, however, that as branching occurs the optimal solution (x, y) of the continuous relaxation changes; therefore, "deep down" in the enumeration tree δ_l may no longer be the optimal choice. In fact, in [21] it is reported that using a convex combination of δ_l and δ_s is sometimes preferable.

3 Perspective relaxations for MIQPs with indicator constraints

We now study the problem of computing "good formulations" of (small-scale) MIQPs of the form (1). We start by considering the basic convex 2×2 MIQP with semi-continuous variables

$$\min \left\{ \, q_{11}x_1^2 + 2q_{12}x_1x_2 + q_{22}x_2^2 \; : \; l_iy_i \leq x_i \leq u_iy_i \;\; , \;\; y_i \in \{0, 1\} \quad i \in \{1 \, , \, 2\} \, \right\} \, , \tag{12}$$

where the Hessian is positive semidefinite, i.e., $q_{11} > 0$, $q_{22} > 0$, and $q_{11}q_{22} \geq (q_{12})^2$ (the case where either $q_{11} = 0$ or $q_{22} = 0$ being, clearly, uninteresting). For simplicity we omit linear terms (both in x and in y) in the objective function, which can of course be present, because they would just pass unchanged through the derivation: the perspective function of a linear function is the original function.

We want to derive the tightest possible reformulation of (12), a task for which there is no lack of theory. For instance, we could use the standard RLT [20]. Also, an in-depth study of the polyhedral structure of the set is available in [14]. For our purposes, however, the following simple tools are perhaps better suited.

3.1 The Perspective Reformulation of Alternatives

Let us consider a more abstract setting, where we have a set K of (indices of) *different* finite-dimensional spaces. That is, we see $x \in \mathbb{R}^n$ as partitioned into $x = [x^k]_{k \in K}$; alternatively, $N = \{1, \ldots, n\}$ is partitioned as $N = \cup_{k \in K} N^k$, with $N^k \cap N^h = \emptyset$ for all $k \neq h$ (and each N^k nonempty). We will require each $x^k \in \mathbb{R}^{|N^k|}$ to be either 0 or to live in a compact set; for our purposes we can assume these to be polyhedra, i.e., $\mathcal{P}^k = \{ \, x^k \; : \; A^k x^k \leq b^k \, \}$, although this is not necessary in general. It is well-known that compactness of the $\mathcal{P}^k$ is equivalent to the fact that their recession cones only contain 0, i.e., $\{ \, x^k \; : \; A^k x^k \leq 0 \, \} = \{0\}$. On each $\mathcal{P}^k$ we have a closed convex function $f^k(x^k) + c^k$. We then consider the *alternatives function* in the global space, where only the variables in any one of the subspaces at a time can be different from 0:

$$f(x) = \begin{cases} f^k(x^k) + c^k & \text{if } x^k \in \mathcal{P}^k \text{ and } x^h = 0 \; \forall \, h \in K \setminus \{k\} \\ 0 & \text{if } x = 0 \\ +\infty & \text{otherwise} \end{cases} \quad . \tag{13}$$

Computing the convex envelope $\overline{co}f(x)$ of (13) is a simple task. Let us define B^k as the $n \times |N^k|$ block-structured matrix with $|K|$ blocks, such that all the blocks are zero except for the one corresponding to the variables in N^k which is the identity matrix. Introducing auxiliary variables

$\bar{x} = [\bar{x}^k]_{k \in K}$ and $\theta = [\theta^k]_{k \in K}$, one can just write from the very definition that

$$\begin{aligned} \overline{co}f(x) \;=\; & \min_{\bar{x},\theta} \textstyle\sum_{k \in K} \theta^k f^k(\bar{x}^k) && \text{(14a)} \\ \text{s.t. } & \textstyle\sum_{k \in K} \theta^k \leq 1 \;\;, && \text{(14b)} \\ & \textstyle\sum_{k \in K} \theta^k B^k \bar{x}^k = x && \text{(14c)} \\ & A^k \bar{x}^k \leq b^k \;\;,\;\; \theta^k \geq 0 \qquad\qquad k \in K && \text{(14d)} \end{aligned}$$

In (14c), due to (13), if for $k \neq h$ one had $x_i > 0$ and $x_j > 0$ for $i \in N^k$ and $j \in N^h$, then $f(x) = +\infty$. Hence, the only feasible way to choose $\bar{x}^k$ is for it to only have nonzero values for $i \in N^k$. In other words, (14c) equivalently reads "$\theta^k \bar{x}^k = x^k$ for all $k \in K$", leading to

$$\overline{co}f(x) \;=\; \min \big\{ \textstyle\sum_{k \in K} \theta^k f^k(x^k/\theta^k) \;:\; \sum_{k \in K} \theta^k \leq 1 \;,\; A^k x^k \leq b^k \theta^k \;,\; \theta_k \geq 0 \quad k \in K \big\} \;\;. \qquad (15)$$

In plain words, the convex envelope of the alternatives is just the sum of the individual convex envelopes plus the simplex constraint "$\sum_{k \in K} \theta^k \leq 1$". If θ^k were binary variables, that alone would guarantee that at most one of the alternatives be chosen. This is precisely how we will use the result. We remark that the above development is closely related to the work of [6].

3.2 The 2-dimensional case

Using the above result we can now easily compute the PR of (12). This simply starts with the (somewhat awkward) reformulation

$$\begin{aligned} \min\; & q_{11}(x_1^1)^2 + q_{22}(x_2^2)^2 + q_{11}(x_1^{12})^2 + 2q_{12}x_1^{12}x_2^{12} + q_{22}(x_2^{12})^2 && \text{(16a)} \\ \text{s.t. } & x_i = x_i^i + x_i^{12} \quad , \quad y_i = y^i + y^{12} \qquad i \in \{1\,,\,2\} && \text{(16b)} \\ & l_i y^i \leq x_i^i \leq u_i y^i \;\;,\;\; l_i y^{12} \leq x_i^{12} \leq u_i y^{12} \qquad i \in \{1\,,\,2\} && \text{(16c)} \\ & y^1 + y^2 + y^{12} \leq 1 && \text{(16d)} \\ & y^1,\; y^2,\; y^{12} \in \{0, 1\}\;. && \text{(16e)} \end{aligned}$$

The aim of (16) is apparent: by enumerating all three possible nonzero configurations that the binary variables can take ($[y_1 = 1, y_2 = 0] \equiv [y^1 = 1]$, $[y_1 = 0, y_2 = 1] \equiv [y^2 = 1]$, $[y_1 = y_2 = 1] \equiv [y^{12} = 1]$), we are forcing upon (12) the alternatives structure of (13). Note that (16b) are not really constraints, but rather ways of recovering the value of the original variables given that of the newly introduced ones; in other words, the problem can be rewritten without the original x_i and y_i, substituting them away using (16b). We can now apply (15), obtaining the PR

$$\begin{array}{ll} \min & q_{11}(x_1^1)^2/y^1 + q_{22}(x_2^2)^2/y^2 + \big[\, q_{11}(x_1^{12})^2 + 2q_{12}x_1^{12}x_2^{12} + q_{22}(x_2^{12})^2 \,\big]/y^{12} \\ \text{s.t.} & \text{(16b)} \;\;,\;\; \text{(16c)} \;\;,\;\; \text{(16d)} \;\;,\;\; y^1,\; y^2,\; y^{12} \geq 0\;. \end{array} \qquad (17)$$

Although (17) is of significantly larger dimension than (12), having 11 variables instead of 4, it also has 4 equality constraints that can be used to project away 4 of the variables. This leads to the more compact

$$\begin{aligned} \min\; & \frac{q_{11}(x_1 - x_1^{12})^2}{y_1 - y^{12}} + \frac{q_{22}(x_2 - x_2^{12})^2}{y_2 - y^{12}} + \frac{1}{y^{12}} \begin{bmatrix} x_1^{12} & x_2^{12} \end{bmatrix} \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix} \begin{bmatrix} x_1^{12} \\ x_2^{12} \end{bmatrix} \\ \text{s.t. } & l_i(y_i - y^{12}) \leq x_i - x_i^{12} \leq u_i(y_i - y^{12}) \;\;,\;\; l_i y^{12} \leq x_i^{12} \leq u_i y^{12} \qquad i \in \{1\,,\,2\} \\ & y_1 + y_2 - y^{12} \leq 1 \;\;,\;\; y_1 \geq y^{12},\; y_2 \geq y^{12},\; y^{12} \geq 0\;. \end{aligned}$$

This formulation uses only three variables more than the original one, hence it may be a reasonable starting point to develop solution approaches actually using these ideas. However, let us mention that any modern solver confronted with (17) would probably do the substitutions itself, so we won't really differentiate between the two. Also, an in-depth polyhedral description of the projection of (17) to the space of the original variables has been provided in [14], which therefore could be applied to avoid the introduction of the new variables. This might be useful also in view of the fact that further variables are necessary if the objective function has to be reformulated in terms of conic constraints to pass the above formulation to a (MI-)SOCP solver; that is, (17) have to be rewritten as

$$\begin{array}{ll} \min & w^1 + w^2 + w^{12} \\ \text{s.t.} & \text{(16b)} \;\;,\;\; \text{(16c)} \;\;,\;\; \text{(16d)} \;\;,\;\; y^1,\; y^2,\; y^{12} \geq 0 \\ & w^1 y^1 \geq q_{11}(x_1^1)^2 \;\;,\;\; w^2 y^2 \geq q_{22}(x_2^2)^2 \;\;,\;\; w^{12} y^{12} \geq \left[\begin{array}{cc} x_1^{12} & x_2^{12} \end{array}\right] \left[\begin{array}{cc} q_{11} & q_{12} \\ q_{12} & q_{22} \end{array}\right] \left[\begin{array}{c} x_1^{12} \\ x_2^{12} \end{array}\right] \end{array}$$

which requires three further variables. However, the use of polyhedral techniques to replace the conic formulation, albeit of possible computational interest, would not change the quality of the obtained lower bounds, which is what this paper is mainly aimed at, and therefore its exploration is left for future research.

3.3 Higher dimensions

The above technique can obviously be used to compute convex envelopes in higher dimensions, although of course the size of these grows exponentially fast. For instance, the 3×3 case amounts to enumerating all $2^3 - 1 = 7$ nonempty subsets of the set $t = \{1\,,\,2\,,\,3\}$, i.e., the possible *configurations* $C(t) = 2^t \setminus \emptyset$ ($= \{\,\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\},\{1,2,3\}\,\}$) of the indices of the three binary variables y_1, y_2 and y_3 which have the value 1, excluding the all-0 case. Then, for each $c \in C(t)$ we define one single variable y^c, plus "copies" x_i^c of x_i for all $i \in c$. The x_i^c variables are naturally partitioned in $2^{|t|} - 1 = 7$ variable-length sub-vectors according to the configuration, i.e., $x = [x^c]_{c \in C(t)}$ where $x^c = [x_i^c]_{i \in c}$. We accordingly define the sub-matrices Q^c of the $|t| \times |t|$ ($= 3 \times 3$) Hessian Q of the problem restricted to the indices in c. This finally yields

$$\begin{array}{lll} \min & \sum_{c \in C(t)} \left[\,(x^c)^T Q^c x^c\,\right] / y^c & \\ \text{s.t.} & x_i = \sum_{c \in C(t)\,:\, i \in c} x_i^c & i \in t \\ & y_i = \sum_{c \in C(t)\,:\, i \in c} y^c & i \in t \\ & l_i y^c \leq x_i^c \leq u_i y^c & c \in C(t) \;\;,\;\; i \in c \\ & \sum_{c \in C(t)} y^c \leq 1 & \\ & y^c \in \{0,1\} & c \in C(t) \end{array} \tag{18}$$

Extending (18) to the generic $k \times k$ case is straightforward: one just has to change the definition of t. However, given the combinatorial explosion in the size of the formulation, it is likely that these ideas can only be of practical use (if ever) for very small values of k. This is similar to what happens in the RLT technique [20] of which this is clearly a special case: while a hierarchy can be defined which provides tighter and tighter relaxations as k grows, the size of the corresponding formulations grows so rapidly in k that only $k = 2$, or occasionally $k = 3$, have ever found practical application.

In our case, a further issue has to be considered: the above reformulations only work for small matrices, while the applications require much larger ones (e.g., n in the hundreds). Since it is clearly impractical to develop formulations with $k = n$, the idea is to extend the approach described in §2 to the $k \geq 2$ case. That is, we seek to (approximately) decompose Q as the sum of several $k \times k$ PSD matrices, to each of which the technique can be separately (and, hopefully, efficiently) applied.

4 SDP approaches for finding approximate k×k decompositions

In this section we explore the direct generalization of the approach described in §2. That is, we define the problem of approximately decomposing the PSD matrix Q in (1a) as the sum of (many, much) smaller matrices as an SDP. We explore both the simple approach where only Q is considered, as well as the exact approach where all the constraints in (1) are taken into account to find the "best" possible decomposition. We of course start with the 2×2 case.

Definition 4.1 *For each $(i,j) = p \in P$ let $E^p = [e_i, e_j] \in \mathbb{R}^{n \times 2}$, where as usual e_h is the h-th vector of the canonical basis of $\mathbb{R}^n$. Given a PSD matrix $Q \in \mathbb{R}^{n \times n}$, we say that Q* admits a 2×2 decomposition *(or equivalently is* 2×2-decomposable *or is 2×2D) if and only if the following set of conic (semidefinite) constraints has a solution with respect to the variables $\Pi^p \in \mathbb{R}^{2 \times 2}$ (i.e., Π^p are 2×2 matrices):*

$$Q = \textstyle\sum_{p \in P} E^p \Pi^p (E^p)^T \tag{19a}$$

$$\Pi^p \succeq 0 \qquad\qquad p \in P \tag{19b}$$

Observation 4.1 *Let $p = (i,j) \in P$ and π^{ij}_{ij} be the offdiagonal entry of the 2×2 matrix Π^{ij}. From equation* (19a) *it is straightforward to derive that if Q admits a 2×2D then $\pi^{ij}_{ij} = Q_{ij}$.*

Clearly, it is possible to restrict P to the set of indices of *nonzero* elements of Q, provided that each row/column has at least a nonzero. Indeed, $Q_{ij} = 0$ implies that $\Pi^{ij}_{12} = Q_{ij} = 0$; then, one can also set $\Pi^{ij}_{11} = \Pi^{ij}_{22} = 0$, as the diagonal elements of the Π^p matrices (which must necessarily be non-negative) can always be increased without violating (19b). For the general case where Q has some all-zero row/column (save for the diagonal), consider that the existence of a 2×2D is invariant w.r.t. symmetric reshuffling of the rows and columns of Q: just pre/post multiply each of the blocks by the appropriate permutation matrix. Hence, w.l.o.g. we can assume that

$$Q = \begin{bmatrix} Q_{11} & 0 \\ 0 & Q_{22} \end{bmatrix} , \tag{20}$$

with Q_{22} diagonal, and each row/column in Q_{11} having at least one nonzero off-diagonal. Hence, one can just decompose Q_{11}, and the 2×2D of Q is then obtained by just adding to that Q_{22} (which is diagonal, and therefore trivially 2×2D).

Thus, detecting a 2×2D is a polynomial-time problem. Analogously, it is easy to write as an SDP the problem of extracting "the largest possible decomposition" of Q. Similarly to (6) one may arbitrarily choose a linear objective function in the Π^p variables; alternatively (and, perhaps, more naturally since the problem is already conic anyway) one might consider

$$\min \left\{ \| \Phi \|^2 \,:\, Q = \Phi + \textstyle\sum_{p \in P} E^p \Pi^p (E^p)^T \;\;,\;\; (19b) \;\;,\;\; \Phi \succeq 0 \right\} . \tag{21}$$

Clearly, the optimal value of (21) is zero if an only if (19) has a solution. Any feasible solution to problem (21) can then be used to define a *2×2 Perspective Reformulation* (2×2PR for short) of (1) as follows:

$$\min x^T\Phi x + q^T x + c^T y + \sum_{p=(i,j)\in P}\left[\Pi^p_{11}\frac{(x^{p,i}_i)^2}{y^{p,i}} + \Pi^p_{22}\frac{(x^{p,j}_j)^2}{y^{p,j}} + \frac{(x^{p,p})^T\Pi^p x^{p,p}}{y^{p,p}}\right] \tag{22a}$$

$$\begin{aligned}
\text{s.t. } & (1b)–(1d) \\
& x_i = x^{p,i}_i + x^{p,p}_i \quad , \quad y_i = y^{p,i} + y^{p,p} & p \in P \;\; , \;\; i \in p \quad (22b)\\
& l_i y^{p,i} \leq x^{p,i}_i \leq u_i y^{p,i} \quad , \quad l_i y^{p,p} \leq x^{p,p}_i \leq u_i y^{p,p} & p \in P \;\; , \;\; i \in p \quad (22c)\\
& y^{p,p} + y^{p,i} + y^{p,j} \leq 1 & p = (i,j) \in P \quad (22d)\\
& y^{p,i},\; y^{p,j},\; y^{p,p} \in \{0,1\} \;\; , \;\; y^{p,p} \in \{0,1\} & p = (i,j) \in P \;. \quad (22e)
\end{aligned}$$

Here, the constraints (22b)–(22e) as well as the right-most sum in the objective function (22a) are obtained by replicating (17) and (16e) for each pair $(i,j) \in P$. However, there is no guarantee that the optimal solution to (21) provides the best lower bound in the corresponding 2×2PR, i.e., the continuous relaxation of (22). Yet, as for the one-dimensional case discussed in §2 it is possible to write the problem of finding the 2×2D that provides the best bound by maximizing the continuous relaxation of (22) over all approximate decompositions (Π, Φ). Interchanging max/min then yields

$$\min_{x,y}\; q^T x + c^T y + \max_{\Phi,\Pi}\,\langle\, \Phi \,,\, xx^T \,\rangle + \sum_{p=(i,j)\in P}\left\langle \begin{bmatrix} \frac{(x^{p,i}_i)^2}{y^{p,i}} & 0 \\ 0 & \frac{(x^{p,j}_j)^2}{y^{p,j}} \end{bmatrix} + \frac{x^{p,p}(x^{p,p})^T}{y^{p,p}} \;,\; \Pi^p \right\rangle \tag{23a}$$

$$\begin{aligned}
\text{s.t. } & (1b)–(1c) \quad , \quad (22b)–(22d) \\
& y^{p,i} \;,\; y^{p,j} \;,\; y^{p,p} \in [0,1] & p = (i,j) \in P \quad (23b)\\
& y \in [0,1]^n & (23c)\\
& Q = \Phi + \textstyle\sum_{p\in P} E^p \Pi^p (E^p)^T \;\; , \;\; (19b) \;\; , \;\; \Phi \succeq 0 \;. & (23d)
\end{aligned}$$

One can now proceed as in the one-dimensional case by computing the dual of the inner maximization problem. This is made slightly easier by defining

$$\hat{O}^{12} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \;\; , \;\; \hat{D}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \;\; , \;\; \hat{D}^2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{24}$$

so as to express the equality constraint in (23d) as

$$\begin{array}{ll}
\langle\, \hat{O}^{12} \,,\, \Pi^{ij} \,\rangle + \langle\, O^{ij} \,,\, \Phi \,\rangle = 2Q_{ij} & (i,j) \in P \\
\sum_{j>i}\langle\, \hat{D}^1 \,,\, \Pi^{ij} \,\rangle + \sum_{j<i}\langle\, \hat{D}^2 \,,\, \Pi^{ji} \,\rangle + \langle\, D^i \,,\, \Phi \,\rangle = Q_{ii} & i \in N
\end{array} \quad ,$$

where $O^{ij} = E^{ij}(E^{ij})^T$. The dual of the inner SDP problem then is

$$\begin{aligned}
\min\; & \textstyle\sum_{p\in P} 2Q_p f_p + \sum_{i\in N} Q_{ii} f_i & (25a)\\
\text{s.t. }\; & \textstyle\sum_{p\in P} O^p f_p + \sum_{i\in N} D^i f_i \succeq xx^T & (25b)\\
& \hat{O}^{12} f_p + \hat{D}^1 f_i + \hat{D}^2 f_j \succeq \begin{bmatrix} \frac{(x^{p,i}_i)^2}{y^{p,i}} & 0 \\ 0 & \frac{(x^{p,j}_j)^2}{y^{p,j}} \end{bmatrix} + \frac{x^{p,p}(x^{p,p})^T}{y^{p,p}} \qquad p = (i,j) \in P \;. & (25c)
\end{aligned}$$

When x and y are variables, the conic constraints (25b) and (25c) are nonlinear. However, they can be transformed into linear constraints by introducing auxiliary variables and constraints, as follows:

$$\begin{bmatrix} 1 & x^T \\ x & \sum_{p \in P} O^p f_p + \sum_{i \in N} D^i f_i \end{bmatrix} \succeq 0 \tag{26a}$$

$$\hat{O}^{12} f_p + \hat{D}^1 f_i + \hat{D}^2 f_j \succeq \begin{bmatrix} w_i^p & 0 \\ 0 & w_j^p \end{bmatrix} + W^p \qquad p = (i,j) \in P \tag{26b}$$

$$\begin{bmatrix} w_i^p & x_i^{p,i} \\ x_i^{p,i} & y^{p,i} \end{bmatrix} \succeq 0 \qquad p \in P \;\; , \;\; i \in p \tag{26c}$$

$$\begin{bmatrix} W^p & x^{p,p} \\ (x^{p,p})^T & y^{p,p} \end{bmatrix} \succeq 0 \qquad p \in P \qquad , \tag{26d}$$

with W^p obviously being 2×2 matrices. All in all, (23) then is

$$\begin{array}{ll} \min & q^T x + c^T y + \sum_{p \in P} 2 Q_p f_p + \sum_{i \in N} Q_{ii} f_i \\ \text{s.t.} & \text{(1b)–(1c)} \;\; , \;\; \text{(22b)–(22d)} \;\; , \;\; \text{(23b)–(23c)} \;\; , \;\; \text{(26)} \end{array} \quad , \tag{27}$$

which is a rather “large” SDP as n grows. As we shall see, actually solving (27) can be rather challenging. However, it is poised to produce a tighter lower bound than (6), and our main interest is in evaluating how significant the improvement in bound quality is. We remark that the dual optimal solution of the constraints (26b) provides the optimal 2×2D Π^p, $p \in P$. Extending both phases of the approach—finding the decomposition and defining the corresponding PR—to the $k \times k$ case for generic k is now almost straightforward, mostly boiling down to defining the appropriate notation. However, our results will show that the approach is already extremely demanding for $k = 2$, and likely even more so when k grows larger. Hence, the detailed derivation of the $k \times k$ case is better left to the Appendix A.

5 Exact 2×2 decompositions

In this section, we give a characterization of the PSD matrices having an exact 2×2 decomposition (in the sense of Definition 4.1) and we show how to construct 2×2 decompositions when they exist. We assume $n \geq 2$ throughout, unless otherwise stated.

We begin with the following observation. Recall that a square matrix $A \in \mathbb{R}^{n \times n}$ is *weakly diagonally dominant* (WDD) if $|a_{ii}| \geq \sum_{j : j \neq i} |a_{ij}|$ for every $i \in N$.

Observation 5.1 *If $Q \succeq 0$ is WDD, then Q is 2×2-decomposable.*

Proof: First, since $Q \succeq 0$, we have $Q_{ii} \geq 0$ for all $i \in N$. Next, note that by Observation 4.1 the non-diagonal entry of Π^{ij} must be equal to Q_{ij} in any feasible 2×2D, therefore system (19) has a solution if and only if the following one is feasible:

$$\sum_{j : j \neq i} \pi_i^{ij} = Q_{ii} \qquad i \in N \tag{28a}$$

$$\pi_i^{ij} \pi_j^{ij} \geq Q_{ij}^2 \qquad (i,j) \in P \tag{28b}$$

$$\pi_i^{ij} \geq 0 \;\; , \;\; \pi_j^{ij} \geq 0 \qquad (i,j) \in P \; , \tag{28c}$$

where π_i^{ij} and π_j^{ij} represent the diagonal entries of Π^{ij}.

For every $i \in N$, arbitrarily choose convex multipliers $\{\alpha_i^{ik}\}_{k : k \neq i} \subseteq \mathbb{R}_+$ such that $\sum_{k : k \neq i} \alpha_i^{ik} = 1$, and for each $j \in N$ with $j \neq i$ define π_i^{ij} by

$$\pi_i^{ij} = |Q_{ij}| + \alpha_i^{ij} \Big(Q_{ii} - \sum_{k : k \neq i} |Q_{ik}| \Big) \quad . \tag{29}$$

By weak diagonal dominance of Q and the fact that $Q_{ii} \geq 0$, we have $\pi_i^{ij} \geq 0$. Moreover,

$$\begin{aligned} \pi_i^{ij}\pi_j^{ij} &\geq |Q_{ij}||Q_{ji}| = Q_{ij}^2 \ \ , \text{ and} \\ \textstyle\sum_{l:l\neq i} \pi_i^{il} &= \textstyle\sum_{l:l\neq i} |Q_{il}| + \left(\sum_{l:l\neq i} \alpha_i^{il}\right)\left(Q_{ii} - \sum_{k:k\neq i} |Q_{ik}|\right) \\ &= \textstyle\sum_{l:l\neq i} |Q_{il}| + Q_{ii} - \sum_{k:k\neq i} |Q_{ik}| = Q_{ii} \ \ . \end{aligned}$$

□

Being WDD is sufficient, but not necessary for a PSD matrix to be 2×2-decomposable, as can be seen from the following example:

$$\begin{bmatrix} 2 & 2 & 1 \\ 2 & 5 & 1 \\ 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \ . \tag{30}$$

However, a necessary and sufficient condition may be obtained from a slight relaxation of weak diagonal dominance. A square matrix $A \in \mathbb{R}^{n\times n}$ is *weakly scaled diagonally dominant* (WSDD) if there exists a positive definite diagonal matrix $D \in \mathbb{R}^{n\times n}$ such that DAD is WDD. In other words, A is WSDD if and only if there is some n-dimensional vector $d > 0$ such that

$$|d_i a_{ii} d_i| \geq \textstyle\sum_{j:j\neq i} |d_i a_{ij} d_j| \quad \equiv \quad |a_{ii}|d_i \geq \sum_{j:j\neq i} |a_{ij}|d_j \qquad \forall\, i \in N \ \ . \tag{31}$$

A straightforward calculation shows that the matrix in (30) is WSDD, for example using $D = diag(\,7/4\,,\,1\,,\,3/2\,)$.

There are many (at least 40) known characterizations of WSDD matrices (see for instance [5, 17, 19] and the references therein). In particular, in the following theorem, the equivalences (2) $\iff$ (3) $\iff$ (4) are well-known. Nevertheless, we include the proofs of these implications for completeness, as well as to motivate Proposition 5.1, in which we give an explicit formula for *constructing* 2×2 decompositions when they exist. The proof we provide for the implication (3) $\implies$ (4) is essentially [19, Theorem 11].

Theorem 5.1 *Given $Q \succeq 0$, let $D = diag(Q)$ and $V = Q - D$. If $Q_{ii} > 0$ for all $i \in N$, then the following are equivalent:*

(1) *Q is 2×2-decomposable;*

(2) *$D \pm |V| \succeq 0$, where $|V|_{ij} = |V_{ij}|$ for every $(i,j) \in N \times N$;*

(3) *$\rho(|I - \bar{Q}|) \leq 1$, where $\bar{Q} = D^{-\frac{1}{2}} Q D^{-\frac{1}{2}}$;*

(4) *Q is WSDD.*

Proof: It is straightforward to show that in the case where Q is reducible, each of (1)–(4) holds for Q if and only if it holds for each of its irreducible components. Hence, we can restrict to the case where Q is irreducible. We prove a cycle of implications.

(1) $\implies$ (2): Note that the system (28) is identical for every matrix $\tilde{Q} \in \mathcal{A}(Q)$, where

$$\mathcal{A}(Q) = \left\{\, \tilde{Q} \in \mathcal{S}_n \ : \ diag(\tilde{Q}) = diag(Q) \ \ , \ \ |\tilde{Q}_{ij}| = |Q_{ij}| \ \ \forall\, i, j \in N \,\right\} \ \ ,$$

and $\mathcal{S}_n$ is the set of $n \times n$ (real-valued) symmetric matrices. Hence, if Q is 2×2D, then so is every $\tilde{Q} \in \mathcal{A}(Q)$. As already mentioned, Q being 2×2D immediately implies that $Q \succeq 0$, hence $\tilde{Q} \succeq 0$ for every $\tilde{Q} \in \mathcal{A}(Q)$. Since $D \pm |V| \in \mathcal{A}(Q)$, the result follows.

(2) $\implies$ (3): Suppose that $D \pm |V| \succeq 0$. Then, since $D \succ 0$, we have

$$0 \preceq D^{-\frac{1}{2}}(D \pm |V|)D^{-\frac{1}{2}} = I \pm D^{-\frac{1}{2}}|Q-D|D^{-\frac{1}{2}} = I \pm |\bar{Q}-I| \ ,$$

where $\bar{Q} = D^{-\frac{1}{2}}QD^{-\frac{1}{2}}$. Hence, $\Lambda(|I-\bar{Q}|) \subseteq [-1,1]$, and therefore $\rho(|I-\bar{Q}|) \leq 1$.
(3) $\implies$ (4): Suppose that $\rho(|I-\bar{Q}|) \leq 1$. Since Q is irreducible, so is $|I-\bar{Q}|$; hence, by the Perron-Frobenius Theorem [13, 16], there exists an eigenvalue $\lambda \in \Lambda(|I-\bar{Q}|)$ such that $\lambda = \rho(|I-\bar{Q}|)$; moreover, there is an associated eigenvector $x > 0$. Thus, for every $i \in N$ we have

$$\textstyle\sum_{j:j\neq i} |\bar{Q}_{ij}|x_j = |I-\bar{Q}|_i x = \lambda x_i \leq x_i = \bar{Q}_{ii}x_i \ .$$

Hence, $\bar{Q}$ is WSDD, and therefore so is Q.
(4) $\implies$ (1): Suppose Q is WSDD. Then, there exists a diagonal matrix $U \succ 0$ such that UQU is WDD. By Observation 5.1, UQU then has a 2×2D, say $UQU = \sum_{(i,j)\in P} E^{ij}\bar{\Pi}^{ij}(E^{ij})^T$. Hence, $Q = \sum_{(i,j)\in P} U^{-1}E^{ij}\bar{\Pi}^{ij}(U^{-1}E^{ij})^T$ which can be rewritten as $Q = \sum_{(i,j)\in P} E^{ij}\Pi^{ij}(E^{ij})^T$ where the diagonal entries of Π^{ij} are given by $\pi_i^{ij} := U_{ii}^{-1}\bar{\pi}_i^{ij}U_{ii}^{-1}$ and $\pi_j^{ij} := U_{jj}^{-1}\bar{\pi}_j^{ij}U_{jj}^{-1}$. This completes the proof. □

Remark 5.1 *After the submission of the initial version of our manuscript, it was brought to our attention that the equivalence between weak scaled diagonal dominance and 2×2-decomposability has been independently demonstrated in [3, Lemma 9] (see also [15, Theorem 3.1] and [2, Remark 2.2]).In fact, the equivalence is a natural consequence of results in [5] characterizing the class of positive semidefinite WSDD matrices as the matrices having factor width at most 2.*

In Theorem 5.1, the assumption that $Q_{ii} > 0$ for all $i \in N$ is not restrictive and is only made for convenience. Indeed, if $Q_{ii} = 0$ for some $i \in N$, the fact that $Q \succeq 0$ entails that $Q_{ij} = Q_{ji} = 0$ for every $j \in N$. Hence, we can assume w.l.o.g. that Q can be partitioned as in (20), where all diagonal elements of Q_{11} are nonzero, and $Q_{22} = 0$ (the zero matrix). It is then straightforward to show that Q has a 2×2D if and only if Q_{11} does.

Proposition 5.1 *Under the hypotheses of Theorem 5.1, assume in addition that Q is both 2×2D and irreducible and let (λ, x) be an eigenpair for $|I-\bar{Q}|$ such that $\lambda = \rho(|I-\bar{Q}|)$ and $x > 0$. Arbitrarily choosing $t_i \in [\lambda, 1]$ for each $i \in N$, a 2×2D of Q is given by*

$$\pi_i^{ij} \ = \ \frac{t_i|Q_{ij}|\sqrt{Q_{ii}}}{\lambda\sqrt{Q_{jj}}}x_i^{-1}x_j + \frac{Q_{ii}(1-t_i)}{n-1} \qquad \forall\ i,\ j \in N\ ,\ i \neq j\ . \tag{32}$$

Proof: First note that $\lambda > 0$, since otherwise Q would be diagonal, and therefore reducible, contradicting the hypothesis. Hence, each π_i^{ij} is well-defined. Next, since $0 \leq t_i \leq 1$ for each $i \in N$, we have $\pi_i^{ij} \geq 0$. In addition,

$$\pi_i^{ij}\pi_j^{ij} \geq \frac{t_i|Q_{ij}|\sqrt{Q_{ii}}}{\lambda\sqrt{Q_{jj}}}x_i^{-1}x_j\frac{t_j|Q_{ij}|\sqrt{Q_{jj}}}{\lambda\sqrt{Q_{ii}}}x_j^{-1}x_i = \frac{t_it_j}{\lambda^2}Q_{ij}^2 \geq Q_{ij}^2,$$

since $t_i \geq \lambda$, $t_j \geq \lambda$. Furthermore, for every $i \in N$,

$$\begin{aligned}
\sum_{j:j\neq i} \pi_i^{ij} \ &= \ \sum_{j:j\neq i}\left(\frac{t_i|Q_{ij}|\sqrt{Q_{ii}}x_j}{\lambda\sqrt{Q_{jj}}x_i} + \frac{Q_{ii}(1-t_i)}{n-1}\right) = \left(\frac{t_i\sqrt{Q_{ii}}}{\lambda x_i}\sum_{j:j\neq i}\frac{|Q_{ij}|x_j}{\sqrt{Q_{jj}}}\right) + Q_{ii}(1-t_i) \\
&= \ \frac{t_iQ_{ii}}{\lambda x_i}\left(\sum_{j:j\neq i}\frac{|Q_{ij}|}{\sqrt{Q_{ii}Q_{jj}}}x_j\right) + Q_{ii}(1-t_i) = \frac{t_iQ_{ii}}{\lambda x_i}|I-\bar{Q}|_i x + Q_{ii}(1-t_i) \\
&= \ t_iQ_{ii} + Q_{ii}(1-t_i) = Q_{ii} \ .
\end{aligned}$$

□

In the case where Q is 2×2-decomposable but reducible, (32) is not necessarily well-defined, since we may have $x_i = 0$ for some $i \in N$. However, in this case Q can be brought, by symmetric exchanges of rows and columns, to a block-diagonal form with k diagonal blocks Q^h, $h = 1, \dots, k$ (cf. (20)), each of which is irreducible. Then, (32) can be applied to each of the blocks Q^h separately, where (x, λ) is the eigenpair associated with $|I - diag(Q^h)^{-\frac{1}{2}} Q^h diag(Q^h)^{-\frac{1}{2}}|$.

It is also possible to characterize when the 2×2D obtained by (32) is unique:

Corollary 5.1 *Under the hypotheses of Theorem 5.1, assume in addition that Q is irreducible. Then Q has a unique 2×2D if and only if $\rho(|I - \bar{Q}|) = 1$.*

The proof of this result is somewhat long, and so it is deferred to Appendix B.

To conclude this section we comment on the connection between our results and the well-known fact that any PSD matrix Q can be written as a non-negative combination of at most n rank-1 PSD matrices, i.e., $Q = \sum_{i \in N} \lambda_i x_i x_i^T$. For example, one may choose $\{x_i\}_{i \in N} \subset \mathbb{R}^n$ to be any orthonormal basis of eigenvectors for Q, and $\{\lambda_i\}_{i \in N} \subset \mathbb{R}_+$ to be the corresponding eigenvalues. If Q has a 2×2D, then it can also be written as the sum of $O(n^2)$ *sparse* matrices. Interestingly, it is always possible to choose the terms of the 2×2D so that, besides being sparse, they are also rank-1. We remark that this result was arrived at independently (and via a shorter proof) in [3, Lemma 9].

Proposition 5.2 *Let $n \geq 3$. For any $n \times n$ 2×2-decomposable matrix Q there exists a 2×2D such that $rank(\Pi^{ij}) \leq 1$ for all $(i, j) \in P$.*

Again, the proof of this result is deferred to Appendix B. It can be seen that the rank-1 2×2D for a given Q is not necessarily unique. For example, it is straightforward to check that the rank of every block in the decomposition (30) is equal to 1. Yet, another possible rank-1 decomposition is:

$$\begin{bmatrix} 2 & 2 & 1 \\ 2 & 5 & 1 \\ 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} \frac{4}{3} & 2 & 0 \\ 2 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} \frac{2}{3} & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & \frac{3}{2} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & \frac{1}{2} \end{bmatrix} .$$

This is tied to the fact that one can choose different objective functions f in (47) (cf. Appendix B), leading to different rank-1 decompositions.

6 Heuristic approaches for approximate 2×2 decomposition

We now combine the results of §5 with those of §2 to propose fast heuristics for finding approximate 2×2Ds of a matrix Q without the need of solving large SDP problems like (21). The observation is that for any $Q \succeq 0$, we know:

- how to select the "largest" diagonal $D \succeq 0$ such that $Q - D \succeq 0$, which just amounts to solving the "small" SDP (6), with any choice of α (or even using $||Q - diag(\delta)||^2$ as the objective function);
- the quick formula (32), just requiring a largest eigenvalue computation, which gives us an exact 2×2D for Q, whenever one exists.

In other words, we want to write $Q = R + X$, where X is 2×2D and $R \succeq 0$, so that R—the *remainder*—is as small as possible. If $\rho(|I - \bar{Q}|) \leq 1$, i.e., Q is 2×2D, we can quit immediately

($X = Q$, $R = D = 0$). Otherwise we can exploit the (obvious) fact that a diagonal matrix is surely 2×2D. We therefore restrict R to have the form

$$R(\varepsilon) = \varepsilon(Q - D)$$

for $\varepsilon \geq 0$ and any fixed D such that $R(1) = Q - D \succeq 0$ (for instance, but not necessarily, the optimal solution to (6)). This choice immediately implies that $R(\varepsilon) \succeq 0$ for any $\varepsilon \geq 0$. Furthermore,

$$X(\varepsilon) = Q - R(\varepsilon) = (1 - \varepsilon)Q + \varepsilon D \ . \tag{33}$$

Clearly, $X(1)$ is 2×2D. Also, for any ε we have a quick way to detect whether or not $X(\varepsilon)$ is 2×2D. Hence we can just do a binary search with $\varepsilon \in [0, 1]$ to look for the smallest ε such that $X(\varepsilon)$ is 2×2D. Having found this ε^*, we then use $X(\varepsilon^*)$ to generate the 2×2D via (32), shouldering the remainder $R(\varepsilon^*)$. The process is independent of how D is computed, only provided that $Q - D \succeq 0$. This is relevant, because one can use both δ_s and δ_l (cf. (9)/(8)).

Interestingly, the process can be iterated. Basically, one is exploring the space of all pairs (X, R) such that $Q = X + R$, X is 2×2D, and $R \succeq 0$, among which we surely know the trivial one $(0, Q)$. Given any feasible pair (X, R), we can easily compute a diagonal D (e.g. by solving (6) with $Q = R$) such that $R - D \succeq 0$: this means that $(X + D, R - D)$, is another feasible pair, clearly better than the previous one (if $D \neq 0$). In other words, we can make an improving step in the direction $(D, -D)$; from there we take a step in the direction $(Q, 0)$ with the above idea, i.e., finding the smallest ε such that $X(\varepsilon)$ is 2×2D. We note, however, that with "obvious" choices of D the process does not iterate long. In fact, assume that D has been obtained by solving (6). Because $R(\varepsilon^*) = \varepsilon^*(Q - D)$, if one solves (6) again with $Q = R(\varepsilon^*)$, clearly the optimal solution can only be $\delta = 0$. Indeed, assume by contradiction that a $D' \neq 0$ such that $\varepsilon^*(Q - D) - D' \succeq 0$ exists: this means that $\varepsilon^*(Q - D - (1/\varepsilon^*)D') \succeq 0$, i.e., $D + (1/\varepsilon^*)D'$ was feasible for (6) before. But, clearly, $D + (1/\varepsilon^*)D'$ is a better solution to (6) than D, whatever reasonable objective function one chooses. Despite this, the approach is able in some cases to find very good solutions in a short time (as shown in the next section).

Although the above heuristic is very efficient in practice (cf. §7), we remark that by changing the requirement that each $X(\varepsilon)$ be 2×2D (hence WSDD) to the stricter requirement that each $X(\varepsilon)$ be WDD, an explicit formula can be obtained for the optimal value of ε that does not require any eigenvalue computations (although at the possible expense of the quality of the bound on (1)). Indeed, by (33) $X(\varepsilon)$ is WDD if and only if

$$\varepsilon d_{ii} + (1 - \varepsilon)Q_{ii} \geq (1 - \varepsilon)\textstyle\sum_{j \neq i} |Q_{ij}| \quad \forall\, i \in N \ . \tag{34}$$

Let $N^- = \{\, i \in N \ : \ v_i := \sum_{j \neq i} |Q_{ij}| - Q_{ii} > 0 \,\}$; if $N^- = \emptyset$ then $Q = X(0)$ is WDD, and hence 2×2D already. Otherwise, since $d_{ii} \geq 0$ for each $i \in N$, (34) holds if and only if

$$\varepsilon d_{ii}/v_i \geq 1 - \varepsilon \quad \forall\, i \in N^- \ .$$

Solving the above for ε, we obtain

$$X(\varepsilon) \text{ is WDD} \quad \equiv \quad \varepsilon \geq 1/(\,1 + \gamma\,) \quad \text{where} \quad \gamma = \min\, \left\{\, d_{ii}/v_i \ : \ i \in N^- \,\right\} \ .$$

Taking $\varepsilon = 1/(1 + \gamma)$, a 2×2D for $X(\varepsilon)$ may be obtained immediately via the formula (29).

We remark that in [1], an alternative decomposition-based approach has been independently developed for finding convex relaxations of (1) which has some features in common with the approach we have outlined in this section. In [1], the authors observed that the quadratic form associated with an arbitrary symmetric matrix Q may be decomposed as follows:

$$x^T Q x = \textstyle\sum_{i=1}^n \left(Q_{ii} - \sum_{j \neq i} |Q_{ij}| \right) x_i^2 + \sum_{i=1}^n \sum_{j=i+1}^n |Q_{ij}| (x_i + sgn(Q_{ij}) x_j)^2 \ . \tag{35}$$

We note that (35) may be obtained from the following decomposition of Q:

$$Q = diag(z) + \sum_{p=(i,j)\in P} E^p \left[\begin{array}{cc} |Q_{ij}| & Q_{ij} \\ Q_{ij} & |Q_{ij}| \end{array} \right] (E^p)^T \ \ , \tag{36}$$

where z is the n-dimensional vector such that $z_i = Q_{ii} - \sum_{j\neq i} |Q_{ij}|$ for each $i \in N$. In [1], a convex relaxation of (a special case of) (1) is obtained by taking a sum of the convex envelopes of each of the (1 and 2-dimensional) terms in (35) with respect to the feasible set of (1).

7 Computational results and conclusions

We now present computational results aimed at assessing whether formulations employing the 2×2PR, i.e., (22), have the potential of improving lower bounds w.r.t. state-of-the-art ones using the PR of the diagonal terms only. For this we have used the Mean-Variance portfolio optimization problem already employed in [8, 9, 10, 11, 12, 21], and available at

`http://www.di.unipi.it/optimize/Data/MV.html .`

The interested reader is referred to these references for details on how the instances were generated, as well as the behaviour of solution approaches based on the PR with diagonal terms only (the covariance matrix Q is often estimated using a *factor model*, i.e., $Q = D + L$, where D is a non-negative diagonal matrix and L is a low-rank PSD matrix, see e.g. [7]). However, we could not use the same instances used there (with $n \in \{200, 300, 400\}$) as some of the approaches do not scale to those sizes; instead, we considered instances with $n \in \{25, 50\}$. We constructed instances mirroring those used in the literature, i.e., (initially) with three different kinds of matrices Q: diagonally dominant ones ("p", or "+" instances), almost but not quite diagonally dominant ones ("z", or "0" instances), and strongly not diagonally dominant ones ("n", or "−" instances). All those instances had $Q > 0$; we also found that matrices with negative (off-diagonal) elements behaved quite differently. Hence, for each instance we produced a second instance by changing the sign of all the off-diagonal elements of Q. These are SDP when the original matrix is of type "p", but not necessarily when it is of type "z" or "n"; hence, the matrices were, whenever necessary, corrected by adding to them the smallest possible diagonal that restored $Q \succeq 0$, *à la* (6). We denote by "o", "y" and "m" the instances thusly produced starting from, respectively, "p", "z" and "n" ones. For each type we produced 10 different instances by changing the seed of the random generator.

We tested 6 different approaches:

1. D_s and D_l denote the bounds obtained by the diagonal <u>PR</u> (5) when δ is obtained by solving (6) and (7), respectively;

2. $2{\times}2_s$ and $2{\times}2_l$ denote the bounds obtained by the <u>2×2PR</u> (the continuous relaxation of (22)) when the 2×2D is obtained by solving (21) and (27), respectively;

3. $2{\times}2^h_s$ and $2{\times}2^h_l$ denote the bounds obtained by the <u>2×2PR</u> (the continuous relaxation of (22)) when the 2×2D is obtained from the heuristic of §6, starting from the diagonal D_s and D_l, respectively.

Tables 1 and 2 report, for each approach, the gap of the corresponding relaxation w.r.t. the optimal integer solution (in percentage) and the running time (in seconds) required to compute it. A number of comments is necessary:

- The optimal integer solution has been obtained by running a B&C solver (in particular, `Cplex 12.7`) on the diagonal PR (5), with the D$_l$ diagonal. While "p", "z" and "n" instances can be solved quickly (cf. e.g. [8, 9] for much larger sizes), the same does not hold for the new "o", "y" and "m" instances, some of which required many days of CPU time to be solved because of the rather weak bounds. Since the optimal solution was obtained with high accuracy (`1e-4` relative), the gaps have been computed as $(ub - lb)/ub$, where ub is the value of the best feasible solution produced by the B&C, and lb the lower bound produced by each $\underline{\text{PR}}$. Most often the gap is rather computed as $(ub - lb)/lb$, which is the safe option when ub can be arbitrarily far from the optimal value. In our case, however, the formula is sensible because ub is very close to the optimal value. Furthermore, this makes comparison between the different lower bounds more accurate. Finally, as the tables will show, for several instances even the best bounding procedure returns (the very weak) $lb = 0$; with the formula we adopted this gives a gap of $1 = 100\%$, whereas the standard formula would be ill-defined.

- Presenting the results is not trivial, because they have a very large variance not only between different classes of instances, but even within the same class. Hence, reporting averages is not a feasible option, as they would hide too much detail; on the other hand, for space reasons we cannot report results for all instances. The compromise we reached is to present individual results, but only for half of the instances (the "odd ones"); the results on the other half are completely analogous, and would not change the overall picture.

- Computing the bound is typically a two-stage process: first either the diagonal or the 2×2D is computed, and then the $\underline{\text{PR}}$ using them is solved. In the tables, "t." is the time for the first, and "tb." that for the second. The exception is $2{\times}2_l$, where one computes at the same time both (but at a rather astonishing price). Also, $2{\times}2_s^h$ and $2{\times}2_l^h$ first compute a diagonal, and then the 2×2D out of it; for these cases, "t." refers to the latter phase, as the time for computing the diagonal can be found in the corresponding columns D_s and D_l.

- Running times are on Intel Core i7@2.5 Ghz. The bound is computed solving ($\underline{5}$) or ($\underline{22}$) formulated as SOCP using `Cplex 12.7`, while the SDP problems are solved with `SeDuMi 1.1`. In some cases, these choice may not be the ones attaining the best performances. For instance, the best solver for computing D_s, according to [11], is DSDP. Also, the SOCP formulation is often not the best approach to compute the $\underline{\text{PR}}$ in the diagonal case, as shown in [8, 9, 12, 21]. However, the running times of ($\underline{22}$), and even more of (27), are so large as to make it hardly relevant to discuss the best solution approaches for the diagonal cases. Furthermore, always using the same solution approaches improves consistency of the results.

Table 1 and 2 paint an uncharacteristically varied panorama. The only constant results are:

- The gaps provided by $2{\times}2_s^h$, are almost always identical, or at least extremely close, to those obtained by $2{\times}2_l^h$ (using D$_l$); when the former provides better gaps the difference is minor, and in a few cases (eg., 25-n-i and 50-z-i) the converse actually happens.

- Most often, the gaps provided by the (very cheap) heuristic, basically irrespective of the choice of the starting diagonal, are identical, or at least extremely close, to those of the much more costly $2{\times}2_s$. Sometimes the SDP-based approach produces visibly better gaps (e.g., 25-n-i and 50-n-i), but when this happens the bound is typically not the best one. Although we don't report them, we can add that the objective function values (in the

Table 1: Results with different decompositions for $n = 25$.

	D_s			D_l			$2\times2_s$			$2\times2_s^h$			$2\times2_l^h$			$2\times2_l$	
	t.	tb.	gap	t.	tb.	gap	t.	tb.	gap	t.	tb.	gap	t.	tb.	gap	t.	gap
25-p-a	0.28	0.04	2.28	0.75	0.04	2.21	5.12	1.14	2.21	3.8e-3	1.81	2.21	2.1e-4	2.24	2.21	6.5e+3	2.21
25-p-c	0.28	0.03	2.48	0.65	0.03	2.36	5.94	1.33	2.30	3.2e-4	1.94	2.30	2.0e-4	2.49	2.30	1.0e+4	2.30
25-p-e	0.28	0.03	1.83	0.63	0.03	1.46	5.82	1.21	3.70	2.9e-4	1.90	3.70	2.2e-4	2.41	3.70	1.1e+4	1.29
25-p-g	0.28	0.03	2.16	0.65	0.03	1.75	5.35	0.94	1.74	2.5e-4	1.93	1.74	2.1e-4	2.55	1.74	8.2e+3	1.71
25-p-i	0.29	0.03	0.71	0.64	0.03	0.56	5.28	0.76	2.33	2.5e-4	1.67	2.33	2.1e-4	2.42	2.33	9.3e+3	0.55
25-o-a	0.51	0.03	3.95	0.68	0.03	2.83	7.31	1.21	2.81	4.2e-3	2.49	2.81	1.2e-3	2.33	2.81	2.5e+3	2.80
25-o-c	0.93	0.03	8.23	0.61	0.03	5.09	7.34	1.17	2.48	4.0e-4	2.67	2.48	7.1e-4	2.55	2.48	2.4e+3	2.47
25-o-e	0.33	0.03	17.26	0.61	0.03	15.30	7.04	0.83	8.90	2.7e-4	2.42	8.90	4.0e-3	2.24	8.90	1.4e+3	8.90
25-o-g	0.29	0.03	10.33	0.63	0.03	8.11	6.95	0.93	3.06	2.5e-4	2.67	3.06	3.3e-4	2.47	3.06	2.2e+3	3.05
25-o-i	0.29	0.02	14.93	0.65	0.02	13.65	7.24	0.83	6.58	3.7e-4	2.14	6.58	2.2e-4	2.14	6.58	1.4e+3	6.58
25-z-a	0.29	0.04	2.39	0.65	0.03	1.84	5.63	0.71	16.18	7.8e-3	2.42	16.26	5.6e-3	2.22	16.26	1.6e+4	1.50
25-z-c	0.29	0.04	3.06	0.64	0.03	2.44	5.78	0.60	15.27	5.7e-3	2.82	15.83	5.3e-3	1.81	15.72	1.9e+4	1.64
25-z-e	0.29	0.03	0.02	0.67	0.03	0.00	5.66	1.21	0.07	4.8e-3	1.50	0.07	5.2e-3	1.65	0.08	1.4e+4	0.00
25-z-g	0.29	0.03	1.55	0.66	0.03	1.32	7.70	1.75	1.33	5.2e-3	3.48	1.39	5.6e-3	2.59	1.55	1.4e+4	1.21
25-z-i	0.29	0.03	0.98	0.65	0.03	0.81	6.05	1.09	0.97	4.6e-3	2.65	1.00	5.0e-3	2.01	0.98	1.4e+4	0.78
25-y-a	0.38	0.01	100.0	0.78	0.01	100.0	8.17	0.72	100.0	2.4e-4	1.27	100.0	2.3e-4	1.20	100.0	1.2e+3	100.0
25-y-c	0.36	0.01	100.0	0.73	0.01	100.0	8.04	0.81	100.0	2.4e-4	1.26	100.0	3.0e-4	1.26	100.0	1.4e+3	100.0
25-y-e	0.36	0.03	2.14	0.69	0.01	2.10	8.51	1.11	0.32	2.5e-4	1.32	0.32	2.4e-4	1.32	0.32	3.3e+3	0.21
25-y-g	0.36	0.04	24.57	0.63	0.04	24.57	8.66	1.42	9.14	2.3e-4	2.28	9.14	2.2e-4	2.29	9.14	3.5e+3	8.71
25-y-i	0.36	0.03	15.96	0.66	0.04	15.97	8.28	1.07	3.49	2.3e-4	1.57	3.49	1.8e-4	1.75	3.49	2.6e+3	1.61
25-n-a	0.29	0.02	2.79	0.65	0.03	1.78	7.91	1.31	11.97	4.9e-3	1.75	12.10	5.4e-3	4.10	11.24	1.5e+4	0.27
25-n-c	0.28	0.03	3.04	0.65	0.02	2.04	8.05	1.21	12.12	5.1e-3	2.04	13.04	4.9e-3	1.84	12.66	1.4e+4	1.42
25-n-e	0.28	0.02	2.00	0.63	0.03	1.39	8.34	1.04	7.94	5.3e-3	1.71	9.81	5.2e-3	1.79	9.74	1.8e+4	0.38
25-n-g	0.29	0.03	4.24	0.63	0.03	4.08	7.25	1.04	4.19	6.2e-3	2.00	4.23	5.5e-3	1.76	4.12	1.4e+4	4.06
25-n-i	0.28	0.03	2.68	0.59	0.03	1.37	8.82	1.81	8.60	5.2e-3	1.74	9.15	6.1e-3	2.78	9.43	1.7e+4	0.35
25-m-a	2.69	0.03	100.0	0.67	0.01	100.0	8.38	0.81	100.0	1.0e-3	1.35	100.0	2.6e-4	1.45	100.0	1.2e+3	100.0
25-m-c	0.57	0.03	99.55	0.69	0.03	99.55	8.74	1.20	98.59	3.0e-4	1.73	98.59	1.9e-4	1.97	98.59	1.5e+3	98.59
25-m-e	1.13	0.02	100.0	0.67	0.01	100.0	7.89	0.71	100.0	5.2e-4	1.20	100.0	2.0e-4	1.23	100.0	1.3e+3	100.0
25-m-g	0.41	0.04	5.32	0.65	0.03	5.32	8.96	1.30	4.79	2.8e-4	1.69	4.79	2.0e-4	1.78	4.79	3.0e+3	4.66
25-m-i	0.38	0.02	100.0	0.65	0.02	100.0	7.95	0.67	100.0	2.5e-4	1.30	100.0	1.9e-4	1.39	100.0	1.3e+3	100.0

Table 2: Results with different decompositions for $n = 50$.

	D_s			D_l			$2{\times}2_s$			$2{\times}2_s^h$			$2{\times}2_l^h$			$2{\times}2_l$	
	t.	tb.	gap	t.	tb.	gap	t.	tb.	gap	t.	tb.	gap	t.	tb.	gap	t.	gap
50-p-a	0.38	0.07	1.47	0.98	0.06	1.23	172.91	13.47	1.20	1.0e-3	39.33	1.20	6.5e-4	39.10	1.20	1.6e+5	1.20
50-p-c	0.31	0.06	2.45	0.95	0.07	2.16	182.99	12.60	2.11	8.6e-4	27.51	2.11	6.7e-4	38.13	2.11	1.6e+5	2.11
50-p-e	0.32	0.06	0.89	0.92	0.06	0.33	173.14	13.08	1.40	7.8e-4	41.42	1.40	7.5e-4	43.97	1.40	2.0e+5	0.26
50-p-g	0.32	0.05	0.83	0.97	0.05	0.73	169.28	14.77	0.72	7.7e-4	45.43	0.72	7.5e-4	43.98	0.72	2.2e+5	0.68
50-p-i	0.32	0.06	1.33	0.95	0.05	1.24	164.57	15.81	1.25	7.6e-4	36.86	1.25	7.0e-4	38.26	1.25	2.6e+5	1.24
50-o-a	0.38	0.06	18.17	0.99	0.14	14.26	172.69	11.25	3.31	7.5e-4	43.13	3.31	7.7e-4	43.69	3.31	1.6e+5	3.31
50-o-c	0.36	0.07	15.71	0.96	0.08	9.32	172.75	10.25	2.44	7.7e-4	42.01	2.44	7.4e-4	39.08	2.44	1.8e+5	2.44
50-o-e	0.34	0.07	21.84	0.96	0.06	17.18	176.68	15.03	7.95	7.9e-4	57.43	7.95	7.4e-4	57.04	7.95	1.4e+5	7.95
50-o-g	0.32	0.06	19.84	0.91	0.06	15.90	175.49	10.04	6.32	6.7e-4	39.26	6.32	7.2e-4	40.58	6.32	1.0e+5	6.31
50-o-i	0.36	0.06	5.24	0.92	0.07	2.35	174.82	12.99	2.27	7.5e-4	33.67	2.27	6.4e-4	28.75	2.27	1.9e+5	2.27
50-z-a	0.32	0.07	0.46	0.98	0.06	0.30	220.37	16.64	0.41	2.1e-2	39.13	0.45	1.8e-2	31.77	0.42	2.4e+5	0.27
50-z-c	0.32	0.06	2.79	1.02	0.06	1.53	229.32	10.11	10.28	2.4e-2	34.27	11.61	2.2e-2	21.71	11.21	2.8e+5	0.62
50-z-e	0.32	0.07	2.86	0.96	0.06	1.16	193.93	8.39	10.97	1.9e-2	38.50	11.45	2.5e-2	22.97	11.30	3.5e+5	0.51
50-z-g	0.39	0.06	3.46	0.94	0.07	2.52	213.51	7.66	11.62	2.1e-2	42.13	12.32	2.3e-2	21.62	12.16	2.2e+5	2.24
50-z-i	0.32	0.06	2.92	0.97	0.07	1.79	226.16	12.10	3.65	2.2e-2	40.59	4.72	1.9e-2	28.12	4.83	2.2e+5	1.36
50-y-a	0.46	0.03	8.80	0.99	0.11	8.80	207.35	15.44	0.16	7.6e-4	69.33	0.16	6.1e-4	67.33	0.16	2.6e+5	0.14
50-y-c	0.43	0.03	99.76	0.97	0.07	99.76	204.27	15.13	98.09	8.4e-4	132.16	98.09	6.9e-4	139.86	98.09	8.6e+4	98.09
50-y-e	0.46	0.03	100.0	0.99	0.02	100.0	190.83	7.56	100.0	9.8e-4	34.98	100.0	6.5e-4	37.44	100.0	8.9e+4	100.0
50-y-g	0.47	0.04	100.0	1.00	0.02	100.0	206.13	8.39	100.0	8.9e-4	37.45	100.0	8.1e-4	39.14	100.0	7.9e+4	100.0
50-y-i	0.45	0.05	94.62	1.06	0.07	94.62	202.14	16.70	73.27	7.7e-4	65.71	73.27	8.3e-4	162.52	73.27	1.6e+5	73.26
50-n-a	0.33	0.06	3.99	0.99	0.10	2.83	252.97	13.72	2.60	1.8e-2	43.35	3.97	2.0e-2	28.47	3.24	3.2e+5	2.43
50-n-c	0.32	0.07	4.67	1.01	0.07	2.85	258.48	13.38	7.68	2.3e-2	47.34	10.85	1.9e-2	26.42	9.89	2.8e+5	1.05
50-n-e	0.32	0.10	3.12	0.99	0.06	1.54	240.48	9.62	9.23	2.1e-2	43.33	11.90	2.5e-2	20.50	10.85	3.0e+5	0.36
50-n-g	0.32	0.07	3.04	1.01	0.08	1.66	247.21	13.88	1.51	3.6e-2	44.48	2.56	2.7e-2	27.39	1.68	2.4e+5	1.45
50-n-i	0.32	0.12	3.92	0.98	0.07	1.98	258.87	14.72	6.75	2.1e-2	44.46	10.10	2.2e-2	26.78	9.07	2.6e+5	0.57
50-m-a	0.48	0.06	59.97	1.04	0.09	59.97	201.50	15.55	15.47	7.5e-4	68.26	15.47	7.6e-4	82.80	15.47	2.2e+5	15.38
50-m-c	0.48	0.11	100.0	0.96	0.07	100.0	200.48	6.85	100.0	7.6e-4	75.53	100.0	9.6e-4	118.41	100.0	8.9e+4	100.0
50-m-e	0.45	0.03	100.0	0.94	0.08	100.0	192.65	6.76	100.0	1.2e-3	53.00	100.0	7.0e-4	130.69	100.0	9.5e+4	100.0
50-m-g	0.49	0.14	36.35	0.96	0.09	36.35	206.09	17.69	6.03	7.6e-4	114.13	6.03	8.4e-4	89.59	6.03	2.9e+5	6.02
50-m-i	0.45	0.07	98.51	1.04	0.07	98.51	200.08	14.65	90.22	8.1e-4	65.97	90.22	9.3e-4	110.19	90.22	1.4e+5	90.22

sense of (21)) of the 2×2D produced by the heuristics are almost indistinguishable from those of the 2×2D of (21).

Also, as expected D_l is always at least as good as D_s, and $2\times2_l$ is always at least as good as the other two 2×2D. Other than that, almost all cases show up. The "optimal" (and extremely costly to obtain) $2\times2_l$ can be barely distinguishable from the "optimal" D_l, and even from the very cheap D_s. It can also be quite close to the other two 2×2D; in some cases, *all* the approaches provide extremely weak bounds. In some cases, $2\times2_l^h$ and $2\times2_s$ are much better than both D_s and D_l; in other cases they are very significantly worse. In some cases $2\times2_l$ is visibly but not dramatically better than the other two 2×2D, in others the gap is abysmal. In general, the difference between the "optimal" choices $\mathrm{D}_l/2\times2_l$ and the corresponding "cheap" ones $\mathrm{D}_s/2\times2_s$ can be anywhere between negligible and humongous.

It is therefore difficult, at this stage, to draw significant conclusions about when using 2×2D could be promising for computationally solving convex MIQPs with semi-continuous variables. The only clear result is that finding the 2×2D by standard SDP approaches does not appear to be feasible for problems of even moderate size. The time for solving ($\underline{22}$) is also rather too large for comfort. Yet, improving the time for solving the $\underline{2\times2\mathrm{PR}}$ and/or the SDP is conceptually possible: both have been done for the diagonal case [11, 8]. Furthermore, the results clearly show that, under the right circumstances, the approach can yield significantly stronger bounds than the best ones available so far. In this sense, our results look more promising than those reported in [14], which were limited to only one class of tridiagonal matrices. Actually making use of those bounds would require overcoming substantial hurdles, relative to both efficiently finding "good" 2×2D (as, in several cases, those provided by the heuristic are not so), and efficiently solving the $\underline{2\times2\mathrm{PR}}$ once this is done. Both aspects are nontrivial, but conceptually possible. Therefore, we believe it is fair to state that the idea proposed in this work warrants further investigation.

A Approximate decompositions in higher dimensions

Similarly to §3.3 we mainly discuss the case of finding 3×3 decompositions, but the arguments can be extended in a straightforward way to the general $k \times k$ case. As already noted, and confirmed by our computational experience (cf. §7), "large" values of k are unlikely to be of any practical significance.

The starting point is just defining the set $T = \{\, (i,j,k) \in N \times N \times N \;:\; i < j < k \,\}$ of all possible triples. To each $t \in T$ we then associate the $n \times |t|$ $(= n \times 3)$ matrix $E^t = [e_i, e_j, e_k]$ and the $|t| \times |t|$ $(= 3\times3)$ matrix Γ^t, which immediately defines the analogous of (21)

$$\min\Big\{\, \|\,\Phi\,\|^2 \;:\; Q = \Phi + \textstyle\sum_{t\in T} E^t\Gamma^t(E^t)^T \;\;,\;\; \Gamma^t \succeq 0 \quad t \in T \;\;,\;\; \Phi \succeq 0 \,\Big\} \;. \tag{37}$$

Any feasible solution of (37) is an approximate 3x3D of Q, which is an exact 3x3D if and only if the optimal value is 0. Given any approximate 3x3D, using the notation defined in §3.3 we can easily define the corresponding 3x3PR:

$$\begin{array}{lll}
\min\ x^T \Phi x + q^T x + c^T y + \sum_{t \in T} \sum_{c \in C(t)} (x^{t,c})^T (\Gamma^t)^c x^{t,c} / y^{t,c} & & (38a) \\
\text{s.t. } (1b) - (1d) & & \\
\quad x_i = \sum_{c \in C(t) \,:\, i \in c} x_i^{t,c} \quad , \quad y_i = \sum_{c \in C(t) \,:\, i \in c} y^{t,c} & t \in T \ \ , \ \ i \in N & (38b) \\
\quad l_i y^{t,c} \leq x_i^{t,c} \leq u_i y^{t,c} & t \in T \ \ , \ \ c \in C(t) \ \ , \ \ i \in c & (38c) \\
\quad \sum_{c \in C(t)} y^{t,c} \leq 1 & t \in T & (38d) \\
\quad y^{t,c} \in \{0, 1\} & t \in T \ \ , \ \ c \in C(t) & (38e)
\end{array}$$

As in §3.3, in (38a) $(\Gamma^t)^c$ denotes the submatrix of Γ^t restricted to the rows and columns corresponding to the indices in c. For instance, for $t = (i, j, k)$, $(\Gamma^t)^{\{i\}} = \Gamma^t_{11}$, $(\Gamma^t)^{\{i,j\}}$ is the (i, j)-th principal 2×2 submatrix of Γ^t, and $(\Gamma^t)^t = \Gamma^t$. Combining (37) with (38) again yields the problem of finding the 3x3D providing the best bound. This again starts with the following min-max analogous to (23)

$$\begin{array}{ll}
\min\limits_{x,y}\ q^T x + c^T y + \max\limits_{\Phi,\Gamma}\ \langle\, \Phi \,,\, x x^T \,\rangle + \sum\limits_{t \in T} \left\langle \sum\limits_{c \in C(t)} \dfrac{\bar{x}^{t,c} (\bar{x}^{t,c})^T}{y^{t,c}} \ , \ \Gamma^t \right\rangle & (39a) \\
\text{s.t. } (1b)\text{–}(1c) \quad , \quad (38b)\text{–}(38d) & \\
\quad y^{t,c} \in [0, 1] \quad t \in T \ \ , \ \ c \in C(t) & (39b) \\
\quad y \in [0, 1]^n & (39c) \\
\quad Q = \Phi + \sum_{t \in T} E^t \Gamma^t (E^t)^T \ \ , \ \ \Gamma^t \succeq 0 \quad t \in T \ \ , \ \ \Phi \succeq 0 & (39d)
\end{array}$$

where $\bar{x}^{t,c}$ in (39a) denotes the $|c|$-vector $x^{t,c}$ extended to a $|t|(= 3)$-vector by filling it with zeroes for the indices $i \notin c$. For any $i \in c$ and $j \in c$ we also define the $|t| \times |t|$ $(= 3{\times}3)$ matrices $D^{t,i}$ having a 1 on the diagonal entry corresponding to the position of index i in t, and $O^{t,ij}$ having a 1 on the two off-diagonal entries corresponding to the position of the pair (i, j) (cf. (24)), so as to express the equality constraint in (39d) by

$$\begin{array}{ll}
\sum_{t \in T : (i,j) \subset t} \langle\, O^{t,ij} \,,\, \Gamma^t \,\rangle + \langle\, O^{ij} \,,\, \Phi \,\rangle = 2 Q_{ij} & (i, j) \in P \\
\sum_{t \in T : i \in t} \langle\, D^{t,i} \,,\, \Gamma^t \,\rangle + \langle\, D^i \,,\, \Phi \,\rangle = Q_{ii} & i \in N
\end{array} .$$

The dual of the inner SDP problem then is

$$\begin{array}{lll}
\min\ \sum_{(i,j) \in P} 2 Q_{ij} f_{ij} + \sum_{i \in N} Q_{ii} f_i & & (40a) \\
\text{s.t. } \sum_{(i,j) \in P} O^{ij} f_{ij} + \sum_{i \in N} D^i f_i \succeq x x^T & & (25b) \\
\quad \sum\limits_{(i,j) \subset t} O^{t,ij} f_{ij} + \sum\limits_{i \in t} D^{t,i} f_i \succeq \sum\limits_{c \in C(t)} \dfrac{\bar{x}^{t,c} (\bar{x}^{t,c})^T}{y^{t,c}} & t \in T & (40b)
\end{array}$$

and the nonlinear constraints (25b) and (40b) can be rewritten as

$$\begin{array}{lll}
\left[\begin{array}{cc} 1 & x^T \\ x & \sum_{(i,j) \in P} O^{ij} f_{ij} + \sum_{i \in N} D^i f_i \end{array} \right] \succeq 0 & & (26a) \\
\sum_{(i,j) \subset t} O^{t,ij} f_{ij} + \sum_{i \in t} D^{t,i} f_i \succeq \sum_{c \in C(t)} \bar{W}^{t,c} & t \in T & (41) \\
\left[\begin{array}{cc} W^{t,c} & x^{t,c} \\ (x^{t,c})^T & y^{t,c} \end{array} \right] \succeq 0 & t \in T \ \ , \ \ c \in C(t) & (42)
\end{array}$$

Each $W^{t,c}$ in (42) is a $|c| \times |c|$ matrix, and $\bar{W}^{t,c}$ in (41) denotes $W^{t,c}$ extended to a $|t| \times |t|$ matrix by filling it with zeroes for the indices $i \notin c$. All in all, (39) then is

$$\begin{aligned} &\min\ q^T x + c^T y + \textstyle\sum_{(i,j)\in P} 2Q_{ij} f_{ij} + \sum_{i \in N} Q_{ii} f_i \\ &\text{s.t. } (1b) - (1c) \ \ , \ \ (38b) - (38d) \ \ , \ \ (39b) - (39c) \ \ , \ \ (26a) \ \ , \ \ (41) - (42) \end{aligned}$$

While the derivation clearly works for any $k \geq 3$, a fortiori the continuous relaxation of such a large SDP is going to be extremely challenging to solve as k grows.

B Proofs of Corollary and Proposition

Proof of Corollary 5.1.

Proof: Let $x > 0$ be the Perron-Frobenius eigenvector for $|I - \bar{Q}|$ (associated with the eigenvalue $\lambda = \rho(|I - \bar{Q}|)$) and define the matrix $W = XD^{-\frac{1}{2}}|Q|D^{-\frac{1}{2}}X$, where $X = diag(x)$ and $D = diag(Q)$. Since $XD^{-\frac{1}{2}} = D^{-\frac{1}{2}}X$ is invertible, there is a one-one correspondence between 2×2D's of W and 2×2D's of $|Q|$ (and thus of Q) whenever either W or $|Q|$ is 2×2-decomposable. Hence, Q has a unique 2×2D if and only if W does. Next, observe that we have

$$\begin{aligned} \rho(|I - \bar{Q}|) \leq 1 \quad &\Longleftrightarrow \quad |I - \bar{Q}|x = \lambda x \leq x \Longleftrightarrow \textstyle\sum_{j:j \neq i} |\bar{Q}_{ij}|x_j \leq x_i = |\bar{Q}_{ii}|x_i \quad \forall\, i \in N \\ &\Longleftrightarrow \quad \textstyle\sum_{j:j \neq i} x_i|\bar{Q}_{ij}|x_j \leq |\bar{Q}_{ii}|x_i^2 \ \ \equiv \ \ \sum_{j:j\neq i} W_{ij} \leq W_{ii} \qquad \forall\, i \in N\ . \end{aligned} \tag{43}$$

Moreover, $\rho(|I - \bar{Q}|) = 1 \Longleftrightarrow \sum_{j:j \neq i} W_{ij} = W_{ii}$. Hence, the proof will be complete when we show that W has a unique 2×2D if and only if $\sum_{j:j\neq i} W_{ij} = W_{ii}$ for every $i \in N$.

For the forward direction, suppose by way of contradiction that W has a unique 2×2D and $\sum_{j:j\neq i} W_{ij} \neq W_{ii}$ for some $i \in N$. Since Q is 2×2-decomposable, $\rho(|I - \bar{Q}|) \leq 1$; hence by (43) we must have $\sum_{j:j\neq i} W_{ij} < W_{ii}$. But then, observe that any two distinct choices of the convex multipliers $\{\alpha_i^{ik}\}_{k:k\neq i}$ in (29) yield different values for the variables π_i^{ij} with $j \neq i$, and hence distinct 2×2D's, contradicting the assumption that the 2×2D of W was unique.

For the backward direction, suppose that $\sum_{j:j\neq i} W_{ij} = W_{ii}$ for every $i \in N$. Then, one possible 2×2D of W is given by setting

$$\pi_i^{ij} = |W_{ij}| \quad \forall\, i,\ j \in N,\ i \neq j\ . \tag{44}$$

Denoting this 2×2D by $[\Pi^{ij}]_{(i,j)\in P}$, assume by way of contradiction that there exists another 2×2D $[\hat{\Pi}^{ij}]_{(i,j)\in P} \neq [\Pi^{ij}]_{(i,j)\in P}$. For each $(i,j) \in P$ define the 2×2 (diagonal) matrix $\Delta^{ij} := \hat{\Pi}^{ij} - \Pi^{ij}$. We denote by Δ_i^{ij} and Δ_j^{ij} the diagonal entries of Δ^{ij}. We claim that $tr(\Delta^{ij}) \geq 0$, with equality holding if and only if $\Delta^{ij} = 0$. Indeed, the claim is clearly true when $\Delta_i^{ij}, \Delta_j^{ij} \geq 0$. So, suppose instead that (without loss of generality) $\Delta_i^{ij} < 0$. Since $0 \preceq \hat{\Pi}^{ij} = \Pi^{ij} + \Delta^{ij}$, using (44) we obtain

$$\begin{aligned} |W_{ij}|^2 \ = \quad &\leq \ \ \hat{\pi}_i^{ij}\hat{\pi}_j^{ij} \ = \ (\pi_i^{ij} + \Delta_i^{ij})(\pi_j^{ij} + \Delta_j^{ij}) \\ &= \ \ (|W_{ij}| + \Delta_i^{ij})(|W_{ij}| + \Delta_j^{ij}) \ = \ |W_{ij}|^2 + |W_{ij}|(\Delta_i^{ij} + \Delta_j^{ij}) + \Delta_i^{ij}\Delta_j^{ij}\ . \end{aligned}$$

Simplifying and rearranging we obtain

$$\Delta_j^{ij}(|W_{ij}| + \Delta_i^{ij}) \geq -\Delta_i^{ij}|W_{ij}|\ . \tag{45}$$

Since $\hat{\Pi}^{ij} \succeq 0$, we must have that

$$|W_{ij}| + \Delta_i^{ij} = \pi_i^{ij} + \Delta_i^{ij} = \hat{\pi}_i^{ij} \geq 0 \; . \tag{46}$$

Moreover, note that if (46) holds with equality, then (45) gives $0 \geq -\Delta_i^{ij}|W_{ij}| = (\Delta_i^{ij})^2$, contradicting $\Delta_i^{ij} < 0$. Hence, the inequality in (46) is strict, and thus (45) is equivalent to

$$\Delta_j^{ij} \geq \frac{-\Delta_i^{ij}|W_{ij}|}{|W_{ij}| + \Delta_i^{ij}} > \frac{-\Delta_i^{ij}|W_{ij}|}{|W_{ij}|} = -\Delta_i^{ij} \; .$$

Hence, $tr(\Delta^{ij}) = \Delta_i^{ij} + \Delta_j^{ij} > 0$, which proves our claim. But note that

$$\textstyle\sum_{(i,j)\in P} tr(\Delta^{ij}) = tr\left(\sum_{(i,j)\in P} \Delta^{ij}\right) = tr\left(\sum_{(i,j)\in P} \hat{\Pi}^{ij} - \sum_{(i,j)\in P} \Pi^{ij}\right) = tr(W - W) = 0 \; .$$

Hence, we must have that $tr(\Delta^{ij}) = 0$ for every $(i,j) \in P$, which by our claim implies that $\Delta^{ij} = 0$ for every $(i,j) \in P$, contradicting the assumption that $[\hat{\Pi}^{ij}]_{(i,j)\in P} \neq [\Pi^{ij}]_{(i,j)\in P}$. This completes the proof of the backwards direction. □

Proof of Proposition 5.2.

Proof: Assume that Q is 2×2-decomposable. Let $\bar{P} = \{S \subseteq N \;:\; |S| = 2\}$ and let $f : \bar{P} \to N$ be defined by

$$f(\{i,j\}) = \begin{cases} n & \text{if } \{i,j\} = \{1,n\} \\ \min\{i,j\} & \text{otherwise} \end{cases} \qquad \forall\; \{i,j\} \in \bar{P} \;\; .$$

For convenience, we will write $f(i,j)$ instead of $f(\{i,j\})$. It is easy to see that f is onto. Indeed, if $i = 1$, then $f(i,2) = i$; if $i = n$, then $f(i,1) = i$; and if $2 \leq i \leq n-1$, then $f(i,n) = i$. With $\Pi = [\Pi^{ij}]_{(i,j)\in P}$ consider the optimization problem

$$\max\left\{\, g(\Pi) = \textstyle\sum_{\{i,j\}\in\bar{P}} \pi_{f(i,j)}^{ij} \;:\; (28) \,\right\} . \tag{47}$$

Since Q has a 2×2D, (47) is non-empty, in addition to being closed and bounded, and therefore it has an optimal solution $\bar{\Pi}$. We claim that for this solution, the inequalities (28b) are all active. Indeed, suppose by way of contradiction that, for some $\{i,j\} \in \bar{P}$,

$$\bar{\pi}_i^{ij}\bar{\pi}_j^{ij} > Q_{ij}^2 \; (\geq 0) \; , \tag{48}$$

where without loss of generality we can assume $f(i,j) = j$. Since f is onto, there exists $r(i) \in N \setminus \{i,j\}$ such that $f(i, r(i)) = i$. So, for any $\epsilon > 0$, we may define the point $\Pi(\epsilon)$ by

$$\pi(\epsilon)_k^{kl} = \begin{cases} \bar{\pi}_k^{kl} - \epsilon & \text{if } k = i \;\;,\;\; l = j \\ \bar{\pi}_k^{kl} + \epsilon & \text{if } k = i \;\;,\;\; l = r(i) \\ \bar{\pi}_k^{kl} & \text{if } k \neq i \text{ or } k = i \text{ and } l \notin \{j, r(i)\} \end{cases} \qquad \forall\; k,\; l \in N,\; k \neq l \; .$$

We claim that for all sufficiently small $\epsilon > 0$, $\Pi(\epsilon)$ is feasible in (47). Indeed, (28c) and (28b) hold since by (48) we have

$$\pi(\epsilon)_i^{ij} = \bar{\pi}_i^{ij} - \epsilon > 0 \qquad \text{and} \qquad \pi(\epsilon)_i^{ij}\pi(\epsilon)_j^{ij} = (\bar{\pi}_i^{ij} - \epsilon)\bar{\pi}_j^{ij} > Q_{ij}^2 \;\; .$$

Furthermore, (28a) holds since

$$\textstyle\sum_{l:\{i,l\}\in\bar{P}} \pi(\epsilon)_i^{il} = (\bar{\pi}_i^{ij} - \epsilon) + (\bar{\pi}_i^{i,r(i)} + \epsilon) + \sum_{l:\{i,l\}\in\bar{P},\, l\neq j,r(i)} \bar{\pi}_i^{i,l} = Q_{ii} \;\; .$$

Moreover,

$$\begin{aligned} g(\Pi(\epsilon)) &= \textstyle\sum_{\{k,l\}\in\bar{P}} \pi(\epsilon)^{kl}_{f(k,l)} \\ &= \pi(\epsilon)^{ij}_{f(i,j)} + \pi(\epsilon)^{i,r(i)}_{f(i,r(i))} + \textstyle\sum_{\{k,l\}\in\bar{P}:\{k,l\}\neq\{i,j\}\,,\,\{k,l\}\neq\{i,r(i)\}} \pi(\epsilon)^{kl}_{f(k,l)} \\ &= \bar{\pi}^{ij}_{j} + (\bar{\pi}^{i,r(i)}_{i} + \epsilon) + \textstyle\sum_{\{k,l\}\in\bar{P}:\{k,l\}\neq\{i,j\}\,,\,\{k,l\}\neq\{i,r(i)\}} \bar{\pi}^{kl}_{f(k,l)} \\ &= \textstyle\sum_{\{k,l\}\in\bar{P}} \bar{\pi}^{kl}_{f(k,l)} + \epsilon = g(\bar{\Pi}) + \epsilon > g(\bar{\Pi}) \ \ , \end{aligned}$$

contradicting the optimality of $\bar{\Pi}$. Hence, the inequalities (28b) must all be active at $\bar{\Pi}$, which implies that the determinant of all $\bar{\Pi}^{ij}$ is zero, i.e., the $\bar{\Pi}^{ij}$ all have rank at most one. □

Acknowledgements

This work has been partly developed while the third author was an ER of the ITN 316647 "MINO: Mixed-Integer Nonlinear Optimization" program funded by the European Union. The first and second authors gratefully acknowledge the financial contribution of the project PRIN 2015B5F27W funded by the Italian Minister for Education, the networking support of the COST Action TD1207 and the ITN 764759 "MINOA: Mixed-Integer Nonlinear Optimization Applications" both of the European Union. We are grateful to the anonymous referees for their valuable comments which have us helped improve the presentation of the paper, as well as for pointing out [2, 3, 15] and suggesting the simplified heuristic described at the end of Section 6.

References

[1] A. Atamtürk and A. Gómez. Strong formulations for quadratic optimization with M-matrices and semi-continuous variables. Technical Report [math.OC], University of California, Berkeley, 2018.

[2] A.A. Ahmadi and A. Majumdar. Some applications of polynomial optimization in operations research and real-time decision making. *Optimization Letters*, 10(4):709–729, 2016.

[3] A.A. Ahmadi and A. Majumdar. DSOS and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization. arXiv:1706.02586 [math.OC], 2017.

[4] A. Billionnet, S. Elloumi, and A. Lambert. Extending the QCR method to the case of general mixed integer programs. *Mathematical Programming*, 131(1):381–401, 2012.

[5] Erik G. Boman, Doron Chen, Ojas Parekh, and Sivan Toledo. On factor width and symmetric H-matrices. *Linear Algebra and its Applications*, 405:239 – 248, 2005.

[6] S. Ceria and J. Soares. Convex programming for disjunctive convex optimization. *Mathematical Programming*, 86:595–614, 1999.

[7] X. T. Cui, X. J. Zheng, S. S. Zhu, and X. L. Sun. Convex relaxations and MIQCQP reformulations for a class of cardinality-constrained portfolio selection problems. *J. Global Optim.*, 56(4):1409–1423, 2013.

[8] A. Frangioni, F. Furini, and C. Gentile. Approximated perspective relaxations: a project&lift approach. *Computational Optimization and Applications*, 63:705–735, 2016.

[9] A. Frangioni, F. Furini, and C. Gentile. Improving the Approximated Projected Perspective Reformulation by Dual Information. Technical report, Dipartimento di Informatica, Università di Pisa, 2016.

[10] A. Frangioni and C. Gentile. Perspective Cuts for a Class of Convex 0–1 Mixed Integer Programs. *Mathematical Programming*, 106(2):225 – 236, 2006.

[11] A. Frangioni and C. Gentile. SDP Diagonalizations and Perspective Cuts for a Class of Nonseparable MIQP. *Operations Research Letters*, 35(2):181 – 185, 2007.

[12] A. Frangioni and C. Gentile. A Computational Comparison of Reformulations of the Perspective Relaxation: SOCP vs. Cutting Planes. *Operations Research Letters*, 37(3):206 – 210, 2009.

[13] F. G. Frobenius. *Über Matrizen aus nicht negativen Elementen.* De Gruyter, Berlin, Boston, 1912.

[14] H. Jeon, J. Linderoth, and A. Miller. Quadratic cone cutting surfaces for quadratic programs with on–off constraints. *Discrete Optimization*, online first, 2014.

[15] A. Majumdar, A.A. Ahmadi, and R. Tedrake. Control and verification of high-dimensional systems with dsos and sdsos programming. In *53rd IEEE Conference on Decision and Control*, pages 394–401, Dec 2014.

[16] O. Perron. Zur theorie der matrices. *Mathematische Annalen*, 64:248–263, 1907.

[17] R.J. Plemmons. M-matrix characterizations. I - Nonsingular M-matrices. *Linear Algebra and its Applications*, 18(2):175 – 188, 1977.

[18] R.T. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, NJ, 1970.

[19] N. Ruozzi and S. Tatikonda. Message-passing algorithms for quadratic minimization. *Journal of Machine Learning*, 14:2287–2314, 2013.

[20] H.D. Sherali and W.P. Adams. A reformulation-linearization technique (rlt) for semi-infinite and convex programs under mixed 0-1 and general discrete restrictions. *Discrete Applied Mathematics*, 157:1319–1333, 2009.

[21] X. Zheng, X. Sun, and D. Li. Improving the Performance of MIQP Solvers for Quadratic Programs with Cardinality and Minimum Threshold Constraints: A Semidefinite Program Approach. *INFORMS Journal on Computing*, 26(4):690–703, 2014.