ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

A. Frangioni, C. Gentile, J. Hungerford

DECOMPOSITIONS OF SEMIDEFINITE MATRICES AND THE PERSPECTIVE REFORMULATION OF NONSEPARABLE QUADRATIC PROGRAMS

R. 16-10, 2016

Antonio Frangioni – Dipartimento di Informatica, Università di Pisa (Italy). Email: frangio@di.unipi.it.

Claudio Gentile – Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Consiglio Nazionale delle Ricerche, Rome (Italy). Email: gentile@iasi.cnr.it.

James Hungerford – RaceTrac, Atlanta, Georgia (USA) jamesthungerford@gmail.com.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

via dei Taurini 19, 00185 ROMA, Italy

tel. ++39-06-49937101/02
fax ++39-06-49937106
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

We study the problem of (approximately) decomposing the hessian matrix of a Mixed-Integer Convex Quadratic Program with semicontinuous variables as the sum of positive semidefinite 2×2 matrices. Solving this problem can enable the use of Perspective Reformulation techniques for obtaining strong lower bounds for the MICQP. We discuss two exact SDP approaches for finding an approximate decomposition, we characterize the set of matrices that have an exact decomposition, and we use the characterization to devise efficient heuristics for obtaining 2×2 decompositions. We present preliminary results on the bound strength for Portfolio Optimization problems, showing that for some classes of problems 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.

1. Introduction and the 1×1 case

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

$$\min\ x^T Q x + q^T x + c^T y \tag{1}$$
$$Ax + By \le b \tag{2}$$
$$l_i y_i \le x_i \le u_i y_i \qquad i \in N \tag{3}$$
$$y_i \in \{0,1\} \qquad i \in N \tag{4}$$

where $x = [x_i]_{i\in N} \in \mathbb{R}^n$ ($N = \{1,2,\ldots,n\}$), $Q \in \mathbb{R}^{n\times n}$ 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 (3) and (4) imply that each x_i is a semicontinuous variable governed by the 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., $-\infty < l_i \le u_i < \infty$). The formulation can be made more complex in several ways, for instance requiring 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 (hopefully, convex) constraints, but we will stick to (1)–(4) for simplicity of notation.

When Q is diagonal, i.e., $x^T Q x = \sum_{i\in N} Q_{ii} x_i^2$, the problem is well-suited for application of the *Perspective Reformulation* (PR). This simply amounts to replacing the objective function with its *convex envelope*, obtained by considering only the semi-continuous constraints (3)–(4):

$$\textstyle\sum_{i\in N} Q_{ii} x_i^2/y_i + q^T x \ \ , \tag{5}$$

where we assume that $x_i^2/y_i = 0$ if $y_i = 0$ (since $y_i = 0 \Longrightarrow x_i = 0$, the extended function is continuous). Note that (5) is just the *perspective function* [12] of the original cost function. Despite its appearance, (5) is convex, and therefore the continuous relaxation of the problem of minimizing (5) subject to (2)–(4)—the *Perspective Relaxation* PR—can be efficiently solved. For instance, one can either iteratively approximate it by using linear approximations (*Perspective Cuts* [5]), or reformulate it as a Second-Order Cone Program and solve it in one blow with existing approaches [7], or even considering new reformulations [3].

In the present paper we are interested in the case where the objective function (1) is *not* separable. A simple but effective extension of the above approach to the non-separable case was proposed in [5, 6], and refined in [15]. The main idea is to reformulate (1)–(4) as

$$\min\left\{\ \textstyle\sum_{i\in N} \delta_i x_i^2 + x^T(Q - diag(\delta))x + q^T x + c^T y\ :\ (2)\text{–}(4)\ \right\}\ , \tag{6}$$

where $\delta \ge 0$ 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 (6), which leads to

$$\min\left\{\ \textstyle\sum_{i\in N} \delta_i x_i^2/y_i + x^T(Q - diag(\delta))x + q^T x + c^T y\ :\ (2)\text{–}(4)\ \right\}\ . \tag{7}$$

The advantage of (7) is that its continuous relaxation, i.e., the PR, often provides a strictly better bound than the continuous relaxation of (1)–(4). Clearly, the quality of the bound depends on δ, and, intuitively, "the larger δ, the better the bound". In [5] a simple and inexpensive way of choosing δ, based on an eigenvalue computation, was used. In [6] an SDP approach was proposed. In particular, given a vector of weights $\alpha = [\alpha_i]_{i\in N} \ge 0$ for the individual components of δ,

finding the "largest" possible δ can be cast as the following dual pair of SemiDefinite Programs (SDP):

$$\begin{array}{c} \max \left\{ \sum_{i\in N} \alpha_i \delta_i \ : \ Q - \sum_{i\in N} D^i \delta_i \succeq 0 \ , \ \delta \geq 0 \right\} \\ \min \left\{ \langle Q \, , \, F \rangle \ : \ diag(F) \geq \alpha \ , \ F \succeq 0 \right\} \end{array} \ , \tag{8}$$

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 (6). In [6], unitary weights were used for simplicity, while an approach for finding the "best possible" δ based on solving the following program, was proposed in [15]:

$$\max_\delta \ \min_{x,y} \ q^T x + c^T y + \sum_{i\in N} \delta_i x_i^2 / y_i + x^T (Q - diag(\delta)) x \tag{9}$$

$$(2)\text{–}(3) \ , \ \delta \geq 0 \ , \ Q - diag(\delta) \succeq 0 \ , \ y \in [0,1]^n \tag{10}$$

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 a convex optimization problem, as the function $\phi(\delta) = \min_{x,y} \{ (9) \ : \ (10) \}$ is clearly concave in δ, being the pointwise infimum of (infinitely many) linear functions in δ (indexed over x and y). Indeed, (9)–(10) can formulated as an SDP. In [15] 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 (9) 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 (9) [12, Corollary 37.3.2], to obtain

$$\min_{x,y} \ \left\{ x^T Q x + q^T x + c^T y + \psi(x,y) \ : \ (2) \ , \ (3) \ , \ y \in [0,1]^n \right\} \ , \tag{11}$$

where

$$\psi(x,y) = \max_\delta \ \left\{ \sum_{i\in N} (x_i^2/y_i - x_i^2)\delta_i \ : \ Q - diag(\delta) \succeq 0 \ , \ \delta \geq 0 \right\} \ . \tag{12}$$

Clearly, (11) 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 (11) as a single SDP it suffices to perform the variable change $\Phi = Q - diag(\delta)$ $(\succeq 0)$ yielding $\delta_i = Q_{ii} - \Phi_{ii}$; then, by defining $P = \{ (i,j) \in N \times N \ : \ i < j \}$ one can write

$$\psi(x,y) = \sum_{i\in N} Q_{ii} x_i^2 / y_i + \left\{ \begin{array}{lll} \max_\Phi & \langle x x^T - V(x,y) \, , \, \Phi \rangle & \\ & \langle O^{ij} \, , \, \Phi \rangle = 2 Q_{ij} & (i,j) \in P \\ & \langle D^i \, , \, \Phi \rangle \leq Q_{ii} & i \in N \\ & \Phi \succeq 0 & \end{array} \right. \tag{13}$$

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 (13) is

$$\min \left\{ \langle Q \, , \, F \rangle \ : \ F \succeq x x^T - V(x,y) \ , \ diag(F) \geq 0 \right\} \ . \tag{14}$$

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

$$F \succeq x x^T - V(x,y) \quad \equiv \quad \left[\begin{array}{cc} 1 & x^T \\ x & F + V(x,y) \end{array} \right] \succeq 0 \, .$$

Analogously, using the well-known

$$y_i \geq 0 \ , \ w_i \geq 0 \ , \ w_i \geq x_i^2 / y_i \quad \equiv \quad \left[\begin{array}{cc} w_i & x_i \\ x_i & y_i \end{array} \right] \succeq 0 \, ,$$

one ends up with the SDP form of (11)

$$\min\ q^T x + c^T y + \textstyle\sum_{i\in N}\ Q_{ii} w_i + \langle\, Q\,,\, F\,\rangle \tag{15}$$

$$(2)\ \ ,\ \ (3)\ \ ,\ \ y \in [0,1]^n\ \ ,\ \ diag(F) \geq 0$$

$$\begin{bmatrix} 1 & x^T \\ x & F + diag(w) \end{bmatrix} \succeq 0 \tag{16}$$

$$\begin{bmatrix} w_i & x_i \\ x_i & y_i \end{bmatrix} \succeq 0 \qquad\qquad i \in N \tag{17}$$

Solving (15)–(17) provides the best possible lower bound and the corresponding optimal solution (x, y). This could be used to compute the optimal δ as in (12), but in fact that is already provided by the dual variables of the $diag(F) \geq 0$ constraint. As solving the large-scale SDP (15)–(17) at all iterations of an enumerative approach to solve the original (1)–(4) is rather costly, 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 all the B&C applied to reformulation (7). This has been shown [4, 15] to be significantly better than using the diagonal, denoted by δ_s, obtained from the (much cheaper) SDP problem (8); 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 [15] it is reported that using a convex combination of δ_l and δ_s is sometimes preferable.

The aim of this paper is to further strengthen the bounds provided by the PR approach by "extracting an even larger part of Q". The idea is to see the above approach in terms of trying to approximate Q as the sum of "small" PSD matrices (i.e., $D^i\delta_i$) to which the PR can be applied. Being that the matrices are only diagonal, it is clear that Q cannot be completely expressed as the sum of these, unless of course Q is itself diagonal. However, by using "only slightly larger" base matrices one indeed has the hope of expressing Q exactly as the sum, or at least to reduce the weight of the residual. Of course, this requires being able to apply the PR technique to more than one x_i at a time.

The structure of the paper is the following. In Section 2 we show how to compute the PR of nonseparable $k \times k$ MIQPs with semicontinuous variables for arbitrary (but, in fact, necessarily "small") k. In Section 3 we study the problem of extracting *approximate* $k \times k$ decompositions of a given PSD matrix Q. We consider both a simple heuristic approach and an exact approach (analogous to the computations of δ_s and δ_l, respectively). In Section 4 we concentrate on the 2×2 case (most likely the only one of practical relevance), characterizing the matrices that have an *exact* 2×2 decomposition, and exploiting this characterization to devise alternative, faster (possibly heuristic) approaches to compute approximate decompositions. Finally, in Section 5 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}. Conversely, given an n-vector d, $diag(d)$ is the $n \times n$ diagonal matrix whose i-th diagonal element is d_i. $\langle\, A\,,\, B\,\rangle = trace(AB)$ whenever $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times m}$, 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. $P = \{\,(i, j) \in N \times N\ :\ i < j\,\}$. $\underline{\mathrm{P}}$ denotes the continuous relaxation of a mixed integer optimization problem P. The remaining notation is fairly standard.

2. The convex envelope of small MIQPs

We now study the problem of computing "good formulations" of small-scale MIQPs of the form (1)–(4). In particular, we will start by considering the basic convex 2×2 MIQP with semi-continuous variables

$$\min \left\{ q_{11}x_1^2 + 2q_{12}x_1x_2 + q_{22}x_2^2 \ : \ l_iy_i \le x_i \le u_iy_i \ \ , \ \ y_i \in \{0,1\} \quad i \in \{1\,,\,2\} \right\}, \qquad (18)$$

where the Hessian is positive semidefinite, i.e., $q_{11} > 0$, $q_{22} > 0$, and $q_{11}q_{22} \ge (q_{12})^2$ (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 (18), a task for which there is no lack of theory. For instance, we could use the standard RLT [14]. Also, an in-depth study of the polyhedral structure of the set is available in [10]. For our purposes, however, the following simple tools are perhaps better suited.

2.1. 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 $k \ne h$ (and each N^k nonempty). We will require each x^k to be either 0 or to live in a compact set; for our purposes we can assume these to be polyhedra, i.e., $\mathcal{P}^k = \{\ x^k \ : \ A^kx^k \le b^k \ \}$, 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., $\{\ x^k \ : \ A^kx^k \le 0 \ \} = \{0\}$. 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:

$$f(x) = \begin{cases} f^k(x^k) + c^k & \text{if } x^k \in \mathcal{P}^k \text{ and } x^h = 0 \ \forall\ h \in K \setminus \{k\} \\ 0 & \text{if } x = 0 \\ +\infty & \text{otherwise} \end{cases} . \qquad (19)$$

Computing the convex envelope $\overline{co} f(x)$ of (19) is a simple task. 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

$$\begin{array}{rll} \overline{co} f(x) \ = \ \min_{\bar{x},\theta} \sum_{k\in K} \theta^k f(\bar{x}^k) & & \\ \qquad \sum_{k\in K} \theta^k \le 1 \ \ , \ \ \sum_{k\in K} \theta^k \bar{x}^k = x & & \\ \qquad A^k x^k \le b^k \theta^k \ \ , \ \ \theta^k \ge 0 & & k \in K \end{array}$$

Note that a term "$0\theta^0$" would be present in the objective function and in the second constraint, while the first constraint should be an equality with an extra variable "θ^0"; clearly, we can treat θ^0 as a slack variable and eliminate it. In the constraint "$\sum_{k\in K} \theta^k \bar{x}^k = x$", $\bar{x}^k$ should in principle be considered as extended to the whole space, whence the sum. However, due to (19), if for $k \ne h$ one had $\bar{x}_i > 0$ and $\bar{x}_j > 0$ for $i \in N^k$ and $j \in N^h$, then $f(\bar{x}) = +\infty$. 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, the constraint "$\sum_{k\in K} \theta^k \bar{x}^k = x$" equivalently reads "$\theta^k \bar{x}^k = x^k$ for all $k \in K$", immediately leading to

$$\overline{co} f(x) \ = \ \min_{\theta} \left\{ \ \sum_{k\in K} \theta^k f^k(x^k/\theta^k) \ : \ \sum_{k\in K} \theta^k \le 1 \ , \ A^kx^k \le b^k\theta^k \ , \ \theta_k \ge 0 \quad k \in K \right\} \ . \quad (20)$$

In plain words, the convex envelope of the alternatives is just the sum of the individual convex envelopes plus the simplex constraint "$\sum_{k\in K}\theta^k \le 1$". 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.

2.2. The 2×2 convex envelope

Using the above result we can now easily compute the PR of (18). This simply starts with the (somewhat awkward) reformulation

$$\min\ q_{11}(x_1^1)^2 + q_{22}(x_2^2)^2 + q_{11}(x_1^{12})^2 + 2q_{12}x_1^{12}x_2^{12} + q_{22}(x_2^{12})^2 \tag{21}$$

$$x_i = x_i^i + x_i^{12} \quad , \quad y_i = y^i + y^{12} \qquad i\in\{1\,,\,2\} \tag{22}$$

$$l_i y^i \le x_i^i \le u_i y^i \ \ , \ \ l_i y^{12} \le x_i^{12} \le u_i y^{12} \qquad i\in\{1\,,\,2\} \tag{23}$$

$$y^1 + y^2 + y^{12} \le 1 \tag{24}$$

$$y^1,\ y^2,\ y^{12} \in \{0,1\}\ . \tag{25}$$

The aim of (21)–(25) 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 (18) the structure of (19). Note that (22) 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 (22). We can now apply (20), obtaining the PR

$$\min\ q_{11}(x_1^1)^2/y^1 + q_{22}(x_2^2)^2/y^2 + \left[\ q_{11}(x_1^{12})^2 + 2q_{12}x_1^{12}x_2^{12} + q_{22}(x_2^{12})^2\ \right]/y^{12} \tag{26}$$

$$(22)\ \ , \ \ (23)\ \ , \ \ (24)\ \ , \ \ y^1,\ y^2,\ y^{12} \ge 0\ . \tag{27}$$

Although (26) is of significantly larger dimension than (18), 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

$$\min\ \frac{q_{11}(x_1 - x_1^{12})^2}{y_1 - y^{12}} + \frac{q_{22}(x_2 - x_2^{12})^2}{y_2 - y^{12}} + \frac{1}{y^{12}}\begin{bmatrix} x_1^{12} & x_2^{12}\end{bmatrix}\begin{bmatrix} q_{11} & q_{12}\\ q_{12} & q_{22}\end{bmatrix}\begin{bmatrix} x_1^{12}\\ x_2^{12}\end{bmatrix}$$

$$l_i(y_i - y^{12}) \le x_i - x_i^{12} \le u_i(y_i - y^{12}) \ \ , \ \ l_i y^{12} \le x_i^{12} \le u_i y^{12} \qquad i\in\{1\,,\,2\}$$

$$y_1 + y_2 - y^{12} \le 1 \ \ , \ \ y_1 \ge y^{12},\ y_2 \ge y^{12},\ y^{12} \ge 0\ .$$

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 (26)–(27) 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 (26)–(27) to the space of the original variables has been provided in [10], 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, (26)–(27) have to be rewritten as

$$\min\ w^1 + w^2 + w^{12} \tag{28}$$

$$(22)\ \ , \ \ (23)\ \ , \ \ (24)\ \ , \ \ y^1,\ y^2,\ y^{12} \ge 0$$

$$w^1 y^1 \ge q_{11}(x_1^1)^2 \ \ , \ \ w^2 y^2 \ge q_{22}(x_2^2)^2 \ \ , \ \ w^{12} y^{12} \ge \begin{bmatrix} x_1^{12} & x_2^{12}\end{bmatrix}\begin{bmatrix} q_{11} & q_{12}\\ q_{12} & q_{22}\end{bmatrix}\begin{bmatrix} x_1^{12}\\ x_2^{12}\end{bmatrix} \tag{29}$$

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.

2.3. Convex envelopes in higher dimension

The above technique can obviously be used to compute convex envelopes in higher dimensions, although of course the size of the corresponding formulations 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

$$\min \sum_{c \in C(t)} \left[(x^c)^T Q^c x^c \right] / y^c \tag{30}$$

$$x_i = \sum_{c \in C(t) \,:\, i \in c} x_i^c \qquad i \in t \tag{31}$$

$$y_i = \sum_{c \in C(t) \,:\, i \in c} y^c \qquad i \in t \tag{32}$$

$$l_i y^c \le x_i^c \le u_i y^c \qquad c \in C(t)\ ,\ \ i \in c \tag{33}$$

$$\sum_{c \in C(t)} y^c \le 1 \tag{34}$$

$$y^c \in \{0, 1\} \qquad c \in C(t) \tag{35}$$

Extending (30)–(35) 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 [14] 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 §1 to the $k \ge 2$ case. That is, we seek to (approximately) decompose Q as the sum of several $k \times k$ PSD matrices with small values of k, to each of which the technique can be separately (and, hopefully, efficiently) applied.

3. Approximate $k \times k$ decomposition of semidefinite matrices

In this section we explore the direct generalization of the approach described in §1. That is, we define the problem of approximately decomposing the PSD matrix Q in (1) 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)–(4) are taken into account to find the "best" possible decomposition.

We of course start with the 2×2 case: for each $(i, j) = p \in P$, we define the $n \times 2$ matrix $E^p = [e_i, e_j]$, where as usual e_h is the h-th vector of the canonical basis of $\mathbb{R}^n$, as well as the

variables $\Pi^p \in \mathbb{R}^{2\times 2}$, i.e., Π^p are 2×2 matrices. Hence, Q admits a *decomposition into 2×2 PSD matrices* (2×2D for short) if and only if the set of conic (semidefinite) constraints

$$Q = \sum_{p\in P} E^p \Pi^p (E^p)^T \tag{36}$$

$$\Pi^p \succeq 0 \qquad\qquad p \in P \tag{37}$$

has a solution. 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 off-diagonal entry. 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 (37). Indeed, when Q has some all-zero row/column (save for the diagonal) is a particular case of reducible matrix, that is Q can be transformed by reshuffling the rows and the columns into a block diagonal matrix $\tilde{Q}$ of the form

$$\tilde{Q} = \left[\begin{array}{cc} Q_{11} & 0 \\ 0 & Q_{22} \end{array} \right] . \tag{38}$$

As $\tilde{Q} = PQP^T$ where P is an appropriate permutation matrix, one can easily see that a 2×2D obtained for $\tilde{Q}$ can be transformed into a 2×2D for Q, and it is possible to obtain a 2×2D for $\tilde{Q}$ by simply finding a 2×2D for each of the blocks Q_{11} and Q_{22}. Thus we can assume that Q is irreducible.

Since (36)–(37) is a set of SOCP constraints, determining the existence of a 2×2D is a polynomial-time problem. Also, it is possible to write as an SDP the problem of extracting "the largest possible decomposition" of Q. Similarly to (8) 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

$$\min \left\{ \| \Phi \|^2 \, : \, Q = \Phi + \sum_{p\in P} E^p \Pi^p (E^p)^T \;\; , \;\; (37) \;\; , \;\; \Phi \succeq 0 \right\} . \tag{39}$$

Clearly, the optimal value of (39) is zero if an only if (36)–(37) has a solution. Any feasible solution to problem (39) can then be used to define a *2×2 Perspective Reformulation* (2×2PR for short) of (1)–(4) as follows:

$$\min x^T \Phi x + q^T x + c^T y + \sum_{p=(i,j)\in P} \left[\Pi^p_{11} \frac{(x^{p,i}_i)^2}{y^{p,i}} + \Pi^p_{22} \frac{(x^{p,j}_j)^2}{y^{p,j}} + \frac{(x^{p,p})^T \Pi^p x^{p,p}}{y^{p,p}} \right] \tag{40}$$

$$(2)\text{–}(4)$$

$$x_i = x^{p,i}_i + x^{p,p}_i \quad , \quad y_i = y^{p,i} + y^{p,p} \qquad p \in P \;\; , \;\; i \in p \tag{41}$$

$$l_i y^{p,i} \le x^{p,i}_i \le u_i y^{p,i} \quad , \quad l_i y^{p,p} \le x^{p,p}_i \le u_i y^{p,p} \qquad p \in P \;\; , \;\; i \in p \tag{42}$$

$$y^{p,p} + y^{p,i} + y^{p,j} \le 1 \qquad p = (i,j) \in P \tag{43}$$

$$y^{p,i}, \; y^{p,j}, \; y^{p,p} \in \{0,1\} \;\; , \;\; y^{p,p} \in \{0,1\} \qquad p = (i,j) \in P \, . \tag{44}$$

However, there is no guarantee that the optimal solution to (39) provides the best lower bound in the corresponding 2×2PR, i.e., the continuous relaxation of (40)–(44). Yet, as for the one-dimensional case discussed in §1 it is possible to write the problem of finding the 2×2D that provides the best bound by maximizing the continuous relaxation of (40)–(44) over all approx-

imate decompositions (Π, Φ). Interchanging max/min then yields

$$\min_{x,y} q^T x + c^T y + \max_{\Phi,\Pi} \langle\, \Phi\,,\, xx^T \,\rangle + \sum_{p=(i,j)\in P} \left\langle \begin{bmatrix} \frac{(x_i^{p,i})^2}{y^{p,i}} & 0 \\ 0 & \frac{(x_j^{p,j})^2}{y^{p,j}} \end{bmatrix} + \frac{x^{p,p}(x^{p,p})^T}{y^{p,p}} \,,\, \Pi^p \right\rangle \tag{45}$$

$$(2)\text{–}(3) \quad , \quad (41)\text{–}(43)$$

$$y^{p,i}\,,\; y^{p,j}\,,\; y^{p,p} \in [0,1] \qquad\qquad p=(i,j)\in P \tag{46}$$

$$y \in [0,1]^n \tag{47}$$

$$Q = \Phi + \textstyle\sum_{p\in P} E^p \Pi^p (E^p)^T \quad , \quad (37) \quad , \quad \Phi \succeq 0\,. \tag{48}$$

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

$$\hat{O}^{12} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \;\;,\;\; \hat{D}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \;\;,\;\; \hat{D}^2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{49}$$

so as to express the equality constraint in (48) as

$$\begin{array}{ll} \langle\, \hat{O}^{12}\,,\, \Pi^{ij} \,\rangle + \langle\, O^{ij}\,,\, \Phi \,\rangle = 2Q_{ij} & (i,j)\in P \\ \sum_{j>i} \langle\, \hat{D}^1\,,\, \Pi^{ij} \,\rangle + \sum_{j<i} \langle\, \hat{D}^2\,,\, \Pi^{ji} \,\rangle + \langle\, D^i\,,\, \Phi \,\rangle = Q_{ii} & i \in N \end{array} \quad , \tag{50}$$

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

$$\min \textstyle\sum_{p\in P} 2Q_p f_p + \sum_{i\in N} Q_{ii} f_i \tag{51}$$

$$\textstyle\sum_{p\in P} O^p f_p + \sum_{i\in N} D^i f_i \succeq xx^T \tag{52}$$

$$\hat{O}^{12} f_p + \hat{D}^1 f_i + \hat{D}^2 f_j \succeq \begin{bmatrix} \frac{(x_i^{p,i})^2}{y^{p,i}} & 0 \\ 0 & \frac{(x_j^{p,j})^2}{y^{p,j}} \end{bmatrix} + \frac{x^{p,p}(x^{p,p})^T}{y^{p,p}} \qquad p=(i,j)\in P\,. \tag{53}$$

When x and y are variables, the conic constraints (52) and (53) are nonlinear. However, they can be transformed into linear constraints by introducing auxiliary variables and constraints, as follows:

$$\begin{bmatrix} 1 & x^T \\ x & \sum_{p\in P} O^p f_p + \sum_{i\in N} D^i f_i \end{bmatrix} \succeq 0 \tag{54}$$

$$\hat{O}^{12} f_p + \hat{D}^1 f_i + \hat{D}^2 f_j \succeq \begin{bmatrix} w_i^p & 0 \\ 0 & w_j^p \end{bmatrix} + W^p \qquad\qquad p=(i,j)\in P \tag{55}$$

$$\begin{bmatrix} w_i^p & x_i^{p,i} \\ x_i^{p,i} & y^{p,i} \end{bmatrix} \succeq 0 \qquad\qquad p\in P \;\;,\;\; i\in p \tag{56}$$

$$\begin{bmatrix} W^p & x^{p,p} \\ (x^{p,p})^T & y^{p,p} \end{bmatrix} \succeq 0 \qquad\qquad p\in P \qquad\qquad , \tag{57}$$

with W^p obviously being 2×2 matrices. All in all, (45)–(48) then is

$$\begin{array}{ll} \min & q^T x + c^T y + \sum_{p\in P} 2Q_p f_p + \sum_{i\in N} Q_{ii} f_i \\ & (2)\text{–}(3) \quad , \quad (41)\text{–}(43) \quad , \quad (46)\text{–}(47) \quad , \quad (54)\text{–}(57) \end{array} \quad , \tag{58}$$

which is a rather "large" SDP as n grows. As we shall see, actually solving (58) can be challenging. However, it is poised to produce a tighter lower bound than (15)–(17), 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 (55) 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 coming 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.

4. A study of 2×2 decomposability

We now concentrate on the 2×2 case and develop a full characterization of the matrices that have an *exact* 2×2D. We then use this to prove some theoretical and algorithmic results on finding exact and approximate 2×2Ds of a given PSD matrix Q.

4.1. Characterizing 2×2 decomposability

In this section we assume $n \geq 2$, unless otherwise stated. We say that an $n \times n$ symmetric real-valued matrix Q has or admits a 2×2D (we also say Q is *2×2-decomposable*) if the SDP system (36)–(37) has a solution. Note that while we have not explicitly asked Q to be PSD, a trivial consequence of the definition is that every 2×2-decomposable matrix *is* PSD, since it is the sum of PSD matrices. Because in (36) each Π^{ij} only influences the four entries (i,i), (i,j), (j,i) and (j,j) of the decomposition, in the following with a slight abuse of notation, and whenever this does not detract from clarity, we will denote by Π^{ij} both the 2×2 matrix, and the $n \times n$ matrix obtained from it as $E^{ij}\Pi^{ij}(E^{ij})^T$.

Now, note that since $\Pi^{ij}_{hk} = 0$ for every $(h,k) \in P$, $(h,k) \neq (i,j)$, by (36) we must have $\Pi^{ij}_{ij} = \Pi^{ij}_{ji} = Q_{ij}$ in any feasible 2×2D. In other words, Q has a 2×2D if and only if there exists a solution to the system

$$\sum_{j:j\neq i} \pi^{ij}_i = Q_{ii} \qquad i \in N \tag{59}$$

$$\pi^{ij}_i \pi^{ij}_j \geq Q^2_{ij} \qquad (i,j) \in P \tag{60}$$

$$\pi^{ij}_i \geq 0 \ , \ \ \pi^{ij}_j \geq 0 \qquad (i,j) \in P \ , \tag{61}$$

where π^{ij}_i and π^{ij}_j represent the values chosen for Π^{ij}_{ii} and Π^{ij}_{jj} (the two diagonal elements in the 2×2 matrix), respectively. Note that in (59)–(61), as well as in the remainder of this section, we use the notation π^{ij}_i and π^{ji}_i interchangeably for any $i, j \in N$ with $i \neq j$. We will now provide a characterization of the class of 2×2-decomposable matrices, begining with a simple observation. Recall that a symmetric matrix A is *weakly diagonally dominant* (WDD) if $|a_{ii}| \geq \sum_{j:j\neq i} |a_{ij}|$ for every $i \in N$.

Lemma 4.1. *If Q is WDD and $Q_{ii} \geq 0$ for all $i \in N$, then Q is 2×2-decomposable.*

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

$$\pi^{ij}_i = |Q_{ij}| + \alpha^{ij}_i \left(Q_{ii} - \textstyle\sum_{k:k\neq i} |Q_{ik}| \right) \ . \tag{62}$$

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

$$\pi^{ij}_i \pi^{ij}_j \geq |Q_{ij}||Q_{ji}| = Q^2_{ij} \ , \text{ and}$$

$$\begin{aligned}\textstyle\sum_{l:l\neq i}\pi_i^{il} &= \textstyle\sum_{l:l\neq i}|Q_{il}| + \left(\sum_{l:l\neq i}\alpha_i^{il}\right)\left(Q_{ii} - \sum_{k:k\neq i}|Q_{ik}|\right) \\ &= \textstyle\sum_{l:l\neq i}|Q_{il}| + Q_{ii} - \sum_{k:k\neq i}|Q_{ik}| = Q_{ii} \ .\end{aligned}$$

□

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

$$\begin{bmatrix} 2 & 2 & 1 \\ 2 & 5 & 1 \\ 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \ . \tag{63}$$

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

$$|d_iQ_{ii}d_i| \geq \textstyle\sum_{j:j\neq i}|d_iQ_{ij}d_j| \quad \equiv \quad |Q_{ii}|d_i \geq \sum_{j:j\neq i}|Q_{ij}|d_j \qquad \forall\, i \in N \ . \tag{64}$$

In the literature (e.g. [13]), (64) is often used as the definition of WSDD. A straightforward calculation shows that the matrix in (63) is WSDD, for example using $D = diag(\,7/4\,,\,1\,,\,3/2\,)$.

We define the *absolute value class* of Q, denoted $\mathcal{A}(Q)$, by

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

where $\mathcal{S}_n$ is the set of $n \times n$ (real-valued) symmetric matrices.

Theorem 4.2. *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) *$\tilde{Q}$ is 2×2-decomposable for every $\tilde{Q} \in \mathcal{A}(Q)$;*

(3) *$\tilde{Q} \succeq 0$ for every $\tilde{Q} \in \mathcal{A}(Q)$;*

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

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

(6) *Q is WSDD.*

Proof. It is straightforward to show that in the case where Q is reducible, each of the six conditions in the Theorem holds for Q if and only if it holds for each irreducible component of Q. Hence, the theorem will be proved once we prove it for the case where Q is irreducible. So, assume Q is irreducible. We prove a cycle of implications.

(1) ⟹ (2): Note that the system (59)–(61) is identical for every matrix $\tilde{Q} \in \mathcal{A}(Q)$. Hence, if Q is 2×2-decomposable, then so is every $\tilde{Q} \in \mathcal{A}(Q)$.

(2) ⟹ (3): As already mentioned, every 2×2-decomposable matrix is PSD, since it can be written as a sum of the PSD matrices Π^{ij}.

(3) ⟹ (4): Trivial, since $D \pm |V| \in \mathcal{A}(Q)$.

(4) ⟹ (5): 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 $\rho(|I - \bar{Q}|) \leq 1$.

(5) $\Longrightarrow$ (6): Suppose that $\rho(|I-\bar{Q}|) \leq 1$. Since Q is irreducible, so is $|I-\bar{Q}|$; hence, by the Perron-Frobenius Theorem [8, 11], 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

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

Hence, $\bar{Q}$ is WSDD, and therefore so is Q.

(6) $\Longrightarrow$ (1): Suppose Q is WSDD. Then, there exists a diagonal matrix $U \succ 0$ such that UQU is WDD. By Lemma 4.1, UQU then has a 2×2D, say $UQU = \sum_{(i,j)\in P} \bar{\Pi}^{ij}$. Hence, a 2×2D of Q is given by $\Pi^{ij} := U^{-1}\bar{\Pi}^{ij}U^{-1}$. This completes the proof. □

Remark 4.1. *The proof of the implication* (5) $\Longrightarrow$ (6) *was taken from [13, Theorem 11], and is only included here for completeness.*

In Theorem 4.2, 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 (38), where all diagonal elements of Q_{11} are nonzero, and $Q_{22} = 0$ (the zero matrix). Thus Q is reducible.

Proposition 4.3. *Under the hypotheses of Theorem 4.2, assume in addition that Q is both 2×2-decomposable 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*

$$\pi_i^{ij} \ = \ \frac{t_i|Q_{ij}|\sqrt{Q_{ii}}}{\lambda\sqrt{Q_{jj}}}x_i^{-1}x_j + \frac{Q_{ii}(1-t_i)}{n-1} \qquad \forall\ i,\ j \in N\ ,\ i \neq j\ . \tag{65}$$

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, (65) 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, $h = 1, \ldots, k$ (cf. (38)), each of which is irreducible. Then, (65) can be applied to each of the blocks Q^h separately, where (λ, x) 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 (65) is unique:

Corollary 4.4. *Under the hypotheses of Theorem 4.2, 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 sparse, they are also rank-1.

Proposition 4.5. *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 (63) is equal to 1. Yet, another possible rank-1 decomposition is:

$$\begin{bmatrix} 2 & 2 & 1 \\ 2 & 5 & 1 \\ 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} \frac{4}{3} & 2 & 0 \\ 2 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} \frac{2}{3} & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & \frac{3}{2} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & \frac{1}{2} \end{bmatrix} \quad . \tag{66}$$

This is tied to the fact that one can choose different objective functions f in (86) (cf. Appendix B), leading to different rank-1 decompositions.

4.2. Heuristic approaches for approximate 2×2 decomposition

We now combine the results of §4.1 with those of §1 to propose fast heuristics for finding approximate 2×2Ds of a matrix Q without the need of solving large SDP problems like (39). The observation is that for any $Q \succeq 0$, we know:

- how to select the "largest" diagonal D such that $Q - D \succeq 0$, which just amounts to solving the "small" SDP (8), with any choice of α (or even using $||Q - diag(\delta)||^2$ as the objective function);
- the quick formula (65), 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 (8)). This choice immediately implies that $R(\varepsilon) \succeq 0$ for any $\varepsilon \geq 0$. Furthermore,

$$X(\varepsilon) = Q - R(\varepsilon) = (1 - \varepsilon)Q + \varepsilon D \quad .$$

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 (65), 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. (12)/(11)).

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 (8) 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 (8). Because $R(\varepsilon^*) = \varepsilon^*(Q - D)$, if one solves (8) 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 (8) before. But, clearly, $D + (1/\varepsilon^*)D'$ is a better solution to (8) 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.

5. Computational results and conclusions

We now present computational results aimed at assessing whether formulations employing the 2×2PR, i.e., (40)–(44), 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 instances of the Mean-Variance portfolio optimization problem, which is a well-known application of significant practical impact and for which several specialized algorithmic approaches have been developed (cf. e.g. [1, 2, 5, 15] and the references therein). The particular instances we use have belong to a family already extensively employed in the literature of MV problems [3, 4, 5, 6, 7, 15], are available at

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

and some of them have found their way in the recently proposed QPLIB [9]. The interested reader is referred to the provided 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). However, we could not use the instances of [3, 4, 5, 6, 7, 15], with $n \in \{200, 300, 400\}$, as some of the approaches do not scale to those sizes. Instead, we considered instances with $n \in \{25, 50\}$, constructed 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 in the "p" case, but not in the other two; hence, the matrices were, whenever necessary, corrected by adding to them the smallest possible diagonal that restored $Q \succeq 0$, a-la (8). 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 compared lower bounds provided by 6 different approaches:

1. D_s and D_l denote those obtained by the diagonal PR (7) when δ is obtained by solving (8) and (9), respectively;

2. $2\times2_s$ and $2\times2_l$ denote those obtained by the $\underline{2\times2\text{PR}}$ (the continuous relaxation of (40)–(44)) when the 2×2D is obtained by solving (39) and (58), respectively;

3. $2\times2_{h,s}$ and $2\times2_{h,l}$ denote those obtained by the $\underline{2\times2\text{PR}}$ (the continuous relaxation of (40)–(44)) when the 2×2D is obtained from the heuristic of §4.2, 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. The optimal integer solution has been obtained by running a B&C solver (in particular, `Cplex 12.7`) on the diagonal PR (7), with the D_l diagonal. It is useful to remark that, while "p", "z" and "n" instances can be solved quickly (cf. e.g. [3, 4] 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. That having been said, since the optimal solution was obtained with high accuracy (the default `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, often for the new "o", "y" and "m" instances very weak bounds were obtained, some of them being $lb = 0$; with the formula we adopted this gives a gap of $1 = 100\%$, whereas the standard formula would be ill-defined.

Presenting the results was not trivial, because they have a very significant variance not only between different classes of instances, but even within the same class. Hence, reporting averages was 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.

We do not report detailed data about running times, because our focus is on the quality of the bounds. Furthermore, total running times are the sum of the one required to obtain the decomposition, plus the one required to solve the $\underline{\text{PR}}$. Both processes are complex; in particular, solving the $\underline{\text{PR}}$ can be done in different ways, with performances potentially differing by orders of magnitude (cf. e.g. [3, 4, 7, 15]). Also, the details of the solution method used for the SDP can have a substantial impact [6]. However, it is worth reporting very aggregated figures giving at least the order of magnitude. On our reference Intel Core i7@2.5 Ghz, `Cplex 12.7` took about 0.03s to solve the $\underline{\text{PR}}$ (7), and 1.5s to solve the 2×2PR (40)–(44) for $n = 25$; for $n = 50$ the running times were instead about 0.07s and 57s, respectively. Clearly, solving (40)–(44) by standard means is not going to scale to large values of n. Similarly, solving (8), (9), (39) and (58) took about 0.21s, 0.88s, 5.5s and 7772s for $n = 25$; the running times were instead about 0.41s, 1.93s, 172s and 241013s for $n = 50$. Thus, finding the 2×2D by standard SDP approaches does not appear to be promising. However, it should be remarked that we only experimented with *dense* matrices: the number of variables and constraints in (40)–(44), (39) and (58) clearly depends to the number of nonzeroes in Q, and therefore the approach can actually be computationally feasible (which does not necessarily imply convenient) for nonseparable but *sparse* problems. The heuristic of §4.2 required only about 0.002s and 0.008s for $n = 25$ and $n = 50$, respectively (possibly on top of the D_s time, but only for the instances not having an exact 2×2D), and therefore clearly scales to large sizes; however, this is not particularly helpful as long as the solution of (40)–(44) remains too costly. Yet, improving the time for solving the $\underline{\text{PR}}$ and/or the SDP is conceptually possible: both have been done for the diagonal case [6, 3]. We therefore focus on the analysis of gaps, in order to gauge whether the whole idea of using

	D_s	D_l	$2\times2_s$	$2\times2_{h,s}$	$2\times2_{h,l}$	$2\times2_l$
25-p-a	2.28	2.21	2.21	2.21	2.21	2.21
25-p-c	2.48	2.36	2.30	2.30	2.30	2.30
25-p-e	1.83	1.46	3.70	3.70	3.70	1.29
25-p-g	2.16	1.75	1.74	1.74	1.74	1.71
25-p-i	0.71	0.56	2.33	2.33	2.33	0.55
25-o-a	3.95	2.83	2.81	2.81	2.81	2.80
25-o-c	8.23	5.09	2.48	2.48	2.48	2.47
25-o-e	17.26	15.30	8.90	8.90	8.90	8.90
25-o-g	10.33	8.11	3.06	3.06	3.06	3.05
25-o-i	14.93	13.65	6.58	6.58	6.58	6.58
25-z-a	2.39	1.84	16.18	16.26	16.26	1.50
25-z-c	3.06	2.44	15.27	15.83	15.72	1.64
25-z-e	0.02	0.00	0.07	0.07	0.08	0.00
25-z-g	1.55	1.32	1.33	1.39	1.55	1.21
25-z-i	0.98	0.81	0.97	1.00	0.98	0.78
25-y-a	100	100	100	100	100	100
25-y-c	100	100	100	100	100	100
25-y-e	2.14	2.10	0.32	0.32	0.32	0.21
25-y-g	24.57	24.57	9.14	9.14	9.14	8.71
25-y-i	15.96	15.97	3.49	3.49	3.49	1.61
25-n-a	2.79	1.78	11.97	12.10	11.24	0.27
25-n-c	3.04	2.04	12.12	13.04	12.66	1.42
25-n-e	2.00	1.39	7.94	9.81	9.74	0.38
25-n-g	4.24	4.08	4.19	4.23	4.12	4.06
25-n-i	2.68	1.37	8.60	9.15	9.43	0.35
25-m-a	100	100	100	100	100	100
25-m-c	99.55	99.55	98.59	98.59	98.59	98.59
25-m-e	100	100	100	100	100	100
25-m-g	5.32	5.32	4.79	4.79	4.79	4.66
25-m-i	100	100	100	100	100	100

Table 1: Gaps (in percentage) of the PR with different decompositions for $n = 25$

more complex decompositions than the diagonal one may possibly have any appeal.

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

- The gaps provided by the heuristic starting from D_s, $2\times2_{h,s}$, are almost always identical, or at least extremely close, to those obtained when starting from D_l, $2\times2_{h,l}$. When $2\times2_{h,l}$ 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 $2\times2_s$. 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 in details, we can add that the objective function values (in the sense of (39)) of the 2×2D produced by the heuristic are almost indistinguishable from those of the 2×2D produced by (39).

	D_s	D_l	$2\times2_s$	$2\times2_{h,s}$	$2\times2_{h,l}$	$2\times2_l$
50-p-a	1.47	1.23	1.20	1.20	1.20	1.20
50-p-c	2.45	2.16	2.11	2.11	2.11	2.11
50-p-e	0.89	0.33	1.40	1.40	1.40	0.26
50-p-g	0.83	0.73	0.72	0.72	0.72	0.68
50-p-i	1.33	1.24	1.25	1.25	1.25	1.24
50-o-a	18.17	14.26	3.31	3.31	3.31	3.31
50-o-c	15.71	9.32	2.44	2.44	2.44	2.44
50-o-e	21.84	17.18	7.95	7.95	7.95	7.95
50-o-g	19.84	15.90	6.32	6.32	6.32	6.31
50-o-i	5.24	2.35	2.27	2.27	2.27	2.27
50-z-a	0.46	0.30	0.41	0.45	0.42	0.27
50-z-c	2.79	1.53	10.28	11.61	11.21	0.62
50-z-e	2.86	1.16	10.97	11.45	11.30	0.51
50-z-g	3.46	2.52	11.62	12.32	12.16	2.24
50-z-i	2.92	1.79	3.65	4.72	4.83	1.36
50-y-a	8.80	8.80	0.16	0.16	0.16	0.14
50-y-c	99.76	99.76	98.09	98.09	98.09	98.09
50-y-e	100	100	100	100	100	100
50-y-g	100	100	100	100	100	100
50-y-i	94.62	94.62	73.27	73.27	73.27	73.26
50-n-a	3.99	2.83	2.60	3.97	3.24	2.43
50-n-c	4.67	2.85	7.68	10.85	9.89	1.05
50-n-e	3.12	1.54	9.23	11.90	10.85	0.36
50-n-g	3.04	1.66	1.51	2.56	1.68	1.45
50-n-i	3.92	1.98	6.75	10.10	9.07	0.57
50-m-a	59.97	59.97	15.47	15.47	15.47	15.38
50-m-c	100	100	100	100	100	100
50-m-e	100	100	100	100	100	100
50-m-g	36.35	36.35	6.03	6.03	6.03	6.02
50-m-i	98.51	98.51	90.22	90.22	90.22	90.22

Table 2: Gaps (in percentage) of the PR with different decompositions for $n = 50$

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 and also of D_s and D_l. 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, $2\times2_{h,l}$ 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 $D_l/2\times2_l$ and the corresponding "cheap" ones $D_s/2\times2_s$ 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. However, 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 [10], 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 2×2PR 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 dimension

Similarly to §2.3 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. §5), "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 (39)

$$\min \left\{ \| \Phi \|^2 \;:\; Q = \Phi + \textstyle\sum_{t\in T} E^t \Gamma^t (E^t)^T \;\;,\;\; \Gamma^t \succeq 0 \quad t \in T \;\;,\;\; \Phi \succeq 0 \right\} . \tag{67}$$

Any feasible solution of (67) 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 §2.3 we can easily define the corresponding 3x3PR:

$$\min\; x^T\Phi x + q^T x + c^T y + \textstyle\sum_{t\in T}\sum_{c\in C(t)} (x^{t,c})^T (\Gamma^t)^c x^{t,c}/y^{t,c} \tag{68}$$

(2)–(4)

$$x_i = \textstyle\sum_{c\in C(t)\,:\, i\in c} x_i^{t,c} \quad , \quad y_i = \sum_{c\in C(t)\,:\, i\in c} y^{t,c} \qquad t \in T \;\;,\;\; i \in N \tag{69}$$

$$l_i y^{t,c} \le x_i^{t,c} \le u_i y^{t,c} \qquad t \in T \;\;,\;\; c \in C(t) \;\;,\;\; i \in c \tag{70}$$

$$\textstyle\sum_{c\in C(t)} y^{t,c} \le 1 \qquad t \in T \tag{71}$$

$$y^{t,c} \in \{0,1\} \qquad t \in T \;\;,\;\; c \in C(t)\,. \tag{72}$$

As in §2.3, in (68) $(\Gamma^t)^c$ 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 (67) with (68)–(72) again yields the problem of finding the 3x3D providing the best bound. This again starts with the following

min-max analogous to (45)–(48)

$$\min_{x,y} q^T x + c^T y + \max_{\Phi,\Gamma} \langle\, \Phi\,,\, xx^T \,\rangle + \sum_{t\in T} \left\langle \sum_{c\in C(t)} \frac{\bar{x}^{t,c}(\bar{x}^{t,c})^T}{y^{t,c}}\,,\, \Gamma^t \right\rangle \tag{73}$$

$$(2)\text{–}(3) \quad , \quad (69)\text{–}(71)$$

$$y^{t,c} \in [0,1] \quad t \in T \quad , \quad c \in C(t) \tag{74}$$

$$y \in [0,1]^n \tag{75}$$

$$Q = \Phi + \textstyle\sum_{t\in T} E^t \Gamma^t (E^t)^T \quad , \quad \Gamma^t \succeq 0 \quad t \in T \quad , \quad \Phi \succeq 0 \tag{76}$$

where $\bar{x}^{t,c}$ in (73) 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. (49)), so as to express the equality constraint in (76) by

$$\begin{array}{ll} \sum_{t\in T:(i,j)\subset t} \langle\, O^{t,ij}\,,\, \Gamma^t \,\rangle + \langle\, O^{ij}\,,\, \Phi \,\rangle = 2Q_{ij} & (i,j) \in P \\ \sum_{t\in T:i\in t} \langle\, D^{t,i}\,,\, \Gamma^t \,\rangle + \langle\, D^{i}\,,\, \Phi \,\rangle = Q_{ii} & i \in N \end{array} \quad . \tag{77}$$

The dual of the inner SDP problem then is

$$\min \ \textstyle\sum_{(i,j)\in P} 2Q_{ij} f_{ij} + \sum_{i\in N} Q_{ii} f_i \tag{78}$$

$$\textstyle\sum_{(i,j)\in P} O^{ij} f_{ij} + \sum_{i\in N} D^i f_i \succeq xx^T \tag{52}$$

$$\sum_{(i,j)\subset t} O^{t,ij} f_{ij} + \sum_{i\in t} D^{t,i} f_i \succeq \sum_{c\in C(t)} \frac{\bar{x}^{t,c}(\bar{x}^{t,c})^T}{y^{t,c}} \qquad t \in T \tag{79}$$

and the nonlinear constraints (52) and (79) can be rewritten as

$$\begin{bmatrix} 1 & x^T \\ x & \sum_{(i,j)\in P} O^{ij} f_{ij} + \sum_{i\in N} D^i f_i \end{bmatrix} \succeq 0 \tag{54}$$

$$\textstyle\sum_{(i,j)\subset t} O^{t,ij} f_{ij} + \sum_{i\in t} D^{t,i} f_i \succeq \sum_{c\in C(t)} \bar{W}^{t,c} \qquad t \in T \tag{80}$$

$$\begin{bmatrix} W^{t,c} & x^{t,c} \\ (x^{t,c})^T & y^{t,c} \end{bmatrix} \succeq 0 \qquad t \in T \ , \ c \in C(t) \tag{81}$$

Each $W^{t,c}$ in (81) is a $|c| \times |c|$ matrix, and $\bar{W}^{t,c}$ in (80) 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, (73)–(76) then is

$$\min q^T x + c^T y + \textstyle\sum_{(i,j)\in P} 2Q_{ij} f_{ij} + \sum_{i\in N} Q_{ii} f_i$$
$$(2)\text{–}(3) \ , \ (69)\text{–}(71) \ , \ (74)\text{–}(75) \ , \ (54) \ , \ (80)\text{–}(81)$$

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 4.4. 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

$$\begin{aligned}\rho(|I-\bar{Q}|) \leq 1 \iff & |I-\bar{Q}|x = \lambda x \leq x \iff \textstyle\sum_{j:j\neq i} |\bar{Q}_{ij}|x_j \leq x_i = |\bar{Q}_{ii}|x_i \quad \forall\, i \in N \\ \iff & \textstyle\sum_{j:j\neq i} x_i|\bar{Q}_{ij}|x_j \leq |\bar{Q}_{ii}|x_i^2 \;\equiv\; \sum_{j:j\neq i} W_{ij} \leq W_{ii} \qquad \forall\, i \in N\,. \end{aligned} \tag{82}$$

Moreover, $\rho(|I-\bar{Q}|) = 1 \iff \sum_{j:j\neq i} W_{ij} = W_{ii}$. 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 (82) 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 (62) 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

$$\pi_i^{ij} = |W_{ij}| \quad \forall\, i,\ j \in N,\ i \neq j\,. \tag{83}$$

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 $n \times n$ (diagonal) matrix $\Delta^{ij} := \hat{\Pi}^{ij} - \Pi^{ij}$. 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 $\Delta_{ii}^{ij}, \Delta_{jj}^{ij} \geq 0$. So, suppose instead that (without loss of generality) $\Delta_{ii}^{ij} < 0$. Since $0 \preceq \hat{\Pi}^{ij} = \Pi^{ij} + \Delta^{ij}$, using (83) we obtain

$$\begin{aligned}|W_{ij}|^2 \;=\; (\hat{\Pi}_{ij}^{ij})^2 \;&\leq\; \hat{\Pi}_{ii}^{ij}\hat{\Pi}_{jj}^{ij} \;=\; (\Pi_{ii}^{ij} + \Delta_{ii}^{ij})(\Pi_{jj}^{ij} + \Delta_{jj}^{ij}) \\ &=\; (|W_{ij}| + \Delta_{ii}^{ij})(|W_{ij}| + \Delta_{jj}^{ij}) \;=\; |W_{ij}|^2 + |W_{ij}|(\Delta_{ii}^{ij} + \Delta_{jj}^{ij}) + \Delta_{ii}^{ij}\Delta_{jj}^{ij}\,.\end{aligned}$$

Simplifying and rearranging we obtain

$$\Delta_{jj}^{ij}(|W_{ij}| + \Delta_{ii}^{ij}) \geq -\Delta_{ii}^{ij}|W_{ij}|\,. \tag{84}$$

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

$$|W_{ij}| + \Delta_{ii}^{ij} = \Pi_{ii}^{ij} + \Delta_{ii}^{ij} = \hat{\Pi}_{ii}^{ij} \geq 0\,. \tag{85}$$

Moreover, note that if (85) holds with equality, then (84) gives $0 \geq -\Delta_{ii}^{ij}|W_{ij}| = (\Delta_{ii}^{ij})^2$, contradicting $\Delta_{ii}^{ij} < 0$. Hence, the inequality in (85) is strict, and thus (84) is equivalent to

$$\Delta_{jj}^{ij} \geq \frac{-\Delta_{ii}^{ij}|W_{ij}|}{|W_{ij}| + \Delta_{ii}^{ij}} > \frac{-\Delta_{ii}^{ij}|W_{ij}|}{|W_{ij}|} = -\Delta_{ii}^{ij}\,.$$

Hence, $tr(\Delta^{ij}) = \Delta_{ii}^{ij} + \Delta_{jj}^{ij} > 0$, 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 4.5. 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 $2 \le i \le n-1$, then $f(i,n) = i$. With $\Pi = [\Pi^{ij}]_{(i,j)\in P}$ consider the optimization problem

$$\max \left\{ \, g(\Pi) = \textstyle\sum_{\{i,j\}\in\bar{P}} \pi^{ij}_{f(i,j)} \ : \ (59)\text{–}(61) \, \right\} . \tag{86}$$

Since Q has a 2×2D, (86) 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 (60) are all active. Indeed, suppose by way of contradiction that, for some $\{i,j\} \in \bar{P}$,

$$\bar{\pi}^{ij}_i \bar{\pi}^{ij}_j > Q^2_{ij} \ (\ge 0) \ , \tag{87}$$

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

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

We claim that for all sufficiently small $\epsilon > 0$, $\Pi(\epsilon)$ is feasible in (86). Indeed, (61) and (60) hold since by (87) we have

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

Furthermore, (59) holds since

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

Moreover,

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

contradicting the optimality of $\bar{\Pi}$. Hence, the inequalities (60) 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 less than or equal to 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 projects PRIN 2012JXB3YF and PRIN 2015B5F27W funded by the Italian Minister for Education, and the networking support of the COST Action TD1207 of the European Union. We are also grateful to Jeff Linderoth and Robert Weismantel for useful discussions.

References

[1] C. Buchheim, M. De Santis, F. Rinaldi, and L. Trieu, "A Frank-Wolfe based branch-and-bound algorithm for mean-risk optimization," arXiv:1507.05914v4 [math.OC], 2017.

[2] D. Di Lorenzo, G. Liuzzi, F. Rinaldi, F. Schoen, and M. Sciandrone, "A concave optimization-based approach for sparse portfolio selection," *Optimization Methods and Software*, vol. 27, no. 6, pp. 983–1000, 2012.

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

[4] 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.

[5] A. Frangioni and C. Gentile, "Perspective Cuts for a Class of Convex 0–1 Mixed Integer Programs," *Mathematical Programming*, vol. 106, no. 2, pp. 225 – 236, 2006.

[6] A. Frangioni and C. Gentile, "SDP Diagonalizations and Perspective Cuts for a Class of Nonseparable MIQP," *Operations Research Letters*, vol. 35, no. 2, pp. 181 – 185, 2007.

[7] A. Frangioni and C. Gentile, "A Computational Comparison of Reformulations of the Perspective Relaxation: SOCP vs. Cutting Planes," *Operations Research Letters*, vol. 37, no. 3, pp. 206 – 210, 2009.

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

[9] F. Furini, E. Traversi, P. Belotti, A. Frangioni, A. Gleixner, N. Gould, L. Liberti, A. Lodi, R. Misener, H. Mittelmann, N. Sahinidis, S. Vigerske, and A. Wiegele, "QPLIB: A library of quadratic programming instances," Optimization Online 5846, 2017.

[10] H. Jeon, J. Linderoth, and A. Miller, "Quadratic cone cutting surfaces for quadratic programs with onoff constraints," *Discrete Optimization*, vol. online first, 2014.

[11] O. Perron, "Zur theorie der matrices," *Mathematische Annalen*, vol. 64, pp. 248–263, 1907.

[12] R. Rockafellar, *Convex Analysis.* Princeton, NJ: Princeton University Press, 1970.

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

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

[15] 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*, vol. 26, no. 4, pp. 690–703, 2014.