SDP-quality bounds via convex quadratic relaxations for global optimization of mixed-integer quadratic programs

Carlos J. Nohra[*]   Arvind U. Raghunathan[†]   Nikolaos V. Sahinidis[‡]

June 28, 2021

Abstract

We consider the global optimization of nonconvex mixed-integer quadratic programs with linear equality constraints. In particular, we present a new class of convex quadratic relaxations which are derived via quadratic cuts. To construct these quadratic cuts, we solve a separation problem involving a linear matrix inequality with a special structure that allows the use of specialized solution algorithms. Our quadratic cuts are nonconvex, but define a convex feasible set when intersected with the equality constraints. We show that our relaxations are an outer-approximation of a semi-infinite convex program which under certain conditions is equivalent to a well-known semidefinite program relaxation. The new relaxations are implemented in the global optimization solver BARON, and tested by conducting numerical experiments on a large collection of problems. Results demonstrate that, for our test problems, these relaxations lead to a significant improvement in the performance of BARON.

1 Introduction

We address the global optimization of problems of the form:

$$\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & x^T Q x + q^T x \\
\text{s.t.} \quad & Ax = b \\
& x_i \in S_i, \ \forall i \in [n] := \{1, \dots, n\}
\end{aligned} \quad (1)$$

where $Q \in \mathbb{R}^{n \times n}$ is a symmetric matrix which may be indefinite, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. For each $i \in [n]$, we assume that S_i is a bounded set given by the union of finitely many closed intervals in $\mathbb{R}$.

The formulation in (1) subsumes many classes of problems such as nonconvex quadratic programs (QPs) and mixed-integer quadratic programs (MIQPs), which arise in applications including facility location and quadratic assignment [9], molecular conformation [15] and max-cut problems [7].

State-of-the-art global optimization solvers rely on branch-and-bound algorithms to solve (1) to global optimality. The efficiency of these algorithms depends to a large extent on the tightness and the computational cost of the relaxations solved for lower bounding.

[*]Department of Chemical Engineering, Carnegie Mellon University (`cnohrakh@andrew.cmu.edu`).

[†]Mitsubishi Electric Research Laboratories (`raghunathan@merl.com`)

[‡]School of Chemical and Biomolecular Engineering, Georgia Institute of Technology (`nikos@gatech.edu`).

Commonly used relaxations for (1) are polyhedral relaxations constructed via factorable programming [11, 18, 21] and reformulation-linearization techniques (RLT) [17], semi-definite programming (SDP) relaxations [2, 4, 19], and convex quadratic relaxations which can be obtained via separable programming [14], d.c. programming [22], or quadratic convex reformulation methods [3].

In a recent paper [12], we derived convex quadratic relaxations of (1) by convexifing the objective function through uniform diagonal perturbations of Q. These perturbations were constructed by solving eigenvalue and generalized eigenvalue problems involving Q and A. Through numerical experiments, we demonstrated that these relaxations are not only inexpensive to solve, but can also provide very tight bounds, significantly improving the performance of branch-and-bound algorithms. Motivated by these results, in the current paper, we consider a related class of convex quadratic relaxations. In particular, we investigate quadratically constrained programming (QCP) relaxations for (1). These relaxations are derived via quadratic cuts obtained from nonuniform diagonal perturbations of Q. We show that these relaxations are at least as tight as the spectral relaxations in [12], and provide a very good approximation of the bounds given by certain SDP relaxations of (1).

An SDP relaxation for (1) obtained after adding RLT inequalities was considered by Saxena et al. [16], who proposed a procedure to project the feasible region of this relaxation onto the space of the original variables. This projection relies on convex quadratic cuts derived from an SDP separation program which is solved by applying a sub-gradient-based algorithm. Even though the relaxations generated through this approach are nearly as tight as the original SDP formulation, the separation problem used to derive the quadratic cuts can be very expensive to solve.

Dong [5] used a different SDP relaxation for (1) which only includes the diagonal RLT inequalities and showed that this SDP is equivalent to a particular semi-infinite program. This semi-infinite formulation served as a motivation to construct convex QCP relaxations for (1) via convex quadratic cuts derived from a semidefinite separation problem. To ensure that the solution of the separation problem is finitely attained, Dong [5] proposed a regularized version of this semidefinite program, and demonstrated that it can be solved very efficiently through a specialized coordinate descent algorithm. Our work is partially inspired by the ideas proposed by Dong [5]. We refine his approach in several directions and make various theoretical and algorithmic contributions.

Our first contribution is a new class of convex QCP relaxations for (1) constructed by using information from both Q and the equality constraints $Ax = b$. These relaxations are derived from a semi-infinite program that generalizes the semi-infinite programming formulation proposed in [5]. Under our approach, we use the matrix $A^T A$ in order to modify the semidefinite constraint of the separation problem solved in [5]. This modification allows us to construct convex QCP relaxations which are at least as tight as those considered in [5]. Unlike the quadratic cuts introduced in [5], the quadratic cuts obtained with our approach are nonconvex. However, as we show in §2.1, these nonconvex quadratic cuts define a convex feasible set when intersected with the equality constraints $Ax = b$, resulting in a convex relaxation of (1).

In our second contribution, we provide conditions under which our semi-infinite programming formulation is equivalent to a well-known SDP relaxation of (1). Moreover, we show that this SDP is the best relaxation in the class of SDP relaxations considered in this paper.

In our third contribution, we present a new analysis of the separation problem used to derive our quadratic cuts. In particular, we provide results on the finite attainment of

2

this SDP by using its dual formulation. These results also apply to the separation problem in [5], which is a special case of ours.

In our fourth contribution, we propose a new regularization approach for the semidefinite separation problem and modify the coordinate descent algorithm introduced in [5] accordingly. Through numerical experiments, we show that the quadratic cuts derived from our regularized separation problem provide a much better approximation of certain SDP bounds than the quadratic cuts obtained from the regularized separation problem proposed in [5].

In order to assess the computational benefits of the proposed techniques, we implement the new quadratic relaxations in the global optimization solver BARON. These relaxations are incorporated into BARON's portfolio of relaxations and invoked according to a dynamic relaxation selection strategy introduced in [12]. We test our implementation on a large set of problems. Numerical results show that the new quadratic relaxations lead to a significant improvement in the performance of BARON, resulting in a new version of this solver which outperforms other state-of-the-art solvers such as CPLEX and GUROBI for many of our test problems.

The remainder of this paper is organized as follows. In §2 we introduce the relaxations considered in this paper and investigate their theoretical properties. In §3, we provide a new analysis on the finite attainment of the semidefinite separation problem and present our regularization approach. In §4 we introduce the version of coordinate minimization algorithm used to solve our regularized separation problem. This is followed by a description of our implementation in §5. In §6, we present an extensive computational study which investigates the effectiveness of our regularization approach, the impact of the proposed relaxations on the performance of BARON, and the relative performance of several state-of-the-art global optimization solvers. Finally, in §7 we present conclusions from this work.

In what follows, we denote by $\mathbb{R}$ and $\mathbb{R}_{\geq 0}$, the set of real numbers and nonnegative real numbers, respectively. We use $\mathbb{1} \in \mathbb{R}^n$ to denote a vector of ones. For a matrix $A \in \mathbb{R}^{m \times n}$, we use $A_{i\cdot}$, $A_{\cdot j}$ and A_{ij} to denote its i-th row, j-th column and (i, j)-th entry, respectively. Let $M, N \in \mathbb{S}^n$, where $\mathbb{S}^n$ is the set of $n \times n$ real, symmetric matrices, and $N \succ 0$. We use $\lambda_{\min}(M)$ to represent the smallest eigenvalue of M. Similarly, we denote by $\lambda_{\min}(M, N)$ the smallest generalized eigenvalue of the problem $Mv = \lambda Nv$, where $v \in \mathbb{R}^n$. The inner product between matrices $M, P \in \mathbb{S}^n$ is denoted by $\langle M, P \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} P_{ij}$. Throughout this paper, we assume that A has rank m and use $Z \in \mathbb{R}^{n \times (n-m)}$ to denote an orthonormal basis for the nullspace of A.

2 Construction and theoretical analysis of convex quadratic relaxations

2.1 A family of semi-infinite programming relaxations

We start by considering the following reformulation of (1):

$$\left. \begin{array}{ll} \min_{x,y,v} & v + q^T x \\ \text{s.t.} & v \geq x^T \left(Q + \text{diag}(d) \right) x - d^T y + \alpha \|Ax - b\|^2 \\ & Ax = b \\ & (x_i, y_i) \in \mathcal{C}_i, \ \forall i \in [n] \end{array} \right\} \tag{2}$$

3

where $y \in \mathbb{R}^n$, $v \in \mathbb{R}$, $d \in \mathbb{R}^n$ is a vector used to perturb the diagonal entries of Q, $\alpha \in \mathbb{R}_{\geq 0}$, and $\mathcal{C}_i := \{(x_i, y_i) \in \mathbb{R}^2 : x_i \in S_i, \ y_i = x_i^2\}$. Define $L_i := \min\{s \in \mathbb{R} : s \in S_i\}$ and $U_i := \max\{s \in \mathbb{R} : s \in S_i\}$. It is simple to show that the convex hull of $\mathcal{C}_i$ is given by (see Proposition 1 in [5]):

$$\text{conv}(\mathcal{C}_i) = \{(x_i, y_i) \in \mathbb{R}^2 : L_i \leq x_i \leq U_i, \ l_i(x_i) \leq y_i \leq u_i(x_i)\} \tag{3}$$

where $l_i(\cdot)$ is the tightest convex extension of x_i^2 when x_i is restricted to S_i (see [20] for convex extensions) and $u_i(\cdot)$ is the concave envelope of x_i^2 over $[L_i, U_i]$. By replacing $\mathcal{C}_i$ with $\text{conv}(\mathcal{C}_i)$ in (2), we obtain the following relaxation of (1):

$$\left. \begin{array}{cl} \min\limits_{(x,y) \in \mathcal{F}, v} & v + q^T x \\[2mm] \text{s.t.} & v \geq x^T (Q + \text{diag}(d)) x - d^T y + \alpha \|Ax - b\|^2 \end{array} \right\} \tag{4}$$

where $\mathcal{F} = \{x, y \in \mathbb{R}^n : Ax = b, \ (x_i, y_i) \in \text{conv}(\mathcal{C}_i), \ \forall i \in [n]\}$. Let $\mathcal{D}_\alpha := \{d \in \mathbb{R}^n : Q + \text{diag}(d) + \alpha A^T A \succcurlyeq 0\}$. Clearly, (4) is a convex problem for any vector $d \in \mathcal{D}_\alpha$. By considering all such vectors, we obtain the following semi-infinite convex program (SICP):

$$\min_{(x,y) \in \mathcal{F}, v} \quad v + q^T x \tag{5a}$$

$$\text{s.t.} \ \ v \geq x^T (Q + \text{diag}(d)) x - d^T y + \alpha \|Ax - b\|^2, \ \forall d \in \mathcal{D}_\alpha. \tag{5b}$$

Since any solution feasible in (2), is feasible in (5) as well, this SICP also is a relaxation of (1). To illustrate this, let $(\bar{x}, \bar{y}, \bar{v})$ be feasible to (2). For each $i \in [n]$, we have $(\bar{x}_i, \bar{y}_i) \in \mathcal{C}_i \subseteq \text{conv}(\mathcal{C}_i)$. Moreover, since $\bar{y}_i = \bar{x}_i^2$, $\forall i \in [n]$, and $A\bar{x} = b$, we have $\bar{v} = \bar{x}^T (Q + \text{diag}(d)) \bar{x} - d^T \bar{y} + \alpha \|A\bar{x} - b\|^2 = \bar{x}^T Q \bar{x}$. By replacing $(\bar{x}, \bar{y}, \bar{v})$ in (5b), it is easy to see this inequality is satisfied $\forall d \in \mathcal{D}_\alpha$.

The quadratic term $\alpha \|Ax - b\|^2$ in (5b) vanishes for any x feasible in (5). As a result, we can drop this term from (5b) in order to obtain the following simplified version of (5):

$$\min_{(x,y) \in \mathcal{F}, v} \quad v + q^T x \tag{6a}$$

$$\text{s.t.} \ \ v \geq x^T (Q + \text{diag}(d)) x - d^T y, \ \forall d \in \mathcal{D}_\alpha. \tag{6b}$$

The quadratic cuts (6b) are not necessarily convex because, for $d \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}_{\geq 0}$, the condition $Q + \text{diag}(d) + \alpha A^T A \succcurlyeq 0$ does not necessarily imply that $Q + \text{diag}(d) \succcurlyeq 0$. However, the quadratic cuts (6b) need not be convex in order for (6) to be a convex optimization problem. As we show in the next proposition, the quadratic cuts (6b) define a convex feasible set when intersected with the constraints $Ax = b$, making (6) a convex relaxation of (1).

Proposition 1. *The semi-infinite program* (6) *is a convex optimization problem.*

Proof. This proof relies on the projection of the feasible set of (6) onto the nullspace of A. Let $\mathcal{H} = \{x \in \mathbb{R}^n : Ax = b\}$. Clearly, any point satisfying $Ax = b$ can be expressed as $x = \hat{x} + Zx_z$, where $\hat{x} \in \mathcal{H}$ and $x_z \in \mathbb{R}^{n-m}$. By using this transformation, (6) can be equivalently written as:

$$\min_{x_z, y, v} \ v + q^T (\hat{x} + Zx_z) \tag{7a}$$

$$\text{s.t.} \ \ v \geq (\hat{x} + Zx_z)^T (Q + \text{diag}(d)) (\hat{x} + Zx_z) - d^T y, \ \forall d \in \mathcal{D}_\alpha \tag{7b}$$

$$L_i \leq \hat{x}_i + e_i^T Z x_z \leq U_i, \ \forall i \in [n] \tag{7c}$$

$$l_i \left(\hat{x}_i + e_i^T Z x_z \right) \leq y_i \leq u_i \left(\hat{x}_i + e_i^T Z x_z \right), \ \forall i \in [n]. \tag{7d}$$

To prove that (7) is convex, it suffices to show that $Z^T \left(Q + \mathrm{diag}(d) \right) Z$ is positive semidefinite. By definition, any vector $d \in \mathcal{D}_\alpha$ satisfies:

$$w^T \left(Q + \mathrm{diag}(d) + \alpha A^T A \right) w \geq 0, \ \forall w \in \mathbb{R}^n. \tag{8}$$

Let $w = Z w_z$, where $w_z \in \mathbb{R}^{n-m}$. For this choice of w, (8) becomes

$$w_z^T Z^T \left(Q + \mathrm{diag}(d) \right) Z w_z \geq 0, \ w_z \in \mathbb{R}^{n-m}. \tag{9}$$

Clearly, (9) holds for all vectors $w_z \in \mathbb{R}^{n-m}$. Hence, $Z^T \left(Q + \mathrm{diag}(d) \right) Z$ is positive semidefinite for any $d \in \mathcal{D}_\alpha$. This completes the proof. $\qquad\square$

By setting $\alpha = 0$ in (6), we obtain the following SICP relaxation of (1) which was considered in [5]:

$$\min_{(x,y) \in \mathcal{F}, v} \ v + q^T x \tag{10a}$$

$$\text{s.t.} \ v \geq x^T \left(Q + \mathrm{diag}(d) \right) x - d^T y, \ \forall d \in \mathcal{D}_0 \tag{10b}$$

where $\mathcal{D}_0 := \{ d \in \mathbb{R}^n : Q + \mathrm{diag}(d) \succcurlyeq 0 \}$. The formulation in (10) served as a motivation in [5] to develop an algorithm to construct convex relaxations for (1) by using a finite number of quadratic cuts of the form (10b). As we demonstrate in §2.4, a similar algorithm can be devised based on the SICP (6).

In (6), the set $\mathcal{D}_\alpha$ is parameterized by the scalar α. An interesting question that arises in this context is how we can choose α to obtain the tightest relaxation of the form (6). This question is addressed by the following proposition.

Proposition 2. *Let α_1 and α_2 be real scalars such that $0 \leq \alpha_1 \leq \alpha_2$. Denote by μ_{SICPda1} and μ_{SICPda2} the optimal objective function values in the SICP (6) for α_1 and α_2, respectively. Define $\mathcal{D}_\infty := \{ d \in \mathbb{R}^n : Z^T (Q + \mathrm{diag}(d)) Z \succcurlyeq 0 \}$. Then, the following holds:*

(i) $\mu_{\mathrm{SICPda2}} \geq \mu_{\mathrm{SICPda1}}$.

(ii) The tightest relaxation of the form (6) is obtained when $\alpha \to \infty$.

(iii) $\lim_{\alpha \to \infty} \mathcal{D}_\alpha = \mathcal{D}_\infty$.

Proof. We start with the proof of (i). Let $\mathcal{D}_{\alpha_1}$ and $\mathcal{D}_{\alpha_2}$ be the sets of diagonal perturbations parameterized by α_1 and α_2, respectively. To prove the claim in (i), it suffices to show that $\mathcal{D}_{\alpha_1} \subseteq \mathcal{D}_{\alpha_2}$. Let $\bar{d} \in \mathcal{D}_{\alpha_1}$. By definition, $\bar{d}$ satisfies:

$$w^T \left(Q + \mathrm{diag}(\bar{d}) + \alpha_1 A^T A \right) w \geq 0, \ \forall w \in \mathbb{R}^n. \tag{11}$$

As $A^T A \succcurlyeq 0$ and $\alpha_2 - \alpha_1 \geq 0$, we have that $(\alpha_2 - \alpha_1) w^T A^T A w \geq 0, \ \forall w \in \mathbb{R}^n$. This condition combined with (11) implies

$$w^T \left(Q + \mathrm{diag}(\bar{d}) + \alpha_2 A^T A \right) w \geq 0, \ \forall w \in \mathbb{R}^n. \tag{12}$$

It follows that $\bar{d} \in \mathcal{D}_{\alpha_2}$. Hence, $\mathcal{D}_{\alpha_1} \subseteq \mathcal{D}_{\alpha_2}$, which completes the proof of (i). The claim in (ii) follows directly from (i). To prove (iii), it suffices to show that, for any $\bar{d} \in \mathcal{D}_\infty$, the following condition holds:

$$\lim_{\alpha \to \infty} w^T \left(Q + \mathrm{diag}(\bar{d}) + \alpha A^T A \right) w \geq 0, \ \forall w \in \mathbb{R}^n. \tag{13}$$

Clearly, any $w \in \mathbb{R}^n$ can be written as $w = w_A + Z w_z$, where $w_A \in \mathrm{range}(A^T)$ and $w_z \in \mathbb{R}^{n-m}$. Suppose that $w_A \neq 0$. Then, $w^T A^T A w = w_A^T A^T A w_A > 0$, and it is easy to show that (13) holds in the limit as $\alpha \to \infty$. Now, assume that $w_A = 0$. Since $AZ = 0$, the left-hand side of (13) reduces to $w_z^T Z^T \left(Q + \mathrm{diag}(\bar{d}) \right) Z w_z$, which is nonnegative because $\bar{d} \in D_\infty$. This proves the claim in (iii). □

A direct consequence of Proposition 2(i) is that, for any $\alpha > 0$, the bound provided by (6) is at least as large as that given by (10).

2.2 Relationship between the semi-infinite and semi-definite formulations

In [5], it was shown that the SICP (10) is equivalent to the following SDP relaxation of (1) (see Section 2 in [5] for details):

$$\min_{(x,y) \in \mathcal{F}, X} \ \langle Q, X \rangle + q^T x \tag{14a}$$

$$\text{s.t.} \ \ X - xx^T \succcurlyeq 0 \tag{14b}$$

$$X_{ii} = y_i, \ \forall i \in [n]. \tag{14c}$$

Motivated by this result, in this section we investigate the relationship between the SICP (6) and the following SDP relaxation of (1):

$$\min_{(x,y) \in \mathcal{F}, X} \ \langle Q, X \rangle + q^T x \tag{15a}$$

$$\text{s.t.} \ \ X - xx^T \succcurlyeq 0 \tag{15b}$$

$$X_{ii} = y_i, \ \forall i \in [n] \tag{15c}$$

$$\langle A^T A, X \rangle - 2(A^T b)^T x + b^T b = 0. \tag{15d}$$

We start by showing that, for any $\alpha \geq 0$, the optimal solution of the SICP (6) is bounded by the optimal solutions of the SDPs (14) and (15).

Proposition 3. *Assume that $\alpha \geq 0$ in (6). Denote by μ_{SICPda}, μ_{SDPd} and μ_{SDPda} the optimal objective function values in (6), (14) and (15), respectively. Then, $\mu_{\mathrm{SDPd}} \leq \mu_{\mathrm{SICPda}} \leq \mu_{\mathrm{SDPda}}$.*

Proof. We start by proving that $\mu_{\mathrm{SDPd}} \leq \mu_{\mathrm{SICPda}}$. Let μ_{SICPd} be the optimal solution of the SICP (10). Since $\alpha \geq 0$, Proposition 2(i) implies that $\mu_{\mathrm{SICPd}} \leq \mu_{\mathrm{SICPda}}$. By Theorem 1 in [5] we have that $\mu_{\mathrm{SICPd}} = \mu_{\mathrm{SDPd}}$. Hence, $\mu_{\mathrm{SDPd}} \leq \mu_{\mathrm{SICPda}}$. To prove that $\mu_{\mathrm{SICPda}} \leq \mu_{\mathrm{SDPda}}$, it suffices to show that for any $(\bar{x}, \bar{y}, \bar{X})$ feasible in (15) and $\bar{d} \in \mathcal{D}_\alpha$, the following condition holds:

$$\langle Q, \bar{X} \rangle \geq \bar{x}^T (Q + \mathrm{diag}(\bar{d})) \bar{x} - \bar{d}^T \bar{y}. \tag{16}$$

To this end, consider the following inequality:

$$\langle Q + \mathrm{diag}(\bar{d}) + \alpha A^T A, \bar{X} - \bar{x} \bar{x}^T \rangle \geq 0 \tag{17}$$

which is valid by the feasibility of $\bar{x}$ and $\bar{X}$ in (15) and the self-duality of the positive semi-definite cone. This inequality can be equivalently written as:

$$\langle Q, \bar{X} \rangle \geq \bar{x}^T (Q + \operatorname{diag}(\bar{d}))\bar{x} - \sum_{i=1}^{n} d_i \bar{X}_{ii} - \alpha \langle A^T A, \bar{X} \rangle + \alpha \bar{x}^T A^T A \bar{x}. \tag{18}$$

From the feasibility of $(\bar{x}, \bar{y}, \bar{X})$ in (15), it follows that $\bar{X}_{ii} = \bar{y}_i$, $\forall i \in [n]$, and $\langle A^T A, \bar{X} \rangle = 2b^T A\bar{x} - b^T b$. Then, the inequality in (18) becomes:

$$\langle Q, \bar{X} \rangle \geq \bar{x}^T (Q + \operatorname{diag}(\bar{d}))\bar{x} - \bar{d}^T \bar{y} + \alpha (A\bar{x} - b)^T (A\bar{x} - b). \tag{19}$$

Since $A\bar{x} - b = 0$, (19) is equivalent to (16), which completes the proof. $\qquad\square$

Now, we consider the case in which $\alpha \to \infty$ in the SICP (6). By Proposition 2(iii), the resulting SICP can be written as:

$$\min_{(x,y)\in\mathcal{F},v} \quad v + q^T x \tag{20a}$$

$$\text{s.t.} \quad v \geq x^T (Q + \operatorname{diag}(d)) x - d^T y, \ \forall d \in \mathcal{D}_\infty. \tag{20b}$$

Proposition 2(ii) implies that (20) is the tightest relaxation of the form (6) that can be constructed for (1). Moreover, from Proposition 3, we know that (15) provides an upper bound on the optimal solution of (20). Therefore, an important question is the existence of conditions under which these two relaxations are equivalent. This question is addressed in the remainder of this section.

In [5], the equivalence between the SICP (10) and the SDP (14) was established by applying strong duality to the SDP (14). However, unlike the SDP (14), the SDP (15) does not admit a strictly feasible solution. To illustrate this, note that, for any x satisfying $Ax = b$, (15d) can be equivalently written as:

$$\langle A^T A, X \rangle - \left(2A^T b\right)^T x + b^T b + \langle A^T A, xx^T \rangle - \langle A^T A, xx^T \rangle = 0 \tag{21a}$$

$$\implies \langle A^T A, X - xx^T \rangle = 0 \tag{21b}$$

which implies that $X - xx^T$ cannot be positive definite for the pairs (x, X) that are feasible in (15). It follows that we cannot apply strong duality to (15). As a result, in order to show the equivalence between the SICP (20) and the SDP (15), we will rely on an auxiliary SDP which we will derive from (15).

We begin by characterizing the set of symmetric matrices $X \in \mathbb{S}^n$ which are feasible in (15).

Proposition 4. *Let the $\bar{x} \in \mathbb{R}^n$ and $\bar{X} \in \mathbb{S}^n$ be feasible in the SDP (15). Then $\bar{X}$ has the form $\bar{X} = \bar{x}\bar{x}^T + Z\bar{W}_z Z^T$, where $\bar{W}_z \in \mathbb{S}^{n-m}$ and $\bar{W}_z \succeq 0$.*

Proof. For any $(\bar{x}, \bar{X})$ feasible in (15) we can express $\bar{X}$ as $\bar{X} = \bar{x}\bar{x}^T + \widetilde{X}$ with $\widetilde{X} \in \mathbb{S}^n$ and $\widetilde{X} \succeq 0$. As mentioned previously, the constraint (15d) implies that $\langle A^T A, \bar{X} - \bar{x}\bar{x}^T \rangle = 0$, which can be equivalently written as $\langle A^T A, \widetilde{X} \rangle = 0$. Let the eigenvalue decomposition of $\widetilde{X}$ be given by $\widetilde{X} = \sum_{i=1}^{p} \lambda_i v_i v_i^T$, where $p \leq n$. Since $\widetilde{X} \succeq 0$, we have that $\lambda_i > 0$, $\forall i = 1, \ldots, p$. By using this decomposition in $\langle A^T A, \widetilde{X} \rangle = 0$ we obtain

$$\langle A^T A, \widetilde{X} \rangle = \sum_{i=1}^{p} \lambda_i \|Av_i\|^2 = 0 \tag{22a}$$

7

$$\implies Av_i = 0, \ \forall i = 1, \ldots, p. \tag{22b}$$

This shows that $\widetilde{X}$ is of the form $Z\bar{W}_z Z^T$ with $\bar{W}_z \in \mathbb{S}^{n-m}$ and $\bar{W}_z \succeq 0$. □

Now, let $\mathcal{H} = \{x \in \mathbb{R}^n : Ax = b\}$. Clearly, any point satisfying $Ax = b$ can be expressed as

$$x = \hat{x} + Zx_z \tag{23}$$

where $\hat{x} \in \mathcal{H}$ and $x_z \in \mathbb{R}^{n-m}$. From Proposition 4 we know that any X feasible in (15) is of the form $X = xx^T + ZW_z Z^T$, where $W_z \in \mathbb{S}^{n-m}$ and $W_z \succeq 0$. By using (23) in the expression for X we obtain $X = \hat{x}\hat{x}^T + \hat{x}(Zx_z)^T + (Zx_z)\hat{x}^T + Zx_z x_z^T Z^T + ZW_z Z^T$. Define $X_z := W_z + x_z x_z^T$ with $X_z \in \mathbb{S}^{n-m}$. Then X can be expressed as

$$X = \hat{x}\hat{x}^T + \hat{x}(Zx_z)^T + (Zx_z)\hat{x}^T + ZX_z Z^T. \tag{24}$$

By substituting (23) and (24) in (15), we can cast this SDP as follows:

$$\min_{x_z, y, X_z} \quad \langle Q, ZX_z Z^T \rangle + (2Q\hat{x} + q)^T (Zx_z) + \hat{x}^T Q\hat{x} + q^T \hat{x} \tag{25a}$$

$$\text{s.t.} \quad X_z - x_z x_z^T \succcurlyeq 0 \tag{25b}$$

$$\hat{x}_i^2 + 2\hat{x}_i \left(e_i^T Zx_z\right) + e_i^T ZX_z Z^T e_i = y_i, \ \forall i \in [n] \tag{25c}$$

$$L_i \le \hat{x}_i + e_i^T Zx_z \le U_i, \ \forall i \in [n] \tag{25d}$$

$$l_i \left(\hat{x}_i + e_i^T Zx_z\right) \le y_i \le u_i \left(\hat{x}_i + e_i^T Zx_z\right), \ \forall i \in [n]. \tag{25e}$$

As we show later in this section, under certain conditions, the SDP in (25) admits a strictly feasible solution. Based on this key observation, in the next theorem we rely on strong duality holding for (25) in order to show the equivalence between the SICP (20) and the SDP (15).

Theorem 1. *Let μ_{SDPda} and $\mu_{\mathrm{SICPda\infty}}$ denote the optimal objective function values in (15) and (20), respectively. Assume that the SDP (25) admits a strictly feasible solution. Then, $\mu_{\mathrm{SDPda}} = \mu_{\mathrm{SICPda\infty}}$.*

Proof. To prove that $\mu_{\mathrm{SDPda}} = \mu_{\mathrm{SICPda\infty}}$, we rely on the SDP (25) and its dual. Denote by μ_{SDPdaz} and μ_{DSDPdaz}, the optimal objective function values of the SDP (25) and its dual, respectively. Since (25) admits a strictly feasible solution, Slater's condition is satisfied by (25), which implies that strong duality holds for this SDP and its dual. Hence, $\mu_{\mathrm{SDPdaz}} = \mu_{\mathrm{DSDPdaz}}$. Moreover, as the SDPs (15) and (25) are equivalent, we have that $\mu_{\mathrm{SDPda}} = \mu_{\mathrm{SDPdaz}} = \mu_{\mathrm{DSDPdaz}}$.

Now, we construct the dual of (25). Let $d_i \in \mathbb{R}$, $i \in [n]$, be the multipliers associated with the constraints (25c). Then, the dual of (25) is given by

$$\max_{d \in \mathbb{R}^n} \left\{ \begin{array}{ll} \min\limits_{x_z, y, X_z} & \langle Q_{d,Z}, X_z \rangle + q_{d,\hat{x}}^T (Zx_z) + k_{d,\hat{x}} - d^T y \\[2mm] \text{s.t.} & X_z - x_z x_z^T \succcurlyeq 0 \\[1mm] & L_i \le \hat{x}_i + e_i^T Zx_z \le U_i, \ \forall i \in [n] \\[1mm] & l_i \left(\hat{x}_i + e_i^T Zx_z\right) \le y_i \le u_i \left(\hat{x}_i + e_i^T Zx_z\right), \ \forall i \in [n] \end{array} \right\} \tag{26}$$

where $Q_{d,Z} = Z^T Q_d Z$, $Q_d = Q + \mathrm{diag}(d)$, $q_{d,\hat{x}} = 2Q_d \hat{x} + q$, and $k_{d,\hat{x}} = \hat{x}^T Q_d \hat{x} + q^T \hat{x}$. For the minimization problem in (26) to be bounded below, we need to choose d such that

$Q_{d,Z} \succcurlyeq 0$, i.e. $d \in \mathcal{D}_\infty := \{d \in \mathbb{R}^n : Z^T(Q + \operatorname{diag}(d))Z \succcurlyeq 0\}$. This restriction on d implies that $X_z = x_z x_z^T$ holds at any optimal solution to the inner minimization problem. As a result, the dual problem in (26) can be simplified as:

$$\max_{d \in \mathcal{D}_\infty} \left\{ \begin{array}{ll} \min_{x_z, y} & (\hat{x} + Z x_z)^T Q_d (\hat{x} + Z z_x) + q^T (\hat{x} + Z x_z) - d^T y \\ \text{s.t.} & L_i \leq \hat{x}_i + e_i^T Z x_z \leq U_i, \ \forall i \in [n] \\ & l_i (\hat{x}_i + e_i^T Z x_z) \leq y_i \leq u_i (\hat{x}_i + e_i^T Z x_z), \ \forall i \in [n]. \end{array} \right\} \tag{27}$$

Since strong duality holds for (25) and its dual (27), both problems attain their optimal objective functions values. This implies that $\exists d^* \in \mathcal{D}_\infty$ such that:

$$\mu_{\text{DSDPdaz}} = \min_{x_z, y, v} \ v + q^T (\hat{x} + Z x_z)$$
$$\left. \begin{array}{rl} \text{s.t.} & v \geq (\hat{x} + Z x_z)^T (Q + \operatorname{diag}(d^*)) (\hat{x} + Z x_z) - d^{*T} y \\ & L_i \leq \hat{x}_i + e_i^T Z x_z \leq U_i, \ \forall i \in [n] \\ & l_i (\hat{x}_i + e_i^T Z x_z) \leq y_i \leq u_i (\hat{x}_i + e_i^T Z x_z), \ \forall i \in [n]. \end{array} \right\} \tag{28}$$

It is easy to show that (28) is a relaxation of (20). To this end, note that by replacing x and $\mathcal{D}_\infty$ in (20) with $\hat{x} + Z x_z$ and $\{d^*\}$, respectively, we obtain (28). Hence, $\mu_{\text{SICPda}\infty} \geq \mu_{\text{DSDPdaz}} = \mu_{\text{SDPdaz}} = \mu_{\text{SDPda}}$. Combining this condition with the result of Proposition 3, we obtain $\mu_{\text{SICPda}\infty} = \mu_{\text{DSDPda}}$, which completes the proof. □

The key assumption in Theorem 1 is the existence of a strictly feasible solution for the SDP (25). In the next proposition, we provide a set of sufficient conditions under which a strictly feasible solution can be constructed for (25).

Proposition 5. *Define* $\mathcal{X} := \{x \in \mathbb{R}^n : Ax = b, \ L_i < x_i < U_i, \ \forall i \in [n]\}$. *Assume that* $\mathcal{X}$ *is nonempty and that the following conditions hold:*

(i) $e_i^T Z \neq 0, \ \forall i \in [n]$.

(ii) $\exists \widetilde{x} \in \mathcal{X}$ *such that* $l_i(\widetilde{x}_i) = \widetilde{x}_i^2$ *and* $l_i(\widetilde{x}_i) < u_i(\widetilde{x}_i), \ \forall i \in [n]$.

Then, the SDP (25) *admits a strictly feasible solution.*

Proof. Define the scalar δ as:

$$\delta := \min_{i \in [n]} \frac{u_i(\widetilde{x}_i) - l_i(\widetilde{x}_i)}{e_i^T Z Z^T e_i}. \tag{29}$$

From assumption (i), it follows that $e_i^T Z Z^T e_i > 0, \ \forall i \in [n]$. Moreover, assumption (ii) implies that $u_i(\widetilde{x}_i) - l_i(\widetilde{x}_i) > 0, \ \forall i \in [n]$. Hence, $0 < \delta < \infty$. Let $\hat{x} = \widetilde{x}$, $\bar{x}_z = 0$, $\bar{y}_i = \widetilde{x}_i^2 + \epsilon e_i^T Z Z^T e_i$, $i \in [n]$, and $\bar{X}_z = \epsilon I_{n-m}$, where $\widetilde{x}$ satisfies the condition in (ii), and $0 < \epsilon < \delta$. It is simple to check that this choice is feasible in the SDP (25), and further, the inequalities (25b), (25d) and (25e) are satisfied strictly. This completes the proof. □

If for a given index i assumption (i) in Proposition 5 fails to hold, then x_i must be a fixed variable in the original problem. To illustrate this, note that any $x \in \mathbb{R}^n$ satisfying $Ax = b$ can be written as $x = \hat{x} + Z x_z$, where $\hat{x}$ satisfies $A\hat{x} = b$ and $x_z \in \mathbb{R}^{n-m}$. The condition $e_i^T Z = 0$ means that $e_i^T Z x_z = 0$, which in turn implies that $x_i = \hat{x}_i$. Hence, for

9

all the indices i for which $e_i^T Z = 0$, the corresponding variables can be eliminated from the original problem in order to obtain a reduced problem for which assumption (i) holds.

The satisfaction of assumption (ii) in Proposition 5 depends on the form of the functions $l_i(x_i)$ and $u_i(x_i)$. It is easy to show that $u_i(x_i)$ is an affine function given by $u_i(x) = (L_i + U_i)x_i - L_i U_i$. On the other hand, the form of $l_i(x_i)$ depends on the structure of the set S_i. If S_i is given by a closed interval, then $l_i(x_i) = x_i^2$, and in this case assumption (ii) is satisfied provided that there exists a vector $x \in \mathbb{R}^n$ such that $Ax = b$ and $L_i < x_i < U_i$, $\forall i \in [n]$. If S_i is given by a set of two discrete points, then $l_i(x_i) = u_i(x_i)$, $\forall x_i \in [L_i, U_i]$, and assumption (ii) fails to hold. This occurs, for example, when x_i is binary, i.e., $S_i = \{0, 1\}$, because in this case $l_i(x_i) = u_i(x_i) = x_i$, $\forall x_i \in [0, 1]$.

2.3 Further insights into the semidefinite relaxation

The SDP relaxation (15) can be derived from (14) by adding the valid equality $(Ax - b)^T(Ax - b) = 0$ and lifting it into the space of (x, X). Clearly, we can construct other SDP relaxations for (1) by including other classes of constraints derived from $Ax = b$. In general, we can apply the following procedure:

(R1) identify a (possibly empty) set $\mathcal{J}$ of quadratic functions of the form $f_j(x) = x^T C_j x + c_j^T x + \gamma_j$, where $C_j \in \mathbb{S}^n, c_j \in \mathbb{R}^n, \gamma_j \in \mathbb{R}$, such that $f_j(x) = 0$ for $x \in \Omega := \{x \in \mathbb{R}^n \mid Ax = b\}$;

(R2) construct an SDP relaxation for (1) as

$$\min_{(x,y) \in \mathcal{F}, X} \quad \langle Q, X \rangle + q^T x \tag{30a}$$

$$\text{s.t.} \quad X - xx^T \succcurlyeq 0 \tag{30b}$$

$$X_{ii} = y_i, \ \forall i \in [n] \tag{30c}$$

$$\langle C_j, X \rangle + c_j^T x + \gamma_j = 0, \ \forall j \in \mathcal{J} \tag{30d}$$

where the constraints (30d) are obtained by lifting the valid equalities $x^T C_j x + c_j^T x + \gamma_j = 0$, $\forall j \in \mathcal{J}$ into the space of (x, X).

There are different types of functions $f_j(x)$ that satisfy the condition in (R1). Some examples that have been considered in the literature are [6]: $(x_j(A_i.x - b_i))$, and $((A_j.x - b_j)(A_i.x - b_i))$, $\left(x^T A_{j.}^T A_i.x - b_j b_i \right)$. A natural question in this context is whether we can improve on the bound given by (15) when restricted to the class of relaxations in (30). We address this question in the remainder of this section by showing that (15) is the best relaxation among the class of relaxations in (30).

We start by recalling the properties of the functions satisfying (R1). The following result follows from Theorem 1 in [6].

Proposition 6. *Let $f(x) = x^T C x + c^T x + \gamma$ be a quadratic function. Then, $f(x) = 0$ for all $x \in \Omega := \{x \in \mathbb{R}^n \mid Ax = b\}$ if and only if $C = A^T W^T + WA$, $c = A^T \beta - 2Wb$, and $\gamma = -b^T \beta$ for some $W \in \mathbb{R}^{n \times m}$ and $\beta \in \mathbb{R}^m$.*

Based on Proposition 6, we can assume without loss of generality that $C_j = A^T W_j^T + W_j A$ for some $W_j \in \mathbb{R}^{n \times m}$. In the next proposition, we show that the feasible set of (15) is a subset of the feasible set of (30). Moreover, we provide conditions on the choice of quadratic functions in $\mathcal{J}$ for these two feasible sets to be identical.

10

Proposition 7. *Let $\mathcal{F}_{\text{SDPda}}$ and $\mathcal{F}_{\text{SDPdaJ}}$ denote the feasible regions of the SDPs in (15) and (30), respectively. Then, the following holds:*

(i) $\mathcal{F}_{\text{SDPda}} \subseteq \mathcal{F}_{\text{SDPdaJ}}$.

(ii) If $\exists \omega_j, j \in \mathcal{J}$ such that $\sum_{j \in \mathcal{J}} \omega_j W_j = A^T$ then $\mathcal{F}_{\text{SDPda}} = \mathcal{F}_{\text{SDPdaJ}}$.

Proof. We first prove (i). From Proposition 4, any $(\bar{x}, \bar{X}) \in \mathcal{F}_{\text{SDPda}}$ satisfies $\bar{X} = \bar{x}\bar{x}^T + Z\bar{W}_z Z^T$, where $\bar{W}_z \in \mathbb{S}^{n-m}$ and $\bar{W}_z \succeq 0$. Then, for any $(\bar{x}, \bar{y}, \bar{X}) \in \mathcal{F}_{\text{SDPda}}$ and $\forall j \in \mathcal{J}$ we have:

$$\langle C_j, \bar{X} \rangle + c_j^T \bar{x} + \gamma_j \tag{31a}$$

$$= \langle C_j, \bar{X} - \bar{x}\bar{x}^T \rangle + \bar{x}^T C_j \bar{x} + c_j^T \bar{x} + \gamma_j \tag{31b}$$

$$= \langle C_j, \bar{X} - \bar{x}\bar{x}^T \rangle = \langle A^T W_j^T + W_j A, Z\bar{W}_z Z^T \rangle = 0 \tag{31c}$$

where (31b) follows from adding and subtracting $\bar{x}^T C_j \bar{x}$, the first equality in (31c) follows from (R1), the second equality in (31c) from Proposition 6 and the final equality from the fact that Z is a basis for the nullspace of A. Thus $(\bar{x}, \bar{y}, \bar{X}) \in \mathcal{F}_{\text{SDPdaJ}}$ proving the claim in (i).

Now, we prove the claim in (ii). Assume that there exist $\omega_j, j \in \mathcal{J}$ such that the condition in (ii) holds. By performing a linear combination of the inequalities in (30d) using ω_j, we obtain that for any $(\bar{x}, \bar{y}, \bar{X}) \in \mathcal{F}_{\text{SDPdaJ}}$:

$$0 = \sum_{j \in \mathcal{J}} \omega_j \left(\langle C_j, \bar{X} \rangle + c_j^T \bar{x} + \gamma_j \right) \tag{32a}$$

$$= \sum_{j \in \mathcal{J}} \omega_j \left(\langle C_j, \bar{X} - \bar{x}\bar{x}^T \rangle + \bar{x}^T C_j \bar{x} + c_j^T \bar{x} + \gamma_j \right) \tag{32b}$$

$$= \sum_{j \in \mathcal{J}} \omega_j \langle C_j, \bar{X} - \bar{x}\bar{x}^T \rangle = 2 \langle A^T A, \bar{X} - \bar{x}\bar{x}^T \rangle \tag{32c}$$

where (32b) follows from adding and subtracting $\bar{x}^T C_j \bar{x}$, the first equality in (32c) follows from (R1), the second equality in (32c) from Proposition 6 and the condition in (ii). Thus $(\bar{x}, \bar{y}, \bar{X}) \in \mathcal{F}_{\text{SDPda}}$ proving the claim in (ii). $\qquad\square$

Now, we are ready to present the main result of this section.

Theorem 2. *Suppose that $\mathcal{J}$ is chosen such that (R1) holds. Let μ_{SDPda}, $\mu_{\text{SICPda}\infty}$, and μ_{SDPdaJ} be the optimal objective function values in (15), (20), and (30), respectively. Then, the following hold:*

(i) $\mu_{\text{SDPdaJ}} \leq \mu_{\text{SDPda}}$

(ii) If the assumption in Theorem 1 holds, then $\mu_{\text{SDPdaJ}} \leq \mu_{\text{SICPda}\infty}$.

Proof. The claim in (i) follows from Proposition 7. If the assumption in Theorem 1 holds, then $\mu_{\text{SDPda}} = \mu_{\text{SICPda}\infty}$ and the claim in (ii) follows from (i). $\qquad\square$

11

2.4   Cutting Surface Algorithm

By replacing $\mathcal{D}_\alpha$ in (6b) with a set $\mathcal{D}_\alpha^{(k)}$ of finite dimension, we can devise an iterative cutting surface algorithm which allows us to derive convex QCP relaxations for (1). At the k-th iteration of this algorithm, the following relaxation is solved:

$$\min_{(x,y)\in\mathcal{F},v}\quad v + q^T x \tag{33a}$$

$$\text{s.t.}\quad v \geq x^T \left(Q + \text{diag}(d)\right) x - d^T y, \ \forall d \in \mathcal{D}_\alpha^{(k)}. \tag{33b}$$

This iterative approach is described in Algorithm 1. At each iteration of this algorithm, a separation problem is solved in order to construct a new quadratic cut of the form (33b). As the number of quadratic cuts increases, the resulting QCP relaxations become tighter. The algorithm terminates when either the maximum number of iterations MaxNC is reached or the new quadratic cut derived at iteration k does not violate the solution of the QCP relaxation constructed at iteration $k-1$.

Note that the parameter α is fixed during the execution of this algorithm. Proposition 2 suggests that we should select a large value of α in order to improve the bound given by (33). We describe a procedure to determine such a value of α in §5.

Algorithm 1 A cutting surface procedure to derive QCP relaxations for (1)

Input: Q, q, A, b, and algebraic expressions for $l(\cdot)$ and $u(\cdot)$.
Output: A lower bound μ_{QCPda} on the optimal solution of (1).
If $m = 0$ **then**
  Set $\alpha = 0$
Else
  Choose a positive value of α according to the procedure described in §5.
End If
Set $\mathcal{D}_\alpha^{(0)} = \{d^{(0)}\}$, where $d^{(0)} \in \mathbb{R}^n$ is a perturbation for which (33) is convex.
Solve (33). Let $(\bar{x}, \bar{y}, \bar{v})$ be an optimal solution to this relaxation.
Set $\mu_{\text{QCPda}} = \bar{v} + q^T \bar{x}$.
For $k = 1$ to MaxNC **do**
  Solve a separation problem to find a new perturbation $d^{(k)}$.
  If the quadratic cut (33b) with $d = d^{(k)}$ violates $(\bar{x}, \bar{y}, \bar{v})$ **then**
    $\mathcal{D}_\alpha^{(k)} \leftarrow \mathcal{D}_\alpha^{(k-1)} \cup \{d^{(k)}\}$
    Solve (33). Let $(\bar{x}, \bar{y}, \bar{v})$ be an optimal solution to this relaxation.
    Set $\mu_{\text{QCPda}} = \bar{v} + q^T \bar{x}$.
  Else
    Terminate
  End If
End For

Another interesting observation about Algorithm 1 is that the parameter α only appears in the separation problem (see §3 for details), since the term $\alpha\|Ax - b\|^2$ is not included in (33b). This is particularly advantageous because it allows us to preserve the sparsity pattern of Q in the quadratic constraints (33b). In addition, by dropping the term $\alpha\|Ax - b\|^2$ from (33b), we prevent the relaxation from becoming ill-conditioned for large values of α.

In order to construct the first relaxation of Algorithm 1, we need to specify an initial perturbation $d^{(0)}$. For simplicity, we set $d^{(0)} = \mu\mathbb{1}$, where $\mu \in \mathbb{R}_{\geq 0}$. For this choice of $d^{(0)}$, (33) becomes:

$$\min_{(x,y)\in\mathcal{F},v} \quad v + q^T x \tag{34a}$$

$$\text{s.t. } v \geq x^T \left(Q + \mu I_n\right) x - \mu\mathbb{1}^T y. \tag{34b}$$

In order to select μ, we consider two cases depending on the value of m:

(i) If $m = 0$, (1) is an unconstrained optimization problem. We run Algorithm 1 only if Q is indefinite and set $\mu = -\lambda_{\min}(Q)$. It is easy to verify that this choice of μ renders $Q + \mu I_n$ positive semidefinite, thus ensuring the convexity of (34).

(ii) If $m > 0$, (1) contains at least one equality constraint. We run Algorithm 1 only if $Z^T Q Z$ is indefinite and set $\mu = -\lambda_{\min}(Z^T Q Z)$. It is simple to check that, for this choice of μ, (34) is a convex problem. To this end, note that the projection of (34) onto the nullspace of A can be obtained from (7) by considering a single quadratic constraint in (7b) and setting $d = \mu\mathbb{1}$. The resulting problem is convex when $Z^T \left(Q + \mu I_n\right) Z \succeq 0$. It is easy to verify that our choice of μ satisfies this condition.

Since (34) contains a single quadratic constraint and $\mu \geq 0$, we can eliminate the variables y and v, and rewrite this QCP as the following quadratic program:

$$\min_{x\in\mathcal{X}} \quad x^T \left(Q + \mu I_n\right) x + q^T x - \mu \sum_{i=1}^{n} u_i(x_i). \tag{35}$$

We refer to the relaxations obtained by setting $\mu = -\lambda_{\min}(Q)$ and $\mu = -\lambda_{\min}(Z^T Q Z)$ in (35) as the *eigenvalue relaxation* and the *eigenvalue relaxation in the nullspace of A*, respectively (see Section 3 in [12] for details). In (ii), we could use the same initial perturbation as in (i). However, as shown in [12], the perturbation given in (ii) can lead to a tighter initial bound.

Observe that the spectral relaxations introduced in [12] are derived through a uniform diagonal perturbation of Q, i.e., by considering a perturbation vector $d \in \mathbb{R}^n$ whose entries are identical to each other. By contrast, the QCP relaxations of Algorithm 1 are derived through a nonuniform diagonal perturbation of Q, i.e., by considering a perturbation vector $d \in \mathbb{R}^n$ whose entries are allowed to differ from each other. Note also that, since the bound provided by (34) is equal to the bound given by (35), by construction, the QCP relaxations solved during the execution in Algorithm 1 are at least as tight at the spectral relaxations considered in [12].

3 Analysis and regularization of the separation problem

We start this section by presenting the separation problem solved in Algorithm 1. Let $(\bar{x}, \bar{y}, \bar{v})$ be an optimal solution to the relaxation (33). Then, in order to construct a quadratic inequality of the form (33b) that is maximally violated by $(\bar{x}, \bar{y}, \bar{v})$, we can solve the following optimization problem:

$$\sup_{d\in\mathcal{D}_\alpha} \quad \sum_{i=1}^{n} \left(\bar{x}_i^2 - \bar{y}_i\right) d_i. \tag{36}$$

This SDP is parametrized by the value of α determined at the beginning of Algorithm 1. Observe also that the separation problem considered in [5] is a particular instance of (36), obtained by setting $\alpha = 0$ in (36). For the remainder of this section, we will cast (36) as:

$$\inf_{d \in \mathcal{D}_\alpha} \eta^T d \tag{37}$$

where $\eta_i := \bar{y}_i - \bar{x}_i^2$, $\forall i \in [n]$. As shown in [5], the attainment of the infimum in (37) is not guaranteed, and may depend on the problem data. We illustrate this behavior through the following example.

Example 1. *Let $Q = \begin{bmatrix} 0 & 2 \\ 2 & -1 \end{bmatrix}$, $A = \begin{bmatrix} 0 & 1 \end{bmatrix}$, and $\alpha = 1$ in (37). Consider the following cases:*

(i) *$\bar{x} = \bar{y} = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}^T$. In this case, the infimum in (37) is attained for $d_1^* = d_2^* = 2$.*

(ii) *$\bar{x} = \bar{y} = \begin{bmatrix} 0.4 & 0 \end{bmatrix}^T$. In this case, the infimum in (37) cannot be attained since it occurs as $d_1 \to 0$ and $d_2 \to \infty$.*

To further analyze the attainment of (37), we construct the dual of this SDP. Let $Y \in \mathbb{S}^n$ be the matrix of dual variables associated with the semidefinite constraint $Q + \text{diag}(d) + \alpha A^T A \succcurlyeq 0$. Then, the dual of (37) is:

$$\sup_{Y \succcurlyeq 0} - \langle Q + \alpha A^T A, Y \rangle \tag{38a}$$

$$\text{s.t. } Y_{ii} = \eta_i, \ \forall i \in [n]. \tag{38b}$$

Let $P := \{i \in [n] : \bar{y}_i = \bar{x}_i^2\}$. From (38b), it follows that $Y_{ii} = 0$, $\forall i \in P$. Hence, if $P \neq \emptyset$, (38) does not admit a strictly feasible solution and, as a result, strong duality may not hold for the primal-dual pair (37),(38). In Example 1, we have $P = \emptyset$ for (i), and $P = \{2\}$ for (ii).

In the separation step of Algorithm 1, we do not need to solve (37) to optimality, but rather derive quadratic cuts that can be used to tighten a relaxation of the form (33). As a result, we can replace (37) with a regularized separation problem constructed in a way such that its optimum is always attained. One option is to regularize (37) as discussed in [5]. To this end, we can add to the objective function of (37) the term $\lambda \sum_{i=1}^n [d_i]_+$, where $\lambda = \sum_{i=1}^n (\bar{y}_i - \bar{x}_i^2)$, and $[d_i]_+$ is equal to d_i if $d_i > 0$, and 0 otherwise. This leads to the following regularized separation problem:

$$\inf_{d \in \mathcal{D}_\alpha} \eta^T d + \lambda \sum_{i=1}^n [d_i]_+. \tag{39}$$

It is simple to show that the infimum in this SDP is always attained (see Proposition 3 in [5] for details). The parameter λ is always positive unless the current relaxation is exact.

In this paper, we propose an alternative regularization for (37). We modify this problem by adding to the objective function the quadratic term $\rho d^T d$, where ρ is a positive scalar. The resulting regularized separation problem is:

$$\inf_{d \in \mathcal{D}_\alpha} \eta^T d + \rho d^T d. \tag{40}$$

As we show in the following proposition, this regularization also gives rise to a separation problem for which the optimum is always attained.

Proposition 8. *Let $\rho > 0$ in (40). Then, the optimal solution to the semidefinite program (40) is always attained at some finite point.*

Proof. This proof relies on strong duality holding for (40) and its dual. Denote by $Y \in \mathbb{S}^n$ the matrix of dual variables associated with the semidefinite constraint $Q + \text{diag}(d) + \alpha A^T A \succcurlyeq 0$. Then, the dual of (40) is:

$$\sup_{Y \succcurlyeq 0} -\langle Q + \alpha A^T A, Y \rangle - \frac{1}{4\rho} \sum_{i=1}^{n} (Y_{ii} - \eta_i)^2. \tag{41}$$

Now, let $\bar{d} = \mu \mathbb{1}$ and $\bar{Y} = I_n$, where $\mu > -\min(0, \lambda_{\min}(Q + \alpha A^T A))$. Clearly, $\bar{d}$ and $\bar{Y}$ are strictly feasible in (40) and (41), respectively. Hence, strong duality holds, and both SDPs attain their optimal solutions. $\square$

4   Solution of the regularized separation problem

In this section, we describe the algorithm that we use to solve the regularized separation problem proposed in §3. This algorithm is a modification of the barrier coordinate minimization method introduced in [5]. To solve (40), our algorithm operates on the following penalized log-det problem:

$$\begin{aligned}
\inf_{d \in \mathbb{R}^n} \quad & f(d; \sigma) := G(d) - \sigma \text{log-det} \left(Q + \text{diag}(d) + \alpha A^T A \right) \\
\text{s.t.} \quad & Q + \text{diag}(d) + \alpha A^T A \succ 0
\end{aligned} \tag{42}$$

where $G(d) = \sum_{i=1}^{n} g_i(d_i)$, $g_i(d_i) = \eta_i d_i + \rho d_i^2$, $\forall i \in [n]$, and σ is a positive penalty parameter. The optimality condition for (42) can be expressed as:

$$\nabla f(d; \sigma) = 0, \quad Q + \text{diag}(d) + \alpha A^T A \succ 0 \tag{43}$$

where the gradient of $f(d; \sigma)$ has the form:

$$\nabla f(d; \sigma) = \nabla G(d) - \sigma \text{diag} \left(\left[Q + \text{diag}(d) + \alpha A^T A \right]^{-1} \right) \tag{44}$$

with $\nabla G(d)_i = \eta_i + 2\rho d_i$, $\forall i \in [n]$. At each iteration of this algorithm, we update a feasible vector $\bar{d}$ and an inverse matrix $V := \left[Q + \text{diag}(\bar{d}) + \alpha A^T A \right]^{-1}$. Based on the optimality condition in (43), we perform coordinate minimization by choosing an index i which corresponds to the entry of $\nabla f(\bar{d}; \sigma)$ with the largest magnitude:

$$i = \arg \max_{j=1,\dots,n} \left\{ \left| \nabla f(\bar{d}; \sigma)_j \right| \right\}. \tag{45}$$

This choice of i leads to the following one-dimensional minimization problem:

$$\Delta d_i^* \in \arg\min_{\Delta d_i} \left\{ f(\bar{d} + \Delta d_i e_i; \sigma) \ : \ Q + \text{diag}(\bar{d} + \Delta d_i e_i) + \alpha A^T A \succ 0 \right\}. \tag{46}$$

As we show in the next proposition, it is possible to find a closed-form expression for the optimal solution of (46).

Proposition 9. *The optimal solution to the one-dimensional problem (46) is:*

$$\Delta d_i^* = -(\phi_i + \tau_i) + \sqrt{(\phi_i - \tau_i)^2 + \kappa} \tag{47}$$

where $\phi_i = 1/(2V_{ii})$, $\tau_i = (\eta_i + 2\rho \bar{d}_i)/(4\rho)$ and $\kappa = \sigma/(2\rho)$.

15

Proof. At the optimal solution of (46) the following holds:

$$\frac{\partial f(\bar{d} + \Delta d_i e_i; \sigma)}{\partial \Delta d_i} = \eta_i + 2\rho(\bar{d}_i + \Delta d_i) - \frac{\sigma V_{ii}}{1 + \Delta d_i V_{ii}} = 0. \tag{48}$$

By solving for Δd_i in (48), we obtain the roots $\Delta d_i^{*(\pm)} = -(\phi_i + \tau_i) \pm \sqrt{(\phi_i - \tau_i)^2 + \kappa}$. It is easy to show that $\Delta d_i^{*(+)}$ is the only one of these two solutions that is feasible in (46). By applying Lemma 1 from [5], we have:

$$Q + \text{diag}(\bar{d} + \Delta d_i e_i) + \alpha A^T A \succ 0 \iff \Delta d_i > -1/V_{ii}. \tag{49}$$

Therefore, for $\Delta d_i^{*(+)}$ to be feasible in (46) we must have $z + \sqrt{z^2 + \kappa} > 0$, where $z = (\phi_i - \tau_i)$. It is simple to check that this condition is always satisfied. To this end, note that $\kappa > 0$. This implies $z + \sqrt{z^2 + \kappa} > z + |z| \geq 0$, $\forall z \in \mathbb{R}$. Using a similar analysis, we can show that $\Delta d_i^{*(-)}$ is infeasible in (46). $\square$

After solving the one-dimensional problem (46), we update $\bar{d}$ as:

$$\bar{d} \leftarrow \bar{d} + \Delta d_i^* e_i \tag{50}$$

and update V using the Sherman-Morrison formula:

$$V \leftarrow V - \frac{\Delta d_i^* V_{\cdot i} V_{\cdot i}^T}{1 + \Delta d_i^* V_{ii}}. \tag{51}$$

In our numerical experiments, we noticed that, for very small values of ρ, some of the entries of $\bar{d}$ become very large after performing the update in (50). This is not surprising because: (i) for very small values of ρ, (40) exhibits a similar behavior to (37), and (ii) as discussed in §3, the finite attainment of (37) is not guaranteed. To address this issue, we propose an adaptive strategy in order to adjust the value of ρ used in (40). At a given iteration of our algorithm, after performing the update in (50), we determine the entry of $\bar{d}$ with the largest magnitude, i.e.:

$$\bar{d}_{\max} = \max_{j=1,\ldots,n} \left\{ |\bar{d}_j| \right\}. \tag{52}$$

If $\bar{d}_{\max}$ is at least an order of magnitude larger than $\mu := -\lambda_{\min}(Q + \alpha A^T A)$, we increase ρ by multiplying it by a factor ρ_{upd}, and restart our algorithm with this new value of ρ. In practice, this adaptive strategy only requires a few restarts before finding a suitable value of ρ. For the first run, we set $\rho = \rho_{\text{init}}$.

Once (42) has been solved within a given precision, we update the penalty parameter σ according to the following condition:

$$\sigma \leftarrow \max\{\sigma_{\min}, \sigma_{\text{upd}}\sigma\} \quad \text{if} \quad \frac{\|\nabla f(\bar{d}; \sigma)\|_2}{\|\eta\|_2} \leq \epsilon_{\text{upd}} \tag{53}$$

where $\nabla f(\bar{d}; \sigma)$ is used as a measure of optimality. We check the relative improvement in the objective function of (40) every $\omega_{\text{check}} n$ iterations, and terminate our algorithm if this relative improvement is smaller than ϵ_{check}.

Our coordinate minimization strategy is summarized in Algorithm 2. If $Q + \alpha A^T A$ is positive semidefinite, then $Z^T Q Z$ is also positive semidefinite and (1) is convex when

restricted to the nullspace of A. In this case, it suffices to solve the continuous relaxation of (1) in order to obtain a lower bound. As a result, the separation procedure outlined in Algorithm 2 is only used if $Q + \alpha A^T A$ is indefinite. We start Algorithm 2 with an initial perturbation $\hat{d} = -1.5\lambda_{\min}(Q + \alpha A^T A)\mathbb{1}$. We set $\text{MaxIter} = 500n$, $\sigma_{\min} = 10^{-5}$, $\sigma_{\text{upd}} = 0.8$, $\epsilon_{\text{upd}} = 0.03$, $\omega_{\text{check}} = 10$ and $\epsilon_{\text{check}} = 10^{-4}$. We initialize ρ_{init} as:

$$\rho_{\text{init}} = 10^{-4} \frac{10^{4\lfloor \log_{10}(\delta_{\max}) \rfloor}}{\max\{1, \lfloor Q_{\max}/100 \rfloor Q_{\max}\}} \tag{54}$$

where $Q_{\max}$ and $\delta_{\max}$ are given by:

$$Q_{\max} = \max_{i=1,\ldots,n,\, j=i,\ldots,n} \{|Q_{ij}|\}, \quad \delta_{\max} = \max_{i=1,\ldots,n} \{U_i - L_i\} \tag{55}$$

and set $\rho_{\text{upd}} = 10$. The initial value of σ_{init} is determined as:

$$\sigma_{\text{init}} = \text{median} \left\{ \left| \frac{\eta_i + 2\rho\hat{d}_i}{V_{ii}} \right| \right\}_{i=1}^{n}. \tag{56}$$

Algorithm 2 Barrier coordinate minimization algorithm used to solve the smooth regularized separation problem (40)

1: **Input**: Q, A, α, and an optimal solution $(\bar{x}, \bar{y}, \bar{v})$ to (33).
2: **Output**: A vector $\bar{d}$ that solves (40).
3: Set $\rho = \rho_{\text{init}}$, where ρ_{init} is determined using (54).
4: Set $\bar{d} = \hat{d}$, where $\hat{d} = 1.5\mu\mathbb{1}$ and $\mu = -\lambda_{\min}(Q + \alpha A^T A)$.
5: Set $\sigma = \sigma_{\text{init}}$, where σ_{init} is calculated according to (56).
6: Calculate $V = \left[Q + \text{diag}(\bar{d}) + \alpha A^T A \right]^{-1}$ and set $k = 0$.
7: **While** $(k < \text{MaxIter})$ **do**
8:     Update $k \leftarrow k + 1$.
9:     Determine an index i according to (45) and calculate Δd_i^* using (47).
10:     Update $\bar{d}$ according to (50) and determine $\bar{d}_{\max}$ using (52).
11:     **If** $(\bar{d}_{\max} > 10\mu)$ **then**
12:         Update $\rho \leftarrow \rho_{\text{upd}}\rho$ and **goto** 4.
13:     **End If**
14:     Update V according to (51).
15:     Adjust σ according to (53).
16:     **If** $(k \bmod (\omega_{\text{check}}n) = 0)$ **then**
17:         Terminate if the improvement in the objective of (40) is smaller than ϵ_{check}.
18:     **End If**
19: **End While**

Even though Algorithm 2 is a variant of the barrier coordinate minimization algorithm introduced in [5], there are two key differences between these two algorithms. First, unlike the algorithm presented in [5], Algorithm 2 does not rely on nonsmooth optimization techniques because the objective function of our regularized separation problem (40) is smooth. Second, the regularization parameter λ used in (39) is fixed throughout the execution of the algorithm proposed in [5]. By contrast, in Algorithm 2, we adaptively adjust the regularization parameter ρ used in (40). As we demonstrate in §6.1, because of this adaptive strategy,

the quadratic cuts derived by solving (40) with our algorithm lead to significantly tighter relaxations than the quadratic cuts obtained through the solution of (39) with the algorithm proposed in [5]. For the sake of completeness, in §A, we provide a detailed description of the barrier coordinate minimization algorithm introduced in [5].

5 Implementation in a branch-and-bound algorithm

BARON's portfolio of relaxations consists of linear, quadratic, nonlinear, and mixed-integer linear programming relaxations [8, 10, 12, 21]. As part of our implementation, we have expanded this portfolio by adding the convex relaxations described in §2.4. These new relaxations are only used if the original model supplied to BARON is of the form (1).

At the root node of the branch-and-bound tree, we solve QCP relaxations of the form (33) by running Algorithm 1 with the maximum number of iterations MaxNC set to 20. In each iteration of this algorithm, we generate quadratic cuts of the form (33b) by solving the regularized separation problem (40) with Algorithm 2. As indicated in §2.4, for $m > 0$, the initial perturbation used in Algorithm 1 is set as $d^{(0)} = \mu \mathbb{1}$, where $\mu = -\lambda_{\min}(Z^T Q Z)$. In our previous paper [12], we showed that it is possible to approximate $\lambda_{\min}(Z^T Q Z)$ without having to explicit compute the basis Z. In particular, we proved that:

$$\lim_{\alpha \to \infty} \lambda_{\min}(Q, I_n + \alpha A^T A) = \min(0, \lambda_{\min}(Z^T Q Z)). \tag{57}$$

Using this result, we set $\mu = \mu(\alpha) := -\lambda_{\min}(Q, I_n + \alpha A^T A)$. From (57), it follows that, for a sufficiently large value of α, $\mu(\alpha)$ will converge to 0 if (1) is convex when restricted to the nullspace of A, or $-\lambda_{\min}(Z^T Q Z)$ otherwise. To find such value of α, we follow the iterative procedure presented in Section 5 of [12]. This is the same value of α that we use in Algorithm 1 and the regularized separation problem (40). Note that this value of α determined at the root-node is used throughout the entire branch-and-bound tree.

Algorithm 1 is only called at the root-node of the branch-and-bound tree. At nodes other than the root-node, instead of solving QCP relaxations, we solve QP relaxations of the form:

$$\min_{(x,y)\in\mathcal{F}} \quad x^T (Q + \operatorname{diag}(d)) x + q^T x - d^T y \tag{58}$$

where $d \in \mathcal{D}_\alpha$. We proceed as follows:

(i) We solve an initial relaxation of the form (58) by setting $d = d^{\text{parent}}$, where d^{parent} is a diagonal perturbation originating from the parent node. Let $(\bar{x}, \bar{y})$ be the optimal solution of this initial QP relaxation and denote by $\bar{\mu}_{\text{QP}}$ its optimal objective function value.

(ii) We use this relaxation solution to construct a new perturbation d^{new} by solving the regularized separation problem (40) with Algorithm 2.

(iii) If $\bar{\mu}_{\text{QP}} - q^T \bar{x} < \bar{x}^T (Q + \operatorname{diag}(d^{\text{new}})) \bar{x} - (d^{\text{new}})^T \bar{y}$, we solve a second relaxation of the form (58) by setting $d = d^{\text{new}}$. Let $(\hat{x}, \hat{y})$ be an optimal solution of this relaxation and denote by $\hat{\mu}_{\text{QP}}$ its optimal objective function value. If $\hat{\mu}_{\text{QP}} \geq \bar{\mu}_{\text{QP}}$ (resp. $\hat{\mu}_{\text{QP}} < \bar{\mu}_{\text{QP}}$), we use the bound $\hat{\mu}_{\text{QP}}$ (resp. $\bar{\mu}_{\text{QP}}$) and pass d^{new} (resp. d^{parent}) to the descendant nodes of the current node.

18

For the descendant nodes of the root-node, we set $d^{\text{parent}} = d^{\text{root}}$, with d^{root} being a surrogate perturbation vector determined as:

$$d^{\text{root}} = \frac{1}{\sum_{i=1}^{\text{NC}} \nu_i} \sum_{i=1}^{\text{NC}} \nu_i d^{(i)}, \tag{59}$$

where NC is the number of quadratic cuts generated during the execution of Algorithm 1 at the root-node, $d^{(i)}$ are the diagonal perturbations, and ν_i are the optimal Lagrange multipliers associated with the quadratic constraints of the last root-node relaxation of the form (33). Note that NC $\leq$ MaxNC, since Algorithm 1 might terminate before reaching the maximum number of iterations MaxNC.

The decision to solve QP relaxations instead of QCP relaxations at nodes other than the root node is motivated by two key observations. First, the convex QCP relaxations of the form (33) are at least an order of magnitude more expensive than the QP relaxations of the form (58). Second, often a single quadratic cut of the form (33b) leads to a significant bound improvement. As a result, there is little gain in running Algorithm 2 more than once. Since the first QP relaxation constructed at the descendant nodes always uses a diagonal perturbation originating from the parent node, the monotonicity of the bounds generated during the branch-and-bound search is guaranteed.

To solve the eigenvalue and generalized eigenvalue problems that arise during the construction of the relaxations discussed above, we use the subroutines included in the linear algebra library LAPACK [1]. At a given node of the branch-and-bound tree, we only consider the variables that have not been fixed in order to construct our relaxations. We solve the convex QCP relaxations with IPOPT and the convex QP relaxations with CPLEX. The relaxation solutions returned by these solvers are used at a given node only if they satisfy the KKT conditions. At nodes at which (1) is convex, we do not use the relaxations described in this section, and solve instead a continuous relaxation of (1) subject to the variable bounds of the current node.

When all the variables in (1) are binary, $l_i(x_i) = u_i(x_i) = x_i$, $\forall i \in [n]$, and as a result, we can eliminate the y variables from (33) and (58). For continuous and general integer variables, we use $l_i(x_i) = x_i^2$, $i \in [n]$. For general integer variables, this choice of $l_i(x_i)$ does not lead to the convex hull of $\mathcal{C}_i$, but it allows us to construct a convex outer-approximation for this set.

Our implementation relies on the dynamic relaxation selection strategy proposed in [12] in order to adjust the frequencies at which we solve polyhedral and quadratic relaxations during the branch-and-bound search. Moreover, if (1) is a binary quadratic program, we use the spectral braching variable selection rule introduced in [12]. The QP relaxations (58) are only used during the branch-and-bound search if, at the root-node, Algorithm 1 gives a tighter bound than BARON's LP relaxation. Otherwise, we disable these QP relaxations and utilize the spectral relaxations proposed in [12].

6 Computational results

In this section, we investigate the impact of the quadratic relaxations proposed in this paper on the performance of branch-and-bound algorithms. We start in §6.1, by showing the effectiveness of the regularization approach discussed in §3. Then, in §6.2, we demonstrate the benefits of the implementation described in §5 on the performance of BARON. Finally, in §6.3, we present a comparison between several state-of-the-art global optimization solvers.

Our experiments are conducted under GAMS 30.1.0 on a 64-bit Intel Xeon X5650 2.66GHz processor with a single-thread. For the experiments described in §6.1, we solve the QP relaxations with CPLEX 12.10, the QCP relaxations with IPOPT 3.12 and the SDP relaxations with MOSEK 9.1.9. For the experiments considered in §6.2–6.3, we consider the following global optimization solvers: ANTIGONE 1.1, BARON 20.4, COUENNE 0.5, CPLEX 12.10, GUROBI 9.0, LINDOGLOBAL 12.0 and SCIP 6.0. In this case we: (i) run all solvers with relative/absolute tolerances of 10^{-6} and a time limit of 500 seconds, and (ii) set the CPLEX option `optimalitytarget` to 3 and the GUROBI option `nonconvex` to 2 to ensure that these two solvers search for a globally optimal solution. We use default settings for other algorithmic parameters.

For our experiments, we consider a large test set consisting of 960 Cardinality Binary Quadratic Programs (CBQPs), 30 Quadratic Semi-Assignment Problems (QSAPs), 246 Box-Constrained Quadratic Programs (BoxQPs), and 315 Equality Integer Quadratic Programs (EIQPs). These test libraries are described in detail in [13].

6.1 Experiments with root-node relaxations

In this section, we provide a numerical comparison between two versions of Algorithm 1 which differ in the separation procedure used to derive the quadratic cuts of the form (33b). We use the following notation for the relaxations considered in this comparison:

(i) EIG: Eigenvalue relaxation, obtained by setting $\mu = -\lambda_{\min}(Q)$ in (35).

(ii) EIGNS: Eigenvalue relaxation in the nullspace of A, obtained by setting $\mu = -\lambda_{\min}(Z^T Q Z)$ in (35).

(iii) SDPd: SDP relaxation (14).

(iv) SDPda: SDP relaxation (15).

(v) QCPnsreg: QCP relaxation (33), where the quadratic cuts (33b) are obtained by solving (39) with the algorithm proposed in [5] (see Algorithm 3 in §A).

(vi) QCPsreg: QCP relaxation (33), where the quadratic cuts (33b) are obtained by solving (40) with Algorithm 2.

In our experiments, we run the two versions of Algorithm 1 by setting the maximum number of iterations MaxNC to 20. We first compare these relaxations by selecting one instance from each of the four libraries that are part of the test set. The results of this comparison are presented in Figures 1a–1d. In these figures, we plot the lower bounds of the QCP relaxations against the number of iterations, and use a dashed vertical line to indicate the iteration number at which each version of Algorithm 1 terminates. We use horizontal lines to represent the lower bounds provided by the spectral and SDP relaxations. As seen in the figures, the quadratic cuts derived by solving (40) with Algorithm 2 lead to significantly tighter QCP relaxations than the quadratic cuts obtained through the solution of (39) with the algorithm proposed in [5]. Under our approach, a few quadratic cuts are sufficient in order to obtain a good approximation of the lower bounds given by the SDP relaxations.

Now, we compare the two versions of Algorithm 1 by considering all the instances contained in each of the test libraries. To this end, we construct performance profiles based

(a) Selected CBQP instance.   (b) Selected QSAP instance.

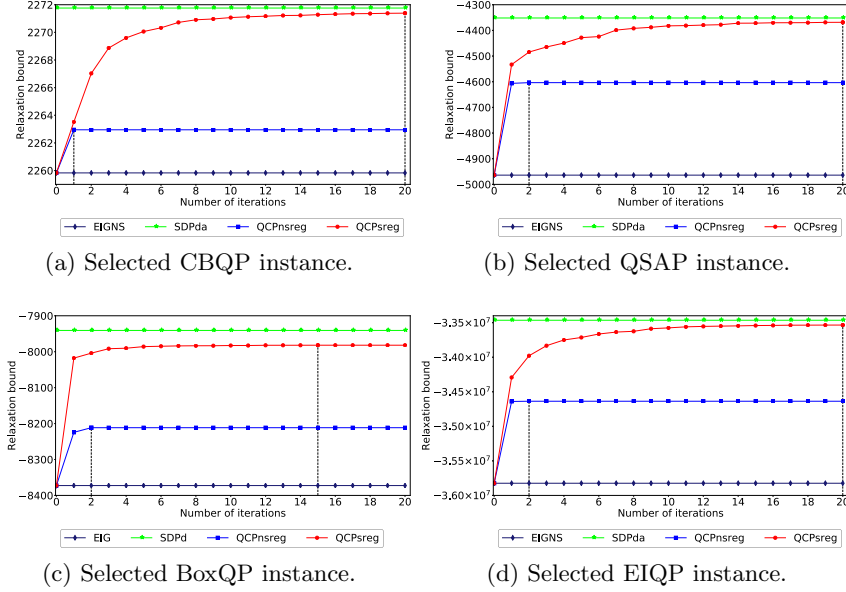(c) Selected BoxQP instance.   (d) Selected EIQP instance.

Figure 1: Comparison between the two versions of the cutting surface algorithm for selected test problems.

on the following root-node relaxation gap:

$$\text{Root gap} = \left(\frac{\mu_{\text{SDP}} - \mu_{\text{QCP}}}{\mu_{\text{SDP}} - \mu_{\text{QP}}}\right) \times 100 \tag{60}$$

where μ_{QCP} is the lower bound given by the last QCP relaxation solved in a given version of Algorithm 1, and μ_{QP} and μ_{SDP} denote the lower bounds provided by the corresponding spectral and SDP relaxations. A smaller gap represents a better approximation of the corresponding SDP bound.

The performance profiles are presented in Figures 2a–2d. These profiles show the percentage of models for which the gap defined in (60) is below a certain threshold. Clearly, the QCP relaxations constructed via our separation procedure provide significantly smaller gaps than the QCP relaxations derived with the separation algorithm proposed in [5].

6.2   Impact of the implementation on BARON's performance

In this section, we demonstrate the benefits the relaxations introduced in this paper on the performance of the global optimization solver BARON. In our experiments, we compare the following versions of BARON 20.4:

 (i) BARONnoqc: Version of BARON for which we disable the quadratic relaxations proposed in this paper. Note that this version of BARON includes the spectral relaxations introduced in [12].

(ii) BARON: Version of BARON which uses the quadratic relaxations proposed in this paper as described in §5.

In this comparison, we exclude from the test set all problems for which the new quadratic relaxations are not activated by BARON during the branch-and-bound search (367 instances). We also eliminate problems that can be solved trivially by both solvers (62

21

(a) 960 CBQP instances.

(b) 30 QSAP instances.
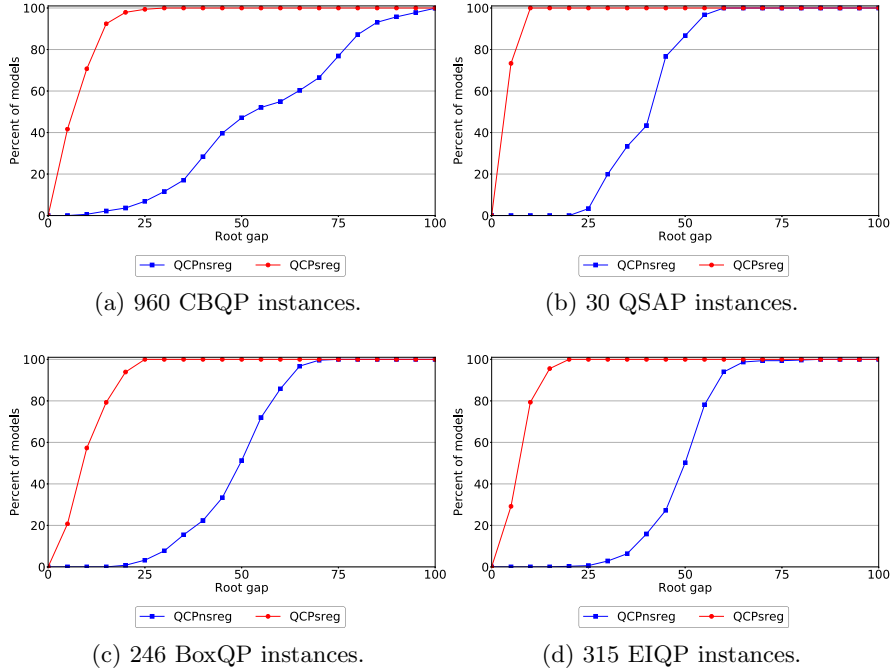
(c) 246 BoxQP instances.

(d) 315 EIQP instances.

Figure 2: Comparison between the two versions of the cutting surface algorithm for all test problems.

instances). A problem is regarded as trivial if it can be solved by both solvers in less than one second. After eliminating all of these problems from the original test set, we obtain a new test set consisting of 1122 instances.

We first consider the nontrivial problems that are solved to global optimality by at least one of the two the versions of the solver (259 instances). For this analysis, we compare the performance of the two solvers by considering the following metrics: (i) CPU time, (ii) total number of nodes in the branch-and-bound tree (iterations), and (iii) maximum number of nodes stored in memory (memory). In this comparison, we say that the two solvers perform similarly if any of these metrics are within 10% of each other. The results are presented in Figures 3a–3c. As the figures indicate, for nearly 90% of the problems considered in this comparison, our implementation leads to a significant reduction in CPU time, number of iterations, and memory requirements. There are a few instances for which BARONnoqc performs slightly better than BARON. These are relatively easy instances which can be solved to global optimality by both solvers in less than 10 seconds. For these instances, the relaxations proposed in this paper provide tighter lower bounds, but the increased computational cost associated with the construction and solution of these relaxations leads to some degradation in performance.

Now, we consider the nontrivial problems that neither of the two solvers are able to solve to global optimality within the time limit (863 instances). In this case, we analyze the performance of these solvers by comparing the relative gaps reported at termination, which are determined as:

$$\text{Relative gap} = \left(\frac{\mu_{\text{UBD}} - \mu_{\text{LBD}}}{\max(|\mu_{\text{LBD}}|, 10^{-3})} \right) \times 100 \tag{61}$$

where μ_{LBD} and μ_{UBD} respectively denote the lower and upper bounds reported by a given

22

solver at termination. In this comparison, we say that two solvers obtain similar gaps if their relative gaps are within 10% of each other. The results are presented in Figure 3d. As seen in the figure, for more than 90% of the problems considered in this comparison, BARON reports significantly smaller gaps than BARONnoqc.
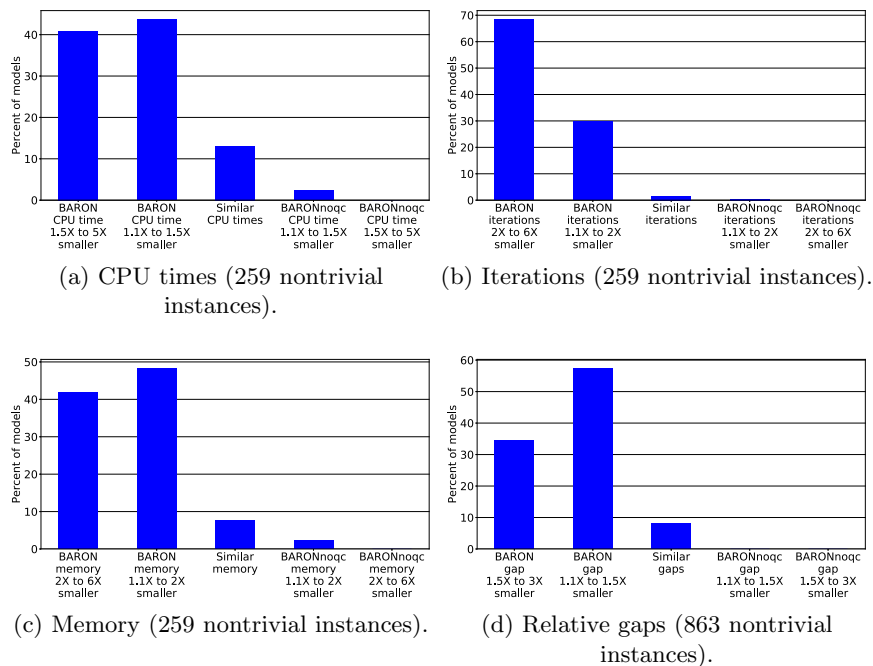

(a) CPU times (259 nontrivial instances).

(b) Iterations (259 nontrivial instances).

(c) Memory (259 nontrivial instances).

(d) Relative gaps (863 nontrivial instances).

Figure 3: One-to-one comparison between BARON and BARONnoqc.

We finish this section by providing a more detailed analysis of the results presented in Figures 3a–3d. To this end, we calculate the shifted geometric means for each of the metrics considered in these figures. We use a shift factor of 1 for the CPU times and relative gaps, and a shift factor of 10 for the total number of nodes and maximum number of nodes stored in memory. The results are presented in Table 1. As seen in the table, BARON significantly outperforms BARONnoqc for each of the considered metrics.

Table 1: Shifted geometric means for BARON and BARONnoqc.

Solver	CPU Time (259 instances)	Iterations (259 instances)	Memory (259 instances)	Relative gaps (863 instances)
BARONnoqc	14.0	926.1	25.9	11.8
BARON	9.9	391.7	13.5	8.7
Improvement (%)	29.6	57.7	47.7	25.7

6.3 Comparison between global optimization solvers

We start this section by comparing several state-of-the-art global optimization solvers through performance profiles. For instances for which a solver can prove global optimality within the time limit, we plot the percentage of models solved within a certain amount

of time. For problems for which a solver cannot prove global optimality within the time limit, we plot the percentage of models for which the relative gap defined in (61) is below a given threshold. These profiles are shown in Figures 4a–4d. As seen in these figures, BARON performs well relative to the other solvers. For the CBQP and QSAP instances, BARON is faster than the other solvers and solves many more problems to global optimality. For problems for which global optimality cannot be proven within the time limit, BARON terminates with smaller gaps than the other solvers.

(a) 960 CBQP instances.

(b) 30 QSAP instances.

(c) 246 BoxQP instances.
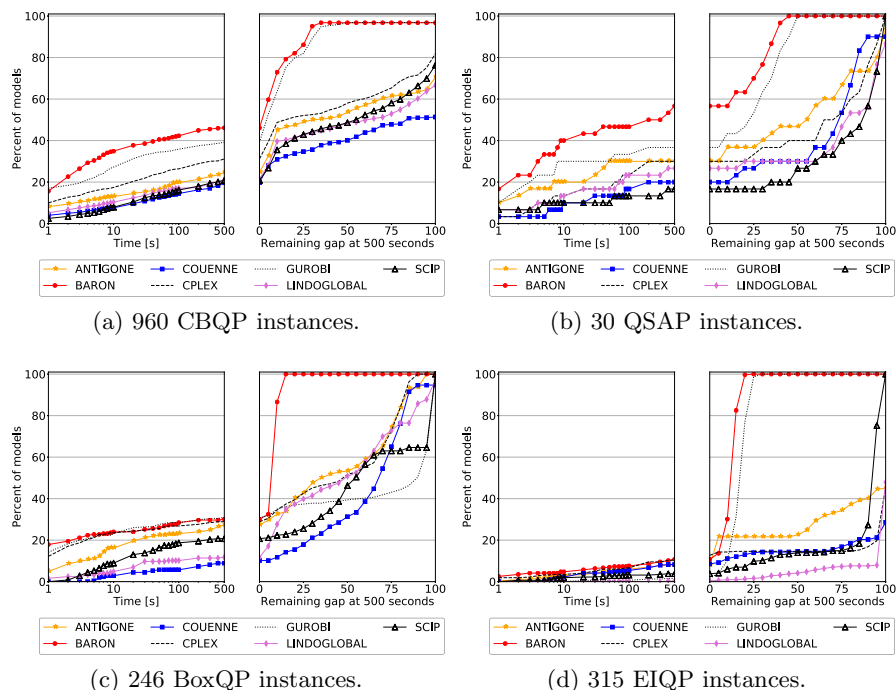
(d) 315 EIQP instances.

Figure 4: Comparison between global optimization solvers.

Next, we provide a more detailed analysis involving BARON, CPLEX and GUROBI. We use the same type of bar plots employed in §6.2. For a one-to-one comparison between BARON and CPLEX, we eliminate from the test set all problems solved trivially by both solvers (124 instances), obtaining a new test set with 1427 instances. In Figure 5a, we consider the nontrivial problems solved to global optimality by at least one of the two solvers (453 instances), whereas in Figure 5b, we consider nontrivial problems that neither solver can solve to global optimality within the time limit (974 instances). As both figures indicate, BARON performs significantly better than CPLEX. For nearly 90% of the instances considered in Figure 5a, BARON is at least 1.1 times faster than CPLEX, whereas for more than 98% of the instances considered in Figure 5b, BARON reports significantly smaller gaps than CPLEX.

Finally, we present a one-to-one comparison between BARON and GUROBI. Once again, we eliminate from the test set all the problems that can be solved trivially by both solvers (183 instances), which leads to a new test set with 1368 instances. In Figure 6a, we consider the nontrivial problems that are solved to global optimality by at least one of the two solvers (391 instances), whereas in Figure 6b, we consider nontrivial problems that neither of the two solvers are able to solve to global optimality within the time limit (977 instances). For more than 80% of the instances considered in Figure 6a, BARON is at least

24

(a) CPU times (453 nontrivial instances).

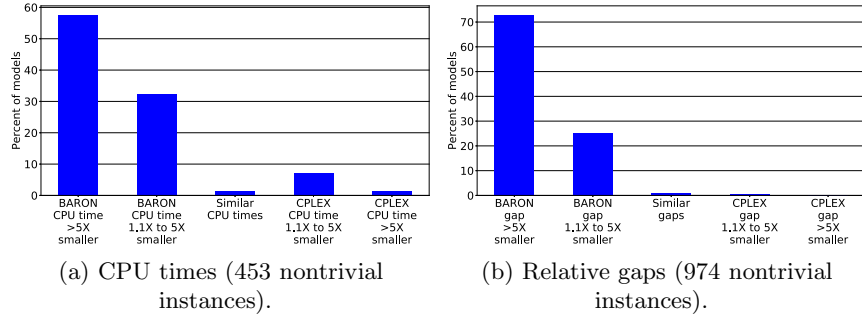(b) Relative gaps (974 nontrivial instances).

Figure 5: One-to-one comparison between BARON and CPLEX.

1.1 faster than GUROBI, whereas for nearly 90% of the instances considered in Figure 6b, BARON terminates with considerably smaller gaps than GUROBI.
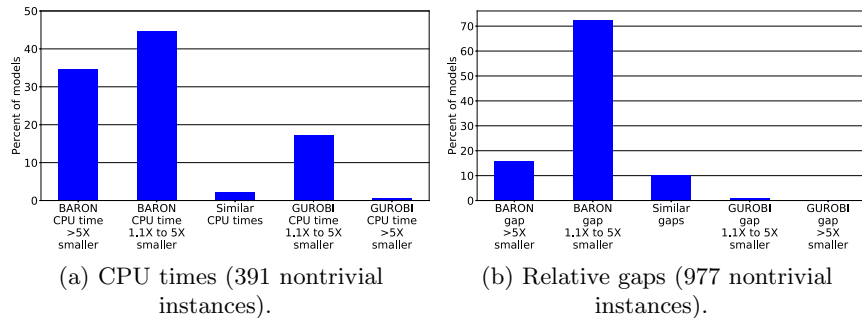

(a) CPU times (391 nontrivial instances).

(b) Relative gaps (977 nontrivial instances).

Figure 6: One-to-one comparison between BARON and GUROBI.

7 Conclusions

We considered the global optimization of nonconvex MIQPs with linear constraints. We introduced a family of convex quadratic relaxations which are constructed via quadratic cuts. We investigated the theoretical properties of these relaxations and showed that they are an outer-approximation of a semi-infinite convex program which under certain conditions is equivalent to a semidefinite program. To assess the benefits of our approach, we incorporated the proposed relaxation techniques into the global optimization solver BARON, and tested our implementation on a large collection of problems. Results demonstrated that, for our test problems, our implementation leads to a very significant improvement in the performance of BARON.

A Barrier coordinate minimization algorithm used to solve the nonsmooth regularized separation problem

In this appendix, we briefly describe the algorithm proposed by Dong [5] to solve the nonsmooth regularized separation problem (39). We then describe our implementation of

this algorithm. The algorithm operates on the following penalized log-det problem

$$
\inf_{d\in\mathbb{R}^n} \quad h(d;\sigma) := \sum_{i=1}^{n} r_i(d_i) - \sigma\log\text{-det}\left(Q + \text{diag}(d) + \alpha A^T A\right)
$$
$$
\text{s.t.} \quad Q + \text{diag}(d) + \alpha A^T A \succ 0
\tag{62}
$$

where $r_i(d_i) = \beta_i d_i$ for $d_i > 0$, $r_i(d_i) = \eta_i d_i$ for $d_i \leq 0$, $\beta_i = \eta_i + \lambda$, $\forall i \in [n]$, and $\sigma > 0$. Each iteration of this algorithm involves the update of a feasible vector $\bar{d}$ and an inverse matrix $V := \left[Q + \text{diag}(\bar{d}) + \alpha A^T A\right]^{-1}$. Based on the optimality condition for (62), this algorithm performs coordinate minimization by choosing an index i determined as:

$$
i = \arg\max_{j=1,\dots,n}\left\{\left|s(\bar{d})_j\right|\right\}, \quad \text{with} \ \ s(\bar{d}) = \arg\min_{u\in\mathbb{R}^n}\left\{\|u\|_2 \ : \ u \in \partial h(\bar{d};\sigma)\right\}.
\tag{63}
$$

where $\partial h(\bar{d};\sigma)$ is the subdifferential of $h(\bar{d};\sigma)$. This choice of i leads to a one-dimensional minimization problem similar to (46) but involving $h(\bar{d} + \Delta d_i e_i; \sigma)$. This problem can be solved analytically to obtain the following formula for Δd_i^* (see Section 4 in [5] for details):

$$
\Delta d_i^* = \begin{cases} \frac{\sigma}{\beta_i} - \frac{1}{V_{ii}}, & \text{if } -\bar{d}_i < -\frac{1}{V_{ii}} \ \text{ or } \ \sigma\frac{V_{ii}}{1-d_i V_{ii}} > \beta_i, \\ -\bar{d}_i, & \text{if } -\bar{d}_i \geq -\frac{1}{V_{ii}} \ \text{ and } \ \eta_i \leq \sigma\frac{V_{ii}}{1-d_i V_{ii}} \leq \beta_i, \\ \frac{\sigma}{\eta_i} - \frac{1}{V_{ii}}, & \text{if } -\bar{d}_i \geq -\frac{1}{V_{ii}} \ \text{ and } \ \sigma\frac{V_{ii}}{1-d_i V_{ii}} < \eta_i. \end{cases}
\tag{64}
$$

After calculating Δd_i^* according to (64), $\bar{d}$ and V are updated using (50) and (51), respectively. Once (62) has been solved within a given precision, the penalty parameter σ is adjusted through a rule similar to (53):

$$
\sigma \leftarrow \max\{\sigma_{\min}, \sigma_{\text{upd}} \cdot \sigma\} \quad \text{if} \quad \frac{\|s(\bar{d})\|_2}{\|\beta\|_2} \leq \epsilon_{\text{upd}}
\tag{65}
$$

where $s(\bar{d})$ is used as a measure of optimality. The relative improvement in the objective function of (39) is checked every $\omega_{\text{check}} n$ iterations, and the algorithm terminates if this relative improvement is smaller than ϵ_{check}.

The entire procedure is summarized in Algorithm 3. In our implementation of this algorithm, we use an initial perturbation $\hat{d} = -1.5\lambda_{\min}(Q + \alpha A^T A)\mathbb{1}$. We set the following parameters by using the values recommended in [5]: $\lambda = \sum_{i=1}^{n}\eta_i$, $\sigma_{\min} = 10^{-5}$, $\sigma_{\text{upd}} = 0.8$, $\epsilon_{\text{upd}} = 0.03$. We use MaxIter $= 500n$, $\omega_{\text{check}} = 10$ and $\epsilon_{\text{check}} = 10^{-4}$. The initial value of σ_{init} is determined as:

$$
\sigma_{\text{init}} = \text{median}\left\{\left|\frac{u_i}{V_{ii}}\right|\right\}_{i=1}^{n}
\tag{66}
$$

where $u_i \in \partial r_i(\hat{d}_i)$, and $\partial r_i(\hat{d}_i)$ is the subdifferential of $r_i(\hat{d}_i)$.

References

[1] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Dongarra, J., Croz, J.D., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: LAPACK Users' guide, vol. 9. Siam (1999)

[2] Bao, X., Sahinidis, N.V., Tawarmalani, M.: Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. Mathematical Programming **129**, 129–157 (2011)

Algorithm 3 Barrier coordinate minimization algorithm used to solve the nonsmooth regularized separation problem (39)

1: **Input**: Q, A, α, and an optimal solution $(\bar{x}, \bar{y}, \bar{v})$ to (33).
2: **Output**: A vector $\bar{d}$ that solves (39).
3: Set $\lambda = \sum_{i=1}^{n} \eta_i$.
4: Set $\bar{d} = \hat{d}$, where $\hat{d} = 1.5\mu\mathbb{1}$ and $\mu = -\lambda_{\min}(Q + \alpha A^T A)$.
5: Set $\sigma = \sigma_{\text{init}}$, where σ_{init} is calculated according to (66).
6: Calculate $V = \left[Q + \text{diag}(\bar{d}) + \alpha A^T A\right]^{-1}$ and set $k = 0$.
7: **While** ($k <$ MaxIter) **do**
8:   Update $k \leftarrow k + 1$.
9:   Determine an index i according to (63) and calculate Δd_i^* using (64).
10:   Update $\bar{d}$ according to (50).
11:   Update V according to (51).
12:   Adjust σ according to (65).
13:   **If** ($k \bmod (\omega_{\text{check}} n) = 0$) **then**
14:     Terminate if the improvement in the objective of (39) is smaller than ϵ_{check}.
15:   **End If**
16: **End While**

---

[3] Billionnet, A., Elloumi, S., Lambert, A.: An efficient compact quadratic convex reformulation for general integer quadratic programs. Computational Optimization and Applications **54**, 141–162 (2013)

[4] Buchheim, C., Wiegele, A.: Semidefinite relaxations for non-convex quadratic mixed-integer programming. Mathematical Programming **141**, 435–452 (2013)

[5] Dong, H.: Relaxing nonconvex quadratic functions by multiple adaptive diagonal perturbations. SIAM Journal on Optimization **26**, 1962–1985 (2016)

[6] Faye, A., Roupin, F.: Partial lagrangian relaxation for general quadratic programming. 4OR **5**, 75–88 (2007)

[7] Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of ACM **42**, 1115–1145 (1995)

[8] Khajavirad, A., Sahinidis, N.V.: A hybrid LP/NLP paradigm for global optimization relaxations. Mathematical Programming Computation **10**, 383–421 (2018)

[9] Koopmans, T.C., Beckmann, M.: Assignment problems and the location of economic activities. Econometrica: Journal of the Econometric Society **25**, 53–76 (1957)

[10] M. Kılınç and N. V. Sahinidis: Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems in BARON. Optimization Methods and Software **33**, 540–562 (2019)

[11] McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems. Mathematical Programming **10**, 147–175 (1976)

[12] Nohra, C.J., Raghunathan, A.U., Sahinidis, N.V.: Spectral relaxations and branching strategies for global optimization of mixed-integer quadratic programs. SIAM Journal on Optimization **31**, 142–171 (2021)

[13] Nohra, C. J. and Raghunathan, A. U. and Sahinidis, N. V.: A test set of quadratic, binary quadratic and integer quadratic programs. Available at `ftp://ftp.merl.com/pub/raghunathan/MIQP-TestSet/`

[14] Pardalos, P.M., Glick, J.H., Rosen, J.B.: Global minimization of indefinite quadratic problems. Computing **539**, 281–291 (1987)

[15] Phillips, A., Rosen, J.: A quadratic assignment formulation of the molecular conformation problem. Journal of Global Optimization **4**, 229–241 (1994)

[16] Saxena, A., Bonami, P., Lee, J.: Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. Mathematical programming **130**, 359–413 (2011)

[17] Sherali, H.D., Adams, W.P.: A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems, *Nonconvex Optimization and its Applications*, vol. 31. Kluwer Academic Publishers, Dordrecht (1999)

[18] Sherali, H.D., Wang, H.: Global optimization of nonconvex factorable programming problems. Mathematical Programming **89**, 459–478 (2001)

[19] Shor, N.: Quadratic optimization problems. Soviet Journal of Computer and Systems Sciences **25**, 1–11 (1987)

[20] Tawarmalani, M., Sahinidis, N.V.: Convex extensions and convex envelopes of l.s.c. functions. Mathematical Programming **93**, 247–263 (2002)

[21] Tawarmalani, M., Sahinidis, N.V.: Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. Mathematical Programming **99**, 563–591 (2004)

[22] Tuy, H.: DC optimization: Theory, methods and algorithms. *In R. Horst and P. M. Pardalos (eds.)*, Handbook of Global Optimization, *Kluwer Academic Publishers, Boston, MA* pp. 149–216 (1995)