Relaxing the Optimality Conditions of Box QP*

Samuel Burer† Jieqiu Chen‡

October 23, 2007

Abstract

We present semidefinite relaxations of nonconvex, box-constrained quadratic programming, which incorporate the first- and second-order necessary optimality conditions. We compare these relaxations with a basic semidefinite relaxation due to Shor, particularly in the context of branch-and-bound to determine a global optimal solution, where it is shown empirically that the new relaxations are significantly stronger. We also establish theoretical relationships between the new relaxations and Shor's relaxation.

1 Introduction

In this paper, we study semidefinite programming (SDP) relaxations for the fundamental problem of minimizing a nonconvex quadratic function over a box:

$$\min \left\{ \frac{1}{2} x^T Q x + c^T x \ : \ 0 \le x \le e \right\}, \tag{1}$$

where $x \in \Re^n$, $Q \in \Re^{n \times n}$, $c \in \Re^n$, and $e \in \Re^n$ is the all-ones vector. Without loss of generality, we assume Q is symmetric. If Q is not positive semidefinite (as we assume in this paper), then (1) is NP-hard (Pardalos and Vavasis, 1991).

There are numerous methods for solving (1) and more general nonconvex quadratic programs, including local methods (Gould and Toint, 2002) and global methods (Pardalos, 1991). For a survey of methods to globally solve (1), see De Angelis et al. (1997) as well as Vandenbussche and Nemhauser (2005a,b) and Burer and Vandenbussche (2006).

Critical to any global optimization method for (1) is the ability to relax (1) into a convex problem, one which hopefully provides a tight lower bound on the optimal value with low computational cost. One standard approach is to linearize the quadratic term $x_i x_j$ via a

*Department of Management Sciences, University of Iowa, Iowa City, IA, 52242-1994, U.S.A. Both authors supported in part by NSF Grant CCF-0545514.

†Email: samuel-burer@uiowa.edu.

‡Corresponding author. Email: jieqiu-chen@uiowa.edu.

single variable X_{ij} and then to enforce implied linear constraints, which link X_{ij} with x_i and x_j, e.g., $0 \le X_{ij} \le \min\{x_i, x_j\}$ (Sherali and Adams, 1997). The resulting relaxation is a linear program. A second approach also linearizes the terms $x_i x_j$ (by introducing a symmetric matrix variable X to replace the aggregate xx^T) but then includes the valid semidefinite inequality $X \succeq xx^T$ to obtain an SDP relaxation.

In this paper, we focus on SDP relaxations of (1) rather than linear ones. In principle, it is always possible to combine linear and semidefinite approaches (yielding better bounds with added computational costs; see Anstreicher (2007)), but the goal of this paper is to improve SDP relaxations.

Our approach is to consider semidefinite relaxations of (1), which incorporate the standard first- and second-order necessary optimality conditions for (1). Vandenbussche and Nemhauser (2005a,b) and Burer and Vandenbussche (2006) have previously considered linear and semidefinite relaxations, respectively, involving only the first-order conditions. The contributions of the current paper are to demonstrate how also to incorporate the second-order conditions and to illustrate the positive effects of doing so.

We point out that Nesterov (2000) has considered incorporating the second-order conditions into SDP relaxations of quadratic optimization over p-norm boxes for $2 \le p < \infty$, i.e., $\{x : \|x\|_p^p \le 1\}$. However, Nesterov strongly uses the fact that the function $\|x\|_p^p$ is smooth for $p \in [2, \infty)$. Our case (p equal to ∞) is wholly different because of the lack of smoothness.

The paper is organized as follows. In Section 2, we review the first- and second-order optimality conditions of (1). In particular, we show how to express the second-order conditions without explicit knowledge of the inactive constraints. This will prove to be a critical ingredient in constructing semidefinite relaxations of the second-order conditions. In Section 3, we review the basic semidefinite relaxation of (1) due to Shor (1987) and then introduce our semidefinite relaxation, which incorporates the optimality conditions. We also construct a relaxation which includes only the second-order conditions (but not the first-order).

We will call the three relaxations just mentioned (SDP_0), (SDP_{12}), and (SDP_2), respectively. The subscript indicates the type of "order" information incorporated in the relaxation. By construction, it will hold that (SDP_{12}) is at least as strong as (SDP_2), which is at least as strong as (SDP_0). On the other hand, (SDP_{12}) takes the most time to solve, while (SDP_0) takes the least.

Continuing in Sections 4 and 5, we study the relationship between (SDP_0) and (SDP_2). We first establish the surprising fact that both achieve the same optimal value. On the other hand, in the context of branch-and-bound to globally solve (1), we demonstrate that (SDP_2) is significantly stronger than (SDP_0)—when both are appropriately tailored for use at any node of the tree. In particular, branching is done by subdividing the entire box into smaller boxes of the type $\{x : l \leq x \leq u\}$, and the relaxations (SDP$_2$) and (SDP$_0$) are modified to incorporate the bounds (l, u) at any particular node in the branch-and-bound tree. In this context, the key insight is that (SDP$_2$) enforces the second-order optimality conditions as a "global" constraint, irrespective of (l, u), whereas (SDP$_0$) only incorporates "local" information based on (l, u). Note that, in Sections 4 and 5, we analyze (SDP$_2$) instead of (SDP$_{12}$) in order to isolate the effect of the second-order conditions in line with the paper's main contributions.

Finally, in Section 6, we show further that (SDP$_0$) and (SDP$_{12}$) also achieve the same optimal value. The proof establishes interesting analytical properties of (SDP$_0$), which are of independent interest.

1.1 Notation and terminology

In this section, we introduce some of the notation that will be used throughout the paper. $\Re^n$ refers to n-dimensional Euclidean space; $\Re^{n\times n}$ is the set of real, $n \times n$ matrices. We let $e_i \in \Re^n$ represent the i-th unit vector. For a set $\mathcal{I}$, $\mathcal{I}^c$ is the complement of $\mathcal{I}$. The norm of a vector $v \in \Re^n$ is denoted by $\|v\| := \sqrt{v^T v}$. Let $\mathcal{I}$ denote an index set. Then $v_{\mathcal{I}}$ is defined as the vector composed of entries of v that are indexed by $\mathcal{I}$; given a matrix $A \in \Re^{n\times n}$, $A_{\mathcal{I}\mathcal{I}}$ is defined as the matrix composed of entries of A whose rows and columns are indexed by $\mathcal{I}$. We denote by $A_{\cdot j}$ and $A_{i\cdot}$ the j-th column and i-th row of A, respectively. The notation $\mathrm{diag}(A)$ is defined as the vector with the diagonal of the matrix A as its entries while $\mathrm{Diag}(v)$ denotes the diagonal matrix with diagonal being v. The inner product of two matrices $A, B \in \Re^{n\times n}$ is defined as $A \bullet B := \mathrm{trace}(A^T B)$. Given two vectors $x, v \in \Re^n$, we denote their Hadamard product by $x \circ v \in \Re^n$, where $[x \circ v]_j = x_j v_j$; an analogous definition applies to the Hadamard product of matrices. Finally, $A \succeq 0$ means matrix A is positive semidefinite, and $A \succ 0$ means A is positive definite.

2 Optimality Conditions

In this section, we first state the standard first- and second-order necessary optimality conditions for (1) involving the set of inactive constraints. Then we derive an expression for the second-order conditions that does not explicitly require knowledge of the inactive constraint set.

For any x satisfying $0 \le x \le e$, define the following sets of inactive constraints:

$$\begin{aligned}
\mathcal{I}_0(x) &:= \{i : x_i > 0\} \\
\mathcal{I}_1(x) &:= \{i : x_i < 1\} \\
\mathcal{I}(x) &:= \{i : 0 < x_i < 1\} = \mathcal{I}_0(x) \cap \mathcal{I}_1(x).
\end{aligned}$$

Note that $\mathcal{I}(x)^c = \mathcal{I}_0(x)^c \cup \mathcal{I}_1(x)^c$ indexes the active constraints at x. Let $y, z \in \Re^n$ denote the Lagrange multipliers for the constraints $e - x \ge 0$ and $x \ge 0$, respectively. For fixed y, z, the Lagrangian of (1) is defined as

$$L(x; y, z) := \frac{1}{2} x^T Q x + c^T x - z^T x - y^T (e - x).$$

With these definitions, the necessary optimality conditions for (1) are

$$\begin{aligned}
0 \le x &\le e && && \text{(2a)} \\
\nabla_x L(x; y, z) = Qx + c - z + y &= 0 && && \text{(2b)} \\
y \ge 0,\ z &\ge 0 && && \text{(2c)} \\
z_i = 0,\ y_j &= 0 && \forall\, i \in \mathcal{I}_0(x),\ \forall\, j \in \mathcal{I}_1(x) && \text{(2d)} \\
v^T \nabla^2_{xx} L(x; y, z) v = v^T Q v &\ge 0 && \forall\, v \in V(x) && \text{(2e)}
\end{aligned}$$

where

$$\begin{aligned}
V(x) &:= \{\ v\ :\ e_i^T v = 0\ \ \forall\ \ i \in \mathcal{I}_0(x)^c,\ \ -e_j^T v = 0\ \ \forall\ \ j \in \mathcal{I}_1(x)^c\ \} \\
&= \{\ v\ :\ v_i = 0\ \ \forall\ \ i \in \mathcal{I}(x)^c\ \}
\end{aligned}$$

is the null space of the Jacobian of the active constraints. By eliminating z and employing other straightforward simplifications, we can rewrite and label (2) as

$$\begin{aligned}
0 \le x &\le e && \text{(primal feasibility)} && \text{(3a)} \\
Qx + c + y \ge 0,\ y &\ge 0 && \text{(dual feasibility)} && \text{(3b)} \\
x \circ (Qx + c + y) = 0,\ y \circ (e - x) &= 0 && \text{(complementary slackness)} && \text{(3c)} \\
Q_{\mathcal{I}(x)\mathcal{I}(x)} &\succeq 0. && \text{(local convexity)} && \text{(3d)}
\end{aligned}$$

Now we give an equivalent form of the local convexity condition (3d), which does not explicitly involve knowledge of $\mathcal{I}(x)$.

Proposition 2.1. *Given x, define*

$$w := x \circ (e - x). \tag{4}$$

Then the local convexity condition (3d) is equivalent to

$$Q \circ ww^T \succeq 0. \tag{5}$$

Proof. For notational convenience, we write $\mathcal{I}$ for $\mathcal{I}(x)$ and D for $\mathrm{Diag}(x)$. We first show the equivalence of (3d) and the inequality

$$(I - D)DQD(I - D) \succeq 0.$$

Assume (3d) holds. By definition, $D(I-D)$ is a diagonal matrix such that, for all $i \in \mathcal{I}(x)^c$, the i-th diagonal entry is 0. For any v, define $\tilde{v} := D(I - D)v$. Then $\tilde{v}_i = 0$ for all $i \in \mathcal{I}^c$ and

$$v^T(I - D)DQD(I - D)v = \tilde{v}^T Q\tilde{v} = \tilde{v}_{\mathcal{I}}^T Q_{\mathcal{I}\mathcal{I}}\tilde{v}_{\mathcal{I}} \geq 0.$$

So $(I-D)DQD(I-D)$ is positive semidefinite. Conversely, assume $(I-D)DQD(I-D) \succeq 0$. Since $D_{\mathcal{I}\mathcal{I}} > 0$ and $[I - D]_{\mathcal{I}\mathcal{I}} > 0$, for any partial vector $\tilde{v}_{\mathcal{I}}$, there exists some v such that the full vector $\tilde{v} := D(I - D)v$ extends $\tilde{v}_{\mathcal{I}}$ and also satisfies $\tilde{v}_{\mathcal{I}^c} = 0$. So

$$\tilde{v}_{\mathcal{I}}^T Q_{\mathcal{I}\mathcal{I}}\tilde{v}_{\mathcal{I}} = \tilde{v}^T Q\tilde{v} = v^T(I - D)DQD(I - D)v \geq 0,$$

which establishes (3d).

Now, the equivalence of (3d) and $Q \circ ww^T \succeq 0$ follows from

$$(I - D)DQD(I - D) = \mathrm{Diag}(w)Q\,\mathrm{Diag}(w) = Q \circ ww^T.$$

□

It follows from Proposition 2.1 that (1) can be reformulated as the following quadratic semidefinite program, which does not depend explicitly on knowledge of the inactive constraints:

$$\min\left\{ \frac{1}{2}x^T Qx + c^T x \ : \ \ \text{(3a–3c)} \ \ (4) \ \ (5) \ \right\}. \tag{6}$$

3 Semidefinite Relaxations

In this section, we first present the basic semidefinite relaxation of (1) due to Shor (1987). Then we introduce semidefinite relaxations of the new formulation (6).

3.1 Shor's bounded relaxation (SDP$_0$)

As is standard in the SDP literature, we can represent (1) in the equivalent form

$$\min \left\{ \frac{1}{2} Q \bullet X + c^T x \ : \ 0 \leq x \leq e, \ \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0, \ X = xx^T \right\}.$$

By dropping the constraint $X = xx^T$, we obtain the relaxation due to Shor:

$$\min \left\{ \frac{1}{2} Q \bullet X + c^T x \ : \ 0 \leq x \leq e, \ \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 \right\}. \tag{7}$$

However, the following well known fact about (7) is easy to prove:

Proposition 3.1. *If $Q \not\succeq 0$, then the optimal value of (7) is $-\infty$.*

The reason why (7) is unbounded is that there is too much freedom for X. We can fix the problem of unboundedness by including some valid linear constraints implied by $X = xx^T$ and $0 \leq x \leq e$, e.g., $\mathrm{diag}(X) \leq x$ (Sherali and Adams, 1997). Adding $\mathrm{diag}(X) \leq x$ to (7), we get a bounded relaxation for (1):

$$\min \left\{ \frac{1}{2} Q \bullet X + c^T x \ : \ 0 \leq x \leq e, \ \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0, \ \mathrm{diag}(X) \leq x \right\}. \tag{SDP$_0$}$$

In particular, the optimal solution set of (SDP$_0$) is nonempty. We consider (SDP$_0$) to be the smallest, simplest semidefinite relaxation of (1).

We remark that Ye (1999) has derived an approximation algorithm for quadratic programming over the box $\{x : -e \leq x \leq e\}$, which is simply a shifted and scaled version of (1). The main tool used by Ye is the equivalent version of (SDP$_0$) for the case $\{x : -e \leq x \leq e\}$.

3.2 Relaxations (SDP$_{12}$) and (SDP$_2$) of the optimality conditions

To relax (6), we consider the matrix

$$\begin{pmatrix} 1 \\ x \\ y \\ w \end{pmatrix} \begin{pmatrix} 1 \\ x \\ y \\ w \end{pmatrix}^T = \begin{pmatrix} 1 & x^T & y^T & w^T \\ x & xx^T & xy^T & xw^T \\ y & yx^T & yy^T & yw^T \\ w & wx^T & wy^T & ww^T \end{pmatrix} \succeq 0$$

and its linearized version

$$M = \begin{pmatrix} 1 & x^T & y^T & w^T \\ x & X & M_{xy}^T & M_{xw}^T \\ y & M_{xy} & Y & M_{yw}^T \\ w & M_{xw} & M_{yw} & W \end{pmatrix} \succeq 0.$$

We can relax the quadratic constraints (3c), (4) and (5) via M. For example, consider the j-th entry of $x \circ (Qx + c + y) = 0$ from (3c), which is

$$x_j(Q_{j\cdot}x + c_j + y_j) = 0.$$

Relaxing it via M, we have

$$Q_{j\cdot}X_{\cdot j} + c_j x_j + [M_{xy}]_{jj} = 0.$$

So, in total, $x \circ (Qx + c + y) = 0$ is relaxed as

$$\mathrm{diag}(QX) + c \circ x + \mathrm{diag}(M_{xy}) = 0.$$

Constraints (4) and (5) can be relaxed in a similar way. Hence, we obtain the following SDP relaxation of (6), which we call (SDP_{12}):

$$\begin{aligned}
\min \quad & \frac{1}{2} Q \bullet X + c^T x && \text{(8a)}\\
\text{s.t.} \quad & 0 \le x \le e, \;\; \text{diag}(X) \le x && \text{(8b)}\\
& Qx + c + y \ge 0, \;\; y \ge 0 && \text{(8c)}\\
& \text{diag}(QX) + c \circ x + \text{diag}(M_{yx}) = 0, \;\; y - \text{diag}(M_{yx}) = 0 && \text{(8d)}\\
& w = x - \text{diag}(X) && \text{(8e)}\\
& Q \circ W \succeq 0 && \text{(8f)}\\
& M \succeq 0. && \text{(8g)}
\end{aligned}$$

We point out that $\text{diag}(X) \le x$ in (8b) is not a relaxation of a particular constraint in (6). Rather, it is added to prevent (SDP_{12}) from being unbounded as with (SDP_0).

We also study a relaxed version of (SDP_{12}), which we call (SDP_2):

$$\begin{aligned}
\min \quad & \frac{1}{2} Q \bullet X + c^T x && \text{(9a)}\\
\text{s.t.} \quad & 0 \le x \le e, \;\; \text{diag}(X) \le x && \text{(9b)}\\
& w = x - \text{diag}(X) && \text{(9c)}\\
& Q \circ W \succeq 0 && \text{(9d)}\\
& \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 && \text{(9e)}\\
& \begin{pmatrix} 1 & w^T \\ w & W \end{pmatrix} \succeq 0. && \text{(9f)}
\end{aligned}$$

In essence, (SDP_2) maintains the minimal set of constraints from (SDP_{12}), which still explicitly relax the second-order optimality conditions. We are interested in (SDP_2) because it captures the second-order optimality information (the main focus of this paper) and because its dimension is significantly lower than that of (SDP_{12}).

4 Equivalence of (SDP_0) and (SDP_2)

In this section, we establish the equivalence of (SDP_0) and (SDP_2). We will use the following generic result:

Lemma 4.1. *Consider two related optimization problems:*

$$\begin{array}{ll} (A) & \min\{\ f(x)\ : x \in P\ \} \\ (B) & \min\{\ f(x)\ : x \in P,\ y \in R(x)\ \}. \end{array}$$

Let x^* be an optimal solution of (A) and suppose $R(x^*) \neq \emptyset$. Then any (x^*, y) with $y \in R(x^*)$ is optimal for (B). Therefore, the optimal value of (B) equals that of (A).

Our first step is to prove the following property of (SDP_0) at optimality.

Lemma 4.2. *Let (x^*, X^*) be an optimal solution of (SDP_0), and define $\mathcal{J} := \{i : X^*_{ii} < x^*_i\}$. Then $Q_{\mathcal{J}\mathcal{J}} \succeq 0$.*

Proof. The argument is based on examining an optimal solution for the dual of (SDP_0). It can be easily verified that the dual is

$$\begin{array}{ll} \max & \lambda - e^T y \\ \text{s.t.} & S = \dfrac{1}{2}\begin{pmatrix} -\lambda & (c+y-z-v)^T \\ c+y-z-v & Q + 2\,\mathrm{Diag}(v) \end{pmatrix} \succeq 0 \\ & y, z, v \geq 0, \end{array}$$

where y, z, v, S, and λ are, respectively, the multipliers for $e - x \geq 0$, $x \geq 0$, $x - \mathrm{diag}(X) \geq 0$, $(1, x^T; x, X) \succeq 0$, and the constraint which fixes the top-left entry of $(1, x^T; x, X)$ to 1.

Note that both (SDP_0) and its dual have nonempty interior. Specifically, the point

$$(x, X) = \left(\frac{1}{2}e, \frac{1}{4}ee^T + \varepsilon I\right)$$

is interior feasible for (SDP_0) for all $\varepsilon \in (0, 1/4)$. In addition, taking v sufficiently positive, λ sufficiently negative, and y, z positive such that $y - z - v$ has sufficiently small norm yields an interior solution of the dual with $S \succ 0$. Because both problems have interiors, strong duality holds. For the remainder of the proof, we let (x, X) and (λ, y, z, v, S) denote specific optimal solutions of the primal and dual.

Due to complementary slackness, $(x - \mathrm{diag}(X)) \circ v = 0$. So $v_{\mathcal{J}} = 0$, and it follows from $S \succeq 0$ that

$$[Q + 2\,\mathrm{Diag}(v)]_{\mathcal{J}\mathcal{J}} = Q_{\mathcal{J}\mathcal{J}} \succeq 0.$$

□

Theorem 4.3. *Let (x^*, X^*) be an optimal solution of (SDP$_0$), and define $w^* := x^* - \operatorname{diag}(X^*)$ and $W^* := w^*(w^*)^T$. Then (x^*, X^*, w^*, W^*) is an optimal solution of (SDP$_2$) with the same optimal value.*

Proof. For notational convenience, we drop the $*$ superscripts. By Lemma 4.1, we need only prove that (x, X, w, W) is feasible for (SDP$_2$), and to do so requires the verification of (9d) since all the other constraints of (SDP$_2$) are satisfied by construction.

Let $\mathcal{J}$ be defined as in Lemma 4.2. Then $w_{\mathcal{J}} > 0$, $w_{\mathcal{J}^c} = 0$, and $Q_{\mathcal{J}\mathcal{J}} \succeq 0$. Note that $[Q \circ W]_{ij} = 0$ if $i \in \mathcal{J}^c$ or $j \in \mathcal{J}^c$. So $Q \circ W = Q \circ ww^T \succeq 0$ is equivalent to $Q_{\mathcal{J}\mathcal{J}} \circ \left(w_{\mathcal{J}} w_{\mathcal{J}}^T\right) = \operatorname{Diag}(w_{\mathcal{J}}) Q_{\mathcal{J}\mathcal{J}} \operatorname{Diag}(w_{\mathcal{J}}) \succeq 0$, which is true because $Q_{\mathcal{J}\mathcal{J}} \succeq 0$ and $w_{\mathcal{J}} > 0$. This proves (9d) and hence the proposition. □

5 Comparison of (SDP$_0$) and (SDP$_2$) Within Branch-and-Bound

We have shown that the SDP relaxation (SDP$_2$), which incorporates the second-order optimality conditions, is equivalent to Shor's bounded relaxation (SDP$_0$). In this section, we empirically compare the two relaxations in the context of branch-and-bound for globally solving (1). In particular, branching is done by recursively subdividing the entire box into smaller boxes of the type $\{x : l \le x \le u\}$, and the relaxations (SDP$_2$) and (SDP$_0$) are modified to incorporate the bounds (l, u) at any particular node in the branch-and-bound tree. The key insight is then that (SDP$_2$) and (SDP$_0$) can have different effects on small boxes, even though they have the same effect on the entire box, i.e., at the root node (as detailed in Section 4). The reason is the explicit use of the second-order optimality conditions in (SDP$_2$), which enforce "global" second-order information at the "local" level. In contrast, (SDP$_0$) only incorporates local information based on (l, u). Overall, our experiments on randomly generated problems illustrate the strength of the bounds produced by (SDP$_2$) over those produced by (SDP$_0$) in this context.

We would like to point out that our intention here is *not* to develop a branch-and-bound method for (1), which outperforms other existing techniques. Rather, our goal is limited to comparing (SDP$_0$) and (SDP$_2$) in order to gauge the effect of incorporating the second-order conditions into SDP relaxations for (1).

5.1 Branch-and-bound for box QP

The branch-and-bound algorithm we consider recursively subdivides the entire box $\{x \in \Re^n : 0 \le x \le e\}$ into smaller and smaller boxes and solves an appropriately tailored SDP relaxation (either (SDP_0) or (SDP_2)) on these smaller boxes. Lower bounds obtained from these relaxations are compared with a global upper bound to fathom as many small boxes as possible from consideration. When fathoming is not possible for a specific small box, that box is further subdivided. Moreover, the global upper bound is improved (where possible) throughout the course of the algorithm.

The performance of the branch-and-bound algorithm is measured in two ways: the total number of nodes in the branch-and-bound tree and the total time to complete the branch-and-bound process. The number of nodes is determined by the quality of both the lower and upper bounds. Our main comparison will be the lower bound calculations (based on either (SDP_0) or (SDP_2)). However, we will also consider two different ways to generate the global upper bound.

Before discussing our algorithm design choices below, we first present the SDP relaxations on the small boxes, which are modified slightly from the corresponding versions on the entire box. Suppose the current node of the branch-and-bound tree corresponds to the box

$$\{\ x \in \Re^n :\ l \le x \le u\ \}.$$

Then the SDP relaxations that correspond to (SDP_0) and (SDP_2) on this box are, respectively:

$$\begin{aligned}
\min \quad & \frac{1}{2} Q \bullet X + c^T x && \text{(10a)}\\
\text{s.t.} \quad & l \le x \le u && \text{(10b)}\\
& \operatorname{diag}(X) - (l+u) \circ x + l \circ u \le 0 && \text{(10c)}\\
& \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 && \text{(10d)}
\end{aligned}$$

$$
\begin{aligned}
\min \quad & \frac{1}{2} Q \bullet X + c^T x && \text{(11a)} \\
\text{s.t.} \quad & l \le x \le u && \text{(11b)} \\
& \operatorname{diag}(X) - (l+u) \circ x + l \circ u \le 0 && \text{(11c)} \\
& w = x - \operatorname{diag}(X) && \text{(11d)} \\
& Q \circ W \succeq 0 && \text{(11e)} \\
& \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 && \text{(11f)} \\
& \begin{pmatrix} 1 & w^T \\ w & W \end{pmatrix} \succeq 0. && \text{(11g)}
\end{aligned}
$$

Both relaxations have the constraint $\operatorname{diag}(X) - (l+u) \circ x + l \circ u \le 0$, which is obtained from relaxing the valid inequality

$$x \circ x - (l+u) \circ x + l \circ u = (x - l) \circ (x - u) \le 0.$$

Note that, when $l = 0$ and $u = e$, this constraint is just $\operatorname{diag}(X) \le x$. So this inequality serves the role of bounding the diagonal of X on the smaller boxes. Relaxation (11) has all the constraints of (10) plus the second-order constraints (9c), (9d) and (9f) of (SDP$_2$). As mentioned at the beginning of this section, explicitly incorporating the "global" second-order optimality conditions will allow (SDP$_2$) to outperform (SDP$_0$) within branch-and-bound.

Now we address the major design choices for the branch-and-bound algorithm:

- **Bounding**. We will compare two types of lower bounds: those given by (10) and those given by (11). Note that a single run of the branch-and-bound algorithm on a single instance will employ one or the other (not both).

  For the global upper bound, we experimented with two ways to improve it at each node of the tree. The first way is to locally solve the small box QP at each node via MATLAB's `quadprog` function. The second way is to simply take the objective value $\frac{1}{2}(x^*)^T Q x^* + c^T x^*$ corresponding to the optimal x^* obtained from the lower bound calculation. Note that a single run of the branch-and-bound algorithm on a single instance will employ one way or the other (not both).

  In total, each instance will be solved four times corresponding to all possible combinations of lower bound and upper bound calculations.

- **Branching**. The branching strategy we use is the standard "bisection via longest edge" (see, for example, Horst and Tuy (1993)). Consider the small box $\{x \in \Re^n : l \le x \le u\}$, which has been selected for branching. We select the longest edge of this box to branch on. More specifically, we choose the index i such that $u_i - l_i$ is the largest among all dimensions. If there is a tie, the smallest such index is chosen. By applying this strategy, we subdivide the box into two smaller boxes:

$$\left\{ x \in \Re^n \ : \ l \le x \le u - \frac{1}{2}(u_i - l_i)e_i \right\},$$
$$\left\{ x \in \Re^n \ : \ l + \frac{1}{2}(u_i - l_i)e_i \le x \le u \right\}.$$

- **Node Selection**. We use a best-bound strategy for selecting the next node to solve in the branch-and-bound tree.

- **Fathoming Tolerance**. A relative optimality tolerance is used for fathoming. For a given tolerance *tol*, a node with lower bound L is fathomed if $(U - L)/\max\{1, \frac{1}{2}(|U| + |L|)\} < tol$, where U is the current global upper bound obtained. In our experiments, we set $tol = 10^{-3}$.

5.2 Implementation and results

For $n = 20$, we generated 100 instances of random data (Q, c) (all entries uniform in $[-1, 1]$) and solved these instances using the branch-and-bound scheme outlined above. In particular, each instance was solved four times corresponding to all possible combinations of lower bound and upper bound calculations. Specifically, we tested each instance with:

(i) lower bounds by (SDP_0) and upper bounds by `quadprog`;

(ii) lower bounds by (SDP_2) and upper bounds by `quadprog`;

(iii) lower bounds by (SDP_0) and upper bounds by extracting x^*;

(iv) lower bounds by (SDP_2) and upper bounds by extracting x^*.

The algorithm was coded in MATLAB and all SDP relaxations were setup and solved using YALMIP (Löfberg, 2004) and SeDuMi (Sturm, 1999). All computations were performed on a Pentium D running at 3.20GHz under the Linux operating system.

Figure 1 contains two log-log plots which depict the number of nodes taken by all runs. The first plot compares (i) with (ii), and the second compares (iii) with (iv). Each figure includes a "$y = x$" line, which divides the plot into two regions. In particular, a point plotted in the lower-right region for a particular instance indicates that (SDP_2) required fewer nodes

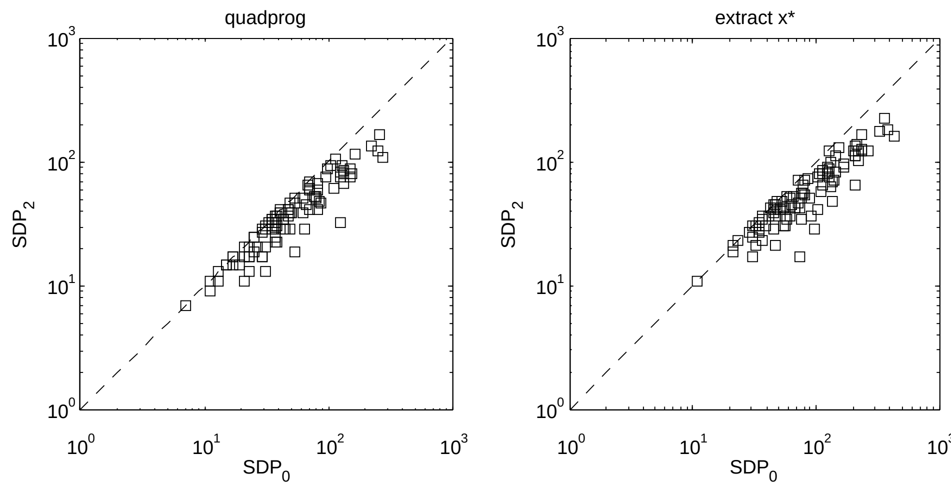

Figure 1: Nodes required by (SDP_0) and (SDP_2) (under two upper-bound calculations).

than (SDP_0). Figure 2 contains similar log-log plots depicting total CPU time (in seconds) of all 400 runs. In this case, a point plotted in the upper-left region indicates that (SDP_2) required more time than (SDP_0).

We summarize our key observations as follows:

- In all runs, the number of nodes required by (SDP_2) is no more than the number required by (SDP_0), which indicates that the former is a stronger relaxation than the latter. For instances that are hard to solve, i.e., instances requiring a lot of nodes, the number of nodes needed by (SDP_2) is significantly less than that of (SDP_0).

- The CPU times required by (SDP_2) are longer than those of (SDP_0) in most cases. However, there are instances where the number of nodes required by (SDP_0) is so large such that the overall CPU time used by (SDP_2) is less.

- The upper bounds obtained via `quadprog` are better than those obtained without it. In runs where `quadprog` was used, both the number of nodes and CPU time were reduced.

- When the global upper bound is calculated by extracting x^*, the ratio

$$\frac{\#\text{ of (SDP}_2\text{) nodes}}{\#\text{ of (SDP}_0\text{) nodes}}$$

averages 0.70 over all 100 instances. On the other hand, when `quadprog` is used, the ratio averages 0.77. This indicates that (SDP_2) contains better information for the global upper bound than (SDP_0). (Still, it is overall better to use `quadprog` as mentioned in the previous item.)

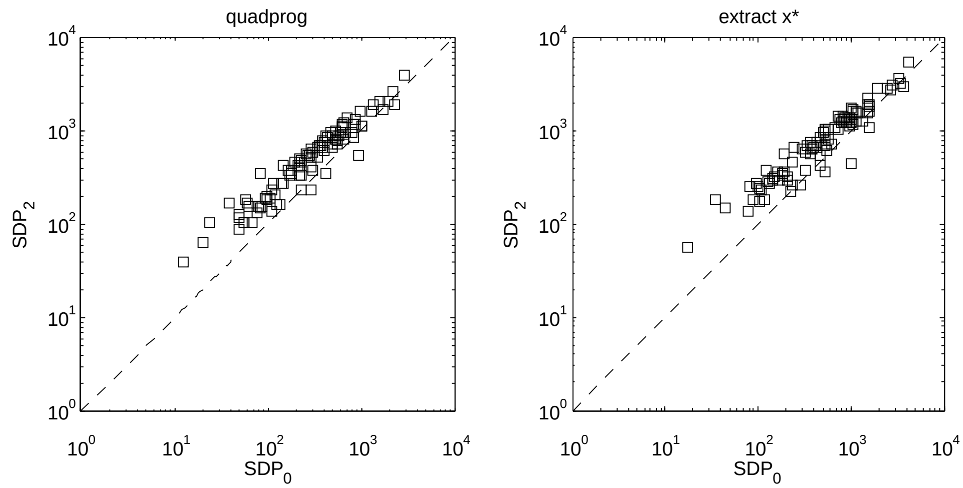

Figure 2: CPU times (in seconds) required by (SDP_0) and (SDP_2) (under two upper-bound calculations).

6 Equivalence of (SDP_0) and (SDP_{12})

In Section 4, we have proved that (SDP_2) is equivalent to (SDP_0). In this section, we show that even (SDP_{12}) is equivalent to (SDP_0). We start by proving some properties of (SDP_0), which will facilitate the proof of equivalence later in Section 6.2, but are also of independent interest.

6.1 Additional properties of (SDP_0)

We will show that every optimal solution (x^*, X^*) of (SDP_0) satisfies the following two inequalities:

$$\operatorname{diag}(QX) + c \circ x \leq 0 \tag{12a}$$

$$Qx + c - (\operatorname{diag}(QX) + c \circ x) \geq 0. \tag{12b}$$

In other words, (12a) and (12b) are redundant for (SDP_0) in the sense that enforcing these inequalities does not change the optimal solution set. This knowledge will help us establish the equivalence between (SDP_{12}) and (SDP_0) in the next subsection.

To prove (12), we start by examining paths of solutions in the feasible set of (SDP_0). Given any feasible (x, X), consider two paths of emanating from (x, X) and depending on a specified index i. Each path is parameterized by $\alpha \geq 0$. We define $(x_1(\alpha), X_1(\alpha))$ and

$(x_2(\alpha), X_2(\alpha))$ by

$$\begin{aligned} x_1(\alpha) &:= x - \alpha x_i e_i \\ X_1(\alpha) &:= X - \alpha e_i X_{\cdot i}^T - \alpha X_{\cdot i} e_i^T + \alpha^2 X_{ii} e_i e_i^T \\ x_2(\alpha) &:= x + \alpha e_i \\ X_2(\alpha) &:= X + \alpha e_i x^T + \alpha x e_i^T + \alpha^2 e_i e_i^T. \end{aligned}$$

Furthermore, for any $\beta \in [0, 1]$, we consider a third path $(x(\alpha), X(\alpha))$, which is a convex combination of $(x_1(\alpha), X_1(\alpha))$ and $(x_2(\alpha), X_2(\alpha))$:

$$\begin{aligned} x(\alpha) &:= \beta x_1(\alpha) + (1 - \beta) x_2(\alpha) \\ X(\alpha) &:= \beta X_1(\alpha) + (1 - \beta) X_2(\alpha). \end{aligned}$$

Our intent is to examine conditions on α and β such that $(x(\alpha), X(\alpha))$ is feasible for (SDP$_0$). We will also be interested in the objective value at $(x(\alpha), X(\alpha))$:

$$f(\alpha) := \frac{1}{2} Q \bullet X(\alpha) + c^T x(\alpha)$$

Proposition 6.1. *Let an index i be specified. Given (x, X) feasible for (SDP$_0$) and $\beta \in [0, 1]$, $(x(\alpha), X(\alpha))$ satisfies the following properties:*

(i) $x(\alpha)$ differs from x only in the i-th entry, and $x(\alpha)_i = x_i + \alpha(1 - \beta - \beta x_i)$;

(ii) $\operatorname{diag}(X(\alpha)) - x(\alpha)$ differs from $\operatorname{diag}(X) - x$ only in the i-th entry, and

$$\begin{aligned} &[\operatorname{diag}(X(\alpha)) - x(\alpha)]_i = \\ &\qquad (1 - \alpha\beta)(X_{ii} - x_i) + \alpha \left[\alpha(\beta X_{ii} + 1 - \beta) + 2(1 - \beta) x_i - (\beta X_{ii} + 1 - \beta)\right]; \end{aligned}$$

(iii) $\begin{pmatrix} 1 & x(\alpha)^T \\ x(\alpha) & X(\alpha) \end{pmatrix}$ is positive semidefinite.

Moreover,

$$f'(0) = \beta(-Q_{\cdot i}^T X_{\cdot i} - c_i x_i) + (1 - \beta)(Q_{\cdot i}^T x + c_i).$$

Proof. It is straightforward to verify the formulas for $x(\alpha)_i$, $[\operatorname{diag}(X(\alpha)) - x(\alpha)]_i$, and $f'(0)$. Now we show that (iii) holds. From the definition of $(x_1(\alpha), X_1(\alpha))$, we see

$$\begin{pmatrix} 1 & x_1(\alpha)^T \\ x_1(\alpha) & X_1(\alpha) \end{pmatrix} = \left(I - \alpha e_i e_i^T\right) \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \left(I - \alpha e_i e_i^T\right) \succeq 0.$$

Furthermore, by using the Schur complement theorem twice, we have

$$\begin{pmatrix} 1 & x_2(\alpha)^T \\ x_2(\alpha) & X_2(\alpha) \end{pmatrix} \succeq 0 \iff \begin{pmatrix} 1 & (x+\alpha e_i)^T \\ x+\alpha e_i & X+\alpha e_i x^T + \alpha x e_i^T + \alpha^2 e_i e_i^T \end{pmatrix} \succeq 0$$
$$\iff (X+\alpha e_i x^T + \alpha x e_i^T + \alpha^2 e_i e_i^T) - (x+\alpha e_i)(x+\alpha e_i)^T \succeq 0$$
$$\iff X - xx^T \succeq 0$$
$$\iff \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0,$$

which is true due to the feasibility of (x, X). Therefore, (iii) follows because $(1, x(\alpha)^T; x(\alpha), X)$ is a convex combination of positive semidefinite matrices. □

The following corollaries are easy to establish.

Corollary 6.2. *Let (x, X) be feasible for (SDP$_0$), and let $\beta = 1$. For a specified index i, $(x(\alpha), X(\alpha))$ is feasible for all $\alpha \in [0, 1]$, and $f'(0) = -[\mathrm{diag}(QX) + c \circ x]_i$.*

Corollary 6.3. *Let (x, X) be feasible for (SDP$_0$) with $x_i < 1$, which guarantees $1+X_{ii}-2x_i > 0$. Also let $\beta = 1/2$. For a specified index i, $(x(\alpha), X(\alpha))$ is feasible for all $\alpha \in \left[0, \frac{1+X_{ii}-2x_i}{1+X_{ii}}\right]$, and $f'(0) = \frac{1}{2}[Qx + c - (\mathrm{diag}(QX) + c \circ x)]_i$.*

We are now ready to prove that every optimal solution (x^*, X^*) of (SDP$_0$) satisfies (12). We need just one additional lemma, whose proof is a straightforward adaptation of the proof of proposition 3.2 in Burer and Vandenbussche (2006):

Lemma 6.4. *Let (x, X) be feasible for (SDP$_0$). Then $x_i = 1$ implies $X_{\cdot i} = x$.*

Theorem 6.5. *Let (x^*, X^*) be optimal for (SDP$_0$). Then (x^*, X^*) satisfies the inequalities (12).*

Proof. We prove the following equivalent statement: suppose feasible (x, X) does not satisfy (12); then (x, X) is not optimal. We break the condition of not satisfying (12) into three subcases: (i) $[\mathrm{diag}(QX) + c \circ x]_i > 0$ for some i; (ii) $[Qx + c - (\mathrm{diag}(QX) + c \circ x)]_i < 0$ for some i and $x_i < 1$; and (iii) $[Qx + c - (\mathrm{diag}(QX) + c \circ x)]_i < 0$ for some i and $x_i = 1$.

In case (i), Corollary 6.2 implies the existence of a feasible path emanating from (x, X) with decreasing objective. Hence, (x, X) is not optimal. Case (ii) follows similarly from Corollary 6.3.

Finally, we show that case (iii) actually cannot occur. Suppose $x_i = 1$. Then by Lemma 6.4,

$$[Qx + c - \mathrm{diag}(QX) - c \circ x]_i = Q_{\cdot i}^T (x - X_{\cdot i}) + c_i(1 - x_i) = 0,$$

which is incompatible with (iii). □

6.2 Proof of equivalence

Note that (SDP$_{12}$) is more constrained than (SDP$_0$). By Lemma 4.1, it suffices to construct a feasible solution to (SDP$_{12}$) based on (x^*, X^*) to establish the equivalence of (SDP$_0$) and (SDP$_{12}$).

We construct the solution for (SDP$_{12}$) by defining

$$
\begin{aligned}
y &:= -(\mathrm{diag}(QX^*) + c \circ x^*) && \text{(13a)}\\
Y &:= yy^T + \epsilon I, && \text{(13b)}\\
w &:= x^* - \mathrm{diag}(X^*),\ W := ww^T && \text{(13c)}\\
M_{xw} &:= wx^{*T},\ M_{yw} := wy^T && \text{(13d)}\\
M &:= \begin{pmatrix} 1 & x^{*T} & y^T & w^T \\ x^* & X^* & {M_{xy}}^T & {M_{xw}}^T \\ y & M_{xy} & Y & {M_{yw}}^T \\ w & M_{xw} & M_{yw} & W \end{pmatrix}, && \text{(13e)}
\end{aligned}
$$

where $\epsilon > 0$ is a sufficiently large constant (more details below). Note that we have not specified M_{xy} yet; we will do so below.

We must check that the solution specified is indeed feasible for (SDP$_{12}$), which requires checking (8b)–(8g). Obviously, (8b) is satisfied by (x^*, X^*). It follows from (12a) and (12b) that (8c) is satisfied by (x^*, X^*, y). The constraint (8e) is satisfied by definition, and Proposition 4.3 illustrates that (8f) is satisfied. It remains to show that (8d) and (8g) hold. These will depend on the choice of ϵ and M_{xy}.

To prove (8d) and (8g), we exhibit an M_{xy} such that $\mathrm{diag}(M_{xy}) = y$ and $M \succeq 0$. We first require the following lemma and proposition:

Lemma 6.6. *For an optimal solution (x^*, X^*) of (SDP$_0$), if ${X_{\cdot i}}^* = x_i^* x^*$, then $y_i(1-x_i^*) = 0$, where y is defined as in (13a).*

Proof. We drop the superscripts $*$ to simplify notation. If $x_i = 1$, then $y_i(1 - x_i) = 0$, and if $x_i = 0$, then $y_i = -(Q_{\cdot i}^T X_{\cdot i} + c_i x_i) = -x_i(Q_{\cdot i}^T x + c_i) = 0$. If $0 < x_i < 1$, we show $y_i = 0$. Let $g_i := Q_{\cdot i}^T x + c_i$. We know $y_i = -x_i g_i \geq 0$, and so $g_i \leq 0$. On the other hand, by (12b), $g_i + y_i = (1 - x_i) g_i \geq 0$, and so $g_i \geq 0$. In total, $g_i = 0$, which ensures $y_i = 0$. □

Proposition 6.7. *Let (x^*, X^*) be an optimal solution of (SDP$_0$), and define y as in (13a). Then there exists $A \in \Re^{n \times n}$ such that*

$$\mathrm{diag}(A(X^* - x^* x^{*T})) = y \circ (e - x^*).$$

Proof. We drop the superscripts $*$ to simplify notation. We show equivalently that there exists a solution A to the system of equations

$$A_{i\cdot}(X - xx^T)_{\cdot i} = y_i(1 - x_i) \quad \forall\, i = 1, \ldots, n.$$

Note that the n equations just listed are separable; so we consider each i separately. If $(X - xx^T)_{\cdot i} \neq 0$, it is obvious that there exists a solution $A_{i\cdot}$; just take $A_{i\cdot}$ equal to $\frac{y_i(1-x_i)}{\|(X-xx^T)_{\cdot i}\|^2}(X - xx^T)^T_{\cdot i}$. On the other hand, if $(X - xx^T)_{\cdot i} = 0$, i.e., $X_{\cdot i} = x_i x$, then we know by Lemma 6.6 that $y_i(1 - x_i) = 0$ and thus $A_{i\cdot}$ can be any vector. □

We define

$$M_{xy} := yx^{*T} + A(X^* - x^* x^{*T}),$$

where A is any matrix as in Proposition 6.7. Then $\operatorname{diag}(M_{xy}) = y \circ x^* + y \circ (e - x^*) = y$, which ensures that (8d) is satisfied. Finally, it remains to show that (8g) holds, i.e., $M \succeq 0$, for this choice of M_{xy}.

From this point, we drop the superscripts $*$ to simplify notation. Note that

$$M = \begin{pmatrix} 1 \\ x \\ y \\ w \end{pmatrix} \begin{pmatrix} 1 \\ x \\ y \\ w \end{pmatrix}^T + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & X - xx^T & (M_{xy} - yx^T)^T & 0 \\ 0 & M_{xy} - yx^T & \epsilon I & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

and so it suffices to show

$$\begin{pmatrix} X - xx^T & (M_{xy} - yx^T)^T \\ M_{xy} - yx^T & \epsilon I \end{pmatrix} = \begin{pmatrix} X - xx^T & \big(A(X - xx^T)\big)^T \\ A(X - xx^T) & \epsilon I \end{pmatrix} \succeq 0.$$

By the Schur complement theorem, this holds if and only if

$$(X - xx^T) - \epsilon^{-1}(X - xx^T)A^T A(X - xx^T) \succeq 0. \tag{14}$$

Consider the following straightforward lemma:

Lemma 6.8. *Suppose $R, S \succeq 0$. Then there exists $\delta > 0$ small enough such that $R - \delta S \succeq 0$ if and only if* $\operatorname{Null}(R) \subseteq \operatorname{Null}(S)$.

Because the null space of $X - xx^T$ is contained in the null space of $(X - xx^T)A^T A(X - xx^T)$, the lemma implies the existence of $\epsilon > 0$ large enough so that (14) holds. Taking such ϵ, we conclude that (8g) is satisfied.

Overall, we have shown that definition (13)—along with the definitions of M_{xy} and ϵ—is feasible for (SDP$_{12}$), which means (SDP$_0$) and (SDP$_{12}$) are equivalent by Lemma 4.1.

7 Conclusion

In this paper, we have introduced new semidefinite relaxations of Box QP: (SDP$_{12}$) and (SDP$_2$). (SDP$_{12}$) is based on relaxing both the first- and second-order necessary optimality conditions; (SDP$_2$) is similar to (SDP$_{12}$) except that it only contains the relaxations of second-order optimality conditions. We have compared these two relaxations with a basic semidefinite relaxation (SDP$_0$) and established the theoretical results that all of the three relaxations achieve the same optimal value. We also empirically compared (SDP$_0$) and (SDP$_2$) and demonstrated that the tailored version of (SDP$_2$) on subdivided boxes is significantly stronger than that of (SDP$_0$) in the context of branch-and-bound, which indicates that incorporation of second-order information in SDP relaxations can help globally solving the problem.

How to relax the standard second-order necessary optimality conditions, however, is not trivial as the conditions involve knowledge of the active/inactive constraint set. We have derived equivalent second-order conditions that do not require explicit knowledge of the inactive constraints set, which is crucial to obtain the semidefinite relaxations. In the future, this technique may be extended to other problems, e.g., quadratic programming over the simplex, and possibly lead to stronger SDP relaxations.

References

K. M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. Manuscript, University of Iowa, May 2007.

S. Burer and D. Vandenbussche. Globally solving box-constrained nonconvex quadratic programs with semidefinite-based finite branch-and-bound. Manuscript, Department of Management Sciences, University of Iowa, Iowa City, IA, USA, November 2006. Revised June 2006 and July 2006. To appear in *Computational Optimization and Applications*.

P. De Angelis, P. Pardalos, and G. Toraldo. Quadratic programming with box constraints. In I. M. Bomze, T. Csendes, R. Horst, and P. Pardalos, editors, *Developments in Global Optimization*, pages 73–94. Kluwer Academic Publishers, 1997.

N. I. M. Gould and P. L. Toint. Numerical methods for large-scale non-convex quadratic programming. In *Trends in industrial and applied mathematics (Amritsar, 2001)*, volume 72 of *Appl. Optim.*, pages 149–179. Kluwer Acad. Publ., Dordrecht, 2002.

R. Horst and H. Tuy. *Global optimization*. Springer-Verlag, Berlin, second edition, 1993. Deterministic approaches.

J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL `http://control.ee.ethz.ch/ joloef/yalmip.php`.

Y. Nesterov. Global quadratic optimization via conic relaxation. In R. Saigal, L. Vandenberghe, and H. Wolkowicz, editors, *Handbook of Semidefinite Programming*. Kluwer Academic Publishers, 2000.

P. Pardalos. Global optimization algorithms for linearly constrained indefinite quadratic problems. *Computers and Mathematics with Applications*, 21:87–97, 1991.

P. M. Pardalos and S. A. Vavasis. Quadratic programming with one negative eigenvalue is NP-hard. *J. Global Optim.*, 1(1):15–22, 1991.

H. D. Sherali and W. P. Adams. *A Reformulation-Linearization Technique (RLT) for Solving Discrete and Continuous Nonconvex Problems*. Kluwer, 1997.

N. Shor. Quadratic optimization problems. *Soviet Journal of Computer and Systems Science*, 25:1–11, 1987. Originally published in *Tekhnicheskaya Kibernetika*, 1:128–139, 1987.

J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, 11/12(1-4):625–653, 1999.

D. Vandenbussche and G. Nemhauser. A polyhedral study of nonconvex quadratic programs with box constraints. *Mathematical Programming*, 102(3):531–557, 2005a.

D. Vandenbussche and G. Nemhauser. A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. *Mathematical Programming*, 102(3):559–575, 2005b.

Y. Ye. Approximating quadratic programming with bound and quadratic constraints. *Math. Program.*, 84(2, Ser. A):219–226, 1999.