Relaxing nonconvex quadratic functions by multiple adaptive diagonal perturbations

Hongbo Dong*

Current version: May 12, 2016
Original version: March 11, 2014

Abstract

The current bottleneck of globally solving mixed-integer (nonconvex) quadratically constrained problems (MIQCPs) is still to construct strong but computationally cheap convex relaxations, especially when dense quadratic functions are present. We propose a cutting-surface method based on multiple diagonal perturbations to derive convex quadratic relaxations for nonconvex quadratic problems with separable constraints. Our relaxations can also be realized as outer-approximations of a semi-infinite convex formulation of a semidefinite relaxation proposed by Buchheim and Wiegele. The corresponding separation problem is a highly structured semidefinite program (SDP) with a convex but nonsmooth objective function. We propose to solve this separation problem with a specialized barrier coordinate minimization algorithm. Numerical experiments on randomly generated instances show that our approach is very promising. We also discuss how to apply our method to more general cases of MIQCPs.

Keywords: Quadratic Programming; Convex Relaxation; Cutting-surface Algorithm;

Mathematics Subject Classification: 90C10, 90C20, 90C22, 90C25, 90C26, 90C30

1 Introduction

In this paper we focus on constructing convex quadratic relaxations for the following class of problems,

$$\min_{x\in\mathcal{R}^n} \; x^TQx + q^Tx \quad s.t. \;\; x_i \in S_i, \; \forall i \in [n] := \{1,..,n\}, \tag{P}$$

where $\mathcal{R}^n$ is the Euclidean space of dimension n and Q is indefinite. For each i, S_i is the union of finitely many closed intervals in $\mathcal{R}$. We also assume that S_i is bounded. With one

*Department of Mathematics, Washington State University, Pullman, WA, 99164, USA

additional variable v, (P) can be equivalently written as the following linear optimization over a convex hull

$$\min_{v,x\in\mathcal{R}^n} \quad v + q^T x \quad s.t. \quad (v,x) \in \mathcal{H}\left(Q; \{S_i\}_{i=1}^n\right),$$

where

$$\mathcal{H}\left(Q; \{S_i\}_{i=1}^n\right) := \mathbf{conv}\left\{(v,x) \middle| v \geq x^T Q x, \quad x_i \in S_i, \forall i \in [n]\right\}.$$

In this paper we discuss methods of generating convex (but nonlinear) valid constraints for $\mathcal{H}\left(Q; \{S_i\}_{i=1}^n\right)$. Such methods can be applied to more general problem classes. For example, consider a mixed-integer quadratically constrained programming (MIQCP) problem,

$$\min_{x\in\mathcal{R}^n} \left\{ c^T x \middle| \begin{array}{rl} x^T Q^{(j)} x + c^{(j)T} x \leq d^{(j)}, & 1 \leq j \leq m \\ x_i \in S_i, & 1 \leq i \leq n \end{array} \right\}. \tag{1}$$

Given nonnegative weights $\alpha_j \geq 0$ for $j = 1, ..., m$, the following condition holds,

$$\left(\sum_{j=1}^m \alpha_j \big(d^{(j)} - c^{(j)T} x\big), x \right) \in \mathcal{H}\big(\sum_{j=1}^m \alpha_j Q^{(j)}; S_1, ..., S_n \big).$$

Therefore methods in this paper can be used to generate (convex) valid constraints for general MIQCPs. There are many studies on constructing valid inequalities for MIQCPs in the lifted variable space (x, xx^T) (see for example [BS12] and [DK15] for recent surveys). We remark that any valid linear inequality in the lifted variable space corresponds to a valid (nonconvex) quadratic constraint in the original space, and hence can be used in the previous aggregation. Effective generation of such an aggregation is beyond the scope of this paper. There are also some applications where solving (P) globally is of interests, including the most well-studied case when all variables are binary. In this case, (P) is equivalent to the max-cut problem in combinatorial optimization, which finds applications in Ising models in statistical physics [LJRR04] and statistical learning [WJ08].

The idea of constructing convex relaxations by diagonal perturbation is not new. The so-called α-BB method is a general method to convexify nonlinear functions by diagonally perturbing their Hessian matrices. See [SWMF12] and references therein. For the important case that $S_i = \{0, 1\}$ for all i, or equivalently, the Max-Cut problem, the problem (P) can be equivalently reformulated as a binary convex quadratic problem by some diagonal perturbations to Q. [RRW10] provides a review of solution approaches for this problem and proposes to incorporate some strong SDP-based relaxations by solving them using a bundle method. Another relevant line of research [FG07, GL10, ZSL14, DL13] focused on globally solving convex quadratic programming with binary indicator variables, when combined with the perspective constraints, diagonal perturbations are also shown to be crucial in this works. Recently, the authors of [DCL15] showed that these ideas are tightly related to the minimax concave penalty (MCP) [Zha10] in the statistical literature.

We remark that in many of these approaches, diagonal perturbations are determined statically using partial problem information, e.g., Hessian matrices and possibly some equality constraints in the original problem formulation. Although in [BEP09] and [ZSL14], the authors determine a single diagonal perturbation by solving a complicated semidefinite program that exploits full problem formulation, these approaches are computationally very costly and only feasible at the root node of a branch-and-bound tree. After branching, the problem structure may change and the computed diagonal perturbation may not be useful anymore. On the contrary, our proposed approach in this paper generates cutting surfaces to separate current relaxed solutions. Furthermore, our separation routine is computationally much cheaper than the SDPs in [BEP09] and [ZSL14], therefore it is flexible enough to be incorporated in a branch-and-bound algorithm to strengthen convex relaxations "on-the-fly".

Many current general purpose global solvers for mixed-integer nonlinear programs are based on the αBB and/or the lifting methodology. For the quadratic case, whenever term $x_i x_j$ is present, one introduces the lifted variable X_{ij} with the following RLT constraints

$$y_{ij}^-(x) \leq X_{ij} \leq y_{ij}^+(x),$$

where $L_i \leq x_i \leq R_i$ and

$$\begin{aligned} y_{ij}^-(x) &:= \max\{R_i x_j + R_j x_i - R_i R_j, L_i x_j + L_j x_i - L_i L_j\}, \\ y_{ij}^+(x) &:= \min\{L_i x_j + R_j x_i - L_i R_j, R_i x_j + L_j x_i - R_i L_j\}. \end{aligned}$$

We remark that when dense quadratic functions are present, this lifting approach generates a lot of additional variables, which may significantly slow down the whole branch-and-bound algorithm. When the full lifted matrix X is introduced to represent the rank one matrix xx^T, the SDP constraint $X - xx^T \succeq 0$ (i.e., $X - xx^T$ is positive semidefinite) is a valid convex constraint. Anstreicher [Ans09] shows that by combining the SDP constraint with RLT inequalities, one obtains much stronger relaxations than each of these methods alone, although the resulting relaxation is generally considered to be computationally expensive. To overcome this difficulty, [SBL11] proposes an illuminating approach which generates convex quadratic cutting surfaces by projecting down the semidefinite and the RLT constraints onto the original variable space. Our work is partially motivated by their work.

Following a different strategy, Burer and Chen [BC12] proposed a competitive branch-and-bound algorithm to solve mixed-binary quadratically constrained programs in certain special forms. Their algorithm is based on an alternating projection augmented Lagrangian algorithm to (approximately) compute the doubly nonnegative relaxations for completely positive reformulations of the original problem. However, this approach is relatively inflexible to represent arbitrary nonconvex structure in S_i (for example, when S_i comprises of integers in a bounded region, or is a union of disjoint intervals), and the bounding algorithm is sensitive to parameter tuning.

Our work is related to the recent work [BW13] by Buchheim and Wiegele. In [BW13], a branch-and-bound algorithm Q-MIST is designed to solve (P) globally, based on solving a "diagonal" SDP relaxation with interior point methods. Q-MIST is shown to compare favorably to Couenne [BLL+09], a general purpose global solver. We establish the connections between our approach and [BW13] in Section 2, and compare the numerical performance in Section 5.2. Another related work is [BT13]. Although their idea is quite different, an SDP that is similar to our separation problem is considered. After submission of our first manuscript, the authors generalized our coordinate minimization algorithm in their subsequent manuscript [BT15].

The full paper is organized as follows. In Section 2 we derive cutting surfaces for (P) using diagonal perturbations. Adding all possible cutting surfaces yields a semi-infinite convex relaxation for (P), which we show is equivalent to the Buchheim-Wiegele SDP relaxation. This semi-infinite formulation motivates us to construct convex relaxations of (P) by iterative separation and adding cutting surfaces. In Section 3 we provide some results on the finite attainment of the corresponding separation problem, as well as subdifferential properties of a nonsmooth problem that is equivalent to the semi-infinite formulation. These results motivate us to solve regularized separation problems in our cutting-surface algorithm. Section 4 is devoted to a specialized primal-barrier coordinate minimization algorithm for regularized separation. Finally, Section 5 includes our numerical results.

2 Convex cutting surfaces by diagonal perturbations

In this section we propose to construct convex quadratic valid constraints for the set $\mathcal{H}(Q; \{S_i\}_{i=1}^n)$ based on diagonal perturbations of matrix Q and convex hulls of the following two dimensional sets D_i,

$$D_i := \{(x_i, x_i^2) | x_i \in S_i\}, \quad \forall i \in [n].$$

Buchheim and Wiegele [BW13] proposed a semidefinite relaxation for (P) by exploiting this convex hull. We will show in this section that our approach is closely related to their work. We denote $L_i := \min\{x | x \in S_i\}$ and $R_i := \max\{x | x \in S_i\}$. Further let us denote $\ell_i(\cdot)$ the lower convex envelope of D_i, i.e., the largest convex function defined on $[L_i, R_i]$, such that $\ell_i(x) \leq x^2, \forall x \in S_i$, and $u_i(\cdot)$ the upper concave envelope, i.e., the smallest concave function defined on $[L_i, R_i]$ such that $u_i(x) \geq x^2, \forall x \in S_i$. We have the following characterizations.

Proposition 1. *Let S_i, D_i, L_i, R_i, $\ell_i(\cdot)$ and $u_i(\cdot)$ be defined as above, then*

1. $\mathbf{conv}(D_i) = \{(x_i, y_i) | \ell_i(x_i) \leq y_i \leq u_i(x_i)\}$;
2. $x_i^2 \leq \ell_i(x_i) \leq u_i(x_i)$, $\forall x_i \in [L_i, R_i]$;

Proof. We prove (1) first. Since $\mathbf{conv}(D_i)$ is the smallest convex set that contains D_i, $\mathbf{conv}(D_i) \subseteq \{(x_i, y_i) | \ell_i(x_i) \le y_i \le u_i(x_i)\}$. To show the opposite inclusion, we assume otherwise that $\ell_i(\bar{x}_i) \le \bar{y}_i \le u_i(\bar{x}_i)$ and $(\bar{x}_i, \bar{y}_i) \notin \mathbf{conv}(D_i)$. Since D_i is compact, there exists scalars a, b and c such that at least one of a and b is non-zero, and

$$a\bar{x}_i + b\bar{y}_i < c, \ \textit{and} \ ax_i + bx_i^2 \ge c, \forall x_i \in S_i.$$

Note the case of $b = 0$ implies $\bar{x}_i \notin [L_i, R_i]$, which contradicts with the implicit assumption that $\ell_i(\bar{x}_i)$ and $u_i(\bar{x}_i)$ are well-defined. If $b > 0$, we rescale such that $b = 1$. Then $x_i^2 \ge -ax_i + c$, $\forall x_i \in S_i$ and $\bar{y}_i < -a\bar{x}_i + c$. One can then verify that $\hat{\ell}_i(x_i) := \max(\ell_i(x_i), -ax_i + c)$ is a larger convex function such that $\hat{\ell}_i(x_i) \le x_i^2$, $\forall x_i \in S_i$, which contradicts with the assumption that $\ell_i(\cdot)$ is the lower convex envelope. The case of $b < 0$ is similar.

To prove (2), note that function x_i^2 is a convex function. By the definition of $\ell_i(\cdot)$, we must have $\max\{x_i^2, \ell_i(x_i)\} \le \ell_i(x_i), \forall x_i \in [L_i, R_i]$, i.e., $x_i^2 \le \ell_i(x_i), \forall x_i \in [L_i, R_i]$. Further $u_i(x_i) - \ell_i(x_i)$ is a concave function such that $u_i(x_i) - \ell_i(x_i) \ge 0$ for all $x_i \in S_i$, including L_i and R_i. Therefore we must have $u_i(x_i) - \ell_i(x_i) \ge 0$ for all $x_i \in [L_i, R_i]$. ☐

As explained in [BW13], it is usually possible to fully characterize $\ell_i(\cdot)$ and $u_i(\cdot)$. In fact, with bounded S_i, $u_i(x_i)$ is always the straight line connecting the two end-points(L_i, L_i^2) and (R_i, R_i^2). If $x_i \in S_i$, $\ell(x_i) = x_i^2$. Otherwise, let p_i and q_i be two points in S_i closest to x_i such that $p_i < x_i < q_i$, then $\ell_i(x_i) = (q_i + p_i)x_i - p_i q_i$.

In this paper we study convex valid constraints obtained by perturbing the quadratic form $x^T Q x$ with separable terms. Note that our construction may seem similar to the QCR method [BEP09], however we aim to design a more flexible cutting-surface approach to be possibly combined with techniques such as bound tightening and other cutting planes in a branch-and-bound framework. Given a vector $d \in \mathcal{R}^n$, consider the inequality

$$v \ge x^T Q x + \sum_{i=1}^{n} \left(d_i x_i^2 - d_i y_i(x_i)\right) = x^T (Q + \mathbf{diag}(d)) x - \sum_{i=1}^{n} d_i y_i(x_i), \tag{2}$$

where $y_i(x_i)$ is some univariate function of x_i, whose form possibly depends on the sign of d_i. We remark that $y_i(x_i)$ can be thought as a "compensating term" for the perturbation x_i^2. Now we consider conditions under which (2) is valid and convex. First of all, it is valid if $d_i(x_i^2 - y_i(x_i)) \le 0, \forall i$. That is, $y_i(x_i) \ge x_i^2$ when $d_i > 0$ and $y_i(x_i) \le x_i^2$ when $d_i < 0$. Secondly, to guarantee the overall convexity, in addition to $Q + \mathbf{diag}(d) \succeq 0$, we require $y_i(x_i)$ to be concave when $d_i > 0$, and convex when $d_i < 0$. Finally, since it is preferable to have $d_i(x_i^2 - y_i(x_i))$ as large (close to 0) as possible, natural choices of $y_i(x_i)$ are the lower and upper envelopes of D_i, i.e.,

$$y_i(x_i) = \begin{cases} \ell_i(x_i), & d_i < 0; \\ u_i(x_i), & d_i > 0. \end{cases}$$

Therefore the following inequality is a convex valid constraint for $\mathcal{H}(Q; \{S_i\}_{i=1}^n)$, given any vector d such that $Q + \mathbf{diag}(d) \succeq 0$,

$$v \geq x^T Q x + \sum_{i:d_i<0} d_i(x_i^2 - \ell_i(x_i)) + \sum_{i:d_i>0} d_i(x_i^2 - u_i(x_i)). \tag{CUT}$$

In the rest of the paper we say a diagonal perturbation vector d is *admissible* if

$$d \in \mathcal{D} := \{d | Q + \mathbf{diag}(d) \succeq 0\}.$$

With all such admissible vectors, a convex relaxation for (P) is the following semi-infinite convex program (SICP):

$$\begin{aligned} \mu_{SICP} = \min_{v,x} \quad & v + q^T x \\ s.t., \quad & v \geq x^T(Q + \mathbf{diag}(d))x - \sum_{i:d_i>0} d_i u_i(x_i) - \sum_{i:d_i<0} d_i \ell_i(x_i), \\ & \qquad \forall d \in \mathcal{D}, \\ & L_i \leq x_i \leq R_i, \quad \forall i. \end{aligned} \tag{SICP}$$

We show that in fact, this semi-infinite relaxation is equivalent to a semidefinite relaxation proposed by Buchheim and Wiegele [BW13],

$$\begin{aligned} \mu_{BW} := \min_{x,X} \quad & \langle Q, X\rangle + q^T x \\ s.t. \quad & X - xx^T \succeq 0, \\ & \ell_i(x_i) \leq X_{ii} \leq u_i(x_i), \quad L_i \leq x_i \leq R_i, \forall i. \end{aligned} \tag{BW}$$

We prove the equivalence in the following theorem by the strong duality of (BW).

Theorem 1. *Let μ_{SICP} and μ_{BW} be the optimal values in (SICP) and (BW), respectively. We have $\mu_{BW} = \mu_{SICP}$.*

Proof. First we show $\mu_{BW} \geq \mu_{SICP}$. Note that (BW) is equivalent to

$$\begin{aligned} \min_{x,v} \quad & v + q^T x \\ s.t. \quad & L_i \leq x_i \leq R_i, \\ & v = \min_X \left\{ \langle Q, X\rangle \;\middle|\; X \succeq xx^T, \ell_i(x_i) \leq X_{ii} \leq u_i(x_i) \right\}. \end{aligned} \tag{3}$$

It suffices to show that for any (x, X) feasible in (BW), and $d \in \mathcal{D}$,

$$\langle Q, X\rangle \geq x^T Q x + \sum_{i:d_i<0} d_i(x_i^2 - \ell_i(x_i)) + \sum_{i:d_i>0} d_i(x_i^2 - u_i(x_i)).$$

By re-arranging terms, this inequality can be written as

$$\langle Q + \mathbf{diag}(d), X - xx^T\rangle - \sum_{i:d_i<0} d_i(X_{ii} - \ell_i(x_i)) - \sum_{i:d_i>0} d_i(X_{ii} - u_i(x_i)) \geq 0,$$

which is valid by the self-duality of the positive semidefinite cone, and the feasibility of (x, X).

We show the other direction, $\mu_{BW} \leq \mu_{SICP}$, by the strong duality and dual attainment of (BW). We introduce Lagrange multipliers $d_i^- \geq 0$ for the constraint $X_{ii} - \ell_i(x_i) \geq 0$, and $d^+ \geq 0$ for $u_i(x_i) - X_{ii} \geq 0$. Then dual problem is

$$\max_{d^+, d^- \in \mathcal{R}^n_+} \left\{ \begin{array}{c} \min_{x,X} \langle Q, X\rangle + q^T x - \sum_i d_i^-(X_{ii} - \ell_i(x_i)) - \sum_i d_i^+(u_i(x_i) - X_{ii}) \\ s.t. X - xx^T \succeq 0, \quad L_i \leq x_i \leq R_i, \quad \forall i. \end{array} \right\}.$$

As (BW) satisfies the Slater's condition (i.e., one can find $\bar{X}$ and $\bar{x}$ such that $\bar{X} \succ \bar{x}\bar{x}^T$ and $\ell_i(\bar{x}_i) < \bar{x}_i < u_i(\bar{x}_i)$, unless $\mathbf{conv}(D_i)$ has an empty interior, which can only happen when $\ell_i(\cdot)$ and $u_i(\cdot)$ are linear functions), strong duality holds and the dual optimal value is attained. For each i, $u_i(x_i) \geq \ell_i(x_i)$ for all $x_i \in [L_i, R_i]$. We can assume that either $d_i^+ = 0$ or $d_i^- = 0$, but not both. Let $d := d^+ - d^-$, we can further restrict d such that $Q + \mathbf{diag}(d) \succeq 0$ as otherwise the inner minimization is unbounded below. Finally it suffices to assume $X - xx^T = 0$. The dual problem can be simplified as

$$\mu_{BW} = \max_{d:Q+\mathbf{diag}(d)\succeq 0} \left\{ \begin{array}{c} \min_x x^T(Q + \mathbf{diag}(d))x + q^T x - \sum_{i:d_i<0} d_i \ell_i(x_i) - \sum_{i:d_i>0} d_i u_i(x_i) \\ s.t. \quad L_i \leq x_i \leq R_i, \quad \forall i. \end{array} \right\}.$$

The dual attainment implies there exists d^*, $Q + \mathbf{diag}(d^*) \succeq 0$, such that

$$\begin{aligned} \mu_{BW} := \min_x \quad & x^T(Q + \mathbf{diag}(d^*))x + q^T x - \sum_{i:d_i^*<0} d_i^* \ell_i(x_i) - \sum_{i:d_i^*>0} d_i^* u_i(x_i) \\ s.t. \quad & L_i \leq x_i \leq R_i. \end{aligned} \tag{4}$$

Therefore $\mu_{BW} \leq \mu_{SICP}$, which completes our proof. □

The semi-infinite formulation (SICP), as well as the proof of Theorem 1, motivates us to construct convex relaxations for (P) in the original variable space using a finite number of valid constraints (CUT). Indeed, as shown in (4), only one single diagonal perturbation $d \in \mathcal{D}$ is needed to obtain the tightest lower bound μ_{BW}, although finding the correct d amounts to solve the semidefinite relaxation (BW). A natural idea is to use a cutting-surface procedure, where one iteratively adds a constraint (CUT) to cut off a previous optimal solution. However, from a computational point of view, it is cumbersome to carry the information of functions $\ell_i(\cdot)$ and $u_i(\cdot)$ in every nonlinear cut. We use a lifting method to overcome this difficulty.

Note that (SICP) can be equivalently written as the following infinite dimensional problem by adding a variable $y_i^{(d)}$ for *each* admissible d and index i,

$$
\begin{aligned}
\mu_{SICP} = \min_{\substack{v, x \in \mathcal{R}^n, \\ y^{(d)} \in \mathcal{R}^n, \forall d \in \mathcal{D}}} \quad & v + q^T x \\
s.t., \quad & v \geq x^T (Q + \mathbf{diag}(d)) x - \sum_i d_i y_i^{(d)}, \qquad \forall d \in \mathcal{D},\ \forall i, \\
& \ell_i(x_i) \leq y_i^{(d)} \leq u_i(x_i), \qquad \forall d \in \mathcal{D},\ \forall i \\
& L_i \leq x_i \leq R_i, \quad \forall i.
\end{aligned}
$$

As for all $d \in \mathcal{D}$, $y_i^{(d)}$ is used as a surrogate of x_i^2, which should be independent of d. By "linking" all such surrogates, we obtain the following semi-infinite convex relaxation of (P),

$$
\begin{aligned}
\min_{v \in \mathcal{R}, x, y \in \mathcal{R}^n} \quad & v + q^T x \\
s.t., \quad & v \geq x^T (Q + \mathbf{diag}(d)) x - d^T y, \quad \forall d \in \mathcal{D}, \\
& \ell_i(x_i) \leq y_i \leq u_i(x_i),\ L_i \leq x_i \leq R_i, \qquad \forall i.
\end{aligned}
\tag{SICP+}
$$

It is straightforward to show that (SICP+) is also a convex relaxation of (P). Indeed, for any x feasible in (P) and $y_i = x_i^2$, $x^T(Q + \mathbf{diag}(d))x - \sum_i d_i y_i = x^T Q x$. Further, (SICP+) provides the same lower bound as (SICP). The proof is quite similar to that of Theorem 1, and we only describe the key idea below.

In fact, (SICP+) can be derived from the following equivalent formulation of (BW) with redundant variables y_i and constraints $y_i = X_{ii}$ for all i.

$$
\begin{aligned}
\mu_{BW} = \min_{x, X, y} \quad & \langle Q, X \rangle + q^T x \\
s.t. \quad & \ell_i(x_i) \leq y_i \leq u_i(x_i), \quad L_i \leq x_i \leq R_i, \forall i, \\
& \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0, \\
& y_i = X_{ii}, \forall i.
\end{aligned}
\tag{5}
$$

By dualizing each of the equality constraints $y_i = X_{ii}$ with a dual variable d_i, and following the same procedure as in the proof of Theorem 1, one obtains (SICP+). By strong duality, (SICP+) provides the same bound as (BW).

We are now ready to present our cutting-surface algorithm by replacing $\mathcal{D}$ in (SICP+) with a dynamically maintained finite set. In the k-th iteration, a convex relaxation of (P) is

$$\mu^{(k)} := \min_{v,x} \quad v + q^T x$$
$$s.t. \quad v \geq x^T \left(Q + \mathbf{diag}(d)\right) x - \sum_{i=1}^{n} d_i y_i, \quad \forall d \in \mathcal{D}^{(k)} \tag{DiagR+}$$
$$\ell_i(x_i) \leq y_i \leq u_i(x_i), \ \forall i$$
$$L_i \leq x_i \leq R_i, \quad \forall i.$$

where $\mathcal{D}^{(k)} \subseteq \mathcal{D}$ is a finite set. Let $(\bar{x}, \bar{y}, \bar{v})$ be an optimal solution to (DiagR+). Then in a separation step, one attempts to find a new admissible diagonal perturbation d^{new} whose associated cutting surface "cuts off" $(\bar{x}, \bar{y}, \bar{v})$. A high level description is provided in Algorithm 1.

Algorithm 1: A cutting-surface algorithm to derive a convex relaxation of (P)

Data: $Q \in \mathcal{S}^n$, $q \in \mathcal{R}^n$, and black box routines to evaluate $\ell_i(\cdot)$ and $u_i(\cdot)$;
Result: A convex quadratic relaxation of (P).

```
D^(1) = {λ · e}, where e is the all-one vector and λ > | min(0, λ_min(Q))|;
for k = 1 to maxIter do
    Solve (DiagR+); Let (x̄, ȳ, v̄) be an optimal solution;
    Separation: Find an admissible diagonal perturbation vector d^new;
    if (CUT) with d = d^new cuts off (x̄, v̄) then
        D^(k+1) ← D^(k) ∪ {d^new};
    else
        Terminate;
    end
end
```

Remark 1. In fact, when a finite set $\mathcal{D}^{(k)}$ is used to replace $\mathcal{D}$ in both (SICP+) and (SICP), the partially lifted formulation (SICP+) is no weaker, and can be strictly stronger than (SICP). This provides another reason of using (SICP+) in our cutting-surface algorithm. We illustrate this by the following small numerical example.

Example 1. *Consider the integer nonconvex quadratic minimization where*

$$Q = \begin{bmatrix} -3 & 3 \\ 3 & 4 \end{bmatrix}, \ q = [-2, 1]^T, \ S_1 = S_2 = \{-1, 0, 1\}.$$

The optimal value of (P) is -8 and is obtained at $[1, -1]^T$. The semidefinite relaxation (BW) is tight, i.e., $\mu_{BW} = -8$. Consider two diagonal perturbation vectors $d^{(1)} = [4, 10]^T$ and $d^{(2)} = [10, -1]^T$. If one replaces $\mathcal{D}$ in (SICP) with $\{d^{(1)}, d^{(2)}\}$, one obtains a lower bound value -10.25. However, if $\mathcal{D}$ in (SICP+) is replaced by $\{d^{(1)}, d^{(2)}\}$, the lower bound is improved to -9.6458.

3 Subdifferential properties and regularized separation

Provided $(\bar{x}, \bar{y}, \bar{v})$ feasible in (DiagR+) for some finite set $\mathcal{D}^{(k)} \subseteq \mathcal{D}$, the separation problem aiming to find the most violated cutting surface is

$$\sup_{d} \sum_{i=1}^{n} \left(\bar{x}_i^2 - \bar{y}_i\right) d_i \quad s.t. \quad Q + \mathbf{diag}(d) \succeq 0. \tag{SEP}$$

Note that solving (SEP) is equivalent to evaluating a function $h(x, y)$ at $(\bar{x}, \bar{y})$, where $h(x, y)$ is defined in the following non-smooth formulation of (SICP+),

$$\min_{x,y} \left\{ h(x,y) + q^T x \middle| \begin{array}{c} \ell_i(x_i) \leq y_i \leq u_i(x_i), \ \forall i \\ L_i \leq x_i \leq R_i, \quad \forall i \end{array} \right\},$$

where $h(x, y) := \sup_{d \in \mathcal{D}} \left\{ x^T \big(Q + \mathbf{diag}(d)\big) x - d^T y \right\}$. In this section we provide some results on the finite attainment of (SEP) and the subdifferential of $h(x, y)$.

As $(\bar{x}, \bar{y}, \bar{v})$ is feasible in (DiagR+), by Proposition 1 we have $\bar{x}_i^2 \leq \bar{y}_i$ for all i. We show that if $\bar{x}_i^2 < \bar{y}_i$ for all i, then (SEP) is finitely attained, and $h(x, y)$ is subdifferentiable at $(\bar{x}, \bar{y})$. Unfortunately we show by a small example that if $\bar{x}_i^2 = \bar{y}_i$ for some i, both properties may fail. We will need the following theorem in convex analysis to calculate the subdifferential of a supremum function.

Theorem 2 (Theorem 4.4.2 [HUL01])**.** *For some compact set $Y \subseteq \mathcal{R}^p$, let $g : \mathcal{R}^q \times Y \mapsto \mathcal{R}$ be a function such that $g(\cdot, d)$ is a convex function from $\mathcal{R}^q$ to $\mathcal{R}$ for each $d \in Y$, and $g(x, \cdot)$ is upper semi-continuous for each $x \in \mathcal{R}^n$. Further let $f(x) = \sup_{d \in Y} g(x; d)$ and assume that $f(x) < +\infty$ for all x. Then*

$$\partial f(x) = \mathbf{conv}\{\cup_d \partial g(\cdot; d) \mid d \in Y, \ s.t. \ g(x; d) = f(x)\}.$$

Proposition 2. *Let $\bar{x}$ and $\bar{y}$ be vectors in $\mathcal{R}^n$ that satisfy the last two sets of constraints in (DiagR+). Let $\mathcal{I} := \{i | \bar{x}_i^2 < \bar{y}_i\}$, and $\mathcal{J} := [n] \setminus \mathcal{I}$ be its complement. If $\mathcal{I} = \emptyset$, then*

1. *the supremum of (SEP) is attained at some finite point d^*, and $h(\bar{x}, \bar{y}) = \bar{x}^T \big(Q + \mathbf{diag}(d^*)\big) \bar{x} - d^{*T} \bar{y}$;*

2. *$h(x, y)$ is subdifferentiable at $(\bar{x}, \bar{y})$, with subdifferential*

$$\partial h(x, y)|_{(\bar{x}, \bar{y})} = \mathbf{conv} \left\{ [2(Q + \mathbf{diag}(d^*))\bar{x}, -d^*]^T \ \middle| \ \forall d^* \ optimal \ in \ (SEP) \right\}.$$

Proof. We first show that if $\mathcal{I} = \emptyset$, the optimal solution to (SEP) is finitely attained. Let $\bar{w}_i = \bar{y}_i - \bar{x}_i^2 > 0$ for each i. A feasible solution to (SEP) is $-\lambda_{min}(Q)e$, where $\lambda_{min}(Q)$ is the minimum eigenvalue of Q, and e is a vector of all ones. If we denote the corresponding objective value as $\hat{\nu} := \lambda_{min}(Q)\sum_i \bar{w}_i$, then any optimal solution to (P) must be in the following compact polytope

$$\{d \in \mathcal{R}^n | \bar{w}^T d \leq -\hat{\nu}, d_i \geq -Q_{ii}, \forall i\}. \tag{6}$$

In fact, an upper bound of d_i can be explicitly obtained as $\frac{-\lambda_{min}(Q)\sum_i \bar{w}_i + \sum_{j\neq i} \bar{w}_j Q_{jj}}{\bar{w}_i}$. Therefore the supremum of (SEP) is finitely attained.

We now consider the subdifferential of $h(x, y)$ at $(\bar{x}, \bar{y})$. To apply Theorem 2, one needs to replace the unbounded set of admissible diagonal perturbations $\{d|Q + \mathbf{diag}(d) \succeq 0\}$ with a compact set. Since $\mathcal{I} = \emptyset$, one can find an open neighborhood of $(\bar{x}, \bar{y})$, denoted by B, and $\epsilon > 0$ such that for any $(x, y) \in B$, $x_i^2 - y_i \leq -\epsilon < 0$ for all i. We further assume without loss of generality that the closure of B, $\mathbf{cl}(B)$, is compact. Again we denote $w_i(x, y) := y_i - x_i^2 > 0$ for each $(x, y) \in B$. By similar construction as in (6), for each $(x, y) \in B$, bounds for d_i in the definition of $h(x, y)$ can be obtained as

$$-Q_{ii} \leq d_i \leq \frac{-\lambda_{min}(Q)\sum_i w_i(x, y) + \sum_{j\neq i} w_j(x, y)Q_{jj}}{w_i(x, y)}, \ \forall i.$$

Since $w_i(x, y)$ is strictly bounded away from 0 for all $(x, y) \in B$, by continuity and compactness of $\mathbf{cl}(B)$, we have

$$\sup_{(x,y)\in \mathbf{cl}B} \frac{-\lambda_{min}(Q)\sum_i w_i(x, y) + \sum_{j\neq i} w_j(x, y)Q_{jj}}{w_i(x, y)} < +\infty, \forall i.$$

Therefore, there exists a compact set $\mathcal{K} \subseteq \mathcal{R}^n$ such that

$$h(x, y) := \sup_d \left\{x^T\big(Q + \mathbf{diag}(d)\big)x - d^T y \;\middle|\; \forall d, Q + \mathbf{diag}(d) \succeq 0, d \in \mathcal{K}\right\}, \forall (x, y) \in B.$$

Then Theorem 2 can be applied to $h(x, y)$ in the open neighborhood B (and by letting $g(x, y; d) = x^T\big(Q + \mathbf{diag}(d)\big)x - d^T y$ and $Y = \mathcal{D} \cap \mathcal{K}$). Then our results on $\partial h(x, y)|_{(\bar{x},\bar{y})}$ follow straightforwardly. □

When $\bar{x}_i^2 = \bar{y}_i$ for some indices i, the finite attainment of (SEP) and the local structure of $h(\bar{x}, \bar{y})$ is much more delicate, depending on the problem data Q. A complete characterization is beyond the scope of this paper. However, we provide a small example where both properties fail.

Example 2. *Let* $Q = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $S_1 = S_2 = \{0, 1\}$ *and* $\bar{x} = \bar{y} = \begin{bmatrix} 1, & 0.5 \end{bmatrix}^T$. *It is easy to verify that* $\ell_i(x_i) = u_i(x_i) = x_i$. *So* $(\bar{x}, \bar{y}, \bar{v})$ *satisfies all assumptions in Proposition 2, and* $\mathcal{I} = \{1\}$, $\mathcal{J} = \{2\}$. *The separation problem (SEP) can be simplified to*

$$\sup_{d_1} \quad -0.25 d_1 \quad s.t. \quad \begin{bmatrix} d_1 & 1 \\ 1 & d_2 \end{bmatrix} \succeq 0.$$

Obviously the supremum takes place when $d_1 \mapsto 0$ *and* $d_2 \mapsto +\infty$. *Now we consider the local structure of* $h(x, y)$ *near* $(\bar{x}, \bar{y})$. *Let* $x_\epsilon = y_\epsilon = \begin{bmatrix} 1-\epsilon, & 0.5 \end{bmatrix}^T$ *for any* $0 \leq \epsilon < 1$,

$$h(x_\epsilon, y_\epsilon) = \sup_{\substack{d_1, d_2 \geq 0, \\ d_1 d_2 \geq 1}} \{(\epsilon^2 - \epsilon)d_1 - 0.25d_2 - (1-\epsilon)\} = \sqrt{\epsilon - \epsilon^2} - (1-\epsilon).$$

The directional derivative of $h(x, y)$ *at* $(\bar{x}, \bar{y})$ *in the direction of* $\left(\begin{bmatrix} -\epsilon, & 0 \end{bmatrix}^T, \begin{bmatrix} -\epsilon, & 0 \end{bmatrix}^T\right)$ *is:*

$$\lim_{\epsilon \mapsto 0+} \frac{h(x_\epsilon, y_\epsilon) - h(\bar{x}, \bar{y})}{-\epsilon} = \lim_{\epsilon \mapsto 0+} \frac{\sqrt{\epsilon - \epsilon^2} + \epsilon}{-\epsilon} = -\infty.$$

So the sub-differential of $\partial h(x, y)$ *is empty at* $(\bar{x}, \bar{y})$, *as there do not exist vectors* $s_1 \in \mathcal{R}^2$ *and* $s_2 \in \mathcal{R}^2$ *such that*

$$h(x_\epsilon, y_\epsilon) \geq h(\bar{x}, \bar{y}) + \langle s_1, \begin{bmatrix} -\epsilon, & 0 \end{bmatrix}^T \rangle + \langle s_1, \begin{bmatrix} -\epsilon, & 0 \end{bmatrix}^T \rangle, \forall \epsilon \in [0, 1).$$

Classical cutting-plane algorithms in non-smooth optimization assume that a subgradient can always be found [HUL01]. However, in practice we find it is not unusual that there are some indices where $\bar{y}_i = \bar{x}_i^2$, and a subgradient may not exist. As our emphasis of this paper is not to provide an algorithm that converges to an optimal solution to (SICP+), but instead, to construct a computationally tractable relaxation and a lower bound of (P), we provide a simple work around that appears to be effective in practice.

For $(\bar{x}, \bar{y}, \bar{v})$ feasible in (DiagR+), we propose to solve the following regularized separation problem

$$\sup_d \sum_{i=1}^n \left(\bar{x}_i^2 - \bar{y}_i\right) d_i - \lambda \sum_{i=1}^n [d_i]_+ \quad s.t. \quad Q + \mathbf{diag}(d) \succeq 0. \tag{SEPreg}$$

where $[d_i]_+$ is a convex function that equals d_i if $d_i > 0$ and 0 otherwise. The finite attainment of (SEPreg) can be easily established.

Proposition 3. *Assuming that* $\bar{x}_i^2 \leq \bar{y}_i$ *for all* i *and* $\lambda > 0$, *the optimal solution to the regularized separation problem (SEPreg) is always attained at some finite point.*

Proof. Note that $-\lambda_{min}(Q)e$ is a feasible solution to (SEPreg). We denote its objective value to be $\hat{\nu}$. Then any optimal solution to (SEPreg) must be in the following set

$$\{d | d_i \geq -Q_{ii}, \forall i \in [n], \ \left(\bar{y}_i - \bar{x}_i^2\right) d_i + \lambda \sum_{i=1}^n [d_i]_+ \leq -\hat{\nu}\}.$$

It is easy to verify that this is a compact set as each d_i is upper-bounded by the last inequality. □

We remark that in our numerical experiments in section 5, λ is chosen to be $\sum_{i=1}^n (\bar{y}_i - \bar{x}_i^2)$, which is always positive unless the current relaxation is already exact.

4 A Barrier Coordinate Minimization Algorithm for Regularized Separation

In this section we propose a specialized iterative algorithm to solve the following problem

$$\min_{d \in \mathcal{R}^n} \sum_{i=1}^{n} g_i(d_i) \;\; s.t. \;\; Q + \mathbf{diag}(d) \succeq 0, \tag{7}$$

where $g_i : \mathcal{R} \mapsto \mathcal{R}$ is a convex scalar function for each i. The regularized separation problem (SEPreg) is obvious a special case where

$$g_i(d_i) = \begin{cases} \bar{y}_i - \bar{x}_i^2 + \lambda & \text{if } d_i \geq 0; \\ \bar{y}_i - \bar{x}_i^2, & otherwise. \end{cases}$$

Our proposed algorithm has the desirable property that an admissible diagonal perturbation $d \in \mathcal{D}$ is always maintained. Therefore it can be terminated as soon as sufficient violation of the corresponding cutting surface is detected for the previous optimal solution $(\bar{x}, \bar{y}, \bar{v})$. Our algorithm is in principle a primal barrier method partially motivated by the block-descent method for general SDPs [WGS12]. We work on the following log-det penalized problem, where the penalty parameter $\sigma > 0$ is updated intelligently.

$$\begin{aligned} \min_{d} \;\; & f(d; \sigma) := \sum_{i=1}^{n} g_i(d_i) - \sigma \log\det(Q + \mathbf{diag}(d)) \\ & Q + \mathbf{diag}(d) \succ 0. \end{aligned} \tag{8}$$

The sub-differential of $f(d; \sigma)$ is

$$\partial f(d; \sigma) = -\sigma \mathbf{diag}\left([Q + \mathbf{diag}(d)]^{-1} \right) + \oplus_i \partial g_i(d_i), \tag{9}$$

where $\oplus_i \partial g_i(d_i)$ is the direct product of sub-differentials of $g_i(\cdot)$. Since the constraint $Q + \mathbf{diag}(d) \succ 0$ defines an open set and cannot be active, the optimality condition of (8) is

$$0 \in \partial f(d), \quad Q + \mathbf{diag}(d) \succ 0. \tag{10}$$

In each iteration, we store and update a feasible vector $\bar{d}$ and a corresponding inverse matrix $V = \left[Q + \mathbf{diag}(\bar{d})\right]^{-1}$. Motivated by the optimality condition (10), we choose an index $i \in \{1, ..., n\}$ with the largest magnitude in the following vector $s(\bar{d})$ to perform the coordinate minimization,

$$s\left(\bar{d}\right) := \min\left\{ \|u\|_2 \;\middle|\; u \in \partial f\left(\bar{d}; \sigma\right) \right\}, \quad i = \arg\max_j \left\{ \left| s\left(\bar{d}\right)_j \right| \right\}. \tag{11}$$

Note that by (9), $s(d)$ can be evaluated in linear time with the information of V, as long as a characterization of $\partial g_i(\cdot)$ is available. With this choice of index i we solve the following one-dimensional minimization problem,

$$\Delta d_i^* \in \arg\min_{\Delta d_i} \left\{ f(\bar{d} + \Delta d_i e_i; \sigma) \;\middle|\; Q + \mathbf{diag}\left(\bar{d} + \Delta d_i e_i\right) \succ 0 \right\}, \tag{12}$$

where e_i is the i-th vector in the canonical basis of $\mathcal{R}^n$. $\bar{d}$ is then updated by $\bar{d} \leftarrow \bar{d} + \Delta d_i^* e_i$ and V is updated by the Sherman-Morrison formula

$$V \leftarrow V - \frac{\Delta d_i^* \cdot v_i v_i^T}{1 + \Delta d_i^* \cdot V_{ii}}, \tag{13}$$

where v_i is the i-th column of the previous matrix V.

To solve the one-dimensional problem (12), we first consider in what range of Δd_i is the feasibility maintained.

Lemma 1. *Suppose that $\bar{d}$ is a vector such that $Q + \mathbf{diag}(\bar{d}) \succ 0$ and $V = \left[Q + \mathbf{diag}(\bar{d})\right]^{-1}$, then for each i, $Q + \mathbf{diag}\left(\bar{d} + \Delta d_i e_i\right) \succ 0$ if and only if $\Delta d_i > -V_{ii}^{-1}$.*

Proof. Without loss of generality we assume $i = n$, and

$$Q + \mathbf{diag}(\bar{d}) := \begin{bmatrix} M & q \\ q^T & Q_{nn} + \bar{d}_n \end{bmatrix}, \quad V = \left[Q + \mathbf{diag}(\bar{d})\right]^{-1} := \begin{bmatrix} \tilde{V} & v_n \\ v_n^T & V_{nn} \end{bmatrix}.$$

Note that $V_{nn} > 0$ as $V \succ 0$. By pre-multiplying $\begin{bmatrix} I & -\frac{v_n}{V_{nn}} \\ 0 & Q_{nn} + \bar{d}_n \end{bmatrix}$ to the equation $V(Q + \mathbf{diag}(\bar{d})) = I$, we obtain

$$\begin{bmatrix} \tilde{V} - \frac{v_n v_n^T}{V_{nn}} & 0 \\ \left(Q_{nn} + \bar{d}_n\right) v_n^T & \left(Q_{nn} + \bar{d}_n\right) V_{nn} \end{bmatrix} \begin{bmatrix} M & q \\ q^T & Q_{nn} + \bar{d}_n \end{bmatrix} = \begin{bmatrix} I & -\frac{v_n}{V_{nn}} \\ 0 & Q_{nn} + \bar{d}_n \end{bmatrix}. \tag{14}$$

Therefore we have $M^{-1} = \tilde{V} - \frac{v_n v_n^T}{V_{nn}}$. Now by the Schur Complement theorem, $Q + \mathbf{diag}\left(\bar{d} + \Delta d_n e_n\right) \succ 0$ if and only if

$$Q_{nn} + \bar{d}_n + \Delta d_n - q^T M^{-1} q > 0 \Leftrightarrow \Delta d_n > -\left(Q_{nn} + \bar{d}_n\right) + q^T \left(\tilde{V} - \frac{v_n v_n^T}{V_{nn}}\right) q. \tag{15}$$

By the upper-right block in (14) and the lower-right block in $V(Q + \mathbf{diag}(\bar{d})) = I$, we have $\left(\tilde{V} - \frac{v_n v_n^T}{V_{nn}}\right) q = -\frac{v_n}{V_{nn}}$ and $v_n^T q + \left(Q_{nn} + \bar{d}_n\right) V_{nn} = 1$, then the condition (15) is equivalent to

$$\Delta d_n > -\left(Q_{nn} + \bar{d}_n\right) - \frac{q^T v_n}{V_{nn}} = -\frac{\left(Q_{nn} + \bar{d}_n\right) V_{nn} + q^T v_n}{V_{nn}} = -\frac{1}{V_{nn}}.$$

□

Therefore to solve (12), it suffices to find Δd_i^* in the open interval $(V_{ii}^{-1}, +\infty)$ such that

$$\sigma \left\{ \left[Q + \mathbf{diag}(\bar{d}) + \Delta d_i E_{ii} \right]^{-1} \right\}_{ii} \in \partial g_i(\bar{d}_i + \Delta d_i). \tag{16}$$

Again by the Sherman-Morrison formula,

$$\sigma \left\{ \left[Q + \mathbf{diag}(\bar{d}) + \Delta d_i E_{ii} \right]^{-1} \right\}_{ii} = \sigma \left(V_{ii} - \frac{\Delta d_i V_{ii}^2}{1 + \Delta d_i \cdot V_{ii}} \right) = \frac{\sigma V_{ii}}{1 + \Delta d_i V_{ii}}. \tag{17}$$

The one-dimensional search (16) can be done very efficiently for our case of regularized separation (SEPreg). Note that in our case

$$\partial g_i(\bar{d}_i + \Delta d_i) = \begin{cases} \alpha_i, & if\ \Delta d_i < -\bar{d}_i; \\ [\alpha_i, \beta_i], & if\ \Delta d_i = -\bar{d}_i; \\ \beta_i, & if\ \Delta d_i \geq -\bar{d}_i; \end{cases} \tag{18}$$

where $\alpha_i := \bar{y}_i - \bar{x}_i^2$ and $\beta_i := \bar{y}_i - \bar{x}_i^2 + \lambda$. Therefore it suffices to find the unique intersection point of a nonlinear curve (17) and the piecewise linear curve (18), under the constraint $\Delta d_i > -\frac{1}{V_{ii}}$ in Lemma 1. Such an intersection point always exist because $\lim\limits_{\Delta d_i \mapsto +\infty} \dfrac{\sigma V_{ii}}{1 + \Delta d_i V_{ii}} = 0$ and $\beta_i > 0$. An explicit formula for Δd_i^* is then given by

$$\Delta d_i^* = \begin{cases} \frac{\sigma}{\beta_i} - \frac{1}{V_{ii}}, & \text{if } -\bar{d}_i < -\frac{1}{V_{ii}} \text{ or } \sigma \frac{V_{ii}}{1 - \bar{d}_i \cdot V_{ii}} > \beta_i; \\ -\bar{d}_i, & \text{if } -\bar{d}_i \geq -\frac{1}{V_{ii}} \text{ and } \alpha_i \leq \sigma \frac{V_{ii}}{1 - \bar{d}_i \cdot V_{ii}} \leq \beta_i; \\ \frac{\sigma}{\alpha_i} - \frac{1}{V_{ii}}, & \text{if } -\bar{d}_i \geq -\frac{1}{V_{ii}} \text{ and } \sigma \frac{V_{ii}}{1 - \bar{d}_i \cdot V_{ii}} < \alpha_i. \end{cases} \tag{19}$$

Figure 1 illustrates the case of $\alpha_i \leq \frac{\sigma V_{ii}}{1 - \bar{d}_i V_{ii}} \leq \beta_i$ and $-\bar{d}_i \geq -\frac{1}{V_{ii}}$, where the intersection takes place at $\Delta d_i^* = -\bar{d}_i$. We further remark that $\alpha_i = 0$ will not introduce any numerical issue because $-\bar{d}_i \geq -\frac{1}{V_{ii}}$ and $\sigma \frac{V_{ii}}{1 - \bar{d}_i \cdot V_{ii}} < 0$ cannot be simultaneously true (since $\sigma > 0$ and $V_{ii} > 0$).

The parameter σ is updated whenever problem (8) is solved to some satisfactory precision. Again we use $s(\bar{d})$ defined in (11) as our measure of optimality, and update σ according to the following rule,

$$\sigma \leftarrow \max(\text{SML\_SIG}, \text{SIG\_UPD} \cdot \sigma), \quad if \quad \frac{s(d)}{\|\beta\|_2} \leq \text{SUBG\_TOL}. \tag{20}$$

SML_SIG is a safe-guard parameter to avoid σ becoming too small. Our algorithm to solve the general form (7) is summarized in Algorithm 2. Note that the most expensive step in each iteration is a single rank one update of V, which takes $O(n^2)$ time with a small constant factor.

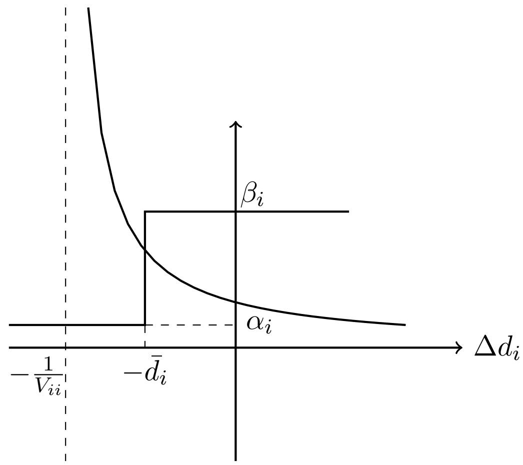

Figure 1: Illustration of the case when the optimal solution to (12) is $\Delta d_i^* = -\bar{d}_i$

Algorithm 2: A barrier coordinate minimization algorithm to solve (7)

Input : Matrix Q, initial $\sigma > 0$ and initial $\bar{d} \in \mathcal{R}^n$ such that $Q + \mathbf{diag}(\bar{d}) \succ 0$
Output: Vector $\bar{d}$ that is feasible and solves (7) approximately.

```
Compute V = [Q + diag(d̄)]^{-1} ;
for k = 1 to maxIter do
    Compute index i := arg max_j { |s(d̄)_j| }, where s(d̄) is defined in (11) ;
    Update d̄_i ← d̄_i + Δd_i^* where Δd_i^* is computed by solving (12) ;
    Update V using (13) ;
    Update σ using rule (20) ;
    Terminate if some termination rule is met;
end
```

4.1 Implementation Details

In our implementation and computational experiments, we set SML_SIG $= 10^{-5}$, SIG_UPD $=$ 0.8 and SUBG_TOL $= 0.03$. We choose initial $\bar{d}$ to be $-1.5\lambda_{\min}(Q)$ times the identity matrix if Q is indefinite, where $\lambda_{\min}(\cdot)$ is the minimal eigenvalue. With Q normalized to have spectral norm 1, initial σ is selected to be the median value of the set

$$\left\{ \left| \frac{u_i}{V_{ii}} \right| \right\}_{i=1}^{n}, \quad \textit{where} \quad u_i \in \partial g_i(\bar{d}_i).$$

The intuition is that we want the information from $g(\cdot)$ and $\log\det(Q + \mathbf{diag}(\cdot))$ to be of similar magnitude at the initial point. In every n iterations, we check our improvement of the objective value, and terminate our algorithm if the relative improvement in last n

iterations is less than a parameter SMLL_PRGRSS, which we set at 5E-4. We implement Algorithm 2 in the C language on a Mac OS X system and exploit Apple's Accelerate framework (to vectorize computation) and their implementation of cblas library whenever necessary.

5 Computational Experiments

We conduct three sets of numerical experiments to validate our contributions. We implement the cutting-surface procedure Algorithm 1 in the MATLAB environment. Convex quadratically constrained relaxations (DiagR+) are solved using the open source interior point code IPOPT [WB06] through the MATLAB interface they provided. Some other separation procedures, i.e., projected RLT cuts [SBL11] used in our third numerical experiment, are implemented using Yalmip [Löf04] and linear programming routines of Gurobi. Sometimes the MATLAB overhead is not negligible, especially when Yalmip is used to prepare inputs to optimization solvers. In these scenarios, we report only the aggregated time used by optimization solvers, and remark that the Yalmip overhead can be avoided in an efficient implementation.

5.1 Algorithm 2 versus interior point methods for SDP to solve (SEP)

SDP relaxations for dense Max-cut problems. When each function $g_i(d_i) = d_i$ in (7), it is the dual of the semidefinite relaxation for the Max-cut problem. Max-cut SDPs are extensively studied and often used to compare different semidefinite programming solvers [Mit]. We randomly generate dense instances and compare algorithm 2 with three implementations of interior point methods, namely, CSDP[Bor99], mc_psdpk, which is a specialized interior point code for max-cut SDPs used in a version of BiqMac solver[RRW10], and DSDP[BYZ00]. Note that Algorithm 2 does not exploit sparsity in matrix Q (the matrix V in Algorithm 2 is usually dense). Hence it is not competitive on many large scale sparse max-cut SDPs in Mittlemann's benchmark library [Mit], where there exist specialized algorithms including specialized SDPLR [BMZ02] and SpeeDP [GPP$^+$12].

Entries in Q are randomly generated from the normal distribution $\mathcal{N}(0,1)$. The matrix is then normalized with spectral norm 1. The comparisons on solution accuracy and solution time are summarized in table 1. All three interior point solvers achieve high accuracy (1E-7), while Algorithm 2 reports approximate solutions with relative error in the order of $10^{-3} \sim 10^{-4}$. DSDP is the most efficient interior point solver under comparison. However Algorithm 2 always finds an approximate solution in about 20% of time used by DSDP.

Instances with nonsmooth objective functions. When each function $g_i(\cdot)$ in (7) is the

Table 1: Interior point methods vs. Algorithm 2 on Max-cut SDPs

		IP methods (sec.)			Alg 2			
n	obj	CSDP	mc	DSDP	obj	T(sec.)	RelErr	SpeedUp
100	84.31	0.28	0.03	0.03	84.34	0.01	4.0e-04	3.3X
200	175.18	0.96	0.16	0.14	175.39	0.03	1.2e-03	4.7X
400	365.06	3.85	0.98	0.53	365.78	0.10	2.0e-03	5.1X
600	567.49	8.93	2.84	1.28	569.47	0.24	3.5e-03	5.4X
800	752.88	17.10	6.63	2.40	756.27	0.45	4.5e-03	5.3X
1000	945.73	28.31	10.70	3.50	951.37	0.69	6.0e-03	5.1X
1500	1425.72	73.49	37.57	9.71	1439.19	1.78	9.5e-03	5.5X

max of two linear functions, for example as in the case of regularized separation (SEPreg),

$$g_i(d_i) = \begin{cases} \alpha_i d_i, & \text{when } d_i \leq 0; \\ \beta_i d_i, & \text{when } d_i > 0, \end{cases}$$

where $0 \leq \alpha_i \leq \beta_i$, (7) can be reformulated into an SDP with $n+1$ additional variables and $2n$ linear constraints. We compare the performance of DSDP and Algorithm 2 in this experiment, where each α_i is generated uniformly in $[0, 0.5]$ and $\{\beta_i\}$ uniformly in $[0.5, 1]$. Table 2 summarizes our results. Compared to the previous Max-cut instances, the additional piecewise linear structure in $g_i(\cdot)$ significantly slows down the interior point solvers, while having little impact on the solution time of Algorithm 2. Overall, we can conclude that Algorithm 2 finds near optimal strictly feasible solutions to (7) much faster than interior point solvers. The benefits are more obvious when there is nonlinear/nonsmooth structure in the objective function of (7).

5.2 Cutting-surface algorithm versus Buchheim-Wiegele SDP

In this section we compare our cutting-surface algorithm with the semidefinite relaxation (BW) in [BW13], on various randomly generated instances with binary, integer, and mixed-integer variables. In [BW13] the authors developed a branch-and-bound algorithm Q-MIST based on solving semidefinite relaxations (BW). It was shown that the Q-MIST algorithm compares favorably to Couenne [BLL+09], a general purpose global solver for mixed-integer nonlinear programs. However, they only tested instances when $n \leq 60$. We observe that the cost of solving (BW) using interior point methods increases significantly when $n > 50$. On such instances our cutting-surface procedure provides lower bounds with similar level of strength in significantly shorter amount of time.

We generate random test instances similarly as in [BW13]. For the sake of completeness we repeat the settings here. Matrix Q is generated randomly by $Q = \sum_{i=1}^{n} \mu_i v_i v_i^T$, where

Table 2: Comparison between DSDP and Algorithm 2 to solve (SEP)

n	DSDP		Alg 2		RelErr	SpeedUp
	obj	T(sec.)	obj	T(sec.)		
100	69.14	0.13	69.19	0.01	8.0e-04	16X
200	133.95	0.50	134.08	0.02	9.0e-04	21X
400	276.90	1.99	277.54	0.10	2.3e-03	20X
600	419.86	4.60	421.73	0.24	4.4e-03	20X
800	557.33	9.45	560.71	0.41	6.1e-03	23X
1000	707.10	17.27	712.85	0.65	8.1e-03	27X
1500	1064.97	42.01	1078.76	1.79	1.3e-02	24X

for a percentage p (parameter used to control the level of convexity of Q), the first $\lfloor pn/100 \rfloor$ number of μ_i are chosen randomly from $[-1, 0]$, and the rest of them are chosen randomly from $[0, 1]$. We generate instances with $n = 50, 70, 100$ and $p = 0.2, 0.5, 0.8$. We also generate cases when Q is positive definite (corresponding to the rows of $p = 1.2$ in our results tables), where all eigenvalues are generated randomly in $[0.2, 1.2]$. Next, each v_i is a random vector of length n with entries uniformly generated from $[-1, 1]$, then normalized such that $\|v_i\|_2 = 1$. Finally the q vector in (P) has all entries uniformly generated from $[-1, 1]$. As a baseline for comparison, we run Cplex 12.6.3 [IBM15] for 300 seconds on each instance with options to aggressively finding feasible solutions and perform solution polishing in the last 60 seconds. We generate five sets of instances, corresponding to five cases of constrained sets S_i, where (1) $S_i = \{0, 1\}$ for all i; (2) $S_i = \{-1, 0, 1\}$ for all i; (3) $S_i = \{-3, ..., , 3\}$ for all i; (4) $S_i = \{-5, ..., 5\}$ for all i; and (5) the mixed-binary-integer-semicontinuous case: $S_i = \{0, 1\}$ for 20% of indices, $S_i = \{-3, ..., 3\}$ for 40% of indices and $S_i = \{0\} \cup [1, 2]$ for the rest of 40% of indices.

For each choice of n, p and sets $\{S_i\}_{i=1}^n$, we generate 10 instances. In each table 3–7, we report the number of instances solved to optimality by Cplex within 300 seconds. Denote the lower and upper bounds provided by Cplex to be LB_{cpx} and UB_{cpx}, respectively. The "Gap closed" column for (BW) is computed by the average of the following measure

$$\frac{\mu_{BW} - LB_{cpx}}{|UB_{cpx} - LB_{cpx}| + |\mu_{BW} - LB_{cpx}|}.$$

This measure equals -100% if (BW) provides a worse lower bound than Cplex, and equals 100% if (BW) closed the gap entirely. In the "T" column of the (BW) section we report the time needed to solve the "initial SDP model" in [BW13] only, where constraints $\ell_i(x_i) \le X_{ii} \le u_i(x_i)$ are replaced by a single constraint $X_{ii} \le (L_i + R_i)x_i - L_i R_i$. This is for a more fair time comparison as the authors of [BW13] add the constraints $\ell_i(x_i) \le X_{ii} \le u_i(x_i)$ iteratively as cutting planes. Finally we report the average number of cutting surfaces added (we limit our algorithm to add at most 20 cutting surfaces, and stop when small progress in bounds is observed), as well as the relative difference from the lower bound μ_{BW}, and the time per iteration in Algorithm 1.

Cplex can solve almost all binary instances with $n = 50$ to optimality, and as expected, performs better on convex instances. However, for those instances that Cplex cannot solve, the semidefinite relaxation (BW) can usually close a significant percentage of the residual gap. For different kinds of constraint sets $\{S_i\}_{i=1}^n$, our cutting-surface algorithm turns out to perform similarly, i.e, with relatively small number of cutting surfaces, we were able to obtain similar level of lower bounds compared to (BW).

n	p	Cplex	(BW)		Cutting-surface		
		(#solved)	Gap closed	T(sec.)	# cuts	Diff vs. (BW)	T(sec.)
50	0.2	10	-100%	3.0	19	0.40%	1.9
	0.5	9	-81.85%	3.0	19	0.20%	1.7
	0.8	10	-100%	3.1	14	0.14%	1.0
	1.2	10	-100%	3.4	19	1.22%	1.0
70	0.2	0	72.77%	12.9	20	0.45%	3.2
	0.5	0	88.78%	13.5	19	0.24%	2.7
	0.8	2	52.55%	13.9	15	0.17%	1.7
	1.2	10	-100%	14.8	18	1.13%	1.4
100	0.2	0	78.80%	70.8	20	0.54%	5.2
	0.5	0	94.51%	73.7	20	0.33%	4.6
	0.8	0	95.14%	77.4	17	0.14%	2.7
	1.2	2	51.15%	81.3	17	1.64%	2.0

Table 3: Comparison with (BW) on binary instances

5.3 BoxQP instances: Comparison with a projection approach by Saxena, Bonami and Lee

In our last numerical experiment, we compare our cutting-surface procedure with a projection method proposed in section 3 of [SBL11] on BoxQP instances, where $S_i = [0, 1], \forall i$. This method is referred to as the projection of the **MIQCP-SDP** relaxation in their paper. The authors also proposed additional cutting planes such as disjunctive cuts and polarity cuts in later sections of their paper, however based on their computational results they concluded that these additional cuts have an inconsequential effect on BoxQP instances.

We remark that when specialized to BoxQP problems, our procedure is similar to the projected **MIQCP-SDP** method in [SBL11] in the following sense:

1. Both Algorithm 1 and the projected **MIQCP-SDP** method generate convex quadratic relaxations with multiple quadratic constraints;

n	p	Cplex	(BW)		Cutting-surface		
		(#solved)	Gap closed	T(sec.)	# cuts	Diff vs. (BW)	T(sec.)
50	0.2	0	52.99%	2.3	18	0.20%	0.7
	0.5	0	49.70%	2.3	18	0.24%	0.9
	0.8	0	60.16%	2.4	18	0.19%	0.9
	1.2	10	-100%	2.5	20	0.22%	1.0
70	0.2	0	56.51%	9.7	18	0.39%	1.1
	0.5	0	55.35%	9.7	19	0.25%	1.1
	0.8	0	59.20%	10.4	19	0.22%	1.3
	1.2	2	12.33%	10.9	19	1.28%	1.5
100	0.2	0	63.15%	51.0	20	0.32%	1.8
	0.5	0	62.24%	53.2	20	0.31%	1.8
	0.8	0	63.32%	55.9	20	0.27%	1.8
	1.2	0	56.13%	60.5	19	0.94%	1.9

Table 4: Comparison with (BW) on integer instances with $S_i = \{-1, 0, 1\}$ for all i

n	p	Cplex	(BW)		Cutting-surface		
		(#solved)	Gap closed	T(sec.)	# cuts	Diff vs. (BW)	T(sec.)
50	0.2	0	48.44%	2.2	14	0.33%	0.7
	0.5	0	48.90%	2.1	12	0.29%	0.6
	0.8	0	54.36%	2.1	13	0.27%	0.7
	1.2	10	-100%	2.0	19	0.76%	1.1
70	0.2	0	53.95%	9.1	13	0.33%	1.0
	0.5	0	56.55%	8.8	12	0.35%	1.0
	0.8	0	60.72%	8.9	13	0.31%	1.0
	1.2	0	50.42%	8.6	18	1.09%	1.6
100	0.2	0	62.26%	50.7	11	0.37%	1.1
	0.5	0	64.88%	46.8	14	0.36%	1.4
	0.8	0	68.26%	46.6	14	0.38%	1.4
	1.2	0	55.06%	45.6	20	0.97%	2.4

Table 5: Comparison with (BW) on integer instances with $S = \{-3, ..., 3\}$ for all i

n	p	Cplex	(BW)		Cutting-surface		
		(#solved)	Gap closed	T(sec.)	# cuts	Diff vs. (BW)	T(sec.)
50	0.2	0	48.02%	2.3	9	0.31%	0.5
	0.5	0	50.28%	2.3	8	0.27%	0.5
	0.8	0	55.67%	2.1	8	0.30%	0.5
	1.2	10	-100%	2.0	19	0.65%	1.1
70	0.2	0	55.59%	9.5	6	0.33%	0.5
	0.5	0	57.14%	9.0	8	0.37%	0.6
	0.8	0	59.51%	9.1	8	0.35%	0.6
	1.2	1	30.05%	8.0	18	0.97%	1.6
100	0.2	0	60.99%	50.0	7	0.33%	0.8
	0.5	0	64.80%	46.6	8	0.38%	0.9
	0.8	0	67.19%	47.6	8	0.32%	0.9
	1.2	0	56.31%	42.2	19	1.12%	2.7

Table 6: Comparison with (BW) on integer instances with $S = \{-5, ..., 5\}$ for all i

n	p	Cplex	(BW)		Cutting-surface		
		(#solved)	Gap closed	T(sec.)	# cuts	Diff vs. (BW)	T(sec.)
50	0.2	0	55.49%	3.0	16	1.01%	1.4
	0.5	0	60.72%	2.6	10	0.54%	0.8
	0.8	0	77.92%	2.9	16	0.41%	1.3
	1.2	10	-100%	3.7	20	0.60%	2.2
70	0.2	0	62.10%	13.3	15	0.98%	2.3
	0.5	0	65.73%	12.8	15	0.49%	2.0
	0.8	0	70.11%	14.1	17	0.29%	2.0
	1.2	1	50.94%	15.7	9	0.55%	1.5
100	0.2	0	67.47%	66.3	16	0.67%	3.8
	0.5	0	60.55%	69.3	16	0.77%	2.7
	0.8	0	61.03%	74.5	10	0.67%	1.6
	1.2	0	93.66%	88.6	10	1.83%	2.3

Table 7: Comparison with (BW) on mixed-integer instances with binary, integer, and semicontinuous variables

2. Both Algorithm 1 and the projected **MIQCP-SDP** method have an underlying semidefinite relaxation model (BW-SDP versus lifted **MIQCP-SDP** relaxation for BoxQP), and produce convex quadratic relaxations that are shown to capture most of the strength of corresponding SDP relaxations;

3. Both Algorithm 1 and the projected **MIQCP-SDP** method employ a first-order feasible approximate method to generate new cutting surfaces (primal-barrier coordinate minimization versus projected subgradient in [SBL11]);

On the other hand, our approach is different from the projected **MIQCP-SDP** method for the following reasons:

1. Our method exploits more structure in S_i, while the projected **MIQCP-SDP** method only exploits variable bounds;

2. For the case of BoxQP problems, our Algorithm 1 essentially exploits only the diagonal RLT constraints
$$0 \leq X_{ii} \leq x_i, \quad \forall i,$$
while ignoring other off-diagonal RLT constraints. Therefore our procedure is theoretically weaker than the (**ProjSDP**) model in Theorem 3 of [SBL11]. However, this loss is remedied by the fact that we can employ a more efficient separation procedure, i.e., Algorithm 2, versus the projected subgradient algorithm in [SBL11], which requires an eigenvalue factorization in each iteration.

In order to further exploit the off-diagonal RLT inequalities, we combine the linear cutting plane procedure (**ProjLP**) in [SBL11] into Algorithm 1. We remark that (**ProjLP**) essentially projects down the full RLT inequalities and generates linear valid inequalities in the original variable space by solving some simple linear programs with $O(n^2)$ number of variables, and is computationally very cheap.

Again motivated by the (**MIQCP-Initial**) reformulation in [SBL11], we augment the (DiagR+) model with a convex inequality generated by splitting Q into its convex and concave parts and introducing an additional scalar variable τ,

$$\begin{aligned}
\min_{v,x,y} \quad & v + q^T x \\
s.t. \quad & v \geq x^T\big(Q + \mathbf{diag}(d)\big)x - d^T y, \quad \forall d \in \mathcal{D}^{(k)} \\
& v \geq x^T Q^+ x + \tau, \\
& \ell_i(x_i) \leq y_i \leq u_i(x_i), \\
& L_i \leq x_i \leq R_i, \quad \forall i \in [n].
\end{aligned}$$

where $Q = Q^+ + Q^-$, $Q^+ = \sum_{i:\lambda_i>0} \lambda_i v_i v_i^T$ and $\{(\lambda_i, v_i)\}$ are the eigen-pairs of Q. Next we enforce the nonconvex constraint $\tau \geq x^T Q^- x$ by separating the following set by using the method of (**ProjLP**) in [SBL11],

$$(x, \tau, v) \in \left\{ (x, \tau, v) \middle| \exists X, \begin{array}{l} \langle Q^+, X\rangle + \tau - v \le 0 \\ \langle Q^-, X\rangle - \tau \le 0 \\ L_i \le x_i \le R_i,\ \forall i \\ y_{ij}^-(x) \le X_{ij} \le y_{ij}^+(x), \forall i, j \end{array} \right\}$$

where

$$y_{ij}^-(x) = \max\{R_i x_j + R_j x_i - R_i R_j, L_i x_j + L_j x_i - L_i L_j\},\ \forall i, j$$
$$y_{ij}^+(x) = \min\{L_i x_j + R_j x_i - L_i R_j, R_i x_j + L_j x_i - R_i L_j\},\ \forall i, j.$$

We name this augmented procedure "Alg 1+" in our later comparison.

Finally we present our numerical results on all 90 BoxQP instances in [VN05] and compare to the results of "W3" method reported in [SBL11], which corresponds to their implementation of the projected **MIQCP-SDP** cutting model (**ProjSDP**). (Though we have no information on the specific machine they are using, it is extremely unlikely that their computer/implementation is several orders of magnitude slower than ours.) Similar to their comparison strategy, we use the gap between the optimal values and the naive RLT relaxations as a baseline, and calculate how much more gap can be closed by more sophisticated bounding procedures (**ProjSDP**) and our aforementioned "Alg 1+" procedure. We report our summary in Table 8 and leave the detailed results of each instance in the Appendix. The "Diff" column is the average difference of the amount of gap closed by these two procedures. A negative number means Alg 1+ is worse. We remark that in all instances, Alg 1+ is only weaker by a small amount, but requires significantly less time to compute. On the other hand, the difference in time required by these two procedures is several orders of magnitude.

Groups	#inst.	Average % gap closed				Average Time (s)	
		SBL	Alg 1	Alg 1+	Diff.	SBL	Alg 1+
spar020*-030*	18	97.14%	91.90%	94.65%	-2.49%	119.73	0.38
spar040*	24	96.37%	89.00%	91.51%	-4.86%	82.31	0.46
spar050*-070*	21	93.41%	87.76%	89.61%	-3.80%	209.92	0.63
spar080*-100*	27	94.24%	92.81%	92.89%	-1.34%	618.74	0.84

Table 8: Summary of comparison with the projected **MIQCP-SDP** procedure in [SBL11] on BoxQP instances

The time reported in the "SBL" column is directly extracted from the paper [SBL11]. Although these numerical experiments were performed at least six years before our experiments, we find it unlikely that the speed-up was entirely due to the improvement of computer hardware. We use a computer with an Intel(R) Xeon(R) CPU with a cpu clock frequency of 2.40GHz, and our implementation does not exploit multi-core parallel computing. We believe the main reason for the huge time difference is that we only search for convex cutting surfaces in a very restricted form, i.e., with Hessian matrices simply diagonal perturbations of the original quadratic function. This restriction greatly simplifies the separation SDP problem one needs to solve. Moreover, this diagonal perturbation approach apparently captures much of problem structure very effectively, e.g., the separability in the constraints $x_i \in S_i,\ \forall i$, and only a small number of iterations is needed to derive a strong relaxation.

One may argue that like all cutting plane procedures, the SBL procedure has a strong tailing effect. Could it be the case that most of the time used by SBL procedure is devoted to closing an insignificant amount of gap? Fortunately, [SBL11] also reports the time needed to close the amount of gap that is only 1% less than the final amount of gap closed, in the columns titled "W3(Adj)" in many of their tables. We remark that in many instances, Alg 1+ provides better bounds than that of "W3(Adj)", including 7 out of 9 largest instances "spar100*", while Alg 1+ is still significantly faster (see table 9). This clearly demonstrates the advantage of Alg 1+ over the projected **MIQCP-SDP** method on BoxQP problems, especially on the larger instances.

Instance	RLT	OPT	% duality gap closed			Time taken (s)	
			W3(Adj)	Alg 1+	Diff.	W3(Adj)	Alg 1+
spar100-025-1	-7660.75	-4027.50	91.36%	91.66%	0.30%	385.64	1.09
spar100-025-2	-7338.50	-3892.56	91.16%	91.90%	0.74%	321.79	1.55
spar100-025-3	-7942.25	-4453.50	92.26%	91.38%	-0.88%	299.23	1.26
spar100-050-1	-15415.75	-5490.00	92.62%	93.88%	1.26%	286.59	0.93
spar100-050-2	-14920.50	-5866.00	93.13%	93.50%	0.37%	288.09	1.11
spar100-050-3	-15564.25	-6485.00	94.81%	94.49%	-0.32%	279.41	0.99
spar100-075-1	-23387.50	-7384.20	94.84%	96.06%	1.22%	366.24	0.92
spar100-075-2	-22440.00	-6755.50	95.47%	96.04%	0.57%	330.70	1.00
spar100-075-3	-23243.50	-7554.00	95.06%	95.49%	0.43%	303.30	1.23

Table 9: Comparison with the projected **MIQCP-SDP** method on 9 largest BoxQP instances

6 Extensions and Final Remarks

We propose a cutting-surface procedure based on multiple diagonal perturbations to derive strong but efficiently solvable convex quadratic relaxations for nonconvex quadratic problems with separable constraints. Our method can be seen as a way to construct outer-approximations using convex quadratic cutting surfaces by exploiting a semi-infinite formulation of an SDP relaxation proposed by Buchheim and Wiegele [BW13]. We provide some results on the subdifferential of an equivalent non-smooth formulation. These results motivate us to solve a regularized separation problem to generate cutting surfaces. We propose to solve the separation problem with a specialized barrier coordinate minimization algorithm. We show that our separation algorithm finds strictly feasible and nearly optimal solutions much faster than interior point methods for SDPs, with some sacrifice on the accuracy. We perform extensive numerical comparisons on randomly generated instances. On quadratic mixed-integer problems, our cutting-surface procedure provides lower bounds of nearly the same strength with the SDP bounds used in [BW13], while our procedure is much faster on problems with dimension greater than 70. Combined with linear projected RLT cutting planes proposed in [SBL11], we also compare our method with a projected **MIQCP-SDP** method proposed by Saxena, Bonami and Lee [SBL11] on the Box-QP instances. Results also show that our method provides slightly weaker bounds but in significantly shorter amount of time.

There are many directions to extend our work to improve branch-and-bound algorithms for mixed-integer nonlinear programs with nonconvex quadratics. First, if there exist linear equality constraints $Ax = b$, our separation algorithm can be revised to exploit this. For example, in (SEP), $Q + \mathbf{diag}(d)$ only needs to be positive semidefinite over the null space of A, although computationally care has to be taken to deal with the case that the primal optimal solution is not finitely attained (indeed similar ideas are also mentioned in [BT13]). Secondly, it is reasonable to expect that when incorporating our diagonal perturbation procedure into a branch-and-bound framework to solve (P) globally, the new algorithm should perform better than Q-MIST, at least on relatively larger instances, as it is not practical to solve (BW) using interior point methods for larger instances. Finally, since our procedure can be thought as a partial lifting procedure that lifts only the diagonal entries X_{ii}, and exploiting one-variable valid constraints $\ell_i(x_i) \leq x_i^2 \leq u_i(x_i)$, it would be interesting to identify important multi-variable valid constraints and generalize our approach to a sparse lifting or sparse perturbation approach.

Acknowledgement

The author wish to thank Angelika Wiegele for sharing an implementation of an interior point method specialized for the Max-cut SDP. He also wishes to thank three anonymous reviewers for careful reading and helpful comments that lead to significant improvements of this paper.

References

[Ans09] Kurt M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43:471–484, 2009.

[BC12] Samuel A. Burer and Jieqiu Chen. Globally solving nonconvex quadratic programming problems via completely positive programming. *Mathematical Programming Computation*, 4:33–52, 2012.

[BEP09] Alain Billionnet, Sourour Elloumi, and Marie-CHristine Plateau. Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The qcr method. *Discrete Applied Mathematics*, 157:1185–1197, 2009.

[BLL+09] Pietro Belotti, Jon Lee, Leo Liberti, François Margot, and Andreas Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optim. Methods and Software*, 24(4-5):597–634, 2009.

[BMZ02] S. Burer, R.D.C. Monteiro, and Y. Zhang. Solving a class of semidefinite programs via nonlinear programming. *Mathematical Programming*, 93:97–122, 2002.

[Bor99] B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.

[BS12] Samuel A. Burer and Anureet Saxena. The MILP Road to MIQCP. In *The IMA Volumes in Mathematics and its Applications*, volume 154, pages 373–405. Springer, 2012.

[BT13] C. Buchheim and E. Traversi. Separable non-convex underestimators for binary quadratic programming. In *Lecture Notes in Computer Science*, volume 7933, pages 236–247, 2013.

[BT15] Christoph Buchheim and Emiliano Traversi. Quadratic 0–1 optimization using separable underestimators. Optimization online., 2015.

[BW13] Christoph Buchheim and Angelika Wiegele. Semidefinite relaxations for non-convex quadratic mixed-integer programming. *Mathematical Programming*, 141:435–452, 2013.

[BYZ00] S. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10:443–461, 2000.

[DCL15] Hongbo Dong, Kun Chen, and Jeff Linderoth. Regularization vs. relaxation: A conic optimization perspective of statistical variable selection. Working paper, Department of MathematicsWashington State University, 2015.

[DK15] Hongbo Dong and Nathan Krislock. Semidefinite approaches for miqcp: Convex relaxations and practical methods. In *Modeling and Optimization: Theory and Applications*, number 147 in Springer Proceedings in Mathematics & Statistics, pages 49–75, 2015.

[DL13] Hongbo Dong and Jeff Linderoth. On valid inequalities for quadratic programming with continuous variables and binary indicators. In *The 16th Conference on Integer Programming and Combinatorial Optimization; Lecture Notes in Computer Science*, volume 7801, pages 169–180, 2013.

[FG07] Antonio Frangioni and Claudio Gentile. SDP diagonalizations and perspective cuts for a class of nonseparable MIQP. *Operations Research Letters*, 35(2):181–185, March 2007.

[GL10] Oktay Günlük and Jeff Linderoth. Perspective reformulations of mixed integer nonlinear programming with indicator variables. *Mathematical Programming (Series B)*, 124(1-2):183–205, 2010.

[GPP+12] Luigi Grippo, Laura Palagi, Mauro Piacentini, Veronica Piccialli, and Giovanni Rinaldi. Speedp: an algorithm to compute sdp bounds for very large max-cut instances. *Math. Program., Ser. B*, 136:353–373, 2012.

[HUL01] J. B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer Sciences & Business Media, 2001.

[IBM15] IBM Corporation. *IBM ILOG CPLEX Optimization Studio 12.6, User Manual*, 2015.

[LJRR04] Frauke Liers, Michael J unger, Gerhard Reinelt, and Giovanni Rinaldi. Computing Exact Ground-States of Hard Ising Spin-Glass Problems by Branch-and-cut. In A. Hartmann and H. Rieger, editors, *New Optimization Algorithms in Physics*, pages 47–68. Wiley, 2004.

[Löf04] J. Löfberg. Yalmip: A toolbox for modeling and optimization in matalb. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[Mit] H. D. Mittelmann. `http://plato.asu.edu/ftp/sparse_sdp.html`.

[RRW10] Franz Rendl, Giovanni Rinaldi, and Angelika Wiegele. Solving Max-Cut to optimality by intersecting semidefinite and polyhedral relaxations. *Math. Program., Ser. A*, 121:307–335, 2010.

[SBL11] Anureet Saxena, Pierre Bonami, and Jon Lee. Convex relaxations of mixed integer quadratically constrained programs: Projected formulations. *Mathematical Programming, Series A*, 130(2):359–413, 2011.

[SWMF12] A. Skjäl, T. Westerlund, R. Misener, and C. A. Floudas. A generalization of the classical αBB convex underestimation via Diagonal and Nondiagonal Quadratic Terms. *Journal of Optimization Theory and Applications*, 154:462–490, 2012.

[VN05] D. Vandenbussche and G. Nemhauser. A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. *Mathematical Programming*, 102(3):559–575, 2005.

[WB06] Andreas Wächter and Lorenz T. Biegler. On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1):25–57, 2006.

[WGS12] Zaiwen Wen, Donald Goldfarb, and Katya Scheinberg. Block Coordinate Descent Methods for Semidefinite Programming. In Miguel F. Anjos and Jean B. Lasserre, editors, *Handbook on Semidefinite, Conic and Polynomial Optimization*, volume 166 of *International Series in Operations Research & Management Science*, pages 533–564. Springer, 2012.

[WJ08] Martin J. Wainwright and Michael Jordan. Graphical models, exponential families and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

[Zha10] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38(2):894–942, 2010.

[ZSL14] Xiaojin Zheng, Xiaoling Sun, and Duan Li. Improving the performance of miqp solvers for quadratic programs with cardinality and minimum threshold constraints: A semidefinite program approach. *INFORMS Journal on Computing*, 26(4):690 – 703, 2014.

Table 10: Full comparison for BoxQP instances

Instance	RLT	OPT	% duality gap closed			Time taken (s)	
			SBL	Alg 1+	Diff.	SBL	Alg 1+
spar020-100-1	-1066.00	-706.50	98.28%	96.85%	-1.43%	43.06	0.52
spar020-100-2	-1289.00	-856.50	94.61%	91.65%	-2.96%	2.49	0.33
spar020-100-3	-1168.50	-772.00	99.98%	99.88%	-0.10%	408.36	0.54
spar030-060-1	-1454.75	-706.00	93.84%	90.75%	-3.09%	13.40	0.19
spar030-060-2	-1699.50	-1377.17	97.35%	95.45%	-1.90%	50.79	0.54
spar030-060-3	-2047.00	-1293.50	95.62%	89.67%	-5.95%	33.92	0.30
spar030-070-1	-1569.00	-654.00	89.88%	88.99%	-0.89%	12.33	0.17
spar030-070-2	-1940.25	-1313.00	98.51%	95.34%	-3.17%	188.12	0.50
spar030-070-3	-2302.75	-1657.40	96.07%	94.59%	-1.48%	31.57	0.47
spar030-080-1	-2107.50	-952.73	95.04%	90.62%	-4.42%	23.57	0.18
spar030-080-2	-2178.25	-1597.00	100.00%	98.73%	-1.27%	226.60	0.45
spar030-080-3	-2403.50	-1809.78	99.20%	98.59%	-0.61%	339.41	0.42
spar030-090-1	-2423.50	-1296.50	99.21%	96.79%	-2.42%	53.39	0.35
spar030-090-2	-2667.00	-1466.84	98.56%	96.10%	-2.46%	56.98	0.44
spar030-090-3	-2538.25	-1494.00	99.88%	99.03%	-0.85%	565.88	0.36
spar030-100-1	-2602.00	-1227.13	98.38%	95.34%	-3.04%	30.28	0.25
spar030-100-2	-2729.25	-1260.50	96.93%	92.34%	-4.59%	18.85	0.28
spar030-100-3	-2751.75	-1511.05	97.16%	93.03%	-4.13%	56.21	0.53
spar040-030-1	-1088.00	-839.50	97.64%	92.02%	-5.62%	117.60	0.79
spar040-030-2	-1635.00	-1429.00	91.60%	74.38%	-17.22%	68.46	0.65
spar040-030-3	-1303.25	-1086.00	93.04%	77.32%	-15.72%	104.80	0.69
spar040-040-1	-1606.25	-837.00	87.85%	83.55%	-4.30%	43.71	0.42
spar040-040-2	-1920.75	-1428.00	99.61%	95.06%	-4.55%	114.57	0.50
spar040-040-3	-2039.75	-1173.50	92.94%	88.12%	-4.82%	35.77	0.34
spar040-050-1	-2146.25	-1154.50	93.71%	87.38%	-6.33%	43.86	0.35
spar040-050-2	-2357.25	-1430.98	95.17%	89.31%	-5.86%	54.14	0.36
spar040-050-3	-2616.00	-1653.63	94.81%	89.95%	-4.86%	44.05	0.39
spar040-060-1	-2872.00	-1322.67	93.47%	88.65%	-4.82%	46.67	0.26
spar040-060-2	-2917.50	-2004.23	96.20%	91.18%	-5.02%	80.14	0.52
spar040-060-3	-3434.00	-2454.50	99.18%	97.06%	-2.12%	134.80	0.82
spar040-070-1	-3144.00	-1605.00	98.85%	95.72%	-3.13%	101.61	0.41
spar040-070-2	-3369.25	-1867.50	98.56%	94.76%	-3.80%	94.96	0.37
spar040-070-3	-3760.25	-2436.50	97.83%	94.15%	-3.68%	112.96	0.41
spar040-080-1	-3846.50	-1838.50	98.43%	94.72%	-3.71%	134.03	0.30
spar040-080-2	-3833.00	-1952.50	98.26%	95.78%	-2.48%	47.06	0.24
spar040-080-3	-4361.50	-2545.50	97.98%	96.11%	-1.87%	83.80	0.86

Continued on next page

Table 10 – continued from previous page

Instance	RLT	OPT	% duality gap closed			Time taken (s)	
			SBL	Alg 1+	Diff.	SBL	Alg 1+
spar040-090-1	-4376.75	-2135.50	98.22%	94.45%	-3.77%	103.96	0.48
spar040-090-2	-4357.75	-2113.00	98.04%	92.53%	-5.51%	83.69	0.33
spar040-090-3	-4516.75	-2535.00	99.00%	97.01%	-1.99%	81.20	0.45
spar040-100-1	-5009.75	-2476.38	98.72%	97.14%	-1.58%	81.56	0.46
spar040-100-2	-4902.75	-2102.50	97.93%	95.72%	-2.21%	121.76	0.41
spar040-100-3	-5075.75	-1866.07	95.87%	94.17%	-1.70%	40.16	0.24
spar050-030-1	-1858.25	-1324.50	96.40%	90.23%	-6.17%	165.74	0.89
spar050-030-2	-2334.00	-1668.00	90.74%	85.47%	-5.27%	79.42	0.50
spar050-030-3	-2107.25	-1453.61	91.45%	83.55%	-7.90%	121.65	0.71
spar050-040-1	-2632.00	-1411.00	97.23%	92.86%	-4.37%	177.96	0.45
spar050-040-2	-2923.25	-1745.76	94.06%	87.88%	-6.18%	85.63	0.40
spar050-040-3	-3273.50	-2094.50	97.53%	93.25%	-4.28%	180.96	0.63
spar050-050-1	-3536.00	-1198.41	87.88%	90.36%	2.48%	50.22	0.36
spar050-050-2	-3500.50	-1776.00	93.13%	89.00%	-4.13%	67.20	0.30
spar050-050-3	-4119.75	-2106.10	95.01%	91.59%	-3.42%	93.62	0.36
spar060-020-1	-1757.25	-1212.00	91.00%	85.57%	-5.43%	163.42	0.77
spar060-020-2	-2238.25	-1925.50	90.22%	85.51%	-4.71%	226.11	1.22
spar060-020-3	-2098.75	-1483.00	85.78%	79.44%	-6.34%	121.83	0.45
spar070-025-1	-3832.75	-2538.91	92.61%	87.48%	-5.13%	249.97	1.17
spar070-025-2	-3248.00	-1888.00	89.79%	86.47%	-3.32%	191.12	0.86
spar070-025-3	-4167.25	-2812.28	90.68%	85.24%	-5.44%	214.40	0.83
spar070-050-1	-7210.75	-3252.50	94.40%	92.10%	-2.30%	240.93	0.69
spar070-050-2	-6620.00	-3296.00	95.77%	93.53%	-2.24%	283.03	0.45
spar070-050-3	-7522.00	-4306.50	99.36%	97.00%	-2.36%	693.28	0.46
spar070-075-1	-11647.75	-4655.50	96.90%	96.06%	-0.84%	365.50	0.58
spar070-075-2	-10884.75	-3865.15	95.57%	94.45%	-1.12%	293.31	0.58
spar070-075-3	-11262.25	-4329.40	96.18%	94.81%	-1.37%	342.92	0.56
spar080-025-1	-4840.75	-3157.00	93.91%	89.06%	-4.85%	524.07	1.16
spar080-025-2	-4378.50	-2312.34	88.14%	87.17%	-0.97%	257.62	0.79
spar080-025-3	-5130.25	-3090.88	91.59%	90.17%	-1.42%	420.61	1.17
spar080-050-1	-9783.25	-3448.10	92.65%	92.42%	-0.23%	355.97	0.45
spar080-050-2	-9270.00	-4449.20	97.50%	95.21%	-2.29%	892.96	0.62
spar080-050-3	-10029.75	-4886.00	95.58%	93.60%	-1.98%	435.41	0.55
spar080-075-1	-15250.75	-5896.00	96.93%	96.02%	-0.91%	387.48	0.64
spar080-075-2	-14246.50	-5341.00	96.95%	95.72%	-1.23%	450.96	0.37
spar080-075-3	-14961.50	-5980.50	96.11%	95.16%	-0.95%	416.32	0.54
spar090-025-1	-6171.50	-3372.50	90.12%	88.36%	-1.76%	408.73	0.90

Continued on next page

Table 10 – continued from previous page

Instance	RLT	OPT	% duality gap closed			Time taken (s)	
			SBL	Alg 1+	Diff.	SBL	Alg 1+
spar090-025-2	-6015.00	-3500.29	89.45%	85.12%	-4.33%	444.30	0.95
spar090-025-3	-6698.25	-4299.00	90.57%	85.10%	-5.47%	446.74	1.16
spar090-050-1	-12584.00	-5152.00	95.02%	93.82%	-1.20%	506.72	0.48
spar090-050-2	-11920.50	-5386.50	96.61%	96.15%	-0.46%	514.05	0.83
spar090-050-3	-12514.00	-6151.00	95.90%	93.56%	-2.34%	991.04	0.45
spar090-075-1	-19054.25	-6267.45	95.66%	95.81%	0.15%	462.16	0.62
spar090-075-2	-18245.50	-5647.50	95.92%	95.40%	-0.52%	784.59	0.60
spar090-075-3	-18929.50	-6450.00	96.11%	95.87%	-0.24%	602.44	0.44
spar100-025-1	-7660.75	-4027.50	92.36%	91.66%	-0.70%	670.15	1.09
spar100-025-2	-7338.50	-3892.56	92.16%	91.90%	-0.26%	538.03	1.55
spar100-025-3	-7942.25	-4453.50	93.26%	91.38%	-1.88%	656.59	1.26
spar100-050-1	-15415.75	-5490.00	93.62%	93.88%	0.26%	757.14	0.93
spar100-050-2	-14920.50	-5866.00	94.13%	93.50%	-0.63%	929.91	1.11
spar100-050-3	-15564.25	-6485.00	95.81%	94.49%	-1.32%	747.46	0.99
spar100-075-1	-23387.50	-7384.20	95.84%	96.06%	0.22%	1509.96	0.92
spar100-075-2	-22440.00	-6755.50	96.47%	96.04%	-0.43%	936.61	1.00
spar100-075-3	-23243.50	-7554.00	96.06%	95.49%	-0.57%	657.84	1.23