J Glob Optim (2018) 70:719–735
https://doi.org/10.1007/s10898-017-0591-0

Quadratic convex reformulation for nonconvex binary quadratically constrained quadratic programming via surrogate constraint

Xiaojin Zheng[1] · **Yutong Pan**[1] · **Xueting Cui**[2]

Received: 28 October 2016 / Accepted: 22 November 2017 / Published online: 26 February 2018

Abstract We investigate in this paper nonconvex binary quadratically constrained quadratic programming (QCQP) which arises in various real-life fields. We propose a novel approach of getting quadratic convex reformulation (QCR) for this class of optimization problem. Our approach employs quadratic surrogate functions and convexifies all the quadratic inequality constraints to construct QCR. The price of this approach is the introduction of an extra quadratic inequality. The "best" QCR among the proposed family, in terms that the bound of the corresponding continuous relaxation is best, can be found via solving a semidefinite programming problem. Furthermore, we prove that the bound obtained by continuous relaxation of our best QCR is as tight as Lagrangian bound of binary QCQP. Computational experiment is also conducted to illustrate the solution efficiency improvement of our best QCR when applied in off-the-shell software.

Keywords Binary QCQP · Semidefinite programming · Quadratic convex reformulation · Global optimization

---

This work was supported by National Natural Science Foundation of China under Grants 11671300, 11371103 and 71501122.

✉ Xueting Cui
cui.xueting@shufe.edu.cn

Xiaojin Zheng
xjzheng@tongji.edu.cn

Yutong Pan
11ytpan@tongji.edu.cn

1 School of Economics and Management, Tongji University, Shanghai 200092, People's Republic of China

2 School of Mathematics, Shanghai University of Finance and Economics, Shanghai 200433, People's Republic of China

1 Introduction

Consider the following quadratically constrained quadratic program (QCQP):

$$\begin{aligned} &\min x^T Q_0 x + c_0^T x, \\ &\text{s.t. } x^T Q_j x + c_j^T x + h_j \leq 0, \quad j = 1, \ldots, J, \end{aligned}$$

where $Q_j, j = 0, 1, \ldots, J$ are symmetric $n \times n$ matrices, $c_0, c_j \in \Re^n, h_j \in \Re, j = 1, \ldots, J$. Problem QCQP is often used in modeling real-world management problems, for example, portfolio selection with multiple risk measures [10,18], Euclidean distance geometry problem [13], facility location problem [17] and so on. Problem QCQP can also be used to model many combinatorial optimization problems.

It is well known that QCQP problem is convex and polynomially solvable when all Q_js, $j = 0, 1, \ldots, J$, are positive semidefinite. This is because problem QCQP can be reformulated as a second-order cone program (SOCP) which can be solved polynomially via interior-point method [21,26]. The other cases where QCQP is polynomially solvable are also investigated in the literature [4,16,27,28]. Unfortunately, problem QCQP is in general NP-hard and its global optima are difficult to reach [12].

The general global optimization methods to solve QCQP problem are under the framework of branch-and-bound (B&B) methods. Based on outer approximations, a B&B method with the subproblems approximated by linear programming was proposed for finding approximate global solutions to QCQP, [1]. Novel B&B algorithms with simplicial partition as a branching strategy were then proposed in [19,23]. Based on the idea of successively linearizing the subproblems using reformulation-and-linearization techniques (RLT) [25], a branch-and-cut algorithm [3] was designed for QCQP.

Note that the major factors affecting the performance of B&B methods are the tightness of the relaxation and the efficiency of solving the relaxation problem at each node of the B&B tree. Thus, different convex relaxations of QCQP are introduced and investigated in the literature. Since semidefinite programming, (SDP), can be polynomially solved by interior-point method, [21,26], SDP relaxation has become an attractive approach for obtaining good convex relaxations. A common way of getting SDP relaxation is to lift x to a symmetric matrix $X = xx^T$ and relax this equation to $X \succeq xx^T$. We denote this type of relaxation by (SP_0). For QCQP with box constraints, the SDP relaxation (SP_0) can be further tightened by adding the RLT constraints. The resulting relaxation is referred to as SDP+RLT, [2]. For QCQP with convex quadratic constraints, a general difference of convex functions (DC) decomposition scheme was proposed to construct a tighter SDP than (SP_0), [30]. In order to get tighter SDP relaxations than (SP_0) for general QCQP, there are various techniques for generating different valid inequalities, [9]. Zheng et al. [29] developed a decomposition–approximation method for generating convex relaxations for nonconvex QCQP and they employed rank-2 semidefinite inequalities to generate a class of new valid inequalities. On the other hand, the main drawback of the strengthened SDP relaxations mentioned above is their huge size, even for the QCQP problems with medium size. Based on the projection techniques, several methods were investigated for efficiently solving the huge-size convex relaxation while maintaining the strength of the relaxation [24].

1.1 Research motivation and main contribution

In this paper, we focus on the following special case of QCQP:

$$\begin{aligned}\text{(QCP)} \quad & \min f(x) = x^T Q_0 x + c_0^T x, \\ & \text{s.t. } g_j(x) = x^T Q_j x + c_j^T x + h_j \le 0, \ \ j = 1, \ldots, J \\ & \quad x \in \mathcal{F} = \{x \in \{0,1\}^n \mid Ax \le b\},\end{aligned} \tag{1}$$

where $Q_j, j = 0, 1, \ldots, J$ are symmetric $n \times n$ matrices, $c_0, c_j \in \Re^n, h_j \in \Re, j = 1, \ldots, J, A \in \Re^{m \times n}, b \in \Re^m$. Note that the constraint $Ax \le b$ may be viewed as m quadratic constraints each with zero quadratic part. The constraint $x \in \{0,1\}^n$ can be replaced by two groups of quadratic constraints $x_i^2 - x_i \le 0$ and $x_i^2 - x_i \ge 0, i = 1, \ldots, n$. Thus, problem (QCP) is a subclass of general QCQP.

Aside from the extensive discussion regarding the branching and/or bounding strategy by exploiting the special structure of the nonconvex QCQP in question, another interesting idea is to construct the mixed-integer quadratic convex reformulation (QCR) of a nonconvex mixed-integer QCQP. Once the QCR is obtained, the reformulation can be solved by off-the-shelf software which is capable of solving convex mixed-integer QCQPs. Ultimately, we succeed in solving nonconvex mixed-integer QCQP by the general-purpose solvers for mixed-integer convex QCQP. The QCR method was first studied in [15]. It was then extended to unconstrained 0–1 quadratic programming (QP) in [5], to equality constrained 0–1 QP in [8], and to the general mixed-integer QP in [6]. It was shown that the best QCR of mixed-integer QP, in the view of the tightness of continuous relaxation, could be obtained by the solution of an SDP relaxation of mixed-integer QP, [6].

For nonconvex mixed-integer QCQP, the quadratic convex reformulation can be obtained by introducing McCornick inequalities, [20], and extra $\mathcal{O}(n^2)$ variables, which leads to a large-scale problem. Furthermore, its best QCR, whose continuous relaxation bound is as tight as possible, can be found based on the dual solution to an SDP relaxation of the reformulation. Meanwhile, due to the introduction of inequalities and extra variables, the size of the SDP relaxation is large even though the original problem is of medium size. Thus, the application of the best QCR for nonconvex mixed-integer QCQP will be limited. One can refer to [7] for more details. Recently, another QCR method, which works only with original variables, for nonconvex binary QCQP was studied in [11]. Their continuous lower bound of the QCR was as tight as an SDP relaxation reduced from the Lagrangian dual of the original problem. The best QCR was also obtained by the dual solution of the SDP relaxation. However, as shown in Proposition 2 in [11], the price of the QCR is to introduce a new variable and two quadratic constraints if there are inequality quadratic constraints.

In fact, the main idea behind the existing QCR is to construct quadratic functions that are zeros at all points of the feasible region, add to the objective function and find corresponding parameters to make the new objective function convex. Thus, the question arises: Can we construct a QCR while involving quadratic functions which are not always zero at all points of the feasible region. In this paper, we succeeded in solving this problem by introducing a surrogate constraint. Our QCR approach has two steps. First, we construct surrogates of all the quadratic functions which are involved in our QCR. Second, we convexify all the quadratic inequalities using any desired method (such as the minimum eigenvalue method). Thus, in our QCR, the price is introducing a quadratic inequality, which is cheaper than that of [11]. The size of the QCR is therefore the same as that of the original problem, which is beneficial to the solution efficiency for the mixed-integer QCQP solvers. We also propose an SDP approach for finding the "best" QCR among our new family, in the sense that the continuous relaxation of this QCR will give the tightest lower bound for (QCP). We show that the lower bound obtained by the continuous relaxation of our "best" QCR is as tight as the Lagrangian bound of (QCP).

The main contributions of this paper are outlined below.

– We propose a QCR method by introducing a surrogate inequality constraint and convexifying all the inequality constraints in the resulting problem.
– We propose an SDP approach for constructing the "best" QCR in terms of the tightness of the lower bound of the corresponding continuous relaxation.
– We prove that the continuous relaxation of our "best" QCR provides a lower bound as tight as the Lagrangian bound of (QCP).

The rest of the paper is organized as follows. We first introduce in Sect. 2 our family of QCR for problem (QCP) via a surrogate constraint. The major obstacle in finding the best QCR among the family will also be presented in this section. To overcome this obstacle, we propose an SDP, the solution of which may be used to obtain the best QCR among our proposed family. In Sect. 3, the tightness of the bound obtained by the continuous relaxation of our best QCR is discussed. In Sect. 4, we conduct computational experiments to illustrate the improved performance of our best QCR. Finally, some concluding remarks are given in Sect. 5.

Notations Throughout the paper, the optimal value of problem $(\cdot)$ is denoted by $v(\cdot)$. We denote by $\Re^n_+$ the nonnegative orthant of $\Re^n$, I_n the identity matrix, and e the all-ones vector. For any $a \in \Re^n$, we denote by $\mathrm{Diag}(a) = \mathrm{Diag}(a_1, \ldots, a_n)$ the diagonal matrix with a_i being the ith diagonal element. We denote by S_n the set of $n \times n$ symmetric matrices, by S_n^+ the set of positive semidefinite matrices of S_n. Notation $M \succeq 0$ means $M \in S_n^+$. The standard inner product in S^n is defined as $A \bullet B = trace(AB) = \sum_{i,j=1}^n A_{ij} B_{ij}$. For a matrix $A \in S^n$, we also denote by $\mathrm{Diag}(A)$ the diagonal matrix with the ith diagonal entries being A_{ii}, and $\mathrm{diag}(A)$ the column vector with the ith element being A_{ii}. Finally, for any $a \in \Re$, we define that $a/0$ is equal to ∞ if $a > 0$, 0 if $a = 0$ and $-\infty$ if $a < 0$.

2 An SDP approach of finding the best QCR

In this section, we introduce a family of QCRs for problem (QCP). The QCR in this family involves quadratic functions which are not always zero at points of the feasible region. We also formulate the problem of finding the best QCR as an SDP.

2.1 A new family of quadratic convex reformulations for QCQP

In this section, we will consider the following equivalent formulation of problem (QCP):

$$
\begin{aligned}
(P) \qquad & \min\ f(x) = c_0^T x, \\
& \text{s.t. } g_j(x) = x^T Q_j x + c_j^T x + h_j \le 0, \quad j = 1, \ldots, J \\
& x \in \mathcal{F},
\end{aligned} \tag{2}
$$

where $Q_j \in S_n$ $(j = 1, \ldots, J)$, $c_0, c_j, h_j \in \Re^n$ $(j = 1, \ldots, J)$, and $\mathcal{F}$ is defined as in (QCP). Notice that our results can be generalized to the case with equality quadratic constraint due to the fact that any equality constraint can be reformulated as two inequality constraints with opposite signs. QCQP with quadratic objective function, i.e., $f(x) = x^T Q_0 x + c_0^T x$, can also be viewed as a subclass of problem (P) by the following transformation

$$\min\{c_0^T x + t \mid x^T Q_0 x \le t, \text{constraints}(2), x \in \mathcal{F}\}.$$

Denote by $(\overline{P})$ the continuous relaxation of problem (P), which is obtained by replacing $x \in \{0,1\}^n$ with $x \in [0,1]^n$ in $\mathcal{F}$. In this section, we assume the following constraint qualification for $(\overline{P})$.

Assumption 1 There is a (relative) interior point in the feasible set of $(\overline{P})$.

This assumption guarantees that if problem $(\overline{P})$ is convex, the strong duality between $(\overline{P})$ and its dual holds.

Now we turn our attention to the construction of the QCR for problem (P). The QCR consists of quadratic functions $g_j(x)$ for $j = 1, \dots, n$. Note that $g_j(x) \le 0$ for all feasible points of (P). We can not apply the idea of QCR directly. On the other hand, for any $u \in \Re^J_+$, it is easy to verify that problem (P) is equivalent to the following problem:

$$\begin{aligned} &\min\ c_0^T x, \\ &\text{s.t. constraints (2)}, \\ &\quad x^T \bar{Q} x + \bar{c}^T x + \bar{h} \le 0, \\ &\quad x \in \mathcal{F}, \end{aligned}$$

where

$$\bar{Q} = \sum_{j=1}^{J} u_j Q_j,\ \bar{c} = \sum_{j=1}^{J} u_j c_j,\ \bar{h} = \sum_{j=1}^{J} u_j h_j. \tag{3}$$

Let θ_j be the minimum eigenvalue of Q_j, for each $j = 1, \dots, J$. Consider the following mixed-integer convex programming problem:

$$\begin{aligned} (MP(u,w)) \qquad &\min\ c_0^T x, \\ &\text{s.t. } x^T(Q_j - \theta_j I_n)x + c_j^T x + h_j + \theta_j e^T x \le 0,\ j = 1, \dots, J \qquad (4) \\ &\quad x^T[\bar{Q} - \text{Diag}(w)]x + \bar{c}^T x + \bar{h} + w^T x \le 0, \\ &\quad x \in \mathcal{F}, \qquad (5) \end{aligned}$$

where $(u, w) \in \Omega$ with

$$\Omega = \left\{ (u, w) \in \Re^J_+ \times \Re^n \mid \sum_{j=1}^{J} u_j Q_j - \text{Diag}(w) \succeq 0 \right\}. \tag{6}$$

Obviously, constraints (4) and (5) are convex constraints according to the choice of parameters (u, w) and θ_js. Due to the fact that $x_i^2 = x_i$, $i = 1, \dots, n$, for all $x \in \mathcal{F}$, problem $(MP(u, w))$ is equivalent to problem (P). Thus, problem $(MP(u, w))$ is a quadratic convex formulation for problem (P) for any $(u, w) \in \Omega$, and can be solved by a general solver for integer quadratic convex problems.

Denote by $(\overline{MP}(u, w))$ the continuous relaxation of problem $(MP(u, w))$. Notice that $(\overline{MP}(u, w))$ is a tractable convex quadratic problem and will provide a lower bound for problem (P). Now we confront the issue of finding the "best" reformulation $(MP(u^*, w^*))$ in the sense that problem $(\overline{MP}(u^*, w^*))$ provides the tightest continuous lower bound among all $(u, w) \in \Omega$. This reformulation may be obtained by solving the problem,

$$(BR) \qquad \max_{(u,w)\in\Omega} v(\overline{MP}(u, w)).$$

If the optimal solution (u^*, w^*) to problem (BR) can be obtained, then the "best" QCR, $(MP(u^*, w^*))$, has a continuous relaxation with efficient lower bound. Thus, the "best" QCR, $(MP(u^*, w^*))$, is more promising to reach the global optima of problem (P) through the existing optimization software when compared with the general formulations.

2.2 The major obstacle of finding the "best" QCR

Due to the convexity of problem $(\overline{MP}(u, w))$ and Assumption 1, problem (BR) can be reduced to a new programming problem via strong duality and lift technique, as shown in the proof of Theorem 1.

Theorem 1 *Problem* (BR) *can be reduced to the following program*

$$
\begin{aligned}
(CP_0)\quad & \max_{u,w,s,\mu,\lambda,\eta,\tau} \; -b^T\mu + s\sum_{j=1}^{J} u_j h_j - e^T\lambda - \tau \\
& \text{s.t.} \begin{pmatrix} s\sum_{j=1}^{J} u_j Q_j - s\mathrm{Diag}(w) & \delta(u, w, s, \mu, \lambda, \eta)/2 \\ \delta(u, w, s, \mu, \lambda, \eta)^T/2 & \tau \end{pmatrix} \succeq 0, \\
& (u, s, \mu, \lambda, \eta) \geq 0, \\
& \sum_{j=1}^{J} u_j Q_j - \mathrm{Diag}(w) \succeq 0,
\end{aligned} \tag{7}
$$

where

$$\delta(u, w, s, \mu, \lambda, \eta) = c_0 + s\sum_{j=1}^{J} u_j c_j + sw + A^T\mu + \lambda - \eta. \tag{8}$$

Proof We first express $(\overline{MP}(u, w))$ by its dual form. Associate the following multipliers to the constraints in $(\overline{MP}(u, w))$:

- $\alpha \in \Re^J_+$ for constraints (4): $x^T(Q_j - \theta_j I_n)x + c_j^T x + h_j + \theta_j e^T x \leq 0,\ j = 1, \ldots, J$;
- $s \geq 0$ for constraint (5): $x^T[\bar{Q} - \mathrm{Diag}(w)]x + \bar{c}^T x + \bar{h} + w^T x \leq 0$;
- $\mu \in \Re^m_+$ for $Ax \leq b$;
- $\lambda \in \Re^n_+$ for $x \leq e$ and $\eta \in \Re^n_+$ for $x \geq 0$.

Let $d(\varpi)$ denote the Lagrangian dual function of $(\overline{MP}(u, w))$ with ϖ being the dual variables introduced above. The Lagrangian dual of $(\overline{MP}(u, w))$ is

$$\max\left\{d(\varpi) \mid (\alpha, s, \mu, \lambda, \eta) \geq 0\right\}. \tag{9}$$

Moreover, we obtain an explicit expression of the Lagrangian dual function,

$$
\begin{aligned}
d(\varpi) = \min_x \Bigg\{ & c_0^T x + \mu^T(Ax - b) - \eta^T x + \lambda^T(x - e) \\
& + s\{x^T[\bar{Q} - \mathrm{Diag}(w)]x + \bar{c}^T x + \bar{h} + w^T x\} \\
& + \sum_{j=1}^{J} \alpha_j[x^T(Q_j - \theta_j I_n)x + c_j^T x + h_j + \theta_j e^T x] \Bigg\}
\end{aligned}
$$

$$
\begin{aligned}
&= \min_x \left\{ x^T \left\{ \sum_{j=1}^J \alpha_j (Q_j - \theta_j I_n) + s \left[\sum_{j=1}^J u_j Q_j - \mathrm{Diag}(w) \right] \right\} x \right. \\
&\quad + \left[c_0 + \sum_{j=1}^J \alpha_j (c_j + \theta_j e) + s \sum_{j=1}^J u_j c_j + sw + A^T \mu + \lambda - \eta \right]^T x \\
&\quad \left. + \left[-b^T \mu + \sum_{j=1}^J \alpha_j h_j + s \sum_{j=1}^J u_j h_j - e^T \lambda \right] \right\} \\
&= \min_x \left\{ s x^T \left[\sum_{j=1}^J u_j Q_j - \mathrm{Diag}(w) \right] x + \left[c_0 + s \sum_{j=1}^J u_j c_j + sw + A^T \mu + \lambda - \eta \right]^T x \right. \\
&\quad \left. + \left[-b^T \mu + s \sum_{j=1}^J u_j h_j - e^T \lambda \right] \right\} \\
&= -b^T \mu + s \sum_{j=1}^J u_j h_j - e^T \lambda + \min_x \widetilde{q}(x),
\end{aligned}
$$

where the second equality holds due to (3) and the third equality holds due to the fact that variable α_j and $\alpha_j \theta_j$ can be absorbed by u_j and w_j, $j = 1, \ldots, J$, respectively, and $\widetilde{q}(x) = s x^T [\sum_{j=1}^J u_j Q_j - \mathrm{Diag}(w)] x + \delta(u, w, s, \mu, \lambda, \eta)^T x$ with $\delta(u, w, s, \mu, \lambda, \eta)$ defined in (8). Thus, the dual problem (9) can be written as

$$
\begin{aligned}
&\max \left\{ -b^T \mu + s \sum_{j=1}^J u_j h_j - e^T \lambda - \tau \right\}, \\
&\text{s.t. } (s, \mu, \eta, \lambda) \geq 0, \\
&\quad s x^T \left[\sum_{j=1}^J u_j Q_j - \mathrm{Diag}(w) \right] x + \delta(u, w, s, \mu, \lambda, \eta)^T x \geq -\tau, \ \forall x \in \Re^n. \qquad (10)
\end{aligned}
$$

Note that (10) is equivalent to constraint (7). Therefore, the dual problem (9) can be reduced to the following problem

$$
(D_s(u, w)) \quad \max \left\{ -b^T \mu + s \sum_{j=1}^J u_j h_j - e^T \lambda - \tau \mid (7), (s, \mu, \lambda, \eta) \geq 0 \right\}.
$$

By Assumption 1 and the conic duality theorem ([26]), the strong duality between $(\overline{MP}(u, w))$ and $(D_s(u, w))$ holds. Therefore, problem (BR) is equivalent to

$$
\max \left\{ v(D_s(u, w)) \mid u \geq 0, \sum_{j=1}^J u_j Q_j - \mathrm{Diag}(w) \succeq 0 \right\},
$$

which is problem (CP_0). □

Note that there are nonlinear parts in problem (CP_0), i.e., $s \sum_{j=1}^J u_j h_j$ in the objective function and constraint (7). Thus problem (CP_0) is nonconvex and can not provide us with the value of the parameters of the "best" QCR in a short time.

2.3 An SDP approach for finding the "best" QCR

The major obstacle to finding the "best" QCR for problem (P) polynomially has been described in the previous subsection. In this section, we propose an SDP approach for polynomially finding the "best" QCR among the proposed QCR family, which will be shown in the following Theorem 2. Theorem 2 enables us to efficiently find the optimal solution to problem (*BR*) via an SDP problem. That is, based on an SDP approach, the "best" QCR for (P) can be found polynomially.

Theorem 2 *Consider the following SDP problem:*

$$(CP) \qquad \max_{u,w,\mu,\lambda,\eta,\tau} \; -b^T\mu + \sum_{j=1}^{J} u_j h_j - e^T\lambda - \tau$$

$$\text{s.t.} \quad \begin{pmatrix} \sum_{j=1}^{J} u_j Q_j - \mathrm{Diag}(w) & \vartheta(u,w,\mu,\lambda,\eta)/2 \\ \vartheta(u,w,\mu,\lambda,\eta)^T/2 & \tau \end{pmatrix} \succeq 0, \tag{11}$$

$$(u,\mu,\lambda,\eta) \geq 0, \tag{12}$$

where

$$\vartheta(u,w,\mu,\lambda,\eta) = c_0 + \sum_{j=1}^{J} u_j c_j + w + A^T\mu + \lambda - \eta. \tag{13}$$

Let $(u^*, w^*, \mu^*, \lambda^*, \eta^*, \tau^*)$ be an optimal solution to problem (CP). Then (u^*, w^*) is an optimal solution to problem (BR).

Proof Denote $(\hat{u}, \hat{w}, \hat{s}, \hat{\mu}, \hat{\lambda}, \hat{\eta}, \hat{\tau})$ as an optimal solution to problem (CP_0). Let $\bar{u} = \hat{s}\hat{u}$, $\bar{w} = \hat{s}\hat{w}$. Then $(\bar{u}, \bar{w}, \hat{\mu}, \hat{\lambda}, \hat{\eta}, \hat{\tau})$ is feasible to problem (CP) and

$$v(CP) \geq -b^T\hat{\mu} + \sum_{j=1}^{J} \bar{u}_j h_j - e^T\hat{\lambda} - \hat{\tau} = -b^T\hat{\mu} + \hat{s}\sum_{j=1}^{J} \hat{u}_j h_j - e^T\hat{\lambda} - \hat{\tau} = v(CP_0).$$

Thus, it suffices to prove that there exists an s^* such that $(u^*, w^*, s^*, \mu^*, \lambda^*, \eta^*, \tau^*)$ is also an optimal solution to problem (CP_0).

Due to (11) and (12), by letting $s^* = 1$, $(u^*, w^*, s^*, \mu^*, \lambda^*, \eta^*, \tau^*)$ is feasible for problem (CP_0). We have

$$\begin{aligned} v(CP_0) &\geq -b^T\mu^* + s^*\sum_{j=1}^{J} u_j^* h_j - e^T\lambda^* - \tau^* \\ &= -b^T\mu^* + \sum_{j=1}^{J} u_j^* h_j - e^T\lambda^* - \tau^* \\ &= v(CP). \end{aligned} \tag{14}$$

Thus, $v(CP) = v(CP_0)$. Furthermore, (14) holds with equality throughout and solution $(u^*, w^*, s^*, \mu^*, \lambda^*, \eta^*, \tau^*)$ with $s^* = 1$ is exactly an optimal solution to problem (CP_0). □

Remark 1 Note that the solution of problem (*BR*), which can be obtained by solving SDP problem (CP) in Theorem 2, depends heavily on the input parameters of problem (P). If some variables in problem (P) are fixed to 0 or 1, we will have different SDP problems (CP)

and then different parameters (u, w). That means that when we apply a branch-and-bound algorithm to the "best" QCR, $(MP(u^*, w^*))$, we can not guarantee that it is still the "best" QCR at nodes other than the root node.

3 Tightness of the continuous relaxation bound of the "best" QCR

In this section, the tightness of the continuous relaxation bound of our "best" QCR will be discussed. As shown in the following Theorem 3, the bound obtained by the continuous relaxation of our "best" QCR is as tight as the Lagrangian bound of (P).

For problem (P), it has been shown that its Lagrangian dual can be reduced to the following SDP problem ([29]):

$$\begin{aligned}(SDP_0)\quad & \min\ c_0^T x\\ & \text{s.t. } Q_j \bullet X + c_j^T x + h_j \le 0,\ j = 1, \cdots, J,\\ & \quad \operatorname{diag}(X) = x,\\ & \quad Ax \le b,\ 0 \le x \le e,\\ & \quad \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0.\end{aligned}$$

The following theorem shows that the continuous relaxation bound of our "best" QCR is as tight as (SDP_0).

Theorem 3 $v(SDP_0) = v(BR)$.

Proof Due to Theorem 2, it suffices to prove $v(CP) = v(SDP_0)$. We first express problem (CP) by its dual form. Associate the following multiplier $\phi = \begin{pmatrix} X & x \\ x^T & t \end{pmatrix} \in S_+^{n+1}$ for constraint (11) in (CP). Let $\widetilde{d}(\phi)$ denote the Lagrangian dual function of (CP). Then the Lagrangian dual of (CP) is

$$\min \left\{\widetilde{d}(\phi) \mid \phi \succeq 0\right\}. \tag{15}$$

An explicit expression of the Lagrangian dual function is,

$$\begin{aligned}\widetilde{d}(\phi) &= \max_{(u,w,\mu,\lambda,\eta,\tau)\ge 0} \left\{ -b^T\mu + \sum_{j=1}^J u_j h_j - e^T\lambda - \tau \right.\\ &\qquad \left. + \begin{pmatrix} X & x \\ x^T & t \end{pmatrix} \bullet \begin{pmatrix} \sum_{j=1}^J u_j Q_j - \operatorname{Diag}(w) & \vartheta(u, w, \mu, \lambda, \eta)/2 \\ \vartheta(u, w, \mu, \lambda, \eta)^T/2 & \tau \end{pmatrix} \right\}\\ &= c_0^T x + \max_{\tau\in\Re}\{(-1+t)\tau\} + \max_{\mu\ge 0}\{\mu^T(Ax - b)\} + \max_{\eta\ge 0}\{-\eta^T x\} + \max_{\lambda\ge 0}\{\lambda^T(x - e)\}\\ &\qquad + \sum_{j=1}^J \max_{u_j\ge 0}\left\{u_j(Q_j \bullet X + c_j^T x + h_j)\right\} + \max_{w\in\Re^n}\left\{\sum_{i=1}^n w_i(-X_{ii} + x_i)\right\}\\ &= \begin{cases} c_0^T x, \ \text{if } \begin{cases} 0 \le x \le e,\ Ax \le b,\ \operatorname{diag}(X) = x, t = 1, \\ Q_j \bullet X + c_j^T x + h_j \le 0,\ j = 1, \ldots, J \end{cases} \\ +\infty, \ \text{otherwise} \end{cases}\end{aligned}$$

Thus, the dual of (CP) has the formulation

$$\begin{aligned} &\min c_0^T x \\ &\text{ s.t. } Q_j \bullet X + c_j^T x + h_j \leq 0, j = 1, \ldots, J \\ &\quad \operatorname{diag}(X) = x, \\ &\quad Ax \leq b, \ 0 \leq x \leq e, \\ &\quad \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0, \end{aligned}$$

which is (SDP_0). Due to Assumption 1 and the conic duality theorem ([26]), the strong duality between (CP) and its dual holds. Therefore, $v(CP) = v(SDP_0)$. □

Remark 2 According to Theorem 3, the continuous relaxation of our "best" QCR is equal to the optimal value of (SDP_0). Thus, in a branch-and-bound algorithm for solving the "best" QCR, $(MP(u^*, w^*))$, the continuous relaxation gap at the root node is the same as the SDP-relaxation gap, which is known to be strong. Although our "best" QCR is no longer the "best" QCR at other nodes of the branch-and-bound algorithm, our preliminary results in Sect. 4 show that the performance of our "best" QCR is much more efficient.

The following example illustrates our "best" QCR of binary QCQP and the tightness of the continuous relaxation bound of our "best" QCR.

Example 1 Consider the following example of [11] with $n = 3$:

$$\begin{aligned} (EP) \ \ \min \ & -x_1 - x_2 - x_3 \\ \text{s.t. } & x \in \{0, 1\}^3, \\ & x_1x_2 \leq 0, \ x_1x_3 \leq 0, \ x_2x_3 \leq 0. \end{aligned} \tag{16}$$

It has been shown in [11] that $v(EP) = -1$ and the optimal value of its dual (SDP_0) is -1 as well.

Denote

$$Q_1 = \begin{pmatrix} 0 & 0.5 & 0 \\ 0.5 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \ Q_2 = \begin{pmatrix} 0 & 0 & 0.5 \\ 0 & 0 & 0 \\ 0.5 & 0 & 0 \end{pmatrix}, \ Q_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.5 \\ 0 & 0.5 & 0 \end{pmatrix},$$

and $c_0 = (-1, -1, -1)^T$. According to Theorem 2, the optimal (u^*, w^*) for the "best" QCR of (EP) can be generated by solving the following SDP,

$$\begin{aligned} \max \ & -e^T\lambda - \tau, \\ \text{s.t. } & \begin{pmatrix} u_1Q_1 + u_2Q_2 + u_3Q_3 - Diag(w) & (c + w + \lambda - \eta)/2 \\ (c + w + \lambda - \eta)^T/2 & \tau \end{pmatrix} \succeq 0, \\ & (u, \lambda, \eta) \in \Re^3_+ \times \Re^3_+ \times \Re^3_+, \ (w, \tau) \in \Re^3 \times \Re. \end{aligned}$$

We get $u^* = (2, 2, 2)^T$ and $w^* = (-1, -1, -1)^T$. It is easily verified that the minimum eigenvalue of matrix Q_j is $\theta_j = -0.5$, for $j = 1, \ 2, \ 3$. Thus, the "best" QCR has the following formulation,

$$
\begin{aligned}
(REP)\quad \min\ & -x_1 - x_2 - x_3 \\
\text{s.t. } & x \in \{0,1\}^3, \\
& 0.5x_1^2 + 0.5x_2^2 + 0.5x_3^2 + x_1x_2 - 0.5x_1 - 0.5x_2 - 0.5x_3 \le 0, \\
& 0.5x_1^2 + 0.5x_2^2 + 0.5x_3^2 + x_1x_3 - 0.5x_1 - 0.5x_2 - 0.5x_3 \le 0, \\
& 0.5x_1^2 + 0.5x_2^2 + 0.5x_3^2 + x_2x_3 - 0.5x_1 - 0.5x_2 - 0.5x_3 \le 0, \\
& x_1^2 + x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3 + 2x_2x_3 - x_1 - x_2 - x_3 \le 0.
\end{aligned}
$$

Denote by $(\overline{REP})$ the continuous relaxation of problem (REP). It can be calculated that $v(\overline{REP}) = -1 = v(SDP_0)$.

4 Numerical analysis

In this section, we conduct computational experiments to demonstrate the effectiveness of the "best" QCR obtained from SDP approach for problem (P).

4.1 Test problems

In our experiment, the testing problems are instances of the following 0–1 QCQP problem:

$$
\begin{aligned}
(TP)\ \min\ & c_0^T x \\
\text{s.t.} & x^T Q_j x + c_j^T x + h_j \le 0,\ \ j = 1, \ldots, J, \\
& x \in \{0,1\}^n.
\end{aligned}
$$

In our test, the data of test problems of (TP) are randomly generated in the following fashion. We set $J = 10$. The entries of c_j are integers uniformly drawn at random from intervals $[-8, 18]$, for each $j = 0, 1, \ldots, J$. We generate Q_j, $j = 1, \ldots, J$, in the same fashion as [22]. For a fixed n, we generate four types of Q_j matrices, classified by different density, i.e., the percentage of nonzero entries. The nonzero elements of matrices Q_j are integers uniformly drawn at random from intervals $[-100, 10]$. We set $h_j = -\lceil 0.5e^T c_j + 0.25e^T Q_j e \rceil$, $j = 1, \ldots, J$. Thus, the optimal value of problem (TP) is guaranteed for our test problems.

4.2 Implementation issues

We compare the performance of the perspective reformulations $(MP(u, w))$ using the following three choices of (u, w):

- (MP_e): the reformulation with $u = e$ and w being the minimum eigenvalue of matrix $\sum_{j=1}^J Q_j$;
- (MP_s): the reformulation with $u = e$ and w calculated by the following small SDP

$$
\begin{aligned}
(SDP_s)\quad \max\ & e^T w \\
\text{s.t. } & \sum_{j=1}^{J} Q_j - \mathrm{Diag}(w) \succeq 0.
\end{aligned}
$$

- (MP_l): the reformulation with (u, w) computed by (CP) in Theorem 2.

The numerical tests were implemented in Matlab R2014a (64 bit) and run on a personal computer equipped with Intel Pentium CPU (3.6GHz) and 16 GB RAM. CVX 1.2 ([14]), which is a MATLAB-based modeling system for convex optimization, is used to model the SDP problems (SDP_s) and (CP). The SDP problems are solved by SeDuMi 1.2 within CVX. All binary QCQP problems are solved by the mixed integer QCQP solver in CPLEX 12.6. CPLEX is called through C++ under Microsoft Visual Studio 2012. We use the default settings of CPLEX 12.6 except that we set the computing time limited to 10,000 s and the number of parallel threads to be single.

4.3 Numerical results

We first compare the continuous bounds of $(MP(u, w))$ with (u, w) computed by (SDP_s) and (CP), respectively, for problem (TP). The comparison results are reported in Table 1. The column "d" in Table 1 denotes the density of the matrices $Q_j s$. We denote by b_s and b_l the continuous bounds of $(MP(u, w))$ with (u, w) computed by (SDP_s) and (CP), respectively. We define "imp.ratio(%)" as the (relative) improvement ratio of b_l over b_s:

$$\text{imp.ratio}(\%) = (b_l - b_s)/|b_s| \times 100.$$

Table 1 Comparison of continuous bounds for (TP)

n	d	b_s	b_l	Imp. ratio (%)	n	d	b_s	b_l	Imp. ratio (%)
25	0.25	−12.77	−8.18	48.48	50	0.25	−29.90	−22.45	39.42
25	0.50	−1.22	2.96	29.48	50	0.50	−33.81	−27.85	37.48
25	0.75	−17.87	−15.00	67.36	50	0.75	−41.96	−36.58	25.10
25	1.00	−7.56	−4.04	52.05	50	1.00	−32.12	−26.65	20.12
75	0.25	−50.35	−40.50	30.29	100	0.25	−75.31	−63.29	17.74
75	0.50	−58.28	−49.45	16.56	100	0.50	−59.09	−46.85	31.29
75	0.75	−65.96	−58.66	14.48	100	0.75	−66.22	−54.10	20.97
75	1.00	−42.13	−32.72	40.09	100	1.00	−67.71	−57.83	17.91
125	0.25	−77.93	−62.66	21.63	150	0.25	−90.27	−72.17	21.87
125	0.50	−78.79	−63.75	23.24	150	0.50	−98.70	−81.35	18.36
125	0.75	−80.44	−65.96	20.92	150	0.75	−93.71	−77.37	22.87
125	1.00	−89.13	−76.33	19.54	150	1.00	−94.28	−77.63	19.05
175	0.25	−119.89	−98.42	23.32	200	0.25	−132.87	−107.26	20.46
175	0.50	−106.53	−86.23	37.50	200	0.50	−131.26	−106.90	22.63
175	0.75	−142.36	−123.62	13.75	200	0.75	−108.60	−84.68	46.87
175	1.00	−156.63	−140.19	10.66	200	1.00	−152.03	−130.58	14.93
225	0.25	−151.21	−123.39	24.16	250	0.25	−130.42	−97.72	30.36
225	0.50	−140.44	−112.35	24.98	250	0.50	−151.62	−121.30	24.44
225	0.75	−165.77	−138.98	20.38	250	0.75	−183.13	−156.02	15.55
225	1.00	−164.76	−139.84	16.19	250	1.00	−172.12	−143.81	18.49
275	0.25	−147.57	−113.93	85.44	300	0.25	−197.75	−160.81	20.37
275	0.50	−220.14	−190.88	14.29	300	0.50	−188.98	−152.60	20.83
275	0.75	−154.02	−120.18	22.82	300	0.75	−230.75	−199.27	16.81
275	1.00	−192.33	−164.62	15.96	300	1.00	−195.29	−159.71	19.50

All the results in Table 1 are the average of five instances for each n and density d. We see from Table 1 that the bound of continuous relaxation of the "best" QCR has a huge improvement over the QCR ($MP(u, w)$) with (u, w) computed by (SDP_s), for different sizes and densities. For the instances with $n = 275$ and $d = 0.25$, the average improvement ratio is up to 85.44%. The smallest average improvement ratio in this table is 10.66%, which is for the instances with $n = 175$ and $d = 1$. All the results in Table 1 indicate that the continuous bound of the "best" QCR with (u, w) computed by (CP), as in Theorem 2, may lead to more accurate applications of branch and bound algorithm than the two other methods.

We furthermore conduct the computation comparison of different QCRs (MP_e), (MP_s) and (MP_l) for the instances with different size and density. Here we also add the QCR derived according to Proposition 2 in [11], which we denote (QCR), to our comparison. The average comparison results of five instances for each size and density are summarized in Table 2. The first column in the table denotes the problem size n. In second column, "d" denotes the density of matrix $Q_j s$, which is same as in Table 2. The columns "Time_s" and "Time_l" are the computation time (in seconds) for finding parameter vector (u, w) via solving SDP problems (SDP_s) and (CP) by CVX, respectively. The column "Time_QCR" is the computation time for finding parameters in (QCR) by solving SDP problem. The column "gap" is the output parameter of CPLEX, which measures the relative gap (in percentage) of the incumbent solution when CPLEX is terminated. Note that the default tolerance of the relative gap in CPLEX is 0.01%. The last columns "time" and "nodes" are the computing time (in seconds) and the number of nodes explored by CPLEX, respectively.

In Table 2, the columns "Time_QCR", "Time_s" and "Time_l" list the time for computing the parameters required in formulations (QCR), (MP_s) and (MP_l). Since these parameters are computed only at the root node, the computing time influences little on the solution efficiency of branch-and-bound procedure. From Table 2, we observe that, for the instance with size $n = 25$ and 50, CPLEX generally terminates with a global solution within 10,000 s. The computing time for (MP_e), (MP_s) and (MP_l) are similar, while CPLEX takes more time for solving formulation (QCR). For instances with $n = 75$, the branch and bound procedure for reformulation (MP_l) by CPLEX will converge in a shorter time when compared with reformulations (QCR), (MP_e) and (MP_s). When problem size $n \geq 100$, CPLEX terminates with global solutions only for the formulation (MP_l) for some instances with $(n, d) =$ (100, 1.00) and (125, 1.00), while it can not converge within the limit of 10,000 s for the other formulations and other instances. Comparing the results displayed in Table 2, we also notice that, for most of instances with $n \geq 100$, the lower bound for (MP_l) when CPLEX stops has a smallest gap to the optimal value among the four reformulations. Especially for all testing instances with $n = 150$ and 200, the average gaps for (MP_l) have obvious advantage by contrast to the other three formulations. It means that formulation (MP_l) could provide approximate solutions with "best" quality for problem (P) among the four formulations at a same cost of computing time. What needs to mention is that, for one of the five instances with $(n, d) = (200, 0.25)$, formulation (QCR) cannot find any feasible solution within 10,000 s, and we do not give the average gap and node in the table. Only for a few groups of instances, the formulation (QCR) has smaller gaps than our "best" QCR (MP_l), for example, $(n, d) =$ (175, 0.50) and (175, 0.75). Based on all the observations, the formulation (MP_l) is generally more promising to find a global solution with a shorter time or an approximate solution with smaller gap with a same time limit. This could be due to the fact that the quality of lower bound generated by the continuous relaxation of (MP_l) has a positive and significant impact on the efficiency of branch and bound procedure.

Table 2 Comparison results of different QCRs for problem (*TP*)

n	d	Time_{QCR}	Time_{s}	Time_{l}	(QCR)			(MP_{e})			(MP_{s})			(MP_{l})		
					Gap	Time	Nodes	Gap	Time	Nodes	Gap	Time	Nodes	Gap	Time	Nodes
25	0.25	0.34	0.14	0.49	0.00	166	1895	0.00	19	4410	0.00	19	4200	0.00	24	508
25	0.50	0.32	0.13	0.45	0.00	657	9064	0.00	446	16, 481	0.00	263	10, 788	0.00	139	8291
25	0.75	0.29	0.16	0.42	0.00	123	1409	0.00	9	601	0.00	9	567	0.00	14	787
25	1.00	0.28	0.13	0.40	0.00	120	983	0.00	18	478	0.00	18	526	0.00	21	302
50	0.25	1.06	0.63	1.24	29.75	8069	11, 582	0.00	3135	8570	0.00	3032	8604	0.00	2629	7449
50	0.50	1.36	0.56	1.18	22.10	5705	7628	11.52	3323	17, 499	9.32	2890	14, 968	7.70	3000	7135
50	0.75	1.15	0.50	1.42	7.77	2825	17, 620	0.00	2310	5188	0.00	1999	4815	0.00	1522	3691
50	1.00	1.19	0.48	1.42	6.97	5308	8504	2.79	3011	7091	2.75	3054	6824	2.32	3009	8598
75	0.25	1.46	0.46	1.73	39.43	10, 000	3711	53.28	8446	6240	51.79	8212	7130	28.30	7218	6688
75	0.50	1.44	0.41	1.67	12.41	10, 000	3602	18.09	10, 000	5725	20.71	10, 000	5595	9.62	6576	55, 118
75	0.75	1.50	0.43	1.67	13.54	8571	2879	19.32	6281	3344	23.31	6172	3637	6.61	4598	2870
75	1.00	1.43	0.40	1.64	10.11	7288	3464	21.40	8486	4998	26.07	8526	4576	8.73	5306	48, 157
100	0.25	2.68	1.34	3.12	15.65	10, 000	1657	120.64	10, 000	2051	38.09	10, 000	2395	14.69	10, 000	2619
100	0.50	2.59	1.34	3.07	34.98	10, 000	1577	61.79	10, 000	2033	56.48	10, 000	2096	29.72	10, 000	43, 905
100	0.75	2.66	1.44	2.95	34.59	10, 000	1503	87.30	10, 000	2183	66.20	10, 000	2579	30.89	10, 000	60, 629
100	1.00	2.83	1.39	3.10	313.47	10, 000	2126	139.29	10, 000	2728	144.17	10, 000	2567	29.58	9903	3070
125	0.25	3.70	1.65	4.27	75.17	10, 000	893	188.47	10, 000	1304	187.98	10, 000	1624	75.84	10, 000	1702
125	0.50	3.07	1.59	3.72	250.84	10, 000	845	157.01	10, 000	1135	292.40	10, 000	1335	84.67	10, 000	1224

Table 2 continued

n	d	Time_{QCR}	Time_{s}	Time_{l}	(QCR)			(MP_{e})			(MP_{s})			(MP_{l})		
					Gap	Time	Nodes	Gap	Time	Nodes	Gap	Time	Nodes	Gap	Time	Nodes
125	0.75	3.34	1.66	3.76	532.60	10, 000	649	183.74	10, 000	983	91.74	10, 000	1312	45.00	10, 000	1336
125	1.00	3.55	1.74	4.22	39.00	10, 000	1026	308.23	8972	1689	192.22	8978	1735	73.33	8226	1559
150	0.25	4.39	2.05	5.19	96.53	10, 000	535	1134.15	10, 000	556	93.52	10, 000	773	40.58	10, 000	883
150	0.50	3.91	2.00	4.67	46.77	10, 000	423	77.34	10, 000	600	81.78	10, 000	699	29.43	10, 000	765
150	0.75	4.27	1.96	4.85	163.26	10, 000	374	156.90	10, 000	695	976.74	10, 000	743	67.84	10, 000	575
150	1.00	4.44	1.95	5.08	31.94	10, 000	560	61.37	10, 000	825	54.76	10, 000	885	27.21	10, 000	942
175	0.25	5.40	2.26	6.69	85.14	10, 000	324	178.54	10, 000	428	102.03	10, 000	463	45.54	10, 000	482
175	0.50	4.86	2.33	5.81	64.61	10, 000	227	1234.22	10, 000	416	188.95	10, 000	446	97.06	10, 000	461
175	0.75	5.05	2.07	5.81	43.57	10, 000	23, 306	86.61	10, 000	432	38.15	10, 000	462	52.03	10, 000	460
175	1.00	5.14	2.25	6.05	12.98	10, 000	361	22.76	10, 000	548	25.48	10, 000	597	6.96	10, 000	41, 718
200	0.25	6.45	2.52	7.71	-	10, 000	-	174.78	10, 000	267	100.77	10, 000	368	54.18	10, 000	390
200	0.50	5.76	2.50	6.69	870.36	10, 000	182	437.69	10, 000	255	161.23	10, 000	268	49.33	10, 000	52, 145
200	0.75	5.90	2.38	7.00	103.65	10, 000	162	133.65	10, 000	278	157.65	10, 000	318	65.69	10, 000	327
200	1.00	6.09	2.38	7.31	26.19	10, 000	238	35.09	10, 000	365	36.89	10, 000	382	10.16	10, 000	422

5 Concluding remarks

In this paper we considered binary QCQP problem which is a special class of nonconvex QCQP problem. We proposed a family of QCR involving quadratic surrogate functions which are not always zero at all points of the feasible region. Our QCR method had two steps. First, we construct surrogates of all the quadratic functions which are involved in our QCR. Second, we convexify all the quadratic inequalities using any desired method (such as the minimum eigenvalue method). Thus, in our QCR, the price is the introduction of a surrogate quadratic inequality. The "best" quadratic convex reformulation among the proposed family of QCRs, in the sense that the lower bound of the corresponding continuous relaxation is tightest, can be obtained by solving an SDP. To illustrate the performance of this "best" QCR, we conducted numerical experiments where its performance was compared with some QCRs in literature. The numerical results indicated that, although it was "best" only at the root node during the procedure of branch and bound algorithm, the "best" QCR indeed improved the solution efficiency for the testing instances with small size, while for the instance with large scale, the "best" QCR can generally help find a solution with lower gap within the permitted computing time. Furthermore, Our idea in this paper can be parallelled to get a tighter QCR if other quadratic valid inequalities for problem (QCP) are involved. We will do further discussion on how to construct quadratic inequalities in our future research.

Acknowledgements The authors would like to thank three anonymous referees for their constructive suggestions and insightful comments, which helped improve the paper substantially.

References

1. Al-Khayyal, F.A., Larsen, C., Van Voorhis, T.: A relaxation method for nonconvex quadratically constrained quadratic programs. J. Global Optim. **6**, 215–230 (1995)
2. Anstreicher, K.M.: Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. J. Global Optim. **43**, 471–484 (2009)
3. Audet, C., Hansen, P., Jaumard, B., Savard, G.: A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. Math. Program. **87**, 131–152 (2000)
4. Beck, A., Eldar, Y.C.: Strong duality in nonconvex quadratic optimization with two quadratic constraints. SIAM J. Optim. **17**, 844–860 (2007)
5. Billionnet, A., Elloumi, S.: Using a mixed integer quadratic programming solver for the unconstrained quadratic 0–1 problem. Math. Program. **109**, 55–68 (2007)
6. Billionnet, A., Elloumi, S., Lambert, A.: Extending the QCR method to general mixed-integer programs. Math. Program. **131**, 381–401 (2012)
7. Billionnet, A., Elloumi, S., Lambert, A.: Exact quadratic convex reformulations of mixed-integer quadratically constrained problems. Math. Program. (2015). https://doi.org/10.1007/s10107-015-0921-2
8. Billionnet, A., Elloumi, S., Plateau, M.: Improving the performance of standard solvers for quadratic 0–1 programs by a tight convex reformulation: the QCR method. Discrete Appl. Math. **157**, 1185–1197 (2009)
9. Burer, S., Saxena, A.: Old wine in new bottle: the milp road to miqcp. Technical Report, Department of Management Sciences University of Iowa (2009). http://www.optimization-online.org/DB_FILE/2009/07/2338.pdf
10. Cui, X.T., Zheng, X.J., Zhu, S.S., Sun, X.L.: Convex relaxations and miqcqp reformulations for a class of cardinality-constrained portfolio selection problems. J. Global Optim. **56**, 1409–1423 (2013)
11. Galli, L., Letchford, A.: A compact variant of the QCR method for quadratically constrained quadratic 0–1 programs. Optim. Lett. **8**, 1213–1224 (2014)
12. Garey, M., Johnson, D.: Computers and Intractability: An Guide to the Theory of NP-Completeness. W.H. Freeman, San Francisco (1979)
13. Gower, J.: Euclidean distance geometry. Math. Sci. **7**(1), 1–14 (1982)

14. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 2.1, (2014). http://cvxr.com/cvx
15. Hammer, P., Rubin, A.: Some remarks on quadratic programming with 0–1 variables. RIRO **3**, 67–79 (1970)
16. Kim, S., Kojima, M.: Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations. Comput. Optim. Appl. **26**, 143–154 (2003)
17. Klose, A., Drexl, A.: Facility location models for distribution system design. Europ. J. Oper. Res. **162**, 4–29 (2005)
18. Kolbert, F., Wormald, L.: Robust portfolio optimization using second-order cone programming. In: Satchell, S. (ed.) Optimizing Optimization: The Next Generation of Optimization Applications & Theory, pp. 3–22. Academic Press and Elsevier, Amsterdam (2010)
19. Linderoth, J.: A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. Math. Program. **103**, 251–282 (2005)
20. McCormick, G.: Computability of global solutions to factorable nonconvex programs: part Iconvex underestimating problems. Math. program. **10**, 147–175 (1976)
21. Nesterov, Y., Nemirovsky, A.: Interior-Point Polynomial Methods in Convex Programming. SIAM, Philadelphia (1994)
22. Pardalos, P.M., Rodgers, G.P.: Computational aspects of a branch and bound algorithm for quadratic zero–one programming. Computing **45**, 131–144 (1990)
23. Raber, U.: A simplicial branch-and-bound method for solving nonconvex all-quadratic programs. J. Global Optim. **13**, 417–432 (1998)
24. Saxena, A., Bonami, P., Lee, J.: Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. Math. Program. **130**, 359–413 (2011)
25. Sherali, H.D., Adams, W.P.: A Reformulation–Linearization Technique for Solving Discrete and Continuous Nonconvex Problems. Kluwer Academic Publishers, Dordrecht (1999)
26. Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Rev. **38**, 49–95 (1996)
27. Ye, Y., Zhang, S.: New results on quadratic minimization. SIAM J. Optim. **14**, 245–267 (2004)
28. Zhang, S.: Quadratic maximization and semidefinite relaxation. Math. Program. **87**, 453–465 (2000)
29. Zheng, X.J., Sun, X.L., Li, D.: Convex relaxations for nonconvex quadratically constrained quadratic programming: matrix cone decomposition and polyhedral approximation. Math. program. **129**, 301–329 (2011)
30. Zheng, X.J., Sun, X.L., Li, D.: Nonconvex quadratically constrained quadratic programming: best DC decompositions and their SDP representations. J. Global Optim. **50**, 695–712 (2011)