A LOW-DIMENSIONAL SDP RELAXATION BASED SPATIAL BRANCH AND BOUND METHOD FOR NONCONVEX QUADRATIC PROGRAMS

Jing Zhou
Department of Applied Mathematics, College of Science
Zhejiang University of Technology
Hangzhou, Zhejiang, 310032, China

Zhibin Deng∗
School of Economics and Management
University of Chinese Academy of Sciences
Key Laboratory of Big Data Mining and Knowledge Management
Chinese Academy of Sciences
Beijing, 100190, China

(Communicated by Wenxun Xing)

Abstract. In this paper, we propose a novel low-dimensional semidefinite programming (SDP) relaxation for convex quadratically constrained nonconvex quadratic programming problems. This new relaxation is derived by simultaneous matrix diagonalization method under the difference of convex decomposition scheme. The highlight of the relaxation is the low dimensionality of the positive semidefinite constraint, which only depends on the number of negative eigenvalues in the objective function. We prove that a mixed SOCP and SDP relaxation appeared in the literature is equivalent to the proposed relaxation, while the proposed relaxation has fewer constraints. We also provide conditions under which the proposed relaxation is as tight as the classical SDP relaxation and provides an optimal value for the original problem. For general cases, a spatial branch-and-bound algorithm is designed for finding a global optimal solution. Extensive numerical experiments support that the proposed algorithm outperforms two cutting-edge algorithms when the problem size is medium or large and the number of negative eigenvalues in the nonconvex objective function is relatively small.

1. Introduction. In this paper, we study the nonconvex quadratic optimization with convex quadratically constraints in the following form:

\[
\begin{align*}
\min & \quad f(x) = x^TPx + p^Tx, \\
\text{s.t.} & \quad x^TQ_ix + q_i^Tx - c_i \leq 0, \quad i = 1, \ldots, m.
\end{align*}
\]  
(QCQP)
where \( P \in \mathbb{R}^{n \times n} \) is a symmetric matrix, \( p \in \mathbb{R}^n \), \( Q_i \in \mathbb{R}^{n \times n} \) is a positive semidefinite matrix, \( q_i \in \mathbb{R}^n \) and \( c_i \in \mathbb{R} \) for \( i = 1, \ldots, m \). In this paper, we assume that the feasible region of (QCQP) is bounded with nonempty relative interior points. (QCQP) involves many classical models, such as box constrained quadratic programming problems [13], standard quadratic programming problems [5], detection of a copositive matrix [23] and trust region problems [6, 11]. (QCQP) has been proved to be NP-hard [18], thus we cannot solve it in polynomial time unless \( P=NP \) except for some subclasses of the problem [20]. As a quadratically constrained quadratic programming problem, the classical semidefinite programming (SDP) relaxation is an effective technique to provide a lower bound for the problem. Zheng et al. [22] proposed an SDP relaxation based on difference of convex (DC) decomposition scheme, and Lu et al. [16] proposed an SDP relaxation based on the eigenvalue decomposition. Chen et al. [10] introduced a polyhedral-SDP relaxation of completely positive (or copositive) programs. Deng et al. [12] presented a conic reformulation and approximation method. However, the dimension of positive semidefinite matrix is \( n^2 \), consequently it will be too expensive when employing SDP relaxation based branch-and-bound algorithm for finding a global optimal solution of high-dimensional problems. Some researchers then proposed other relaxations. For example, Kim et al. [15] proposed second-order cone programming (SOCP) relaxation methods for a nonconvex quadratic optimization problem and proved that the SOCP relaxation is a tool whose tightness is between SDP relaxation and the lift-and-project linear programming (LP) relaxation. Burer et al. [8] introduced a mixed SOCP and SDP relaxation, which balanced the solution time and bound quality.

Taking advantage of DC decomposition schemes [22] and Burer et al.’s research of mixed SOCP-SDP relaxation [8], we propose a new low-dimensional SDP relaxation in which the dimension of positive semidefinite matrix is \( r^2 \), where \( r \) is the number of negative eigenvalues of \( P \). The results on simultaneous diagonalization of two positive semidefinite matrices [21] and matrix block diagonalization method [8] are critical tools to derive the proposed relaxation. The advantage of the proposed relaxation is that we only introduce one low-dimensional positive semidefinite matrix through DC decomposition scheme rather than dividing a large positive semidefinite matrix into several small blocks which was first proposed by Burer et al. [8]. And we prove that the proposed relaxation is equivalent to a special SOCP-SDP relaxation, but the latter one has one more positive semidefinite matrix of size \( (n-r)^2 \) than ours. Therefore, the proposed relaxation is easier to solve than Burer et al.’s. We further prove that when there exists a \( \eta \in \mathbb{R} \) such that \( P + \eta Q_1 \succeq 0 \) or \( P + \eta Q_1 \preceq 0 \) and \( Q_i \) is null matrix for \( i = 2, \ldots, m \), the proposed relaxation is as tight as the classical SDP relaxation. For general cases, we propose an effective spatial branch-and-bound algorithm via exploiting the property that \( X = xx^T \) if and only if \( X \succeq xx^T \) and \( \text{diag}(X) = x \) (ref. [7]). In the proposed algorithm, we only need to add \( r \) reformulation-linearization technique (RLT) constraints to the SDP relaxation rather than \( n \) as in [7]. Lu et al. [16] also added \( r \) RLT constraints to the SDP relaxation, but the coefficients of RLT constraints in the proposed relaxation is much sparser than Lu’s. We compare the proposed algorithm with two cutting-edge SDP based branch-and-bound algorithms [7, 16] using similar branching strategies. Numerical results show that the proposed algorithm is more effective when \( n \) is large, \( m \) is small and \( r \leq \lceil \frac{2n}{3} \rceil \).
The rest of this paper is organized as follows: Section 2 presents a low-dimensional SDP relaxation for \((\text{QCQP})\) and is compared with SOCP-SDP relaxation and classical SDP relaxation, respectively. Section 3 states a spatial branch-and-bound algorithm for finding a global optimal solution of \((\text{QCQP})\). Extensive numerical experiments are carried out in Section 4 to show that the proposed algorithm outperforms two cutting-edge ones in the literature in some cases. The conclusions are given in Section 5.

The following notations are adopted throughout the paper. \(\mathbb{R}^n\) denotes the \(n\)-dimensional real vector and \(\mathbb{R}^{n \times n}\) denotes the \(n \times n\) real matrix. Given a real symmetric matrix \(X\), \(X \succeq 0\) means \(X\) is positive semidefinite and \(X \preceq 0\) means \(X\) is negative semidefinite. For \(n \times n\) real matrices \(A = (A_{ij})\) and \(B = (B_{ij})\), \(A \bullet B = \text{trace}(A^T B) = \sum_{i,j=1}^{n} A_{ij} B_{ij}\) and \(A_i\) denotes the \(i\)-th row of \(A\). Given a vector \(a \in \mathbb{R}^n\), \(\text{diag}(a)\) corresponds to an \(n \times n\) diagonal matrix with its diagonal equal to \(a\). \(\lfloor b \rfloor\) and \(\lceil b \rceil\) represent that \(b \in \mathbb{R}\) is rounded down and up to the nearest integer, respectively.

2. A new low-dimensional SDP relaxation and comparisons. The classical SDP relaxation of \((\text{QCQP})\) is (ref. [17]):

\[
\begin{align*}
\min & \quad P \bullet X + p^T x, \\
\text{s.t.} & \quad Q_i \bullet X + q_i^T x - c_i \leq 0, \quad i = 1, \ldots, m, \\
& \quad X \succeq xx^T.
\end{align*}
\]

When \(n\) is large, it is computationally expensive to solve the above SDP relaxation problem. In order to achieve a good trade-off between the bound and computational time, we aim to introduce a low-dimensional SDP relaxation for \((\text{QCQP})\). Let \(r\) be the number of negative eigenvalues of \(P\) and without loss of generality, we suppose that the first \(r\) eigenvalues of \(P\) are negative. Thus \(P = P_1 - P_2\) with \(P_1 = \sum_{j=r+1}^{n} \sigma_j g_j g_j^T\) and \(P_2 = \sum_{j=1}^{r} \sigma_j g_j g_j^T\) where \(\sigma_j \geq 0, -\sigma_1, \ldots, -\sigma_r, \sigma_{r+1}, \ldots, \sigma_n\) are eigenvalues and \(g_j, j = 1, \ldots, n\), are corresponding eigenvectors of \(P\). Then the nonconvex objective function can be rewritten as \(f(x) = x^T P_1 x + p^T x = x^T P_1 x - x^T P_2 x + p^T x\). Obviously, the difficulty of solving \((\text{QCQP})\) is dealing with the nonconvex term \(-x^T P_2 x\). This motivates us to construct the low-dimensional SDP relaxation by focusing on convexifying the nonconvex function \(-x^T P_2 x\). The tool we used to achieve this purpose is simultaneous diagonalization.

Ben-Tal et al. [2] presented that, for the problem of minimizing a quadratic objective function subject to one or two quadratic constraints, it is possible to derive an equivalent SOCP relaxation when the quadratic terms are simultaneously diagonalizable. Zhou et al. [25] proposed a simultaneous diagonalization based SOCP relaxation for convex quadratic program with linear complementarity constraints and proved that it is equivalent to the SDP relaxation when the objective matrix is positive definite. They all showed that simultaneous diagonalization is an effective technique when designing the convex relaxation. Furthermore, Burer et al. [8] presented a matrix block diagonalization technique in order to design an effective mixed SOCP-SDP relaxation for the quadratically constrained quadratic program. Thus, we will employ the two techniques together to design the convex relaxation in this paper.
Lemma 2.1. ([21]) If $P$ and $Q$ are both positive semidefinite matrices, then there exists a nonsingular matrix $F$ such that $F^TF$ and $F^TQF$ are both diagonal matrices.

Because $Q_1 \succeq 0$ and $P_2 \succeq 0$ with $r = \text{rank}(P_2)$, the proof of Lemma 2.1 in [21] gives a specific way to find a nonsingular matrix $F \in \mathbb{R}^{n \times n}$ such that $F^TQ_iF = \Sigma = \text{diag}(\mu_1, \ldots, \mu_n)$ and $F^TP_2F = \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix}$ with \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_r) \), $\lambda_i > 0$, $i = 1, \ldots, r$ and $\mu_j \geq 0$, $j = 1, \ldots, n$.

Besides, as $Q_i \succeq 0$, $i = 2, \ldots, m$, if $k_i = \text{rank}(Q_i) \geq 1$, then following from Algorithm 1 in [8], we can decompose $F^TQ_iF = B_i + \begin{bmatrix} D_i & 0 \\ 0 & 0 \end{bmatrix}$ with $D_i \in \mathbb{R}^{r \times r}$ as follows:

1. Find $L^i \in \mathbb{R}^{n \times k_i}$ such that $F^TQ_iF = (L^i)(L^i)^T$.
2. Define $S^i = \text{span}\{(L^i_j)^T : j = r+1, \ldots, n\}$ and $k_i = \text{dim}(S^i)$. Construct $V^i \in \mathbb{R}^{k_i \times k_i}$ with columns forming an orthonormal basis of $S^i$.
3. Define $B_i = L^iV^i(L^iV^i)^T$ and $D_i$ is the submatrix of the first $r$ rows and $r$ columns of the matrix $M^i = F^TQ_iF - B_i$.

Therefore, $B_i \succeq 0$, $M^i = L^i(\text{Id}_{k_i} - V^iV^i)^T(L^i)^T \succeq 0$ and $M_{jk} = 0$ if $j > r$ or $k > r$ as showed by Proposition 6 in [8]. Thus $F^TQ_iF = B_i + \begin{bmatrix} D_i & 0 \\ 0 & 0 \end{bmatrix}$ with $D_i \in \mathbb{R}^{r \times r}$ and $B_i \succeq 0$, $D_i \geq 0$.

By setting $x = F \begin{bmatrix} w \\ v \end{bmatrix}$ with $w \in \mathbb{R}^r$ and $v \in \mathbb{R}^{n-r}$, (QCQP) can be reformulated as

\[
\begin{aligned}
\min & \quad \begin{bmatrix} w \\ v \end{bmatrix}^TF^TP_1F \begin{bmatrix} w \\ v \end{bmatrix} - \sum_{j=1}^{r} \lambda_j w_j^2 + p^T F \begin{bmatrix} w \\ v \end{bmatrix}, \\
\text{s.t.} & \quad \sum_{j=1}^{r} \mu_j w_j^2 + \sum_{j=1}^{n-r} \mu_{r+j} v_j^2 + q_1^T F \begin{bmatrix} w \\ v \end{bmatrix} - c_1 \leq 0, \\
& \quad w^T D_i w + \begin{bmatrix} w \\ v \end{bmatrix}^T B_i \begin{bmatrix} w \\ v \end{bmatrix} + q_i^T F \begin{bmatrix} w \\ v \end{bmatrix} - c_i \leq 0, \quad i = 2, \ldots, m.
\end{aligned}
\]

(Ref)

Note that (Ref) still cannot be solved in polynomial time due to the nonconvex term $- \sum_{j=1}^{r} \lambda_j w_j^2$ in the objective function. But it can be relaxed into the following form:

\[
\begin{aligned}
\min & \quad \begin{bmatrix} w \\ v \end{bmatrix}^TF^TP_1F \begin{bmatrix} w \\ v \end{bmatrix} - \sum_{j=1}^{r} \lambda_j W_{jj} + p^T F \begin{bmatrix} w \\ v \end{bmatrix}, \\
\text{s.t.} & \quad \sum_{j=1}^{r} \mu_j W_{jj} + \sum_{j=1}^{n-r} \mu_{r+j} v_j^2 + q_1^T F \begin{bmatrix} w \\ v \end{bmatrix} - c_1 \leq 0, \\
& \quad D_i \bullet W + \begin{bmatrix} w \\ v \end{bmatrix}^T B_i \begin{bmatrix} w \\ v \end{bmatrix} + q_i^T F \begin{bmatrix} w \\ v \end{bmatrix} - c_i \leq 0, \quad i = 2, \ldots, m, \\
& \quad W - ww^T \succeq 0.
\end{aligned}
\]

(LowSDP)
Note that, in the relaxation (LowSDP), the size of $W$ is $r \times r$. Thus it is indeed a low-dimensional SDP relaxation if $r$ is small. Next, we will clarify the superiority of exploiting the techniques of simultaneous diagonalization and matrix block diagonalization to construct $F$, $B$ and $D$ for the proposed relaxation.

**Remark 1.** We choose $F$ such that $F^T P F$ and $F^T Q_i F$ are both diagonal, which has its superiority when $Q_i$ is null matrix for $i = 2, \ldots, m$. In this case, the proposed low-dimensional SDP relaxation degenerates into a SOCP relaxation as the low-dimensional SDP constraint $W \succ ww^T$ degenerates into $W_{jj} \geq w_j^2$ for $j = 1, \ldots, r$. However, it performs better than the classical SOCP relaxation proposed by Kim et al. [15] due to the fact that the auxiliary variables in the classical SOCP relaxation could be unbounded if no RLT constraints are added, while the auxiliary variables $W_{jj}$ in (LowSDP) are usually bounded by the constraint $\sum_{j=1}^{r} \mu_j W_{jj} + \sum_{j=1}^{r} \mu_{r+j} v_j^2 + q_i^T F \begin{bmatrix} w \\ v \end{bmatrix} - c_i \leq 0$ if $\mu_j > 0$. For example, for the trust-region problem with linear inequality constraints which will be introduced in Section 4, (LowSDP) is as tight as (SDP) (a special case of Theorem 2.4). The numerical results also show that the simultaneous diagonalization technique indeed improves the relaxation tightness.

Burer et al. [8] proved that $B_i$'s generated by employing the above method have the so-called “minimum positive semidefinite” property. That is $B_i \preceq B_i \in G_i$ for $i = 2, \ldots, m$, where $G_i = \{ B_i | B_i \succeq 0, F^T Q_i F - B_i = \begin{bmatrix} D_i & 0 \\ 0 & 0 \end{bmatrix} \text{ with } D_i \in \mathbb{R}^{r \times r} \}$. Define $h(B_2, \ldots, B_m)$ as the optimal value of (LowSDP) with $B_i$ replaced by $\bar{B}_i \in G_i$ for $i = 2, \ldots, m$. Next proposition shows that the $B_i$'s obtained by Burer et al.'s method result in the “best” (LowSDP) in terms of lower bound.

**Proposition 2.2.** $h(B_2, \ldots, B_m) \geq h(\bar{B}_2, \ldots, \bar{B}_m)$ for all $\bar{B}_i \in G_i$, $i = 2, \ldots, m$.

**Proof.** Since all the $B_i$'s have the “minimum positive semidefinite” property, $B_i \preceq B_i$ for all $\bar{B}_i \in G_i$. For any feasible solution $(w, v, W)$ of (LowSDP), we have $B_i - \bar{B}_i = F^T Q_i F - \bar{B}_i - (F^T Q_i F - B_i) = \begin{bmatrix} \bar{D}_i - D_i & 0 \\ 0 & 0 \end{bmatrix}$ and $(\bar{D}_i - D_i) \bullet W + \begin{bmatrix} w \\ v \end{bmatrix}^T (\bar{B}_i - B_i) \begin{bmatrix} w \\ v \end{bmatrix} = (\bar{B}_i - B_i) \bullet \left( \begin{bmatrix} W \\ v w^T \\ v v^T \end{bmatrix} - \begin{bmatrix} w \\ v \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix}^T \right) \leq 0$. Then, $\bar{D}_i \bullet W + \begin{bmatrix} w \\ v \end{bmatrix}^T \bar{B}_i \begin{bmatrix} w \\ v \end{bmatrix} + q_i^T F \begin{bmatrix} w \\ v \end{bmatrix} - c_i = D_i \bullet W + \begin{bmatrix} w \\ v \end{bmatrix}^T B_i \begin{bmatrix} w \\ v \end{bmatrix} + q_i^T F \begin{bmatrix} w \\ v \end{bmatrix} - c_i + (\bar{D}_i - D_i) \bullet W + \begin{bmatrix} w \\ v \end{bmatrix}^T (\bar{B}_i - B_i) \begin{bmatrix} w \\ v \end{bmatrix} \leq 0$. It shows that $(w, v, W)$ is also a feasible solution of (LowSDP) when $B_i$ is replaced by $\bar{B}_i \in G_i$ for $i = 2, \ldots, m$. Thus $h(B_2, \ldots, B_m) \geq h(\bar{B}_2, \ldots, \bar{B}_m)$. \hfill $\square$

To the best of our knowledge, only the SOCP-SDP relaxation proposed by Burer et al. [8] is similar to the proposed relaxation. We will compare the difference between these two relaxations in the next subsection and show that they are equivalent under certain conditions.

2.1. **Connections to SOCP-SDP relaxation.** In [8], Burer et al. constructed a mixed SOCP-SDP relaxation for (QCQP). Specifically, they decomposed $P = G_0 + (P - G_0)$ and $Q_i = G_i + (Q_i - G_i)$, $i = 1, \ldots, m$, such that $G_i$'s are block-diagonal. In this subsection, we will show that, when $G_i$'s are 2-block diagonal,
Burer et al.’s mixed SOCP-SDP relaxation is actually equivalent to the proposed relaxation (LowSDP).

We first derive Burer et al.’s mixed SOCP-SDP relaxation. In the above, we have showed that
\[ F^T P F = F^T P_1 F - F^T P_2 F = F^T P_1 F - \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix}, \]
\[ F^T Q_1 F = G_1 = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \text{ with } \Sigma_1 = \text{diag}(\mu_1, \ldots, \mu_r) \in \mathbb{R}^{r \times r}, \Sigma_2 = \text{diag}(\mu_{r+1}, \ldots, \mu_n) \in \mathbb{R}^{(n-r) \times (n-r)}, \]
and \( F^T Q_i F = B_i + \begin{bmatrix} D_i & 0 \\ 0 & 0 \end{bmatrix} \) with \( D_i \succeq 0 \) and \( B_i \succeq 0 \), \( i = 2, \ldots, m \).

Because \( F^T P_1 F \geq 0 \), it follows from Algorithm 1 in [8] that there is a matrix \( A \in \mathbb{R}^{(n-r) \times (n-r)} \) such that \( A \succeq 0 \) and \( F^T P_1 F - \begin{bmatrix} 0 & 0 \\ 0 & A \end{bmatrix} \geq 0 \). Hence, \( F^T P F = G_0 + \begin{bmatrix} 0 & 0 \\ 0 & A \end{bmatrix} \) with \( G_0 = \begin{bmatrix} -\Lambda & 0 \\ 0 & A \end{bmatrix} \). Again, because \( B_i \succeq 0 \), we can find a matrix \( H_i \in \mathbb{R}^{(n-r) \times (n-r)} \) such that \( H_i \succeq 0 \) and \( B_i - \begin{bmatrix} 0 & 0 \\ 0 & H_i \end{bmatrix} \succeq 0 \) for \( i = 2, \ldots, m \). Then we have \( F^T Q_i F = G_i + \begin{bmatrix} B_i - \begin{bmatrix} 0 & 0 \\ 0 & H_i \end{bmatrix} \end{bmatrix} \) with \( G_i = \begin{bmatrix} D_i & 0 \\ 0 & H_i \end{bmatrix} \).

Obviously, \( G_i \) is a 2-block diagonal matrix for \( i = 0, 1, \ldots, n \). Consequently, we have the following mixed SOCP-SDP relaxation:

\[
\begin{align*}
\min & \quad A \bullet V - \Lambda \bullet W + \begin{bmatrix} w \\ v \end{bmatrix}^T \left( F^T P_1 F - \begin{bmatrix} 0 & 0 \\ 0 & A \end{bmatrix} \right) \begin{bmatrix} w \\ v \end{bmatrix} + p^T F \begin{bmatrix} w \\ v \end{bmatrix}, \\
\text{s.t.} & \quad \Sigma_1 \bullet W + \Sigma_2 \bullet V + q_i^T F \begin{bmatrix} w \\ v \end{bmatrix} - c_i \leq 0, \\
& \quad D_i \bullet W + H_i \bullet V + \begin{bmatrix} w \\ v \end{bmatrix}^T \left( B_i - \begin{bmatrix} 0 & 0 \\ 0 & H_i \end{bmatrix} \right) \begin{bmatrix} w \\ v \end{bmatrix} + q_i^T F \begin{bmatrix} w \\ v \end{bmatrix} - c_i \leq 0, \quad i = 2, \ldots, m, \\
& \quad W \succeq w w^T, \\
& \quad V \succeq v v^T.
\end{align*}
\]

(\text{SOCP-SDP})

By comparing the formulations of (SOCP-SDP) and (LowSDP), the semidefinite constraint \( V \succeq v v^T \) is relieved in the proposed relaxation (LowSDP). However, the next theorem shows that the two relaxations actually have the same optimal value.

**Theorem 2.3.** (SOCP-SDP) and (LowSDP) share the same optimal value.

**Proof.** On one hand, if \((w, v, W, V)\) is a feasible solution of (SOCP-SDP), then obviously \((w, v, W; v v^T)\) is also a feasible solution of (SOCP-SDP) due to \( H_i \succeq 0 \) and \( \Sigma_2 \succeq 0 \). And it is straightforward to see that \((w, v, W)\) is a feasible solution of (LowSDP). Since \( A \succeq 0 \) and \( V \succeq v v^T \), we have \( A \bullet V - \Lambda \bullet W + \begin{bmatrix} w \\ v \end{bmatrix}^T \left( F^T P_1 F - \begin{bmatrix} 0 & 0 \\ 0 & A \end{bmatrix} \right) \begin{bmatrix} w \\ v \end{bmatrix} + p^T F \begin{bmatrix} w \\ v \end{bmatrix} \geq A \bullet v v^T - \Lambda \bullet W + \begin{bmatrix} w \\ v \end{bmatrix}^T \left( F^T P_1 F - \begin{bmatrix} 0 & 0 \\ 0 & A \end{bmatrix} \right) \begin{bmatrix} w \\ v \end{bmatrix} + p^T F \begin{bmatrix} w \\ v \end{bmatrix} = -\Lambda \bullet W + \begin{bmatrix} w \\ v \end{bmatrix}^T F^T P_1 F \begin{bmatrix} w \\ v \end{bmatrix} + p^T F \begin{bmatrix} w \\ v \end{bmatrix} \). Therefore, the optimal value of (LowSDP) is no more than that of (SOCP-SDP).

On the other hand, if \((w, v, W)\) is a feasible solution of (LowSDP), then it is easy to prove that \((w, v, W; v v^T)\) is also feasible to (SOCP-SDP), thus the optimal value of (SOCP-SDP) is no more than that of (LowSDP).

\[ \square \]
Theorem 2.3 points out that (LowSDP) provides a lower bound for (QCQP) as tight as the mixed SOCP-SDP relaxation (SOCP-SDP). But (LowSDP) can be solved within less computational time than (SOCP-SDP) since there is one more semidefinite constraint $V \succeq vv^T$ in (SOCP-SDP). In order to demonstrate this, we generate some random instances and solve the two relaxations of these instances. The results are summarized in Table 1. For various combinations of $n$, $m$ and $r$, the CPU time for solving the proposed relaxation (LowSDP) is much smaller than (SOCP-SDP).

A prominent feature of Burer et al.’s framework is that it allows one to decompose $P$ and $Q_i$’s into different block-diagonal structures for the balance of speed and bound quality. In what follows, we will use an example to illustrate that the proposed relaxation (LowSDP) can be tighter than the equational 3-block SOCP-SDP relaxation. Therefore, the proposed relaxation is not a special case under Burer et al.’s framework.

Example 1. Let $n = 6$, $m = 2$, $r = 3$, $q_1$ is a null vector, $Q_1 = I_n$ where $I_n$ is the $n$ dimensional identity matrix, $c_1 = c_2 = 1$, $P$ is a diagonal matrix with the diagonal elements $(-1.0773, -0.3140, -2.3756, 1.0699, 1.2419, 0.6097)$,

$$p = (0.2766, 0.2903, 0.9581, -0.5336, 1.2941, -0.9449)^T,$$

$$\begin{bmatrix}
0.8199 & 0.1804 & 0.1432 & 0.1165 & -0.1187 & -0.4699 \\
0.1804 & 2.8031 & 0.4943 & -0.2983 & -0.9563 & -0.6905 \\
0.1432 & 0.4943 & 0.7943 & -0.0090 & -0.1920 & -0.2661 \\
0.1165 & -0.2983 & -0.0090 & 1.5849 & 0.3673 & -0.5846 \\
-0.1187 & -0.9563 & -0.1920 & 0.3673 & 1.0677 & -0.1157 \\
-0.4699 & -0.6905 & -0.2661 & -0.5846 & -0.1157 & 2.0164 \\
\end{bmatrix},$$

$$Q_2 = (-0.9937, 0.6358, -0.2616, -0.6223, -1.2748, 0.6276)^T.$$

$Q_2$ is decomposed into the following two forms in order to derive (LowSDP) and the equational 3-blocks SOCP-SDP relaxation of original problem, respectively.

$$\begin{bmatrix}
0.6224 & -0.3060 & 0 & 0 & 0 & 0 \\
-0.3060 & 0.7167 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.6679 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.6679 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.6679 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.6666
\end{bmatrix} + \begin{bmatrix}
0.1975 & 0.4864 & 0.1432 & 0.1165 & -0.1187 & -0.4699 \\
0.4864 & 2.0864 & 0.4943 & -0.2983 & -0.9563 & -0.6905 \\
0.1432 & 0.4943 & 1.264 & -0.0090 & -0.1920 & -0.2661 \\
0.1165 & -0.2983 & -0.0090 & 1.5849 & 0.3673 & -0.5846 \\
-0.1187 & -0.9563 & -0.1920 & 0.3673 & 1.0677 & -0.1157 \\
-0.4699 & -0.6905 & -0.2661 & -0.5846 & -0.1157 & 2.0164 \\
\end{bmatrix},$$

and

$$\begin{bmatrix}
0.6224 & -0.3060 & 0 & 0 & 0 & 0 \\
-0.3060 & 0.7167 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.6679 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.6679 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.6679 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.6666
\end{bmatrix} + \begin{bmatrix}
0.1975 & 0.4864 & 0.1432 & 0.1165 & -0.1187 & -0.4699 \\
0.4864 & 2.0864 & 0.4943 & -0.2983 & -0.9563 & -0.6905 \\
0.1432 & 0.4943 & 1.264 & -0.0090 & -0.1920 & -0.2661 \\
0.1165 & -0.2983 & -0.0090 & 1.5849 & 0.3673 & -0.5846 \\
-0.1187 & -0.9563 & -0.1920 & 0.3673 & 1.0677 & -0.1157 \\
-0.4699 & -0.6905 & -0.2661 & -0.5846 & -0.1157 & 2.0164 \\
\end{bmatrix}.$$

The lower bounds of (LowSDP) and the equational 3-blocks SOCP-SDP relaxation are -2.6681 and -3.4251, respectively.

2.2. Comparison with classical SDP relaxation. In general, the proposed relaxation (LowSDP) provides a weaker lower bound than (SDP). However, we will show that (LowSDP) is equivalent to the classical SDP relaxation (SDP) in two cases.

Theorem 2.4. (LowSDP) and (SDP) share the same optimal value when
Proof. We would first prove (i) and then show that (ii) is a special case of (i).

(i) Since there exists a nonsingular matrix $F$ such that $F^T PF$ and $F^T Q_i F$ are both diagonal matrices, $F^T Q_i F = \begin{bmatrix} D_i & 0 \\ 0 & C_i \end{bmatrix}$ with $D_i \in \mathbb{R}^{r \times r}$ and $C_i \in \mathbb{R}^{(n-r) \times (n-r)}$ for $i = 2, \ldots, m$, or

(ii) There exists an $\eta \in \mathbb{R}$ such that $P + \eta Q_1 \succeq 0$ or $P + \eta Q_1 \preceq 0$ and $Q_i$ is null matrix for $i = 2, \ldots, m$.

Proof. We would first prove (i) and then show that (ii) is a special case of (i).

(i) Since there exists a nonsingular matrix $F$ such that $F^T PF$ and $F^T Q_i F$ are both diagonal matrices and $F^T Q_i F = \begin{bmatrix} D_i & 0 \\ 0 & C_i \end{bmatrix}$ with $D_i \in \mathbb{R}^{r \times r}$ and $C_i \in \mathbb{R}^{(n-r) \times (n-r)}$ for $i = 2, \ldots, m$, $F^T PF$ can be written as $F^T PF = \text{diag}(\xi_1, \ldots, \xi_n)$ with $\xi_i < 0$ for $i = 1, \ldots, r$, $\xi_i \geq 0$ for $i = r + 1, \ldots, n$ and $F^T Q_i F$ can be written as $F^T Q_i F = \begin{bmatrix} D_1 & 0 \\ 0 & C_1 \end{bmatrix}$ with $D_1 \in \mathbb{R}^{r \times r}$ and $C_1 \in \mathbb{R}^{(n-r) \times (n-r)}$ for format consistency with $F^T Q_i F$, $i = 2, \ldots, m$. If $(w, v, W)$ is a feasible solution of (LowSDP), then we define $x = F \begin{bmatrix} w \\ v \end{bmatrix}$ and $X = F \begin{bmatrix} W & wv^T \\ vw^T & vv^T \end{bmatrix} F^T$. It is easy to prove that $X \succeq xx^T$ and

\begin{align*}
(1) \quad P \bullet X + p^T x &= F^T PF \bullet \begin{bmatrix} W & wv^T \\ vw^T & vv^T \end{bmatrix} + p^T F \begin{bmatrix} w \\ v \end{bmatrix} \\
&= \sum_{j=1}^{r} \xi_j W_{jj} + \sum_{j=1}^{n-r} \xi_{r+j} v_j^2 + p^T F \begin{bmatrix} w \\ v \end{bmatrix},
\end{align*}

\begin{align*}
(2) \quad Q_i \bullet X + q_i^T x - c_i &= F^T Q_i F \bullet \begin{bmatrix} W & wv^T \\ vw^T & vv^T \end{bmatrix} + q_i^T F \begin{bmatrix} w \\ v \end{bmatrix} - c_i \\
&= D_i \bullet W + C_i \bullet vv^T + q_i^T F \begin{bmatrix} w \\ v \end{bmatrix} - c_i \leq 0, \quad i = 1, \ldots, m.
\end{align*}

It follows that $(x, X)$ is a feasible solution of (SDP). Thus the optimal value of (SDP) is no more than the one of (LowSDP).
Conversely, if \((x, X)\) is a feasible solution of \((SDP)\), then we define 
\[
\begin{bmatrix}
w \\
v
\end{bmatrix} = F^{-1} x,
\]
\[
F^{-1} X (F^T)^{-1} = \begin{bmatrix} G_1 & G_2 \\ G_3 & G_4 \end{bmatrix} \quad \text{and} \quad W = G_1.
\]
It is straightforward to show that \((w, v, W)\) is a feasible solution of \((LowSDP)\). Hence, \((LowSDP)\) and \((SDP)\) share the same optimal value.

(ii) Lemma 2.1 implies \(P + \eta Q_1\) and \(Q_1\) can be simultaneously diagonalized under the condition that there exists an \(\eta\) such that \(P + \eta Q_1 \succeq 0\) or \(P + \eta Q_1 \preceq 0\). Consequently, there exists a nonsingular matrix \(F\) such that \(F^T P F\) and \(F^T Q_1 F\) are both diagonal matrices. This shows that (ii) is a special case of (i). □

Actually, the case (i) in Theorem 2.4 indicates that \((LowSDP)\) and \((SDP)\) are equivalent when \(P\) and \(Q_i\), \(i = 1, \ldots, m\), are simultaneously diagonalizable.

**Proposition 2.5.** When \(m = 1\) and there exists an \(\eta \in \mathbb{R}\) such that \(P + \eta Q_1 \succeq 0\) or \(P + \eta Q_1 \preceq 0\), \((LowSDP)\) and \((QCQP)\) have the same optimal value.

**Proof.** The optimal value of \((SDP)\) equals to the one of \((QCQP)\) when \(m = 1\) (ref. [20]), thus Theorem 2.4 implies that the optimal value of \((LowSDP)\) is the same as the one of \((QCQP)\). □

3. A low-dimensional SDP relaxation based spatial branch-and-bound algorithm. In this section, we develop a spatial branch-and-bound algorithm for \((QCQP)\) based on the proposed relaxation \((LowSDP)\). We first give the optimality condition of the proposed relaxation.

**Lemma 3.1.** Let \((w, v, W)\) be an optimal solution of \((LowSDP)\), then 
\[
x = F \begin{bmatrix} w \\ v \end{bmatrix}
\]
is a feasible solution of \((QCQP)\). If \(W_{jj} = w_j^2\) for \(j = 1, \ldots, r\), then 
\[
x = F \begin{bmatrix} w \\ v \end{bmatrix}
\]
is an optimal solution of \((QCQP)\).

**Proof.** If \(W_{jj} = w_j^2\) for \(j = 1, \ldots, r\) and \(W - w w^T \succeq 0\), then \(W = w w^T\). That implies \(W\) is a rank-one matrix and the relaxation \((LowSDP)\) is tight under this condition. Therefore, 
\[
x = F \begin{bmatrix} w \\ v \end{bmatrix}
\]
is an optimal solution of \((QCQP)\). □

Lemma 3.1 shows that if \((LowSDP)\) is not tight, then there must exist some \(j \in \{1, \ldots, r\}\) such that \(W_{jj} > w_j^2\). Then we can design a branching strategy by selecting the index with maximal \(\lambda_j (W_{jj} - w_j^2)\).

Anstreicher [1] pointed out that incorporating the RLT inequalities into SDP relaxations could provide tighter bounds for \((QCQP)\) than applying RLT or SDP alone. Hence, if \(l_j \leq w_j \leq u_j\), the so-called RLT constraints \(W_{jj} \leq (l_j + u_j) w_j - l_j u_j\), \(j = 1, \ldots, r\), can be added into \((LowSDP)\) to improve the lower bound.

**Remark 2.** We only add \(r\) RLT constraints to the relaxation \((LowSDP)\) rather than \(n\) as in [7]. This relieves the computational burden for solving \((LowSDP)\) with RLT constraints, especially when \(r\) is small. The proposed relaxation not only lowers the dimension of SDP constraint but also reduces the number of RLT constraints.

**Remark 3.** In [16], Lu et al. also add \(r\) RLT constraints to tight the relaxation. Their RLT constraints have the form of \(g_j^T X g_j \leq (l_j + u_j) g_j^T x - l_j u_j\), where \(g_j\) is
Lemma 3.2. The number of variables involved in each of Lu’s RLT constraints is between 2 and \(\frac{n(n+3)}{2}\), depending on the number of nonzero elements of \(g_i\). In contrast, there are only two variables \(W_{ij}\) and \(w_j\) involved in RLT constraints of the proposed relaxation. Therefore, the coefficients of the RLT constraints are sparser and easier to tackle than Lu’s.

The spatial branch-and-bound algorithm is presented in Algorithm 1. The framework of the proposed algorithm includes the following four steps:

(1) **Initialization.** The initial lower bound \(l_j^0\) and upper bound \(u_j^0\) for \(w_j\), \(j = 1, \ldots, r\), are obtained by solving the following 2r convex quadratic programming problems:

\[
v_j^0 / u_j^0 = \min / \max w_j,
\]

s.t.
\[
\begin{bmatrix}
w \\
v
\end{bmatrix}^T F^T Q_i F \begin{bmatrix}
w \\
v
\end{bmatrix} + q_i^T F \begin{bmatrix}
w \\
v
\end{bmatrix} - c_i \leq 0, i = 1, \ldots, m. \quad (w_j\text{-Bound})
\]

(2) **The node selection strategy.** We use the classical “best-first” selection strategy, i.e., we select the one with the lowest bound among the live subproblems.

(3) **The variable selection strategy and branching rule.** Let \((w^p, v^p, W^p)\) be the solution of (LowSDP) at the current node over the box \([l^p, u^p]\). Set \(j^*\) to be the index with maximal \(\lambda_j(W_{jj}^p - (w_j^p)^2)\). Then we split the box \([l^p, u^p]\) into two sub-boxes \([l^a, u^a]\) and \([l^b, u^b]\) with \(l^a = l^p\), \(u^a = u^p\), \(u_{j^*}^a = \frac{u_{j^*}^p + v_{j^*}^p}{2}\), \(l^b = l^p\), \(u^b = u^p\), \(l_{j^*}^b = \frac{v_{j^*}^p + u_{j^*}^p}{2}\). Then two new subproblems are generated over the two new sub-boxes \([l^a, u^a]\) and \([l^b, u^b]\), respectively.

(4) **Lower Bound.** As pointed out by the proposed branching rule, each enumeration node is over a box \([l, u]\). We compute the lower bound \(lb\) for each node via solving (LowSDP) with RLT constraints \(W_{ij} \leq (l_j + u_j)w_j - l_ju_j, j = 1, \ldots, r\).

We claim that, if problem (QCQP) is feasible and bounded, then the proposed algorithm terminates in finite iterations and for any \(\epsilon > 0\), it returns a \(U^*\) such that \(|U^* - V_{opt}| \leq \epsilon\), where \(V_{opt}\) denotes the optimal value of (QCQP).

**Lemma 3.2.** Assume the problem \(\{w^p, v^p, W^p, lb^p, l^p, u^p\}\) in node \(p\) is chosen from \(D\) in Line 11 of Algorithm 1. Let \(ub^p = f \left( \begin{bmatrix} w^p \\ v^p \end{bmatrix} \right)\). For any \(\epsilon > 0\), there exists a \(\delta > 0\) such that if \(\max_{j \in \{1, 2, \ldots, r\}} (u_j^p - l_j^p) \leq \delta\), then Algorithm 1 terminates in Line 13.

**Proof.** \(ub^p - lb^p = f \left( \begin{bmatrix} w^p \\ v^p \end{bmatrix} \right) - lb^p = \sum_{j=1}^{r} \lambda_j(W_{jj}^p - (w_j^p)^2) \leq \sum_{j=1}^{r} \frac{\lambda_j(u_j^p - l_j^p)^2}{2} \leq \frac{2\epsilon^2}{\max_{j \in \{1, 2, \ldots, r\}} \lambda_j}. \) For any \(\epsilon > 0\), we set \(\delta = \frac{2\epsilon \max_{j \in \{1, 2, \ldots, r\}} \lambda_j}{\sqrt{\max_{j \in \{1, 2, \ldots, r\}} \lambda_j}}\), then \(U^* - lb^p \leq ub^p - lb^p \leq \epsilon\). Consequently, Algorithm 1 terminates in Line 13.

**Theorem 3.3.** [24] Assume problem (QCQP) is feasible and bounded and \(V_{opt}\) is the optimal value. Algorithm 1 returns \((x^*, U^*)\) after taking at most

\[
N_c = \prod_{j=1}^{r} \left[ \frac{\sqrt{\max_{j \in \{1, 2, \ldots, r\}} \lambda_j}}{\epsilon} (u_j^0 - l_j^0) \right]
\]

iterations such that \(|U^* - V_{opt}| \leq \epsilon\) for any given \(\epsilon > 0\).

**Proof.** Under the condition that problem (QCQP) is feasible and bounded, the initial box \([l_0^0, u_0^0]\) is bounded and nonempty. In every iteration of Algorithm 1,
Algorithm 1 A Spatial Branch-and-Bound Algorithm for Solving (QCQP)

Require: An instance of (QCQP) and a given error tolerance $\epsilon > 0$. Set iteration step $p = 1$.
1: Solve (LowSDP) with $[l^0, u^0]$ for $F$ and $u^0$.
2: if (LowSDP) is infeasible, then
3: (QCQP) is infeasible and terminate.
4: end if
5: Solve (LowSDP) with $[l^0, u^0]$ for its optimal value $lb^0$ and optimal solution $(w^0, v^0, W^0)$. Let $U^* = f\left(F\left[w^0\right]\right)$ and $x^* = F\left[w^0\right]$. 
6: Construct a set $\mathcal{D}$ and insert $\{w^0, v^0, W^0, lb^0, l^0, u^0\}$ into it.
7: loop
8: if $\mathcal{D} = \emptyset$, then
9: return $(x^*, U^*)$ and terminate.
10: end if
11: Choose a problem from $\mathcal{D}$ using the node selection strategy, denoted as $\{w^p, v^p, W^p, lb^p, l^p, u^p\}$ such that $lb^p = \min\{lb^k | lb^k \in \mathcal{D}\}$. Delete it from $\mathcal{D}$.
12: if $U^* - lb^p \leq \varepsilon$, then
13: return $(x^*, U^*)$ and terminate.
14: end if
15: Set $p \leftarrow p + 1$.
16: Choose $j^*$ by the variable selection strategy.
17: Set $l^p = l^p, u^a = u^p, u^p = \frac{v^p + u^p}{2}, l^0 = l^p, u^b = u^p, l^p = \frac{v^p + u^p}{2}$. 
18: if (LowSDP) with $[l^0, u^0]$ is feasible, then
19: Solve (LowSDP) with $[l^0, u^0]$ for its optimal value $lb^a$ and optimal solution $(w^a, v^a, W^a)$ and denote $ub^a = f\left(F\left[w^a\right]\right)$. 
20: if $ub^a < U^*$, then
21: $U^* = ub^a$ and $x^* = F\left[w^a\right]$.
22: end if
23: if $U^* - lb^a > \varepsilon$, then
24: insert $\{w^a, v^a, W^a, lb^a, l^0, u^a\}$ into $\mathcal{D}$.
25: end if
26: end if
27: if (LowSDP) with $[l^b, u^b]$ is feasible, then
28: Solve (LowSDP) with $[l^b, u^b]$ for its optimal value $lb^b$ and optimal solution $(w^b, v^b, W^b)$ and denote $ub^b = f\left(F\left[w^b\right]\right)$. 
29: if $ub^b < U^*$, then
30: $U^* = ub^b$ and $x^* = F\left[w^b\right]$.
31: end if
32: if $U^* - lb^b > \varepsilon$, then
33: insert $\{w^b, v^b, W^b, lb^b, l^0, u^b\}$ into $\mathcal{D}$.
34: end if
35: end if
36: end loop

if it does not terminate at Line 13, then we split a chosen box in half along the direction perpendicular to the chosen edge to generate two new sub-boxes. After the $k$-th iteration, the initial box $[l^0, u^0]$ will be split into $k + 1$ sub-boxes. If
Algorithm 1 does not terminate at Line 13, we claim that, for each sub-box \([l^p, u^p]\) among all \(k + 1\) sub-boxes, \(u^p_j - l^p_j \geq \min\{\frac{\delta}{\sqrt{2}}, u^0_j - l^0_j\}\) with \(\delta = \frac{2\sqrt{\epsilon}}{\sqrt{\lambda_{\text{max}}(1, 2, \ldots, r)}}\) for \(j = 1, \ldots, r\). Actually, if there exists a \(j^*\) such that \(u^p_{j^*} - l^p_{j^*} \leq \frac{\delta}{\sqrt{2}}\), then \(\lambda_{j^*}(W^p_{j^*})^2 - (u^0_{j^*} - l^0_{j^*})^2 \leq \lambda_{j^*}(\frac{\delta}{\sqrt{2}})^2 \leq \frac{\epsilon}{\sqrt{2}}\). This implies that the \(j^*\)-th edge is never selected as a branching direction. Otherwise, if the node \(p\) is chosen by the node selection strategy at Line 11, Algorithm 1 does not terminate at Line 13 and \(j^*\) is chosen by the variable selection strategy at Line 16, we have \(U^* - lb^p \leq ub^p - lb^p \leq \epsilon\) following from Lemma 3.2. This contradicts with the assumption that Algorithm 1 does not terminate at Line 13 at node \(p\). Hence, the total volume of all the \(k + 1\) sub-boxes is no more than that of the initial box \([l^0, u^0]\). Hence, Algorithm 1 returns \((x^*, U^*)\) such that \(|U^* - V_{\text{opt}}| < \epsilon\) after at most \(N_\epsilon\) iterations.

4. Numerical Experiments. In this section, we compare the proposed algorithm with ED algorithm [16] and BW algorithm [7] on two classes of (QCQP) problems: trust-region problem with linear inequality constraints and the generalized Celis-Dennis-Tapia problem. In Lu et al.’s work [16], they have shown that their method outperformed the commercial solver Baron for the quadratic programming problems. As the results of our experiment will show, the proposed algorithm is comparable with Lu’s method. Besides, some preliminary tests show that the proposed algorithm outperformed Baron and Couenne. Therefore, we omit the comparison with these two commercial solvers. In order to drop the easy instances in the experiment, we only select the instances which cannot be solved within one iteration. The algorithms are implemented in MATLAB R2013b on a PC with Window 7 and 2.50 GHZ Inter Dual Core CPU processors. The initial bounds \(l^0\) and \(u^0\) are computed by Cplex 12.6 and the SDP relaxations are solved by Sedumi [19]. In all algorithms, the error tolerance is set to be \(\epsilon = 1\times 10^{-5}\). Ten instances are generated for each given problem size in Tables 2 and 3. The average number of iterations (aver it), average CPU time in seconds (aver CPU) and standard deviation of CPU time (std) are displayed for each algorithm in the Tables 2 and 3.

4.1. The trust-region problem with linear inequality constraints. Consider the trust-region problem with linear inequality constraints,

\[
\begin{align*}
\min & \quad x^T P x + p^T x, \\
\text{s.t.} & \quad \|x\| \leq 1, \\
& \quad Bx \leq \beta,
\end{align*}
\]

where \(P \in \mathbb{R}^{n \times n}\) is a symmetric matrix, \(p \in \mathbb{R}^n\), \(B \in \mathbb{R}^{m \times n}\) and \(\beta \in \mathbb{R}^m\). (TRL) is an equivalent reformulation of the problem in [14] which arises in many scientific and engineering fields such as regularization of certain ill-posed problems [3]. The instances are generated as follows (ref. [6, 22]). The entries of \(P\) and \(p\) are independently generated from the normal distribution, and \(P\) is replaced by its real symmetric parts; the parameters in the linear constraints are set as \(B = (B_{ij})\) with \(B_{ij}\) uniformly sampled between 0 and 50, and \(\beta = B_{e/n}\), where \(e\) is the all-one vector. Table 2 lists the comparison results with various \(n, m\) and \(r\).
Table 2. Comparisons on trust-region problems with linear inequality constraints

| (n, m, r) | Proposed Algorithm | RO Algorithm | BW Algorithm |
|----------|--------------------|--------------|--------------|
|          | aver iter | aver CPU | std | aver iter | aver CPU | std | aver iter | aver CPU | std |
| (100,2,33) | 10.3 | 21.7 | 3.3 | 8.6 | 0.6 | 14.5 | 4.6 | 229.4 | 8.4 | (150,6,75) | 10.5 | 16.1 | 1.5 | 10.5 | 81.3 | 13.7 | 44.2 | 86.5 | 175.0 |
| (200,2,66) | 9.6 | 37.5 | 2.2 | 9.0 | 213.6 | 42.7 | 36.5 | 133.2 | 190.6 |
| (100,6,66) | 10.2 | 6.8 | 0.6 | 10.1 | 24.5 | 3.4 | 182.1 | 200.2 | 270.6 |
| (150,6,50) | 8.8 | 17.2 | 1.7 | 8.4 | 70.9 | 26.3 | 67.4 | 158.5 | 184.0 |
| (200,6,66) | 9.3 | 36.5 | 2.7 | 9.3 | 237.1 | 52.8 | 76.2 | 426.2 | 1077.7 |
| (100,10,33) | 10.1 | 7.1 | 0.9 | 10.0 | 22.3 | 4.5 | 124.6 | 113.7 | 119.7 |
| (150,10,50) | 10.4 | 17.7 | 2.0 | 10.1 | 78.1 | 17.5 | 569.7 | 1583.9 | 4553.0 |
| (200,10,66) | 9.5 | 36.2 | 6.1 | 9.3 | 196.6 | 69.5 | 308.0 | 1336.0 | 2984.3 |

$\bar{x}$ is set to be $(QCQP)$. The instances for the experiment are generated as follows (ref. [6, 16]).

and consequently the generalized Celis-Dennis-Tapia problem is a special case of identification problems. An ellipsoidal constraint is also a convex quadratic constraint CDT problem is an extension of the trust-region subproblem. It involves minimizing a quadratic function over the intersection of several ellipsoids [4, 6]. Generalized CDT problem is a classical problem arising frequently in nonlinear parameter identification problems. An ellipsoidal constraint is also a convex quadratic constraint and consequently the generalized Celis-Dennis-Tapia problem is a special case of (QCQP). The instances for the experiment are generated as follows (ref. [6, 16]).

The first constraint is set to be $x^T x \leq 1$. The entries of $P$, $p$, $Q_i$ and $q_i = 2, \ldots, m$, are independently generated from the standard normal distribution. Then, $P$ and $Q_i$ are replaced by their real symmetric parts. $P$ is then decomposed as $P = VDV^T$ where $D$ is the diagonal matrix of eigenvalues and $V$ is the orthogonal matrix whose columns are the corresponding eigenvectors. Set $D_{kk} = -|D_{kk}|$ for $k = 1, \ldots, r$ and $\bar{D}_{kk} = |D_{kk}|$ for $k = r + 1, \ldots, n$, then let $P = VDV^T$. Similarly, we can decompose $Q_i$ as $Q_i = V^tD^t(V^t)^T$, then set $D^t = |D^t|$ and $Q_i = V^tD^t(V^t)^T$. At last, we generate a vector $\alpha \in \mathbb{R}^n$ with unit length from the standard normal distribution, and scale all the $Q_i$'s and $q_i$'s by $|\alpha|^2 Q_i, \alpha + q_i^T \alpha|$ and $\sqrt{|\alpha|^2 Q_i, \alpha + q_i^T \alpha}$, respectively. $c_i$ is set to be 1 for $i = 2, \ldots, m$. Table 3 lists the comparison results with various combination of $n$, $m$ and $r$.

It can be observed from Table 3 that when $m \leq 5$, although the average iterations of the proposed algorithm is close to ED algorithm, the CPU time is less for most of the instances. Also, the proposed algorithm clearly outperforms BW algorithm both in the number of iterations and CPU time. The results also show that the proposed algorithm is the dominating one when $n$ is medium or large, $m$ and $r$ are
| (n, m, r)     | Proposed algorithm | ED Algorithm | BW Algorithm |
|--------------|--------------------|--------------|--------------|
|              | aver iter | aver CPU | std | aver iter | aver CPU | std | aver iter | aver CPU | std |
| (100,2,33)   | 10.8      | 10.8     | 1.1  | 10.7      | 24.8     | 3.8  | 132.8      | 154.0     | 102.2 |
| (150,2,50)   | 9.5       | 29.0     | 1.6  | 9.5       | 81.9     | 6.7  | 55.9       | 163.1      | 197.1 |
| (200,2,66)   | 9.4       | 62.1     | 6.2  | 9.3       | 199.4    | 32.9 | 52.5       | 235.3      | 227.5 |
| (100,3,33)   | 10.9      | 17.3     | 2.0  | 9.6       | 24.3     | 6.9  | 86.1       | 114.0      | 187.5 |
| (150,3,50)   | 11.8      | 60.5     | 5.6  | 10.5      | 99.9     | 14.9 | 65.8       | 221.9      | 244.6 |
| (200,3,66)   | 12.1      | 125.0    | 42.8 | 11.6      | 257.0    | 131.5 | 125.3      | 675.5      | 1103.2 |
| (100,4,33)   | 12.1      | 28.2     | 7.2  | 11.7      | 32.8     | 9.5  | 270.1      | 355.3      | 431.9 |
| (150,4,50)   | 9.9       | 76.2     | 6.6  | 9.2       | 90.6     | 12.5 | 34.6       | 175.9      | 168.3 |
| (200,4,66)   | 11.1      | 233.4    | 36.8 | 10.3      | 284.6    | 69.4 | 345.1      | 2594.1     | 6676.9 |
| (100,5,33)   | 11.0      | 41.6     | 4.7  | 10.3      | 42.0     | 4.9  | 72.7       | 117.9      | 139.5 |
| (150,5,50)   | 10.9      | 138.3    | 21.5 | 10.5      | 129.8    | 10.5 | 74.8       | 371.2      | 507.1 |
| (200,5,66)   | 10.9      | 345.3    | 60.5 | 10.8      | 344.1    | 68.4 | 96.8       | 904.7      | 1064.0 |
| r = \{ 4 \}  |           |          |     |           |          |     |           |          |     |
| (100,2,50)   | 10.0      | 14.5     | 2.3  | 10.3      | 36.1     | 7.9  | 129.5      | 151.4      | 147.9 |
| (150,2,75)   | 10.0      | 41.0     | 3.8  | 10.0      | 130.9    | 18.4 | 52.0       | 153.9      | 218.3 |
| (200,2,100)  | 10.6      | 89.9     | 11.6 | 10.2      | 331.4    | 73.7 | 30.2       | 146.6      | 55.8  |
| (100,3,50)   | 9.8       | 22.7     | 1.2  | 9.8       | 40.4     | 2.9  | 22.8       | 35.1       | 34.9  |
| (150,3,75)   | 9.9       | 65.4     | 6.4  | 9.8       | 131.6    | 22.7 | 33.2       | 112.5      | 142.5 |
| (200,3,100)  | 9.9       | 158.1    | 4.9  | 9.9       | 366.5    | 32.1 | 48.6       | 371.6      | 393.2 |
| (100,4,50)   | 9.4       | 36.3     | 5.0  | 9.0       | 40.0     | 5.9  | 48.0       | 75.1       | 103.9 |
| (150,4,75)   | 12.5      | 131.6    | 33.0 | 10.9      | 153.7    | 46.4 | 211.4      | 841.7      | 1729.6 |
| (200,4,100)  | 10.2      | 249.1    | 17.8 | 9.7       | 398.9    | 56.8 | 139.2      | 1049.6     | 1096.1 |
| (100,5,50)   | 9.7       | 49.0     | 5.7  | 9.4       | 47.3     | 6.0  | 42.7       | 73.3       | 96.1  |
| (150,5,75)   | 10.4      | 148.4    | 11.5 | 9.6       | 174.1    | 15.9 | 107.0      | 434.8      | 427.6 |
| (200,5,100)  | 10.3      | 479.0    | 38.6 | 9.8       | 550.5    | 59.2 | 101.1      | 931.0      | 830.7 |
| r = \{ 4 \}  |           |          |     |           |          |     |           |          |     |
| (100,2,66)   | 10.3      | 19.3     | 2.5  | 10.1      | 42.9     | 8.3  | 146.1      | 168.6      | 181.8 |
| (150,2,100)  | 13.4      | 60.1     | 28.1 | 10.0      | 157.0    | 28.9 | 76.8       | 198.9      | 264.0 |
| (200,2,133)  | 10.5      | 110.6    | 7.4  | 10.6      | 465.5    | 53.3 | 105.0      | 519.6      | 876.1 |
| (100,3,66)   | 9.7       | 25.9     | 2.5  | 9.2       | 42.7     | 6.6  | 146.4      | 179.5      | 399.1 |
| (150,3,100)  | 9.6       | 77.3     | 8.8  | 9.5       | 166.7    | 37.3 | 60.4       | 198.0      | 287.0 |
| (200,3,133)  | 10.0      | 181.9    | 9.0  | 9.3       | 453.2    | 67.2 | 114.1      | 658.1      | 1102.6 |
| (100,4,66)   | 10.1      | 39.1     | 3.5  | 9.5       | 50.2     | 5.0  | 59.0       | 101.3      | 132.1 |
| (150,4,100)  | 10.2      | 128.5    | 10.3 | 9.8       | 204.7    | 26.5 | 91.3       | 337.7      | 430.5 |
| (200,4,133)  | 10.7      | 341.6    | 45.8 | 10.4      | 575.1    | 95.7 | 243.8      | 1497.5     | 1597.9 |
| (100,5,66)   | 11.1      | 59.2     | 9.5  | 10.7      | 62.9     | 11.6 | 80.7       | 523.1      | 862.5 |
| (150,5,100)  | 9.6       | 181.1    | 12.0 | 9.5       | 224.3    | 27.6 | 113.2      | 443.2      | 546.9 |
| (200,5,133)  | 9.7       | 476.0    | 30.9 | 9.6       | 612.3    | 72.8 | 75.2       | 699.2      | 589.2 |
relatively small. In addition, the comparison of standard deviation of CPU times for the three algorithms shows that the proposed algorithm is more robust than the other two algorithms.

5. Conclusion. The classical SDP relaxation provides a tight lower bound for quadratically constrained quadratic program, but it requires heavy computational effort due to the high dimension of semidefinite constraint. Based on the difference of convex decomposition scheme and mixed SOCP-SDP relaxation method, we derive a low-dimensional SDP relaxation by utilizing the simultaneous diagonalization. We compare the proposed relaxation with SOCP-SDP relaxation and classical SDP relaxation, respectively, and show that it is as tight as the classical SDP relaxation under certain cases. We also prove that the proposed relaxation is equivalent to a special SOCP-SDP relaxation, but is relatively easier to solve since the proposed relaxation has one fewer semidefinite matrix constraint. Furthermore, we develop a spatial branch-and-bound algorithm based on the proposed relaxation. The computational results show that the proposed algorithm is superior to two state-of-the-art algorithms for trust-region problem with linear inequality constraints. For the generalized Celis-Dennis-Tapia problem, the proposed algorithm performs better when problem size is large and the number of quadratic constraints is small.

Acknowledgments. This work has been supported by National Natural Science Foundation of China under Grant Numbers 11701512, 11501543, Zhejiang Provincial Natural Science Foundation of China under Grant Number LQ16A010010 and the Swiss Government Excellence Scholarship.

REFERENCES

[1] K. Anstreicher, Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming, Journal of Global Optimization, 43 (2009), 471–484.
[2] A. Ben-Tal and D. Hertog, Hidden conic quadratic representation of some nonconvex quadratic optimization problems, Mathematical Programming, 143 (2014), 1–29.
[3] ø. Bergmann and T. Steihaug, Solving trust-region subproblem augmented with linear inequality constraints, Optimization Methods and Software, 28 (2013), 26–36.
[4] D. Bienstock, A note on polynomial solvability of the CDT problem, SIAM Journal on Optimization, 26 (2016), 488–498.
[5] I. M. Bomze, M. Locatelli and F. Tardella, New and old bounds for standard quadratic optimization: Dominance, equivalence and incomparability, Mathematical Programming, 115 (2008), 31–64.
[6] I. M. Bomze and M. L. Overton, Narrowing the difficulty gap for the Celis-Dennis-Tapia problem, Mathematical Programming, 151 (2015), 459–476.
[7] C. Buchheim and A. Wiegele, Semidefinite relaxations for non-convex quadratic mixed-integer programming, Mathematical Programming, 141 (2013), 435–452.
[8] S. Burer, S. Kim and M. Kojima, Faster, but weaker, relaxations for quadratically constrained quadratic programs, Computational Optimization and Applications, 59 (2014), 27–45.
[9] S. Burer and B. Yang, The trust region subproblem with non-intersecting linear constraints, Mathematical Programming, 149 (2015), 253–264.
[10] J. Chen and S. Burer, Globally solving nonconvex quadratic programming problems via completely positive programming, Mathematical Programming Computation, 4 (2012), 33–52.
[11] J. Dai, S.-C. Fang and W. Xing, Recovering optimal solutions via SOC-SDP relaxation of trust region subproblem with nonintersecting linear constraints, Journal of Industrial and Management Optimization, (2018).
[12] Z. Deng, S.-C. Fang, Q. Jin and C. Lu, Conic approximation to nonconvex quadratic programming with convex quadratic constraints, Journal of Global Optimization, 61 (2015), 459–478.
[13] D. Y. Gao, Solutions and optimality criteria to box constrained nonconvex minimization problems, *Journal of Industrial and Management Optimization*, 3 (2007), 293–304.

[14] V. Jeyakumar and G. Y. Li, Trust-region problems with linear inequality constraints: Exact SDP relaxation, global optimality and robust optimization, *Mathematical Programming*, 147 (2014), 171–206.

[15] S. Kim and M. Kojima, Second order cone programming relaxation of nonconvex quadratic optimization problems, *Optimization Methods and Software*, 15 (2001), 201–224.

[16] C. Lu, Z. Deng and Q. Jin, An eigenvalue decomposition based branch-and-bound algorithm for nonconvex quadratic programming problems with convex quadratic constraints, *Journal of Global Optimization*, 67 (2017), 475–493.

[17] Z. Q. Luo, W. K. Ma, A. M. C. So, Y. Ye and S. Zhang, Semidefinite relaxation of quadratic optimization problems, *IEEE Signal Processing Magazine*, 27 (2010), 20–34.

[18] P. M. Pardalos and S. A. Vavasis, Quadratic programming with one negative eigenvalue is NP-hard, *Journal of Global Optimization*, 1 (1991), 15–22.

[19] J. F. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, *Optimization Methods and Software*, 11/12 (1999), 625–653.

[20] J. F. Sturm and S. Zhang, On cones of nonnegative quadratic functions, *Mathematics of Operations Research*, 28 (2003), 246–267.

[21] J. Wang, J. Lu and Y. Feng, Congruence diagonalization of two Hermite matrices simultaneously, *International Journal of Algebra*, 4 (2010), 1119–1125.

[22] X. Zheng, X. Sun and D. Li, Nonconvex quadratically constrained quadratic programming: Best DC decompositions and their SDP representations, *Journal of Global Optimization*, 50 (2011), 695–712.

[23] J. Zhou, Z. Deng, S.-C. Fang and W. Xing, Detection of a copositive matrix over a p-th order cone, *Pacific Journal of Optimization*, 10 (2014), 593–611.

[24] J. Zhou, S.-C. Fang and W. Xing, Conic approximation to quadratic optimization with linear complementarity constraints, *Computational Optimization and Applications*, 66 (2017), 97–122.

[25] J. Zhou and Z. Xu, A simultaneous diagonalization based SOCP relaxation for convex quadratic programs with linear complementarity constraints, *Optimization Letters* (2018).

Received October 2018; 1st revision November 2018; 2nd revision December 2018.

E-mail address: zhoujing@zjut.edu.cn
E-mail address: zhibindeng@ucas.edu.cn