Journal of Optimization Theory and Applications (2022) 194:246–264
https://doi.org/10.1007/s10957-022-02018-x

On Indefinite Quadratic Optimization over the Intersection of Balls and Linear Constraints

Temadher A. Almaadeed[1] · Saeid Ansary Karbasy[2] · Maziar Salahi[2] · Abdelouahed Hamdi[1]

Received: 5 April 2021 / Accepted: 26 February 2022 / Published online: 30 March 2022

Abstract
In this paper, we study the minimization of an indefinite quadratic function over the intersection of balls and linear inequality constraints (QOBL). Using the hyperplanes induced by the intersection of each pair of balls, we show that the optimal solution of QOBL can be found by solving several extended trust-region subproblems (e-TRS). To solve e-TRS, we use the alternating direction method of multipliers approach and a branch and bound algorithm. Numerical experiments show the efficiency of the proposed approach compared to the CVX and the extended adaptive ellipsoid-based algorithm.

Keywords Quadratically constrained quadratic optimization problems · Extended trust region subproblems · Nonconvex optimization

Mathematics Subject Classification 49J53 · 49K99

1 Introduction

Quadratically constrained quadratic optimization (QCQO) problems arise in various applications and are among the well-studied optimization problems [5, 9, 11, 16, 24, 27, 33]. Special cases of QCQO include the well-known trust region subproblem (TRS) and extended TRS (e-TRS). Though TRS is nonconvex, it has the necessary

Communicated by Qianchuan Zhao.

✉ Abdelouahed Hamdi
abhamdi@qu.edu.qa

1 Department of Mathematics, Statistics and Physics, College of Arts and Sciences, Qatar University, Doha, Qatar

2 Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

and sufficient optimality conditions and exact semidefinite relaxation (SDR) [15, 26]. However, for e-TRS, the necessary and sufficient optimality conditions and the SDR hold under certain assumptions [12, 13, 22]. A variant of QCQO that is minimizing a quadratic function subject to the intersection of the inside and outside of several balls with extra linear constraints is studied in [7]. The authors proposed a Branch and Bound (BB) algorithm to solve it and reported preliminary numerical results.

In this paper, we study a special case of the problem in [7] that minimizes a quadratic function subject to the intersection of several balls and linear inequality constraints (QOBL). Variants of this problem appear for example in solving nonconvex source localization problems and numerical solution of parameter identification [6, 8]. As a special case of QCQO, one may apply algorithms such as the Extended Adaptive Ellipsoid-based (EAE) algorithm to solve QOBL [16, 23]. We show that QOBL can be reduced to m e-TRS using the hyperplanes induced by the intersection of each pair of balls constraints. To solve e-TRS, we utilize alternating direction method of multipliers (ADMM) and the BB algorithm of [7]. The rest of the paper is organized as follows. In Sect. 2, we give our main results, namely reducing QOBL to m e-TRS. In Sect. 3, we briefly discuss the ADMM [10] for solving e-TRS. Finally, in Sect. 4, numerical results are given to show the efficiency of the proposed approach in comparison with some existing algorithms.

2 Main Results

Consider the following quadratic optimization problem with ball and linear inequality constraints:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Ax + a^T x \\ & ||x - c_i||^2 \le \delta_i^2, \quad i \in \mathcal{I} := \{1, \ldots, m\}, \\ & b_k^T x \le \beta_k, \quad k = 1, \ldots, p, \end{aligned} \tag{QOBL}$$

where $A \in \mathbb{R}^{n\times n}$ is a symmetric matrix, $a, c_i, b_k \in \mathbb{R}^n$, $\beta_k \in \mathbb{R}$ and $\delta_i \in \mathbb{R}_+$. When $m = 1$ and $p = 0$, QOBL reduces to the well-known TRS [15] and when $m = 1$ and $p \ge 1$, it reduces to the following e-TRS:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Ax + a^T x \\ & ||x - c||^2 \le \delta^2, \\ & b_k^T x \le \beta_k, \quad k = 1, \ldots, p, \end{aligned} \tag{p-eTRS}$$

that has been widely studied in recent years [1, 2, 12, 13, 17, 21, 22, 25, 28–31].

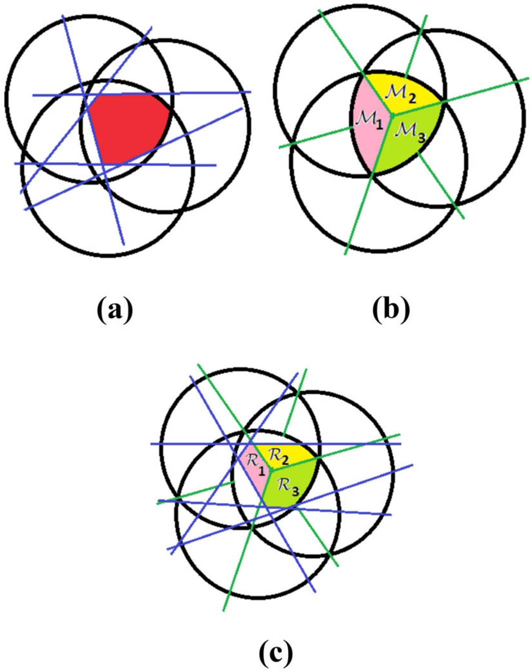

Fig. 1 Graphical representation of notations. **a** The feasible region of QOBL, where the blue lines are linear constraints. **b** $\mathcal{M}_i, i = 1, 2, 3$. **c** $\mathcal{R}_i$, $i = 1, 2, 3$

The following notations are used throughout this section:

$$
\begin{aligned}
&\mathcal{B}_i = \{x \mid ||x - c_i||^2 \le \delta_i^2\}, \quad \partial\mathcal{B}_i = \{x \mid ||x - c_i||^2 = \delta_i^2\},\\
&\mathcal{P} = \{x \mid b_k^T x \le \beta_k, \quad k = 1, \ldots, p\},\\
&\mathcal{M} = \bigcap_{i=1}^{m} \mathcal{B}_i, \quad \mathcal{R} = \mathcal{M} \cap \mathcal{P},\\
&\mathcal{M}_i = \{x \mid x \in \mathcal{B}_i,\ 2(c_i - c_k)^T x \le \alpha_{ik},\ \forall k \in \mathcal{I} \setminus \{i\}\},\\
&\alpha_{ik} = c_i^T c_i - c_k^T c_k - \delta_i^2 + \delta_k^2,\ \ \alpha_{ik} = -\alpha_{ki},\\
&\mathcal{R}_i = \mathcal{M}_i \cap \mathcal{P}.
\end{aligned}
$$

For clarity, we have also shown them in Fig. 1. In the following lemma, we discuss a case where QOBL is infeasible.

Lemma 2.1 *If there exist $i, j \in \mathcal{I}$ such that $||c_i - c_j|| > \delta_i + \delta_j$, then* QOBL *is infeasible (see Fig.* 2a*).*

Proof Let $||c_i - c_j|| > \delta_i + \delta_j$ and $x \in \mathcal{B}_i$, then

$$
\begin{aligned}
||x - c_j|| =& ||x - c_j + c_i - c_i|| \ge ||c_j - c_i|| - ||x - c_i|| > \delta_j + \delta_i - ||x - c_i|| \ge \delta_j\\
&\Longrightarrow ||x - c_j|| > \delta_j \Longrightarrow \mathcal{B}_i \cap \mathcal{B}_j = \emptyset.
\end{aligned}
$$

□

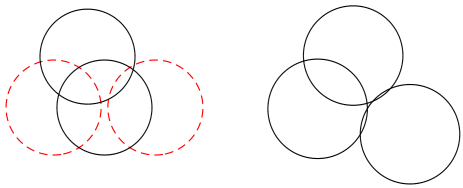

Fig. 2 Infeasible QOBL examples

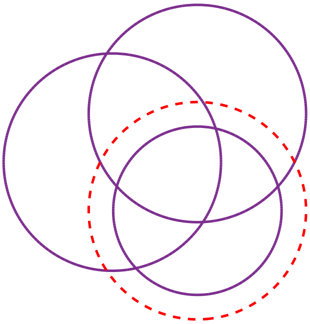

Fig. 3 Red ball is redundant

We will discuss other cases where QOBL becomes infeasible in the rest of the paper (see Fig. 2b). The following lemma discusses the redundancy of ball constraints.

Lemma 2.2 *Let $\delta_i \leq \delta_j$. If $||c_i - c_j|| \leq \delta_j - \delta_i$, then constraint $||x - c_j||^2 \leq \delta_j^2$ is redundant (see Fig. 3).*

Proof Let $x \in \mathcal{B}_i$, then

$$||x - c_j|| = ||x - c_j + c_i - c_i|| \leq ||c_j - c_i|| + ||x - c_i|| \leq \delta_j - \delta_i + ||x - c_i|| \leq \delta_j$$
$$\Longrightarrow \mathcal{B}_i \subset \mathcal{B}_j.$$

Therefore, constraint $||x - c_j||^2 \leq \delta_j^2$ is redundant. □

Following Lemma 2.2, we make the following assumption for the rest of the paper.

Assumption 1 For all $i \in \mathcal{I}$, there is no $j \in \mathcal{I} \setminus \{i\}$ such that $\mathcal{B}_i \subseteq \mathcal{B}_j$ and the Slater condition holds for QOBL. Also, we assume $m \geq 3$.

As noted earlier, the case with $m = 1$ corresponds to the well-studied p-eTRS, see for example [2, 7, 12, 22, 30] and the case with $m = 2$ is handled by a similar approach as in [4]. In the following results, our goal is to characterize the feasible region of QOBL as the union of the feasible region of m, $(m + p - 1)-$eTRS. The first result shows that if $\mathcal{M}_j$ is nonempty, then it has a point on the boundary of $\mathcal{B}_j$.

Lemma 2.3 *Suppose QOBL satisfies Assumption 1. If $\mathcal{M}_j \neq \emptyset$, then there exists $y \in \mathcal{M}_j$ such that $||y - c_j||^2 = \delta_j^2$.*

Proof Let $x \in \mathcal{M}_j$, then $||x - c_j||^2 \leq \delta_j^2$ and $2(c_j - c_i)^T x \leq \alpha_{ji}$ $\forall i \in \mathcal{I} \setminus \{j\}$. If $||x - c_j||^2 < \delta_j^2$, since $2(c_j - c_i)^T x \leq \alpha_{ji}$ $\forall i \in \mathcal{I} \setminus \{j\}$, we have $||x - c_i||^2 - \delta_i^2 \leq ||x - c_j||^2 - \delta_j^2 < 0$. Then, there exist $d \in \mathbb{R}^n$ and $\epsilon > 0$, such that for $y = x + \epsilon d$ we have

$$||y - c_i||^2 - \delta_i^2 \leq ||y - c_j||^2 - \delta_j^2 = 0, \quad \forall i \in \mathcal{I} \setminus \{j\}.$$

Therefore, $||y - c_j||^2 = \delta_j^2$ and $2(c_j - c_i)^T y \leq \alpha_{ji}$ $\forall i \in \mathcal{I} \setminus \{j\}$, which completes the proof. □

In the previous lemma, we showed that when $\mathcal{M}_j$ is nonempty, it intersects the boundary of a ball. In the following lemma, we will show that when $p = 0$ the intersection of $\mathcal{M}_j$ with the boundary of a ball is a part of the boundary of the feasible region of QOBL.

Lemma 2.4 *Suppose QOBL satisfies Assumption 1 and $p = 0$. Then, we have $\mathcal{M}_j \bigcap \partial \mathcal{B}_j = \partial \left(\bigcap_{i=1}^m \mathcal{B}_i \right) \bigcap \partial \mathcal{B}_j$.*

Proof ($\Longrightarrow$) Note that

$$\partial \left(\bigcap_{i=1}^{m} \mathcal{B}_i \right) = \bigcup_{t=1}^{m} \{x | \ ||x - c_t||^2 = \delta_t^2, \ \ ||x - c_i||^2 \leq \delta_i^2, \ \forall i \in \mathcal{I} \setminus \{t\}\}. \qquad (1)$$

Let $x \in \mathcal{M}_j \bigcap \partial \mathcal{B}_j$, then $||x - c_j||^2 = \delta_j^2, \ \ 2(c_j - c_i)^T x \leq \alpha_{ji}, \ \forall i \in \mathcal{I} \setminus \{j\}$. Further,

$$\begin{aligned} 2(c_j - c_i)^T x \leq \alpha_{ji} &\Longrightarrow 2c_j^T x - 2c_i^T x \leq c_j^T c_j - c_i^T c_i + \delta_i^2 - \delta_j^2 \\ &\Longrightarrow x^T x + c_i^T c_i - 2c_i^T x - \delta_i^2 \leq x^T x + c_j^T c_j - 2c_j^T x - \delta_j^2 \\ &\Longrightarrow ||x - c_i||^2 - \delta_i^2 \leq ||x - c_j||^2 - \delta_j^2. \end{aligned}$$

Now, from $||x - c_j||^2 = \delta_j^2$, we have $||x - c_i||^2 \leq \delta_i^2$ for all $i \in \mathcal{I} \setminus \{j\}$, and from (1), we have $x \in \partial \left(\bigcap_{i=1}^m \mathcal{B}_i \right)$. Thus, $\mathcal{M}_j \bigcap \partial \mathcal{B}_j \subseteq \partial \left(\bigcap_{i=1}^m \mathcal{B}_i \right) \bigcap \partial \mathcal{B}_j$.

($\Longleftarrow$) Now, suppose that $x \in \partial \left(\bigcap_{i=1}^m \mathcal{B}_i \right) \bigcap \partial \mathcal{B}_j$. Then, from (1), there exists $j \in \mathcal{I}$ such that $||x - c_j||^2 = \delta_j^2$ and $||x - c_i||^2 \leq \delta_i^2$ for all $i \in \mathcal{I} \setminus \{j\}$ or

$$||x - c_j||^2 = \delta_j^2, \ \ 2(c_j - c_i)^T x \leq \alpha_{ji}, \ \forall i \in \mathcal{I} \setminus \{j\}.$$

This implies $x \in \mathcal{M}_j \bigcap \partial \mathcal{B}_j$. Thus $\partial \left(\bigcap_{i=1}^m \mathcal{B}_i \right) \bigcap \partial \mathcal{B}_j \subseteq \mathcal{M}_j \bigcap \partial \mathcal{B}_j$. □

The following theorem enables us to find redundant ball constraints containing the feasible region, but does not completely contain any of the other ball constraints (see Fig. 4). We should note that these types of redundant ball constraints are not of the type discussed in Lemma 2.2.

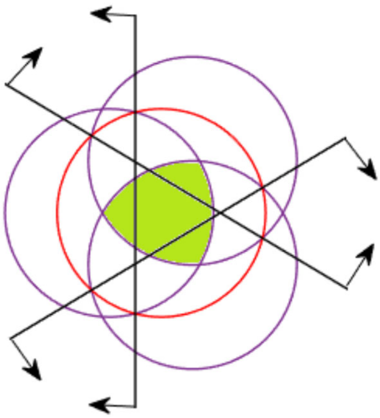

Fig. 4 Red ball is called $\mathcal{B}_1$ and $\mathcal{M}_1$ is empty

Theorem 2.1 *Suppose QOBL satisfies Assumption 1 and $p = 0$. Then, $\mathcal{M}_j = \emptyset$ if and only if $\bigcap_{i=1,i\neq j}^m \mathcal{B}_i \subseteq \mathcal{B}_j$ and $\partial \left(\bigcap_{i=1,i\neq j}^m \mathcal{B}_i\right) \bigcap \partial\mathcal{B}_j = \emptyset$.*

Proof ($\Longleftarrow$) Let $\bigcap_{i=1,i\neq j}^m \mathcal{B}_i \subseteq \mathcal{B}_j$ and $\partial \left(\bigcap_{i=1,i\neq j}^m \mathcal{B}_i\right) \bigcap \partial\mathcal{B}_j = \emptyset$. By contradiction, suppose $\mathcal{M}_j \neq \emptyset$, then from Lemma 2.3, $\mathcal{M}_j \bigcap \partial B_j \neq \emptyset$. Further by Lemma 2.4 and $\bigcap_{i=1}^m \mathcal{B}_i = \bigcap_{i=1,i\neq j}^m \mathcal{B}_i$, we have

$$\mathcal{M}_j \bigcap \partial\mathcal{B}_j = \partial \left(\bigcap_{i=1,i\neq j}^m \mathcal{B}_i\right) \bigcap \partial\mathcal{B}_j. \tag{2}$$

Now, since $\partial \left(\bigcap_{i=1,i\neq j}^m \mathcal{B}_i\right) \bigcap \partial\mathcal{B}_j = \emptyset$, from (2) $\mathcal{M}_j \bigcap \partial\mathcal{B}_j = \emptyset$ which is a contradiction. Thus, $\mathcal{M}_j = \emptyset$.

($\Longrightarrow$) Let $\mathcal{M}_j = \emptyset$. Suppose by contradiction, there exists $x \in \bigcap_{i=1,i\neq j}^m \mathcal{B}_i \setminus \mathcal{B}_j$ such that $||x - c_j||^2 \geq \delta_j^2$. Since $\bigcap_{i=1}^m \mathcal{B}_i \neq \emptyset$, by Assumption 1 there exists $y \in \bigcap_{i=1}^m \mathcal{B}_i$ such that $||y - c_j||^2 \leq \delta_j^2$. Now, let $z_\lambda = \lambda y + (1 - \lambda)x$, then there exist λ^* such that $||z_{\lambda^*} - c_j||^2 = \delta_j^2$. Since $z_{\lambda^*} \in \bigcap_{i=1,i\neq j}^m \mathcal{B}_i$, we have $||z_{\lambda^*} - c_i||^2 \leq \delta_i^2$, $\forall i \in \mathcal{I} \setminus \{j\}$. Then

$$||z_{\lambda^*} - c_i||^2 - \delta_i^2 \leq 0 = ||z_{\lambda^*} - c_j||^2 - \delta_j^2 \Longrightarrow 2(c_j - c_i)^T z_{\lambda^*} \leq \alpha_{ji},\ \forall i \in \mathcal{I} \setminus \{j\}.$$

This means $z_{\lambda^*} \in \mathcal{M}_j$, which is a contradiction with $\mathcal{M}_j = \emptyset$. Therefore,

$$\bigcap_{i=1,i\neq j}^m \mathcal{B}_i \subseteq \mathcal{B}_j.$$

Also, since $\mathcal{M}_j = \emptyset$, we have $\mathcal{M}_j \bigcap \partial\mathcal{B}_j = \emptyset$. Then from Lemma 2.4, $\partial \left(\bigcap_{i=1}^m \mathcal{B}_i\right) \bigcap \partial\mathcal{B}_j = \emptyset$. □

The following theorem, which is the main result of this paper, shows that the feasible region of QOBL is the union of the feasible region of m, $(m+p-1)-$eTRS (see Fig. 5).

Theorem 2.2 *Suppose QOBL satisfies Assumption 1. Then,* $\mathcal{R} = \bigcup_{i=1}^{m} \mathcal{R}_i$.

Proof ($\Longrightarrow$) Suppose $x \in \mathcal{R}$, then $x \in \bigcap_{i=1}^{m} \mathcal{B}_i$ and $x \in \mathcal{P}$. Without loss of generality, we assume that

$$||x - c_1||^2 - \delta_1^2 \leq ||x - c_2||^2 - \delta_2^2 \leq \cdots \leq ||x - c_m||^2 - \delta_m^2.$$

Thus,

$$2(c_m - c_i)^T x \leq \alpha_{mi}, \text{ for all } i \in \mathcal{I} \setminus \{m\} \Longrightarrow x \in \mathcal{M}_m \Longrightarrow x \in \bigcup_{i=1}^{m} \mathcal{M}_i.$$

Since $x \in \mathcal{P}$,

$$\begin{aligned} x \in \left(\bigcup_{i=1}^{m} \mathcal{M}_i\right) \bigcap \mathcal{P} &\Longrightarrow x \in \bigcup_{i=1}^{m} \left(\mathcal{M}_i \bigcap \mathcal{P}\right) \\ &\Longrightarrow x \in \bigcup_{i=1}^{m} \mathcal{R}_i \Longrightarrow \mathcal{R} \subseteq \bigcup_{i=1}^{m} \mathcal{R}_i. \end{aligned} \tag{3}$$

($\Longleftarrow$) Let $x \in \bigcup_{i=1}^{m} \mathcal{R}_i$, then there exists $k \in \mathcal{I}$ such that $x \in \mathcal{R}_k = \mathcal{M}_k \cap \mathcal{P}$. Also

$$x \in \mathcal{M}_k \Longrightarrow 2(c_k - c_i)^T x \leq \alpha_{ki}, \ \forall i \in \mathcal{I} \setminus \{k\},$$

or

$$||x - c_i||^2 - \delta_i^2 \leq ||x - c_k||^2 - \delta_k^2, \ \forall i \in \mathcal{I} \setminus \{k\}.$$

Furthermore, $x \in \mathcal{R}_k$ implies that $||x - c_k||^2 \leq \delta_k^2$. Thus

$$||x - c_i||^2 \leq \delta_i^2, \ \forall i \in \mathcal{I} \Longrightarrow x \in \mathcal{M} \Longrightarrow x \in \mathcal{M} \cap \mathcal{P} \Longrightarrow x \in \mathcal{R}.$$

This implies $\bigcup_{i=1}^{m} \mathcal{R}_i \subseteq \mathcal{R}$. □

Therefore, from Theorem 2.2, solving QOBL reduces to solve m, $(m+p-1)-$eTRS as follows for all $i \in \mathcal{I}$ (see Fig. 5):

$$\begin{aligned} \min \quad & x^T A x + a^T x \\ s.t. \quad & x \in \mathcal{R}_i. \end{aligned} \tag{PR_i}$$

Using Theorem 2.1, in the following lemma, we show that infeasible (PR_i) means a redundant ball constraint (see Fig. 6).

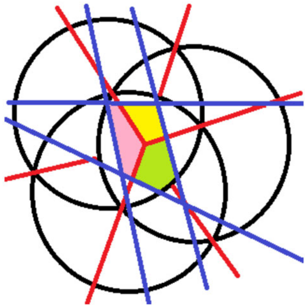

Fig. 5 Feasible region of QOBL when $m = 3$ and $p = 4$. The blue lines are the linear constraints of QOBL

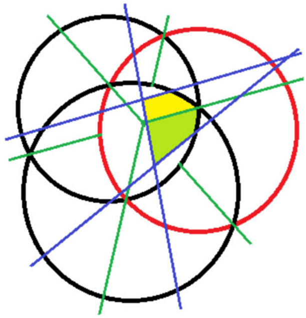

Fig. 6 Red ball is redundant and $\mathcal{R}_i$ related to it is empty

Lemma 2.5 *The* (PR_i) *is feasible if and only if* $||x^*_{c_i} - c_i||^2 \le \delta_i^2$, *where* $x^*_{c_i}$ *is the optimal solution of the following convex quadratic problem:*

$$\begin{aligned} \min \quad & ||x - c_i||^2 && (CR_i) \\ s.t. \quad & 2(c_i - c_j)^T x \le \alpha_{ij}, \quad j \in \mathcal{I} \setminus \{i\}, \\ & x \in \mathcal{P}. \end{aligned}$$

Moreover, If $||x^*_{c_i} - c_i||^2 > \delta_i^2$ *or* (CR_i) *is infeasible, then the* i*th ball constraint is redundant (see Fig. 6).*

Proof The feasibility of (PR_i) is straightforward. If $||x^*_{c_i} - c_i||^2 > \delta_i^2$ or (CR_i) is infeasible, then $\mathcal{R}_i$ is empty, and from Theorem 2.1, the ith ball constraint is redundant. □

Corollary 2.1 *If* $||x^*_{c_i} - c_i||^2 > \delta_i^2$ *or* (CR_i) *is infeasible for all* $i \in \mathcal{I}$, *then* QOBL *is infeasible.*

Based on the previous results, the algorithm for solving QOBL can be outlined as follows.

QOBL algorithm

Step 1: If there exist $i, j \in \mathcal{I}$ such that $||c_i - c_j|| > \delta_i + \delta_j$, then QOBL is infeasible, stop; else go to Step 2.
Step 2: For all $i, j \in \mathcal{I}$ for which $||c_i - c_j|| \leq \delta_j - \delta_i$ and $\delta_j \geq \delta_i$ remove i from $\mathcal{I}$.
Step 3: Solve (CR_i) for all $i \in \mathcal{I}$. For all $i \in \mathcal{I}$ for which $||x^*_{c_i} - c_i||^2 > \delta_i^2$ or feasible region of (CR_i) is infeasible remove i from $\mathcal{I}$. If $\mathcal{I} = \emptyset$, then QOBL is infeasible, stop; else go to Step 4.
Step 4: Solve (PR_i) for all $i \in \mathcal{I}$, and save x_i^*, f_i^*, its optimal solution and optimal objective value.
Step 5: $f_k^* = \min_{i \in \mathcal{I}} \; f_i^*$ and x_k^* are the optimal objective value and global optimal solution of QOBL, respectively.

As we see, the main computational costs of algorithm is solving m, $(m+p-1)-$eTRS. In the next section, we discuss the solution approach for p-eTRS.

3 Solving p-eTRS

As mentioned in the introduction, p-eTRS has been widely studied in recent years. The BB algorithm of [7] is a recent efficient algorithm to solve p-eTRS that we use in our numerical experiments. Also, we utilize the ADMM approach that has been widely used to solve various classes of optimization problems [3, 10, 19, 20, 30, 32]. Consider the following ith $(m+p-1)-$eTRS ($i \in \mathcal{I}$) that arises in the QOBL algorithm:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Ax + a^T x \\ & ||x - c_i||^2 \leq \delta_i^2, \\ & 2(c_i - c_j)^T x \leq \alpha_{ij}, \quad j \in \mathcal{I} \setminus \{i\}, \\ & b_k^T x \leq \beta_k, \quad k = 1, \ldots, p. \end{aligned} \tag{$(m+p-1)-$eTRS}$$

One can write it in the following equivalent form:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Ax + a^T x \\ & ||x - c_i||^2 \leq \delta_i^2, \\ & 2(c_i - c_j)^T z \leq \alpha_{ij}, \quad j \in \mathcal{I} \setminus \{i\}, \\ & b_k^T z \leq \beta_k, \quad k = 1, \ldots, p \\ & x = z. \end{aligned} \tag{4}$$

Now to define the ADMM steps, consider the augmented Lagrangian of (4) as follows:

$$L\left(x, z, \lambda\right) = \frac{1}{2}x^T A x + a^T x + \lambda^T \left(x - z\right) + \frac{\rho}{2}||x - z||^2,$$

where λ_i's are Lagrange multipliers and $\rho \in \mathbb{R}_+$ is the appropriate penalty parameter. Let z_k be a feasible point for $(m + p - 1)-$eTRS that is obtained by solving m (CR_i). The ADMM iterations are as follows:

- **Step 1:** $x^{k+1} = \underset{||x-c_i||^2 \le \delta_i^2}{\operatorname{argmin}} \quad L\left(x, z^k, \lambda^k\right).$
- **Step 2:** $z^{k+1} = \underset{\substack{b_k^T z \le \beta_k,\ k=1,\ldots,p \\ 2(c_i-c_j)^T z \le \alpha_{ij},\ j \in \mathcal{I}\setminus\{i\}}}{\operatorname{argmin}} \quad L\left(x^{k+1}, z, \lambda^k\right).$
- **Step 3:** $\lambda^{k+1} = \lambda^k + \gamma\rho\left(x^{k+1} - z^{k+1}\right)$, where $\gamma \in (0, 1)$ is a constant.

In Step 1, we solve the following TRS:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T \left(A + \rho I_n\right) x + \left(a + \lambda - \rho z^k\right)^T x \\ & ||x - c_i||^2 \le \delta_i^2. \end{aligned} \tag{5}$$

Let x^{k+1} be the optimal solution of (5). In Step 2, we solve the following problem:

$$\begin{aligned} \min \quad & \frac{\rho}{2}z^T z - (\lambda + \rho x^{k+1})^T z \\ & 2(c_i - c_j)^T z \le \alpha_{ij}, \quad j \in \mathcal{I} \setminus \{i\} \\ & b_k^T z \le \beta_k, \quad k = 1, \ldots, p. \end{aligned} \tag{6}$$

As we see, if $\rho \ge -\lambda_{\min}(A)$, then in Step 1 and Step 2, we have convex optimization problems, where $\lambda_{\min}(A)$ is the smallest eigenvalue of A.

It should also be noted that the convergence results for the ADMM algorithms under some mild assumptions are established in [10, 20, 30, 32] for different classes of optimization problems. The convergence of ADMM to the first-order stationary point is given in the following theorem.

Theorem 3.1 ([30]). *Let (x^*, z^*, λ^*) be any accumulation point of $\{(x^k, z^k, \lambda^k)\}$ generated by the ADMM. Then by boundedness assumptions of $\{\lambda^k\}$ and $\sum_{k=0}^{\infty} ||\lambda^{k+1} - \lambda^k||^2 < \infty$, x^* satisfies the first-order stationary conditions.*

4 Numerical Results

In this section, we compare the QOBL algorithm with CVX [18] (solves the semidefinite programming (SDP) relaxation of QOBL) and the EAE algorithm [16, 23]. The

Table 1 Notations in the tables

Notation	Description
n	Dimension of problem
m	Number of ball constraints
Den	Density of A
CPU(ADMM)	Run time of the QOBL algorithm with ADMM
CPU(BB)	Run time of the QOBL algorithm with the BB algorithm of [7]
CPU(CVX)	Run time of CVX
CPU(EAE)	Run time of the EAE algorithm of [16]
F_{ADMM}	Objective value of the QOBL algorithm with ADMM
F_{BB}	Objective value of the QOBL algorithm with the BB algorithm [7]
F_{CVX}	Objective value of CVX
F_{EAE}	Objective value of EAE algorithm of [16]

SDP relaxation of QOBL is as follows:

$$\begin{aligned} \min \quad & \mathrm{Trace}(AX) + a^T x \\ s.t. \quad & \mathrm{Trace}(X) - 2c_i^T x + ||c_i||^2 - \delta_i^2 \leq 0, \quad i \in \mathcal{I}, \\ & b_k^T x \leq \beta_k, \quad k = 1, \ldots, p, \\ & X \succeq xx^T, \end{aligned} \tag{SDP}$$

which is exact when

$$\dim\Big(\mathrm{Ker}\ (A - \lambda_{\min}(A) I_n)\Big) \geq m + p + 1, \tag{7}$$

where I_n is the identity matrix and $\mathrm{Ker}(A) := \{d \in \mathbb{R}^n | Ad = 0\}$ [14]. To solve p-eTRS within the QOBL algorithm, we use the BB algorithm of [7] and ADMM. Implementation is done in MATLAB R2017a on a 2.50 GHz laptop with 8 GB of RAM, and the results in tables are the average of 10 runs for each dimension. It is worth noting that p-eTRSs inside the QOBL algorithm can be solved in parallel. We report the results for both parallel and non-parallel implementations. (CPU time in parentheses are for the parallel version.) The used machine allows solving two p-eTRSs in parallel.

We generate instances of QOBL such that the Slater condition holds. To this end, first we generate a random matrix $C \in \mathbb{R}^{n \times m}$. Let c_i, $i \in \mathcal{I}$ be the columns of the matrix C. Then, we set $y \in \mathbb{R}^n$ as the convex combination of the columns of C, i.e.,

$$y = \sum_{i=1}^{m} \lambda_i c_i, \quad \text{such that} \quad \sum_{i=1}^{m} \lambda_i = 1, \quad \lambda_i \geq 0. \tag{8}$$

Next, we set $\delta_i = ||c_i - y|| + \epsilon_i \ \forall i \in \mathcal{I}$, where $\epsilon_i \in (0, 1)$.

Table 2 Comparison of the QOBL algorithm with CVX when SDP relaxation of QOBL is exact

n	m	CPU(ADMM)	CPU(CVX)	$F_{\text{ADMM}} - F_{\text{CVX}}$
5	3	2.06 (0.38)	2.91	-5.86×10^{-8}
10	7	1.88 (1.11)	2.22	-4.03×10^{-8}
20	12	2.04 (1.95)	2.28	-7.17×10^{-8}
50	20	7.19 (5.12)	2.53	-3.21×10^{-8}
70	30	10.15 (8.31)	3.25	-4.17×10^{-8}
100	10	9.86 (4.06)	3.04	-7.25×10^{-7}
200	10	15.43 (10.56)	5.61	-5.49×10^{-8}
300	10	21.36 (16.57)	12.53	-7.62×10^{-8}
500	10	155.13 (32.55)	–	–
1000	5	85.27 (36.89)	–	–

"–" in all tables means the algorithm cannot solve the problem

- **Test class 1:** In this class, we compare the QOBL algorithm when using ADMM with CVX (that solves the SDP relaxation). To do so, we consider $m < n$ and generate $A \in \mathbb{R}^{n\times n}$ randomly such that multiplicity of its smallest eigenvalue is greater than m and $p = 0$. Therefore, the SDP relaxation is exact. The results are reported in Table 2. As we see, for dimensions $50 \leq n \leq 300$, CVX is better in terms of CPU time, while for the rest of the problems, the QOBL algorithm is faster and CVX cannot solve larger problems. The parallel version of the QOBL algorithm also shows significant CPU time reduction for larger problems.
- **Test class 2:** In this class, we compare the QOBL algorithm and EAE algorithm of [16]. To solve p-eTRS inside the QOBL algorithm, we use the BB algorithm of [7] and ADMM. We generate $A \in \mathbb{R}^{n\times n}$ randomly and we set $p = 0$. The results are summarized in Table 3. As we see, the EAE algorithm is able to solve problems for $n \leq 100$ and except for one instance, the non-parallel QOBL algorithm with ADMM is always faster than it, while they have almost equal objective values. Also, when $m \leq 20$, the non-parallel QOBL algorithm with the BB algorithm is faster than the EAE algorithm in terms of CPU time. When the number of ball constraints is increasing, the QOBL algorithm with ADMM is better than the QOBL algorithm with the BB algorithm in terms of CPU time, while having almost equal objective values. Here, also we see significant time reduction of the parallel QOBL algorithm. Also, in both parallel and non-parallel versions, the QOBL algorithm with ADMM is faster than the QOBL algorithm with the BB algorithm for $m > 10$.
- **Test class 3:** In this class, we compare the QOBL algorithm and EAE algorithm, when $p \neq 0$. By considering y as in (8), we add linear inequality constraints as follows:

  1- Generate $b_k \in \mathbb{R}^n$ for $k = 1, \ldots, p$ randomly,
  2- Let $\beta_k = b_k^T y + \epsilon$ where $\epsilon \in (0, 1)$.

Table 3 Comparison of the QOBL algorithm with the EAE algorithm when $m \geq 5$ and $p = 0$

n	m	CPU(BB)	CPU(ADMM)	CPU(EAE)	$F_{ADMM} - F_{BB}$	$F_{ADMM} - F_{EAE}$
5	5	2.38 (0.35)	4.63 (0.87)	8.54	-1.12×10^{-8}	-9.00×10^{-7}
	10	2.96 (0.45)	5.70 (2.11)	9.31	-3.38×10^{-10}	-5.54×10^{-7}
	20	5.9 (1.65)	10.91 (2.95)	9.38	-5.86×10^{-8}	-2.38×10^{-7}
	50	68.55 (8.65)	22.75 (5.24)	70.31	-1.17×10^{-8}	-2.31×10^{-7}
	100	138.28 (35.26)	121.11 (25.62)	661.94	-1.65×10^{-7}	-4.27×10^{-7}
10	5	2.85 (0.32)	5.94(1.91)	11.35	-1.31×10^{-7}	-1.16×10^{-7}
	10	7.82 (3.68)	7.03 (3.54)	10.18	-9.16×10^{-7}	-1.41×10^{-6}
	20	33.72 (8.56)	18.80 (6.73)	150.36	-2.55×10^{-8}	-1.65×10^{-6}
	50	86.64 (36.48)	40.70 (20.25)	1008.95	-7.15×10^{-8}	-6.32×10^{-6}
	100	> 3000 $(*)^2$	179.07 (72.89)	1152.65	–	-5.85×10^{-6}
20	5	2.39 (0.41)	6.58 (2.88)	21.54	-1.80×10^{-7}	-6.62×10^{-7}
	10	17.52 (1.45)	11.57 (5.61)	32.25	-1.79×10^{-7}	-1.70×10^{-7}
	20	> 3000 (32.87)	81.79 (8.91)	350.26	–	-5.97×10^{-6}
	50	> 3000 $(*)$	82.28(32.26)	1027.26	–	−0.2440
	100	> 3000 $(*)$	496.52 (110.65)	1028.81	–	−0.8615
50	5	3.46 (0.34)	10.26 (2.47)	65.13	-1.35×10^{-7}	-1.93×10^{-7}
	10	25.41 (2.89)	18.49 (6.37)	72.26	-3.94×10^{-7}	-1.56×10^{-7}
	20	> 3000 (192.94)	72.68(19.44)	94.12	–	-1.15×10^{-7}
	50	> 3000 $(*)$	125.59 (76.73)	1237.72	–	−2.9912
	100	> 3000 $(*)$	850.36 (245.35)	1489.81	–	−6.2349

Table 3 continued

n	m	CPU(BB)	CPU(ADMM)	CPU(EAE)	$F_{\text{ADMM}} - F_{\text{BB}}$	$F_{\text{ADMM}} - F_{\text{EAE}}$
100	5	3.12 (0.82)	15.63 (2.15)	295.33	-2.25×10^{-7}	-1.33×10^{-7}
	10	41.04 (11.66)	50.33 (9.33)	334.37	-1.08×10^{-7}	-5.48×10^{-6}
	20	> 3000 (905.35)	56.17 (27.98)	453.94	–	−0.0014
	50	> 3000 (∗)	225.76 (138.10)	1039.45	–	−12.568
200	5	3.7029 (1.51)	34.85 (6.73)	–	1.36×10^{-7}	–
	10	67.14 (32.77)	73.06 (15.48)	–	-1.91×10^{-7}	–
	20	241.75 (∗)	56.85 (51.75)	–	2.31×10^{-7}	–
	50	> 3000 (∗)	149.59 (132.88)	–	–	–
500	5	37.37 (10.40)	16.67(16.20)	–	-2.13×10^{-7}	–
	10	12.36 (625.25)	51.36 (33.71)	–	-6.58×10^{-7}	–
	20	720.16 (∗)	180.61 (123.48)	–	-3.25×10^{-7}	–
	50	> 5000 (∗)	594.12 (394.73)	–	–	–
1000	5	132.73(∗)	360.06 (44.78)	–	-1.71×10^{-7}	–
	10	1654.73 (∗)	375.60 (99.42)	–	-1.68×10^{-7}	–
	20	> 5000 (∗)	975.36 (265.64)	–	–	–

(∗) means the code of [7] either gives error or exponential number of nodes needed to solve p-eTRS

Table 4 Comparison of the QOBL algorithm with the EAE algorithm when $p \neq 0$

n	m	p	CPU(BB)	CPU(ADMM)	CPU(EAE)	$F_{ADMM} - F_{BB}$	$F_{ADMM} - F_{EAE}$
5	5	5	2.08 (0.37)	4.70 (1.15)	874.49	2.10×10^{-8}	-4.66×10^{-7}
	10	10	6.28 (0.96)	6.75 (1.33)	1004.45	-4.63×10^{-8}	-4.66×10^{-8}
	20	20	6.45 (2.83)	11.03 (1.66)	1020.81	-4.52×10^{-9}	-1.51×10^{-6}
	50	50	11.61 (3.69)	9.18 (2.35)	439.26	-3.61×10^{-8}	-7.22×10^{-7}
10	5	5	2.32 (0.48)	8.41 (2.22)	1001.91	7.96×10^{-8}	−1.21
	10	10	28.04 (6.89)	15.09 (4.82)	1004.71	-1.21×10^{-8}	−0.6144
	20	20	264.18 (162.34)	11.73 (4.61)	1018.21	-6.52×10^{-8}	−0.1564
	50	50	796.73 (242.39)	33.08 (10.39)	1019.10	-1.40×10^{-7}	−0.1114
20	5	5	5.55 (0.84)	11.70 (2.49)	47.76	-1.73×10^{-7}	-1.74×10^{-8}
	10	10	366.27 (134.69)	11.96 (4.61)	1002.50	-2.06×10^{-8}	−6.1934
	20	20	> 3000 (∗)	31.32 (10.71)	1024.40	–	−4.0826
	50	50	> 3000 (∗)	240.07 (30.65)	1068.40	–	−2.85
50	5	5	2.67(1.56)	8.28 (3.50)	75.34	1.01×10^{-8}	-9.15×10^{-8}
	10	10	1829.40 (824.36)	16.34 (8.04)	87.27	1.38×10^{-8}	-1.13×10^{-8}
	20	20	> 3000 (∗)	134.21 (25.89)	1026.80	–	−19.52
	50	50	> 3000 (∗)	463.37 (165.51)	1014.01	–	−6.7394

Table 4 continued

n	m	p	CPU(BB)	CPU(ADMM)	CPU(EAE)	$F_{ADMM} - F_{BB}$	$F_{ADMM} - F_{EAE}$
100	5	5	3.53 (2.03)	8.44 (4.10)	341.59	1.34×10^{-7}	-9.26×10^{-7}
	10	10	> 3000 $(*)$	30.12 (11.71)	411.88	–	-9.88×10^{-7}
	20	20	> 3000 $(*)$	149.22 (35.63)	1159.50	–	−90.6684
	50	50	> 3000 $(*)$	710.01 (295.63)	–	–	–
200	5	5	8.42(5.99)	20.00 (7.87)	–	-1.25×10^{-7}	–
	10	10	> 5000 $(*)$	53.60 (18.21)	–	–	–
	20	20	> 5000 $(*)$	350.87 (75.69)	–	–	–
500	5	5	80.28(41.72)	37.96 (21.38)	–	-1.19×10^{-7}	–
	10	10	> 5000 $(*)$	111.12 (49.70)	–	–	–
	20	20	> 5000 $(*)$	442.56 (145.86)	–	–	–
1000	5	5	190.85(146.62)	138.15 (38.64)	–	-3.51×10^{-7}	–
	10	10	> 5000 $(*)$	314.97 (104.72)	–	–	–
	20	20	> 5000 $(*)$	1695.30 (346.41)	–	–	

Table 5 Results of parallel QOBL algorithm when $m \geq n$ and $p = 0$

n	m	CPU(ADMM)
50	200	417.1
	300	1016.7
	500	2074.5
100	200	1641.3
	300	4872.2
	500	7351.8
200	200	4904.9
	300	9366.4
	500	15452.3
300	300	18404.3
	400	24544.6
	500	43344.2
500	500	52724.3
	600	112726.1
	1000	225741.6

Therefore, y as given in (8) is an interior point of QOBL. The corresponding results are summarized in Table 4. A similar observation as in the previous tables also hold here and the QOBL algorithm in overall, performs better than the EAE algorithm. Also, when the number of ball and linear constraints, and dimensions are increasing, the QOBL algorithm with ADMM is the best among all.

– **Test class 4:** In this class, we apply the parallel QOBL algorithm to instances when $m \geq n$. We generate $A \in \mathbb{R}^{n\times n}$ randomly, and we set $p = 0$. To solve $(m + p - 1)-$eTRS inside the QOBL algorithm, we use ADMM. The results are summarized in Table 5 that can be further enhanced by running on cluster machines.

5 Conclusions

In this paper, we studied an indefinite quadratic minimization problem with balls and linear inequality constraints (QOBL). We showed that by solving several extended trust-region subproblems (e-TRS), the optimal solution of QOBL can be found. Our experiments showed that when SDP relaxation is exact, the new approach is better than CVX for larger dimensions. For general instances, our comparison with the EAE algorithm of [7] showed that the new approach is significantly faster. Also using ADMM for solving e-TRS, inside the QOBL algorithm, for majority of problems is faster than the BB algorithm of [7]. Parallelization also is another important feature of the QOBL algorithm.

Acknowledgements The authors would like to thank the reviewer for useful comments and suggestions and Qatar University for supporting their project under Grant NCBP-QUCP-CAS-2020-1.

Funding Open Access funding provided by the Qatar National Library.

References

1. Almaadeed, T.A., Taati, A., Salahi, M., Hamdi, A.: The generalized trust-region sub-problem with additional linear inequality constraints—two convex quadratic relaxations and strong duality. Symmetry **12**(8), 1369 (2020)
2. Ansary Karbasy, S., Hamdi, A., Salahi, M., Taati, A.: An efficient algorithm for large-scale extended trust-region subproblems with non-intersecting linear constraints. Optim. Lett. 1–22
3. Ansary Karbasy, S., Salahi, M.: A hybrid algorithm for the two-trust-region subproblem. Comput. Appl. Math. **38**(3), 1–19 (2019)
4. Ansary Karbasy, S., Salahi, M.: Quadratic optimization with two ball constraints. Numer. Algebra Control Optim. **10**(2), 165 (2020)
5. Anstreicher, K.M.: Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. J. Global Optim. **43**(2–3), 471–484 (2009)
6. Banks, H.T., Kunisch, K.: Estimation Techniques for Distributed Parameter Systems. Springer, New York (2012)
7. Beck, A., Pan, D.: A branch and bound algorithm for nonconvex quadratic optimization with ball and linear constraints. J. Global Optim. **69**(2), 309–342 (2017)
8. Beck, A., Stoica, P., Li, J.: Exact and approximate solutions of source localization problems. IEEE Trans. Signal Process. **56**(5), 1770–1778 (2008)
9. Ben-Tal, A., Nemirovski, A.: Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications. Society for Industrial and Applied Mathematics (2001)
10. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Mach. Learn. **3**(1), 1–122 (2010)
11. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2009)
12. Burer, S., Anstreicher, K.M.: Second-order-cone constraints for extended trust-region subproblems. SIAM J. Optim. **23**(1), 432–451 (2013)
13. Burer, S., Yang, B.: The trust region subproblem with non-intersecting linear constraints. Math. Program. **149**(1), 253–264 (2015)
14. Burer, S., Ye, Y.: Exact semidefinite formulations for a class of (random and non-random) nonconvex quadratic programs. Math. Program. **181**, 1–17 (2019)
15. Conn, A.R., Gould, N.I., Toint, P.L.: Trust Region Methods. SIAM (2000)
16. Deng, Z., Fang, S.C., Jin, Q., Lu, C.: Conic approximation to nonconvex quadratic programming with convex quadratic constraints. J. Global Optim. **61**(3), 459–478 (2015)
17. Fallahi, S., Salahi, M., Karbasy, S.A.: On SOCP/SDP formulation of the extended trust region subproblem. Iran. J. Oper. Res. **9**(2), 03–14 (2018)
18. Grant, M., Boyd, S.: CVX: MATLAB software for disciplined convex programming, version 2.1 (2014)
19. Hajinezhad, D., Shi, Q.: Alternating direction method of multipliers for a class of nonconvex bilinear optimization: convergence analysis and applications. J. Global Optim. **70**(1), 261–288 (2018)
20. Hong, M., Luo, Z.Q., Razaviyayn, M.: Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. SIAM J. Optim. **26**(1), 337–364 (2016)
21. Hsia, Y., Sheu, R.L.: Trust region subproblem with a fixed number of additional linear inequality constraints has polynomial complexity. arXiv preprint arXiv:1312.1398 (2013)

22. Jeyakumar, V., Li, G.: Trust-region problems with linear inequality constraints: exact SDP relaxation, global optimality and robust optimization. Math. Program. **147**(1), 171–206 (2014)
23. Keyanpour, M., Osmanpour, N.: On solving quadratically constrained quadratic programming problem with one non-convex constraint. Opsearch **55**(2), 320–336 (2018)
24. Lipp, T., Boyd, S.: Variations and extension of the convex-concave procedure. Optim. Eng. **17**(2), 263–287 (2016)
25. Locatelli, M.: Exactness conditions for an SDP relaxation of the extended trust region problem. Optim. Lett. **10**(6), 1141–1151 (2016)
26. Nesterov, Y., Wolkowicz, H., Ye, Y.: Semidefinite Programming Relaxations of Nonconvex Quadratic Optimization. Handbook of Semidefinite Programming, pp. 361–419. Springer, Boston (2000)
27. Park, J., Boyd S.: General heuristics for nonconvex quadratically constrained quadratic programming. arXiv preprint arXiv:1703.07870 (2017)
28. Salahi, M., Fallahi, S.: Trust region subproblem with an additional linear inequality constraint. Optim. Lett. **10**(4), 821–832 (2016)
29. Salahi, M., Taati, A.: A fast eigenvalue approach for solving the trust region subproblem with an additional linear inequality. Comput. Appl. Math. **37**(1), 329–347 (2018)
30. Salahi, M., Taati, A.: Alternating direction method of multipliers for the extended trust region subproblem. Iran. J. Numer. Anal. Optim. **7**(1), 107–117 (2017)
31. Salahi, M., Taati, A., Wolkowicz, H.: Local nonglobal minima for solving large-scale extended trust-region subproblems. Comput. Optim. Appl. **66**(2), 223–244 (2017)
32. Xu, L., Yu, B., Zhang, Y.: An alternating direction and projection algorithm for structure-enforced matrix factorization. Comput. Optim. Appl. **68**(2), 333–362 (2017)
33. Zheng, X.J., Sun, X.L., Li, D.: Nonconvex quadratically constrained quadratic programming: best D.C. decomposition and their SDP represetations. J. Global Optim. **50**, 695–712 (2011)