Communications in Combinatorics and Optimization
Vol. xx, No. x (xxxx), pp. 1-13
https://doi.org/10.22049/cco.2025.30641.2561

CCO
Commun. Comb. Optim.

Research Article

Quadratic optimization with a ball and a reverse ball constraints

Abdelouahed Hamdi[1,†], Maziar Salahi[2*], Saeid Ansary Karbasy[2,§], Temadher Almaadeed[1,‡]

[1]Department of Mathematics and Statistics, College of Arts and Sciences, Qatar University, Doha, Qatar
[†]abhamdi@qu.ed.qa
[‡]t.alassiry@qu.ed.qa

[2]Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran
[*]salahim@guilan.ac.ir
[§]saeidansary144@yahoo.com

Received: 20 May 2025; Accepted: 31 July 2025
Published Online: 10 August 2025

Abstract: In this paper, we study a quadratic minimization problem over the intersection of a ball and a reverse ball constraints that includes generalized trust-region subproblem (TRS). Using the structure of the problem, we prove that it can be solved to global optimality by solving at most three TRS or two TRS with an extra linear constraint. Then we present an efficient TRS-based algorithm to solve it. Computational experiments illustrate that our new algorithm outperforms the ones in the literature, specially the algorithm for generalized TRS, on three widely used test classes.

Keywords: nonconvex quadratic program, trust-region subproblem, SDP relaxation.

AMS Subject classification: 90C22, 90C26

1. Introduction

Variants of nonconvex quadratically constrained quadratic programming (QCQP) problem due to their importance are studied in the literature [1, 4, 6, 8, 10, 19, 21]. A widely used technique for tackling QCQPs involves semidefinite programming (SDP) relaxation. This approach often yields exact solutions in specific scenarios, such as the

* *Corresponding Author*

trust-region subproblem (TRS), which is a crucial component in addressing broader nonlinear programming challenges [12]. However, in general the SDP relaxation is not exact. In a most recent work, Burer and Ye gave some sufficient conditions under which SDP relaxation of general QCQPs is exact [9]. For more details on the SDP relaxation and related works on QCQPs, we refer to [24] and references therein.
Some studies also took advantages of the problem structure and developed more specialized efficient algorithms to solve the underlying QCQPs. For example, in [22] the authors have studied a nonconvex quadratic optimization problem with two quadratic constraints, one of them is convex and developed a two-parameters eigenvalue based algorithm to solve it. In [7] the authors have developed a branch and bound algorithm for a quadratic program with balls and reverse ball constraints that solves TRSs at each node. A global search algorithm is developed in [16] which integrates branch and bound method and alternative direction method to solve a variant of QCQPs with a few negative eigenvalues and convex constraints. In [3] the authors developed an efficient generalized eigenvalue-based algorithm for a QCQP with one constraint. Also, in [25] the authors studied a similar problem to [16] and by utilizing simultaneous diagonalization and difference of convex decomposition, developed an efficient second order cone relaxation to solve it.
In this paper, we consider the following quadratic minimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Ax + a^T x \\ & ||x - c_1||^2 \leq \delta_1^2, \\ & ||x - c_2||^2 \geq \delta_2^2, \end{aligned} \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is a symmetric matrix, $a, x, c_1, c_2 \in \mathbb{R}^n$, and $\delta_1, \delta_2 \in \mathbb{R}$. It is a special case of QCQPs and when $c_1 = c_2$ it contains generalized TRS (GTRS) [20]

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Ax + a^T x \\ & l \leq x^T Bx - 2c^T x \leq u, \end{aligned} \tag{1.2}$$

as a special case for positive definite B. An application of (1.1) for example is the robust portfolio optimization models, where the covariance matrix is indefinite due to estimation errors or missing data [8, 11]. The goal here is to minimize risk (variance) subject to lower and upper bounds on the portfolio weights norm which is in the form of (1.1). Other subjects that (1.1) might appear are the so called robust sparse signal recovery and sensor localization problems [7, 15].
The SDP relaxation of (1.1) is as follows:

$$\begin{aligned} \min_{x,X} & \frac{1}{2}\operatorname{trace}(AX) + a^T x \\ & \operatorname{trace}(BX) - 2c_1^T x + c_1^T c_1 \leq \delta_1^2, \\ & \operatorname{trace}(X) - 2c_2^T x + c_2^T c_2 \geq \delta_2^2, \\ & X \succeq xx^T. \end{aligned} \tag{1.3}$$

When the Lagrangian has positive semidefinite Hessian at global solution, this relaxation is exact [18]. However, SDP suffers from high computational costs specially for large-scale problems, which is $O(n^{4.5}\log\frac{1}{\epsilon})$ where ϵ is the desired accuracy. Thus, in this paper, using the structure of problem, we prove that for solving (1), it is sufficient to solve at most three TRSs or two extended TRS (eTRS) (TRS with an additional linear constraint), which is $O(n^3)$. The rest of the paper is organized as follows. Section 2 gives the main results and new algorithm. Section 3, discusses the computational experiments conducted to evaluate the performance of the algorithm compared to the similar algorithms in the literature. Finally, in Section 4 we conclude the paper by summarizing the key findings and suggesting some future research directions.

2. Mains results and algorithm

Here, we show that (1.1) can be solved by solving three TRSs or two eTRSs for which efficient generalized eigenvalue based algorithms are developed in [2, 23]. To do so, we use the following notations:

- $\mathcal{B} = \{x \mid ||x-c_1||^2 \le \delta_1^2\}$, $\partial\mathcal{B} = \{x \mid ||x-c_1||^2 = \delta_1^2\}$,
- $\mathcal{E} = \{x \mid ||x-c_2||^2 \ge \delta_2^2\}$, $\mathcal{E}^c = \{x \mid ||x-c_2||^2 < \delta_2^2\}$,
- $\partial\mathcal{E} = \{x \mid ||x-c_2||^2 = \delta_2^2\}$, $\mathcal{M} = \mathcal{B}\cap\mathcal{E}$,
- $\mathcal{U} = \{x \mid x\in\partial\mathcal{B},\ (c_1-c_2)^T x \ge \alpha_{12}\}$,
- $\mathcal{V} = \{x \mid x\in\partial\mathcal{E},\ (c_1-c_2)^T x \ge \alpha_{12}\}$,
- $\alpha_{12} = \frac{1}{2}\left(c_1^T c_1 - c_2^T c_2 - \delta_1^2 + \delta_2^2\right)$.

The following lemma discusses the case where the second constraint is redundant as shown in Fig. 1(c).

Lemma 1. *If $||c_1-c_2|| > \delta_1+\delta_2$, then $||x-c_2||^2 \ge \delta_2^2$ is redundant.*

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

$$\begin{aligned}||x-c_2|| =& ||x-c_2+c_1-c_1|| \ge ||c_2-c_1|| - ||x-c_1|| > \delta_2+\delta_1-||x-c_1|| > \delta_2\\ &\Longrightarrow ||x-c_2|| > \delta_2 \Longrightarrow x\in\mathcal{E}.\end{aligned}$$

Thus, $\mathcal{B}\subset\mathcal{E}$ and then $\mathcal{B}\cap\mathcal{E}=\mathcal{B}$. □

The infeasible case (Fig. 1(a)) and singleton feasible region also are discussed in the following lemma.

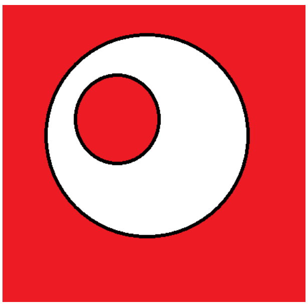
Lemma 2. *Let $\delta_1 \leq \delta_2$. If $||c_1 - c_2|| < \delta_2 - \delta_1$, then (1.1) is infeasible. If $||c_1 - c_2|| = \delta_2 - \delta_1$, and $\delta_2 > \delta_1$ then the feasible region of (1.1) is a singleton.*

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

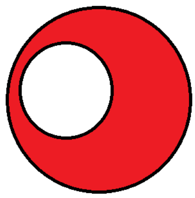
$$||x - c_2|| = ||x - c_2 + c_1 - c_1|| \leq ||c_2 - c_1|| + ||x - c_1|| < \delta_2 - \delta_1 + ||x - c_1|| \leq \delta_2.$$

Thus, $x \notin \mathcal{E}$ and then $\mathcal{B} \cap \mathcal{E} = \emptyset$. The second part follows directly from triangle inequality.

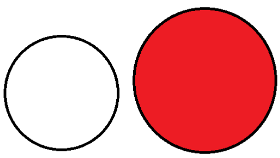
□



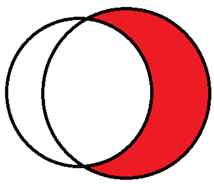
(a) The first constraint circle is inside the circle related to the second constraint, empty feasible region ($\mathcal{B} \subseteq \mathcal{E}^c$).



(b) The second constraint circle is inside the first circle, nonempty feasible region ($\mathcal{E}^c \subseteq \mathcal{B}$).



(c) Two disjoint circles, nonempty feasible region ($\mathcal{B} \subseteq \mathcal{E}$).



(d) The two circles intersect, nonempty feasible region ($\mathcal{B} - \mathcal{E}^c \neq \emptyset$, $\mathcal{E}^c - \mathcal{B} \neq \emptyset$).

Figure 1. ***All possible cases for* (1.1) *in* $\mathbb{R}^2$.**

From Lemma 2, if $\mathcal{B} \subseteq \mathcal{E}^c$ then (1.1) is infeasible (Fig 1(a)). Suppose this is not the case, then the following three cases may occur for x^*, the optimal solution of (1.1):

- **Case 1:** $\mathcal{E}^c \subseteq \mathcal{B}$ (Fig 1(b)). It is clear that in this case $\delta_2 < \delta_1$. The following three subcases may occur:

- **Case 1.1:** x^* is the optimal solution of the following TRS with equality constraint:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Ax + a^T x \\ & ||x - c_1||^2 = \delta_1^2. \end{aligned} \tag{2.1}$$

- **Case 1.2:** x^* is the optimal solution of the following TRS with equality constraint:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Ax + a^T x \\ & ||x - c_2||^2 = \delta_2^2. \end{aligned} \tag{2.2}$$

- **Case 1.3:** x^* is strictly between the boundary of the two circles i.e., $||x^* - c_1||^2 < \delta_1^2$ and $||x^* - c_2||^2 > \delta_2^2$. Therefore, A is positive definite ([17]) and x^* is the optimal solution of the following problem:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Ax + a^T x \\ & ||x - c_1||^2 \le \delta_1^2. \end{aligned} \tag{2.3}$$

- **Case 2:** $\mathcal{B} \subseteq \mathcal{E}$ (Fig 1(c)). From Lemma 1, constraint $||x - c_2||^2 \ge \delta_2^2$ is redundant. Then, for solving (1.1), it is sufficient to solve (2.3).
- **Case 3:** $\mathcal{B} - \mathcal{E}^c \ne \emptyset$, $\mathcal{E}^c - \mathcal{B} \ne \emptyset$ and $\mathcal{B} \cap \mathcal{E} \ne \emptyset$ (Fig 1(d)). In this case, the following two cases may occur:
- **Case 3.1:** $||x^* - c_1||^2 < \delta_1^2$ and $||x^* - c_2||^2 > \delta_2^2$.
This is similar to the Case 1.3.
- **Case 3.2:** $x^* \in \partial\mathcal{M}$. This case is discussed in the following theorem.

Theorem 1. *Let $\mathcal{B} - \mathcal{E}^c \ne \emptyset$, $\mathcal{E}^c - \mathcal{B} \ne \emptyset$ and $\mathcal{B} \cap \mathcal{E} \ne \emptyset$ (Fig 1(d)). Then $\partial\mathcal{M} = \mathcal{U} \cup \mathcal{V}$.*

Proof. ($\Longrightarrow$) Let $x \in \partial\mathcal{M}$, then

$$||x - c_1||^2 = \delta_1^2 \ \text{ or } \ ||x - c_2||^2 = \delta_2^2.$$

- **Case 1.1:** $||x - c_1||^2 = \delta_1^2$. Since $\partial\mathcal{M} \subset \mathcal{M}$, we have $x \in \mathcal{M}$, thus we have $||x - c_2||^2 \ge \delta_2^2$. This further implies that

$$\begin{aligned} & ||x - c_1||^2 - \delta_1^2 = 0 \le ||x - c_2||^2 - \delta_2^2 \\ & \Longrightarrow x^T x - 2c_1^T x + c_1^T c_1 - \delta_1^2 \le x^T x - 2c_2^T x + c_2^T c_2 - \delta_2^2 \\ & \Longrightarrow (c_1 - c_2)^T x \ge \alpha_{12} \Longrightarrow x \in \mathcal{U}. \end{aligned} \tag{2.4}$$

- **Case 1.2:** $||x-c_2||^2 = \delta_2^2$. Also, we have $||x-c_1||^2 \le \delta_1^2$, thus

$$\begin{aligned} & 0 = ||x-c_2||^2 - \delta_2^2 \ge ||x-c_1||^2 - \delta_1^2 \\ & \implies x^Tx - 2c_2^Tx + c_2^Tc_2 - \delta_2^2 \le x^Tx - 2c_1^Tx + c_1^Tc_1 - \delta_2^2 \\ & \implies (c_1-c_2)^Tx \ge \alpha_{12} \implies x \in \mathcal{V}. \end{aligned} \tag{2.5}$$

  From (2.4) and (2.5), we have $\partial\mathcal{M} \subseteq \mathcal{U} \cup \mathcal{V}$.

($\Longleftarrow$) Now, let $x \in \mathcal{U} \cup \mathcal{V}$.

- **Case 2.1:** Let $x \in \mathcal{U}$, then

$$||x-c_1||^2 = \delta_1^2, \quad (c_1-c_2)^Tx \ge \alpha_{12},$$

  and

$$||x-c_1||^2 - \delta_1^2 = 0 \le ||x-c_2||^2 - \delta_2^2 \implies x \in \partial\mathcal{M}. \tag{2.6}$$

- **Case 2.2:** Let $x \in \mathcal{V}$. From definition of $\mathcal{V}$, we have

$$||x-c_2||^2 = \delta_2^2, \quad (c_1-c_2)^Tx \ge \alpha_{12},$$

  therefore

$$0 = ||x-c_2||^2 - \delta_2^2 \ge ||x-c_1||^2 - \delta_1^2 \implies x \in \partial\mathcal{M}.$$

From (2.6) and (2), we have $\mathcal{U} \cup \mathcal{V} \subseteq \partial\mathcal{M}$. □

According to Theorem 1, the optimal solution in Case 3.2 can be found by solving the following two eTRSs:

$$\begin{aligned} \min \quad & \frac{1}{2}x^TAx + a^Tx \\ & ||x-c_1||^2 = \delta_1^2, \\ & (c_1-c_2)^Tx \ge \alpha_{12}, \end{aligned} \tag{2.7}$$

and

$$\begin{aligned} \min \quad & \frac{1}{2}x^TAx + a^Tx \\ & ||x-c_2||^2 = \delta_2^2, \\ & (c_1-c_2)^Tx \ge \alpha_{12}. \end{aligned} \tag{2.8}$$

The above solution procedure for solving (1.1) is summarized in Algorithm 1.

Algorithm 1

Step 0: A (matrix), a (vector), c_1, c_2 (vectors), δ_1, δ_2 (positive scalars).
Outpute: Optimal solution x^* or infeasibility status.
Step 1: Check the feasibility
if $\delta_2 \geq \delta_1$ **and** $\|c_1 - c_2\| < \delta_2 - \delta_1$ **then**
  return "Problem (1.1) is infeasible"
end if
Step 2 (Case 1):
if $\delta_2 \leq \delta_1$ **and** $\|c_1 - c_2\| < \delta_1 - \delta_2$ **then**
  $x_1^* \leftarrow$ Solve subproblem (2.1)
  $x_2^* \leftarrow$ Solve subproblem (2.2)
  $x_3^* \leftarrow$ Solve subproblem (2.3)
Step 3: Select the best solution
  $x^* \leftarrow \text{argmin}_{x \in \{x_1^*, x_2^*, x_3^*\}} \left(\frac{1}{2}x^T Ax + a^T x\right)$
  return x^*
end if
Step 4 (Case 2):
if $\|c_1 - c_2\| > \delta_1 + \delta_2$ **then**
  $x^* \leftarrow$ Solve subproblem (2.3)
  return x^*
end if
Step 5 (Case 3):
$x^* \leftarrow$ Solve subproblem (2.3)
if $x^* \in \mathcal{M}$ **then**
  return x^*
else
  $x_4^* \leftarrow$ Solve subproblem (2.7)
  $x_5^* \leftarrow$ Solve subproblem (2.8)
  $x^* \leftarrow \text{argmin}_{x \in \{x_4^*, x_5^*\}} \left(\frac{1}{2}x^T Ax + a^T x\right)$
  return x^*
end if

3. Numerical experiments

In this section, we compare Algorithm 1 with the AEA algorithm from [13], the RW algorithm from [20], CVX software [14], and the 'fmincon' function from MATLAB across various classes of test problems. It is important to note that CVX is employed only when the SDP relaxation is exact. Additionally, the RW algorithm from [20] is used for comparison when the GTRS instances are generated. All experiments are conducted in MATLAB R2015a on a laptop with a 2.50 GHz processor and 8 GB of RAM. The notations used in the tables can be found in Table 1.

- **Class 1:**
  The following lemma is used to generate test instances of this class.

  Lemma 3 (Lemma 2, [5]). *Let A be a symmetric matrix with $\lambda_1 < \min\{0, \lambda_2\}$. Denote the eigenvector corresponding to λ_1 by v_1. Then, there exists a linear term* **a**

Table 1. Tables' Notations

Notation	Description
n	Dimension of problem
Den	Density of A
CPU(NA)	CPU time of Algorithm 1
CPU(Fmin)	CPU time of the 'fmincon' function of MATLAB
CPU(CVX)	CPU time of CVX
CPU(AEA)	CPU time of the AEA algorithm of [13]
CPU(RW)	CPU time of the RW algorithm of [13]
F_{NA}	Objective function value of Algorithm 1
F_{Fmin}	Objective function value of 'fmincon' function in matlab
F_{CVX}	Objective function value of CVX
F_{AEA}	Objective function value of the AEA algorithm of [13]
F_{RW}	Objective function value of the RW algorithm of [20]
OOM	MATLAB run to out of memory

such that the vector $(v_1 + c_1)$ *represents a local non-global minimum (LNGM) of* (3.1)*:*

$$\begin{aligned} \min \quad & \frac{1}{2} x^T A x + \mathbf{a}^T x \\ & ||x - c_1||^2 \leq \delta_1^2. \end{aligned} \tag{3.1}$$

Moreover $(-v_1 + c_1)$ *is the global solution of (3.1).*

Using this lemma, first we randomly generate a TRS instance of the form (3.1) that has an LNGM. We consider $c_2 = (2\tau - 1)v_1 + \tau c_1$ such that $\tau \geq 1$ and $2|1-\tau|\|v_1\| < \delta_2 < 2\tau\|v_1\|$. Then, the constraint $||x - c_2||^2 \geq \delta_2^2$ is added such that the global minimizer of TRS, $(-v_1 + c_1)$, to be infeasible but the LNGM [23], $(v_1 + c_1)$, remains feasible (See Fig. 2). The corresponding results are summarized in Table 2, where we compare the Algorithm 1 with the AEA algorithm of [13] and 'fmincon' function of MATLAB. From this table, we can conclude that Algorithm 1 solves all instance, while AEA and 'fmincon' are able to solve instances with $n \leq 100$. Algorithm 1 is the fastest, 'fmincon' is the second one and AEA has the worst CPU time.

- **Class 2:**
  First, we generate random TRS instances of the form (3.1) for which hard case 2 occurs [12]. Let x_1^* and x_2^* be two optimal solutions of (3.1). Also let $c_2 = (2\tau - 1)x_1^* + \tau c_1$ such that $\tau \geq 1$ and $2|1-\tau|\|v_1\| < \delta_2 < 2\tau$. In this case, x_1^* is outside of the feasible region of (1.1) and x_2^* is in the feasible region (Fig. 3). Since the optimal solution of TRS (3.1) is inside the feasible region, then the Lagrangian has positive definite Hessian, and as a result SDP relaxation is exact. The comparison results with CVX which solves SDP relaxation, and the AEA algorithm are reported in Table 3. As can be seen, Algorithm 1 solves all instances below 1 second, CVX solves instances up to dimension 500 and AEA

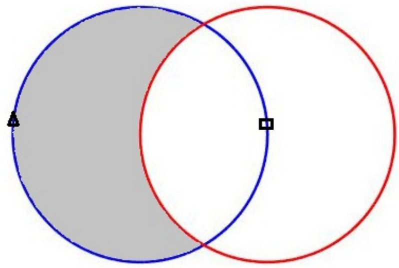


Figure 2. □: ***Global solution of TRS*** (3.1), △: ***LNGMs of TRS*** (3.1).

Table 2. Comparison of objective values and CPU times of Algorithm 1 with the AEA algorithm of [13] and 'fmincon' function of MATLAB for Class 1 instances.

n	CPU(NA)	CPU(AEA)	$F_{NA} - F_{AEA}$	CPU(Fmin)	$F_{NA} - F_{Fmin}$
5	0.25	8.45	3.21×10^{-9}	0.58	-1.99×10^{-6}
10	0.27	11.06	2.58×10^{-9}	0.64	-3.99×10^{-6}
15	0.25	15.50	8.69×10^{-9}	0.67	-3.99×10^{-6}
25	0.25	24.87	4.28×10^{-8}	0.69	-3.99×10^{-6}
30	0.26	29.67	3.84×10^{-8}	0.70	-3.99×10^{-6}
50	0.27	55.54	1.84×10^{-8}	0.85	-7.91×10^{-6}
100	0.27	205.95	4.26×10^{-7}	0.97	-1.98×10^{-4}
150	0.27	–	–	1.28	−0.07
200	0.27	–	–	1.48	2.98
500	0.27	–	–	–	–
700	0.28	–	–	–	–
1000	0.32	–	–	–	–

solves instances up to dimension 15. CVX has the second best CPU time and AEA has the worst.

- **Class 3:**
  In this class, we generate random instances in the form of GTRS to compare with the RW algorithm designed for GTRS [20] as well. Let A be a positive

Table 3. Comparison of objective values and CPU times of Algorithm 1 with the AEA algorithm of [13] and CVX software for Class 2 instances.

n	CPU(NA)	CPU(AEA)	$F_{NA} - F_{AEA}$	CPU(SDP)	$F_{NA} - F_{SDP}$
5	0.03	1001.81	-9.35×10^{-9}	2.06	-7.53×10^{-8}
10	0.06	1053.65	-5.48×10^{-9}	2.11	-4.72×10^{-8}
15	0.09	1008.81	-7.25×10^{-9}	2.15	-7.60×10^{-8}
20	0.08	–	–	2.12	-1.65×10^{-8}
25	0.08	–	–	2.15	-2.39×10^{-8}
30	0.08	–	–	2.18	-3.53×10^{-7}
50	0.08	–	–	2.26	-3.82×10^{-7}
100	0.08	–	–	2.85	-4.38×10^{-6}
200	0.09	–	–	5.51	-4.47×10^{-6}
300	0.14	–	–	12.73	-4.35×10^{-6}
500	0.16	–	–	152.28	-3.85×10^{-6}
700	0.3	–	–	OOM	–
1000	0.9	–	–	OOM	–

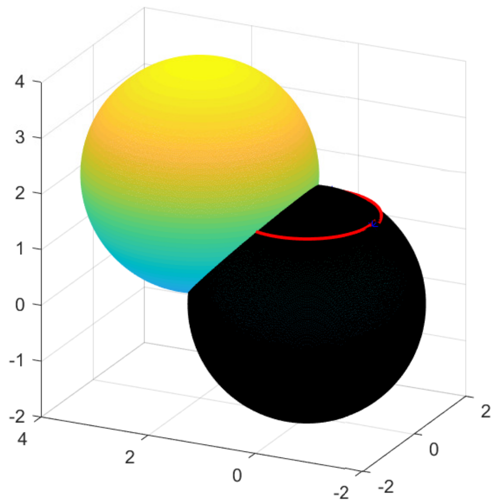


Figure 3. *□: Global solution of TRS, △: LNGMs of TRS.*

semidefinite matrix, c_1 and c_2 be zero vectors and $\delta_1 > 1$. We generate $\bar{x}$ randomly such that $\|\bar{x}\| < \delta_1$. Then we let $a = -A\bar{x}$, thus $\bar{x}$ becomes the optimal solution of TRS (3.1). Now, we choose δ_2 such that $\|\bar{x}\| < \delta_2 < \delta_1$ and thus the optimal solution of TRS (3.1) is not in the feasible region of problem (1.1) (Fig 4). The comparison results are reported in Table 4. As we see, for dimensions $n \leq 3000$, Algorithm 1 and RW yield nearly identical objective function values; however, Algorithm 1 has better CPU time. For $n > 3000$, Algorithm 1 finds global solution while the RW algorithm is not able to do so. As before, AEA algorithm has the worst CPU time and is not able to solve instnces for $n > 100$.

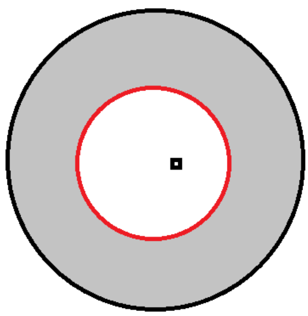


Figure 4. *□: Global solution of TRS.*

Table 4. Comparison of objective values and CPU times of Algorithm 1 with the AEA algorithm of [13] and RW algorithm of [20] for Class 3 instances.

n	Den	CPU(NA)	CPU(AEA)	$F_{NA} - F_{AEA}$	CPU(RW)	$F_{NA} - F_{RW}$
5	1	0.27	13.27	6.75×10^{-9}	0.58	1.45×10^{-12}
10	1	0.27	14.79	2.13×10^{-9}	0.58	1.11×10^{-12}
50	1	0.32	49.64	6.81×10^{-8}	0.84	-1.72×10^{-9}
100	1	0.38	170.03	4.62×10^{-7}	0.96	-2.66×10^{-8}
200	1	0.57	–	–	1.75	-7.63×10^{-10}
300	1	0.62	–	–	1.95	-1.61×10^{-9}
700	1	1.52	–	–	4.23	-2.97×10^{-11}
1500	1	7.31	–	–	16.60	-4.18×10^{-10}
2000	0.1	3.75	–	–	9.34	-1.81×10^{-8}
3000	0.1	4.88	–	–	29.64	-2.93×10^{-8}
4000	0.1	14.82	–	–	35.34	−28.89
5000	0.1	4.35	–	–	30.08	−3.61
7000	0.001	6.85	–	–	54.42	−3.65
10000	0.001	22.40	–	–	174.37	−3.54
20000	0.0001	31.45	–	–	–	–
20000	0.0001	54.69	–	–	–	–
50000	0.0001	204.34	–	–	–	–
100000	0.00001	359.69	–	–	–	–

4. Conclusions

In this paper, we examine a quadratic minimization problem constrained by the intersection of a ball and a reverse ball. By leveraging the problem's structure, we demonstrate that it can be effectively addressed by solving either three trust-region subproblems or two trust-region subproblems with an additional linear constraint. This approach not only simplifies the optimization process but also enhances computational efficiency. We conducted extensive experiments to evaluate the performance of our proposed method in comparison to existing techniques found in the literature. The results indicate that our approach consistently outperforms traditional methods, providing faster convergence and improved accuracy across various test cases. Furthermore, our findings suggest that the structured nature of the nonconvex quadratic minimization problem allows for a more refined exploration of the feasible region and facilitating better optimization outcomes. Extending this approach to the case where there are several ball constraints could be considered as a future work.

Conflict of Interest: The authors declare that they have no conflict of interest.

Data Availability: Data sharing is not applicable to this article as no data sets were generated or analyzed during the current study.

References

[1] E. Abbe, A.S. Bandeira, and G. Hall, *Exact recovery in the stochastic block model*, IEEE Trans. Inf. Theory. **62** (2015), no. 1, 471–487. https://doi.org/10.1109/TIT.2015.2490670.

[2] S. Adachi, S. Iwata, Y. Nakatsukasa, and A. Takeda, *Solving the trust-region subproblem by a generalized eigenvalue problem*, SIAM J. Optim. **27** (2017), no. 1, 269–291.
https://doi.org/10.1137/16M1058200.

[3] S. Adachi and Y. Nakatsukasa, *Eigenvalue-based algorithm and analysis for non-convex QCQP with one constraint*, Math. Program. **173** (2019), no. 1, 79–116.
https://doi.org/10.1007/s10107-017-1206-8.

[4] T.A. Almaadeed, S. Ansary Karbasy, M. Salahi, and A. Hamdi, *On indefinite quadratic optimization over the intersection of balls and linear constraints*, J. Optim. Theory Appl. **194** (2022), no. 1, 246–264.
https://doi.org/10.1007/s10957-022-02018-x.

[5] S. Ansary Karbasy and M. Salahi, *A hybrid algorithm for the two-trust-region subproblem*, Comput. Appl. Math. **38** (2019), no. 3, Article ID: 115.
https://doi.org/10.1007/s40314-019-0864-y.

[6] ———, *Quadratic optimization with two ball constraints*, Numer. Algebra Control Optim. **10** (2020), no. 2, p165.
https://doi.org/10.3934/naco.2019046.

[7] A. Beck and D. Pan, *A branch and bound algorithm for nonconvex quadratic optimization with ball and linear constraints*, J. Glob. Optim. **69** (2017), no. 2, 309–342.
https://doi.org/10.1007/s10898-017-0521-1.

[8] A. Ben-Tal, L.E. Ghaoui, and A. Nemirovski, *Robust optimization*, Princeton Series in Applied Mathematics, Princeton University Press, 2009.

[9] Samuel Burer and Yinyu Ye, *Exact semidefinite formulations for a class of (random and non-random) nonconvex quadratic programs*, Math. Program. **181** (2020), no. 1, 1–17.
https://doi.org/10.1007/s10107-019-01367-2.

[10] E.J. Candes, Y.C. Eldar, T. Strohmer, and V. Voroninski, *Phase retrieval via matrix completion*, SIAM review **57** (2015), no. 2, 225–251.
https://doi.org/10.1137/151005099.

[11] S. Ceria and R.A. Stubbs, *Incorporating estimation errors into portfolio selection: Robust portfolio construction*, pp. 270–294, Springer International Publishing, Cham, 2016.

[12] A.R. Conn, N.I.M. Gould, and P.L. Toint, *Trust Region Methods*, SIAM Philadelphia, 2000.

[13] Z. Deng, S.C. Fang, Q. Jin, and C. Lu, *Conic approximation to nonconvex quadratic programming with convex quadratic constraints*, J. Glob. Optim. **61** (2015), no. 3, 459–478.
https://doi.org/10.1007/s10898-014-0195-x.

[14] M. Grant and S. Boyd, *Cvx: Matlab software for disciplined convex programming*, Available from: https://cvxr.com/cvx, 2017.

[15] Z. Lu and Y. Zhang, *Sparse approximation via penalty decomposition methods*, SIAM J. Optim. **23** (2013), no. 4, 2448–2478.
https://doi.org/10.1137/100808071.

[16] H. Luo, X. Bai, G. Lim, and J. Peng, *New global algorithms for quadratic programming with a few negative eigenvalues based on alternative direction method and convex relaxation*, Math. Program. Comput. **11** (2019), no. 1, 119–171. https://doi.org/10.1007/s12532-018-0142-9.

[17] J.J. Moré and D.C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Stat. Comput. **4** (1983), no. 3, 553–572. https://doi.org/10.1137/0904038.

[18] Y. Nesterov, H. Wolkowicz, and Y. Ye, *Semidefinite programming relaxations of nonconvex quadratic optimization*, Handbook of Semidefinite Programming: Theory, Algorithms, and Applications (H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds.), Springer US, Boston, MA, 2000, pp. 361–419.

[19] E. Phan-huy Hao, *Quadratically constrained quadratic programming: Some applications and a method for solution*, Z. Operations Res. **26** (1982), no. 1, 105–119. https://doi.org/10.1007/BF01917102.

[20] T.K. Pong and H. Wolkowicz, *The generalized trust region subproblem*, Comput. Optim. Appl. **58** (2014), no. 2, 273–322. https://doi.org/10.1007/s10589-013-9635-7.

[21] N. Rujeerapaiboon, K. Schindler, D. Kuhn, and W. Wiesemann, *Size matters: Cardinality-constrained clustering and outlier detection via conic optimization*, SIAM J. Optim. **29** (2019), no. 2, 1211–1239. https://doi.org/10.1137/17M1150670.

[22] S. Sakaue, Y. Nakatsukasa, A. Takeda, and S. Iwata, *Solving generalized CDT problems via two-parameter eigenvalues*, SIAM J. Optim. **26** (2016), no. 3, 1669–1694. https://doi.org/10.1137/15100624X.

[23] M. Salahi, A. Taati, and H. Wolkowicz, *Local nonglobal minima for solving large-scale extended trust-region subproblems*, Comput. Optim. Appl. **66** (2017), no. 2, 223–244. https://doi.org/10.1007/s10589-016-9867-4.

[24] A. Wang, *On quadratically constrained quadratic programs and their semidefinite program relaxations*, Ph.D. thesis, Carnegie Mellon University, 2022.

[25] J. Zhou, D. Zhang, L. Wang, and Z. Xu, *A new SOCP relaxation of nonconvex quadratic programming problems with a few negative eigenvalues*, J. Comput. Appl. Math. **423** (2023), Article ID: 114944. https://doi.org/10.1016/j.cam.2022.114944.