Open Mathematics

Research Article

Hongwei Jiao* and Rongjiang Chen*

A parametric linearizing approach for quadratically inequality constrained quadratic programs

https://doi.org/10.1515/math-2018-0037
Received December 1, 2017; accepted February 20, 2018.

Abstract: In this paper we propose a new parametric linearizing approach for globally solving quadratically inequality constrained quadratic programs. By utilizing this approach, we can derive the parametric linear programs relaxation problem of the investigated problem. To accelerate the computational speed of the proposed algorithm, an interval deleting rule is used to reduce the investigated box. The proposed algorithm is convergent to the global optima of the initial problem by subsequently partitioning the initial box and solving a sequence of parametric linear programs relaxation problems. Finally, compared with some existing algorithms, numerical results show higher computational efficiency of the proposed algorithm.

Keywords: Quadratically inequality constrained quadratic programs, Global optimization, Parametric linearizing technique, Interval deleting rule

MSC: 90C20, 90C26, 65K05

1 Introduction

In this paper we consider the following quadratically inequality constrained quadratic programs:

$$(\mathrm{QICQP}): \begin{cases} \min\ H_0(z) = \sum\limits_{k=1}^{n} d_k^0 z_k + \sum\limits_{j=1}^{n}\sum\limits_{k=1}^{n} p_{jk}^0 z_j z_k \\ \text{s.t.}\ \ H_i(z) = \sum\limits_{k=1}^{n} d_k^i z_k + \sum\limits_{j=1}^{n}\sum\limits_{k=1}^{n} p_{jk}^i z_j z_k \le b_i,\ i = 1, \ldots, m, \\ \qquad z \in Z^0 = \{z \in R^n : l^0 \le z \le u^0\}, \end{cases}$$

where p_{jk}^i, d_k^i and b_i are all arbitrary real numbers; $l^0 = (l_1^0, \ldots, l_n^0)^T$, $u^0 = (u_1^0, \ldots, u_n^0)^T$. The investigated problem (QICQP) has a broad applications in investment portfolio, management decision, route optimization, engineering optimization, production planning and so on. In addition, the investigated problem (QICQP) usually owns multiple local optima which are not global optima, that is to say, in this kind of problems there are important theoretical and computational complexities. Therefore, it is very necessary to present an effective global optimization algorithm for solving the (QICQP).

In last decades, for the problem (QICQP) and its special cases many methods have been developed and described in the existent literature. For example, semi-definite relaxation method [1], reformulation-

***Corresponding Author: Hongwei Jiao:** Postdoctoral Research Base, Henan Institute of Science and Technology, Xinxiang 453003, China and Postdoctoral Research Station of Control Science and Engineering, Henan University of Science and Technology, Luoyang 471023, China, E-mail: jiaohongwei@126.com
***Corresponding Author: Rongjiang Chen:** Postdoctoral Research Base, Henan Institute of Science and Technology, Xinxiang 453003, China, E-mail: chenrj0324@126.com

convexification approach [2], branch-and-reduce approaches [3-7], approximation algorithms [8-10], simplicial branch-and-bound method [11], branch-and-cut method [12], rectangle branch-and-bound algorithms [13,14], robust solution approach [15], and so on. In addition, some algorithms for geometric programming [16-20] and multiplicative (or fractional) programming [21-25] also can be used to solve the (QICQP). Although these methods can be used to solve the investigated problem (QICQP) or its special forms, less work has been still done for globally solving the investigated quadratically inequality constrained quadratic programs.

In this paper, we will present a new branch-and-bound algorithm for globally solving the (QICQP). Firstly, we present a new parametric linearizing technique. By utilizing this method, we can convert the (QICQP) into a parametric linear programs relaxation problem, which can be used to compute the lower bounds of the optimal values of the initial problem (QICQP) and its subproblems. Secondly, based on the branch-and-bound framework, by successive partitioning of the initial box and by solving those derived parametric linear programs relaxation problems, a new branch-and-bound algorithm is designed for globally solving the (QICQP). Thirdly, to accelerate the computational efficiency of the proposed branch-and-bound algorithm, an interval deleting rule is used to reduce the investigated box. Fourthly, the proposed algorithm is convergent to the global optima of the initial problem (QICQP) by successively partitioning of the initial box and by solving those derived parametric linear programs relaxation problems. Finally, compared with some existent algorithms, numerical results demonstrate the computational efficiency of the proposed algorithm.

The remaining sections of this article are organized as follows. First of all, we present a new parametric linearizing technique for deriving the parametric linear programs relaxation problem of the (QICQP) in Section 2. Secondly, based on the branch-and-bound framework, by combing the derived parametric linear programs relaxation problem with the interval deleting rule, a branch-and-bound algorithm is established for globally solving the (QICQP) in Section 3. Thirdly, compared with the existent methods, some numerical examples in existent literatures are used to verify the computational efficiency of the proposed algorithm in Section 4. Finally, some concluding remarks are presented.

2 New parametric linearizing approach

In this section, we propose a new parametric linearizing approach for deriving the parametric linear programs relaxation problem of the (QICQP). The detailed parametric linearizing approach is presented as follows:

Assume that $Z = \{(z_1, z_2, \ldots, z_n)^T \in R^n : l_j \le z_j \le u_j, j = 1, \ldots, n\} \subseteq Z^0$, $\lambda = (\lambda_{jk})_{n\times n} \in R^{n\times n}$ is a symmetric matrix, and $\lambda_{jk} \in \{0, 1\}$. For convenience, for any $z \in Z$, for any $k \in \{1, 2, \ldots, n\}$, some expressions are introduced as follows:

$$
\begin{aligned}
&z_k(\lambda_{kk}) = l_k + \lambda_{kk}(u_k - l_k),\\
&z_k(1-\lambda_{kk}) = l_k + (1-\lambda_{kk})(u_k - l_k),\\
&h_{kk}(z) = z_k^2,\\
&\underline{h}_{kk}(z, Z, \lambda_{kk}) = [z_k(\lambda_{kk})]^2 + 2z_k(\lambda_{kk})[z_k - z_k(\lambda_{kk})],\\
&\overline{h}_{kk}(z, Z, \lambda_{kk}) = [z_k(\lambda_{kk})]^2 + 2z_k(1-\lambda_{kk})[z_k - z_k(\lambda_{kk})].
\end{aligned}
$$

Obviously, we have $z_k(0) = l_k$, $z_k(1) = u_k$.

Theorem 2.1. *For any $k \in \{1, 2, \ldots, n\}$, for any $z \in Z$, consider the functions $h_{kk}(z)$, $\underline{h}_{kk}(z, Z, \lambda_{kk})$ and $\overline{h}_{kk}(z, Z, \lambda_{kk})$, then, the following conclusions hold:*

$$\underline{h}_{kk}(z, Z, \lambda_{kk}) \le h_{kk}(z) \le \overline{h}_{kk}(z, Z, \lambda_{kk}); \tag{1}$$

$$\lim_{\|u-l\|\to 0}[h_{kk}(z) - \underline{h}_{kk}(z, Z, \lambda_{kk})] = 0 \tag{2}$$

and

$$\lim_{\|u-l\|\to 0}[\overline{h}_{kk}(z, Z, \lambda_{kk}) - h_{kk}(z)] = 0. \tag{3}$$

Proof. (i) By the mean value theorem, for any $z \in Z$, there exists a point $\xi_k = \alpha z_k + (1-\alpha) z_k(\lambda_{kk})$, where $\alpha \in [0,1]$, such that

$$z_k^2 = [z_k(\lambda_{kk})]^2 + 2\xi_k[z_k - z_k(\lambda_{kk})].$$

If $\lambda_{kk} = 0$, then we have

$$\xi_k \geq l_k = z_k(\lambda_{kk}) \text{ and } z_k - z_k(\lambda_{kk}) = z_k - l_k \geq 0.$$

If $\lambda_{kk} = 1$, then it follows that

$$\xi_k \leq u_k = z_k(\lambda_{kk}) \text{ and } z_k - z_k(\lambda_{kk}) = z_k - u_k \leq 0.$$

Thus, we can get that

$$\begin{aligned} h_{kk}(z) &= z_k^2 \\ &= [z_k(\lambda_{kk})]^2 + 2\xi_k[z_k - z_k(\lambda_{kk})] \\ &\geq [z_k(\lambda_{kk})]^2 + 2z_k(\lambda_{kk})[z_k - z_k(\lambda_{kk})] \\ &= \underline{h}_{kk}(z, Z, \lambda_{kk}). \end{aligned}$$

Similarly, if $\lambda_{kk} = 0$, then we have

$$\xi_k \leq u_k = z_k(1-\lambda_{kk}) \text{ and } z_k - z_k(\lambda_{kk}) = z_k - l_k \geq 0.$$

If $\lambda_{kk} = 1$, then it follows that

$$\xi_k \geq l_k = z_k(1-\lambda_{kk}) \text{ and } z_k - z_k(\lambda_{kk}) = z_k - u_k \leq 0.$$

Thus, we can get that

$$\begin{aligned} h_{kk}(z) &= z_k^2 \\ &= [z_k(\lambda_{kk})]^2 + 2\xi_k[z_k - z_k(\lambda_{kk})] \\ &\leq [z_k(\lambda_{kk})]^2 + 2z_k(1-\lambda_{kk})[z_k - z_k(\lambda_{kk})] \\ &= \overline{h}_{kk}(z, Z, \lambda_{kk}). \end{aligned}$$

Therefore, for any $z \in Z$, we have that

$$\underline{h}_{kk}(z, Z, \lambda_{kk}) \leq h_{kk}(z) \leq \overline{h}_{kk}(z, Z, \lambda_{kk}).$$

(ii) Since

$$\begin{aligned} h_{kk}(z) - \underline{h}_{kk}(z, Z, \lambda_{kk}) &= z_k^2 - \{[z_k(\lambda_{kk})]^2 + 2z_k(\lambda_{kk})[z_k - z_k(\lambda_{kk})]\} \\ &= (z_k - z_k(\lambda_{kk}))^2 \\ &\leq (u_k - l_k)^2, \end{aligned} \tag{4}$$

we have

$$\lim_{\|u-l\| \to 0} [h_{kk}(z) - \underline{h}_{kk}(z, Z, \lambda_{kk})] = 0.$$

Also since

$$\begin{aligned} \overline{h}_{kk}(z, Z, \lambda_{kk}) - h_{kk}(z) &= [z_k(\lambda_{kk})]^2 + 2z_k(1-\lambda_{kk})[z_k - z_k(\lambda_{kk})] - z_k^2 \\ &= (z_k(\lambda_{kk}) + z_k)(z_k(\lambda_{kk}) - z_k) \\ &\quad + 2z_k(1-\lambda_{kk})(z_k - z_k(\lambda_{kk})) \\ &= [z_k - z_k(\lambda_{kk})][2z_k(1-\lambda_{kk}) - z_k(\lambda_{kk}) - z_k] \\ &= [z_k - z_k(\lambda_{kk})][z_k(1-\lambda_{kk}) - z_k(\lambda_{kk})] \\ &\quad + [z_k - z_k(\lambda_{kk})][z_k(1-\lambda_{kk}) - z_k] \\ &\leq 2(u_k - l_k)^2. \end{aligned} \tag{5}$$

Therefore, it follows that

$$\lim_{\|u-l\| \to 0} [\overline{h}_{kk}(z, Z, \lambda_{kk}) - h_{kk}(z)] = 0.$$

The proof is completed. □

Without loss of generality, for any $z \in Z$, for any $j \in \{1, 2, \dots, n\}$, $k \in \{1, 2, \dots, n\}$, $j \neq k$, we define

$$\begin{aligned}
&z_j(\lambda_{jk}) = l_j + \lambda_{jk}(u_j - l_j),\\
&z_k(\lambda_{jk}) = l_k + \lambda_{jk}(u_k - l_k),\\
&z_j(1-\lambda_{jk}) = l_j + (1-\lambda_{jk})(u_j - l_j),\\
&z_k(1-\lambda_{jk}) = l_k + (1-\lambda_{jk})(u_k - l_k),\\
&(z_j - z_k)(\lambda_{jk}) = (l_j - u_k) + \lambda_{jk}(u_j - l_k - l_j + u_k),\\
&(z_j - z_k)(1-\lambda_{jk}) = (l_j - u_k) + (1-\lambda_{jk})(u_j - l_k - l_j + u_k).
\end{aligned}$$

Obviously, we have $(z_j - z_k)(0) = l_j - u_k$, $(z_j - z_k)(1) = u_j - l_k$.

In a similar way as in Theorem 2.1, we can get the following Theorem 2.2:

Theorem 2.2. *For each $j = 1, 2, \dots, n$, $k = 1, 2, \dots, n$, for any $z \in Z$, we have:*
(i) The following inequalities hold:

$$\begin{aligned}
&[z_j(\lambda_{jk})]^2 + 2z_j(\lambda_{jk})[z_j - z_j(\lambda_{jk})] \le z_j^2 \le [z_j(\lambda_{jk})]^2 + 2z_j(1-\lambda_{jk})[z_j - z_j(\lambda_{jk})],\\
&z_k(\lambda_{jk})]^2 + 2z_k(\lambda_{jk})[z_k - z_k(\lambda_{jk})] \le z_k^2 \le [z_k(\lambda_{jk})]^2 + 2z_k(1-\lambda_{jk})[z_k - z_k(\lambda_{jk})],\\
&(z_j - z_k)^2 \le [(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(1-\lambda_{jk})[z_j - z_k - (z_j - z_k)(\lambda_{jk})],\\
&(z_j - z_k)^2 \ge [(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(\lambda_{jk})[z_j - z_k - (z_j - z_k)(\lambda_{jk})].
\end{aligned}$$

(ii) The following limitations hold:

$$\lim_{\|u-l\|\to 0}[z_j^2 - \{[z_j(\lambda_{jk})]^2 + 2z_j(\lambda_{jk})[z_j - z_j(\lambda_{jk})]\}] = 0,$$

$$\lim_{\|u-l\|\to 0}[[z_j(\lambda_{jk})]^2 + 2z_j(1-\lambda_{jk})[z_j - z_j(\lambda_{jk})] - z_j^2] = 0,$$

$$\lim_{\|u-l\|\to 0}[z_k^2 - \{[z_k(\lambda_{jk})]^2 + 2z_k(\lambda_{jk})[z_k - z_k(\lambda_{jk})]\}] = 0,$$

$$\lim_{\|u-l\|\to 0}[[z_k(\lambda_{jk})]^2 + 2z_k(1-\lambda_{jk})[z_k - z_k(\lambda_{jk})] - z_k^2] = 0,$$

$$\lim_{\|u-l\|\to 0}[[(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(1-\lambda_{jk})[z_j - z_k - (z_j - z_k)(\lambda_{jk})] - (z_j - z_k)^2] = 0,$$

$$\lim_{\|u-l\|\to 0}[(z_j - z_k)^2 - \{[(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(\lambda_{jk})[(z_j - z_k) - (z_j - z_k)(\lambda_{jk})]\}] = 0.$$

Proof. (i) From the inequality (1), replacing λ_{kk} by λ_{jk}, and replacing z_k by z_j, we can get that

$$[z_j(\lambda_{jk})]^2 + 2z_j(\lambda_{jk})[z_j - z_j(\lambda_{jk})] \le z_j^2 \le [z_j(\lambda_{jk})]^2 + 2z_j(1-\lambda_{jk})[z_j - z_j(\lambda_{jk})].$$

From the inequality (1), replacing λ_{kk} by λ_{jk}, we can get that

$$[z_k(\lambda_{jk})]^2 + 2z_k(\lambda_{jk})[z_k - z_k(\lambda_{jk})] \le z_k^2 \le [z_k(\lambda_{jk})]^2 + 2z_k(1-\lambda_{jk})[z_k - z_k(\lambda_{jk})].$$

From (1), replacing λ_{kk} and z_k by λ_{jk} and $(z_j - z_k)$, respectively, we can get that

$$(z_j - z_k)^2 \le [(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(1-\lambda_{jk})[(z_j - z_k) - (z_j - z_k)(\lambda_{jk})],$$

$$(z_j - z_k)^2 \ge [(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(\lambda_{jk})[(z_j - z_k) - (z_j - z_k)(\lambda_{jk})].$$

(ii) From the limitations (2) and (3), replacing λ_{kk} and z_k by λ_{jk} and z_j, we have

$$\lim_{\|u-l\|\to 0}[z_j^2 - \{[z_j(\lambda_{jk})]^2 + 2z_j(\lambda_{jk})[z_j - z_j(\lambda_{jk})]\}] = 0$$

and

$$\lim_{\|u-l\|\to 0}[[z_j(\lambda_{jk})]^2 + 2z_j(1-\lambda_{jk})[z_j - z_j(\lambda_{jk})] - z_j^2] = 0.$$

From the limitations (2) and (3), replacing λ_{kk} by λ_{jk}, it follows that

$$\lim_{\|u-l\|\to 0}[z_k^2 - \{[z_k(\lambda_{jk})]^2 + 2z_k(\lambda_{jk})[z_k - z_k(\lambda_{jk})]\}] = 0$$

and

$$\lim_{\|u-l\|\to 0}[[z_k(\lambda_{jk})]^2 + 2z_k(1-\lambda_{jk})[z_k - z_k(\lambda_{jk})] - z_k^2] = 0.$$

By the limitations (2) and (3), replacing λ_{kk} and z_k by λ_{jk} and $(z_j - z_k)$, respectively, we can get that

$$\lim_{\|u-l\|\to 0}[[(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(1-\lambda_{jk})[z_j - z_k - (z_j - z_k)(\lambda_{jk})] - (z_j - z_k)^2] = 0$$

and

$$\lim_{\|u-l\|\to 0}[(z_j - z_k)^2 - \{[(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(\lambda_{jk})[z_j - z_k - (z_j - z_k)(\lambda_{jk})]\}] = 0.$$

The proof is completed. □

Without loss of generality, for any $z \in Z$, for any $j \in \{1, 2, \ldots, n\}$, $k \in \{1, 2, \ldots, n\}$, $j \neq k$, define

$$\begin{aligned}
h_{jk}(z) &= z_j z_k = \tfrac{z_j^2 + z_k^2 - (z_j - z_k)^2}{2},\\
\underline{h}_{jk}(z, Z, \lambda_{jk}) &= \tfrac{1}{2}\{[z_j(\lambda_{jk})]^2 + 2z_j(\lambda_{jk})[z_j - z_j(\lambda_{jk})]\\
&\quad + [z_k(\lambda_{jk})]^2 + 2z_k(\lambda_{jk})[z_k - z_k(\lambda_{jk})]\\
&\quad - \{[(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(1-\lambda_{jk})[z_j - z_k - (z_j - z_k)(\lambda_{jk})]\}\},\\
\overline{h}_{jk}(z, Z, \lambda_{jk}) &= \tfrac{1}{2}\{[z_j(\lambda_{jk})]^2 + 2z_j(1-\lambda_{jk})[z_j - z_j(\lambda_{jk})]\\
&\quad + [z_k(\lambda_{jk})]^2 + 2z_k(1-\lambda_{jk})[z_k - z_k(\lambda_{jk})]\\
&\quad - \{[(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(\lambda_{jk})[z_j - z_k - (z_j - z_k)(\lambda_{jk})]\}\}.
\end{aligned}$$

Theorem 2.3. *For each $k = 1, 2, \ldots, n$, consider the functions $\underline{h}_{jk}(z, Z, \lambda_{jk})$, $h_{jk}(z)$ and $\overline{h}_{jk}(z, Z, \lambda_{jk})$, then, for any $z \in Z$, we have the following conclusions:*

$$\underline{h}_{jk}(z, Z, \lambda_{jk}) \le h_{jk}(z) \le \overline{h}_{jk}(z, Z, \lambda_{jk}), \tag{6}$$

$$\lim_{\|u-l\|\to 0}[h_{jk}(z) - \underline{h}_{jk}(z, Z, \lambda_{jk})] = 0 \tag{7}$$

and

$$\lim_{\|u-l\|\to 0}[\overline{h}_{jk}(z, Z, \lambda_{jk}) - h_{jk}(z)] = 0. \tag{8}$$

Proof. First of all, from the conclusions (i) of Theorem 2.2, it follows that

$$\begin{aligned}
h_{jk}(z) &= z_j z_k = \tfrac{z_j^2 + z_k^2 - (z_j - z_k)^2}{2}\\
&\ge \tfrac{1}{2}\{[z_j(\lambda_{jk})]^2 + 2z_j(\lambda_{jk})[z_j - z_j(\lambda_{jk})]\\
&\quad + [z_k(\lambda_{jk})]^2 + 2z_k(\lambda_{jk})[z_k - z_k(\lambda_{jk})]\\
&\quad - \{[(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(1-\lambda_{jk})[z_j - z_k - (z_j - z_k)(\lambda_{jk})]\}\}\\
&= \underline{h}_{jk}(z, Z, \lambda_{jk})
\end{aligned}$$

and

$$\begin{aligned}
h_{jk}(z) &= z_j z_k = \tfrac{z_j^2 + z_k^2 - (z_j - z_k)^2}{2}\\
&\le \tfrac{1}{2}\{[z_j(\lambda_{jk})]^2 + 2z_j(1-\lambda_{jk})[z_j - z_j(\lambda_{jk})]\\
&\quad + [z_k(\lambda_{jk})]^2 + 2z_k(1-\lambda_{jk})[z_k - z_k(\lambda_{jk})]\\
&\quad - \{[(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(\lambda_{jk})[z_j - z_k - (z_j - z_k)(\lambda_{jk})]\}\}\\
&= \overline{h}_{jk}(z, Z, \lambda_{jk}).
\end{aligned}$$

Secondly, from the inequalities (4) and (5), we have

$$\begin{aligned}
h_{jk}(z) - \underline{h}_{jk}(z, Z, \lambda_{jk}) &= z_j z_k - \underline{h}_{jk}(z, Z, \lambda_{jk}) \\
&= \tfrac{z_j^2 + z_k^2 - (z_j - z_k)^2}{2} - \tfrac{1}{2}\{[z_j(\lambda_{jk})]^2 + 2z_j(\lambda_{jk})[z_j - z_j(\lambda_{jk})] \\
&\quad + [z_k(\lambda_{jk})]^2 + 2z_k(\lambda_{jk})[z_k - z_k(\lambda_{jk})] \\
&\quad - \{[(z_j - z_k)(\lambda_{jk})]^2 + 2(z_j - z_k)(1 - \lambda_{jk})[z_j - z_k \\
&\quad - (z_j - z_k)(\lambda_{jk})]\}\}, \\
&= \tfrac{1}{2}[z_j^2 - \{[z_j(\lambda_{jk})]^2 + 2z_j(\lambda_{jk})[z_j - z_j(\lambda_{jk})]\}] \\
&\quad + \tfrac{1}{2}[z_k^2 - \{[z_k(\lambda_{jk})]^2 + 2z_k(\lambda_{jk})[z_k - z_k(\lambda_{jk})]\}] \\
&\quad + \tfrac{1}{2}\{\{[(z_j - z_k)(\lambda_{jk})]^2 \\
&\quad + 2(z_j - z_k)(1 - \lambda_{jk})[z_j - z_k - (z_j - z_k)(\lambda_{jk})]\} \\
&\quad - (z_j - z_k)^2\} \\
&\le \tfrac{1}{2}(u_j - l_j)^2 + \tfrac{1}{2}(u_k - l_k)^2 + (u_k + u_j - l_j - l_k)^2.
\end{aligned}$$

Thus, we can get that $\lim_{\|u-l\|\to 0}[h_{jk}(z) - \underline{h}_{jk}(z, Z, \lambda_{jk})] = 0$.

Also from the inequalities (4) and (5), we get that

$$\begin{aligned}
\overline{h}_{jk}(z, Z, \lambda_{jk}) - h_{jk}(z) &= \overline{h}_{jk}(z, Z, \lambda_{jk}) - z_j z_k \\
&= \tfrac{1}{2}\{[z_j(\lambda_{jk})]^2 + 2z_j(1 - \lambda_{jk})[z_j - z_j(\lambda_{jk})] \\
&\quad + [z_k(\lambda_{jk})]^2 + 2z_k(1 - \lambda_{jk})[z_k - z_k(\lambda_{jk})] \\
&\quad - \{[(z_j - z_k)(\lambda_{jk})]^2 \\
&\quad + 2(z_j - z_k)(\lambda_{jk})[z_j - z_k - (z_j - z_k)(\lambda_{jk})]\}\} \\
&\quad - \tfrac{z_j^2 + z_k^2 - (z_j - z_k)^2}{2} \\
&= \tfrac{1}{2}[[z_j(\lambda_{jk})]^2 + 2z_j(1 - \lambda_{jk})[z_j - z_j(\lambda_{jk})] - z_j^2] \\
&\quad + \tfrac{1}{2}[[z_k(\lambda_{jk})]^2 + 2z_k(1 - \lambda_{jk})[z_k - z_k(\lambda_{jk})] - z_k^2] \\
&\quad + \tfrac{1}{2}\{(z_j - z_k)^2 - \{[(z_j - z_k)(\lambda_{jk})]^2 \\
&\quad + 2(z_j - z_k)(\lambda_{jk})[z_j - z_k - (z_j - z_k)(\lambda_{jk})]\}\} \\
&= (z_j - z_j(\lambda_{jk}))(z_k - z_k(\lambda_{jk})) \\
&\le (u_j - l_j)^2 + (u_k - l_k)^2 + (u_k + u_j - l_k - l_j)^2.
\end{aligned}$$

Thus, it follows that $\lim_{\|u-l\|\to 0}[\overline{h}_{jk}(z, Z, \lambda_{jk}) - h_{jk}(z)] = 0$. □

Without loss of generality, for any $Z = [l, u] \subseteq Z^0$, for any parameter matrix $\lambda = (\lambda_{jk})_{n\times n}$, for any $z \in Z$ and $i \in \{0, 1, \ldots, m\}$, we let

$$\underline{f}_{kk}^i(z, Z, \lambda_{kk}) = \begin{cases} p_{kk}^i \underline{h}_{kk}(z, Z, \lambda_{kk}), \text{ if } p_{kk}^i > 0, \\ p_{kk}^i \overline{h}_{kk}(z, Z, \lambda_{kk}), \text{ if } p_{kk}^i < 0, \end{cases}$$

$$\overline{f}_{kk}^i(z, Z, \lambda_{kk}) = \begin{cases} p_{kk}^i \overline{h}_{kk}(z, Z, \lambda_{kk}), \text{ if } p_{kk}^i > 0, \\ p_{kk}^i \underline{h}_{kk}(z, Z, \lambda_{kk}), \text{ if } p_{kk}^i < 0, \end{cases}$$

$$\underline{f}_{jk}^i(z, Z, \lambda_{jk}) = \begin{cases} p_{jk}^i \underline{h}_{jk}(z, Z, \lambda_{jk}), \text{ if } p_{jk}^i > 0,\ j \ne k, \\ p_{jk}^i \overline{h}_{jk}(z, Z, \lambda_{jk}), \text{ if } p_{jk}^i < 0,\ j \ne k, \end{cases}$$

$$\overline{f}_{jk}^i(z, Z, \lambda_{jk}) = \begin{cases} p_{jk}^i \overline{h}_{jk}(z, Z, \lambda_{jk}), \text{ if } p_{jk}^i > 0,\ j \ne k, \\ p_{jk}^i \underline{h}_{jk}(z, Z, \lambda_{jk}), \text{ if } p_{jk}^i < 0,\ j \ne k. \end{cases}$$

$$H_i^L(z, Z, \lambda) = \sum_{k=1}^n (d_k^i z_k + \underline{f}_{kk}^i(z, Z, \lambda_{kk})) + \sum_{j=1}^n \sum_{k=1, k\ne j}^n \underline{f}_{jk}^i(z, Z, \lambda_{jk}).$$

$$H_i^U(z, Z, \lambda) = \sum_{k=1}^n (d_k^i z_k + \overline{f}_{kk}^i(z, Z, \lambda_{kk})) + \sum_{j=1}^n \sum_{k=1, k\ne j}^n \overline{f}_{jk}^i(z, Z, \lambda_{jk}).$$

Theorem 2.4. *For $\forall\, z \in Z = [l, u] \subseteq Z^0$, for any given parameter matrix $\lambda = (\lambda_{jk})_{n\times n}$, for each $i = 0, 1, \ldots, m$, we have the following conclusions:*

$$H_i^L(z, Z, \lambda) \le H_i(z) \le H_i^U(z, Z, \lambda),$$

$$\lim_{\|u-l\|\to 0}[H_i(z) - H_i^L(z, Z, \lambda)] = 0$$

and

$$\lim_{\|u-l\|\to 0}[H_i^U(z, Z, \lambda) - H_i(z)] = 0.$$

Proof. First of all, from the inequalities (1) and (6), for any $j, k \in \{1, \dots, n\}$, we have

$$\underline{f}_{kk}^i(z, Z, \lambda_{kk}) \le p_{kk}^i z_k^2 \le \overline{f}_{kk}^i(z, Z, \lambda_{kk}), \tag{9}$$

$$\underline{f}_{jk}^i(z, Z, \lambda_{jk}) \le p_{jk}^i z_j z_k \le \overline{f}_{jk}^i(z, Z, \lambda_{jk}). \tag{10}$$

By the above inequalities (9) and (10), for any $z \in Z \subseteq Z^0$, we can get that

$$\begin{aligned} H_i^L(z, Z, \lambda) &= \sum_{k=1}^n (d_k^i z_k + \underline{f}_{kk}^i(z, Z, \lambda_{kk})) + \sum_{j=1}^n \sum_{k=1,k\neq j}^n \underline{f}_{jk}^i(z, Z, \lambda_{jk}) \\ &\le \sum_{k=1}^n d_k^i z_k + \sum_{k=1}^n p_{kk}^i z_k^2 + \sum_{j=1}^n \sum_{k=1,k\neq j}^n p_{jk}^i z_j z_k = H_i(z) \\ &\le \sum_{k=1}^n (d_k^i z_k + \overline{f}_{kk}^i(z, Z, \lambda_{kk})) + \sum_{j=1}^n \sum_{k=1,k\neq j}^n \overline{f}_{jk}^i(z, Z, \lambda_{jk}) \\ &= H_i^U(z, Z, \lambda). \end{aligned}$$

Therefore, we have $H_i^L(z, Z, \lambda) \le H_i(z) \le H_i^U(z, Z, \lambda)$.

Secondly,

$$\begin{aligned} H_i(z) - H_i^L(z, Z, \lambda) &= \sum_{k=1}^n d_k^i z_k + \sum_{k=1}^n p_{kk}^i z_k^2 + \sum_{j=1}^n \sum_{k=1,k\neq j}^n p_{jk}^i z_j z_k \\ &\quad -[\sum_{k=1}^n d_k^i z_k + \sum_{k=1}^n \underline{f}_{kk}^i(z, Z, \lambda_{kk}) + \sum_{j=1}^n \sum_{k=1,k\neq j}^n \underline{f}_{jk}^i(z, Z, \lambda_{jk})] \\ &= \sum_{k=1}^n [p_{kk}^i z_k^2 - \underline{f}_{kk}^i(z, Z, \lambda_{kk})] \\ &\quad + \sum_{j=1}^n \sum_{k=1,k\neq j}^n [p_{jk}^i z_j z_k - \underline{f}_{jk}^i(z, Z, \lambda_{jk})] \\ &= \sum_{k=1,p_{kk}^i>0}^n p_{kk}^i [h_{kk}(z) - \underline{h}_{kk}(z, Z, \lambda_{kk})] \\ &\quad + \sum_{k=1,p_{kk}^i<0}^n p_{kk}^i [h_{kk}(z) - \overline{h}_{kk}(z, Z, \lambda_{kk})] \\ &\quad + \sum_{j=1}^n \sum_{k=1,k\neq j,p_{jk}^i>0}^n p_{jk}^i [h_{jk}(z) - \underline{h}_{jk}(z, Z, \lambda_{jk})] \\ &\quad + \sum_{j=1}^n \sum_{k=1,k\neq j,p_{jk}^i<0}^n p_{jk}^i [h_{jk}(z) - \overline{h}_{jk}(z, Z, \lambda_{jk})]. \end{aligned}$$

From the limitations (2),(3),(7) and (8), $\lim_{\|u-l\|\to 0}[h_{kk}(z) - \underline{h}_{kk}(z, Z, \lambda_{kk})] = 0$, $\lim_{\|u-l\|\to 0}[\overline{h}_{kk}(z, Z, \lambda_{kk}) - h_{kk}(z)] = 0$, $\lim_{\|u-l\|\to 0}[h_{jk}(z) - \underline{h}_{jk}(z, Z, \lambda_{jk})] = 0$ and $\lim_{\|u-l\|\to 0}[\overline{h}_{jk}(z, Z, \lambda_{jk}) - h_{jk}(z)] = 0$.

Therefore, it follows that

$$\lim_{\|u-l\|\to 0}[H_i(z) - H_i^L(z, Z, \lambda)] = 0.$$

Similarly, we can prove that

$$\lim_{\|u-l\|\to 0}[H_i^U(z, Z, \lambda) - H_i(z)] = 0.$$

The proof is completed. □

By Theorem 2.4, we can construct the parametric linear programs relaxation problem (PLPRP) of the (QICQP) over Z as follows:

$$(\textbf{PLPRP}): \begin{cases} \min\ H_0^L(z, Z, \lambda), \\ \text{s.t.}\ \ H_i^L(z, Z, \lambda) \le b_i,\ i = 1, \dots, m, \\ \qquad z \in Z = \{z : l \le z \le u\}. \end{cases}$$

where

$$H_i^L(z, Z, \lambda) = \sum_{k=1}^{n}(d_k^i z_k + \underline{f}_{kk}^i(z, Z, \lambda_{kk})) + \sum_{j=1}^{n}\sum_{k=1,k\neq j}^{n}\underline{f}_{jk}^i(z, Z, \lambda_{jk}).$$

Based on the former parametric linearizing technique, each feasible solution of the (QICQP) must be also feasible to the (PLPRP) in the sub-region Z; and the minimum value of the (PLPRP) must be less than or equal to that of the (QICQP) in the sub-region Z. Hence, the (PLPRP) offers a reliable lower bound for the minimum value of the (QICQP) in the sub-region Z. In addition, Theorem 2.4 ensures that the optimal solution of the (PLPRP) will sufficiently approximate the optimal solution of the (QICQP) as $\|u - l\| \to 0$, and this guarantees the global convergence of the proposed algorithm.

3 Branch-and-bound global optimization algorithm

In this section, a new branch-and-bound global optimization algorithm is proposed for solving the (QICQP). In this algorithm, there are the following several important techniques: branching, bounding the lower bound, bounding upper bound and interval deleting.

Branching: The branching step will generate a more refined box partition. Here we choose a typical box-bisection method, which is sufficient to ensure the global convergence of the proposed branch-and-bound method. For any selected box $Z' = [l', u'] \subseteq Z^0$. Set $\eta \in \arg\max\{u_i' - l_i' : i = 1, 2, \ldots, n\}$, by partitioning $[\underline{z}_\eta', \overline{z}_\eta']$ into $[\underline{z}_\eta', (\underline{z}_\eta' + \overline{z}_\eta')/2]$ and $[(\underline{z}_\eta' + \overline{z}_\eta')/2, \overline{z}_\eta']$, we can subdivide Z' into two new sub-boxes Z'^1 and Z'^2.

Bounding the lower bound: For each sub-box $Z \subseteq Z^0$, which has not been deleted, the bounding the lower bound step needs to solve the parametric linear programs relaxation problem over each sub-box, and denote by $LB_s = \min\{LB(Z)|Z \in \Omega_s\}$, where Ω_s denotes the set of sub-box which has not been deleted after s iteration.

Bounding the upper bound: The bounding upper bound step needs to judge the feasibility of the midpoint of each investigated sub-box Z and the optimal solution of the (PLPRP) over the investigated sub-box Z, where $Z \in \Omega_s$. In addition, we need to calculate the objective function values of each known feasible solutions for the (QICQP), and denote by $UB_s = \min\{H_0(z) : z \in \Theta\}$ the best upper bound, where Θ is the known feasible point set.

Interval deleting: To improve the convergent speed of the branch-and-bound algorithm, an interval deleting rule is introduced as follows. For convenience, for any $z \in Z, i \in \{0, 1, \ldots, m\}, q \in \{1, \ldots, n\}$, and denote by HUB the current upper bound of the (QICQP), we let

$$H_i^L(z, Z, \lambda) = \sum_{j=1}^{n}\alpha_{ij}(\lambda)z_j + \beta_i(\lambda), \quad RLB_i(\lambda) = \sum_{j=1}^{n}\min\{\alpha_{ij}(\lambda)l_j, \alpha_{ij}(\lambda)u_j\} + \beta_i(\lambda).$$

Theorem 3.1. *For any investigated sub-box $Z = (Z_j)_{1\times n} \subseteq Z^0$, we have the following conclusions:*
(i) If $RLB_0(\lambda) > HUB$, then the whole investigated sub-box Z should be deleted.
(ii) If $RLB_0(\lambda) \leq HUB$, then: for any $q \in \{1, 2, \ldots, n\}$, if $\alpha_{0q}(\lambda) > 0$, the interval Z_q should be replaced by $[l_q, \frac{HUB - RLB_0(\lambda) + \min\{\alpha_{0q}(\lambda)l_q, \alpha_{0q}(\lambda)u_q\}}{\alpha_{0q}(\lambda)}] \cap Z_q$; if $\alpha_{0q}(\lambda) < 0$, the interval Z_q should be replaced by $[\frac{HUB - RLB_0(\lambda) + \min\{\alpha_{0q}(\lambda)l_q, \alpha_{0q}(\lambda)u_q\}}{\alpha_{0q}(\lambda)}, u_q] \cap Z_q$.
(iii) If $RLB_i(\lambda) > b_i$ for some $i \in \{1, \ldots, m\}$, then the whole investigated sub-box Z should be deleted.
(iv) If $RLB_i(\lambda) \leq b_i$ for each $i \in \{1, \ldots, m\}$, then, for any $q \in \{1, 2, \ldots, n\}$, if $\alpha_{iq}(\lambda) > 0$, the interval Z_q can be replaced by $[l_q, \frac{b_i - RLB_i(\lambda) + \min\{\alpha_{iq}(\lambda)l_q, \alpha_{iq}(\lambda)u_q\}}{\alpha_{iq}(\lambda)}] \cap Z_q$; if $\alpha_{iq}(\lambda) < 0$, the interval Z_q can be replaced by $[\frac{b_i - RLB_i(\lambda) + \min\{\alpha_{iq}(\lambda)l_q, \alpha_{iq}(\lambda)u_q\}}{\alpha_{iq}(\lambda)}, u_q] \cap Z_q$.

Proof. In a similar way as in the proof of Theorem 3 in [14], we may draw the conclusions for Theorem 3.1, so here it is omitted. □

From Theorem 3.1, we can construct an interval deleting step to compress the investigated box for improving the convergent speed of the proposed branch-and-bound algorithm.

3.1 New branch-and-bound algorithm

For any sub-box $Z^s \subseteq Z^0$, we denote by $LB(Z^s)$ the optimal value of the (PLPRP) over the sub-box Z^s, and denote by $z^s = z(Z^s)$ the optimal solution of the (PLPRP) over the sub-box Z^s. Based on the branch-and-bound framework, combining the former branching step, bounding the lower bound step, bounding upper bound step and interval deleting step together, a new branch-and-bound algorithm is designed as follows.

Branch-and-Bound Algorithm Steps

Initializing step. Given the initial convergent error ϵ, the initial randomly generated parameter matrix λ.

Solve the (PLPRP) over the initial box Z^0 to obtain its optimal solution z^0 and optimal value $LB(Z^0)$, denote by the initial lower bound $LB_0 = LB(Z^0)$. If z^0 is a feasible solution of the (QICQP), we denote by the initial upper bound $UB_0 = H_0(z^0)$. Otherwise, we denote by the initial upper bound $UB_0 = +\infty$.

If $UB_0 - LB_0 \leq \epsilon$, the proposed algorithm terminates, z^0 is a global ϵ-optimal solution of the initial problem (QICQP). Otherwise, set $\Omega_0 = \{Z^0\}$, $\Lambda = \varnothing$, $s = 1$.

Branching step. Let $UB_s = UB_{s-1}$. Partition the investigated sub-box Z^{s-1} into two sub-boxes $Z^{s,1}$, $Z^{s,2}$ by the selected branching rule, and denote by $\Lambda = \Lambda \cup \{Z^{s-1}\}$ the set of the deleting sub-boxes.

Interval deleting step. For each investigated sub-box $Z^{s,t}$, $t = 1, 2$, use the former interval deleting rule to compress the investigated sub-box, still denote by $Z^{s,t}$ the remaining sub-box.

Bounding the lower bound step. For each remaining sub-box $Z^{s,t}$, where $t = 1, 2$, solve the (PLPRP) over $Z^{s,t}$ to obtain its optimal solution $z^{s,t}$ and optimal value $LB(Z^{s,t})$, and let $\Omega_s = \{Z | Z \in \Omega_{s-1} \cup \{Z^{s,1}, Z^{s,2}\}, Z \notin \Lambda\}$ and $LB_s = \min\{LB(Z) | Z \in \Omega_s\}$.

Bounding the upper bound step. For each sub-box $Z^{s,t}$, if its midpoint z^{mid} is the feasible point of the initial problem (QICQP), let $\Theta := \Theta \cup \{z^{mid}\}$, denote by the new upper bound $UB_s = \min_{z \in \Theta} H_0(z)$; if the optimal solution $z^{s,t}$ of the (PLPRP) is the feasible point of the initial problem (QICQP), denote by the new upper bound $UB_s = \min\{UB_s, H_0(z^{s,t})\}$, and denote by z^s the best existent feasible point such that $UB_s = H_0(z^s)$.

Terminating judgement step. If $UB_s - LB_s \leq \epsilon$, the proposed algorithm terminates, z^s is a global ϵ-optimal solution of the initial problem (QICQP). Otherwise, denote by $s = s + 1$, and go to the Branching step.

3.2 Global convergence of the proposed algorithm

Without loss of generality, we assume that v is the global optimal value of the initial problem (QICQP). If the proposed algorithm terminates after s finite iterations, where s is a finite number such that $s \geq 0$, then it follows that

$$UB_s \leq LB_s + \epsilon.$$

From the bounding the upper bound step of the proposed algorithm, we know that there must exist a feasible point z^s of the initial problem (QICQP) such that

$$v \leq UB_s = H_0(z^s).$$

By the branch-and-bound structure of the proposed algorithm, we have

$$LB_k \leq v.$$

Combining the above several inequalities together, it follows that

$$v \leq UB_s = H_0(z^s) \leq LB_s + \epsilon \leq v + \epsilon.$$

Therefore, z^s is an ϵ-global optimal solution of the initial problem (QICQP).

If the proposed algorithm does not terminate after finite iterations, for this case, the detailed convergent conclusions are given as follows.

Theorem 3.2. *If the proposed algorithm does not terminate after finite iterations, then it will generate an infinite partitioning sequence $\{Z^s\}$ of the initial box Z^0, and any accumulation point of the sequence $\{Z^s\}$ will be a global optimum solution of the initial problem (QICQP).*

Proof. First of all, in the proposed algorithm the selected branching method is the bisection of box, so that the branching process is exhaustive, that is to say, the branching step will ensure that the intervals of all variables tend to 0, i.e., $\|u - l\| \to 0$.

Secondly, from Theorem 2.4, the optimal solution of the (PLPRP) will sufficiently approximate the optimal solution of the (QICQP) as $\|u - l\| \to 0$, and this ensures that the limitation $\lim_{s\to\infty}(UB_s - LB_s) = 0$ holds. So that the bounding operation is consistent.

Thirdly, in the proposed algorithm the subdivided box which attains the actual lower bound is selected for further partition at the later immediate iteration, so that the used selecting operation is bound improving.

From [26, Theorem IV.3], the sufficient condition of global convergence of the branch-and-bound algorithm is that the branching method is exhaustive, the bounding method is consistent and the selecting method is improvement, therefore, the proposed algorithm is convergent to the global optimal solution of the initial (QICQP). □

4 Numerical experiments

Given the convergent error $\epsilon = 10^{-6}$ and the parameter matrix $\lambda = (\lambda_{jk})_{n\times n} \in R^{n\times n}$, where $\lambda_{jk} \in \{0, 1\}$, compared with the existing methods, several numerical examples in existing literature are tested on microcomputer, the procedure is coded in C++ software, the parametric linear programs relaxation problems are solved by the simplex method. These examples and their numerical results are listed as follows. In the following Tables 1 and 2, the number of iteration and running time in seconds for the algorithm are represented by “Iteration" and “Time(s)", respectively.

Example 4.1 ([16])**.**

$$\begin{cases} \min\ H_0(z) = z_1 \\ \text{s.t. } H_1(z) = \frac{1}{4}z_1 + \frac{1}{2}z_2 - \frac{1}{16}z_1^2 - \frac{1}{16}z_2^2 \le 1, \\ \quad H_2(z) = \frac{1}{14}z_1^2 + \frac{1}{14}z_2^2 - \frac{3}{7}z_1 - \frac{3}{7}z_2 \le -1, \\ \quad 1 \le z_1 \le 5.5,\ 1 \le z_2 \le 5.5. \end{cases}$$

Example 4.2 ([16])**.**

$$\begin{cases} \min\ H_0(z) = z_1z_2 - 2z_1 + z_2 + 1 \\ \text{s.t. } H_1(z) = 8z_2^2 - 6z_1 - 16z_2 \le -11, \\ \quad H_2(z) = -z_2^2 + 3z_1 + 2z_2 \le 7, \\ \quad 1 \le z_1 \le 2.5,\ 1 \le z_2 \le 2.225. \end{cases}$$

Example 4.3 ([4,5,17])**.**

$$\begin{cases} \min\ H_0(z) = z_1^2 + z_2^2 \\ \text{s.t. } H_1(z) = 0.3z_1z_2 \ge 1, \\ \quad 2 \le z_1 \le 5,\ 1 \le z_2 \le 3. \end{cases}$$

Example 4.4 ([5,14,17,18]).

$$\begin{cases} \min\ H_0(z) = z_1 \\ \text{s.t. } H_1(z) = 4z_2 - 4z_1^2 \le 1, \\ \quad H_2(z) = -z_1 - z_2 \le -1, \\ \quad 0.01 \le z_1, z_2 \le 15. \end{cases}$$

Example 4.5 ([4,6,14]).

$$\begin{cases} \min\ H_0(z) = 6z_1^2 + 4z_2^2 + 5z_1z_2 \\ \text{s.t. } H_1(z) = -6z_1z_2 \le -48, \\ \quad 0 \le z_1, z_2 \le 10. \end{cases}$$

Example 4.6 ([19]).

$$\begin{cases} \min\ H_0(z) = -z_1 + z_1z_2^{0.5} - z_2 \\ \text{s.t. } H_1(z) = -6z_1 + 8z_2 \le 3, \\ \quad H_2(z) = 3z_1 - z_2 \le 3, \\ \quad 1 \le z_1, z_2 \le 1.5. \end{cases}$$

Example 4.7 ([14,20]).

$$\begin{cases} \min\ H_0(z) = -4z_2 + (z_1 - 1)^2 + z_2^2 - 10z_3^2 \\ \text{s.t. } H_1(z) = z_1^2 + z_2^2 + z_3^2 \le 2, \\ \quad H_2(z) = (z_1 - 2)^2 + z_2^2 + z_3^2 \le 2, \\ \quad 2 - \sqrt{2} \le z_1 \le \sqrt{2},\ 0 \le z_2, z_3 \le \sqrt{2}. \end{cases}$$

Table 1. Numerical comparisons for Examples 4.1-4.7

Example	Refs.	Optimal value	Optimal solution	Iteration	Time(s)
1	our	1.177124344	(1.177124344, 2.177124344)	22	0.0103
	[16]	1.177124327	(1.177124327, 2.177124353)	434	1.0000
2	our	-0.999999106	(2.000000, 1.000000)	21	0.0079
	[16]	-1.0	(2.000000, 1.000000)	24	0.0129
3	our	6.778046494	(2.000000000, 1.666747279)	12	0.0039
	[4]	6.777778340	(2.000000000, 1.666666667)	30	0.0068
	[5]	6.777782016	(2.000000000, 1.666666667)	40	0.0320
	[17]	6.7780	(2.00003, 1.66665)	44	0.1800
4	our	0.500000000	(0.500000000, 0.500000000)	25	0.0070
	[5]	0.500004627	(0.5 0.5)	34	0.0560
	[14]	0.500000442	(0.500000000, 0.500000000)	37	0.0193
	[17]	0.5	(0.5, 0.5)	91	0.8500
	[18]	0.5	(0.5, 0.5)	96	1.0000
5	our	118.392375925	(2.560178568, 3.125000000)	46	0.0294
	[4]	118.383672050	(2.555409888, 3.130613160)	49	0.0744
	[6]	118.383756475	(2.5557793695, 3.1301646393)	210	0.7800
	[14]	118.383671904	(2.555745855, 3.130201688)	59	0.0385
6	our	-1.162882315	(1.499977112, 1.5)	37	0.0756
	[19]	-1.16288	(1.5, 1.5)	84	0.1257
7	our	-11.363636364	(1.0, 0.181815071, 0.983332741)	98	0.1672
	[14]	-11.363636364	(1.0, 0.181818470, 0.983332113)	420	0.2845
	[20]	-10.35	(0.998712,0.196213,0.979216)	1648	0.3438

Example 4.8 ([3,14])**.**

$$\begin{cases} \max\ H_0(z) = \sum\limits_{i=1}^{n} z_i^2 \\ \text{s.t.}\ \ H_j(z) = \sum\limits_{i=1}^{j} z_i \le j,\ j = 1, 2, \ldots, n, \\ \quad z_i \ge 0,\ \ i = 1, 2, \ldots, n. \end{cases}$$

Table 2. Numerical results for Example 4.8

Refs.	Dimension n	Optimal value	Iteration	Time(s)
This paper	5	25.0	11	0.01632
	10	100.0	30	0.22646
	20	400.0	86	4.35751
	30	900.0	204	31.2676
	40	1600.0	300	80.161
[3]	5	25.0	141	10.11
	10	100.0	283	21.86
	20	400.0	651	47.00
	30	900.0	965	106.33
[14]	5	25.0	12	0.01818
	10	100.0	32	0.30216
	20	400.0	88	6.01095
	30	900.0	206	44.4965
	40	1600.0	302	98.122

Compared with the existing algorithms, the numerical results for examples 1-8 show that the proposed algorithm can be used to globally solve the quadratically inequality constrained quadratic programs with higher computational efficiency.

5 Concluding remarks

In this paper, we propose a new branch-and-bound algorithm for globally solving the quadratically inequality constrained quadratic programs. In this algorithm, we present a new parametric linearizing technique, which can be used to derive the parametric linear programs relaxation problem of the investigated problem (QICQP). To accelerate the computational speed of the proposed branch-and-bound algorithm, an interval deleting rule is used to reduce the investigated box. By subsequently partitioning the initial box and solving a sequence of parametric linear programs relaxation problems, the proposed algorithm is convergent to the global optima of the initial problem (QICQP). Finally, compared with some existing algorithms, numerical results show higher computational efficiency of the proposed algorithm.

Acknowledgement: This paper is supported by the National Natural Science Foundation of China (11671122, 11471102), the China Postdoctoral Science Foundation (2017M622340), the Basic and Frontier Technology Research Program of Henan Province (152300410097), the Cultivation Plan of Young Key Teachers in Colleges and Universities of Henan Province (2016GGJS-107), the Higher School Key Scientific Research Projects of Henan Province (18A110019,17A110021), the Major Scientific Research Projects of Henan Institute of Science and Technology (2015ZD07), the High-level Scientific Research Personnel Project for Henan Institute of Science and Technology (2015037), Henan Institute of Science and Technology Postdoctoral Science Foundation.

References

[1] Burer S., Vandenbussche D., A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. Math. Program., 2008, 113, 259-282.

[2] Sherali H.D., Tuncbilek C.H., A reformulation-convexification approach for solving nonconvex quadratic programming problems. J. Glob. Optim., 1995, 7, 1-31.

[3] Gao Y., Xue H., Shen P., A new box branch-and-reduce approach for solving nonconvex quadratic programming problems. Appl. Math. Comput., 2005, 168, 1409-1418.

[4] Jiao H., Liu S., Lu N., A parametric linear relaxation algorithm for globally solving nonconvex quadratic programming, Appl. Math. Comput., 2015, 250, 973-985.

[5] Shen P., Liu L., A global optimization approach for quadratic programs with nonconvex quadratic constraints. Chin. J. Eng. Math., 2008, 25, 923-926.

[6] Gao Y., Shang Y., Zhang L., A branch and reduce approach for solving nonconvex quadratic programming problems with quadratic constraints. OR Transactions, 2005, 9(2), 9-20.

[7] Jiao H., Liu S., An efficient algorithm for quadratic sum-of-ratios fractional programs problem, Numer. Func. Anal. Opt. 2017, 38(11), 1426-1445.

[8] Fu M., Luo Z.Q., Ye Y., Approximation algorithms for quadratic programming. J. Comb. Optim., 1998, 2, 29-50.

[9] Shen P., Zhang T., Wang C., Solving a class of generalized fractional programming problems using the feasibility of linear programs, J. Inequal. Appl., 2017, 2017, 147. https://doi.org/10.1186/s13660-017-1420-1

[10] Shen P., Wang C., Linear decomposition approach for a class of nonconvex programming problems, J. Inequal. Appl. 2017, 2017, 74. https://doi.org/10.1186/s13660-017-1342-y

[11] Linderoth J., A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. Math. Program., 2005, 103, 251-282.

[12] Audet C., Hansen P., Jaumard B., Savard G., A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. Math. Program., 2000, 87, 131-152.

[13] Qu S.-J., Zhang K.-C., Ji Y., A global optimization algorithm using parametric linearization relaxation. Appl. Math. Comput., 2007, 186, 763-771.

[14] Jiao H., Chen Y., A global optimization algorithm for generalized quadratic programming, J. Appl. Math., 2013, 2013, 215312. Doi:10.1155/2013/215312.

[15] Shen P., Duan Y., Ma Y., A robust solution approach for nonconvex quadratic programs with additional multiplicative constraints. Appl. Math. Comput., 2008, 201, 514-526.

[16] Shen P., Jiao H., A new rectangle branch-and-pruning appproach for generalized geometric programming. Appl. Math. Comput., 2006, 183, 1027-1038.

[17] Wang Y., Liang Z., A deterministic global optimization algorithm for generalized geometric programming. Appl. Math. Comput., 2005, 168, 722-737.

[18] Wang Y.J., Zhang K.C., Gao Y.L., Global optimization of generalized geometric programming. Comput. Math. Appl., 2004, 48, 1505-1516.

[19] Shen P., Linearization method of global optimization for generalized geometric programming. Appl. Math. Comput., 2005, 162, 353-370.

[20] Shen P., Li X., Branch-reduction-bound algorithm for generalized geometric programming. J. Glob. Optim., 2013, 56(3), 1123-1142.

[21] Jiao H., Liu S., Zhao Y., Effective algorithm for solving the generalized linear multiplicative problem with generalized polynomial constraints, Appl. Math. Model., 2015, 39, 7568-7582.

[22] Jiao H., Liu S., Range division and compression algorithm for quadratically constrained sum of quadratic ratios, Comput. Appl. Math., 2017, 36(1), 225-247.

[23] Hou Z., Jiao H., Cai L., Bai C., Branch-delete-bound algorithm for globally solving quadratically constrained quadratic programs, Open Math., 2017, 15, 1212-1224.

[24] Wang C.-F., Liu S.-Y., Shen P.-P., Global minimization of a generalized linear multiplicative programming, Appl. Math. Model., 2012, 36(6), 2446-2451.

[25] Wang C.-F., Bai Y.-Q., Shen P.-P., A practicable branch-and-bound algorithm for globally solving linear multiplicative programming, Optimization, 2017, 66, 397-405.

[26] Horst R., Tuy H., Global Optimization: Deterministic Approaches, second ed., Springer, Berlin, Germany, 1993.