

Article

A Relaxed and Bound Algorithm Based on Auxiliary Variables for Quadratically Constrained Quadratic Programming Problem

Chenyang Hu ^{1,2}, Yuelin Gao ^{2,3,*} , Fuping Tian ¹ and Suxia Ma ^{1,3}

¹ School of Mathematics and Information Sciences, North Minzu University, Yinchuan 750021, China; huchenyang1997@163.com (C.H.); fptian808@163.com (F.T.); suxiama0220@163.com (S.M.)

² Ningxia Province Cooperative Innovation Center of Scientific Computing and Intelligent Information Processing, North Minzu University, Yinchuan 750021, China

³ Ningxia Province Key Laboratory of Intelligent Information and Data Processing, North Minzu University, Yinchuan 750021, China

* Correspondence: gaoyuelin@263.net; Tel.: +86-139-9510-0900

Abstract: Quadratically constrained quadratic programs (QCQP), which often appear in engineering practice and management science, and other fields, are investigated in this paper. By introducing appropriate auxiliary variables, QCQP can be transformed into its equivalent problem (EP) with non-linear equality constraints. After these equality constraints are relaxed, a series of linear relaxation subproblems with auxiliary variables and bound constraints are generated, which can determine the effective lower bound of the global optimal value of QCQP. To enhance the compactness of sub-rectangles and improve the ability to remove sub-rectangles, two rectangle-reduction strategies are employed. Besides, two ϵ -subproblem deletion rules are introduced to improve the convergence speed of the algorithm. Therefore, a relaxation and bound algorithm based on auxiliary variables are proposed to solve QCQP. Numerical experiments show that this algorithm is effective and feasible.

Keywords: global optimization; quadratically constrained quadratic program; branch and bound method; relaxation technique



Citation: Hu, C.; Gao, Y.; Tian, F.; Ma, S. A Relaxed and Bound Algorithm Based on Auxiliary Variables for Quadratically Constrained Quadratic Programming Problem. *Mathematics* **2022**, *10*, 270. <https://doi.org/10.3390/math10020270>

Academic Editors: Mihai Postolache, Jen-Chih Yao and Yonghong Yao

Received: 22 December 2021

Accepted: 14 January 2022

Published: 16 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In this paper, we investigate the following quadratically constrained quadratic program (QCQP):

$$\begin{aligned} \min \quad & f_0(x) = x^T Q_0 x + c_0^T x \\ \text{s.t.} \quad & f_s(x) = x^T Q_s x + c_s^T x \leq d_s, \quad s = 1, 2, \dots, N, \\ & x \in D, \end{aligned}$$

where $Q_s = (q_{sij})_{n \times n} \in \mathbb{R}^{n \times n}$ is real symmetric, $c_s = (c_{si})_{n \times 1} \in \mathbb{R}^n$, $d_s \in \mathbb{R}$, $D = \{x \in \mathbb{R}^n | Ax \leq b\}$ with $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Furthermore, we assume that the set D is non-empty bounded and that the feasible region $F = \{x \in D | f_s(x) = x^T Q_s x + c_s^T x \leq d_s, s = 1, 2, \dots, N\}$ has at least one interior point (i.e., Slater condition holds). Clearly, $F \subseteq D$.

Problem QCQP has a wide range of applications in such fields as wireless communication, network, radar, signal processing [1,2] and so on. Besides, some of the solutions of this problem can also be applied to solve the nonlinear singular models [3] and nonlinear mathematical based medical smoking model [4,5]. It is also often employed to construct many models of management problems, such as portfolio selection problem [6,7], minimum distance problem [8], location problem [9], 0–1 quadratic programming problem [10], maximum cut problem [11], trust-region subproblem [12,13], etc. If all the matrices Q_s , $s = 0, 1, 2, \dots, N$ are semi-positive definite, QCQP is a convex quadratic program with convex quadratic constraints (CQPCQC), which can be reconstructed into a second-order cone programming problem (SOCP) that can be solvable in polynomial time [14,15]. In other words, the problem CQPCQC is solvable in polynomial time. Of course, other special

QCQP problems in reference [11,16–18] can also be solved in polynomial time. Generally speaking, QCQP is NP-hard, and its complexity is mainly reflected in the non-convexity of quadratic objective function and the feasible region with non-connectivity. Therefore, most of these problems cannot be solved in polynomial time, and it is difficult to search for its global optimal solution.

Over the decades, many scholars have studied QCQP problems and proposed global optimization methods which generally include branch-and-bound methods [19–21], outer approximation method [22], Lagrangian decomposition [23], Benders decomposition [24,25] and their hybrid methods. Branch-and-bound (BB) algorithm is a typical enumeration method [19,20,26], which usually obtains a new lower bound for the optimal value of the original problem and tries to update the upper bound by solving the relaxation subproblem in each enumeration. During the enumeration, the compactness of the relaxation subproblem is one of the main factors. Therefore, BB Algorithms embedded with different types of relaxation problems have been investigated in many pieces of literature about QCQP. Among them, the semidefinite programming (SDP) [27] relaxation method, which can be solved in polynomial time by the interior point method, has attracted the attention of many researchers. Besides, another relaxation method widely studied is linear programming relaxation such as reconstruction linearization technique (RLT) [26], which was adopted in [28,29]. Based on the idea of RLT, a branch and cut algorithm for solving QCQP were proposed in [30]. To find the global approximate optimal solution for QCQP, the author of [31] developed a branch and bound algorithm by mixing the outer approximation method with the linear relaxation approximation subproblem. In [20,32], two branch-and-bound algorithms with linear relaxation, which are based on the method of simply dividing the feasible region, can also solve QCQP globally. By adopting the quadratic proxy function, literature [33] convexifies all quadratic inequality constraint functions, and thus presents a new algorithm based on the quadratic convex reconstruction method. All these methods mentioned above can solve QCQP and its variants well.

In this paper, by introducing the auxiliary variable $Y = xx^T$, the problem QCQP is equivalently converted into an equivalent problem with non-linear equality constraints. After the equality constraint is relaxed, a linear relaxation subproblem with auxiliary variables and bounded constraints is generated for QCQP, and then the lower bound of the global optimal value of QCQP is determined. Then, by combining two rectangle-reduction strategies and two ϵ -subproblem-deletion rules, a relaxation and bound algorithm for QCQP are proposed. Numerical experiments and analysis of the results show that the proposed algorithm is feasible and effective, and its computational performance is better than the software package BARON [34], so it is also a promising algorithm.

The follow-up arrangement of this paper is as follows. In Section 2, the construction method of linear relaxation subproblem with auxiliary variables and bound constraints are given. The rectangle-reduction strategies are presented respectively in Section 3. In Section 4, the proposed algorithm is described and its finite convergence is proved under given precision. Section 5 lists the numerical results and reveals the feasibility and effectiveness of the proposed algorithm. The last section is the conclusion.

2. Bounded Relaxation Technique

By introducing variable $Y = xx^T = (x_i x_j)_{n \times n}$, QCQP can be reformulated as

$$\begin{aligned}
 \min \quad & g_0(Y, x) = Q_0 \cdot Y + c_0^T x \\
 \text{(EP) : } \quad & s.t. \quad g_s(Y, x) = Q_s \cdot Y + c_s^T x \leq d_s, \quad s = 1, 2, \dots, N, \\
 & Y = xx^T = (x_i x_j)_{n \times n}, \\
 & x \in D,
 \end{aligned}$$

where “ \cdot ” denotes the inner product of two matrices. Further, for each $j = 1, 2, \dots, n$, let $\bar{x}_j^0 = \min_{x \in X} x_j$, $\bar{x}_j^0 = \max_{x \in X} x_j$, $j = 1, 2, \dots, n$, then we can construct an initial rectangle

$H^0 = \{x \in \mathbb{R}^n | \underline{x}_j^0 \leq x_j \leq \bar{x}_j^0, j = 1, 2, \dots, n\} = \prod_{j=1}^n [\underline{x}_j^0, \bar{x}_j^0]$ that satisfies $H^0 \supseteq X \supseteq F$. Of course, \underline{x}_j^0 and \bar{x}_j^0 can be obtained by solving linear programs. Also, if D is a bounded convex set, \underline{x}_j^0 and \bar{x}_j^0 must be obtained by solving convex programs.

Next, we mainly study the problem EP, whose non-convexity is mainly reflected in the nonlinear equality constraint $Y = xx^T$. Suppose $H = \prod_{j=1}^n [x_j, \bar{x}_j]$ is denoted as H^0 or any subrectangle of H^0 , then each element of $Y = (y_{ij})_{n \times n}$, y_{ij} , satisfies $\underline{y}_{ij} \leq y_{ij} = x_i x_j \leq \bar{y}_{ij}$, where $\underline{y}_{ij} = \min\{x_i \underline{x}_j, x_i \bar{x}_j, \underline{x}_i \bar{x}_j, \bar{x}_i \bar{x}_j\}$, $\bar{y}_{ij} = \max\{\underline{x}_i \underline{x}_j, \underline{x}_i \bar{x}_j, \bar{x}_i \bar{x}_j\}$. Thus, we obtain the linear relaxation problem of EP over H :

$$\begin{aligned}
 & \min g_0(Y, x) = Q_0 \cdot Y + c_0^T x \\
 \text{(LRP}_H\text{)} : & \quad \text{s.t. } g_s(Y, x) = Q_s \cdot Y + c_s^T x \leq d_s, s = 1, 2, \dots, N, \\
 & \quad \underline{y}_{ij} \leq y_{ij} \leq \bar{y}_{ij}, i, j = 1, 2, \dots, n, \\
 & \quad x \in D \cap H,
 \end{aligned}$$

which of course also relaxes the following subproblem of QCQP on H :

$$\begin{aligned}
 & \min f_0(x) = x^T Q_0 x + c_0^T x \\
 \text{(QCQP}_H\text{)} : & \quad \text{s.t. } f_s(x) = x^T Q_s x + c_s^T x \leq d_s, s = 1, 2, \dots, N, \\
 & \quad x \in D \cap H.
 \end{aligned}$$

Obviously, any feasible solution of QCQP_H is feasible for QCQP.

For convenience, we denote the feasible regions of QCQP_H and LRP_H as F_H and \bar{F}_H respectively. In addition, throughout this paper, $\epsilon > 0$ denotes the convergence accuracy of the proposed algorithm, $\|\cdot\|$ denotes the Euclidean norm of “.”.

The following five lemmas illustrate the relationship between feasible solutions and optimal solutions of LRP_H and QCQP_H.

Lemma 1. *If (\tilde{Y}, \bar{x}) is a feasible solution of LRP_H with $\tilde{Y} = \bar{x}\bar{x}^T$, then \bar{x} is feasible for QCQP_H.*

Proof. Since (\tilde{Y}, \bar{x}) is a feasible solution of LRP_H with $\tilde{Y} = \bar{x}\bar{x}^T$, it follows that $\bar{x}^T Q_s \bar{x} + c_s^T \bar{x} = Q_s \cdot \tilde{Y} + c_s^T \bar{x} \leq d_s$ for $s = 1, 2, \dots, N$. This means $\bar{x} \in F_H$. □

Lemma 2. *If (\tilde{Y}, \bar{x}) is an optimal solution of LRP_H with $\tilde{Y} = \bar{x}\bar{x}^T$, then \bar{x} is a global optimal solution of QCQP_H.*

Proof. Since (\tilde{Y}, \bar{x}) is an optimal solution of LRP_H with $\tilde{Y} = \bar{x}\bar{x}^T$, then we have $g_0(Y, x) \geq g_0(\tilde{Y}, \bar{x}) = f_0(\bar{x})$ for any $(Y, x) \in \bar{F}_H$. Let $\tilde{Y} = \tilde{x}\tilde{x}^T$ for any $\tilde{x} \in F_H$, then $(\tilde{Y}, \tilde{x}) \in \bar{F}_H$ is obvious. Thus, for any $\tilde{x} \in F_H$, it follows that $f_0(\tilde{x}) = g_0(\tilde{Y}, \tilde{x}) \geq g_0(\tilde{Y}, \bar{x}) = f_0(\bar{x})$, which means \bar{x} is an optimal solution to QCQP_H. □

Definition 1. *If there is a $(\tilde{Y}, \bar{x}) \in \bar{F}_H$ such that $\bar{x} \in F_H$ and $f_0(\bar{x}) \leq \min_{x \in F_H} f_0(x) + \epsilon$, then \bar{x} is called an ϵ -globally optimal solution for QCQP_H.*

Definition 2. *If there is a $(\tilde{Y}, \bar{x}) \in \bar{F}_H$ such that $|f_s(\bar{x}) - g_s(\tilde{Y}, \bar{x})| \leq \epsilon$ for $s = 1, \dots, N$ and satisfies $f_0(\bar{x}) \leq \min_{x \in F_H} f_0(x) + \epsilon$, then \bar{x} is called a forced ϵ -globally optimal solution for QCQP_H.*

Definition 3. *If (\tilde{Y}, \bar{x}) is an optimal solution of LRP_H with $|f_s(\bar{x}) - g_s(\tilde{Y}, \bar{x})| \leq \epsilon$ for $s = 0, 1, \dots, N$, then LRP_H is called a forced ϵ -approximation problem for QCQP_H.*

In Definition 1, (\tilde{Y}, \bar{x}) is a feasible solution of LRP_H. In fact, there is no absolute optimal solution in practice, so we just need to find the approximate optimal solution under the

required tolerance $\epsilon > 0$. If a feasible point \bar{x} satisfies the inequality $f_0(\bar{x}) \leq \min_{x \in F_H} f_0(x) + \epsilon$, it means that $f_0(\bar{x})$ is close enough to the optimal value $\min_{x \in F_H} f_0(x)$ of the problem QCQP_H to be usable under the given tolerance. In Definition 2, if $|f_s(\bar{x}) - g_s(\bar{Y}, \bar{x})| \leq \epsilon$ for every $s = 1, \dots, N$, which means that \bar{x} is very close to the feasible domain F_H of QCQP_H. On this basis, if there is such a solution \bar{x} that also satisfies the inequality $f_0(\bar{x}) \leq \min_{x \in F_H} f_0(x) + \epsilon$, we call it a forced ϵ -globally optimal solution, which is inferior to the ϵ -global optimal solution. For some problems with more complex feasible regions, feasible solutions may not always be available, so it may be a better choice to have an infeasible forced ϵ -globally optimal solution to save more computation. It can be seen from Definition 3 that when $|f_s(\bar{x}) - g_s(\bar{Y}, \bar{x})| \leq \epsilon$ for $s = 0, 1, \dots, N$, not only \bar{x} is close to the feasible region of QCQP_H, but also the optimal value of the relaxation problem LRP_H is close to the objective function value $f_0(\bar{x})$ of the problem QCQP_H. In addition, it is not too difficult to deduce that $f_0(\bar{x}) \leq g_0(\bar{Y}, \bar{x}) + \epsilon = \min_{(Y,x) \in \bar{F}_H} g_0(Y, x) + \epsilon \leq \min_{(Y,x) \in \bar{G}_H} g_0(Y, x) + \epsilon = \min_{x \in F_H} f_0(x) + \epsilon$, where the feasible region of EP_H is \bar{G}_H that satisfies $\bar{G}_H \subseteq \bar{F}_H$.

Lemma 3. Let (\bar{Y}, \bar{x}) be an optimal solution to LRP_H. If $\|\bar{Y} - \bar{x}\bar{x}^T\| \leq \frac{\epsilon}{\max\{\max\{\|Q_s\|:s=0,1,\dots,N\},1\}}$, then LRP_H is a forced ϵ -approximation problem for QCQP_H.

Proof. Since $\|\bar{Y} - \bar{x}\bar{x}^T\| \leq \frac{\epsilon}{\max\{\max\{\|Q_s\|:s=0,1,\dots,N\},1\}}$, then for each $s = 0, 1, \dots, N$, we have

$$\begin{aligned} |f_s(\bar{x}) - g_s(\bar{Y}, \bar{x})| &= |Q_s \cdot \bar{x}\bar{x}^T + c_s^T \bar{x} - Q_s \cdot \bar{Y} - c_s^T \bar{x}| \\ &= |Q_s \cdot \bar{x}\bar{x}^T - Q_s \cdot \bar{Y}| \\ &= |Q_s \cdot (\bar{x}\bar{x}^T - \bar{Y})| \\ &\leq \|Q_s\| \|\bar{x}\bar{x}^T - \bar{Y}\| \\ &\leq \frac{\epsilon \|Q_s\|}{\max\{\max\{\|Q_s\| : s = 0, 1, \dots, N\}, 1\}} \\ &\leq \frac{\epsilon \|Q_s\|}{\max\{\|Q_s\|, 1\}} \\ &\leq \epsilon. \end{aligned} \tag{1}$$

Therefore, LRP_H is a forced ϵ -approximation problem for QCQP_H. □

Lemma 4. Let (\bar{Y}, \bar{x}) be an optimal solution to LRP_H. If $\|\bar{Y} - \bar{x}\bar{x}^T\| \leq \frac{\epsilon}{\max\{\max\{\|Q_s\|:s=0,1,\dots,N\},1\}}$, then \bar{x} is a forced ϵ -globally optimal solution for QCQP_H.

Proof. Since $\|\bar{Y} - \bar{x}\bar{x}^T\| \leq \frac{\epsilon}{\max\{\max\{\|Q_s\|:s=0,1,\dots,N\},1\}}$, it follows from Formula (1) that

$$|f_s(\bar{x}) - g_s(\bar{Y}, \bar{x})| = |Q_s \cdot (\bar{x}\bar{x}^T - \bar{Y})| \leq \frac{\epsilon \|Q_s\|}{\max\{\|Q_s\|, 1\}} \leq \epsilon, \quad s = 0, 1, \dots, N. \tag{2}$$

Let $x_H \in \min_{x \in F_H} f_0(x)$ and $Y_H = x_H(x_H)^T$, then $g_0(Y_H, x_H) = f_0(x_H)$ and $(Y_H, x_H) \in \bar{F}_H$ are obvious. It follows from the optimality of (\bar{Y}, \bar{x}) on LRP_H that

$$f_0(x_H) = g_0(Y_H, x_H) \geq g_0(\bar{Y}, \bar{x}). \tag{3}$$

Besides, it knows from Formula (2) that

$$-\epsilon \leq f_0(\bar{x}) - g_0(\bar{Y}, \bar{x}) \leq \epsilon. \tag{4}$$

By combining Formulas (3) and (4), we have

$$f_0(\bar{x}) \leq g_0(\bar{Y}, \bar{x}) + \epsilon \leq f_0(x_H) + \epsilon = \min_{x \in F_H} f_0(x) + \epsilon. \tag{5}$$

Thus, the formulas (2) and (5) indicate that \bar{x} is a strong ϵ -globally optimal solution for QCQP_H. □

Lemma 3 gives a sufficient condition that LRP_H is a forced ϵ -approximation problem of QCQP_H; under this condition, Theorem 4 states that the optimal solution of problem LRP_H can also provide a forced ϵ -globally optimal solution for QCQP_H.

Lemma 5. *Let (\bar{Y}, \bar{x}) be an optimal solution to LRP_H. If $\|\bar{Y} - \bar{x}\bar{x}^T\| \leq \frac{\epsilon}{\max\{\|Q_0\|, 1\}}$ and $\bar{x} \in F_H$, then \bar{x} is an ϵ -globally optimal solution for QCQP_H.*

Proof. Since $\|\bar{Y} - \bar{x}\bar{x}^T\| < \frac{\epsilon}{\max\{\|Q_0\|, 1\}}$ and $\bar{x} \in F_H$, it easily deduced that

$$f_0(\bar{x}) - g_0(\bar{Y}, \bar{x}) = Q_0 \cdot (\bar{x}\bar{x}^T - \bar{Y}) \leq \frac{\epsilon \|Q_0\|}{\max\{\|Q_0\|, 1\}} \leq \epsilon. \tag{6}$$

Let $x_H \in \min_{x \in \bar{F}_H} f_0(x)$ and $Y_H = x_H(x_H)^T$, then $g_0(Y_H, x_H) = f_0(x_H)$ and $(Y_H, x_H) \in \bar{F}_H$. It follows from the optimality of (\bar{Y}, \bar{x}) on LRP_H that

$$f_0(x_H) = g_0(Y_H, x_H) \geq g_0(\bar{Y}, \bar{x}). \tag{7}$$

Therefore, by combining formulas (6) and (7), it once again derives (5), which also means \bar{x} is an ϵ -globally optimal solution for QCQP_H. □

Lemma 5 gives a sufficient condition that the optimal solution of LRP_H can provide an ϵ -globally optimal solution for QCQP_H.

Remark 1. *In our proposed algorithm(see Algorithm 3), over a certain rectangle H, when the ϵ -globally optimal solution or forced ϵ -globally optimal solution of the subproblem QCQP_H is found, if H is deleted, some computational costs will be saved, because such two kinds of solutions are already very close to the global optimal solution of QCQP_H.*

3. Rectangle-Reduction Strategy

In this section, two rectangle-reduction strategies are given, which can improve the computational efficiency of the algorithm as much as possible.

3.1. Rectangle Reduction Technique Based on Linear Constraints

For any sub-rectangle $H = \prod_{j=1}^n [x_j, \bar{x}_j] \subseteq H^0$, we assume that all linear constraints of the problem QCQP_H can be expanded as:

$$\sum_{j=1}^n a_{lj}x_j \leq b_l, \quad l \in \{1, 2, \dots, m\}.$$

Below, the pseudo-code for rectangle-reduction technique based on linear constraints (RRTLTC) (see Algorithm 1) in [21] is given.

Algorithm 1 Rectangle-Reduction Technique based on Linear Constraints

Require: $H = \prod_{j=1}^n [x_j, \bar{x}_j]$.

- 1: Set $I := \{1, 2, \dots, m\}$.
- 2: **for** $l = 1 \rightarrow m$ **do**
- 3: Compute $rU_l := \sum_{j=1}^n \max\{a_{lj}x_j, a_{lj}\bar{x}_j\}$ and $rL_l := \sum_{j=1}^n \min\{a_{lj}x_j, a_{lj}\bar{x}_j\}$.
- 4: **if** $rL_l > b_l$ **then**
- 5: Problem QCQP_H has no feasible solution and the algorithm stops; delete the rectangle H .
- 6: **end if**
- 7: **if** $rU_l \leq b_l$ **then**
- 8: Set $I := I \setminus \{l\}$ and delete the l th linear inequality constraint for QCQP_H.
- 9: **end if**
- 10: **if** $rU_l > b_l \geq rL_l$ **then**
- 11: **for** $j = 1 \rightarrow n$ **do**
- 12: **if** $a_{lj} > 0$ **then**
- 13: $\bar{x}_j := \min\{\bar{x}_j, \frac{b_l - rL_l + \min\{a_{lj}x_j, a_{lj}\bar{x}_j\}}{a_{lj}}\}$
- 14: **end if**
- 15: **if** $a_{lj} < 0$ **then**
- 16: $x_j := \max\{x_j, \frac{b_l - rL_l + \max\{a_{lj}x_j, a_{lj}\bar{x}_j\}}{a_{lj}}\}$
- 17: **end if**
- 18: **end for**
- 19: **end if**
- 20: **end for**
- 21: **return** $H = \prod_{j=1}^n [x_j, \bar{x}_j]$ or \emptyset .

3.2. Rectangle Reduction Technique Based on Quadratic Constraints

To further reduce or delete $H = \prod_{j=1}^n [x_j, \bar{x}_j]$, the following rectangle-reduction Theorem based on quadratic constraints is given.

Theorem 1. Let $\gamma_{si} = d_s - \eta_s + \min\{c_{si}x_i, c_{si}\bar{x}_i\}$ for $s = 1, 2, \dots, N, i = 1, 2, \dots, n$, then for any $s = 1, 2, \dots, N$, if $\gamma_{si} < \min\{c_{si}x_i, c_{si}\bar{x}_i\}$, there is no optimal solution to problem QCQP over H ; otherwise, if there is an index $i \in \{1, 2, \dots, n\}$ such that $c_{si} > 0$ and $\gamma_{si} < c_{si}\bar{x}_i$, there is no optimal solution for problem QCQP over \hat{H} ; if there is an index $i \in \{1, 2, \dots, n\}$ such that $c_{si} < 0$ and $\gamma_{si} < c_{si}x_i$, there is no optimal solution for problem QCQP over \tilde{H} , where

$$\eta_s = \sum_{i=1}^n \sum_{j=1}^n \min\{q_{sij}y_{ij}, q_{sij}\bar{y}_{ij}\} + \sum_{i=1}^n \min\{c_{si}x_i, c_{si}\bar{x}_i\},$$

$$y_{ij} = \min\{x_i x_j, x_i \bar{x}_j, \bar{x}_j x_i, \bar{x}_j \bar{x}_i\}, \bar{y}_{ij} = \max\{x_i x_j, x_i \bar{x}_j, \bar{x}_j x_i, \bar{x}_j \bar{x}_i\},$$

$$\hat{H} = \prod_{j=1}^n \hat{H}_j \text{ with } \hat{H}_j = \begin{cases} [x_j, \bar{x}_j], & j \neq i \\ (\frac{\gamma_{si}}{c_{si}}, \bar{x}_i] \cap [x_i, \bar{x}_i], & j = i, \end{cases}$$

$$\tilde{H} = \prod_{j=1}^n \tilde{H}_j \text{ with } \tilde{H}_j = \begin{cases} [x_j, \bar{x}_j], & j \neq i \\ [x_i, \frac{\gamma_{si}}{c_{si}}) \cap [x_i, \bar{x}_i], & j = i. \end{cases}$$

Proof. If $\gamma_{si} < \min\{c_{si}x_i, c_{si}\bar{x}_i\}$, the inequality $\eta_s > d_s$ is obvious. Therefore, problem QCQP_H is unsolvable.

For any $x \in \hat{H}$ and a $s \in \{1, 2, \dots, N\}$, there must be an index $i \in \{1, 2, \dots, n\}$ such that $\frac{\gamma_{si}}{c_{si}} < x_i \leq \bar{x}_i$. It follows from $c_{si} > 0$ that $\gamma_{si} < c_{si}x_i$. Therefore, by using the Definition of γ_{si} and η_s , it can be concluded that $d_s < \eta_s - c_{si}x_i + c_{si}x_i < \sum_{i=1}^n \sum_{j=1}^n q_{sij}x_i x_j + \sum_{i=1}^n c_{si}x_i$, which means that for any $x \in \hat{H}$, the s th quadratic constraint of problem QCQP is violated.

For any $x \in \tilde{H}$ and a $s \in \{1, 2, \dots, N\}$, there must be an $i \in \{1, 2, \dots, n\}$ such that $\bar{x}_i \leq x_i < \frac{\gamma_{si}}{c_{si}}$. It follows from $c_{si} < 0$ that $\gamma_{si} < c_{si}x_i$. we also have $d_s < \eta_s - c_{si}\bar{x}_i + c_{si}x_i < \sum_{i=1}^n \sum_{j=1}^n q_{sij}x_i x_j + \sum_{i=1}^n c_{si}x_i$, which means that for any $x \in \tilde{H}$, the s th quadratic constraint of problem QCQP is violated. □

Next, by using Theorem 1, the following pseudo-code for the rectangle-reduction technique based on quadratic constraints (RRTQC) (see Algorithm 2) is also given.

Algorithm 2 Rectangle-Reduction Technique based on Quadratic Constraints

Require: $H = \prod_{j=1}^n [x_j, \bar{x}_j]$.

- 1: **for** $i = 1 \rightarrow n$ **do**
- 2: **for** $j = 1 \rightarrow n$ **do**
- 3: Compute $\omega_{ij} = \{x_i x_j, x_i \bar{x}_j, x_j \bar{x}_i, \bar{x}_i \bar{x}_j\}$, $y_{ij} = \min \omega_{ij}$, $\bar{y}_{ij} = \max \omega_{ij}$.
- 4: **end for**
- 5: **end for**
- 6: **for** $s = 1 \rightarrow N$ **do**
- 7: Compute $\eta_s = \sum_{i=1}^n \sum_{j=1}^n \min\{q_{sij} y_{ij}, q_{sij} \bar{y}_{ij}\} + \sum_{i=1}^n \min\{c_{si} x_i, c_{si} \bar{x}_i\}$.
- 8: **if** $\eta_s > d_s$ **then**
- 9: Problem QCQP_H has no feasible solution and the algorithm stops; delete the rectangle H .
- 10: **else**
- 11: **for** $i = 1 \rightarrow n$ **do**
- 12: **if** $c_{si} > 0$ **then**
- 13: $\bar{x}_i = \min\{\frac{\gamma_{si}}{c_{si}}, \bar{x}_i\}$
- 14: **end if**
- 15: **if** $c_{si} < 0$ **then**
- 16: $x_i = \max\{\frac{\gamma_{si}}{c_{si}}, x_i\}$
- 17: **end if**
- 18: **end for**
- 19: **end if**
- 20: **end for**
- 21: **return** $H = \prod_{j=1}^n [x_j, \bar{x}_j]$ or \emptyset .

4. Algorithm and Its Convergence

By embedding the two rectangular reduction algorithms given in the previous section into the branch-and-bound scheme, we develop a new global optimization algorithm for solving QCQP. In addition, in each iteration of the branch-and-bound algorithm, one or two new linear relaxation subproblems are generated, whose optimal values are not lower than the current lower bound. Therefore, the lower bound does not decrease in the current iteration. The update of the upper bound is performed by solving linear relaxation subproblems, and it is not difficult to conclude that the upper bound will not increase in the current iteration. Based on this idea, we give the pseudo-code of the relaxation-and-bound algorithm (RBA) in Algorithm 3 below.

Lemma 6. *In the above algorithm RBA, let (Y^k, x^k) be an optimal solution to problem LRP_{H^k} with $H^k \subseteq H^0$, there are three conclusions as follows:*

- (a) *if $\|Y^k - x^k(x^k)^T\| = 0$, x^k is a global optimal solution to problem QCQP_{H^k};*
- (b) *if $\|Y^k - x^k(x^k)^T\| \leq \frac{\epsilon}{\max\{\|Q_0\|, 1\}}$ and $x^k \in F_{H^k}$, x^k is an ϵ -global optimal solution to problem QCQP_{H^k};*
- (c) *if $\|Y^k - x^k(x^k)^T\| \leq \frac{\epsilon}{\max\{\max\{\|Q_s\|:s=0,1,\dots,N\}, 1\}}$, x^k is a forced ϵ -globally optimal solution to problem QCQP_{H^k}.*

Proof. (a) Since $\|Y^k - x^k(x^k)^T\| = 0$, then $Y^k = x^k(x^k)^T$. It follows from Lemma 2 that x^k is a global optimal solution to problem QCQP_{H^k}. Besides, conclusions (b) and (c) are derived from Lemmas 4 and 5 respectively, which are not stated here. □

A further explanation of the pseudo-code of the algorithm RBA is given in Remarks 2–8 below.

Remark 2. *The above algorithm is based on the branch and bound algorithm framework, and all branching operations are performed according to the standard bisection method described in line 18.*

Remark 3. The termination criterion of the algorithm adopts $U^k - L^k \leq \epsilon$ with upper bound U^k and lower bound L^k . When line 16 is violated, the algorithm does not iterate and the termination criterion is established.

Remark 4. If line 28 of the algorithm is violated, it indicates that the node in the branch and bound tree corresponding to sub-problem $LRP_{H^{ki}}$ is empty, so it is meaningless to consider $LRP_{H^{ki}}$ further.

Remark 5. If line 30 of the algorithm is violated, it indicates that the optimal value L^{ki} of sub-problem $LRP_{H^{ki}}$ is larger than the current upper bound U^k , and the node corresponding to $LRP_{H^{ki}}$ is not further considered, which also implies the execution of the pruning operation; besides, even if x^{ki} is feasible (i.e. $x^{ki} \in F$), it cannot be adopted to update the upper bound U^k , because in this subproblem, there is $f_0(x^{ki}) \geq L^{ki} > U^k$.

Algorithm 3 Relaxation-and-Bound Algorithm

Require: Given a QCQP instance and an error tolerance $\epsilon > 0$.
 1: Construct the initial rectangle $H^0 = \prod_{j=1}^n [x_j^0, \bar{x}_j^0]$ by solving $2n$ linear programs $\underline{x}_j^0 = \min_{x \in X} x_j$ and $\bar{x}_j^0 = \max_{x \in X} x_j$ for each $j = 1, 2, \dots, n$; in this process, all feasible solutions of problem QCQP obtained are stored in the set W ; if there is no feasible solution for QCQP, set $W = \emptyset$.
 2: if $W = \emptyset$ then Set $U^0 = +\infty$,
 3: else Set $U^0 = \min\{f_0(x) : x \in W\}$, $\hat{x}^0 \in \{x \in \mathbb{R}^n | f_0(x) = U^0, x \in W\}$, $\hat{y}^0 = \hat{x}^0(x^0)^T$.
 4: end if
 5: for $i = 1 \rightarrow n$ do
 6: for $j = 1 \rightarrow n$ do
 7: Compute $\omega_{ij}^0 = \{x_i^0 \underline{x}_j^0, x_i^0 \bar{x}_j^0, \bar{x}_i^0 \underline{x}_j^0, \bar{x}_i^0 \bar{x}_j^0\}$, $\underline{v}_{ij}^0 = \min \omega_{ij}^0$, $\bar{v}_{ij}^0 = \max \omega_{ij}^0$.
 8: end for
 9: end for
 10: Solve LRP_{H^0} to obtain its optimal solution (y^0, x^0) and its optimal value L^0 .
 11: if $x^0 \in F$ then $\bar{U} = f_0(x^0)$.
 12: if $\bar{U} < U^0$ then Set $U^0 = \bar{U}$, $\hat{x}^0 = x^0$, $\hat{y}^0 = y^0$.
 13: end if
 14: end if
 15: Set $T = \emptyset$, $k := 0$.
 16: while $U^k - L^k > \epsilon$ do
 17: Set $id = \arg \max\{\bar{x}_j^k - \underline{x}_j^k : j = 1, 2, \dots, n\}$ and let $\bar{x}_{id} = \frac{1}{2}(\underline{x}_{id}^k + \bar{x}_{id}^k)$.
 18: Construct two subrectangles:

$$H^{k1} = \prod_{j=1}^{id-1} [\underline{x}_j^k, \bar{x}_j^k] \times [\underline{x}_{id}^k, \bar{x}_{id}^k] \times \prod_{j=id+1}^n [\underline{x}_j^k, \bar{x}_j^k], H^{k2} = \prod_{j=1}^{id-1} [\underline{x}_j^k, \bar{x}_j^k] \times [\bar{x}_{id}^k, \bar{x}_{id}^k] \times \prod_{j=id+1}^n [\underline{x}_j^k, \bar{x}_j^k].$$

19: for $i = 1 \rightarrow 2$ do
 20: Compress or delete rectangle H^{ki} by using algorithms RRTLTC and RRTQC.
 21: if $H^{ki} \neq \emptyset$ then
 22: for $i = 1 \rightarrow n$ do
 23: for $j = 1 \rightarrow n$ do
 24: Compute $\omega_{ij}^{ki} = \{x_i^{ki} \underline{x}_j^k, x_i^{ki} \bar{x}_j^k, \bar{x}_i^{ki} \underline{x}_j^k, \bar{x}_i^{ki} \bar{x}_j^k\}$, $\underline{v}_{ij}^{ki} = \min \omega_{ij}^{ki}$, $\bar{v}_{ij}^{ki} = \max \omega_{ij}^{ki}$.
 25: end for
 26: end for
 27: Solve $LRP_{H^{ki}}$ to identify the feasibility of this problem.
 28: if $LRP_{H^{ki}}$ is feasible then
 29: its optimal solution (y^{ki}, x^{ki}) and optimal value L^{ki} must be obtained.
 30: if $L^{ki} \leq U^k$ then
 31: if $x^{ki} \in F$ then let $\bar{U} = f_0(x^{ki})$.
 32: if $\bar{U} < U^k$ then set $U^k = \bar{U}$, $\hat{x}^k = x^{ki}$, $\hat{y}^k = y^{ki}$.
 33: end if
 34: if $\|y^{ki} - x^{ki}(x^{ki})^T\| > \frac{\epsilon}{\max\{\|Q_0\|, 1\}}$ then
 35: Put $\{H^{ki}, L^{ki}, (y^{ki}, x^{ki})\}$ into T .
 36: end if
 37: else
 38: if $\|y^{ki} - x^{ki}(x^{ki})^T\| > \frac{\epsilon}{\max\{\max\{\|Q_s\| : s=0,1,\dots,N\}, 1\}}$ then
 39: Put $\{H^{ki}, L^{ki}, (y^{ki}, x^{ki})\}$ into T .
 40: else
 41: let $\bar{U} = f_0(x^{ki})$
 42: if $\bar{U} < U^k$ then set $U^k = \bar{U}$, $\hat{x}^k = x^{ki}$, $\hat{y}^k = y^{ki}$.
 43: end if
 44: end if
 45: end if
 46: end if
 47: end if
 48: end if
 49: end for
 50: Set $k \leftarrow k + 1$; $L^k = \min\{L^k, L^k\} \in T$.
 51: Choose a subproblem $\{H^k, L^k, (y^k, x^k)\}$ such that $H^k = \prod_{j=1}^n [\underline{x}_j^k, \bar{x}_j^k] \in \{H | \{H, L^k, (y^k, x^k)\} \in T\}$.
 52: Set $T = T \setminus \{H^k, L^k, (y^k, x^k)\}$.
 53: end while
 54: return $(\hat{y}^k, \hat{x}^k), U^k$.

Remark 6. According to lines 34–35 of the above algorithm RBA, only all nodes with ϵ -globally optimal solutions are deleted; also, lines 38–39 imply that only nodes with forced ϵ -globally optimal

solutions are deleted. It can be found that when these two stages are not executed, it is actually what we call the “ ϵ -subproblem-deletion rule” in action.

Remark 7. If line 32 or 42 is satisfied, the updating of the upper bound is performed, and the updating of the lower bound is performed online 50.

Remark 8. Line 51 is the node selection operation of the next iteration. Here, we adopt the node corresponding to the first subproblem $\{H^k, L^k, (Y^k, x^k)\}$ with $H^k = \prod_{j=1}^n [x_j^k, \bar{x}_j^k] \in \{H | \{H, L^k, (Y^k, x^k)\} \in T\}$.

To ensure that the two ϵ -subproblem-deletion rules mentioned in Remark 6 can be realized, it is necessary that as rectangle $H^k \subseteq H^0$ is gradually thinned, $\|Y^k - x^k(x^k)^T\|$ gradually approaches zero, where (Y^k, x^k) is an optimal solution to problem LRP_{H^k} . For this reason, the following Lemma is given.

Lemma 7. For any $H^k \subseteq H^0$, let $\Delta(H^k) = \max\{\bar{x}_j^k - \underline{x}_j^k : j = 1, 2, \dots, n\}$ for $k \in \mathbb{N}$. Let (Y^k, x^k) be an optimal solution to problem LRP_{H^k} , then $\|Y^k - x^k(x^k)^T\| \rightarrow 0$ as $\Delta(H^k) \rightarrow 0$.

Proof. Since (Y^k, x^k) is an optimal solution to problem LRP_{H^k} with $H^k \subseteq H^0$. Let $\omega_{ij}^k = \{x_i^k x_j^k, x_i^k \bar{x}_j^k, \underline{x}_i^k \bar{x}_j^k, \bar{x}_i^k x_j^k\}$ for any $i, j \in \{1, 2, \dots, n\}$, it follows that

$$\begin{aligned} \max\{x_i^k x_j^k - \omega_{ij}^k\} &= \max\{x_i^k x_j^k - \{x_i^k x_j^k, x_i^k \bar{x}_j^k, \underline{x}_i^k \bar{x}_j^k, \bar{x}_i^k x_j^k\}\} \\ &= \max\{x_i^k x_j^k - x_i^k x_j^k, x_i^k x_j^k - x_i^k \bar{x}_j^k, x_i^k x_j^k - \underline{x}_i^k \bar{x}_j^k, x_i^k x_j^k - \bar{x}_i^k x_j^k\} \\ &= \max\{0, x_i^k(x_j^k - \bar{x}_j^k), x_j^k(x_i^k - \bar{x}_i^k), x_i^k(x_j^k - \bar{x}_j^k) + \bar{x}_j^k(x_i^k - \bar{x}_i^k)\} \quad (8) \\ &\leq \max\{0, x_i^k(x_j^k - \bar{x}_j^k), x_j^k(x_i^k - \bar{x}_i^k), x_i^k(x_j^k - \bar{x}_j^k) + \bar{x}_j^k(x_i^k - \bar{x}_i^k)\} \\ &\leq |x_i^k| |\bar{x}_j^k - x_j^k| + |\bar{x}_j^k| |\bar{x}_i^k - x_i^k|, \end{aligned}$$

$$\begin{aligned} \max\{x_i^k \bar{x}_j^k - \omega_{ij}^k\} &= \max\{x_i^k \bar{x}_j^k - x_i^k x_j^k, x_i^k \bar{x}_j^k - x_i^k \bar{x}_j^k, x_i^k \bar{x}_j^k - \underline{x}_i^k \bar{x}_j^k, x_i^k \bar{x}_j^k - \bar{x}_i^k \bar{x}_j^k\} \\ &= \max\{x_i^k(\bar{x}_j^k - x_j^k), 0, x_i^k(\bar{x}_j^k - x_j^k) + \bar{x}_j^k(x_i^k - \bar{x}_i^k), \bar{x}_j^k(x_i^k - \bar{x}_i^k)\} \quad (9) \\ &\leq \max\{x_i^k(\bar{x}_j^k - x_j^k), 0, \bar{x}_j^k(\bar{x}_j^k - x_j^k) + \bar{x}_j^k(x_i^k - \bar{x}_i^k), \bar{x}_j^k(x_i^k - \bar{x}_i^k)\} \\ &\leq |\bar{x}_j^k| |\bar{x}_j^k - x_j^k| + |\bar{x}_j^k| |\bar{x}_i^k - x_i^k|, \end{aligned}$$

$$\begin{aligned} \max\{\underline{x}_i^k \bar{x}_j^k - \omega_{ij}^k\} &= \max\{\underline{x}_i^k \bar{x}_j^k - x_i^k x_j^k, \underline{x}_i^k \bar{x}_j^k - x_i^k \bar{x}_j^k, \underline{x}_i^k \bar{x}_j^k - \underline{x}_i^k \bar{x}_j^k, \underline{x}_i^k \bar{x}_j^k - \bar{x}_i^k \bar{x}_j^k\} \\ &= \max\{\underline{x}_i^k(\bar{x}_j^k - x_j^k), \underline{x}_i^k(\bar{x}_j^k - x_j^k) + \bar{x}_j^k(x_i^k - \bar{x}_i^k), 0, \bar{x}_j^k(\underline{x}_i^k - \bar{x}_i^k)\} \quad (10) \\ &\leq \max\{\bar{x}_j^k(\bar{x}_j^k - x_j^k), \bar{x}_j^k(\bar{x}_j^k - x_j^k) + \bar{x}_j^k(x_i^k - \bar{x}_i^k), 0, \bar{x}_j^k(\underline{x}_i^k - \bar{x}_i^k)\} \\ &\leq |\bar{x}_j^k| |\bar{x}_j^k - x_j^k| + |\bar{x}_j^k| |\bar{x}_i^k - x_i^k| \end{aligned}$$

and

$$\begin{aligned} \max\{\bar{x}_i^k x_j^k - \omega_{ij}^k\} &= \max\{\bar{x}_i^k x_j^k - x_i^k x_j^k, \bar{x}_i^k x_j^k - x_i^k \bar{x}_j^k, \bar{x}_i^k x_j^k - \underline{x}_i^k \bar{x}_j^k, \bar{x}_i^k x_j^k - \bar{x}_i^k x_j^k\} \\ &= \max\{\bar{x}_i^k(x_j^k - x_i^k) + x_i^k(\bar{x}_j^k - x_j^k), \bar{x}_i^k(\bar{x}_j^k - x_j^k), \bar{x}_i^k(\bar{x}_j^k - x_j^k), 0\} \quad (11) \\ &\leq \max\{\bar{x}_i^k(\bar{x}_j^k - x_j^k) + x_i^k(\bar{x}_j^k - x_j^k), \bar{x}_i^k(\bar{x}_j^k - x_j^k), \bar{x}_i^k(\bar{x}_j^k - x_j^k), 0\} \\ &\leq |\bar{x}_i^k| |\bar{x}_j^k - x_j^k| + |\bar{x}_i^k| |\bar{x}_i^k - x_i^k|. \end{aligned}$$

Now, let $\underline{y}_{ij}^k = \min \omega_{ij}^k, \bar{y}_{ij}^k = \max \omega_{ij}^k$ for any $i, j \in \{1, 2, \dots, n\}$, it follows that

$$\underline{y}_{ij}^k \leq y_{ij}^k = x_i^k x_j^k \leq \bar{y}_{ij}^k \text{ for } Y^k = (y_{ij}^k)_{n \times n}. \quad (12)$$

Upon formulas (8)–(12), it can be deduced that

$$\begin{aligned}
 y_{ij}^k - x_i^k x_j^k &\leq \bar{y}_{ij}^k - \underline{y}_{ij}^k \\
 &= \max \omega_{ij}^k - \min \omega_{ij}^k \\
 &= \max\{\omega_{ij}^k - \min \omega_{ij}^k\} \\
 &= \max\{\{x_i^k x_j^k, x_i^k \bar{x}_j^k, \underline{x}_j^k \bar{x}_i^k, \bar{x}_i^k \bar{x}_j^k\} - \min \omega_{ij}^k\} \\
 &= \max \max\{x_i^k x_j^k - \omega_{ij}^k, x_i^k \bar{x}_j^k - \omega_{ij}^k, \underline{x}_j^k \bar{x}_i^k - \omega_{ij}^k, \bar{x}_i^k \bar{x}_j^k - \omega_{ij}^k\} \\
 &= \max\{\max\{\underline{x}_i^k \underline{x}_j^k - \omega_{ij}^k\}, \max\{\underline{x}_i^k \bar{x}_j^k - \omega_{ij}^k\}, \\
 &\quad \max\{\bar{x}_i^k \bar{x}_j^k - \omega_{ij}^k\}, \max\{\bar{x}_i^k \bar{x}_j^k - \omega_{ij}^k\}\} \\
 &\leq \max\{|\underline{x}_i^k| |\bar{x}_j^k - \underline{x}_j^k| + |\underline{x}_j^k| |\bar{x}_i^k - \underline{x}_i^k|, |\bar{x}_i^k| |\bar{x}_j^k - \underline{x}_j^k| + |\underline{x}_j^k| |\bar{x}_i^k - \underline{x}_i^k|, \\
 &\quad |\underline{x}_i^k| |\bar{x}_j^k - \underline{x}_j^k| + |\bar{x}_j^k| |\bar{x}_i^k - \underline{x}_i^k|, |\bar{x}_i^k| |\bar{x}_j^k - \underline{x}_j^k| + |\bar{x}_j^k| |\bar{x}_i^k - \underline{x}_i^k|\} \\
 &\leq \max\{|\underline{x}_i^k|, |\bar{x}_i^k|\} \Delta(H^k) + \max\{|\underline{x}_j^k|, |\bar{x}_j^k|\} \Delta(H^k) \\
 &= \max\{|\underline{x}_i^k|, |\bar{x}_i^k|, |\underline{x}_j^k|, |\bar{x}_j^k|\} \Delta(H^k).
 \end{aligned}
 \tag{13}$$

By (13), it follows that

$$\begin{aligned}
 \|Y^k - x^k(x^k)^T\| &= \|(y_{ij}^k)_{n \times n} - (x_i^k x_j^k)_{n \times n}\| \\
 &= \|(y_{ij}^k - x_i^k x_j^k)_{n \times n}\| \\
 &\leq \|(\max\{|\underline{x}_i^k|, |\bar{x}_i^k|, |\underline{x}_j^k|, |\bar{x}_j^k|\} \Delta(H^k))_{n \times n}\| \\
 &= \|(\max\{|\underline{x}_i^k|, |\bar{x}_i^k|, |\underline{x}_j^k|, |\bar{x}_j^k|\})_{n \times n}\| \Delta(H^k).
 \end{aligned}
 \tag{14}$$

Thus, $\|Y^k - x^k(x^k)^T\| \rightarrow 0$ as $\Delta(H^k) \rightarrow 0$. \square

Theorem 2. (a) If the algorithm terminates within finite iterations, an approximately optimal solution for QCQP is found. (b) If the algorithm generates an infinite sequence of iterations, then any accumulation point of the sequence $\{\hat{x}^k\}_{k \in \mathbb{N}}$ is a globally optimal solution to QCQP.

Proof. (a) If the algorithm is finite, assume it stops at the k th iteration. From the termination rule of line 16, we know that

$$U^k - L^k = f_0(\hat{x}^k) - L^k \leq \epsilon. \tag{15}$$

Assuming that the global optimal solution is x^* , we know that

$$U^k = f_0(\hat{x}^k) \geq f_0(x^*) \geq L^k. \tag{16}$$

Hence, it follows from inequalities (15) and (16) that

$$f_0(x^*) + \epsilon \geq L^k + \epsilon \geq f_0(\hat{x}^k). \tag{17}$$

and then part (a) has been proven.

(b) If the algorithm is infinite, and an infinite sequence $\{\hat{x}^k\}_{k \in \mathbb{N}}$ is generated for the QCQP problem by solving the linear relaxation problem LRP_{H^k} , the sequence of optimal solutions for the corresponding problem LRP_{H^k} is $\{(\hat{Y}^k, \hat{x}^k)\}_{k \in \mathbb{N}}$. Without loss of generality, assume that the rectangular sequence $\{H^k = [\underline{x}^k, \bar{x}^k]\}_{k \in \mathbb{N}}$ satisfies $\hat{x}^k \in H^k$ and $H^{k+1} \subset H^k$. In our algorithm, the rectangles are divided continuously into two parts of equal width,

then $\bigcap_{k=1}^{\infty} H^k = \{\hat{x}^k\}_{k \in \mathbb{N}}$. Thus $\lim_{k \rightarrow \infty} H^k = \lim_{k \rightarrow \infty} \hat{x}^k$. Let $\lim_{k \rightarrow \infty} H^k = \lim_{k \rightarrow \infty} \hat{x}^k = x^*$, it follows from Lemma 7 that $\lim_{k \rightarrow \infty} \|\hat{Y}^k - \hat{x}^k(\hat{x}^k)^T\| = 0$, which means

$$Y^* := \lim_{k \rightarrow \infty} \hat{Y}^k = \lim_{k \rightarrow \infty} \hat{x}^k(\hat{x}^k)^T = x^*(x^*)^T. \tag{18}$$

Based on the process of determining upper and lower bounds of the algorithm, we have

$$L^k = g_0(\hat{Y}^k, \hat{x}^k) \leq f_0(x^*) \leq f_0(\hat{x}^k) = U^k, k = 1, 2, \dots \tag{19}$$

Since the sequence $\{L^k = g_0(\hat{Y}^k, \hat{x}^k)\}$ is nondecreasing and bounded, and $\{U^k = f_0(\hat{x}^k)\}$ is decreasing and bounded, they are convergent sequences. Taking the limit on both sides of (19), we have

$$\lim_{k \rightarrow \infty} L^k = \lim_{k \rightarrow \infty} g_0(\hat{Y}^k, \hat{x}^k) \leq f_0(x^*) \leq \lim_{k \rightarrow \infty} f_0(\hat{x}^k) = \lim_{k \rightarrow \infty} U^k. \tag{20}$$

Upon the continuity of function $f(x)$, it follows from (18)-(20) that

$$\lim_{k \rightarrow \infty} L^k = g_0(Y^*, x^*) = f_0(x^*) = \lim_{k \rightarrow \infty} f_0(\hat{x}^k) = \lim_{k \rightarrow \infty} U^k.$$

So, the sequence $\{\hat{x}^k\}$, of which any accumulation point x^* is a global optimal solution of the QCQP problem. \square

In Theorem 2, conclusion (a) merely proves that the proposed algorithm, at the termination of finite iterations, returns an approximately global optimal solution related to $\epsilon > 0$ of the problem QCQP. It is not certain which of the two classes of solutions given in Definitions 1 and 2 is the returned solution. In fact, for a given precision $\epsilon > 0$, the algorithm RBA eventually terminates infinite iterations and returns either an ϵ -global optimal solution or a forced ϵ -globally optimal solution to the problem QCQP. The following Theorem 3 is specified.

Theorem 3. For a given $\epsilon > 0$, if the subproblem $\{H^k, L^k, (Y^k, x^k)\}$ satisfies

$$\Delta(H^k) \leq \frac{\epsilon}{\max\{\max\{\|Q_s\| : s = 0, 1, \dots, N\} \|(\max\{|\underline{x}_i^k|, |\bar{x}_i^k|, |\underline{x}_j^k|, |\bar{x}_j^k|\})_{n \times n}\|, 1\}}, \tag{21}$$

the algorithm RBA terminates at line 16; at the same time, if $\hat{x}^k \in F$, \hat{x}^k is an ϵ -globally optimal solution of QCQP; otherwise, \hat{x}^k is a forced ϵ -globally optimal solution of QCQP.

Proof. When BBA runs to line 16, the subproblem $\{H^k, L^k, (Y^k, x^k)\}$ satisfies

$$\begin{aligned} 0 &\leq f_0(x^k) - g_0(Y^k, x^k) \\ &= Q_0 \cdot (x^k(x^k)^T - Y^k) \\ &\leq \|Q_0\| \|x^k(x^k)^T - Y^k\| \\ &\leq \|Q_0\| \|(\max\{|\underline{x}_i^k|, |\bar{x}_i^k|, |\underline{x}_j^k|, |\bar{x}_j^k|\})_{n \times n}\| \Delta(H^k) \\ &\leq \max\{\|Q_0\| \|(\max\{|\underline{x}_i^k|, |\bar{x}_i^k|, |\underline{x}_j^k|, |\bar{x}_j^k|\})_{n \times n}\|, 1\} \Delta(H^k) \\ &\leq \max\{\max\{\|Q_s\| : s = 0, 1, \dots, N\} \|(\max\{|\underline{x}_i^k|, |\bar{x}_i^k|, |\underline{x}_j^k|, |\bar{x}_j^k|\})_{n \times n}\|, 1\} \Delta(H^k). \end{aligned} \tag{22}$$

Note that the third-to-last inequality of Formula (22) follows from Formula (14) above. By combining Formulas (21) and (22), we have

$$0 \leq f_0(x^k) - g_0(Y^k, x^k) \leq \epsilon, \tag{23}$$

Moreover, the subproblem $\{H^k, L^k, (Y^k, x^k)\}$ also satisfies

$$g_0(Y^k, x^k) = L^k \leq U^k = f_0(\bar{x}^k) \leq f_0(x^k). \tag{24}$$

Further, it follows from Formulas (23) and (24) that $0 \leq U^k - L^k \leq f_0(x^k) - g_0(Y^k, x^k) \leq \epsilon$, which means that the algorithm terminates at line 16. Since L^k is the smallest lower bound at the current iteration, then $\min_{x \in F} f_0(x) + \epsilon \geq L^k + \epsilon \geq U^k = f_0(\hat{x}^k)$, which shows that \hat{x}^k is an ϵ -globally optimal solution of QCQP for $\hat{x}^k \in F$. Besides, if $\hat{x}^k \notin F$, \hat{x}^k must come from line 42 of the algorithm RBA, so it knows that $\|\hat{Y}^k - \hat{x}^k(\hat{x}^k)^T\| \leq \frac{\epsilon}{\max\{\max\{\|Q_s\|:s=0,1,\dots,N\},1\}}$. Then, it follows from (2) that

$$|f_s(\hat{x}^k) - g_s(\hat{Y}^k, \hat{x}^k)| = |Q_s \cdot (\hat{x}^k(\hat{x}^k)^T - \hat{Y}^k)| \leq \frac{\epsilon \|Q_s\|}{\max\{\|Q_s\|, 1\}} \leq \epsilon \text{ for each } s = 1, \dots, N.$$

which means that \hat{x}^k is a forced ϵ -globally optimal solution of QCQP for $\hat{x}^k \notin F$. \square

Theorem 4. *Given an error tolerance $\epsilon > 0$, the algorithm RBA returns a forced ϵ -globally optimal solution for QCQP in at most*

$$\prod_{j=1}^n \left\lceil \frac{\max\{\max\{\|Q_s\| : s = 0, 1, \dots, N\} \|(\max\{|\underline{x}_i^0|, |\bar{x}_i^0|, |\underline{x}_j^0|, |\bar{x}_j^0|\})_{n \times n}\|, 1\}}{\epsilon} (\bar{x}_j^0 - \underline{x}_j^0) \right\rceil$$

times, where $\lceil a \rceil$ represents the smallest integer greater than a .

Proof. Based on Theorem 3, the specific proof is similar to Theorem 3 in [27] and is thus omitted. \square

Remark 9. *If $\Delta(H^k) \leq \frac{\epsilon}{\max\{\|Q_0\| \|(\max\{|\underline{x}_i^k|, |\bar{x}_i^k|, |\underline{x}_j^k|, |\bar{x}_j^k|\})_{n \times n}\|, 1\}}$ and $\hat{x}^k \in F$, the penultimate inequality of formula (22) has been satisfied, so it is not difficult to verify that \hat{x}^k is already an ϵ -globally optimal solution to QCQP; at the same time, the algorithm RBA returns \hat{x}^k in at most*

$$\prod_{j=1}^n \left\lceil \frac{\max\{\|Q_0\| \|(\max\{|\underline{x}_i^0|, |\bar{x}_i^0|, |\underline{x}_j^0|, |\bar{x}_j^0|\})_{n \times n}\|, 1\}}{\epsilon} (\bar{x}_j^0 - \underline{x}_j^0) \right\rceil$$

times.

5. Numerical Experiment and Analysis

In this section, we adopt several test examples to verify the feasibility of the proposed algorithm. Note that all examples are non-convex problems. We compile and execute the code on MATLAB9.0.0.341360(R2016a), respectively. All experimental procedures were performed on a desktop computer with Inter(R) Core(TM) i7-6700, @3.40GHz power processor, 16.00GB memory, and Microsoft Win7 operating system. In the numerical experiment, all linear programs are solved by using the *linprog* solver in Matlab. Besides, CVX 2.2 [35] is adopted to execute and solve all convex optimization problems in Matlab.

To demonstrate the practical application of the proposed algorithm, we first consider the following balanced transportation problem presented in [36].

$$\min \frac{\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}}{\sum_{i=1}^m \sum_{j=1}^n d_{ij} x_{ij}}, \text{ s.t. } \sum_{j=1}^n x_{ij} = a_i, \sum_{i=1}^m x_{ij} = b_j, x_{ij} \geq 0,$$

where, $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$, a_i denotes the i th source, b_j denotes the j th destination, c_{ij} denotes the unit cost from i th source to j th destination, d_{ij} denotes the unit preference from i th source to j th destination and the variable x_{ij} denotes the amount of source that the i th source supplies to the j th destination, where $i = 1, 2, \dots, m, j = 1, 2, \dots, n$. By adopting the data in [36], a known mathematical model of the balanced transport problem can be formulated as

$$\begin{aligned} \min & \frac{9x_{11} + 12x_{12} + 7x_{13} + 6x_{14} + 11x_{21} + 9x_{22} + 17x_{23} + 6x_{24} + 5x_{31} + 4x_{32} + 3x_{33} + 9x_{34}}{8x_{11} + 10x_{12} + 12x_{13} + 9x_{14} + 6x_{21} + 4x_{22} + 8x_{23} + 11x_{24} + 9x_{31} + 13x_{32} + 11x_{33} + 7x_{34}}, \\ \text{s.t.} & \quad x_{11} + x_{12} + x_{13} + x_{14} = 12, \quad x_{21} + x_{22} + x_{23} + x_{24} = 19, \\ & \quad x_{31} + x_{32} + x_{33} + x_{34} = 17, \quad x_{11} + x_{21} + x_{31} = 3, \\ & \quad x_{12} + x_{22} + x_{32} = 22, \quad x_{13} + x_{23} + x_{33} = 18, \\ & \quad x_{14} + x_{24} + x_{34} = 5, \quad x_{ij} \geq 0, i = 1, 2, 3, j = 1, 2, 3, 4. \end{aligned}$$

Clearly, it follows from the model above that $\sum_{i=1}^4 a_i = 48 = \sum_{j=1}^4 b_j$.

Further, let

$$\begin{aligned} x &= (x_{11}, x_{12}, x_{13}, x_{14}, x_{21}, x_{22}, x_{23}, x_{24}, x_{31}, x_{32}, x_{33}, x_{34})^T, \\ C &= (9, 12, 7, 6, 11, 9, 17, 6, 5, 4, 3, 9)^T, D = (8, 10, 12, 9, 6, 4, 8, 11, 9, 13, 11, 7)^T, \\ b &= (12, 19, 17, 3, 22, 18, 5)^T, A = (A_1, A_2, A_3, A_4, A_5, A_6, A_7)^T, \\ A_1 &= (1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0)^T, A_2 = (0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0)^T, \\ A_3 &= (0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1)^T, A_4 = (1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0)^T, \\ A_5 &= (0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0)^T, A_6 = (0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0)^T, \\ A_7 &= (0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1)^T. \end{aligned}$$

Then, the above model can be rewritten into the following form:

$$\min \frac{C^T x}{D^T x}, \text{ s.t. } Ax = b, x_i \geq 0, i = 1, 2, \dots, 12.$$

which is equivalent to the following quadratic programming problem with a quadratic constraint [37]:

$$\min t, \text{ s.t. } x \in X, C^T x - tD^T x \leq 0, \underline{t} \leq t \leq \bar{t},$$

where $X = \{x \in \mathbb{R}^{12} | Ax = b, x_i \geq 0, i = 1, 2, \dots, 12\}$, $\underline{t} = \min_{x \in X} C^T x / \max_{x \in X} D^T x$, $\bar{t} = \max_{x \in X} C^T x / \min_{x \in X} D^T x$. Then, after solving the balanced transport problem by the proposed algorithm with $\epsilon = 5 \times 10^{-4}$, we obtain the global optimal solution $x^* = (0.0, 0.0, 12.0, 0.0, 3.0, 11.0, 5.0, 0.0, 11.0, 6.0, 0.0)^T$ and the optimal value 0.6553 in 1822.4674 s after 12,549 iteration.

To further test the proposed algorithm, the following 10 test examples are solved by this algorithm and the algorithms in [38,39], and the numerical results are listed in Table 1. The numerical results recorded in Table 2 are obtained by solving a series of random instances generated by Example 11 in four methods(our algorithm, the algorithms in [38,39], and the commercial software package BARON [34]).

In Tables 1 and 2, the meanings of the symbols in the headers of these two tables are as follows: Iter: number of iterations or the average number of iterations; ϵ : tolerance; CPU: CPU running time or average CPU running time; m : number of quadratic constraints; n : Number of variables; r : The number of negative eigenvalues in the matrix Q_0 ; "–": Some algorithms cannot solve the problem within 3600 s in all cases. Moreover, the tolerance ϵ is set to 5×10^{-4} in Examples 1-10 and 5×10^{-3} in Example 11, respectively.

Table 1. Numerical results in Examples 1–10.

Ex.	Ref.	Solution		Optimum	Iter	CPU
1	ours	(5.0000, 1.0000)	−16	24	0.7985	
	[38]	(5.0000, 1.0000)	−16	4	0.0312	
	[39]	(5.0000, 1.0000)	−16	2	0.0234	
2	ours	(1.1772, 2.1769)		1.1771	159	7.5143
	[38]	(1.1771, 2.1771)		1.1771	28	0.2141
	[39]	(1.1771, 2.1771)		1.1771	17	0.1459
3	ours	(2.0000, 1.0000)	−1	87	2.7883	
	[38]	(2.0000, 1.0000)	−1	19	0.1523	
	[39]	(2.0000, 1.0000)	−1	1	0.0295	
4	ours	(2.0000, 1.6667)		6.7778	222	7.8239
	[38]	(2.0000, 1.6667)		6.7778	8	0.1621
	[39]	(2.0000, 1.6667)		6.7778	26	0.4421
5	ours	(0.5003, 0.4996)		0.5	154	5.1439
	[38]	(0.5000, 0.5000)		0.5	34	0.6258
	[39]	(0.5000, 0.5000)		0.5	22	0.4287
6	ours	(2.5540, 3.1323)		118.3835	992	56.5885
	[38]	(2.5557, 3.1303)		118.3837	52	0.7591
	[39]	(2.5554, 3.1307)		118.3837	43	0.6657
7	ours	(1.5, 1.4998)	−1.1629	1941	78.1576	
	[38]	(1.5, 1.5)	−1.1629	18	0.3905	
	[39]	(1.5, 1.5)	−1.1629	13	0.3524	
8	ours	(2, 0)	−2	1	0.1018	
	[38]	(2, 0)	−2	1	0.1118	
	[39]	(2, 0)	−2	1	0.1073	
9	ours	(2, 0)		−2	21	0.7572
	[38]	(2, 0)	−2	17	0.3026	
	[39]	(2, 0)	−2	10	0.1906	
10	ours	(1.0000, 0.1817, 0.9829)−11.3636	17,319	770.00		
	[38]	(1.0000, 0.1817, 0.9829)−11.3636	1882	83.5967		
	[39]	(1.0000, 0.1817, 0.9829)−11.3636	1063	71.3283		

Example 1. [38,39]

$$\min -x_1^2 + x_1x_2 + x_2^2 + x_1 - 2x_2, \quad s.t. \begin{cases} x_1 + x_2 \leq 6, \\ -2x_1^2 + x_2^2 + 2x_1 + x_2 \leq -4, \\ 1 \leq x_1, x_2 \leq 6. \end{cases}$$

Example 2. [38,39]

$$\min x_1, \quad s.t. \begin{cases} -1/16x_1^2 - 1/16x_2^2 + 1/4x_1 + 1/2x_2 \leq 1, \\ 1/14x_1^2 + 1/14x_2^2 - 3/7x_1 - 3/7x_2 \leq -1, \\ 1 \leq x_1, x_2 \leq 5.5. \end{cases}$$

Example 3. [38,39]

$$\min x_1x_2 - 2x_1 + x_2 + 1, \quad s.t. \begin{cases} 8x_1^2 - 6x_1 - 16x_2 \leq -11, \\ -x_2^2 + 3x_1 + 2x_2 \leq 7, \\ 1 \leq x_1 \leq 2.5, 1 \leq x_2 \leq 2.225. \end{cases}$$

Example 4. [38,39]

$$\min x_1^2 + x_2^2, \quad s.t. \begin{cases} 0.3x_1x_2 \geq 1, \\ 2 \leq x_1 \leq 5, 1 \leq x_2 \leq 3. \end{cases}$$

Example 5. [38,39]

$$\min x_1, \quad s.t. \begin{cases} 4x_2 - 4x_1^2 \leq 1, \\ -x_1 - x_2 \leq -1, \\ 0.01 \leq x_1, x_2 \leq 15. \end{cases}$$

Example 6. [38,39]

$$\min 6x_1^2 + 4x_2^2 + 5x_1x_2, \quad s.t. \begin{cases} 6x_1^2 + 4x_2^2 + 5x_1x_2 \\ -6x_1x_2 \leq -48, \\ 0 \leq x_1, x_2 \leq 10. \end{cases}$$

Example 7. [38,39]

$$\min -x_1 + x_1x_2^{0.5} - x_2, \quad s.t. \begin{cases} -6x_1^2 + 8x_2 \leq 3, \\ 3x_1 - x_2 \leq 3, \\ 1 \leq x_1, x_2 \leq 1.5. \end{cases}$$

Example 8. [38,39]

$$\min -x_1^2 + x_1 + x_2^2, \quad s.t. \begin{cases} x_1^2 + x_2^2 \leq 4, \\ (x_1 + x_2)^2 + x_2^2 - 2x_1 \leq 0, \\ 0 \leq x_1, x_2 \leq 2. \end{cases}$$

Example 9.

$$\min -x_1^2 + x_1 + x_2^2, \quad s.t. \begin{cases} x_1^2 + x_2^2 \leq 4, \\ x_1^2 + 1/4x_2^2 - 4x_1 \leq 0, \\ x_1 + x_2 \leq 2, \\ 0 \leq x_1, x_2 \leq 5. \end{cases}$$

Example 10.

$$\min -4x_2 + (x_1 - 1)^2 + x_2^2 - 10x_3^2, \quad s.t. \begin{cases} x_1^2 + x_2^2 + x_3^2 \leq 2, \\ (x_1 - 1)^2 + x_2^2 + x_3^2 \leq 2, \\ 2 - \sqrt{2} \leq x_1 \leq \sqrt{2}, 0 \leq x_2, x_3 \leq \sqrt{2}. \end{cases}$$

From Table 1, it can be observed that all the optimal values obtained by our algorithm in solving these 10 examples are at least consistent with those of the algorithms in [38,39], while the optimal value obtained in solving Example 6 is superior to the other two algorithms. Furthermore, it is not difficult to know that our algorithm consumes the most computing resources, mainly because the linear relaxation subproblem of our algorithm is the worst among these algorithms. Nevertheless, these results are sufficient to demonstrate that the proposed algorithm is effective and feasible for solving QCQP.

For convenience, we represent the algorithm in [38] as JCBB, and the algorithm in [39] as JLBB.

Example 11.

$$\min f_0(x) = x^T Q_0 x, \quad s.t. f_s(x) = x^T Q_s x + c_s^T x \leq d_s, \quad s = 1, 2, \dots, m,$$

where, Q_0, Q_s, c_s and d_s are generated as follows:

- For each $s = 0, 1, \dots, m$, the matrix W_s is generated, in which each element is randomly generated in the interval $[-1,1]$.

- Set $R_s = 0.5(W_s + (W_s)^T)$ for each $s = 0, 1, \dots, m$.
- For each $s = 0, 1, \dots, m$, by using eigenvalue decomposition, $R_s = (P_s)^T D_s P_s$ is generated. Also, it can be noted that D_s is a diagonal matrix.
- The first r ($r \leq n$) diagonal elements of matrix D_0 are replaced by the r numbers randomly generated in the interval $[-10,0]$, and the last $n - r$ diagonal elements of D_0 are replaced by the numbers randomly generated in $[0,10]$.
- For each $s = 1, \dots, m$, replace all diagonal elements of the matrix D_s by randomly generating n numbers in $[1,100]$.
- For each $s = 0, 1, \dots, m$, let $Q_s = (P_s)^T D_s P_s$.
- For each $s = 1, \dots, m$, all elements of the n -dimensional vector c_s are generated randomly in $[-100, 100]$, and the real number d_s is generated randomly in $[1,50]$.

The above construction method of Example 11 shows that the feasible region of this problem consists of m convex quadratic constraints. Therefore, the feasible region of Example 11 is convex, so $2n$ convex optimizations with linear objective functions need to be solved to construct the initial rectangle H^0 . By using Example 11 and each set of parameters (n, m, r) , 10 random examples are generated and solved by algorithms RBA, JCBB, JLBB, and BARON, and then their average results are recorded in Table 2.

From Table 2, it can be observed that our algorithm can be applied to QCQP problems with different forms of objective functions. The CPU running time and iterations of the algorithm are positively correlated with the number n of decision variables. For fixed parameters m and n , the more the number r of negative eigenvalues of matrix Q_0 , the number of iterations and CPU running time increase accordingly. Unfortunately, the computational performance of algorithm RBA is indeed inferior to the algorithms JCBB and JLBB, and when $(m, n, r) = (15, 10, 5), (20, 20, 6)$, our algorithm cannot solve the problem within 3600 s. However, our algorithm has a better numerical result than BARON. In particular, BARON can no longer solve the problem in 3600 s at $(m, n, r) = (7, 5, 3)$, but our algorithm can solve four more groups of problems. By comparing the relaxation techniques of algorithms JCBB and JLBB, it is not difficult to find that these two algorithms adopt a tighter relaxation strategy than our algorithm, which is the main reason why the two algorithms are better than our algorithm. In fact, our linear relaxation subproblem is only obtained by simply relaxing the boundary of the introduced variables, which shows that our algorithm is promising, and maybe we can incorporate some relaxation techniques(e.g., SDP or RLT) into our relaxation method to develop better algorithms.

Table 2. Numerical results in Example 11.

(m, n, r)	Iter				CPU			
	RBA	JCBB	JLBB	BARON	RBA	JCBB	JLBB	BARON
(5,3,1)	2659.5	839.5	445.4	22,551.3	73.9537	8.7589	7.0782	893.1392
(5,3,2)	3676.4	731.2	378.7	23,815.4	109.0161	7.8601	6.9026	952.2138
(5,3,3)	7546.9	1163.7	581.2	23,322.3	227.1233	11.9196	9.9312	887.3607
(5,5,1)	2968.2	11,938.3	5945.1	182,711.7	1303.9179	149.3092	111.3539	1776.7202
(5,5,3)	30,895.5	11,209.5	6148.6	151,395.5	1335.6709	133.7491	113.0717	1837.5112
(5,5,5)	47,219.6	18,681.0	8296.7	203,510.9	2247.0565	237.5699	157.5038	3228.4427
(7,5,1)	14,117.3	11,017.9	4859.4	205,995.1	687.3498	228.5107	148.5903	3348.2776
(7,5,3)	12,538.2	12,736.9	6232.3	—	579.1162	264.7193	198.5944	—
(10,3,3)	12,657.3	2548.1	1296.4	—	440.3908	27.5611	20.9512	—
(10,10,3)	35,602.1	36,255.8	16,117.3	—	2302.4471	1456.7120	972.1402	—
(10,10,5)	49,875.4	27,257.3	14,686.2	—	3343.4321	1271.3252	985.0749	—
(15,10,5)	—	69,293.8	36,052.9	—	—	3215.0414	2846.0178	—
(20,20,6)	—	—	36,693.8	—	—	—	3544.4622	—

In summary, the above numerical experiments are sufficient to prove the effectiveness and feasibility of the proposed algorithm, and although it is not as good as the algorithms in [38,39], it is better than the commercial software package BARON.

6. Conclusions and Discussion

This paper mainly develops a global optimization algorithm that can solve quadratically constrained quadratic programs. A new linear relaxation technique is proposed by simply extending the feasible domain of the equivalent problem. To speed up the convergence of the algorithm, a new rectangular reduction technique based on quadratic constraint is proposed on the basis of the existing linear constraint-rectangular reduction technique. It is proved that the proposed algorithm converges finitely, and an ϵ -global optimal solution or a forced ϵ -globally optimal solution of the original problem is obtained. Numerical results demonstrate the effectiveness and feasibility of the algorithm. Other techniques that can be integrated into this algorithm are also being investigated. In the future, we will try to generalize and apply the Definitions 1–3 in Section 2 to other optimization problems with quadratic functions, such as the sum of quadratic ratios.

Author Contributions: C.H. carried out the methodology, investigation, and writing the draft. Y.G. supervised the research, gave the methodology, and edited and reviewed the final draft. F.T. and S.M. performed the experiments, and reviewed the final draft. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant (11961001), the Construction Project of first-class subjects in Ningxia higher Education (NXYLXK2017B09), and the Major proprietary funded project of North Minzu University (ZDZX201901).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: We declare there is no conflict of interest.

References

1. He, S.; Luo, Z.Q.; Nie, J.; Zhang, S.Z. Semidefinite relaxation bounds for indefinite homogeneous quadratic optimization. *SIAM J. Optim.* **2008**, *19*, 503–523. [[CrossRef](#)]
2. Manskani, E.; Sidiropoulos, N.D.; Luo, Z.; Tassiulas, L. Convex approximation techniques for joint multiuser downlink beamforming and admission control. *IEEE Trans. Wirel. Commun.* **2008**, *7*, 2682–2693. [[CrossRef](#)]
3. Sabir, Z.; Wahab, H.A.; Guirao, J.L.G. A novel design of Gudermannian function as a neural network for the singular nonlinear delayed, prediction and pantograph differential models. *Math. Biosci. Eng.* **2022**, *19*, 663–687. [[CrossRef](#)] [[PubMed](#)]
4. Sabir, Z.; Raja, M.A.Z.; Alnahdi, A.S.; Jeelani, M.B.; Abdelkawy, M.A. Numerical investigations of the nonlinear smoke model using the Gudermannian neural networks. *Math. Biosci. Eng.* **2021**, *19*, 351–370. [[CrossRef](#)]
5. Saeed, T.; Sabir, Z.; Alhodaly, M.S.; Alsulami, H.H.; Sánchez, Y.G. An advanced heuristic approach for a nonlinear mathematical based medical smoking model. *Results Phys.* **2022**, *32*, 105137. [[CrossRef](#)]
6. Kolbert, F.; Wormald, L. Robust portfolio optimization using second-order cone programming. *Optim. Optim.* **2010**, *1*, 3–22.
7. Gower, J.C. Euclidean distance geometry. *Math. Sci.* **1982**, *7*, 1–14.
8. Klose, A.; Drexler, A. Facility location models for distribution system design. *Eur. J. Oper. Res.* **2005**, *162*, 4–29. [[CrossRef](#)]
9. Vandembussche, D.; Nemhauser, G.L. A polyhedral study of nonconvex quadratic programs with box constraints. *Math. Program.* **2005**, *102*, 531–557. [[CrossRef](#)]
10. Fortin, C.; Wolkowicz, H. The trust region subproblem and semidefinite programming. *Optim. Methods Softw.* **2004**, *19*, 41–67. [[CrossRef](#)]
11. Goemans, M.X.; Williamson, D.P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM* **1995**, *42*, 1115–1145. [[CrossRef](#)]
12. Jeyakumar, V.; Li, G.Y. Trust-region problems with linear inequality constraints: Exact SDP relaxation, global optimality and robust optimization. *Math. Program.* **2014**, *147*, 171–206. [[CrossRef](#)]
13. Bomze, I.M.; Locatelli, M.; Tardella, F. New and old bounds for standard quadratic optimization: Dominance, equivalence and incomparability. *Math. Program.* **2008**, *115*, 31–64. [[CrossRef](#)]
14. Vandenberghe, L.; Boyd, S. Semidefinite programming. *SIAM Rev.* **1996**, *38*, 49–95. [[CrossRef](#)]
15. Beck, A.; Eldar, Y.C. Strong duality in nonconvex quadratic optimization with two quadratic constraints. *SIAM J. Optim.* **2006**, *17*, 844–860. [[CrossRef](#)]
16. Lu, C.; Deng, Z.B.; Zhou, J.; Guo, X.L. A sensitive-eigenvector based global algorithm for quadratically constrained quadratic programming. *J. Glob. Optim.* **2019**, *73*, 371–388. [[CrossRef](#)]

17. Kim, S.; Kojima, M. Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations. *Comput. Optim. Appl.* **2003**, *26*, 143–154. [[CrossRef](#)]
18. Vandenbussche, D.; Nemhauser, G.L. A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. *Math. Program.* **2005**, *102*, 559–575. [[CrossRef](#)]
19. Burer, S.; Vandenbussche, D. A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Math. Program.* **2008**, *113*, 259–282. [[CrossRef](#)]
20. Linderoth, J. A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. *Math. Program.* **2005**, *103*, 251–282. [[CrossRef](#)]
21. Gao, Y.; Wei, F. A new bound-and-reduce approach of nonconvex quadratic programming problems. *Appl. Math. Comput.* **2015**, *250*, 298–308. [[CrossRef](#)]
22. Reemtsen, R. Some outer approximation methods for semi-infinite optimization problems. *J. Comput. Appl. Math.* **1994**, *53*, 87–108. [[CrossRef](#)]
23. Elloumi, S.; Faye, A.; Soutif, E. Decomposition and Linearization for 0–1 Quadratic Programming. *Ann. Oper. Res.* **2000**, *99*, 79–93. [[CrossRef](#)]
24. Benders, J.F. Partitioning Procedures for Solving Mixed-Variables Programming Problems. *Numer. Math.* **1962**, *4*, 238–252. [[CrossRef](#)]
25. Geoffrion, A.M. Generalized Benders decomposition. *J. Optimiz. Theory Appl.* **1972**, *10*, 237–260. [[CrossRef](#)]
26. Sherali, H.D.; Adams, W.P. A reformulation-linearization technique for solving discrete and continuous nonconvex problems. *Comput. Math. Appl.* **1999**, *38*, 288.
27. Lu, C.; Deng, Z.B.; Jin, Q. An eigenvalue decomposition based branch-and-bound algorithm for nonconvex quadratic programming problems with convex quadratic constraints. *J. Glob. Optim.* **2017**, *67*, 475–493. [[CrossRef](#)]
28. Sherali, H.D.; Adams, W.P. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discret. Math.* **1990**, *3*, 411–430. [[CrossRef](#)]
29. Sherali, H.D. Reformulation-Linearization Methods for Global Optimization. In *Encyclopedia of Optimization*; Floudas, C.A., Pardalos, P.M. Eds.; Springer: Boston, FL, USA, 2001; pp. 2182–2186.
30. 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. [[CrossRef](#)]
31. Al-Khayyal, F.A.; Larsen, C.; Van, V.T. A relaxation method for nonconvex quadratically constrained quadratic programs. *J. Glob. Optim.* **1995**, *6*, 215–230. [[CrossRef](#)]
32. Raber, U. A simplicial branch-and-bound method for solving nonconvex all-quadratic programs. *J. Glob. Optim.* **1998**, *13*, 417–432. [[CrossRef](#)]
33. Zheng, X.; Pan, Y.; Cui, X. Quadratic convex reformulation for nonconvex binary quadratically constrained quadratic programming via surrogate constraint. *J. Glob. Optim.* **2018**, *70*, 719–735. [[CrossRef](#)]
34. Sahinidis, N. BARON User Manual v.21.1.13[EB/OL]. 2021. Available online: [Http://minlp.com](http://minlp.com) (accessed on 6 January 2022).
35. Grant, M.; Boyd, S. CVX: Matlab Software for Disciplined Convex Programming, Version 2.2. Available online: <http://cvxr.com/cvx/download> (accessed on 15 December 2021).
36. Sivri, M.; Emiroglu, I.; Guler, C.; Tasci, F. A solution proposal to the transportation problem with the linear fractional objective function. In Proceedings of the Fourth International Conference on Modelling, Simulation and Applied Optimization, Kuala Lumpur, Malaysia, 19–21 April 2011.
37. Jiao, H.W.; Liu, S.Y. A new linearization technique for minimax linear fractional programming. *Intern. J. Comp. Math.* **2014**, *91*, 1730–1743. [[CrossRef](#)]
38. Jiao, H.W.; Chen, Y.Q. A Global Optimization Algorithm for Generalized Quadratic Programming. *J. Appl. Math.* **2013**, *2013*, 1–9. [[CrossRef](#)]
39. Jiao, H.W.; Liu, S.Y.; Lu, N. A parametric linear relaxation algorithm for globally solving nonconvex quadratic programming. *Appl. Math. Comput.* **2015**, *250*, 973–985. [[CrossRef](#)]