symmetry

MDPI

Article

An Effective Global Optimization Algorithm for Quadratic Programs with Quadratic Constraints

Dongwei Shi *, Jingben Yin and Chunyang Bai

School of Mathematical Sciences, Henan Institute of Science and Technology, Xinxiang 453003, China; jingbenyin@126.com (J.Y.); bchyang@163.com (C.B.)
* Correspondence: shidongwei@hist.edu.cn

Received: 13 January 2019; Accepted: 7 March 2019; Published: 22 March 2019

Abstract: This paper will present an effective algorithm for globally solving quadratic programs with quadratic constraints. In this algorithm, we propose a new linearization method for establishing the linear programming relaxation problem of quadratic programs with quadratic constraints. The proposed algorithm converges with the global optimal solution of the initial problem, and numerical experiments show the computational efficiency of the proposed algorithm.

Keywords: quadratic programs with quadratic constraints; global optimization; new linearization method; branch-and-bound

1. Introduction

Quadratic programs with quadratic constraints (QPWQC) have attracted the attention of many researchers for several decades. On the one hand, it is since these classes of problems have a broad applications in multistage shipping, path planning, finance, and portfolio optimization, among others. [1–11]. On the other hand, it is because these classes of problems exist as important theoretical complexities and computational difficulties, that is to say, they are known to generally possess multiple local optimal solutions, which are not optimal solutions.

In the last several decades, many algorithms have been developed for globally solving the (QPWQC) and its special cases, such as branch-and-bound method [12,13], approximation approach [14], robust approach [15], branch-reduce-bound algorithm [16–19], geometric programming approach [20–23], and others. Except for the above approaches, some global optimization algorithms [24–38] for linear multiplicative programming problems and generalized linear fractional programming problems can be used to solve the quadratic programs with quadratic constraints (QPWQC) considered in this paper. Although these algorithms can be employed to solve the QPWQC and its special cases, less work has been done for globally solving the QPWQC considered in this paper.

In this paper, first of all, by making use of the characteristics of simple variable quadratic function, we construct a new linearization method for establishing the linear programming relaxation problem of the QPWQC. Next, we present a global optimization algorithm based on the branch-and-bound scheme for solving the QPWQC. Finally, the global convergence of the proposed algorithm is proved, and numerical experimental results demonstrate the higher computational efficiency of the proposed algorithm.

The main features of the proposed algorithm are given as follows. (1) A new linearization method is proposed for systematically converting the QPWQC into a sequence of linear programming relaxation problems, and the solutions of these linear programming relaxation problems can infinitely approximate the global optimal solution of the original QPWQC by subdividing the linear relaxation of the feasible region of the QPWQC and solving a series of linear programming relaxation problems. (2) The constructed linear programming relaxation problems are embedded within a branch-and-bound framework, which can be effectively solved by any efficient linear programming method. (3) Combining the proposed linear programming relaxation problem with the branch-and-bound framework, an effective algorithm is proposed for solving the problem of QPWQC. (4) Compared with the exist algorithms [37,39–47], numerical results show that the proposed algorithm in this paper can be used to globally solve the QPWQC with higher computational efficiency.

The remaining sections of this paper are organized as follows. Firstly, the aim of Section 2 is to propose a new linearization method for establishing the linear programming relaxation problem of the initial QPWQC. Secondly, based on the branch-and-bound scheme, Section 3 proposes a global optimization algorithm, and its global convergence is proved. Thirdly, compared with the existing methods, Section 4 describes some numerical examples to show the computational efficiency of the proposed algorithm. Finally, some conclusions are given.

2. New Linearization Method for Deriving Linear Programming Relaxation Problem

In this paper, the mathematical modeling of quadratic programs with quadratic constraints is given as follows:

$$(\text{QPWQC})\begin{cases} \min\ \psi_0(x) = \sum\limits_{k=1}^{n} c_k^o x_k + \sum\limits_{j=1}^{n}\sum\limits_{k=1}^{n} d_{ij}^0 x_j x_k \\ \ s.t.\ \ \psi_i(x) = \sum\limits_{k=1}^{n} c_k^i x_k + \sum\limits_{j=1}^{n}\sum\limits_{k=1}^{n} d_{ij}^i x_j x_k \leq b_i,\ i = 1, 2, \ldots, m, \\ x \in X^0 = \{\ x \in R^n :\ l^0 \leq x \leq u^0\}, \end{cases} \tag{1}$$

where d_{jk}^i, c_k^i, and b_i are all arbitrary real numbers; $l^0 = (l_1^0, \ldots, l_n^0)^T > -\infty$, $u^0 = (u_1^0, \ldots, u_n^0)^T < +\infty$.

In this section, we construct a new linearization method for deriving the linear programming relaxation problem of the QPWQC, and the detailed construction process of the linearization method is described as follows.

For convenience, we assume without loss of generality that $X = \{(x_1, x_2, \ldots, x_n)^T \in R^n : l_j \leq x_j \leq u_j, j = 1, 2, \ldots, n\} \subseteq X^0$.

Theorem 1. *For any $x \in X$, $k \in \{1, 2, \ldots, n\}$, we consider the functions x_k^2, $u_k^2 + 2u_k(x_k - u_k)$ and $u_k^2 + 2l_k(x_k - u_k)$, we have the following conclusions:*

$$u_k^2 + 2u_k(x_k - u_k) \leq x_k^2 \leq u_k^2 + 2l_k(x_k - u_k); \tag{2}$$

$$\lim_{\|u-l\|\to 0}\{x_k^2 - [u_k^2 + 2u_k(x_k - u_k)]\} = 0 \tag{3}$$

$$\lim_{\|u-l\|\to 0}\{u_k^2 + 2l_k(x_k - u_k) - x_k^2\} = 0 \tag{4}$$

Proof. (i) From the mean value theorem, there exists a point $\xi_k = \alpha l_k + (1-\alpha)u_k \in [l_k, u_k]$, where $\alpha \in [0, 1]$, which satisfies that

$$x_k^2 = u_k^2 + 2\xi_k(x_k - u_k). \tag{5}$$

From $l_k \leq \xi_k \leq u_k$, it follows that

$$u_k^2 + 2l_k(x_k - u_k) \geq u_k^2 + 2\xi_k(x_k - u_k) = x_k^2 \geq u_k^2 + 2u_k(x_k - u_k). \tag{6}$$

(ii) From

$$x_k^2 - [u_k^2 + 2u_k(x_k - u_k)] = u_k^2 - x_k^2 \leq (u_k - l_k)^2, \tag{7}$$

it follows that

$$\lim_{\|u-l\|\to 0}\{x_k^2-[u_k^2+2u_k(x_k-u_k)]\}=0. \tag{8}$$

Also from

$$u_k^2+2l_k(x_k-u_k)-x_k^2=(x_k-u_k)[2l_k-u_k-x_k]\le 2(u_k-l_k)^2. \tag{9}$$

Therefore, we have

$$\lim_{\|u-l\|\to 0}\{u_k^2+2l_k(x_k-u_k)-x_k^2\}=0. \tag{10}$$

The proof is completed. □

From the conclusion (2), it follows that

$$u_j^2+2u_j(x_j-u_j)\le x_j^2\le u_j^2+2l_j(x_j-u_j), \tag{11}$$

$$(x_j-x_k)^2\ge (u_j-l_k)^2+2(u_j-l_k)[x_j-x_k-(u_k-l_k)], \tag{12}$$

$$(x_j-x_k)^2\le (u_j-l_k)^2+2(l_j-u_k)[x_j-x_k-(u_k-l_k)]. \tag{13}$$

From the conclusions (3) and (4), it follows that

$$\lim_{\|u-l\|\to 0}\{x_j^2-[u_j^2+2u_j(x_j-u_j)]\}=0, \tag{14}$$

$$\lim_{\|u-l\|\to 0}\{u_j^2+2l_j(x_j-u_j)-x_j^2\}=0, \tag{15}$$

$$\lim_{\|u-l\|\to 0}\{(u_j-l_k)^2+2(l_j-u_k)[x_j-x_k-(u_k-l_k)]-(x_j-x_k)^2\}=0. \tag{16}$$

and

$$\lim_{\|u-l\|\to 0}\{(x_j-x_k)^2-\{\,(u_j-l_k)^2+2(u_j-l_k)[x_j-x_k-(u_j-l_k)]\}\}=0. \tag{17}$$

For any $x\in X$, $j,k\in\{1,2,\dots,n\}, j\neq k$, without loss of generality, we define

$$\begin{aligned}\underline{\psi}_{jk}(x)=\tfrac{1}{2}\quad &\{u_j^2+2u_j(x_j-u_j)+u_k^2+2u_k(x_k-u_k)\\ &-\{(u_j-l_k)^2+2(l_j-u_k)[x_j-x_k-(u_k-l_k)]\}\}]\end{aligned} \tag{18}$$

and

$$\begin{aligned}\overline{\psi}_{jk}(x)=\tfrac{1}{2}\quad &\{u_j^2+2l_j(x_j-u_j)+u_k^2+2l_k(x_k-u_k)\\ &-\{(u_j-l_k)^2+2(u_j-l_k)[x_j-x_k-(u_k-l_k)]\}\}],\end{aligned} \tag{19}$$

Theorem 2. *For any $x\in X$, $j,k\in\{1,2,\dots,n\}, j\neq k$, consider the functions $\underline{\psi}_{jk}(x)$, x_jx_k and $\overline{\psi}_{jk}(x)$, the following conclusions hold:*

$$\underline{\psi}_{jk}(x)\le x_jx_k=\frac{1}{2}[x_j^2+x_k^2-(x_j-x_k)^2]\le\overline{\psi}_{jk}(x), \tag{20}$$

$$\lim_{\|u-l\|\to 0}[x_jx_k-\underline{\psi}_{jk}(x)]=0, \tag{21}$$

and

$$\lim_{\|u-l\|\to 0}[\overline{\psi}_{jk}(x)-x_jx_k]=0. \tag{22}$$

Proof. (i) By the conclusions of Theorem 1, it follows that

$$\begin{aligned}\overline{\psi}_{jk}(x) = & \tfrac{1}{2}\{u_j^2 + 2l_j(x_j - u_j) + u_k^2 + 2l_k(x_k - u_k) \\ & -\{(u_j - l_k)^2 + 2(u_j - l_k)[x_j - x_k - (u_k - l_k)]\}\} \\ & \geq \tfrac{1}{2}[x_j^2 + x_k^2 - (x_j - x_k)^2] = x_j x_k \\ & \geq \tfrac{1}{2}\{u_j^2 + 2u_j(x_j - u_j) + u_k^2 + 2u_k(x_k - u_k) \\ & -\{(u_j - l_k)^2 + 2(l_j - u_k)[x_j - x_k - (u_j - l_k)]\}\} \\ & = \underline{\psi}_{jk}(x).\end{aligned} \tag{23}$$

(ii) From the inequalities (7) and (9), we have

$$\begin{aligned}x_j x_k - \underline{\psi}_{jk}(x) = \tfrac{1}{2} \quad & [x_j^2 + x_k^2 - (x_j - x_k)^2] - \tfrac{1}{2}\{u_j^2 + 2u_j(x_j - u_j) + u_k^2 + 2u_k(x_k - u_k) \\ & -\{(u_j - l_k)^2 + 2(l_j - u_k)[x_j - x_k - (u_j - l_k)]\}\} \\ & \leq \tfrac{1}{2}(u_j - l_j)^2 + \tfrac{1}{2}(u_k - l_k)^2 + (u_k + u_j - l_j - l_k)^2.\end{aligned} \tag{24}$$

Thus, we can get that $\lim\limits_{\|u-l\|\to 0}[x_j x_k - \underline{\psi}_{jk}(x)] = 0.$ □

Also from the proof of Theorem 2 and the inequalities (7) and (9), we get that

$$\begin{aligned}\overline{\psi}_{jk}(x) - x_j x_k = \tfrac{1}{2} \quad & \{u_j^2 + 2l_j(x_j - u_j) + u_k^2 + 2l_k(x_k - u_k) \\ & -\{(u_j - l_k)^2 + 2(u_j - l_k)[x_j - x_k - (u_j - l_k)]\}\} \\ & -\tfrac{1}{2}[x_j^2 + x_k^2 - (x_j - x_k)^2] \\ & \leq (u_j - l_j)^2 + (u_k - l_k)^2 + \tfrac{1}{2}(u_k + u_j - l_k - l_j)^2.\end{aligned} \tag{25}$$

Thus, we can get that $\lim\limits_{\|u-l\|\to 0}[\overline{\psi}_{jk}(x) - x_j x_k] = 0.$

Without loss of generality, for any $X = [l, u] \subseteq X^0$ for any $x \in X$, and $i \in \{0, 1, 2, \ldots, m\}$ we let

$$\underline{f}_{kk}^i = \begin{cases} d_{kk}^i\{u_k^i + 2u_k(x_k - u_k)\}, & if\ d_{kk}^i > 0\,, \\ d_{kk}^i\{u_k^i + 2l_k(x_k - u_k)\}, & if\ d_{kk}^i < 0, \end{cases} \tag{26}$$

$$\underline{f}_{jk}^i = \begin{cases} d_{jk}^i \underline{\psi}_{jk}(x), & if\ d_{jk}^i > 0\,, j \neq k, \\ d_{jk}^i \overline{\psi}_{jk}(x), & if\ d_{jk}^i < 0\,, j \neq k, \end{cases} \tag{27}$$

$$\psi_i^L(x) = \sum_{k=1}^{n}(c_k^i x_k + \underline{f}_{kk}^i(x)) + \sum_{j=1}^{n}\sum_{k=1, k\neq j}^{n}\underline{f}_{jk}^i(x). \tag{28}$$

Theorem 3. *For any $x \in X = [l, u] \subseteq X^0$, for each $i = 0, 1, 2, \ldots, m$, we get that $\psi_i^L(x) \leq \psi_i(x)$ and* $\lim\limits_{\|u-l\|\to 0}[\psi_i(x) - \psi_i^L(x)] = 0.$

Proof. (i) From (2) and (12), we have

$$\underline{f}_{kk}^i \leq d_{kk}^i x_k^2 \leq \overline{f}_{kk}^i(x) \quad and \quad \underline{f}_{jk}^i \leq d_{jk}^i x_j x_k \leq \overline{f}_{jk}^i(x). \tag{29}$$

By (29), it follows that $\psi_i^L(x) \leq \psi_i(x)$.

(ii)

$$\begin{aligned}
\psi_i(x) - \psi_i^L(x) = & \sum_{k=1}^{n} c_k^i x_k + \sum_{k=1}^{n} d_{kk}^i x_k^2 + \sum_{j=1}^{n} \sum_{k=1,k\neq j}^{n} d_{jk}^i x_j x_k \\
& - \left[\sum_{k=1}^{n} c_k^i x_k + \sum_{k=1}^{n} \underline{f}_{kk}^i(x) + \sum_{j=1}^{n} \sum_{k=1,k\neq j}^{n} \underline{f}_{jk}^i(x) \right] \\
& = \sum_{k=1}^{n} (d_{kk}^i x_k^2 - \underline{f}_{kk}^i(x)) + \sum_{j=1}^{n} \sum_{k=1,k\neq j}^{n} \left[d_{jk}^i x_j x_k - \underline{f}_{jk}^i(x) \right] \\
& = \sum_{k=1,d_{kk}^i>0}^{n} d_{kk}^i \{x_k^2 - [u_k^2 + 2u_k(x_k - u_k)]\} \\
& + \sum_{k=1,d_{kk}^i<0}^{n} d_{kk}^i \{x_k^2 - [u_k^2 + 2l_k(x_k - u_k)]\} \\
& + \sum_{j=1}^{n} \sum_{k=1,k\neq j,d_{kk}^i>0}^{n} d_{jk}^i \left[x_j x_k - \underline{\psi}_{jk}(x) \right] \\
& + \sum_{j=1}^{n} \sum_{k=1,k\neq j,d_{kk}^i<0}^{n} d_{jk}^i \left[x_j x_k - \overline{\psi}_{jk}(x) \right]
\end{aligned} \tag{30}$$

From (3), (4), (14), and (15), we get

$$\lim_{\|u-l\|\to 0} \{x_k^2 - [u_k^2 + 2u_k(x_k - u_k)]\} = 0, \tag{31}$$

$$\lim_{\|u-l\|\to 0} \{[u_k^2 + 2l_k(x_k - u_k)] - x_k^2\} = 0, \tag{32}$$

$$\lim_{\|u-l\|\to 0} [x_j x_k - \underline{\psi}_{jk}(x)] = 0 \tag{33}$$

and

$$\lim_{\|u-l\|\to 0} [\overline{\psi}_{jk}(x) - x_j x_k] = 0. \tag{34}$$

Therefore, we have

$$\lim_{\|u-l\|\to 0} [\psi_i(x) - \psi_i^L(x)] = 0. \tag{35}$$

The proof is completed. □

By Theorem 3, we can establish the linear programming relaxation problem (LPRP) of the QPWQC over X as follows:

$$(\text{LPRP}): \begin{cases} \min \psi_0^L(x) = \sum_{k=1}^{n} (c_k^0 x_k + \underline{\varphi}_{kk}^0(x)) + \sum_{j=1}^{n} \sum_{k=1,k\neq j}^{n} \underline{\varphi}_{jk}^0(x), \\ s.t.\ \ \psi_i^L(x) = \sum_{k=1}^{n} (c_k^i x_k + \underline{\varphi}_{kk}^i(x)) + \sum_{j=1}^{n} \sum_{k=1,k\neq j}^{n} \underline{\varphi}_{jk}^i(x) \le b_i, \quad i = 1,2,\ldots,m, \\ x \in X = \{x : l \le x \le u\}. \end{cases} \tag{36}$$

From the construction process of the former linearizing method, it is obvious that for any given X, each feasible point of the QPWQC is also feasible to the LPRP, and the optimal value of the LPRP is less than or equal to that of QPWQC. Therefore, the LPRP offers a reliable lower bound for the optimal value of the QPWQC. Except for the above approach, Theorem 3 also ensures the global convergence of the proposed algorithm.

3. New Global Optimization Algorithm

In this section, based on the former LPRP, we shall present a new global optimization algorithm for solving the QPWQC. In this algorithm, there are the following several key operations: branching, bounding, and space reduction.

Firstly, we choose a simple branching operation, which is called an interval bisection method. For any selected box $X' = [l', u'] \subseteq X^0$. Let $\delta \in \operatorname{argmax}\{u_i{}' - l_i{}' : i = 1, 2, \ldots, n\}$, subdivide $[\underline{x}_\delta{}' - \overline{x}_\delta{}']$ into $[\underline{x}_\delta{}', (\underline{x}_\delta{}' + \overline{x}_\delta{}')/2]$ and $[(\underline{x}_\delta{}' + \overline{x}_\delta{}')/2, \overline{x}_\delta{}']$, X' can be subdivide into X'^1 and X'^2. The selected branching operation is sufficient to ensure the global convergence of this algorithm.

Secondly, for each investigated sub-box $X \subseteq X^0$, we must solve the LPRP, and set $LB_s = \min\{LB(X)|X \in \Omega_s\}$, where Ω_s is still not fathomed as a sub-box set. In order to update the upper bound, we need to fathom the feasible point, and set Θ be the known feasible point set and $UB_s = \min\{\psi_0(x)|x \in \Theta\}$, to be the existent best upper bound.

In addition, we can introduce an interval reduction operation from Theorem 3 [6] to improve the convergent speed of the proposed algorithm.

3.1. Steps for Global Optimization Algorithm

For any investigated box $X^s \subseteq X^0$, let $LB(X^s)$ and $x^s = x(X^s)$ be the optimal value and optimal solution of the LPRP over X^s. Based on the branch-and-bound scheme and the former LPRP, a new global optimization algorithm is described as follows.

Algorithm Steps:

Step 1. Set $\varepsilon = 10^{-6}$, solve the (LPRP) over X^0 to obtain its optimal solution x^0 and the optimal value $LB(X^0)$, respectively.

Let the lower bound $LB_0 = LB(X^0)$. If x^0 is feasible to the QPWQC, let the upper bound be $UB_0 = \psi_0(x^0)$, otherwise let the initial upper bound be $UB_0 = +\infty$.

If $UB_0 - LB_0 \leq \varepsilon$, let the global ε– optimal solution of the QPWQC be x^0, otherwise let $\Omega_0 = \{X^0\}$, $\Lambda = \phi$, $s = 1$.

Step 2. Let the upper bound be $UB_s = UB_{s-1}$, partition X^{s-1} into $X^{s,1}$ and $X^{s,2}$, and let $\Lambda = \Lambda \cup \{X^{s-1}\}$ be the deleted sub-boxes set.

For each $X^{s,t}, t = 1, 2$, utilize the interval reduction method to compress the investigated box, and let $X^{s,t}$ be the remaining box.

For each remaining box $X^{s,t}, t = 1, 2$, solve the LPRP to obtain its optimal solution $x^{s,t}$ and optimal value $LB(X^{s,t})$, respectively.

Set $\Omega_s = \{X \big| X \in \Omega_{s-1} \cup \{X^{s,1}, X^{s,2}\}, X \notin \Lambda\}$ and $LB_s = \min\{LB(X)|X \in \Omega_s\}$.

Step 3. For each $X^{s,t}, t = 1, 2$, if x^{mid} is the feasible point of the QPWQC, let $\Theta := \Theta \cup \{x^{mid}\}$, and let the new upper bound $UB_s = \min_{x \in \Theta}\{\psi_0(x)\}$; if $x^{s,t}$ is feasible to the QPWQC, let the new upper bound $UB_s = \min\{UB_s, \psi_0(x^{s,t})\}$, and let the best known feasible point be x^s, which satisfies $UB_s = \psi_0(x^s)$.

Step 4. If $UB_s - LB_s \leq \varepsilon$, then we let the ε–global optimal solution of the QPWQC be x^s, otherwise let $s = s + 1$, and return to Step 2.

3.2. Global Convergence of the Proposed Algorithm

If the proposed algorithm terminates after finite iterations, then, when it terminates, we can obtain the global optimal solution of the QPWQC. Otherwise, the proposed algorithm will generate an infinite sequence, whose limitation is the global optimal solution of the QPWQC; the detailed proof is given as follows.

Theorem 4. *If the proposed algorithm does not terminate after finite iterations, then the proposed algorithm will generate an infinite sequence $\{X^s\}$, whose accumulation point will be the global optimal solution of the QPWQC.*

Proof. First of all, in the proposed algorithm, the selected branching method is the rectangle bisection, which is exhaustive, and which guarantees that the intervals of all variables converge to 0.

Secondly, as $\|u - l\| \to 0$, from the conclusions of Theorem 3, it follows that the LPRP will sufficiently approximate the QPWQC, which is to say, $\lim_{s \to \infty}(UB_s - LB_s) = 0$, i.e., the proposed algorithm satisfies that the bounding operation is consistent. Thirdly, in the proposed algorithm, the subdivided box which achieved the actual lower bound is immediately selected for the later partition, and the proposed algorithm satisfies that the selected operational bound is improving. By Theorem 4.3 of Reference [39], the proposed branch-and-bound algorithm satisfies the global convergent sufficient condition. Hence, the proposed algorithm converges to the global optimal solution of the QPWQC. □

4. Numerical Experiments

Let $\varepsilon = 10^{-6}$ be the convergence error. Some numerical examples in recent literature are solved in C++ program on microcomputer, and the simplex approach is employed to solve the LPRP. Compared with the existent algorithms, these numerical examples are given as follows, and their computational results are listed in Tables 1 and 2.

Example 1 (Reference [40])

$$(\mathrm{LPRP}): \begin{cases} \min \psi_0^L(x) = \sum\limits_{k=1}^{n} (c_k^0 x_k + \underline{\varphi}_{kk}^0(x)) + \sum\limits_{j=1}^{n} \sum\limits_{k=1,k\neq j}^{n} \underline{\varphi}_{jk}^0(x), \\ s.t.\ \ \psi_i^L(x) = \sum\limits_{k=1}^{n} (c_k^i x_k + \underline{\varphi}_{kk}^i(x)) + \sum\limits_{j=1}^{n} \sum\limits_{k=1,k\neq j}^{n} \underline{\varphi}_{jk}^i(x) \leq b_i, \quad i = 1, 2, \ldots, m, \\ x \in X = \{x : l \leq x \leq u\}. \end{cases}$$

$$\begin{cases} \min \psi_0(x) = x_1 \\ s.t.\ \ \psi_1(x) = \frac{1}{4}x_1 + \frac{1}{2}x_2 - \frac{1}{16}x_1^2 - \frac{1}{16}x_2^2 \leq 1, \\ \psi_2(x) = -\frac{3}{7}x_1 - \frac{3}{7}x_2 + \frac{1}{14}x_1^2 + \frac{1}{14}x_2^2 \leq -1, \\ 1 \leq x_1 \leq 5.5,\ \ 1 \leq x_2 \leq 5.5, \end{cases}$$

Example 2 (Reference [40])

$$\begin{cases} \min \psi_0(x) = x_1 x_2 - 2x_1 + x_2 + 1 \\ s.t.\ \ \psi_1(x) = -6x_1 - 16x_2 + 8x_2^2 \leq -11, \\ \psi_2(x) = 3x_1 + 2x_2 - x_2^2 \leq 7, \\ 1 \leq x_1 \leq 2.5,\ \ 1 \leq x_2 \leq 2.225. \end{cases}$$

Example 3 (References [37,41,42])

$$\begin{cases} \min \psi_0(x) = x_1^2 + x_2^2 \\ s.t.\ \ \psi_1(x) = 0.3x_1 x_2 \geq 1, \\ 2 \leq x_1 \leq 5,\ \ 1 \leq x_2 \leq 3. \end{cases}$$

Example 4 (References [41–44])

$$\begin{cases} \min \psi_0(x) = x_1 \\ s.t.\ \ \psi_1(x) = 4x_2 - 4x_1^2 \leq 1, \\ \psi_2(x) = -x_1 - x_2 \leq -1, \\ 0.01 \leq x_1, x_2 \leq 15. \end{cases}$$

Example 5 (Reference [45])

$$\begin{cases} \min \psi_0(x) = 6x_1^2 + 4x_2^2 + 5x_1 x_2 \\ s.t.\ \ \psi_1(x) = -6x_1 x_2 \leq -48, \\ 0 \leq x_1, x_2 \leq 10. \end{cases}$$

Example 6 (Reference [46])

$$\begin{cases} \min \psi_0(x) = -x_1 + x_1 x_2^{0.5} - x_2 \\ s.t. \ \ \psi_1(x) = -6x_1 + 8x_2 \le 3, \\ \psi_2(x) = 3x_1 - x_2 \le 3, \\ 1 \le x_1, x_2 \le 1.5. \end{cases}$$

Example 7 (References [26,43])

$$\begin{cases} \min \psi_0(x) = 4x_2 + (x_1 - 1)^2 + x_2 - 10x_3^2 \\ s.t. \ \ \psi_1(x) = x_1^2 + x_2^2 + x_3^2 \le 2, \\ \psi_2(x) = (x_1 - 2)^2 + x_2^2 + x_3^2 \le 2, \\ 2 - \sqrt{2} \le x_1 \le \sqrt{2}, \ \ 0 \le x_2, x_3 \le \sqrt{2}. \end{cases}$$

Table 1. Numerical comparisons for Examples 1–7.

Example	Refs.	Optimal Value	Optimal Solution	Iteration	Time (s)
1	ours	1.177124990	(1.177124344, 2.177124344)	22	0.0091
	[40]	1.177124327	(1.177124327, 2.177124353)	434	1.0000
2	ours	−0.999999202	(2.000000, 1.000000)	22	0.0085
	[40]	−1.0	(2.000000, 1.000000)	24	0.0129
3	ours	6.777809491	(2.000000000, 1.666676181)	13	0.0038
	[37]	6.777778340	(2.000000000, 1.666666667)	30	0.0068
	[41]	6.777782016	(2.000000000, 1.666666667)	40	0.0320
	[42]	6.7780	(2.00003, 1.66665)	44	0.1800
4	ours	0.500000600	(0.500000000, 0.500000000)	26	0.0061
	[41]	0.500004627	(0.5 0.5)	34	0.0560
	[42]	0.5	(0.5, 0.5)	91	0.8500
	[43]	0.500000442	(0.500000000, 0.500000000)	37	0.0193
	[44]	0.5	(0.5, 0.5)	96	1.0000
5	ours	118.381493268	(2.564162744, 3.119857633)	70	0.0435
	[45]	118.383756475	(2.5557793695, 3.1301646393)	210	0.7800
6	ours	−1.162882315	(1.499977112, 1.5)	37	0.0412
	[46]	−1.16288	(1.5, 1.5)	84	0.1257
7	ours	−11.363635682	(1.0,0.181818133, 0.983332175)	229	0.3919
	[43]	−11.363636364	(1.0,0.181818470, 0.983332113)	420	0.2845
	[26]	−10.35	(0.998712, 0.196213, 0.979216)	1648	0.3438

Table 2. Computational results for Example 8.

(n,m)	Algorithm of [47]	This Paper
	Computational Time (s)	Computational Time (s)
(4, 6)	2.37678	1.9894
(5, 11)	6.39897	4.9867
(14, 6)	9.22732	6.4567
(18, 7)	15.8410	11.6856
(20, 5)	11.9538	8.9802
(35, 10)	74.8853	56.7866
(37, 9)	77.1476	45.6324
(45, 8)	86.7174	65.6845
(46, 5)	44.2502	32.2150
(60, 11)	315.659	216.534

Comparing with the existent algorithms, numerical results show that the proposed algorithm has the higher computational efficiency.

To demonstrate robustness of the proposed algorithm, we give a large-scale random numerical example as follows.

Example 8. (Reference [47])

$$\begin{cases} \min\ \psi_0(x) = \sum\limits_{k=1}^{n} c_k^o x_k + \sum\limits_{j=1}^{n}\sum\limits_{k=1}^{n} d_{ij}^0 x_j x_k \\ s.t. \quad \psi_i(x) = \sum\limits_{k=1}^{n} c_k^i x_k + \sum\limits_{j=1}^{n}\sum\limits_{k=1}^{n} d_{ij}^i x_j x_k \le b_i,\ i = 1, 2, \ldots, m, \\ x \in X^0 = \{\, x \in R^n : \ l^0 \le x \le u^0 \}, \end{cases}$$

where $c_k^0, k = 1, 2, \ldots, n$, is randomly generated in [0, 1], $d_{kj}^0, k = 1, 2, \ldots, n, j = 1, 2, \ldots, n$, is randomly generated in [0, 1]; $c_k^i, i = 1, \ldots, m, k = 1, 2, \ldots, n$, is randomly generated in $[-1, 0]$, $d_{kj}^i, k = 1, 2, \ldots, n, j = 1, 2, \ldots, n$, is randomly generated in $[-1, 0]$, $b_i, i = 1, 2, \ldots, m$, is randomly generated in $[-300, -90]$. In the Example 8, 'n' denotes the number of variables while 'm' denotes the number of constraints. Numerical results about the Example 8 are given in the Table 2.

5. Concluding Remarks

This paper presents an effective algorithm for globally solving quadratic programs with quadratic constraints. In this algorithm, a new linearization method is constructed for deriving the linear programming relaxation problem of the QPWQC. The proposed algorithm converges to the global optimal solution of the initial problem of QPWQC, and numerical experimental results show the higher computational efficiency of the proposed algorithm.

Author Contributions: D.S., J.Y. and C.B. conceived and worked together to achieve this work.

Funding: This research was funded by the Science and Technology Project of Henan Province (192102210114, 182102310941), the Key Scientific Research Project of Universities of Henan Province (18A110019, 17A110021, 16A110013, 16A110014).

Acknowledgments: The authors would like to express their sincere thanks to the responsible editor and the anonymous referees for their valuable comments and suggestions, which have greatly improved the earlier version of our paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhu, D.; Huang, H.; Yang, S.X. Dynamic task assignment and path planning of multi-auv system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace. *IEEE Trans. Cybern.* **2012**, *43*, 504–514.
2. Jiao, H.; Liu, S. Range division and compression algorithm for quadratically constrained sum of quadratic ratios. *Comput. Appl. Math.* **2017**, *36*, 225–247. [CrossRef]
3. Cao, X.; Yu, A.L. Multi-auv cooperative target search algorithm in 3-D underwater workspace. *J. Navig.* **2017**, *53*, 1–19. [CrossRef]
4. Cao, X.; Zhu, D.Q. Multi-AUV task assignment and path planning with ocean current based on biological inspired self-organizing map and velocity synthesis algorithm. *Intell. Autom. Soft Comput.* **2017**, *23*, 31–39. [CrossRef]
5. Wieczorek, Ł.; Ignaciuk, P. Continuous Genetic Algorithms as Intelligent Assistance for Resource Distribution in Logistic Systems. *Data* **2018**, *3*, 68. [CrossRef]
6. Salami, M.; Movahedi Sobhani, F.; Ghazizadeh, M. Short-Term Forecasting of Electricity Supply and Demand by Using the Wavelet-PSO-NNs-SO Technique for Searching in Big Data of Iran's Electricity Market. *Data* **2018**, *3*, 43. [CrossRef]
7. Faris, H. A Hybrid Swarm Intelligent Neural Network Model for Customer Churn Prediction and Identifying the Influencing Factors. *Information* **2018**, *9*, 288. [CrossRef]

8. Stojčić, M.; Pamučar, D.; Mahmutagić, E.; Stević, Ž. Development of an ANFIS Model for the Optimization of a Queuing System in Warehouses. *Information* **2018**, *9*, 240. [CrossRef]
9. Lee, P.; Kang, S. An Interactive Multiobjective Optimization Approach to Supplier Selection and Order Allocation Problems Using the Concept of Desirability. *Information* **2018**, *9*, 130. [CrossRef]
10. Jain, S.; Bisht, D.C.S.; Mathpal, P.C. Particle swarm optimised fuzzy method for prediction of water table elevation fluctuation. *Int. J. Data Anal. Tech. Strateg.* **2018**, *10*, 99–110. [CrossRef]
11. Sun, H.; Tian, Y. Using improved genetic algorithm under uncertain circumstance of site selection of O2O customer returns. *Int. J. Data Anal. Tech. Strateg.* **2018**, *10*, 241–256. [CrossRef]
12. Shen, P.; Zhang, T.; Wang, C. Solving a class of generalized fractional programming problems using the feasibility of linear programs. *J. Inequal. Appl.* **2017**, *147*. [CrossRef]
13. Jiao, H.; Liu, S. An efficient algorithm for quadratic sum-of-ratios fractional programs problem. *Numer. Funct. Anal. Optim.* **2017**, *38*, 1426–1445. [CrossRef]
14. Fu, M.; Luo, Z.Q.; Ye, Y. Approximation algorithms for quadratic programming. *J. Comb. Optim.* **1998**, *2*, 29–50. [CrossRef]
15. Shen, P.; Wang, C. Linear decomposition approach for a class of nonconvex programming problems. *J. Inequal. Appl.* **2017**, *74*. [CrossRef]
16. 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. [CrossRef]
17. Jiao, H.; Chen, Y.Q.; Cheng, W.X. A Novel Optimization Method for Nonconvex Quadratically Constrained Quadratic Programs. *Abstr. Appl. Anal.* **2014**, *2014*, 698489. [CrossRef]
18. Zhao, Y.; Liu, S. Global optimization algorithm for mixed integer quadratically constrained quadratic program. *J. Comput. Appl. Math.* **2017**, *319*, 159–169. [CrossRef]
19. Jiao, H.; Chen, R. A parametric linearizing approach for quadratically inequality constrained quadratic programs. *Open Math.* **2018**. [CrossRef]
20. Shen, P.; Huang, B.; Wang, L. Range division and linearization algorithm for a class of linear ratios optimization problems. *J. Comput. Appl. Math.* **2019**, *350*, 324–342. [CrossRef]
21. Shen, P.P.; Li, X.A.; Jiao, H.W. Accelerating method of global optimization for signomial geometric programming. *J. Comput. Appl. Math.* **2008**, *214*, 66–77. [CrossRef]
22. Shen, P.; Huang, B. Global algorithm for solving linear multiplicative programming problems. *Optim. Lett.* **2019**. [CrossRef]
23. Jiao, H.; Wang, Z.; Chen, Y. Global optimization algorithm for sum of generalized polynomial ratios problem. *Appl. Math. Model.* **2013**, *37*, 187–197. [CrossRef]
24. Chen, Y.; Jiao, H. A nonisolated optimal solution of general linear multiplicative programming problems. *Comput. Oper. Res.* **2009**, *36*, 2573–2579. [CrossRef]
25. Shen, P.-P.; Lu, T. Regional division and reduction algorithm for minimizing the sum of linear fractional functions. *J. Inequal. Appl.* **2018**, *63*. [CrossRef]
26. Shen, P.; Li, X. Branch reduction bound algorithm for generalized geometric programming. *J. Glob. Optim.* **2013**, *56*, 1123–1142. [CrossRef]
27. Jiao, H.; Chen, Y.; Yin, J. Optimality condition and iterative thresholding algorithm for l(p)-regularization problems. *SpringerPlus* **2016**, *5*, 1873. [CrossRef]
28. Shen, P.; Yang, L.; Liang, Y. Range division and contraction algorithm for a class of global optimization problems. *J. Glob. Optim.* **2014**, *242*, 116–126. [CrossRef]
29. Zhao, Y.; Liu, S. An efficient method for generalized linear multiplicative programming problem with multiplicative constraints. *SpringerPlus* **2016**, *5*, 1302. [CrossRef]
30. Jiao, H.W.; Liu, S.Y.; Zhao, Y.F. Effective algorithm for solving the generalized linear multiplicative problem with generalized polynomial constraints. *Appl. Math. Model.* **2015**, *39*, 7568–7582. [CrossRef]
31. Shen, P.P.; Bai, X.D. Global optimization for generalized geometric programming problems with discrete variables. *Optimization* **2013**, *62*, 895–917. [CrossRef]
32. Jiao, H. A branch and bound algorithm for globally solving a class of nonconvex programming problems. *Nonlinear Anal. Theory Methods Appl.* **2009**, *70*, 1113–1123. [CrossRef]
33. Jiao, H.; Liu, S. A new linearization technique for minimax linear fractional programming. *Int. J. Comput. Math.* **2014**, *91*, 1730–1743. [CrossRef]

34. Jiao, H.; Guo, Y.; Shen, P. Global optimization of generalized linear fractional programming with nonlinear constraints. *Appl. Math. Comput.* **2006**, *183*, 717–728. [CrossRef]
35. Jiao, H.; Liu, S.; Yin, J.; Zhao, Y. Outcome space range reduction method for global optimization of sum of affine ratios problem. *Open Math.* **2016**, *14*, 736–746. [CrossRef]
36. Jiao, H.W.; Liu, S.Y. A practicable branch and bound algorithm for sum of linear ratios problem. *Eur. J. Oper. Res.* **2015**, *243*, 723–730. [CrossRef]
37. Jiao, H.; Liu, S.; Lu, N. A parametric linear relaxation algorithm for globally solving nonconvex quadratic programming. *Appl. Math. Comput.* **2015**, *250*, 973–985. [CrossRef]
38. Shen, P.; Jiao, H. Linearization method for a class of multiplicative programming with exponent. *Appl. Math. Comput.* **2006**, *183*, 328–336. [CrossRef]
39. Horst, R.; Tuy, H. *Global Optimization: Deterministic Approaches*, 2nd ed.; Springer: Berlin, Germany, 1993.
40. Shen, P.; Jiao, H. A new rectangle branch-and-pruning appproach for generalized geometric programming. *Appl. Math. Comput.* **2006**, *183*, 1027–1038.
41. Shen, P.; Liu, L. A global optimization approach for quadratic programs with nonconvex quadratic constraints. *Chin. J. Eng. Math.* **2008**, *25*, 923–926.
42. Wang, Y.; Liang, Z. A deterministic global optimization algorithm for generalized geometric programming. *Appl. Math. Comput.* **2005**, *168*, 722–737. [CrossRef]
43. Jiao, H.; Chen, Y. A global optimization algorithm for generalized quadratic programming. *J. Appl. Math.* **2013**, *2013*, 215312. [CrossRef]
44. Wang, Y.J.; Zhang, K.C.; Gao, Y.L. Global optimization of generalized geometric programming. *Comput. Math. Appl.* **2004**, *48*, 1505–1516. [CrossRef]
45. Gao, Y.; Shang, Y.; Zhang, L. A branch and reduce approach for solving nonconvex quadratic programming problems with quadratic constraints. *OR Trans.* **2005**, *9*, 9–20.
46. Shen, P. Linearization method of global optimization for generalized geometric programming. *Appl. Math. Comput.* **2005**, *162*, 353–370. [CrossRef]
47. Qu, S.-J.; Ji, Y.; Zhang, K.-C. A deterministic global optimization algorithm based on a linearizing method for nonconvex quadratically constrained programs. *Math. Comput. Model.* **2008**, *48*, 1737–1743. [CrossRef]

© 2019 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 (http://creativecommons.org/licenses/by/4.0/).