Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 594693, 6 pages
http://dx.doi.org/10.1155/2013/594693

Research Article

A Branch and Bound Reduced Algorithm for Quadratic Programming Problems with Quadratic Constraints

Yuelin Gao, Feifei Li, and Siqiao Jin

Institute of Information & System Science, Beifang University of Nationalities, Yinchuan 750021, China

Correspondence should be addressed to Yuelin Gao; gaoyuelin@263.net

Received 15 July 2013; Accepted 19 September 2013

Academic Editor: Dongdong Ge

We propose a branch and bound reduced algorithm for quadratic programming problems with quadratic constraints. In this algorithm, we determine the lower bound of the optimal value of original problem by constructing a linear relaxation programming problem. At the same time, in order to improve the degree of approximation and the convergence rate of acceleration, a rectangular reduction strategy is used in the algorithm. Numerical experiments show that the proposed algorithm is feasible and effective and can solve small- and medium-sized problems.

1. Introduction

Quadratic programming problems with quadratic constraints play a very important role in global optimization because quadratic functions are relatively simple functions among all nonlinear functions, and quadratic functions can approach many other functions. Therefore, it is necessary for us to research quadratic problems for researching nonlinear problems better, and quadratic programming problems with quadratic constraints have an important applications in Science and technology. Then, in spite of researching local optimization problems or global optimization problems, quadratic programming problems have got extensive attention; it is obvious that researching this kind of problems is very necessary. In this paper, we consider the following quadratic programming problems with quadratic constraints:

$$\begin{aligned} \min \quad & f^0(x) = x^T Q^0 x + \left(d^0\right)^T x + c_0, \\ \text{s.t.} \quad & f^i(x) = x^T Q^i x + \left(d^i\right)^T x + c_i \le 0, \\ & \qquad i = 1, 2, \ldots, p, \\ & x \in S = \left\{x \in R^n : l \le x \le u\right\}, \end{aligned} \tag{QP}$$

where $Q^i = (q^i_{j_1 j_2})_{n\times n}$ are n-dimension symmetric matrices, $d^i = (d^i_1, d^i_2, \ldots, d^i_n)^T \in R^n$, $l \in R^n$, $u \in R^n$, $c_i \in R$, and $i = 0, 1, \ldots, p$.

In recent years, many researchers have researched this kind of problems and made certain progress. In [1], an effective lower bound of the optimal value of original problem is provided using Lagrange lower estimate, and the local optimal solutions are obtained by Newton methods; then to accelerate the convergence of the global optimal solutions, the local Newton methods are used. A decompose-approach method is put forward in [2]. Literature [3] organically combines the outer approximation method with the branch and bound technique and presents a new branch-reduce algorithm. Literature [4] combines the cutting plane algorithm with the branch and bound algorithm, and puts forwards a new algorithm. Literature [5] presents a branch and bound algorithm by the linear lower function of the bilinear function. Based on [5], literature [6] puts forward a branch-reduce method aiming at objective function and constraint conditions of the linear relaxation programming. A simplex branch and bound algorithm is raised in [7]. There are many different methods for solving quadratic programming problems with quadratic constraints in [8–15].

The rest of this paper is organized as follows. In Section 2, we give the linear relaxation programming problem (LP) of the problem (QP). In Section 3, we give the rectangle subdivision and reduce strategy. We explain the branch and bound algorithm in detail in Section 4, and the convergence

of the algorithm is proved. Finally, some numerical results turn out the effectiveness of the present algorithm.

2. Linear Relaxation Programming

In this section, we construct a linear relaxation programming problem of the original problem.

Assume that $\lambda^i_{\min}$ is the minimum eigenvalue of the matrices Q^i, for $i = 1, 2, \ldots, p$. If $\lambda^i_{\min} \geq 0$, let $\theta_i = 0$; otherwise, let $\theta_i = |\lambda^i_{\min}| + \tau^i$, where $\tau^i \geq 0$; then $Q^i + \theta_i I$ is semipositive definite.

On the rectangle $S^k = \{x \in R^n : l^k \leq x \leq u^k\}$, for each i, we construct a linear lower function $f^i(x)$ on S^k.

We have

$$\begin{aligned} f^i(x) &= x^T Q^i x + \left(d^i\right)^T x + c_i \\ &= x^T \left(Q^i + \theta_i I\right) x + \left(d^i\right)^T x + c_i - \theta_i \|x\|^2 \\ &= \left(x - l^k\right)^T \left(Q^i + \theta_i I\right)\left(x - l^k\right) + \left(d^i\right)^T x + c_i - \theta_i \sum_{j=1}^n x_j^2 \\ &\quad + 2\left(l^k\right)^T \left(Q^i + \theta_i I\right) x - \left(l^k\right)^T \left(Q^i + \theta_i I\right) l^k. \end{aligned} \tag{1}$$

Suppose that l^k_j and u^k_j are the jth indicators of l^k and u^k, respectively. We know that, for each $j \in \{1, 2, \ldots, n\}$, a linear lower function of $-x_j^2$ is $-(u^k_j + l^k_j)x_j + u^k_j l^k_j$ on the interval $[l^k_j, u^k_j]$. Therefore,

$$\begin{aligned} \varphi_{S^k}(x) &\triangleq \sum_{j=1}^n \left(-\left(u^k_j + l^k_j\right) x_j + u^k_j l^k_j\right) \\ &= -\left(l^k + u^k\right)^T x + \left(l^k\right)^T u^k \end{aligned} \tag{2}$$

is a linear lower function of $-\sum_{j=1}^n x_j^2$ on the rectangle $[l^k, u^k]$; we construct the following linear function:

$$l^i_{S^k}(x) = \left(a^i_{S^k}\right)^T x + b^i_{S^k}, \tag{3}$$

where

$$\begin{aligned} a^i_{S^k} &= d^i + 2\left(Q^i + \theta_i I\right) l^k - \theta_i \left(l^k + u^k\right), \\ b^i_{S^k} &= c_i - \left(l^k\right)^T \left(Q^i + \theta_i I\right) l^k + \theta_i \left(l^k\right)^T u^k. \end{aligned} \tag{4}$$

We can obtain the following two theorems.

Theorem 1. *For each $i \in \{0, 1, \ldots, p\}$, let $Q^i + \theta_i I$ be semipositive definite. For each $i \in \{0, 1, \ldots, p\}$, the linear function $l^i_{S^k}(x)$ is a lower function of $f^i(x)$ on the rectangle S^k; that is $f^i(x) \geq l^i_{S^k}(x)$, for all $x \in S^k$.*

Proof. From the formula (1) and the definitions of the functions $\varphi_{S^k}(x)$ and $l^i_{S^k}(x)$, for each $i \in \{0, 1, \ldots, p\}$, we have

$$f^i(x) \geq \left(x - l^k\right)^T \left(Q^i + \theta_i I\right)\left(x - l^k\right) + l^i_{S^k}(x), \quad \forall x \in S^k. \tag{5}$$

Moreover, the matrix $Q^i + \theta_i I$ is semipositive definite; then,

$$\left(x - l^k\right)^T \left(Q^i + \theta_i I\right)\left(x - l^k\right) \geq 0, \quad \forall x \in S^k. \tag{6}$$

Consequently, $f^i(x) \geq l^i_{S^k}(x)$, for all $x \in S^k$, $i \in \{0, 1, \ldots, p\}$. □

Theorem 2. *Assume that $\rho(Q^i + \theta_i I)$ is the spectral radius of the rectangle $Q^i + \theta_i I$; then*

$$\begin{aligned} &\max\left\{\left|f^i(x) - l^i_{S^k}(x)\right| : x \in S^k\right\} \\ &\quad \leq \left(\rho\left(Q^i + \theta_i I\right) + \theta_i\right)\left\|u^k - l^k\right\|^2, \\ &\qquad i \in \{0, 1, \ldots, p\}. \end{aligned} \tag{7}$$

Proof. From the formula (1) and the definitions of the functions $\varphi_{S^k}(x)$ and $l^i_{S^k}(x)$, we have

$$\begin{aligned} &f^i(x) - l^i_{S^k}(x) \\ &\quad = \left(x - l^k\right)^T \left(Q^i + \theta_i I\right)\left(x - l^k\right) + \theta_i\left(-\|x\|^2 - \varphi_{S_k}(x)\right) \\ &\quad \leq \rho\left(Q^i + \theta_i I\right)\left\|u^k - l^k\right\|^2 + \theta_i\left|\left(x - l^k\right)^T \left(u^k - x\right)\right| \\ &\quad \leq \rho\left(Q^i + \theta_i I\right)\left\|u^k - l^k\right\|^2 + \theta_i\left\|u^k - l^k\right\|^2 \\ &\quad = \left(\rho\left(Q^i + \theta_i I\right) + \theta_i\right)\left\|u^k - l^k\right\|^2. \end{aligned} \tag{8}$$

Hence, the conclusion is established. □

Therefore, from Theorem 1, we obtain the linear relaxation programming problem of (QP) on the rectangle S^k:

$$\begin{aligned} \min \quad & l^0_{S^k}(x) \\ \text{s.t.} \quad & l^i_{S^k}(x) \leq 0, \quad i = 1, 2, \ldots, p, \\ & x \in S^k. \end{aligned} \tag{$LP\left(S^k\right)$}$$

Solving the problem $\left(LP\left(S^k\right)\right)$, its optimal value is obtained, which is a lower bound of the global optimum of the problem (QP) on the rectangle S^k.

3. The Subdivision and Reduction of the Rectangle

In this section, we give the bisection and reduction methods of the rectangle. Let $S^k = \{l^k \leq x \leq u^k\}$ be a rectangle on R^n, and $x^k \in S^k$.

3.1. The Subdivision of the Rectangle. The method of the subdivision of the rectangle is described as follows.

(i) Select the longest edge of the rectangle S^k; that is $U^k_s - L^k_s = \max\{U^k_j - L^k_j : j = 1, 2, \ldots, n\}$.

(ii) Let $V_s^k = (U_s^k + L_s^k)/2$. Then

$$S^{k1} = \prod_{j=1}^{s-1} \left[L_j^k, U_j^k\right] \times \left[L_s^k, V_s^k\right] \times \prod_{j=s+1}^{n} \left[L_j^k, U_j^k\right],$$
$$S^{k2} = \prod_{j=1}^{s-1} \left[L_j^k, U_j^k\right] \times \left[V_s^k, U_s^k\right] \times \prod_{j=s+1}^{n} \left[L_j^k, U_j^k\right]. \tag{9}$$

3.2. The Reduction of the Rectangle. Based on [8], in order to improve the convergence of the algorithm, we give two pruning methods of problem (LP). For all $S^k = \{x \in R^n : l^k \le x \le u^k\} \subseteq S$, $S_j^k = [l_j^k, u_j^k]$, suppose that the objective function of $\left(LP\left(S^k\right)\right)$ is $\varphi_0^k(x) = \sum_{j=1}^n c_j^k x_j + c_0^k$, the constraint functions are $\sum_{j=1}^n a_{ij}^k x_j \le b_i^k$, and the upper bound of (QP) is denoted by UB; let

$$rC^k = \sum_{j=1}^{n} \min\left\{c_j^k l_j^k, c_j^k u_j^k\right\}, \qquad rL_i^k = \sum_{j=1}^{n} \min\left\{a_{ij}^k l_j^k, a_{ij}^k u_j^k\right\}, \quad i = 1, \ldots, p. \tag{10}$$

Theorem 3 (see [8]). *For any $S^k \subseteq S$, if $rC^k + c_0^k > UB$, then there is no optimal solution of (QP) on S^k; otherwise, if $c_r^k > 0$ $(r \in \{1, \ldots, n\})$, then there is no optimal solution of (QP) on $\overline{S}^k = (\overline{S}_j^k)_{n\times 1}$; if $c_r^k < 0$ $(r \in \{1, \ldots, n\})$, then there is no optimal solution of (QP) on $\underline{S}^k = (\underline{S}_j^k)_{n\times 1}$, where*

$$\overline{S}_j^k = \begin{cases} S_j^k, & j \ne r,\ j = 1, \ldots, n, \\ \left(\dfrac{UB - c_0^k - rC^k + c_j^k l_j^k}{c_j^k}, u_j^k\right] \cap S_j^k, & j = r, \end{cases}$$
$$\underline{S}_j^k = \begin{cases} S_j^k, & j \ne r,\ j = 1, \ldots, n, \\ \left[l_j^k, \dfrac{UB - c_0^k - rC^k + c_j^k u_j^k}{c_j^k}\right) \cap S_j^k, & j = r. \end{cases} \tag{11}$$

Theorem 4 (see [8]). *For any $i = 1, \ldots, p$, if $rL_i^k > b_i^k$, then there is no optimal solution of (QP) on S^k; otherwise, if $a_{ir}^k > 0$ $(r \in \{1, \ldots, n\})$, then there is no optimal solution of (QP) on $\overline{\overline{S}}^k = (\overline{\overline{S}}_j^k)_{n\times 1}$; if $a_{ir}^k < 0$ $(r \in \{1, \ldots, n\})$, then there is no optimal solution of (QP) on $\underline{\underline{S}}^k = (\underline{\underline{S}}_j^k)_{n\times 1}$, where*

$$\overline{\overline{S}}_j^k = \begin{cases} S_j^k, & j \ne r,\ j = 1, \ldots, n, \\ \left(\dfrac{b_i^k - rL_i^k + a_{ij}^k l_j^k}{a_{ij}^k}, u_j^k\right] \cap S_j^k, & j = r, \end{cases}$$
$$\underline{\underline{S}}_j^k = \begin{cases} S_j^k, & j \ne r,\ j = 1, \ldots, n, \\ \left[l_j^k, \dfrac{b_i^k - rL_i^k + a_{ij}^k u_j^k}{a_{ij}^k}\right) \cap S_j^k, & j = r. \end{cases} \tag{12}$$

From Theorems 3 and 4, we can construct the following pruning rules to delete or reduce the rectangle S^k.

Rule 1. Compute rC^k, if $rC^k + c_0^k > UB$, then S^k is deleted; otherwise, for any $j = 1, \ldots, n$.

If $c_j^k > 0$, let $u_j^k = \min\{u_j^k, (UB - c_0^k - rC^k + c_j^k l_j^k)/c_j^k\}$.

If $c_j^k < 0$, let $l_j^k = \max\{l_j^k, (UB - c_0^k - rC^k + c_j^k u_j^k)/c_j^k\}$.

Rule 2. Compute rL_i^k, if $rL_i^k > b_i^k$, then S^k is deleted; otherwise, for any $j = 1, \ldots, n$.

If $a_{ij}^k > 0$, let $u_j^k = \min\{u_j^k, (b_i^k - rL_i^k + a_{ij}^k l_j^k)/a_{ij}^k\}$.

If $a_{ij}^k < 0$, let $l_j^k = \max\{l_j^k, (b_i^k - rL_i^k + a_{ij}^k u_j^k)/a_{ij}^k\}$, where $i = 1, \ldots, p$.

4. The Algorithm Description and Convergence Analysis

Next, we can describe a branch and bound reduced algorithm of problem (QP) as follows.

Suppose when the iteration proceeds in step k, the feasible region of the problem (QP) is denoted by D, Q represents the feasible set at present, S^k represents the divided rectangle soon, the set of remained rectangle after pruning is denoted by T, and the current lower bound and upper bound of the global optimal value of the problem (QP) are denoted by α_k and β_k, respectively.

Step 1 (initializing). Set $\varepsilon > 0$, and let $T = \{S\}$, $k = 1$, $S^k = S$, and $\beta_k = \infty$. Solving the problem $\left(LP\left(S^k\right)\right)$, its optimal solution and optimal value are denoted by x^k and $\beta(S^k)$, respectively. Let $\beta_k = \beta(S^k)$; then β_k is a lower bound of global optimal value of the problem (QP); if $x^k \in D$, let $Q = Q \cup \{x^k\}$, the upper bound is $Q = Q \cup \{x^k\}$, and find a current optimal solution $x^* \in \arg\min \alpha_k$.

Step 2 (termination rule). If there was a condition satisfying between $\alpha_k - \beta_k \le \varepsilon$ $(k = 1, 2, \ldots)$ or $T = \emptyset$, then stop; the global optimal solution x^* and the global optimal value $f^0(x^*)$ are outputted; otherwise, go to the next step.

Step 3 (selection rule). Select a rectangle which has a minimum lower bound in the rectangle set T; that is, $S^k = \arg\min \beta_k$.

Step 4 (subdivision rule). Using the subdivision method in the former section, then the rectangle S^k can be divided into subrectangles S^{k1} and S^{k2}, and $\operatorname{int} S^{k1} \cap \operatorname{int} S^{k2} = \emptyset$.

Step 5 (reduction technique). Reducing the subrectangles after dividing using the reduction method in the former

section, without loss of generality, the new rectangles after reduction are also denoted by S^{kj}, $j \in \Gamma$, where Γ is the index set of the rectangles after reduction.

Step 6 (bounding rule). Lower bound is $\beta_k^* = \min\{\beta_k : k = 1, 2, \ldots\}$; upper bound is $\alpha_k^* = \min\{f^0(x), x \in Q\}$.

The current best feasible solution is $x^* \in \arg\min\{f^0(x) : x \in Q\}$.

Step 7 (pruning rule). Let $T = T \setminus \{S : \overline{\beta}_k(S) \geq \alpha_k^*, S \in T\}$.

Step 8. Set $k = k + 1$; go to Step 2.

Theorem 5. *(a) If the algorithm terminates in limited steps, then x^k is a ε-global optimal solution of problem (QP).*

(b) For each $k \geq 1$, let x^k be the solution after step k. If the algorithm is infinite, then $\{x^k\}$ is a feasible solution sequence of problem (QP), and any accumulation is a global optimal solution of problem (QP), and $\lim_{k\to\infty}\alpha_k = \lim_{k\to\infty}\beta_k = v$.

Proof. (a) If the algorithm is finite, suppose that it terminates in step k $(k \geq 1)$. Because x^k is obtained by solving $\left(LP\left(S^k\right)\right)$, then $x^k \in S^k \subseteq S$, and x^k is a feasible solution of problem (QP). When $\alpha_k - \beta_k \leq \varepsilon$, the algorithm terminate. From Steps 1 and 6, we have $f^0(x^k) - \beta_k \leq \varepsilon$; from the algorithm, $\beta_k \leq v$, where v is the global optimal value of problem (QP). Because x^k is a feasible solution of problem (QP), so $f^0(x^k) \geq v$. Thus

$$v \leq f^0\left(x^k\right) \leq v + \varepsilon. \tag{13}$$

Therefore, x^k is a ε-global optimal solution of problem (QP).

(b) If the algorithm is infinite, then it produces a solution sequence $\{x^k\}$ of problem (QP), where for each $k \geq 1$, x^k is obtained by solving problem $\left(LP\left(S^k\right)\right)$. For each $S^k \subseteq S$, for the optimal solution $x^k \in S^k \subseteq S$, the sequence $\{x^k\}$ constitute a solution sequence of problem (QP); from the iteration of the algorithm, we have

$$\beta_k \leq v \leq \alpha_k = f^0\left(x^k\right), \quad k = 1, 2, \ldots. \tag{14}$$

Because the series $\{\beta_k\}$ do not decrease and have an upper bound, and $\{\alpha_k\}$ do not increase and have a lower bound, then the series $\{\beta_k\}$ and $\{\alpha_k\}$ are both convergent. Taking the limits on both sides of (14), we have

$$\lim_{k\to\infty}\beta_k \leq v \leq \lim_{k\to\infty}\alpha_k = \lim_{k\to\infty} f^0\left(x^k\right). \tag{15}$$

Let $\lim_{k\to\infty}\beta_k = \overline{\beta}$, $\lim_{k\to\infty}\alpha_k = \overline{\alpha}$; then the formula (15) converts into

$$\overline{\beta} \leq v \leq \lim_{k\to\infty} f^0\left(x^k\right) = \overline{\alpha}. \tag{16}$$

Without loss of generality, assume that the sequence of rectangle $\{S^k = [l^k, u^k]\}$ satisfy $x^k \in S^k$ and $S^{k+1} \subset S^k$. In our algorithm, the rectangles are divided into two equal parts continuously; then $\bigcap_{k=1}^{\infty} S^{k+1} = \{x^k\}$, and because of the continuity of function $f^0(x)$,

$$\overline{\beta} = v = \overline{\alpha} = \lim_{k\to\infty} f^0\left(x^k\right) = f^0\left(x^*\right), \tag{17}$$

So any accumulation of $\{x^k\}$ is a global optimal solution of problem (QP). □

5. Numerical Experiment

Several experiments are given to turn out the feasibility and effectiveness of our algorithm.

Example 1.

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 \\ \text{s.t.} \quad & 0.3x_1x_2 \geq 1, \\ & 2 \leq x_1 \leq 5, \\ & 1 \leq x_2 \leq 3. \end{aligned} \tag{18}$$

From the algorithm, the initial rectangle is $S^1 = \left[\begin{smallmatrix} 2.0000 & 5.0000 \\ 1.0000 & 3.0000 \end{smallmatrix}\right]$; first, we solve the problem $LP(S^1)$, its optimal solution is $x^1 = (2.0000; 3.0000)$, and optimal value is $\beta_1 = \beta(S^1) = 4.9996$; then 4.9996 is a lower bound of the global optimal value of problem (QP). Because $x^1 = (2.0000; 3.0000)$ is feasible, then $Q = [2.0000; 3.0000]$ is a set of current feasible solutions, and the current upper bound is $\alpha_1 = f^0(x^1) = 13.0000$; the current optimal solution is $x^* = x^1 = (2.0000; 3.0000)$.

After that, based on our selection rule, select the rectangle with the minimum lower bound S^1 to divide; then S^1 is divided into two subrectangles $S^{1,1} = \left[\begin{smallmatrix} 2.0000 & 3.5000 \\ 1.0000 & 3.0000 \end{smallmatrix}\right]$ and $S^{1,2} = \left[\begin{smallmatrix} 3.5000 & 5.0000 \\ 1.0000 & 3.0000 \end{smallmatrix}\right]$ from the dividing method in Section 3.1; then reduce the rectangles using the reduction technique in Section 3.2, and the new rectangle after reduction is denoted by $S^2 = S^{1,1} = \left[\begin{smallmatrix} 2.0000 & 3.5000 \\ 1.0000 & 3.0000 \end{smallmatrix}\right]$. Solving the linear relaxation programming problem LP on the rectangle S^2, its optimal value is $\beta_2 = \beta(S^{1,1}) = 4.9996$; then the lower bound of the original problem is not updated, also being 4.9996. Next, we choose S^2 to divide, until $\cdots$ the 15th iteration, $S^{14} = \left[\begin{smallmatrix} 2.0000 & 2.0408 \\ 1.6538 & 1.7019 \end{smallmatrix}\right]$; solve the linear relaxation programming problem $LP(S^{14})$, its optimal solution is $(2.0000; 1.6665)$, and optimal value is 6.7765; while, the current upper bound is 6.8151, the current optimal solution is $(2.0000; 1.6778)$. Because $|6.8151 - 6.7765| < 0.1$, it satisfies our termination rule; then the optimal value of the original problem is 6.8151, the lower bound of the optimal value is 6.7765, and the optimal solution is $x = (2.0000; 1.6778)$; here the lower bound of the optimal value is also approximate optimal value, where the accuracy is $\varepsilon = 0.1$.

Table 2 shows the different results of Example 1 under different accuracy.

Table 1

Example	The optimal solution within accuracy or one solution among solutions				
	x_1	x_2	x_3	x_4	x_5
1	2.0000	1.6667			
2	2.5576	3.1279			
3	2.0000	1.6667			
4	1.0000	5.5000			
5	1.5000	1.2247			
6	1.0156	1.5594			
7	0.4267	0.5879			
8	78.0000	33.0001	29.9958	44.9998	36.7753

Example	Approximate optimal value	Iterations	CPU (s)
1	6.7778	33	8.301128
2	118.3837	49	32.696565
3	6.7778	29	6.676444
4	1.0000	9	3.162106
5	−1.1629	17	4.429806
6	−31.8878	130	54.024140
7	−3.3304	20	5.410943
8	10128	98	193.921992

Table 2: Different results of Example 1 under different accuracy.

	Example 1	
ε	Approximate optimal value	Optimal value
$1.0e-2$	6.777772334392922	6.784953802104409
$1.0e-3$	6.777777695638590	6.778685210977349
$1.0e-4$	6.777777810403491	6.777777840618791

Example 2.

$$\begin{aligned} \min\ & 6x_1^2 + 4x_2^2 + 5x_1x_2 \\ \text{s.t.}\ & -6x_1x_2 \le -48, \\ & 0 \le x_1,\ x_2 \le 10. \end{aligned} \tag{19}$$

The optimal value is 118.3838.

Example 3.

$$\begin{aligned} \min\ & x_1^2 + x_2^2 \\ \text{s.t.}\ & -0.3x_1x_2 \le -1, \\ & -x_1 - x_2 \le 1, \\ & x \in X^0 = \{2 \le x_1 \le 5, 1 \le x_2 \le 3\}. \end{aligned} \tag{20}$$

The optimal value is 6.7778.

Example 4.

$$\begin{aligned} \min\ & x_1 \\ \text{s.t.}\ & \frac{1}{4}x_1 + \frac{1}{2}x_2 - \frac{1}{6}x_2^2 - \frac{1}{6}x_1^2 \le 1, \\ & \frac{1}{14}x_1^2 + \frac{1}{14}x_2^2 - \frac{3}{7}x_1 - \frac{3}{7}x_2 \le 1, \\ & 1 \le x_1 \le 5.5,\ 1 \le x_2 \le 5.5. \end{aligned} \tag{21}$$

The optimal value is 1.0000.

Example 5.

$$\begin{aligned} \min\ & -x_1 + x_1x_2^{0.5} - x_2 \\ \text{s.t.}\ & -6x_1 + 8x_2 \le 3, \\ & 3x_1 - x_2 \le 3, \\ & x \in X^0 = \{x \mid 0 \le x_i \le 1.5,\ i = 1, 2\}. \end{aligned} \tag{22}$$

The optimal value is −1.1629.

Example 6.

$$\begin{aligned} \min\ & 6x_1^2 + 4x_2^2 + 2.5(x_1 + x_2)^2 - 2.5(10x_1 + 10x_2) \\ \text{s.t.}\ & 3(x_1 - x_2)^2 - 3(10x_1 + 10x_2) \le -48, \\ & 0 \le x_1,\ x_2 \le 10. \end{aligned} \tag{23}$$

The optimal value is −31.8878.

Example 7.

$$\begin{aligned} \min\ & 21x_1^2 + 34x_1x_2 - 24x_2^2 + 2x_1 - 14x_2 \\ \text{s.t.}\ & 2x_1^2 + 4x_1x_2 + 2x_2^2 + 8x_1 + 6x_2 - 9 \le 0, \\ & -5x_1^2 - 8x_1x_2 - 5x_2^2 - 4x_1 + 4x_2 + 4 \le 0, \\ & x_1 + 2x_2 \le 2,\ x \in [0, 1]^2. \end{aligned} \tag{24}$$

The optimal value is −3.3205.

Example 8.

$$\begin{aligned}
\min\ & 5.3578x_3^2 + 0.8357x_1x_5 + 37.2392x_1 \\
\text{s.t.}\ & 2.584\times10^{-5}x_3x_5 - 6.663\times10^{-3}x_2x_5 \\
& \quad -7.34\times10^{-5}x_1x_4 \le 1, \\
& 8.53007\times10^{-4}x_2x_5 + 9.395\times10^{-5}x_1x_4 \\
& \quad -3.3.85\times10^{-4}x_3x_5 \le 1, \\
& \quad -x_2x_5 - 0.42x_1x_2 - 0.30586x_3^2 \\
& \le -1.3303294\times10^3, \\
& \quad -x_3x_5 - 0.2668x_1x_3 - 0.40584x_3x_4 \\
& \le -2.2751327\times10^3, \\
& 2.4186\times10^{-4}x_2x_5 + 1.0159\times10^{-4}x_1x_2 \\
& \quad +7.379\times10^{-5}x_3^2 \le 1, \\
& 2.9955\times10^{-4}x_3x_5 + 7.992\times10^{-5}x_1x_3 \\
& \quad +1.2157\times10^{-4}x_3x_4 \le 1, \\
& x \in X^0 = \{x \mid 78 \le x_1 \le 102, 33 \le x_2 \le 45, 27 \\
& \qquad \le x_i \le 45, i = 3,4,5\}.
\end{aligned} \tag{25}$$

The optimal value is 1.0128×10^4.

We choose $\varepsilon = 1.0\varepsilon - 4$; then the approximate optimal value satisfying accuracy and the CPU running time are obtained; the results are shown in Table 1.

6. Conclusion

In this paper, we presented a branch and bound reduced algorithm for solving the quadratic programming problems with quadratic constraints. By constructing a linear relaxation programming problem, the lower bound of the optimal value of original problem can be obtained. Meanwhile, we used a rectangle reduction technique to improve the degree of approximation and the convergence rate of acceleration. Numerical experiments show the effectiveness of our algorithm.

Acknowledgment

The work is supported by the Foundation of National Natural Science of China under Grant no. 11161001.

References

[1] T. Van Voorhis, "A global optimization algorithm using Lagrangian underestimates and the interval Newton method," *Journal of Global Optimization*, vol. 24, no. 3, pp. 349–370, 2002.

[2] X. J. Zheng, X. L. Sun, and D. Li, "Convex relaxations for nonconvex quadratically constrained quadratic programming: matrix cone decomposition and polyhedral approximation," *Mathematical Programming B*, vol. 129, no. 2, pp. 301–329, 2011.

[3] Y. Gao, H. Xue, and P. Shen, "A new rectangle branch-and-reduce approach for solving nonconvex quadratic programming problems," *Applied Mathematics and Computation*, vol. 168, no. 2, pp. 1409–1418, 2005.

[4] C. Audet, P. Hansen, B. Jaumard, and G. Savard, "A branch and cut algorithm for nonconvex quadratically constrained quadratic programming," *Mathematical Programming A*, vol. 87, no. 1, pp. 131–152, 2000.

[5] S.-J. Qu, Y. Ji, and K.-C. Zhang, "A deterministic global optimization algorithm based on a linearizing method for nonconvex quadratically constrained programs," *Mathematical and Computer Modelling*, vol. 48, no. 11-12, pp. 1737–1743, 2008.

[6] H. Wu and K. Zhang, "A new accelerating method for global non-convex quadratic optimization with non-convex quadratic constraints," *Applied Mathematics and Computation*, vol. 197, no. 2, pp. 810–818, 2008.

[7] J. Linderoth, "A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs," *Mathematical Programming B*, vol. 103, no. 2, pp. 251–282, 2005.

[8] H. Tuy and N. T. Hoai-Phuong, "A robust algorithm for quadratic optimization under quadratic constraints," *Journal of Global Optimization*, vol. 37, no. 4, pp. 557–569, 2007.

[9] X. L. Sun, J. L. Li, and H. Z. Luo, "Convex relaxation and Lagrangian decomposition for indefinite integer quadratic programming," *Optimization*, vol. 59, no. 5-6, pp. 627–641, 2010.

[10] M. Salahi, "Convex optimization approach to a single quadratically constrained quadratic minimization problem," *Central European Journal of Operations Research (CEJOR)*, vol. 18, no. 2, pp. 181–187, 2010.

[11] X. J. Zheng, X. L. Sun, and D. Li, "Nonconvex quadratically constrained quadratic programming: best D.C. decompositions and their SDP representations," *Journal of Global Optimization*, vol. 50, no. 4, pp. 695–712, 2011.

[12] X. Bao, N. V. Sahinidis, and M. Tawarmalani, "Semidefinite relaxations for quadratically constrained quadratic programming: a review and comparisons," *Mathematical Programming B*, vol. 129, no. 1, pp. 129–157, 2011.

[13] D. S. Kim, N. N. Tam, and N. D. Yen, "Solution existence and stability of quadratically constrained convex quadratic programs," *Optimization Letters*, vol. 6, no. 2, pp. 363–373, 2012.

[14] S. Burer and H. Dong, "Representing quadratically constrained quadratic programs as generalized copositive programs," *Operations Research Letters*, vol. 40, no. 3, pp. 203–206, 2012.

[15] X. J. Zheng, X. L. Sun, D. Li, and Y. F. Xu, "On zero duality gap in nonconvex quadratic programming problems," *Journal of Global Optimization*, vol. 52, no. 2, pp. 229–242, 2012.