OPSEARCH (2018) 55:320–336
https://doi.org/10.1007/s12597-018-0334-0

THEORETICAL ARTICLE

On solving quadratically constrained quadratic programming problem with one non-convex constraint

Mohammad Keyanpour[1] · Naser Osmanpour[1]

Accepted: 14 February 2018 / Published online: 3 March 2018

Abstract In this paper we consider a quadratically constrained quadratic programming problem with convex objective function and many constraints in which only one of them is non-convex. This problem is transformed to a parametric quadratic programming problem without any non-convex constraint and then by solving the parametric problem via an iterative scheme and updating the parameter in each iteration, the solution of the problem is achieved. The convergence of the proposed method is investigated. Numerical examples are given to show the applicability of the new method.

Keywords Adaptive ellipsoid based method · Quadratically constrained quadratic programming · Non-convex constraint · Semidefinite programming

1 Introduction

In this paper, we study the quadratically constrained quadratic programming (QCQP) problem as follows:

$$\text{(QCQP)} \quad \begin{array}{l} \min\limits_{x \in R^n} f(x) \\ g(x) \leq 0 \\ h_j(x) \leq 0 \quad j = 1, 2, \ldots, m. \end{array} \tag{1.1}$$

where $f(x) = x^T Qx + 2p^T x + r$ is a coercive function, $g(x) = x^T Q_0 x + 2p_0^T x + r_0$

✉ Mohammad Keyanpour
kianpour@guilan.ac.ir

Naser Osmanpour
naser.osmanpour@gmail.com

[1] Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

and $h_j(x) = x^T Q_j x + 2p_j^T x + r_j$ for $j = 1, 2, \ldots, m$. Q and Q_j are $n \times n$ real symmetric matrices, p and p_j are real vectors in $\mathbb{R}^n$ and $\mathrm{c}, \mathrm{c_j} \in \mathbb{R}$ are real numbers. QCQP appears in many areas of science and engineering such as wireless communications and networking, [7], radar [17], and signal processing [10]. Some important subclasses of QCQP problem are also the trust region problem [1], Max-Cut problem, 0–1 quadratic programming problem and box constrained quadratic programming problem [13]. It is well known that QCQP is NP-Hard, it means that this problem cannot be solved in polynomial time unless P = NP [6].

If all of the matrices $Q, Q_0, Q_1, \ldots, Q_m$ are positive semidefinite, problem (1.1) becomes a convex programming problem which is solvable in polynomial time (within a given precision level) using the semidefinite programming methods [13]. If at least one of them isn't positive semidefinite then finding the global optimal solution of QCQP is much more difficult and in general this is an open problem. In this article we will consider the case of QCQP problem (1.1) in which one non-convex constraint $g(x) \leq 0$ exists. The feasible set for this case may even be non-convex.

Having a non-convex constraint in programming is first studied by Rosen in [18]. Rosen studied the problem of linear programs with an additional reverse convex constraint; and Hillestad and Jacobsen in [9] gave characterizations of optimal solutions and proposed an algorithm based on optimality properties (see also [11, 12]).

A non-convex quadratic programming problem with two quadratic constraints is also studied in [21] and using the semidefinite programming (SDP) approach a family of solvable subclasses of non-convex quadratic programming problem is identified. In [20] a non-convex quadratic programming problem is also reformulated into a linear conic programming problem and based on the matrix decomposition method and SDP techniques, some polynomial-time solvable subclasses of QCQP are identified.

Lu et al. [14] have proposed a new method for reformulating QCQP into a relaxation linear conic programming and solved the problem by using the SDP approach. To find a global optimal solution, they investigated the relationship between its Lagrangian multipliers and related linear conic programming problem. In order to apply this method, some conditions should be imposed on QCQP problem such as semidefinite condition. So we cannot use this method for any QCQP problem.

The copositive representation of a quadratic programming problem is another approach to find an approximate solution [14]. In [22] Shi and Jin presented the optimality and copositivity conditions for determining whether or not a given KKT solution is globally optimal and then proposed a local search based scheme to find the global optimal solution. But the limitation of this method is that it is not also convenient for any QCQP problems and it could find only a lower bound of optimal value of objective function.

The main contribution of this paper is to propose a numerical procedure to find a global optimal solution of the non-convex QCQP problem (1.1) with one non-convex constraint and probably many convex constraints. At first, the problem (1.1) is changed to an equivalent parametric problem. Then by finding the minimizer of parametric problem and updating the parameter, an iterative method called Parametric QCQP method (PQCQP) is proposed.

This work was intended as an attempt to motivate the use of adaptive ellipsoid based method for solving a class of non-convex QCQP problems. One of the advantages of this method is to determine the infeasibility of problem (1.1). The new method is an iterative method which combines a bisection scheme, adaptive ellipsoid based method, and SDP relaxation.

The rest of this paper is arranged as follows: In Sect. 2 we provide detail of PQCQP method and present a new algorithm. Section 3 contains convergence of the new method and more theoretical results. The computational results on randomly generated problem are reported in Sect. 4. Conclusions are given in Sect. 5.

The following notations are adopted in this paper. Let S^n denote the set of real symmetric matrices of size n, and $\mathrm{S}^{\mathrm{n}}_{+}$ the set of positive semidefinite matrices of size n, $\mathrm{S}^{\mathrm{n}}_{++}$ the set of positive definite matrices. Given a vector $x \in \mathbb{R}^n$, x_i denotes the ith entry of x. For a matrix Y, $\mathrm{Y_{ij}}$ denotes the (i,j)th entry of Y. For any two matrices $M = (M_{ij})$ and $N = (N_{ij})$ in S^n, the inner product of these two matrices is defined by $M * N = \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} N_{ij}$.

2 New method for solving QCQP

Consider problem (1.1). If all constraints and objective function are convex, then we can solve it by using the second order cone programming efficiently [8]. If at least one of the constraints is non-convex then solving problem (1.1) is difficult, because feasible set may be non-convex. We consider the case of QCQP problems with one non-convex constraint $g(x) \leq 0$. The epigraph form of problem (1.1) is as follows

$$\begin{aligned} \alpha^* = \min_{x,\alpha} \alpha & \\ f(x) &\leq \alpha \\ g(x) &\leq 0 \\ h_j(x) &\leq 0 \quad j = 1, 2, \ldots, m. \end{aligned} \tag{2.1}$$

Let $\Lambda(\alpha) = \{x | f(x) \leq \alpha,\ g(x) \leq 0,\ h_j(x) \leq 0,\ j = 1, 2, \ldots, m\}$. The following lemma concerns the relationship between problems (1.1) and (1.2).

Lemma 1 *Consider problem* (1.1) *and* (2.1)*, we have:*

1. *Problem* (1.1) *is feasible if and only if* $\Lambda(\alpha)$ *is a nonempty set for some* $\alpha \in \mathbb{R}$.
2. *Problem* (1.1) *is solvable if* $\Lambda(\alpha)$ *is a nonempty compact set for some* $\alpha \in \mathbb{R}$.

Proof

1. Let $\Lambda = \{x | g(x) \leq 0,\ h_j(x) \leq 0,\ j = 1, 2, \ldots, m\}$. First suppose that $\Lambda(\alpha)$ is a nonempty set for some $\alpha \in \mathbb{R}$. Since $\Lambda(\alpha) \subseteq \Lambda$, the proof is evident.
   Conversely, assume that problem (1.1) is feasible and $\bar{x}$ is a feasible solution of problem (1.1). we take $\alpha = f(\bar{x})$. Thus $\Lambda(\alpha)$ is a nonempty set.

2. Take $\alpha \in \mathbb{R}$ such that $\Lambda(\alpha)$ is a nonempty compact set. Since $\Lambda(\alpha)$ is compact set and the objective function f is continuous, so by Weierstrass theorem there exists a minimizing point $\bar{x} \in \Lambda(\alpha)$ for the problem $\min_{x \in \Lambda(\alpha)} f(x)$. We claim that $\bar{x}$ is the optimal solutions of problem (1.1), thus solvability is established. To prove this claim, we assume, on contrary, that there exists a point $\hat{x} \in \Lambda \backslash \Lambda(\alpha)$, such that $f(\hat{x}) < f(\bar{x})$. Since $f(\bar{x}) < \alpha$, we have $f(\hat{x}) < \alpha$ contradicting the assumption that $\hat{x} \notin \Lambda(\alpha)$. □

Now we introduce a new function as below:

$$\begin{aligned} \mathrm{F}(\alpha) = \min_{x \in R^n} & \, g(x) \\ \text{s.t.} \quad & f(x) \leq \alpha \quad j = 1, 2, \ldots, m \end{aligned} \tag{2.2}$$

It is seen that F(α) can be computed by solving the QCQP problem in which the objective function is non-convex and the constraints are convex.

One see that $\mathrm{F}(\alpha) \leq 0$ if and only if $\Lambda(\alpha) \neq \emptyset$. Therefore we have:

$$\begin{aligned} \alpha^* = \min_{\alpha} & \, \alpha \\ & F(\alpha) \leq 0 \end{aligned}$$

Let an interval $[\alpha_1, \alpha_2]$ including the optimal value α^* be given. We call $[\alpha_1, \alpha_2]$ trust interval. Here a question arises how to determine a trust interval? To answer this question, we compute endpoints of a trust interval as follows:

Right endpoint For finding the right endpoint, we need to a feasible solution x_0. Let $\alpha_2 = f(x_0)$ as a right endpoint of trust interval. Since x_0 is a feasible solution for problem (1.1), so $\alpha^* \leq \alpha_2$.

To obtain a feasible solution x_0, consider the following problem:

$$\begin{aligned} g^* = \min_{x \in R^n} & \, g(x) \\ & h_j(x) \leq 0 \quad j = 1, 2, \ldots, m. \end{aligned} \tag{2.3}$$

For solving non-convex problem (2.3), we use AEA method which will be discussed later. We take x_0, the optimal solution of problem (2.3). In the next theorem, we state the relationship between the problems (1.1) and (2.3).

Theorem 2 *The problem* (1.1) *is feasible if and only if* $g^* \leq 0$.

Proof Suppose that the problem (1.1) is feasible. So

$$\exists \bar{x} \in \mathbb{R}^n : g(\bar{x}) \leq 0, \quad h_j(\bar{x}) \leq 0 \quad for\ j = 1, 2, \ldots, m.$$

Therefore we have $g^* \leq g(\bar{x}) \leq 0$.

Conversely, let x^* be an optimal solution of problem (2.3) and $g^* \leq 0$. We have

$$g(x^*) \leq 0, \quad h_j(x^*) \leq 0 \quad for\ j = 1, 2, \ldots, m.$$

So the problem (1.1) is feasible. □

Corollary 3 *The problem (1.1) is infeasible if and only if* $g^* > 0$.

Left endpoint Consider the following problem equivalent to problem (1.1):

$$\begin{aligned} &\min Q * X + 2p^T x + r \\ &\quad Q_0 * X + 2p_0^T x + r_0 \le 0 \\ &\quad Q_j * X + 2p_j^T x + r_j \le 0 \qquad j = 1, 2, \ldots, m \\ &\quad X = xx^T \succeq 0 \end{aligned} \tag{2.4}$$

The last constraint i.e. $X = xx^T$ is non-convex. We can directly relax it to the convex constraint $X \succeq xx^T$, therefore the convex relaxation of problem (2.4) is as follows:

$$\begin{aligned} &\min Q * X + 2p^T x + r \\ &\qquad Q_0 * X + 2p_0^T x + r_0 \le 0 \\ &\qquad Q_j * X + 2p_j^T x + r_j \le 0 \\ (\text{SDPR}) &\qquad X \succeq xx^T \qquad j = 1, 2, \ldots, m \\ &\qquad X \succeq 0 \end{aligned} \tag{2.5}$$

Problem (2.5) is called SDP Relaxation problem (SDPR).

Lemma 4 ([4]) *Let* $V(QCQP)$ *and* $V(SDPR)$ *denote the optimal values of optimization problems (1.1) and (2.5) respectively. Then we have* $V(SDPR) \le V(QCQP)$.

Let Y^* be an optimal solution of problem (2.5) then $\alpha_1 = H_0 Y^*$ is a left endpoint of trust interval. According to Lemma 2, we have $\alpha_1 \le \alpha^*$.

So we could find a trust interval $[\alpha_1, \alpha_2]$. Now a question arises: how to tighten this trust interval? For answering this question, we propose a method that turns out to be similar to the bisection method. At each step, the method divides the interval in two by computing the midpoint $\alpha_3 = \frac{(\alpha_1 + \alpha_2)}{2}$ of the interval and the value of the function $F(\alpha_3)$. Therefore we have two intervals $[\alpha_1, \alpha_3]$ and $[\alpha_3, \alpha_2]$. If $F(\alpha_3) \le 0$, we select the subinterval $[\alpha_1, \alpha_3]$, otherwise we select $[\alpha_3, \alpha_2]$.

According to above discussion, the proposed method is as follows:

Algorithm1 (PQCCP):

Initialization. Given $\varepsilon > 0$.

Step 1. Determine α_1 by calculating optimal value of problem (2.5).

Step 2. Solve problem (2.3) and set $\alpha_2 = f(x_0)$. If $g^* > 0$, then stop; problem (1.1) is infeasible.

Step 3. Let $\alpha_3 = \frac{(\alpha_1 + \alpha_2)}{2}$.

Step 4. Solve problem (2.2) with $\alpha = \alpha_3$ to compute $F(\alpha_3)$.

Step 5. If $F(\alpha_3) \le 0$ then set $\alpha_2 = \alpha_3$, otherwise set $\alpha_1 = \alpha_3$.

Step 6. If $|\alpha_2 - \alpha_1| < \varepsilon$ then terminate, otherwise go to Step 3.

In Step 4 for calculating $F(\alpha_3)$, we should solve problem (2.2). We use the adaptive ellipsoid based method (AEA) [4] in which its details are given in the following.

We recall that $\Lambda(\alpha) = \{x \in R^n | f(x) \leq \alpha,\ h_j(x) \leq 0,\ j = 1, \ldots, m\}$ is a subset of the feasible set of problem (1.1), that α is obtained from Algorithm 1. For the remainder of this section, we will assume that the parameter α is given. For simplicity of notation, we denote $\Lambda(\alpha)$ briefly by Λ. According to [16], problem (2.2) is equivalent to the following linear conic programming problem:

$$\text{(CP)} \quad \begin{array}{l} \min H' * Y \\ Y_{11} = 1 \\ Y \in D^*_\Lambda \end{array} \tag{2.6}$$

where $H' = [0\, p^T; p\, Q]$ and $D^*_\Lambda = cone\left\{ Y \in S^{n+1} \middle| Y = \begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}^T \text{ for some } x \in \Lambda \right\}$.

The cone of nonnegative quadratic functions D_ξ over a union of ellipsoids $\xi \subseteq R^n$ and its dual cone D^*_ξ is introduced in [4], such that $S^{n+1}_+ \subseteq D_\xi \subseteq D_\Lambda$ and $D^*_\Lambda \subseteq D^*_\xi \subseteq S^{n+1}_+$. A tighter lower bound could be obtained, By replacing the untractable cone D^*_Λ in problem (CP) with the tractable cone D^*_ξ instead of S^{n+1}_+.

Let $\mathcal{F} = \{\mathcal{F}^1_e, \ldots, \mathcal{F}^k_e\}$ be a collection of full-dimensional ellipsoids, where

$$\mathcal{F}^j_e = \left\{ x \in R^n \middle| x^T A_j x + 2b_j^T x + c_j \leq 0 \right\} \tag{2.7}$$

where $A_j \in S^n_{++}$, $b_j \in R^n$ and $c_j \in R$, for $j = 1, \ldots, k$. Let $F = \bigcup_{j=1}^k \mathcal{F}^j_e$. So F is an ellipsoidal cover of Λ, if $\Lambda \subseteq F$. From [2] each cone $D_{\mathcal{F}^j_e}$ has an LMI representation. The cone of nonnegative quadratic functions over F and its dual cone are defined as follows:

$$D_F = \left\{ U \in S^{n+1} \middle| \begin{bmatrix} 1 \\ x \end{bmatrix}^T U \begin{bmatrix} 1 \\ x \end{bmatrix} \geq 0 \quad \text{for all } x \in F \right\} \tag{2.8}$$

$$D^*_F = cone\left\{ Y \in S^{n+1} \middle| Y = \begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}^T \quad \textit{for some } x \in F \right\} \tag{2.9}$$

Theorem 5 ([15])

1. *If $\Lambda \subseteq F$, for some $\alpha \in \mathbb{R}$, then $D_F \subseteq D_\Lambda$ and $D^*_\Lambda \subseteq D^*_F$.*
2. *If $F = \bigcup_{j=1}^k \mathcal{F}^j_e$, then $D_F = \bigcap_{j=1}^k D_{\mathcal{F}^j_e}$ and $D^*_F = \sum_{j=1}^k D^*_{\mathcal{F}^j_e}$.*

From Theorem 5, LMI representations of D_F^* is as follows.

Corollary 6 ([15]) *Let sets $\mathcal{F}_e^j, F$, D_F and D_F^* be defined as in (2.7)–(2.9). Then for any matrix $Y \in S^{n+1}$,, $Y \in D_F^*$ if and only if*

$$Y = Y^1 + Y^2 + \cdots + Y^k, \quad \begin{bmatrix} c_j & b_j^T \\ b_j & A_j \end{bmatrix} * Y^j \leq 0, Y^j \in S^{n+1} \quad \text{for } j = 1, \ldots, k.$$

Now we can relax problem (CP) to the following linear conic problem:

$$\begin{aligned} \min\ & H' * Y \\ & Y_{11} = 1 \\ & Y \in D_F^* \end{aligned} \tag{2.10}$$

According to Corollary 6, problem (2.10) can be specifically rewritten as

$$(\text{RCP}) \quad \begin{aligned} & \min H' * Y \\ & Y = Y^1 + Y^2 + \cdots + Y^k, \quad Y_{11} = 1 \\ & \begin{bmatrix} c_j & b_j^T \\ b_j & A_j \end{bmatrix} * Y^j \leq 0, \quad Y^j \in S^{n+1}, \quad j = 1, \cdots, k. \end{aligned} \tag{2.11}$$

By applying the reformulation–linearization technique (RLT) to problem (2.11), a tighter lower bound for problem (2.2) could be obtained. The problem (2.11) is changed as follows:

$$(\text{RCP-RLT}) \quad \begin{aligned} & \min H' * Y \\ & Y = Y^1 + Y^2 + \cdots + Y^k, \quad Y_{11} = 1 \\ & \begin{bmatrix} r_i & p_i^T \\ p_i & Q_i \end{bmatrix} * Y^j \leq 0, \quad i = 0, 1, \ldots, m, \quad j = 1, \ldots, k. \\ & \begin{bmatrix} c_j & b_j^T \\ b_j & A_j \end{bmatrix} * Y^j \leq 0, \quad Y^j \in S^{n+1}, \quad j = 1, \ldots, k. \end{aligned} \tag{2.12}$$

The next theorem shows that problem (RCP-RLT) indeed provides a lower bound for problem (2.2) if $\Lambda \subseteq F$.

Theorem 7 ([4]) *Let F and $\mathcal{F}_e^j$ be defined in (2.7), if $\Lambda \subseteq F$, then*

$$V(RCP) \leq V(RCP - RLT) \leq V(CP) = V(P).$$

Theorem 8 ([4]) *Let $\mathcal{F}_\mathrm{e}^\mathrm{j}$ be defined in (2.7), if the set $\mathcal{F}_\mathrm{e}^\mathrm{j} \cap \Lambda$ has a nonempty interior for $i = 1, \ldots, k$, then problem (RCP-RLT) is strongly feasible.*

Therefore, the ellipsoids $\mathcal{F}_e^j$ are needed to an efficient arrangement to cover Λ. In order to detect which ellipsoid $\mathcal{F}_e^j$ in F should be refined, the concepts of "most sensitive points" and "most sensitive ellipsoids" are introduced. In order to detect the sensitive regions, the following theorem is needed.

Theorem 9 ([4]) *If* $Y^* = (Y^1)^* + (Y^2)^* + \cdots + (Y^k)^*$ *is an optimal solution to problem (RCP-RLT), then, for* $(Y^i)^* \neq 0$ *with* $i \in \{1, \ldots, k\}$*, we have*

$$(Y^1)^* = \sum_{s=1}^{n_i} \alpha_{is} \begin{bmatrix} 1 \\ x^{is} \end{bmatrix} \begin{bmatrix} 1 \\ x^{is} \end{bmatrix}^T$$

for some $n_i \in \{1, \ldots, n+1\}$, $\alpha_{is} > 0$ *and* $x^{is} \in \mathcal{F}_e^j$. *In this case,* Y^* *can be decomposed into*

$$Y^* = \sum_{i:(Y^i)^* \neq 0} \sum_{s=1}^{n_i} \alpha_{is} \begin{bmatrix} 1 \\ x^{is} \end{bmatrix} \begin{bmatrix} 1 \\ x^{is} \end{bmatrix}^T$$

With $\sum_{i:(Y^i)^* \neq 0} \sum_{s=1}^{n_i} \alpha_{is} = 1$

Definition 1 ([4]). For the decomposition

$$Y^* = \sum_{i:(Y^i)^* \neq 0} \sum_{s=1}^{n_i} \alpha_{is} \begin{bmatrix} 1 \\ x^{is} \end{bmatrix} \begin{bmatrix} 1 \\ x^{is} \end{bmatrix}^T \tag{2.13}$$

x^* is the *most sensitive point* if

$$x^* = \underset{\{x^{is}:(Y^i)^* \neq 0;\ s=1,2,\ldots,n_i\}}{\text{argmin}} \left\{ (x^{is})^T Q_0 x^{is} + 2(p_0)^T x^{is} + r_0 \right\} \tag{2.14}$$

The minimum objective value among all of the sensitive points is x^*. Note that if there are multiple sensitive points that having the same minimum objective value, the sensitive point x^* is selected as the one having the smallest index in i with the smallest index in s as a tie breaker. Denote the corresponding index i by t, then the ellipsoid $\mathcal{F}_e^t$ contain sensitive point x^* is named the most sensitive ellipsoid. The next theorem is about connection between the sensitive point x^* and optimal solution of problem (2.2).

Theorem 10 ([4]) *Assume* Y^* *is the optimal solution to problem (RCP-RLT) with the most sensitive point* x^**, then*

$$\begin{bmatrix} 1 \\ x^* \end{bmatrix} \begin{bmatrix} 1 \\ x^* \end{bmatrix}^T * H' \leq V(P) \tag{2.15}$$

Moreover, if $x^* \in \Lambda$*, then the matrix* $\begin{bmatrix} 1 \\ x^* \end{bmatrix} \begin{bmatrix} 1 \\ x^* \end{bmatrix}^T$ *is optimal to problem (CP) and* x^* *is optimal to problem* (2.2).

According to Theorem 8, if $x^* \in \Lambda$, then x^* is an optimal solution of problem (2.2). Otherwise x^* is a lower bound of problem (2.2). In this case, to get a better lower bound we have to refine the ellipsoid cover F. But it does not need to refine F everywhere. A question that arises is how to determine which ones need to refine? To answer this question we use fewer ellipsoids involved in problem (RCP-RLT).

Since the most sensitive point x^* in the most sensitive ellipsoid $\mathcal{F}_e^t$ in F has the lowest objective value, therefore only this ellipsoid is need to refine.

The lower bound of problem (2.2) is improved by refining ellipsoids around the region of x^*.The following definition is to easily manage the ellipsoids in the set F.

Definition 2 ([4]) For a given rectangle set $T = [u, v] = \{x \in R^n | u_i \leq x_i \leq v_i\}$, define the corresponding ellipsoid $\mathcal{F}_e^T$ generated by T as

$$\mathcal{F}_e^T = \left\{ x \in R^n \middle| \sum_{i=1}^{n} \frac{(2x_i - v_i - u_i)^2}{(v_i - u_i)^2} \leq n \right\} \tag{2.16}$$

It is easy to verify that $T \subseteq \mathcal{F}_e^T$.

Let $T = \bigcup_{i=1}^{k} \{T_i\}$ and $\mathfrak{T}$ be the union of the rectangle sets in T. Suppose $\mathfrak{T}$ be a rectangle cover of Λ that

$$\Lambda \subseteq \mathfrak{T} = \bigcup_{i=1}^{k} T_i.$$

Then the set $F = \bigcup_{j=1}^{k} \mathcal{F}_e^j$, whose member $\mathcal{F}_e^j$ is a full-dimensional ellipsoid generated by the rectangle set T_i, respectively, is an ellipsoid cover of Λ. In order to make this method converge quickly, a good initial rectangle set cover is necessary. In order to fulfill this purpose, consider the following problems:

$$\left(I_{min}^i\right) \quad \begin{array}{l} \min\limits_{x \in R^n} x_i \\ f(x) \leq \alpha \\ h_j(x) \leq 0 \quad j = 1, 2, \ldots, m \end{array} \tag{2.17}$$

and

$$\left(I_{max}^i\right) \quad \begin{array}{l} \max\limits_{x \in R^n} x_i \\ f(x) \leq \alpha \\ h_j(x) \leq 0 \quad j = 1, 2, \ldots, m \end{array} \tag{2.18}$$

For $i = 1, 2, \ldots, n$ problems (2.17) and (2.18) are convex programming, hence they can be solved efficiently. Since the feasible region Λ is closed and bounded, therefore problems (2.17) and (2.18) have finite optimal solutions. Denote the optimal solutions of problems (2.17) and (2.18) to be u_i^1 and v_i^1 respectively. The rectangle set $T_1 = [u^1, v^1]$ is chosen as the initial rectangle set covering the feasible region Λ and the ellipsoid $\mathcal{F}_e^1$ generated from T_1 is the initial ellipsoid cover F of Λ. The most sensitive point x^*, the most sensitive ellipsoid $\mathcal{F}_e^t$ and the rectangle set T_t generating the most sensitive ellipsoid $\mathcal{F}_e^t$ is also detected. Then, this rectangle set is split by half. Let $\mathrm{id} = \arg \max\limits_{\{i=1,\ldots,n\}} \{v_i^t - u_i^t\}$, then T_t is split into $T_{t1} = [u^{t1}, v^{t1}]$ and $T_{t2} = [u^{t2}, v^{t2}]$, where $u^{t1} = u^t$, $v^{t2} = v^t$, $v_i^{t1} = v_i^t, u_i^{t2} = u_i^t$, for $i \neq \mathrm{id}$, and $v_{id}^{t1} = u_{id}^{t2} = \frac{u_{id}^t + v_{id}^t}{2}$. Two ellipsoids $\mathcal{F}_e^{t1}$ and $\mathcal{F}_e^{t2}$ are generated from T_{t1} and T_{t2} according to

$$\mathcal{F}_e^{t1} = \left\{ x \in R^n \middle| \sum_{i=1}^{n} \frac{\left(2x_i - v_i^{t1} - u_i^{t1}\right)^2}{\left(v_i^{t1} - u_i^{t1}\right)^2} \leq n \right\} \tag{2.19}$$

and

$$\mathcal{F}_e^{t2} = \left\{ x \in R^n \middle| \sum_{i=1}^{n} \frac{\left(2x_i - v_i^{t2} - u_i^{t2}\right)^2}{\left(v_i^{t2} - u_i^{t2}\right)^2} \leq n \right\}. \tag{2.20}$$

Let $\mathrm{Int}(T_i \cap \Lambda) \neq \emptyset$ for any rectangle set T_i in T, then one of the following occurs:

1. $\mathrm{Int}(T_{t1} \cap \Lambda) \neq \emptyset, \quad \mathrm{Int}(T_{t2} \cap \Lambda) = \emptyset$
2. $\mathrm{Int}(T_{t2} \cap \Lambda) \neq \emptyset, \quad \mathrm{Int}(T_{t1} \cap \Lambda) = \emptyset$
3. $\mathrm{Int}(T_{t1} \cap \Lambda) \neq \emptyset, \quad \mathrm{Int}(T_{t2} \cap \Lambda) \neq \emptyset.$

In the case 1 because $T_{t2} \cap \Lambda$ don't have common interior, thus the rectangle set T_{t2} should be eliminated from the rectangle set cover $\mathfrak{T}$ for further consideration. When the case 1 is occur the rectangle set T_{t2} should be eliminated. In order to determine which rectangle set should be eliminated, consider the following problems:

$$\left(I_{min}^{id}\right) \quad \begin{array}{l} \min\limits_{x \in R^n} x_{id} \\ f(x) \leq \alpha \\ h_j(x) \leq 0 \quad j = 1, 2, \ldots, m \end{array} \tag{2.21}$$

and

$$\left(I_{max}^{id}\right) \quad \begin{array}{l} \max\limits_{x \in R^n} \quad x_{id} \\ f(x) \leq \alpha \\ h_j(x) \leq 0 \quad j = 1, 2, \ldots, m \end{array} \tag{2.22}$$

Problems (2.21) and (2.22) are also convex programming, Therefore they can be solved efficiently. Denote the optimal value of Problems (2.21) and (2.22) to be φ and ψ, respectively. The rectangle sets in T is changed as the following way.

$$\mathfrak{T} = \mathfrak{T} \backslash \{T_t\} \cup \{T_{t2}\}, \quad if \quad \varphi \geq \frac{u_{id}^t + v_{id}^t}{2} \tag{2.23}$$

$$\mathfrak{T} = \mathfrak{T} \backslash \{T_t\} \cup \{T_{t1}\}, \quad if \quad \psi \leq \frac{u_{id}^t + v_{id}^t}{2} \tag{2.24}$$

$$\mathfrak{T} = \mathfrak{T} \backslash \{T_t\} \cup \{T_{t1}\} \cup \{T_{t2}\}, \quad otherwise \tag{2.25}$$

Theorem 11 ([3]) *The set* $T_i \cap \Lambda$ *has a nonempty interior for each rectangle set* T_i *in* T *if the rectangle sets are added into* T *according to* (2.23)–(2.25).

Since each T_i has a common interior with Λ, each ellipsoid $\mathcal{F}_e^i$, generated from T_i, also has a common interior with Λ. Consider the following corollary:

Corollary 12 ([3]) *The set $\mathcal{F}_e^i \cap \Lambda$ has a nonempty interior for each $\mathcal{F}_e^i$ generated by the rectangle set T_i in T.*

Consider the following problem (I^c):

$$(I^c) \quad \begin{array}{ll} \min\limits_{x \in R^n} & \|x - x^*\|_\infty \\ f(x) \le \alpha & \\ h_j(x) \le 0 & j = 1, 2, \ldots, m \end{array} \tag{2.26}$$

The optimal value of problem (I^c) indicates the distance of the current sensitive point x^* to the feasible region Λ. It is clearly that problem (I^c) is a convex programming problem. We denote the optimal solution of (I^c) by $\bar{x}^1$.

Algorithm2 (Adaptive Ellipsoid-based Algorithm (AEA))

Initialization: Solve problems (I_{min}^i) and (I_{max}^i) defined by (2.17) and (2.18) for $i = 1, 2, \cdots, n$ to get the initial rectangle set T_1 and the corresponding ellipsoid $\mathcal{F}_e^1$. Set $\varepsilon > 0$ be the tolerance, $T = \{T_1\}$, $F = \{\mathcal{F}_e^1\}$ and $\bar{\mathfrak{T}} = \emptyset$. Let lower bound $low = -\infty$, upper bound $upp = +\infty$ and approximate solution $\tilde{x} = 0 \in R^n$.

Step 1: Solve problem (RCP-RLT) defined by (2.12) with the approximation cone D_F^*, where F is defined by (2.7). Assume the optimal solution to problem (RCPRLT) is $Y^* = (Y^1)^* + (Y^2)^* + \cdots + (Y^k)^*$. Return the optimal value of problem (RCP-RLT) as l^*. Set $low = \max\{low, l^*\}$. If $|upp - low| \le \varepsilon$, stop and output $\tilde{x}$.

Step 2: Decompose Y^* according to Theorem 8 to obtain the most sensitive point x^* and the most sensitive ellipsoid $\mathcal{F}_e^t = \left\{ x \in R^n \,\middle|\, \sum_{i=1}^n \frac{(2x_i - v_i^t - u_i^t)^2}{(v_i^t - u_i^t)^2} \le n \right\} \in F$, which is generated from the rectangle set $T_t = [u^t, v^t] \in T$. If $x^* \in \Lambda(\alpha)$, stop and output x^*.

Step 3: Set $T = T \backslash \{T_t\}$, $F = F \backslash \{\mathcal{F}_e^t\}$ and $id = \arg\max\limits_{\{i=1,\ldots,n\}} \{v_i^t - u_i^t\}$ .Generate ellipsoids $\mathcal{F}_e^{t1}$ and $\mathcal{F}_e^{t2}$ according to (2.19) and (2.20), respectively. Let φ and ψ be the optimal value of problems (I_{min}^{id}) and (I_{max}^{id}) defined by (2.21) and (2.22), respectively.
– If $\varphi \ge \frac{u_{id}^t + v_{id}^t}{2}$,set $F = F \cup \{\mathcal{F}_e^{t2}\}$, $T = T \cup \{[u^{t2}, v^{t2}]\}$ and $\bar{T} = \bar{T} \cup \{[u^{t2}, v^{t2}]\}$.
– If $\psi \le \frac{u_{id}^t + v_{id}^t}{2}$,set $F = F \cup \{\mathcal{F}_e^{t1}\}$, $T = T \cup \{[u^{t1}, v^{t1}]\}$ and $\bar{T} = \bar{T} \cup \{[u^{t1}, v^{t1}]\}$.
Otherwise $F = F \cup \{\mathcal{F}_e^{t1}\} \cup \{\mathcal{F}_e^{t2}\}$ and $T = T \cup \{[u^{t1}, v^{t1}]\} \cup \{[u^{t2}, v^{t2}]\}$.

Step 4: Solve problem (I^c) defined by (2.26) to obtain $\bar{x}^1$. Set $\bar{x}^2 = \sum_{i:(Y^i)^* \ne 0} \sum_{s=1}^{n_i} \alpha_{is} x^{is}$. If $\min\{p(\bar{x}^1), p(\bar{x}^2)\} < upp$, set $upp = \min\{p(\bar{x}^1), p(\bar{x}^2)\}$ and $\tilde{x} = \operatorname{argmin}_{\{\bar{x}^1, \bar{x}^2\}}\{p(\bar{x}^1), p(\bar{x}^2)\}$. If $|upp - low| < \varepsilon$, stop and output $\tilde{x}$. Otherwise, go to *Step 1*.

3 Convergence of PQCQP method

In this section we will prove that the PQCQP method is convergent to an optimal solution. At first we investigate some properties of function $F(\alpha)$.

Let us rewrite $F(\alpha)$ as follows:

$$F(\alpha) = \min\left\{ g(x) \middle| x \in K(\alpha) \cap \left(\bigcap_{j=1}^{m} T_j \right) \right\} \tag{3.1}$$

where $K(\alpha) = \{x|f(x) \leq \alpha\}$ and $T_j = \{x|h_j(x) \leq 0\}$ for $j = 1, 2, \ldots, m$.

Lemma 13 *$F(\alpha)$ is a decreasing function of α.*

Proof If $\alpha_n \leq \alpha_{n+1}$ then $K(\alpha_n) \subseteq K(\alpha_{n+1})$, so $F(\alpha_{n+1}) \leq F(\alpha_n)$. □

Some properties of $K(\alpha)$ are established in the following lemma.

Lemma 14 *For any $\alpha \in \mathbb{R}$, we have:*

1. *$K(\alpha)$ is a convex set.*
2. *$K(\alpha)$ is a compact set.*

Proof

1. Suppose $x_1, x_2 \in K(\alpha)$ and $\lambda \in [1, 0]$ be an arbitrary constant. So we have

$$f(x_1) \leq \alpha, f(x_2) \leq \alpha.$$

Since the function $f(x)$ is convex, we have

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2) \leq \alpha$$

Thus $x_1 + (1-\lambda)x_2 \in K(\alpha)$.

2. We notice that the continuity of f implies the closedness of the set $K(\alpha)$. Thus, it remains only to show that the set $K(\alpha)$ is bounded. We prove this by contradiction. Suppose that there is an $\alpha \in \mathbb{R}$ such that the set $K(\alpha)$ is unbounded. Then there must exist a sequence $\{x^p\} \subset K(\alpha)$ with $\|x^p\| \to \infty$. But then by the coercivity of f, we must also have $f(x^p) \to \infty$. This contradicts the fact that $f(x^p) \leq \alpha$ for all $= 1, 2, \ldots$. Therefore the set $K(\alpha)$ must be bounded. □

Theorem 15 *Let $K(\alpha)$ be a nonempty set. Then $F(\alpha)$ is well-defined.*

Proof According to Lemma 14, $K(\alpha)$ is a compact set. It is sufficient to prove that $T = \bigcap_{j=1}^{m} T_j$ is closed. It is equivalent to show that every T_j $(j = 1, 2, \ldots, m)$ is closed. We remember that $T_j = \{x|h_j(x) \leq 0\}$. Since $h_j(x) \in (-\infty, 0]$ and $h_j(x)$ is a continuous function, thus T_j is closed set. Therefore $K(\alpha) \cap \left(\bigcap_{j=1}^{m} T_j\right)$ is a compact set.

$g(x)$ is continuous function, so we conclude that $\min g(x)$ on $K(\alpha) \cap \left(\bigcap_{j=1}^{m} T_j\right)$ always exists. Therefore $F(\alpha)$ is well-defined.

Lemma 16 *Let $\{\alpha_n\}_{n=1}^{\infty}$ be a sequence generated by the PQCQP method, then $K(\alpha_n)$ is a nonempty set for any $n \in \mathbb{N}$.*

Proof According to Algorithm 1, the proof is obvious.

Theorem 17 *Let* $\{\alpha_n\}_{n=1}^{\infty}$ *be a sequence generated by the PQCQP method then we have:*

1. $\{F(\alpha_n)\}_{n=1}^{\infty}$ *is convergent.*
2. $\{\alpha_n\}_{n=1}^{\infty}$ *is convergent.*
3. *If* z^* *is the optimal value of problem* (1.1) *then* $\alpha_n \to z^*$.

Proof

1. In Theorems 15 and 16 we have shown that $F(\alpha)$ always exists and is well-defined. Now it is sufficient to prove that the new method is convergent. According to Lemma 11, the sequence $\{F(\alpha_n)\}_{n=1}^{\infty}$ is decreasing. Because $F(\alpha_n)$ is bounded from below to $F(\alpha_2)$, thus $\{F(\alpha_n)\}_{n=1}^{\infty}$ is convergent.
2. Suppose $l > 0$ and $\alpha_{n+1} = (\alpha_n + \alpha_{n-1})/2$. If $|\alpha_2 - \alpha_1| < l$ then $|\alpha_{n+1} - \alpha_n| < \frac{l}{2^n}$. So $|\alpha_{n-1} - \alpha_n| \to 0$ as $n \to \infty$.
3. Since $z^* \in [\alpha_{n-1}, \alpha_n]$. So

$$\alpha_{n-1} \le z^* \le \alpha_n. \tag{3.2}$$

According to Step 2 of Algorithm 1, we have $\alpha_{n+1} = (\alpha_n + \alpha_{n-1})/2$.Therefore

$$\alpha_{n-1} \le \alpha_{n+1} \le \alpha_n \tag{3.3}$$

From (3.2), (3.3) and the proof of part 2, we conclude

$$|\alpha_n - z^*| \le |\alpha_n - \alpha_{n-1}| < \frac{l}{2^{n-1}}.$$

Hence $|\alpha_n - z^*| \to 0$ as $n \to \infty$. □

4 Computational results

In this section, we report some computational results of the PQCQP method. The new method was implemented using MATLAB R2013a on a PC with Intel Core i7 CPU and 16G memory. In the Algorithm 1 and AEA, CVX [8] is used to solve all convex programming problems. The test problems are considered as follows [23]:

$$\begin{aligned} \min_{x \int R^n} \; & x^T Q^{m+1} x + 2p^{{m+1}^T} x + r^{m+1} \\ & x^T Q^0 x + 2p^{0^T} x + r^0 \le 0 \\ & x^T Q^j x + 2p^{j^T} x + r^j \le 0 \quad j = 1, 2, \ldots, m \end{aligned} \tag{4.1}$$

where the matrices $Q^j = O^j D^j (O^j)^T$ for some orthogonal matrices O^j and diagonal matrices $D^j \in S^n_{++} (j = 1, 2, \ldots, m)$. The parameters in the test problems are randomly generated as follows.

The matrix $O^j = P^j_1 P^j_2 P^j_3$, in which $P^j_i = I - 2\frac{\omega_i \omega_i^T}{\|\omega^2\|}$, $i = 1, 2, 3$. The components of vector $\omega_i \in R^n$ are random numbers in [− 1, 1] and I *is* the n-dimensional identity matrix. The matrices D^j are considered in the following way:

$D^0 = Diag(D^0_1, \ldots, D^0_n)$ with $D^0_i \in [-50, 0]$ for $i = 1, \ldots, \lfloor \frac{n}{2} \rfloor$ and $D^0_i \in [0, 50]$ for $i = \lfloor \frac{n}{2} \rfloor + 1, \ldots, n$.

$D^j = Diag(D^j_1, \ldots, D^j_n)$ with $D^j_i \in [0, 50]$ for $j = 1, \ldots, m + 1$.

Also, we set $p^0 = (p^0_1, \ldots, p^0_n)$ with $p^0_i \in [-10, 10]$, $p^j = (p^j_1, \ldots, p^j_n)$ with $p^j_i \in [-50, 0]$, and $r^j \in [-5, 0]$ for $j = 1, \cdots, m + 1$, and $r^0 \in [-5, 0]$.

In order to demonstrate the validity of the PQCQP method, we use the global optimization package BARON [19] to obtain the optimal value.

The termination condition was chosen as $|\alpha_2 - \alpha_1| < \varepsilon$ with $\varepsilon = 10^{-5}$. Since academic version BARON could solve only problems with at most $m = n = 10$, so for $m = n = 10$, 40 random test problems were generated and the results are listed in Table 1, including the optimal value of SDP relaxation problem corresponding to problem (1.1) (to see how good is optimal solution) and the optimal values (labeled by "OPT") and the CPU time (in seconds) required for solving problems by BARON and PQCQP-AEA.

We summarize results of Table 1 in Fig. 1, by method that proposed by Dolan and More in [5]. Figure 1 plots the function

$$\pi_s(\tau) = \frac{1}{|P|} |\{p \in P : r_{p,s} \le \tau\}|.$$

where P denotes the set of problems used for a given numerical experiment and $r_{p,s}$ denotes the ratio the amount of CPU time needed to solving problem p with method s and the least amount of CPU time needed for solving problem p.

The value of $\pi_s(1)$ is probability that the method s will win over test problems. In Fig. 1, we see that PQCQP-AEA method is successful in about 85% of problems and the CPU time for solving problem by PQCQP-AEA is less than CPU time for solving problem by BARON. The Table 1 indicates that PQCQP-AEA could obtain comparable accurate solutions with BARON in an efficient manner.

We report numerical results for some different m and n with the PQCQP-AEA in Table 2.

Table 2 summarizes the average number of iterations and the average CPU time for 50 test problems for cases n = 15 and m = 10, n = m=15, n = 30 and m = 20, n = 50 and m = 10, n = 50 and m = 30 and n = m=50. From Tables 1 and 2, we conclude that the PQCQP-AEA can be successfully applied to problem (1.1).

Table 1 Numerical results for 40 test problems

Instance	SDP relaxation OPT	BARON		PQCQP-AEA	
		OPT	CPU time (s)	OPT	CPU time (s)
1	− 24.5124	− 6.0289	252	− 6.0289	128
2	− 36.7411	− 21.1969	251	− 21.1969	142
3	− 41.2014	− 9.4298	251	− 9.4298	117
4	22.1991	− 18.7130	251	− 18.7130	124
5	− 16.2410	− 10.4258	251	− 10.4258	196
6	− 88.1540	− 10.9253	252	− 10.6132	878
7	− 50.1490	− 20.0025	251	− 20.0025	134
8	− 18.3027	− 15.9799	251	− 15.9799	130
9	− 15.9801	− 15.6225	251	− 15.6225	131
10	− 18.6233	− 18.1946	250	− 18.1946	128
11	− 150.2804	− 10.0157	252	− 9.7245	870
12	− 10.0024	− 9.6108	251	− 9.6108	116
13	− 22.3114	− 16.2552	253	− 16.2552	138
14	− 29.4570	− 12.2657	253	− 12.2657	144
15	− 27.1153	− 16.0992	253	− 16.0992	142
16	− 8.0152	− 7.7217	251	− 7.7217	121
17	− 105.2177	− 23.9440	250	− 23.9440	281
18	− 61.3008	− 21.0915	250	− 21.0915	142
19	− 32.1041	− 19.5248	251	− 19.5248	130
20	− 11.2962	− 8.3339	252	− 8.3339	122
21	− 45.5142	− 14.8133	265	− 14.8133	154
22	− 8.0019	− 6.6665	253	− 6.6665	122
23	− 33.0145	− 18.5253	251	− 18.5253	131
24	− 19.7680	− 10.9821	251	− 10.9821	130
25	− 26.3456	− 17.1209	251	− 17.1209	130
26	− 13.9993	− 9.7844	251	− 9.7844	189
27	− 85.2431	− 11.3432	252	− 11.3432	334
28	− 29.1777	− 18.0511	251	− 18.0511	133
29	− 20.8457	− 16.9599	251	− 16.9599	132
30	− 13.6844	− 11.8552	252	− 11.8552	139
31	− 25.1623	− 10.6704	254	− 10.6704	138
32	− 321.5770	− 10.6902	255	− 10.6004	891
33	− 12.2016	− 9.8198	253	− 9.8198	118
34	− 31.2257	− 13.6658	257	− 13.6658	143
35	− 51.2026	− 12.1915	252	− 12.1915	202
36	− 73.2999	− 29.2332	252	− 29.2332	150
37	− 77.5614	− 8.3456	252	− 8.3456	253
38	− 46.1844	− 27.0819	252	− 27.0819	142
39	− 47.8110	− 18.1924	251	− 18.1924	134
40	− 16.3708	− 9.7463	251	− 9.7463	124

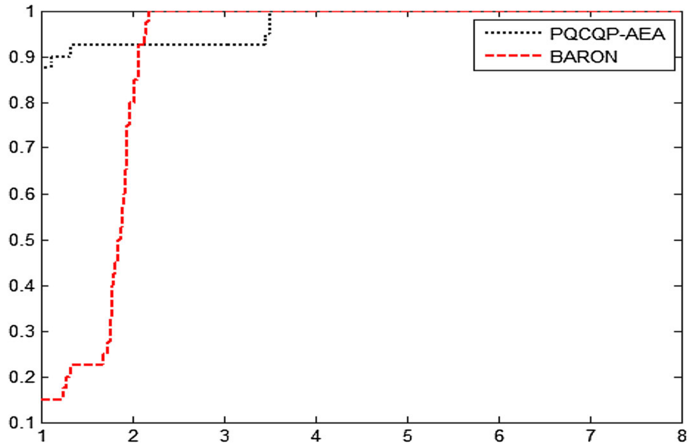

Fig. 1 Performance profile for CPU time

Table 2 Numerical results for random test

n	m	Iteration	CPU time
15	10	10.8	119.6
15	15	12	142
30	20	12.9	655.2
50	10	13.6	942
50	30	13.9	1105
50	50	15	1524

5 Conclusion

In this paper, we have proposed a numerical procedure to find a global optimal solution of the non-convex QCQP problem (1.1) with one non-convex constraint. For solving this problem, we change the problem (1.1) to an equivalent parametric problem (PQCQP) that all of constraints in PQCQP problem (2.2) are convex. The convergence of PQCQP method is investigated. Computational results show that PQCQP-AEA can be successfully applied to problems (1.1).

As for future work, PQCQP method would be extended for the cases of problem (1.1) which objective function and constraints are non-convex.

References

1. Beck, A.: Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB. SIAM, Philadelphia (2014)
2. Ben-Tal, A., Nemirovski, A.: Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications. SIAM, Philadelphia (2001)
3. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2009)

4. Deng, Zh, Fang, Sh, Jin, Q., Lu, Q.C.: Conic approximation to non-convex quadratic programming with convex quadratic constraints. J. Glob. Optim. **61**, 459–478 (2015)
5. Dolan, E., More, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and co, San Francisco (1979)
7. Gershman, A.B., Sidiropoulos, N.D., Shahbazpanahi, S., Bengtsson, M., Ottersten, B.: Convex optimization based beamforming. IEEE Signal Process. Mag. **27**(3), 62–75 (2010)
8. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 1.21. Stanford University. http://cvxr.com/cvx/ (2011)
9. Hillestad, R.J., Jacobsen, S.E.: Linear programs with an additional reverse convex constraint. Appl. Math. Optim. **6**, 257–269 (1980)
10. Konar, A., Sidiropoulos, N., Mehanna, O.: Parametric frugal sensing of power spectra for moving average models. IEEE Trans. Signal Process. **63**(5), 1073–1085 (2015)
11. Kuno, T., Shi, J.: Linear programs with an additional separable concave constraint. J. Appl. Math. Decis. Sci. **8**(3), 155–174 (2004)
12. Liu, Sh-M, Papavassilopoulos, G.P.: A parallel algorithm for linear programs with an additional reverse convex constraint. J. Parallel Distrib. Comput. **45**, 91–103 (1997)
13. Lobo, M.S., Vandenberghe, L., Boyd, S., Lebret, H.: Applications of second-order cone programming. Linear Algebra Appl **284**(1), 193–228 (1998)
14. Lu, C., Fang, S.C., Jin, Q., Wang, Z., Xing, W.: KKT solution and conic relaxation for solving quadratically constrained quadratic programming problems. SIAM J. Optim. **21**, 1475–1490 (2011)
15. Lu, C., Jin, Q., Fang, S.-C., Wang, Z., Xing, W.: Adaptive computable approximation to cones of nonnegative quadratic functions. Optimization **63**(6), 955–980 (2014)
16. Luo, Z., Ma, W., So, A., Ye, Y., Zhang, S.: Semidefinite relaxation of quadratic optimization problems. IEEE Signal Process. Mag. **27**, 20–34 (2010)
17. De Maio, A., Nicola, S., De Huang, Y., Zhang, Y.S., Farina, A.: Code design to optimize radar detection performance under accuracy and similarity constraints. IEEE Trans. Signal Process. **56**(11), 5618–5629 (2008)
18. Rosen, J.B.: Iterative solution of nonlinear optimal control problems. SIAM J. Control. **4**, 223–244 (1966)
19. Sahinidis, N.V.: BARON: a general purpose global optimization software package. J. Glob. Optim. **8**, 201–205 (1996)
20. Sturm, J.F., Zhang, S.: On cones of nonnegative quadratic functions. Math. Oper. Res. **28**, 246–267 (2003)
21. Ye, Y., Zhang, S.: New results on quadratic minimization. SIAM J. Optim. **14**, 245–267 (2003)
22. Shi, Z., Jin, Q.: Second order optimality conditions and reformulations for non-convex quadratically constrained quadratic programming problems. J. Ind. Manag. Optim. **10**(3), 871–882 (2014)
23. Zheng, X.J., Sun, X.L., Li, D.: Non-convex quadratically constrained quadratic programming: best D.C.decomposition and their SDP representations. J. Glob. Optim. **50**, 695–712 (2011)