Noname manuscript No.
(will be inserted by the editor)

Global Optimization for Nonconvex Programs via Convex Proximal Point Method

Yuanyi Zhao · Wenxun Xing

Received: date / Accepted: date

Abstract The nonconvex program plays an important role in the field of optimization and has a lot of applications in practice. However, for general nonconvex programming problems, the lack of verifiable global optimal conditions and the multiple local minimizers make global optimization hard in computation. In this paper, a convex proximal point algorithm (CPPA) is considered for globally solving nonconvex programming problems. We prove that every accumulation point of CPPA is a stationary point and the initial point of CPPA is key to get the global minimum. Serval sufficient conditions for the initial point selection are provided for CPPA getting the global minimum. Motivated by these sufficient conditions, CPPA is applied for the nonconvex quadratic programming problem with convex quadratic constraints with the initial point getting from its Lagrangian dual problem. Numerical results show that the possibility to get the global minimum is much higher than that of randomly selecting initial points.

Keywords Proximal point method · Nonconvex programming · Global optimization · Lagrangian duality

1 Introduction

Consider the following nonconvex programming problem:

$$\begin{aligned} \min \quad & f(x) \\ s.t. \quad & x \in \mathcal{X} \end{aligned} \tag{1}$$

where $f(x)$ is a proper real-valued twice differentiable nonconvex function, and $\mathcal{X}$ is convex, closed and bounded.

This work is supported by the National Science Foundation of China Grant #11771243.

✉Yuanyi Zhao
Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China.
E-mail: zhao-yy18@mails.tsinghua.edu.cn.

Wenxun Xing
Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China.
E-mail: wxing@tsinghua.edu.cn.

The nonconvex programming problem (1) is common and important in the field of optimization. A well-known subclass is the nonconvex quadratic programming problem with convex quadratic constraints, which is denoted as NQPCQC [17], such as the box constrained quadratic programming problem [27] and the trust-region problem [7]. For general nonconvex programming problems, it is hard to get their global minimizers in computation. For instance, NQPCQC is known to be NP-hard in general, even for cases that the quadratic form matrix of the objective function has only one negative eigenvalue [23]. Up to our knowledge, the non-existence of verifiable characterizations in polynomial computable time for global optimal solutions and the multiple local minimizers of nonconvex programming problems make global optimization a great challenge.

The proximal point method was introduced probably first by B. Martinet [20] for solving convex programming problems, and extended by R. T. Rockafellar [25], D. P. Bertsekas and P. Tseng [4], etc. The general iteration formula is described as

$$x_{k+1} \in \arg\min_{x\in\mathcal{X}}\{f(x) + \frac{1}{2c_k}||x - x_k||^2\} \tag{2}$$

with $k = 0, 1, \cdots$, where x_0 is an arbitrary starting point and c_k is a positive scalar parameter.

For convex programming problems, the proximal point methods, including their models, convergent results and applications, have been well researched [5]. Meanwhile, other proximal-like methods have also been considered exclusively for convex problems, such as the dual proximal algorithm [5], proximal bundle method [12], ADMM [5], and proximal gradient method [2], etc. The success of the proximal point methods in convex optimization greatly stimulates us to use these methods for solving nonconvex programming problems.

Proximal and proximal-like methods have been applied for nonconvex programming problems. M. Fukushima and H. Mine [9] in 1981 applied a proximal point algorithm for a nonconvex programming problem by linearizing the nonconvex function at each iteration. H. Apolinario et al [1] and G. Bento et al [3] gave proximal point methods for multiobjective minimization problems and presented their convergence results. In [24] and [29], proximal ADMM methods were proposed for a structured nonconvex programming problem and a linearly constrained nonconvex minimization. In [16], a proximal bundle method was presented for an unconstrained nonconvex problem, and it was applied to the nonsmooth nonconvex constrained optimization problem by using the improvement function in [13,14]. In view of the applications in [8,10,11,15,22,26,28], it has been shown that the proximal point method is an efficient tool that can transform a given problem into a sequence of easily solved subproblems and many popular algorithms are adopted then. But the global optimization issue is seldom addressed in these works.

For globally solving a nonconvex programming problem $\min\{f(u) : u \in \mathcal{R}^n\}$, A. Kaplan and R. Tichatschke [15] considered the proximal point method and proved that the iterations converge to a globally optimal solution under the following conditions, the initial point is selected from a level set $\Omega_c = \{u : f(u) \le c\}$ which satisfies that $||\nabla f(u)|| > d > 0$ for any $u \in \Omega_c \setminus \Omega_{c_0}$ for some $c_0 \in [f^*, c)$ where f^* is the optimal objective value of the problem, Ω_{c_0} is convex, f is convex on Ω_{c_0} and other two conditions.

The main contributions of this paper are listed below.

In terms of computability, the subproblem (2) may not be easily solvable as it is not a convex optimization problem. In our consideration of using the proximal point method (2) for finding a globally minimizer of (1), we first select a small c_k to make the objective function $f(x) + \frac{1}{2c_k}||x - x_k||^2$ convex in (2) at each iteration, which is called a convex proximal point algorithm (abbreviated as CPPA). Then (2) is a constrained convex optimization problem and thus is assumed being easily solved by convex optimization algorithms. Like the convergence properties for convex optimization problems, we find that every accumulation point of our CPPA is a stationary point of the problem and it depends on the initial point for finding a globally minimizer.

Based on these convergence properties, three special neighbours for the initial point location are provided for getting a global minimizer of (1). The first one is a neighbour closer enough to a global minimizer, then an $\epsilon-$global minimizer can be obtained even though $f(x)$ is not convex over it. The second one is a convex neighbor which contains a global minimizer and $f(x)$ is convex over it, and the third one is a convex neighbor which contains a global minimizer and $f(x)$ may not be convex but there is no stationary point over it.

Motivated by our above theoretical results, we finally apply our idea to NQPCQC problems, develop a proximal point algorithm by selecting an initial point from an optimal solution of its Lagrangian dual problem. Numerical experiments show that the possibility to get the global minimum is higher than that of randomly selecting initial points.

General notations are listed as follows. Let $\mathcal{R}^n$ denote the n-dimensional Euclidean space, $\mathcal{R}^n_+$ represent the first orthant of $\mathcal{R}^n$, and $\|\cdot\|$ denote the Euclidean norm. $x_k \in \mathcal{R}^n$ denotes the point at the k-th iteration. $\mathcal{S}^n$ denotes the set of $n \times n$ symmetric matrices and $\mathcal{S}^n_+$ denotes the set of semi-definite positive matrices. $\nabla f(x)$ is used to represent the gradient of f at x, $\nabla^2 f(x)$ is used to denote the Hessian. $\mathcal{X}$ is the feasible domain for the problem (1). In addition, let us denote by f^* the optimal value of (1), i.e.,

$$f^* = \inf_{x \in \mathcal{X}} f(x)$$

and by $\mathcal{X}^*$ the set of global minimizers of (1),

$$\mathcal{X}^* = \arg \inf_{x \in \mathcal{X}} f(x)$$

The rest of this paper is organized as follows. In Section 2, we give theoretical results about our proximal point method. In Section 3, the NQPCQC problem is taken as an example and a dual initial point algorithm is presented. Numerical results are presented in Section 4 and conclusions are provided in Section 5.

2 Our CPPA and convergence results

We consider (1) under the following four assumptions throughout this paper.
(i) The objective function $f(x)$ is a proper real-valued twice differentiable nonconvex function.
(ii) The set $\mathcal{X}$ is convex, closed and bounded.

(iii) The minimum eigenvalue λ^* of the Hessian $\nabla^2 f(x)$ is finite and negative.
(iv) The parameter c_k, $k = 0, 1, \cdots$ satisfy

$$0 < M \leq c_k \leq -\frac{1}{\lambda^*}. \tag{3}$$

where M is a lower bound of c_k.

Remark 1 *In our CPPA, (2) is assumed a convex optimization problem, so we give the assumption (3). Then convex optimization softwares can be used for solving (2).*

Under the four assumptions above for (1) and (2), we give our algorithm as follows.

Convex Proximal Point Algorithm (CPPA)

Input and output. Input an instance of (1) and an error tolerance $\epsilon > 0$. Output a solution x_k and its value $f(x_k)$.

S0. Select an initial point $x_0 \in \mathcal{R}^n$.
S1. **Loop step.** If one termination criterion is satisfied, then exits this loop. Otherwise, for $k = 0, 1, 2, \cdots$, select a parameter c_k such that (3) is satisfied and (2) is a convex optimization problem, then obtain x_{k+1} by solving (2). Termination criteria are $\|x_{k+1} - x_k\| \leq \epsilon$ or $|f(x_{k+1}) - f(x_k)| \leq \epsilon$.
S2. Stop and output x_k and $f(x_k)$.

To state the convergence results of our method, preliminary definitions are listed below.

Definition 1 *(Tangent direction.) Given any $x \in \mathcal{X}$ in (1), d is a tangent direction at x if there is a sequence of vectors $\{d_m\}_{m=1}^{+\infty} \subseteq \mathcal{R}^n$ and a sequence of positive numbers $\{\theta_m\}_{m=1}^{+\infty} \subseteq \mathcal{R}_+$ such that the following conditions are satisfied*

$$x + \theta_m d_m \in \mathcal{X}, \quad \lim_{m \to +\infty} d_m = d, \quad \lim_{m \to +\infty} \theta_m = 0. \tag{4}$$

Furthermore, the following set $\mathcal{T}(x)$ is defined as the set of tangent directions at $x \in \mathcal{X}$,

$$\mathcal{T}(x) = \{d \in \mathcal{R}^n | d \text{ is a tangent direction at } x\}. \tag{5}$$

Definition 2 *(Stationary point.) A point $x \in \mathcal{X}$ is a stationary point of (1) if it satisfies*

$$\nabla f(x)^T d \geq 0, \quad \forall d \in \mathcal{T}(x) \tag{6}$$

where $\mathcal{T}(x)$ is the set of tangent directions at x.

A naive result between a local minimizer and a stationary point is stated as follows.

Theorem 1 *If $x^* \in \mathcal{X}$ is a local minimizer of (1), then x^* is a stationary point of (1).*

Proof. For any $d \in \mathcal{T}(x^*)$, the result is obvious if $d = 0$, let us assume that $d \neq 0$. From (4), we have a sequence of vectors $\{d_m\}_{m=1}^{+\infty} \subseteq \mathcal{R}^n$ and a sequence of positive numbers $\{\theta_m\}_{m=1}^{+\infty} \subseteq \mathcal{R}_+$ such that

$$x^* + \theta_m d_m \in \mathcal{X}, \quad \lim_{m \to +\infty} d_m = d, \quad \lim_{m \to +\infty} \theta_m = 0.$$

Meanwhile we have

$$f(x^* + \theta_m d_m) - f(x^*) = \theta_m \nabla f(x^*)^T d_m + \frac{\theta_m^2}{2} d_m^T \nabla^2 f(\xi) d_m,$$

where ξ is between x^* and $x^* + \theta_m d_m$.

Because $x^* \in \mathcal{X}$ is a local minimizer of (1), there exists $N > 0$, and for $m > N$

$$f(x^* + \theta_m d_m) - f(x^*) \geq 0.$$

Namely

$$\theta_m \nabla f(x^*)^T d_m + \frac{\theta_m^2}{2} d_m^T \nabla^2 f(\xi) d_m \geq 0,$$

$$\nabla f(x^*)^T d_m + \frac{\theta_m}{2} d_m^T \nabla^2 f(\xi) d_m \geq 0.$$

When m tends to $+\infty$, we have

$$\nabla f(x^*)^T d \geq 0,$$

which ends our proof. □

Theorem 1 shows that a local minimizer of (1) must be a stationary point. From our next basic convergence theorem, it is proved that any limitation point of the sequence generated by our CPPA is a stationary point. Thus our proximal point method does not exclude local minimizers.

Theorem 2 *(Basic convergence theorem.) Under our four assumptions and the sequence generated by our CPPA is denoted as* $\{x_k | k = 0, 1, \dots\}$*, we have*
(i) the functional value sequence $\{f(x_k) | k = 0, 1, \dots\}$ *is monotonically decreasing and convergent,*
(ii) $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$,
(iii) x_k *is a stationary point of the problem (1) if* $x_k = x_{k+1}$ *holds for some* k,
(iv) and any accumulation point of the sequence $\{x_k\}$ *when* $c_k = c_0, k = 1, 2\dots$ *is a stationary point of (1).*

Proof. (i) From (2) we have

$$f(x_{k+1}) + \frac{1}{2c_k} ||x_{k+1} - x_k||^2 \leq f(x_k), k = 0, 1, \cdots. \tag{7}$$

So $\{f(x_k)\}$ is monotonically decreasing. By the assumptions (i) and (ii), $f(x)$ is differential and $\mathcal{X}$ is bounded and closed, so $\{f(x_k)\}$ is convergent and its limitation denoted as f_∞ is finite.

(ii) From (7) we have

$$\frac{1}{2c_k} ||x_{k+1} - x_k||^2 \leq f(x_k) - f(x_{k+1}), k = 0, 1, \cdots,$$

and consequently

$$\sum_{k=0}^{\infty} \frac{1}{2c_k} ||x_{k+1} - x_k||^2 \leq f(x_0) - f_\infty < +\infty.$$

Thus $\sum_{k=1}^{\infty} \frac{1}{2c_k}||x_{k+1} - x_k||^2$ is convergent, and then

$$\lim_{k\to\infty} \frac{1}{2c_k}||x_{k+1} - x_k||^2 = 0.$$

Since the parameter c_k satisfies (3), then $\lim_{k\to\infty} ||x_{k+1} - x_k|| = 0$ is proven.

(iii) We adapt the reduction to absurdity to prove it. Suppose x_k be not a stationary point when $x_k = x_{k+1}$ for some $k \geq 0$. By Definition 2, there exists a $d \in \mathcal{T}(x_k)$ such that $\nabla f(x_k)^T d < 0$. For $d \in \mathcal{T}(x_k)$, there exists $\{d_m\}_{m=1}^{+\infty} \subseteq \mathbb{R}^n$, a sequence of positive numbers $\{\theta_m\}_{m=1}^{+\infty} \subseteq R_+$ such that

$$y_m = x_k + \theta_m d_m \in \mathcal{X}, \quad \lim_{m\to+\infty} d_m = d, \quad \lim_{m\to+\infty} \theta_m = 0. \tag{8}$$

Then

$$\begin{aligned}
&f(y_m) + \frac{1}{2c_k}||y_m - x_k||^2 \\
&= f(x_k + \theta_m d_m) + \frac{1}{2c_k}||\theta_m d_m||^2 \\
&= f(x_k) + \theta_m \nabla f(x_k)^T d_m + \frac{\theta_m^2}{2} d_m^T \nabla^2 f(\xi) d_m + \frac{1}{2c_k}||\theta_m d_m||^2 \\
&= f(x_k) + \theta_m(\nabla f(x_k)^T d_m + \frac{\theta_m}{2} d_m^T \nabla^2 f(\xi) d_m + \frac{\theta_m}{2c_k}||d_m||^2)
\end{aligned}$$

where ξ is between y_m and x_k.

Together with (8) and $\nabla f(x_k)^T d < 0$, we know that there exists a $N > 0$ such that the following inequality is satisfied for all $m \geq N$,

$$\nabla f(x_k)^T d_m + \frac{\theta_m}{2} d_m^T \nabla^2 f(\xi) d_m + \frac{\theta_m}{2c_k}||d_m||^2 < 0.$$

Then, for any positive number θ_m,

$$\begin{aligned}
f(y_m) + &\tfrac{1}{2c_k}||y_m - x_k||^2 - f(x_k) \\
&= \theta_m(\nabla f(x_k)^T d_m + \tfrac{\theta_m}{2} d_m^T \nabla^2 f(\xi) d_m + \tfrac{\theta_m}{2c_k}||d_m||^2) \\
&< 0,
\end{aligned}$$

and

$$f(y_m) + \frac{1}{2c_k}||y_m - x_k||^2 < f(x_k). \tag{9}$$

Due to (9) and $x_k = x_{k+1}$, we have

$$f(y_m) + \frac{1}{2c_k}||y_m - x_k||^2 < f(x_{k+1}) + \frac{1}{2c_k}||x_{k+1} - x_k||^2, \tag{10}$$

which contradicts to $x_{k+1} = x_k$ with the iterative formula (2). Thus (iii) is proven.

(iv) Since the feasible $\mathcal{X}$ is bounded and closed, the sequence $\{x_k | k = 0, 1, \dots\} \subseteq \mathcal{X}$ is bounded too and has convergent subsequences. Suppose $\{x_{k_i}\}$ be a subsequence whose corresponding limitation point x_{k_∞} is not a stationary point. With the same arguments as in the proof of (iii), there exist $d \in \mathcal{T}(x_{k_\infty})$, $\{\theta_m\}$ and $\{d_m\}$ such that $\nabla f(x_{k_\infty})^T d < 0$ and (8) is satisfied by replacing x_k by x_{k_∞}. Under the condition

$c_k = c_0, k = 1, 2, \ldots$, we get a similar result like (9) in the following form for $m > N$ and $y_m = x_{k_\infty} + \theta_m d_m \in \mathcal{X}$

$$f(y_m) + \frac{1}{2c_{k_\infty}}||y_m - x_{k_\infty}||^2 < f(x_{k_\infty}) \tag{11}$$

where $0 < c_{k_\infty} = c_0 < +\infty$.

For any k_i and getting from the optimality of (2), we have

$$f(x_{k_i+1}) + \frac{1}{2c_0}||x_{k_i+1} - x_{k_i}||^2 \leq f(y_m) + \frac{1}{2c_0}||y_m - x_{k_i}||^2.$$

With the first result (i) of this theorem and the monotonically decreasing of $\{f(x_k)\}$, we have

$$\begin{aligned} f(x_{k_{(i+1)}}) + \frac{1}{2c_0}||x_{k_i+1} - x_{k_i}||^2 &\leq f(x_{k_i+1}) + \frac{1}{2c_0}||x_{k_i+1} - x_{k_i}||^2 \\ &\leq f(y_m) + \frac{1}{2c_0}||y_m - x_{k_i}||^2. \end{aligned}$$

Then

$$f(x_{k_{(i+1)}}) + \frac{1}{2c_0}||x_{k_i+1} - x_{k_i}||^2 \leq f(y_m) + \frac{1}{2c_0}||y_m - x_{k_i}||^2 \tag{12}$$

Notice the difference of foot-indexes of $f(x_{k_{(i+1)}})$ and $f(x_{k_i+1})$ in the above equation, with the first inequality getting from the monotonically decreasing of $\{f(x_k)\}$. Taking limitation on both sides of (12) for i, and together with the result (ii) of this theorem, we get

$$f(x_{k_\infty}) \leq f(y_m) + \frac{1}{2c_0}||y_m - x_{k_\infty}||^2, \tag{13}$$

which is a contradiction to (11). So x_{k_∞} is a stationary point. □

Remark 2 *In terms of (iii) in Theorem 2 and (7), the two conditions,* $x_k = x_{k+1}$ *and* $f(x_k) = f(x_{k+1})$*, are used as termination criteria in our CPPA.*

Unfortunately, a stationary point may not be a minimizer or may be the point with the worst value. See the following simple example.

Example 1 *A nonconvex optimization problem is* $\min_{-0.5\leq x\leq 1} -x^2$*. It is easy to see in Fig. 1 that* $x^* = 1$ *is a global minimizer and* $x^* = -0.5$ *is a local minimizer. If setting* $c_k = \frac{1}{4}$*, then (2) is formulated as*

$$x_{k+1} \in \underset{x\in[-0.5,1]}{\arg\min}\,\{-x^2 + 2(x - x_k)^2\} = \underset{x\in[-0.5,1]}{\arg\min}\,\{x^2 - 2x_k x + 2x_k^2\}.$$

If the initial point is set as $x_0 = 0$*, then* $x_1 = x_0$*. Our iteration is stopped and* $x_0 = 0$ *is a stationary point. The stationary point* $x_0 = 0$ *is not a global or local minimizer, and* $f(0) = 0$ *gets its highest value. Obviously, we get a local minimizer* $x_1 = -0.5$ *if choosing an initial point* $-0.5 \leq x_0 < 0$*, and a global minimizer* $x_1 = 1$ *if choosing an initial point* $0 < x_0 \leq 1$*. So CPPA gets a local or global minimizer if a small perturbation is given for the initial* $x_0 = 0$*.*

Being hinted at this example and aiming at getting an global minimizer of (1), we give three sufficient conditions for initial point selection of CPPA.

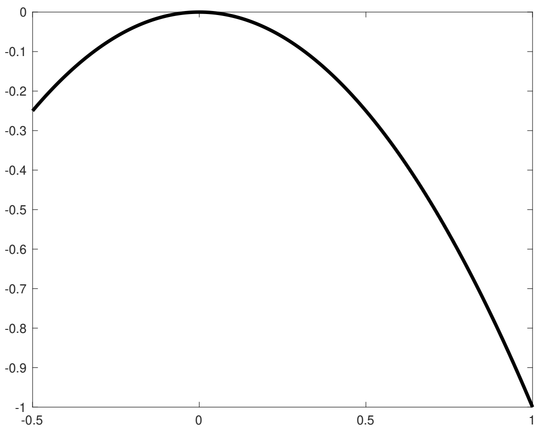

Fig. 1 Curve of $-x^2$ over $[-0.5, 1]$

Lemma 1 *Under our four assumptions and the sequence generated by CPPA is denoted as* $\{x_k | k = 0, 1, \dots\}$*, we have* $||x_{k+1} - x_k|| \le \delta$ *if* $||x_k - x^*|| \le \delta$ *for some* k*, where* $x^* \in X^*$ *is one global minimizer and* $\delta > 0$*.*
Proof. As $x_{k+1} \in \arg\min_{x \in \mathcal{X}} \{f(x) + \frac{1}{2c_k} ||x - x_k||^2\}$, we have

$$f(x_{k+1}) + \frac{1}{2c_k} ||x_{k+1} - x_k||^2 \le f(x^*) + \frac{1}{2c_k} ||x^* - x_k||^2. \tag{14}$$

If $||x_{k+1} - x_k|| > \delta$, then

$$f(x_{k+1}) + \frac{1}{2c_k} ||x_{k+1} - x_k||^2 > f(x_{k+1}) + \frac{\delta^2}{2c_k}. \tag{15}$$

Since $x^* \in X^*$ is a global minimizer, then $f(x^*) \le f(x_{k+1})$ holds. Together with $||x^* - x_k|| \le \delta$ and (15), we have

$$f(x^*) + \frac{1}{2c_k} ||x^* - x_k||^2 \le f(x_{k+1}) + \frac{\delta^2}{2c_k} < f(x_{k+1}) + \frac{1}{2c_k} ||x_{k+1} - x_k||^2, \tag{16}$$

which is a contradiction to (14). Thus $||x_{k+1} - x_k|| \le \delta$, which ends our proof. □

Remark 3 *The above lemma is quite interesting. If an* x_k *is near a global minimizer within distance* δ*, then the step size of the next* x_{k+1} *of CPPA is not larger than* δ*. Together with* $f(x_{k+1}) < f(x_k)$ *of (i) in Theorem 2, we can draw a picture to show the interesting result. When* x_k *is trapped in a round basin with diameter* δ *and its boundary is in a contour with height no less than that of any interior point of the basin, then the next step* x_{k+1} *can not escape from the basin if there is a* δ *width great-wall around the boundary. For any two basins with the above shape, the sequence of CPPA will not jump from one basin to another. Then the sequence always be trapped around a global minimizer if the initial point is selected in a basin containing the minimizer.*

In view of real computation, we give the following two definitions.

Definition 3 *(ϵ-distinguishable point.) Given any $\epsilon > 0$ and $x_1, x_2 \in \mathcal{X}$, x_1, x_2 are ϵ-distinguishable points if $|f(x_1) - f(x_2)| > \epsilon$.*

Definition 4 *(ϵ-global minimizer.) Given any $\epsilon > 0$, $x \in \mathcal{X}$ is an ϵ-global minimizer of (1) if there exists a global minimizer $x^* \in \mathcal{X}^*$ such that $|f(x) - f(x^*)| \leq \epsilon$.*

Theorem 3 *Under our four assumptions and the sequence generated by CPPA is denoted as $\{x_k | k = 0, 1, \dots\}$, for a given $\epsilon > 0$, if an initial point x_0 and a global minimizer $x^* \in X^*$ satisfy the following condition*

$$||x_0 - x^*||^2 \leq 2c_0\epsilon$$

Then any accumulation point of the sequence $\{x_k\}$ is an ϵ-global minimizer of (1).
Proof. For any convergent subsequence $\{x_{k_i}\}$ and its corresponding limitation point x_{k_∞}, let $k_1 = 0$ without loss of generality. From (2) and (i) of Theorem 2, we have

$$\begin{aligned} f(x_{k_i}) + \tfrac{1}{2c_0}||x_1 - x_0||^2 &\leq f(x_1) + \tfrac{1}{2c_0}||x_1 - x_0||^2 \\ &\leq f(x^*) + \tfrac{1}{2c_0}||x^* - x_0||^2 \\ &\leq f(x^*) + \epsilon \end{aligned}$$

for any $i \geq 2$. Namely

$$f(x_{k_i}) - f(x^*) \leq \epsilon.$$

Since x^* is a global minimizer of (1), then we have $f(x_{k_i}) \geq f(x^*)$. Therefore

$$f(x^*) \leq f(x_{k_i}) \leq f(x^*) + \epsilon,$$

i.e.,

$$|f(x_{k_\infty}) - f(x^*)| \leq \epsilon$$

where x_{k_∞} is the limitation point of $\{x_{k_i}\}$. □

The above theorem shows that if an initial point x_0 is close enough to a global minimizer, then an ϵ-global minimizer can be got by CPPA. Obviously from $||x_0 - x^*||^2 \leq 2c_0\epsilon$, the larger c_0 is, the wider area is at which x_0 can be selected for getting an ϵ-global minimizer. But c_0 is restricted by (3) to keep the convexity of the objective function of (2). So if the lower bound λ^* is closer to 0, then we can get an ϵ-global minimizer by our CPPA with higher probability.

To get a global minimizer or an ϵ-global minimizer with higher probability, we consider the following special regions.

Definition 5 *(Global optimality locally convex set) $D(z, \delta) = \{x \in R^n \mid ||x - z|| \leq \delta\}$, where $\delta > 0$, is a global optimality locally convex set if it satisfies the following conditions.*
(i) (Global optimality) There exists an $x^* \in \mathcal{X}^*$ such that $x^* \in D(z, \delta)$.
(ii) (Feasibility) $D(z, \delta) \subseteq \mathcal{X}$.
(iii) (Trapping) For a given $\overline{x} \in D(z, \delta)$, we have $x \in D(z, \delta)$ whenever $f(x) < f(\overline{x})$ and $||x - \overline{x}|| \leq 2\delta$ for $x \in \mathcal{X}$.
(iv) (Local convexity) $f(x) \geq f(y) + (x - y)^T \nabla f(y)$ for any $x, y \in D(z, \delta)$.

The following example shows that the above definition is reasonable.

Example 2 *For the nonconvex problem* $\min_{-\frac{\pi}{2}\leq x\leq 2\pi}\sin x$*, the global minimum set* $\mathcal{X}^*=\{-\frac{\pi}{2},\frac{3\pi}{2}\}$ *as shown in Fig. 2.* $D(-\frac{\pi}{4},\frac{\pi}{4})$, $D(-\frac{3\pi}{8},\frac{\pi}{8})$ *are two global optimality locally convex sets containing* $-\frac{\pi}{2}$*, and* $D(\frac{3\pi}{2},\frac{\pi}{2})$, $D(\frac{3\pi}{2},\frac{\pi}{4})$, $D(\frac{5\pi}{4},\frac{\pi}{4})$ *are three global optimality locally convex sets containing* $\frac{3\pi}{2}$.

To verify that $D(-\frac{\pi}{4},\frac{\pi}{4})=[-\frac{\pi}{2},0]$ *satisfies Definition 5, it is easily checked that (i)* $x^*=-\frac{\pi}{2}\in D(-\frac{\pi}{4},\frac{\pi}{4})$*, (ii)* $D(-\frac{\pi}{4},\frac{\pi}{4})\subseteq\mathcal{X}=[-\frac{\pi}{2},2\pi]$ *and (iv)* $\sin x$ *is convex over* $D(-\frac{\pi}{4},\frac{\pi}{4})$.

To verify the remaining (iii), for a given $\bar{x}\in D(-\frac{\pi}{4},\frac{\pi}{4})$*, we imply* $-\frac{\pi}{2}\leq x\leq\frac{\pi}{2}$ *when* $x\in\mathcal{X}$ *and* $\|x-\bar{x}\|\leq\frac{\pi}{2}$*. As* $\sin\bar{x}$ *being an increasing function over* $D(-\frac{\pi}{4},\frac{\pi}{4})=[-\frac{\pi}{2},0]$*, we imply that* $x\in[-\frac{\pi}{2},0]=D(-\frac{\pi}{4},\frac{\pi}{4})$ *if* $f(x)<f(\bar{x})$ *and* $-\frac{\pi}{2}\leq x\leq\frac{\pi}{2}$*. Then (iii) is checked.*

With the similar arguments as above, The remaining four global optimality locally convex sets can be similarly checked.

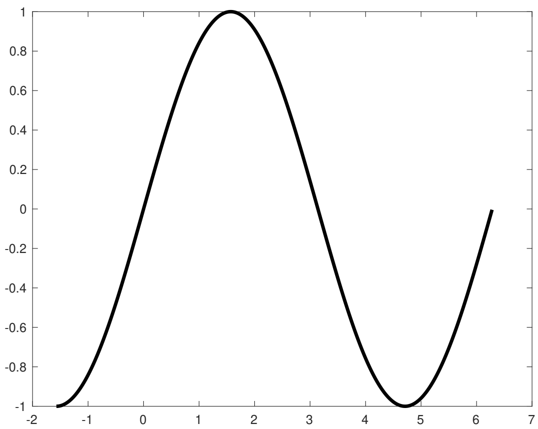

Fig. 2 Curve of $\sin x$ over $[-\frac{\pi}{2},2\pi]$

Theorem 4 *Under our four assumptions and the sequence generated by CPPA is denoted as* $\{x_k|k=0,1,\dots\}$*, then* $\{x_k\}$ *converges to a global minimizer in* $D(z,\delta)\cap\mathcal{X}^*$ *if an initial point* x_0 *is selected from one* $D(z,\delta)$.

Proof. Firstly, we prove $\{x_k|k=0,1,\dots\}\subseteq D(z,\delta)$. For any $x_k\in D(z,\delta)$ and $x^*\in X^*\cap D(z,\delta)$, we have

$$||x_k-x^*||\leq||x_k-z||+||z-x^*||\leq 2\delta. \tag{17}$$

By Lemma 1, we have

$$||x_{k+1}-x_k||\leq 2\delta \tag{18}$$

where $x_{k+1}\in\arg\min_{x\in\mathcal{X}}\{f(x)+\frac{1}{2c_k}||x-x_k||^2\}$. Meanwhile, we have $x_{k+1}=x_k$ with $f(x_{k+1})=f(x_k)$ or $x_{k+1}\neq x_k$ with $f(x_{k+1})<f(x_k)$ by

$$f(x_{k+1})+\frac{1}{2c_k}||x_{k+1}-x_k||^2\leq f(x_k).$$

When $x_{k+1} \neq x_k$ with $f(x_{k+1}) < f(x_k)$ and $x_k \in D(z,\delta)$, we have $\|x_k - x^*\| \leq 2\delta$ by (17) and then $x_{k+1} \in D(z,\delta)$ by (18) and (iii) of Definition 5. When $x_{k+1} = x_k$ with $f(x_{k+1}) = f(x_k)$, obviously $x_{k+1} \in D(z,\delta)$ if $x_k \in D(z,\delta)$. By the induction, we have $\{x_k | k = 0,1,\dots\} \subseteq D(z,\delta)$.

Secondly, we prove, for any $y \in D(z,\delta)$,

$$||x_{k+1} - y||^2 \leq ||x_k - y||^2 - 2c_k(f(x_{k+1}) - f(y)) - ||x_k - x_{k+1}||^2. \tag{19}$$

By the above result $\{x_k | k = 0,1,\dots\} \subseteq D(z,\delta)$, $x_{k+1} \in \arg\min\limits_{x\in\mathcal{X}}\{f(x) + \frac{1}{2c_k}||x - x_k||^2\}$ of (2) is equivalent to

$$x_{k+1} \in \arg\min_{x\in D(z,\delta)}\{f(x) + \frac{1}{2c_k}||x - x_k||^2\}. \tag{20}$$

Obviously, for any $y \in D(z,\delta)$,

$$||x_k - y||^2 = ||x_k - x_{k+1}||^2 + 2(x_k - x_{k+1})^T(x_{k+1} - y) + ||x_{k+1} - y||^2. \tag{21}$$

As $D(z,\delta)$ is a ball with interior points, the Slater condition is satisfied, and thus the KKT condition is satisfied

$$\begin{cases} \nabla f(x_{k+1}) + \frac{x_{k+1} - x_k}{c_k} + 2\lambda(x_{k+1} - z) = 0 \\ \lambda(||x_{k+1} - z||^2 - \delta^2) = 0,\ \lambda \geq 0. \end{cases} \tag{22}$$

When x_{k+1} is an interior of $D(z,\delta)$, i.e., $x_{k+1} \in \text{int}D(z,\delta)$, from (22), we have $\lambda = 0$ and $\nabla f(x_{k+1}) = \frac{x_k - x_{k+1}}{c_k}$. Together with (iv) of Definition 5, we have for any $y \in D(z,\delta)$

$$f(y) \geq f(x_{k+1}) + \frac{(x_k - x_{k+1})^T(y - x_{k+1})}{c_k},$$

and with (21)

$$||x_k - y||^2 \geq ||x_k - x_{k+1}||^2 + 2c_k(f(x_{k+1}) - f(y)) + ||x_{k+1} - y||^2,$$

which is easily reformulated to (19).

When x_{k+1} is at the boundary of $D(z,\delta)$, i.e., $x_{k+1} \in \text{bdry}D(z,\delta)$, from (22) and for any $y \in D(z,\delta)$, we have

$$(y - x_{k+1})^T\nabla f(x_{k+1}) = \frac{(y - x_{k+1})^T(x_k - x_{k+1})}{c_k} + 2\lambda(y - x_{k+1})^T(z - x_{k+1}). \tag{23}$$

Since $x_{k+1} \in \text{bdry}D(z,\delta)$, $y \in D(z,\delta)$, z is at the center of $D(z,\delta)$ and $D(z,\delta)$ is a convex set, we have

$$(y - x_{k+1})^T(z - x_{k+1}) \geq 0,$$

and consequently from (23)

$$(y - x_{k+1})^T\nabla f(x_{k+1}) \geq \frac{(y - x_{k+1})^T(x_k - x_{k+1})}{c_k}.$$

Together with (iv) of Definition 5, we have for any $y \in D(z,\delta)$

$$f(y) \geq f(x_{k+1}) + (y - x_{k+1})^T\nabla f(x_{k+1}) \geq f(x_{k+1}) + \frac{(y - x_{k+1})^T(x_k - x_{k+1})}{c_k}.$$

Together with (21), we get (19).

Finally, we prove that the sequence $\{x_k\}$ converges to a global minimizer in $D(z,\delta)\cap\mathcal{X}^*$ in two cases, finite and infinite of the sequence.

In the case of a finite sequence, CPPA stops at a k such that $x_k = x_{k+1}$, and

$$f(x_{k+1}) = \min_{x\in D(z,\delta)}\{f(x) + \frac{1}{2c_k}\|x - x_k\|^2\}. \tag{24}$$

By the reduction to absurdity, if $f(x_{k+1}) > f^* = f(x^*)$, then from (iv) of Definition 5, we have

$$0 > f(x^*) - f(x_{k+1}) = f(x^*) - f(x_k) \geq \nabla f(x_k)^T(x^* - x_k). \tag{25}$$

Denoting $d = x^* - x_k$, we have $x_k + \alpha d \in D(z,\delta) \subseteq \mathcal{X}$ with $0 \leq \alpha \leq 1$ as $D(z,\delta)$ is a convex set. By (25), there exists a $0 < \bar{\alpha} < 1$ such that

$$\nabla f(x_k)^T d + \frac{\bar{\alpha}}{2} d^T \nabla^2 f(x_k) d + \frac{\bar{\alpha}}{2c_k}\|d\|^2 + \bar{\alpha} o(\|d\|^2) < 0.$$

Denoting $y = x_k + \bar{\alpha} d$ and together with Taylor formula at x_k, we have

$$\begin{aligned}
&f(y) + \frac{1}{2c_k}\|y - x_k\|^2 \\
&= f(x_k) + \bar{\alpha}\nabla f(x_k)^T d + \frac{\bar{\alpha}^2}{2} d^T\nabla^2 f(x_k) d + \bar{\alpha}^2 o(\|d\|^2) + \frac{1}{2c_k}\|y - x_k\|^2 \\
&= f(x_k) + \bar{\alpha}\left[\nabla f(x_k)^T d + \frac{\bar{\alpha}}{2} d^T\nabla^2 f(x_k) d + \frac{\bar{\alpha}}{2c_k}\|d\|^2 + \bar{\alpha} o(\|d\|^2)\right] \\
&< f(x_k) = f(x_{k+1}),
\end{aligned}$$

which is contradictory to (24). So $f(x_{k+1}) = f^*$ and x_{k+1} is a global minimizer.

In the case of an infinite sequence, we first prove $f_\infty = f^*$, where f_∞ is the limitation of $\{f(x_k)\}$ from the existence result of (i) in Theorem 2, and then prove that $\{x_k\}$ is convergent.

Denote $\bar{x}$ as the accumulated point of a convergent subsequence $\{x_{k_i}\}$ of $\{x_k\}$. Then $f(\bar{x}) = f_\infty$ and $\bar{x} \in D(z,\delta)$ as $D(z,\delta)$ is a bounded and closed set. From (19), we have, for the global minimizer $x^* \in D(z,\delta)$,

$$||x_{k+1} - x^*||^2 \leq ||x_k - x^*||^2 - 2c_k(f(x_{k+1}) - f(x^*))$$

and consequently, for any integer $N > 0$,

$$||x_{N+1} - x^*||^2 + 2\sum_{k=0}^{N} c_k(f(x_{k+1}) - f(x^*)) \leq ||x_0 - x^*||^2.$$

So

$$2\sum_{k=0}^{N} c_k(f(x_{k+1}) - f(x^*)) \leq ||x_0 - x^*||^2,$$

and

$$2\sum_{k=0}^{\infty} c_k(f(x_{k+1}) - f(x^*)) \leq ||x_0 - x^*||^2. \tag{26}$$

By the reduction to absurdity, if $f_\infty > f^*$, then $f(x_{k+1}) - f(x^*) \geq f_\infty - f(x^*) = f(\bar{x}) - f(x^*) > 0$ by the monotonically nonincreasing of $\{f(x_k)\}$. Together with $\sum_{k=0}^{\infty} c_k = \infty$ by our assumption (3), (26) is a contradictory equation. So $f_\infty = f^*$.

Also from (19), we have

$$||x_{k+1} - \bar{x}||^2 \leq ||x_k - \bar{x}||^2 - 2c_k(f(x_{k+1}) - f(\bar{x})) \leq ||x_k - \bar{x}||^2,$$

as $f(x_k) \geq f(\bar{x})$ and $c_k > 0$ for any $k \geq 0$. Furthermore, for any given k_i and any $k \geq k_i$,

$$||x_k - \bar{x}||^2 \leq ||x_{k_i} - \bar{x}||^2,$$

which implies the convergence of $\{x_k\}$ and ends our proof. □

Similar to Definition 5, we consider a more general set defined as follows.

Definition 6 *(Global optimality set)* $D(z, \delta) = \{x \in \mathbb{R}^n \mid ||x - z|| \leq \delta\}$ *is called a global optimality set if it satisfies the following conditions.*
(i) There exists an $x^* \in \mathcal{X}^*$ *such that* $x^* \in D(z, \delta)$.
(ii) $D(z, \delta) \subseteq \mathcal{X}$.
(iii) For any given $\bar{x} \in D(z, \delta)$, $x \in D(z, \delta)$ *whenever* $f(x) < f(\bar{x})$ *and* $||x - \bar{x}|| \leq 2\delta$ *for* $x \in \mathcal{X}$.
(iv) There is no other stationary point in $D(z, \delta)$ *except of the points in* $\mathcal{X}^*$.

The following example shows its reasonability of the definition.

Example 3 *(***Continued of Example 1***) For the problem* $\min_{-0.5 \leq x \leq 1} -x^2$*, it is easily checked that* $D(a, 1-a)$ *with* $\frac{1}{2} < a < 1$ *is a global optimality set.*

Theorem 5 *Under our four assumptions and the sequence generated by CPPA is denoted as* $\{x_k | k = 0, 1, \ldots\}$*, then any accumulation point of* $\{x_k\}$ *is a global minimizer in* $D(z, \delta) \cap \mathcal{X}^*$ *if an initial point* x_0 *is selected from one global optimality set* $D(z, \delta)$.

Similar to the arguments as in the proof of Theorem 4, we get the same result (20). Based on (iv) of Theorem 2 and (iv) of Definition 6, we get the convergent result. The proof is omitted here.

3 A dual initial point for NQPCQC problem

For the NQPCQC problem

$$\begin{aligned} \min \quad & f(x) = x^T Q_0 x + c_0^T x \\ s.t. \quad & g_i(x) = x^T Q_i x + c_i^T x - b_i \leq 0 \quad i = 1, 2, \cdots, m \\ & x \in \mathbb{R}^n, \end{aligned} \tag{27}$$

where Q_0 is an $n \times n$ symmetric matrix with at least one negative eigenvalue, $Q_i \in \mathcal{S}_+^n, i = 1, \ldots, m$ are semi-definite positive symmetric matrices, $c_i \in \mathbb{R}^n, i = 0, 1, \ldots, m$, and $b_i \in \mathbb{R}, i = 1, 2, \ldots, m$.

Let $f(x) = x^T Q_0 x + c_0^T x$ and $\mathcal{X} = \{x \in \mathbb{R}^n | g_i(x) \leq 0, i = 1, 2, \cdots, m\}$ in line with the notations in (1). Theoretically, we assume that $\mathcal{X}$ is bounded and strictly feasible.

Obviously, (27) with the above assumptions meets our four assumptions and is a subclass of (1). Therefore, we can apply our CPPA to this problem. From the discussions in Section 2, the initial point selection is key for finding a global minimizer.

Due to the continuality of $f(x)$, a naive idea is to find a point x_0 such that $f(x_0)$ and f^* are closer enough. As the optimal value of the Lagrangian dual always provides a small duality gap for f^*, the Lagrangian dual problem is considered to find an initial point. We use the techniques of Lu et al [19] to fix the Lagrangian multipliers and the dual optimal value.

For (27), the Lagrangian function is, for any $\lambda \in \mathbb{R}^m_+$,

$$L(x,\lambda) = x^T(Q_0 + \sum_{i=1}^m \lambda_i Q_i)x + (c_0 + \sum_{i=1}^m \lambda_i c_i)^T x - \sum_{i=1}^m \lambda_i b_i.$$

Its Lagrangian dual problem $\max_{\lambda\in\mathbb{R}^m_+} \min_{x\in\mathbb{R}^n} L(x,\lambda)$ is reformulated as a semi-definite positive programming problem as follows, which can be solved in polynomial time,

$$\begin{aligned} \max \quad & t \\ s.t. \quad & \begin{pmatrix} 2(-\sum_{i=1}^m \lambda_i b_i - t) & (c_0 + \sum_{i=1}^m \lambda_i c_i)^T \\ c_0 + \sum_{i=1}^m \lambda_i c_i & 2(Q_0 + \sum_{i=1}^m \lambda_i Q_i) \end{pmatrix} \in \mathcal{S}^{n+1}_+ \\ & t \in \mathbb{R}, \quad \lambda \in \mathbb{R}^m_+. \end{aligned} \tag{28}$$

Based on the assumptions that $\mathcal{X}$ is bounded and strictly feasible, (28) is attainable (see [6] for details). We solve (28) and obtain the corresponding optimal duality value t^*, then solve the following semi-definite positive programming problem to obtain an unique λ^* (see [19] for details.)

$$\begin{aligned} \max \quad & e^T\lambda \\ s.t. \quad & \begin{pmatrix} 2(-\sum_{i=1}^m \lambda_i b_i - t^*) & (c_0 + \sum_{i=1}^m \lambda_i c_i)^T \\ c_0 + \sum_{i=1}^m \lambda_i c_i & 2(Q_0 + \sum_{i=1}^m \lambda_i Q_i) \end{pmatrix} \in \mathcal{S}^{n+1}_+ \\ & \lambda \in \mathbb{R}^m_+. \end{aligned} \tag{29}$$

The Lagrangian dual problem is then equivalent to $\min_{x\in\mathbb{R}^n} L(x,\lambda^*)$ with a convex function $L(x,\lambda^*)$ and its KKT condition is

$$2(Q_0 + \sum_{i=1}^m \lambda_i^* Q_i)y + c_0 + \sum_{i=1}^m \lambda_i^* c_i = 0.$$

To find a closest point to the feasible set $\mathcal{X}$ satisfying the above KKT condition, we solve the following convex quadratic programming problem

$$\begin{aligned} \min \quad & (x-y)^T(x-y) \\ s.t. \quad & g_i(x) = x^T Q_i x + c_i^T x - b_i \le 0 \quad i = 1,2,\cdots,m \\ & 2(Q_0 + \sum_{i=1}^m \lambda_i^* Q_i)y + c_0 + \sum_{i=1}^m \lambda_i^* c_i = 0 \\ & x, y \in \mathbb{R}^n, \end{aligned} \tag{30}$$

and get an optimal solution (x^*, y^*). Then y^* is selected as an initial point for CPPA. We conclude the above initial point selecting procedure as the following algorithm.

Dual Initial Point Algorithm(DIPA):

Input and output: Input an instance of the NQPCQC problem with a bounded feasible domain, and a given error tolerance $\epsilon > 0$. Output an initial point x_0.

S0. Input an instance of the NQPCQC problem and a tolerance $\epsilon > 0$.
S1. Compute (28) and denote its optimal objective value as t^*.
S2. Compute (29) and denote its optimal solution as λ^*.
S3. Solve (30) and denote its optimal solution as (x^*, y^*).
S4. Select $x_0 = y^*$.

When the initial point is selected by DIPA or randomly selected in our CPPA, we denote the algorithm as D-CPPA or R-CPPA respectively. In next section, D-CPPA and R-CPPA are compared on some randomly generated instances.

4 Numerical experiments

In this section, numerical experiments are adapted on some randomly generated instances of the $[0,1]^n$ box constrained quadratic programming (BQP) problem, the nonconvex quadratic programming problem with multiple ellipsoidal constraints (ECQP), and the homogeneous nonconvex quadratic programming problem with homogeneous convex quadratic constraints (HQCQP). The branch-and-bound algorithm (denoted as BB) of [18] is used to give the optimal solution and its objective value and D-CPPA is evaluated comparing with the branch-and-bound algorithm. We also compare D-CPPA with R-CPPA to show the stability and the possibility to get the global minimum.

The algorithms are implemented in Matlab R2018a on a laptop PC with Intel Core i5 (3.10GHz) and 8GB memory. CVX 2.1 is used to solve the convex subproblems (28), (29), (30), and (2). The tolerance of CVX is set to its default accuracy $[\bar{\epsilon}^{\frac{1}{2}}, \bar{\epsilon}^{\frac{1}{2}}, \bar{\epsilon}^{\frac{1}{4}}]$, where $\bar{\epsilon} = 2.22 \times 10^{-16}$ is the machine accuracy. The error tolerance of D-CPPA, BB and R-CPPA remains the same as in [18] with $\epsilon = 5 \times 10^{-4}$ for convenience of the adoption and comparison of the branch-and-bound algorithm.

The eigenvalue decomposition of Q is realized directly using the Matlab function $eig(Q)$. Denote Y_{D-CPPA}, Y_{R-CPPA} and Y_{BB} as the objective values computed by D-CPPA, R-CPPA and BB respectively, and define the relative errors of D-CPPA and R-CPPA by $\frac{|Y_{D-CPPA} - Y_{BB}|}{|Y_{BB}|}$ and $\frac{|Y_{R-CPPA} - Y_{BB}|}{|Y_{BB}|}$ respectively in the following numerical experiments.

4.1 The BQP problem

The BQP problem is an abbreviation of the $[0,1]^n$ box constrained quadratic programming problem as follows.

$$\begin{aligned} \min \quad & f(x) = x^T Q x + c^T x \\ s.t. \quad & x \in [0,1]^n, \end{aligned}$$

where Q is indefinite and $c \in \mathbb{R}^n$.

The generating procedure of test instances is similar to the one in [17]. A symmetric matrix $Q^* \in \mathcal{S}^n$ and a vector $c^* \in \mathbb{R}^n$ are generated with each entry of them is uniformly distributed in $[-100, 100]$. Then compute the eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$ of Q^*. For a given integer r with $1 \leq r < n$, let $v = \frac{1}{2}(\lambda_r + \lambda_{r+1})$. Set $Q = Q^* - vI$ and $c = c^* + \frac{v}{2}e$. It can be seen that Q always has exactly r negative eigenvalues when the instances with $\lambda_r = \lambda_{r+1}$ are dropped.

Table 1 Comparison of D-CPPA and BB on BQP problem

Size	D-CPPA CPU		BB CPU		D-CPPA W	
(n, r)	Average	σ	Average	σ	Relative error	CPU
(20,4)	15.0674	11.5242	7.3267	3.5924	6.85E-03	61.9160
(20,10)	15.4728	9.0726	22.0199	20.0264	3.24E-02	33.5230
(20,16)	6.3771	1.8460	24.6449	17.4734	3.86E-02	9.6318
(30,6)	19.2165	8.8044	27.4649	20.0285	2.52E-06	50.1690
(30,15)	32.0578	16.1136	194.4803	194.2549	5.60E-02	68.5840
(30,24)	8.8780	2.0864	134.0300	240.7631	2.34E-02	12.9690
(50,10)	47.7690	27.4143	584.4270	482.5466	1.16E-02	120.0200
(50,25)	44.8632	19.6245	2382.0680	1171.8684	1.49E-02	85.0710
(50,40)	12.3352	3.6887	2553.6600	1080.7102	3.45E-02	20.4580

σ: the standard deviation of the CPU time.
D-CPPA CPU: CPU time in second for D-CPPA.
BB CPU: CPU time in second for BB.
D-CPPA W: the worst case of D-CPPA.

Various configurations of (n, r) are set for Q, where n is the dimension of variables and r represents the number of negative eigenvalues of Q. For a given n, set $r = 0.2n$, $0.5n$, and $0.8n$. For each (n,r), 20 instances are generated in our experiments. From our experience of numerical experiments, the CPU time of BB is more than one hour when n is as large as n=100, and the performance of D-CPPA, R-CPPA and BB keeps a similar trend as n increases. Thus we set n = 20, 30, 50. In addition, under the background of the above generating procedure of test instances and parameters, the absolute value of the optimal value computed via BB, $|Y_{BB}|$ is mostly between 10^2 and 10^3.

Numerical results for the comparison of D-CPPA and BB in [18] are listed in Table 1. Generally, the average CPU time of D-CPPA is much less than that of BB, especially when (n, r) is in a large size. For example, the average CPU time of D-CPPA is 44.8632s and that of BB is 2382.0680s when $(n, r) = (50, 25)$. D-CPPA is acceptable as an approximation algorithm for the BQP problem with all the worst-case relative errors are within the scale of 10^{-2}. Besides, the CPU time of D-CPPA is also more stable than that of BB in terms of the standard deviation σ. For the D-CPPA, 70% of the total 180 instances get their relative error within $\epsilon = 5 \times 10^{-4}$.

Numerical results for comparison of D-CPPA and R-CPPA are listed in Table 2. An accuracy index, abbreviated as Acc and defined as the percentage of instances with their relative error within the error tolerance ϵ, is used for the comparison of the two algorithms.

In Table 2, we split the CPU time of D-CPPA into two parts T_1 and T_2, where T_2 represents the CPU time of getting an initial point using DIPA proposed in Section 3, T_1 represents the CPU time of CPPA with the obtained initial point, and $T_2 + T_1$ is the CPU time of D-CPPA. For R-CPPA, a random initial point is selected with uniformly distribution in $[-1, 1]^n$.

From Table 2, we find that the initial point obtained by DIPA is much closer than the random initial point to the global optimal solution of BB, and more importantly, the accuracy of D-CPPA for different (n, r) is much higher than that of R-CPPA. In general, the accuracy of D-CPPA is 70% and R-CPPA is 36.67% of the totally 180 randomized instances. So the initial point selected by DIPA makes CPPA more effective. More specifically, the number of negative eigenvalues affects the accuracy of D-CPPA and the accuracy of D-CPPA decreases as the number of negative eigenvalues r increases. For example, when $n = 30, r = 6, 15, 24$, the corresponding accuracy of D-CPPA shows a downward trend of 100%, 70%, 55%. In addition, the average number of iterations of D-CPPA is much less than that of R-CPPA and the CPU times of both algorithms are not much different. Moreover, the CPU time T_1 of CPPA occupies most of the total CPU time of D-CPPA, while the CPU time T_2 of DIPA only occupies a small part. Based on the above results, we conclude that D-CPPA is more efficient than R-CPPA on BQP problem.

Table 2 Comparison of D-CPPA and R-CPPA on BQP problem

Size	D-CPPA					R-CPPA			
(n, r)	Iters	T_1	T_2	Dist	Acc	Iters	CPU	Dist	Acc
(20,4)	15.8500	12.7811	2.2863	0.7168	95.00	21.3000	17.1411	4.0094	80.00
(20,10)	17.2500	13.3771	2.0958	0.9013	70.00	39.1500	30.0145	4.3006	40.00
(20,16)	5.6000	4.2045	2.1725	0.8281	60.00	12.5000	9.6974	4.5735	0.00
(30,6)	17.3000	16.7215	2.4950	0.8492	100.00	26.6500	25.6280	4.8212	85.00
(30,15)	31.6500	29.4686	2.5892	1.2452	70.00	39.7000	36.7099	5.2104	30.00
(30,24)	6.9000	6.3950	2.4831	1.1420	55.00	14.6000	13.5665	5.6980	0.00
(50,10)	35.5500	44.6218	3.1472	1.2292	95.00	39.8500	49.7705	6.2840	90.00
(50,25)	34.1000	41.6914	3.1718	1.5222	45.00	71.7500	87.4324	6.8874	5.00
(50,40)	7.5500	9.2120	3.1233	1.5005	35.00	16.2500	19.9017	7.1853	0.00

CPU time is in second.
Dist: Euclidean distance between the initial point and the optimal solution computed by BB.
Acc: percentage of the relative error within $\epsilon = 5 \times 10^{-4}$.
Iters: number of iterations.

4.2 The ECQP problem

The ECQP problem is an abbreviation of the nonconvex quadratic programming problem with ellipsoidal constraints as follows.

$$\begin{aligned} \min \quad & f(x) = x^T Q x + c^T x \\ s.t. \quad & x^T Q_i x + c_i^T x - b_i \le 0, i = 1, 2, \ldots, m \\ & x \in \mathbb{R}^n, \end{aligned}$$

where Q is indefinite, $c \in \mathbb{R}^n$, Q_i is definite positive, $c_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for $i = 1, 2, \ldots, m$.

The procedure of randomly generating the test instances is the same as that in [17]. The matrices Q and Q_i are generated with the following 3 steps. Firstly, matrices Q^* and Q_i^*, $i = 1, 2, \cdots, m$ are uniformly generated with each entry being in $[-1, 1]$. Then $Q^* = PD^*P^T$ and $Q_i^* = P_i D_i^* P_i^T$ are decomposed. Diagonal matrices D and D_i are uniformly generated with the first r diagonal entries of D being in $[-10, 0]^r$, the rest diagonal entries of D being in $[0, 10]^{n-r}$, and diagonal entries of D_i being in $[1, 100]^n$. Finally, let $Q = PDP^T$ and $Q_i = P_i D_i P_i^T$. c_i is uniformly generated from $[-100, 100]^n$, b_i is uniformly generated from $[1, 50]$ for $i = 1, \cdots, m$ and c is uniformly generated from $[-1, 1]^n$. It is easy to verify that all instances generated by the above procedures are strictly feasible with $x = 0$ being a strictly feasible solution, i.e. the Slater's condition holds. Since Q_i is positive definite, $i = 1, 2, \cdots, m$, then the feasible domain is bounded. So the Lagrangian dual problem is solvable. In our computation, few instances have zero duality gap between the primal ECQP and its Lagrangian dual, especially when n is large. The instances with positive gap are selected in our numerical experiments.

For each configuration (n,m,r), where n is the dimension of the variables, m is the number of constraints, and r represents the number of negative eigenvalues of Q, set $n = 20$, 30 and 50, m=0.5n and m=n, and $r = 0.2n$, $0.5n$ and $0.8n$. For each (n,m,r), 20 instances are randomly generated, i.e., 360 instances in total, to test the performance of the algorithms. In this case, the absolute value of the optimal value computed via BB, $|Y_{BB}|$ is mostly less than 10.

The comparison of BB and D-CPPA is listed in Table 3. The dominance of D-CPPA in CPU time appears when (n, m, r) gets large. For example, when $n = 50, m = 50, r = 40$, the average CPU time of D-CPPA is 28.7227s and BB is 1207.3964s. D-CPPA is acceptable as an approximation algorithm for the ECQP problem with all the worst-case relative errors are in scale of 10^{-4} to 10^{-1}. Besides, the CPU time of D-CPPA is also more stable than that of BB from the standard deviation.

Table 3 Comparison of D-CPPA and BB on ECQP problem

Size	D-CPPA CPU		BB CPU		D-CPPA W	
(n,m,r)	Average	σ	Average	σ	Relative error	CPU
(20,10,4)	9.1895	3.9332	2.5845	2.6701	1.96E-04	23.5980
(20,10,10)	9.3454	3.9669	5.6721	4.1041	3.85E-04	17.9240
(20,10,16)	11.2753	5.8086	6.9655	5.9261	4.14E-04	31.9820
(20,20,4)	10.4956	3.5005	8.9651	14.8548	1.35E-01	21.7390
(20,20,10)	11.8863	6.1823	21.7167	65.5153	7.42E-02	28.8500
(20,20,16)	12.8256	8.6865	34.9873	51.6375	1.36E-02	45.5700
(30,15,6)	15.9402	15.3578	7.6827	8.0969	2.17E-01	30.5760
(30,15,15)	13.4529	9.8570	19.9115	24.4300	2.98E-04	48.7790
(30,15,24)	9.1082	5.3818	32.9166	91.4044	4.36E-04	26.4130
(30,30,6)	15.8071	5.9219	25.9032	28.9929	2.56E-03	32.5740
(30,30,15)	19.1165	7.5710	271.4170	596.5184	3.84E-02	33.8990
(30,30,24)	15.7306	6.3600	332.5909	675.4637	4.32E-03	31.3850
(50,25,10)	25.4016	8.7970	50.4508	44.2209	6.08E-04	45.5040
(50,25,25)	15.8936	3.5856	47.4300	80.2643	4.09E-04	23.6350
(50,25,40)	16.5230	7.1435	37.2739	32.2867	1.17E-04	43.4850
(50,50,10)	38.3019	12.5743	344.4689	300.0124	8.85E-04	62.7560
(50,50,25)	33.8834	11.5362	858.2318	1144.2082	4.01E-04	59.5740
(50,50,40)	28.7227	8.9746	1207.3964	1979.0644	1.51E-02	59.1190

σ represents the standard deviation of the CPU time
D-CPPA CPU: CPU time in second for D-CPPA.
BB CPU: CPU time in second for BB.
D-CPPA W: the worst case of D-CPPA.

The accuracy of D-CPPA and R-CPPA for different (n, m, r) is listed in Table 4. We still split the CPU time of D-CPPA into two parts T_1 and T_2, where T_2 represents the CPU time of calculating an initial point using DIPA, T_1 represents the CPU time of CPPA with the initial point obtained by DIPA, and $T_2 + T_1$ is the CPU time of D-CPPA. For R-CPPA, a randomly initial point is selected with uniform distribution in $[-1, 1]^n$. The initial point obtained by DIPA is much closer than the randomly initial point to the global optimal solution, and the accuracy of D-CPPA for most instances is higher than that of R-CPPA. In general, the accuracy of D-CPPA is 94.17% and R-CPPA is 85.28%. In addition, for different (n, m, r) in Table 4, the average number of iterations and the time T_1 of D-CPPA is always less than those of R-CPPA. Therefore, the initial points given by DIPA make CPPA more effective to get a global minimizer. However, the occupation of T_2 in CPU time for

Table 4 Comparison of D-CPPA and R-CPPA on ECQP problem

Size	D-CPPA					R-CPPA			
(n,m,r)	Iters	T_1	T_2	Dist	Acc	Iters	CPU	Dist	Acc
(20,10,4)	10.5000	7.2224	1.9671	0.6523	100.00	13.8000	9.4891	2.8419	75.00
(20,10,10)	11.4000	7.5315	1.8139	0.4755	100.00	26.3500	17.6356	2.7818	95.00
(20,10,16)	13.6000	9.2109	2.0643	0.5037	100.00	27.1000	18.4499	2.7500	90.00
(20,20,4)	9.1500	8.2847	2.2109	0.4788	80.00	17.0500	15.3557	2.6969	60.00
(20,20,10)	10.6500	9.6554	2.2309	0.4035	85.00	18.3000	16.4396	2.6361	65.00
(20,20,16)	11.8000	10.5550	2.2705	0.3909	95.00	23.1000	20.4938	2.7535	90.00
(30,15,6)	17.1000	13.6334	2.3069	0.5790	90.00	21.2500	16.8623	3.3284	80.00
(30,15,15)	14.0000	11.1521	2.3008	0.4249	100.00	25.1500	20.0648	3.3692	95.00
(30,15,24)	8.5500	6.7616	2.3457	0.2585	100.00	26.3000	21.0074	3.3734	100.00
(30,30,6)	11.2500	12.9593	2.8477	0.4772	85.00	19.1500	22.0659	3.2542	70.00
(30,30,15)	14.1500	16.2077	2.9087	0.4806	85.00	23.3500	26.8015	3.2876	70.00
(30,30,24)	11.2500	12.8061	2.9244	0.3465	95.00	27.3500	31.3675	3.2349	95.00
(50,25,10)	17.6500	19.2200	6.1817	0.6530	95.00	30.6500	33.0688	4.3471	90.00
(50,25,25)	8.9500	9.6382	6.2555	0.3138	100.00	22.0500	23.7217	4.2989	100.00
(50,25,40)	9.0500	9.8387	6.6843	0.2727	100.00	25.7500	28.9166	4.2379	100.00
(50,50,10)	15.1500	27.1295	11.1724	0.4867	95.00	21.3500	37.8773	4.1461	75.00
(50,50,25)	12.9000	22.8199	11.0634	0.4152	100.00	23.6500	41.6770	4.1779	95.00
(50,50,40)	11.4000	18.1998	10.5227	0.3431	90.00	23.9500	37.2235	4.1847	90.00

CPU time is in second.
Dist: Euclidean distance between the initial point and the optimal solution computed by BB.
Acc: percentage of the relative error within $\epsilon = 5 \times 10^{-4}$.
Iters: number of iterations.

DIPA increases as n gets large. So improving the running time of DIPA or designing a new strategy for getting good initial points are important for future study.

4.3 The HQCQP problem

The HQCQP problem is an abbreviation of the homogeneous nonconvex quadratic programming problem with convex homogeneous quadratic constraints which has the following form

$$\begin{aligned} \min \quad & f(x) = x^T Q x \\ s.t. \quad & g_i(x) = x^T Q_i x - 1 \leq 0 \quad i = 1, 2, \cdots, m \\ & x \in \mathbb{R}^n \end{aligned} \tag{31}$$

where Q is indefinite, Q_i is semidefinite positive for $i = 1, \cdots, m$. This problem is an important subclass of the NQPCQC problem and has lots of applications in communication systems [17,21].

Test instances are generated as follows. Q_i, $i = 1, \cdots, m$ in the constraints, are generated in the following steps. Firstly, generate a symmetric matrix B_i with each entry being uniform distribution in $[-1, 1]$. Then compute the eigenvalue decomposition $B_i = VDV^T$. Finally, get a diagonal matrix D^* by setting $D^*_{ii} = |D_{ii}|$ for $i = 1, \cdots, n$, and set $Q_i = VD^*V^T$. Q in the objective function is generated with the same procedure of that for BQP.

For various configurations (n,m,r), let n= 20, 30 and 50, m=0.5n and m=n, and $r = 0.2n$, 0.5n and 0.8n. For each (n,m,r), 20 random instances are selected, hence 360 instances are in total. Based on the above procedure, the magnitude of $|Y_{BB}|$ is about 10^{-1}.

By (30), 0 is always an optimal solution and hence is an initial point obtained by DIPA. Meanwhile, 0 is also a feasible solution for (31). By the convex proximal model (2), it is obvious that the next iterative point is still 0. Now the CPPA always outputs a stationary point 0. Hinted at a small perturbation at 0 in Example 1, which makes the CPPA with the perturbed initial point getting a local or global minimizer, our CPPA is implemented by selecting an uniformly distributed initial point in $[-0.001, 0.001]^n$, which is also denoted as D-CPPA.

Comparison the performance of BB and D-CPPA is presented in Table 5. The similar trend as those for the BQP and ECQP problems is seen. The dominance of D-CPPA in CPU time appears when (n, m, r) gets large. For example, when $n = 50, m = 25, r = 40$, the average CPU time of D-CPPA is 22.6164s and BB is 194.4116s. D-CPPA is acceptable as an approximation algorithm for the HQCQP problem with all the worst-case relative errors are within scale of 10^{-2}. The CPU time of D-CPPA is also more stable than that of BB from the standard deviation in Table 5.

Table 5 Comparison of D-CPPA and BB on HQCQP problems

Size	D-CPPA CPU		BB CPU		D-CPPA W	
(n, m, r)	Average	σ	Average	σ	Relative error	CPU
(20,10,4)	3.9090	0.8128	1.4931	0.9508	2.06E-03	5.9552
(20,10,10)	8.4289	4.6900	3.3880	3.3594	7.06E-02	23.1550
(20,10,16)	10.5109	3.7000	7.9001	12.9227	4.88E-02	22.3910
(20,20,4)	6.3858	4.3404	1.3681	1.0484	1.89E-03	22.3810
(20,20,10)	12.5840	5.7581	7.8460	8.3531	1.20E-02	22.7170
(20,20,16)	15.4745	7.4723	22.7192	23.1020	5.52E-02	38.4990
(30,15,6)	5.6777	1.8936	2.7104	1.8656	8.09E-03	11.6720
(30,15,15)	12.3132	6.0897	7.5080	7.8917	2.09E-02	28.1200
(30,15,24)	18.0328	13.6175	29.0698	54.7022	2.86E-02	68.3810
(30,30,6)	8.2492	4.0212	2.7975	2.2580	6.37E-03	20.7780
(30,30,15)	12.4959	5.4398	23.4776	36.4774	8.38E-03	27.9570
(30,30,24)	23.7293	12.9174	138.4013	236.4851	3.32E-02	58.1250
(50,25,10)	9.1327	6.9606	11.1673	11.8059	4.36E-02	35.9700
(50,25,25)	18.3331	7.7355	77.2613	103.6280	4.10E-02	39.6460
(50,25,40)	22.6164	9.2301	194.4116	263.9626	3.08E-02	46.0040
(50,50,10)	13.2330	5.5086	14.5774	11.4913	4.18E-02	26.7090
(50,50,25)	21.5481	6.7176	91.6895	100.1529	1.98E-02	30.0930
(50,50,40)	45.0248	25.6568	319.9264	292.7532	6.45E-03	120.3900

σ represents the standard deviation of the CPU time
D-CPPA CPU: CPU time in second for D-CPPA.
BB CPU: CPU time in second for BB.
D-CPPA W: the worst case of D-CPPA.

For R-CPPA, the initial point is selected with uniform distribution in $[-1, 1]^n$. The randomly initial point of D-CPPA is much closer to the global optimal solution than the random initial point for R-CPPA as shown in Table 6. Consequently, the accuracy of D-CPPA is higher than that of R-CPPA. In general, the accuracy of D-CPPA is 57.78% and R-CPPA is 40.56%. Therefore selecting an initial point near a global minimizer is helpful for global optimization. In addition, the average number of iterations and the CPU time of D-CPPA is less than those of R-CPPA for most cases of (n, m, r).

5 Conclusions

Based on the above theoretical and numerical results, we give the following conclusions. D-CPPA costs less CPU time than that of BB when the size of the problem gets large, can be used as an approximation algorithm for the NQPCQC problem,

Table 6 Comparison of D-CPPA and R-CPPA on HQCQP problem

Size	D-CPPA				R-CPPA			
(n, m, r)	Iters	CPU	Dist	Acc	Iters	CPU	Dist	Acc
(20,10,4)	6.1500	3.9090	0.7515	90.00	9.7500	6.0674	2.6481	80.00
(20,10,10)	13.6000	8.4289	0.8097	60.00	23.0500	14.3152	2.6927	50.00
(20,10,16)	17.1000	10.5109	0.8615	70.00	39.6500	24.3074	2.7304	50.00
(20,20,4)	7.6000	6.3858	0.7348	80.00	12.4500	10.4111	2.6714	65.00
(20,20,10)	14.6500	12.5840	0.7686	70.00	22.6000	19.1127	2.8108	45.00
(20,20,16)	18.5500	15.4745	0.7925	60.00	33.4000	27.8431	2.6753	45.00
(30,15,6)	7.6000	5.6777	0.6899	70.00	15.3500	11.4225	3.2740	35.00
(30,15,15)	16.2000	12.3132	0.7325	40.00	24.6500	18.5576	3.3500	30.00
(30,15,24)	24.6000	18.0328	0.7494	55.00	32.7000	23.9814	3.2874	35.00
(30,30,6)	7.6500	8.2492	0.6784	85.00	13.2500	14.3118	3.2815	65.00
(30,30,15)	11.4500	12.4959	0.6957	65.00	24.9500	27.1242	3.1736	45.00
(30,30,24)	22.1000	23.7293	0.7133	35.00	34.7500	37.4680	3.2831	20.00
(50,25,10)	9.1500	9.1327	0.6154	55.00	20.4000	20.6321	3.9856	45.00
(50,25,25)	18.0000	18.3331	0.6363	35.00	33.1500	33.9726	4.1850	20.00
(50,25,40)	20.5500	22.6164	0.6464	30.00	45.4000	47.6821	4.2048	15.00
(50,50,10)	7.9000	13.2330	0.6001	55.00	17.1000	28.6951	4.0821	45.00
(50,50,25)	12.7000	21.5481	0.6147	55.00	32.0500	54.5036	4.2484	30.00
(50,50,40)	23.8500	45.0248	0.6222	30.00	45.5500	82.6975	4.1283	10.00

CPU time is in second.
Dist: Euclidean distance between the initial point and the optimal solution computed by BB.
Acc: percentage of the relative error within $\epsilon = 5 \times 10^{-4}$.
Iters: number of iterations.

and is more stable in CPU time than that of BB. To globally solve (1) using our CPPA, the key point is to have a good initial point near a global minimizer. Overall, the proximal point algorithm is an effective tool for globally solving a nonconvex optimization problem provided with the consideration of the initial point selection.

References

1. Apolinário H.C.F., Papa Quiroz E.A., Oliveira P.R., A scalarization proximal point method for quasiconvex multiobjective minimization, J. Glob. Optim, 64, 79-96(2016) https://doi.org/10.1007/s10898-015-0367-3
2. Bauschke H.H., Combettes P.L., Convex Analysis and Monotone Operator Theory in Hilbert Spaces, 399-411. Springer, New York(2011)
3. Bento G.C., Ferreira O.P., Sousa Junior V.L., Proximal point method for a special class of nonconvex multiobjective optimization functions, Optim. Lett, 12, 311-320(2018) https://doi.org/10.1007/s11590-017-1114-0
4. Bertsekas D.P., Tseng P., Partial proximal minimization algorithms for convex programming, SIAM J. Optim, 4, 551-572(1994) https://doi.org/10.1137/0804031
5. Bertsekas D.P., Convex Optimization Algorithms, 233-300. Athena Scientific, Massachusetts (2015)
6. Fang S.-C., Xing W., Linear Conic Optimiation (in Chinese), 142-172. Science Press, Beijing, China (2013)
7. Fortin C., Wolkowicz H., The trust region subproblem and semidefinite programming, Optimization methods and software, 19, 41-67(2004) https://doi.org/10.1080/10556780410001647186
8. Fuduli A., Gaudioso M., Giallombardo G., Minimizing nonconvex nonsmooth functions via cutting planes and proximity control, SIAM J. Optim, 14, 743-756(2004) https://doi.org/10.1137/s1052623402411459
9. Fukushima M., Mine H., A generalized proximal point algorithm for certain nonconvex minimization problems, International Journal Of Systems Science, 12, 989-1000(1981) https://doi.org/10.1080/00207728108963798

10. Hajinezhad D., Shi Q., Alternating direction method of multipliers for a class of nonconvex bilinear optimization: convergence analysis and applications, J. Glob. Optim, 70, 261-288(2018)
https://doi.org/10.1007/s10898-017-0594-x
11. Hajinezhad D., Hong M., Perturbed proximal primal-dual algorithm for nonconvex nonsmooth optimization, Math. Program, 176, 207-245(2019)
https://doi.org/10.1007/s10107-019-01365-4
12. Hintermüller M., A proximal bundle method based on approximate subgradients, Computational Optimization and Applications, 20, 245-266(2001)
https://doi.org/10.1023/A:1011259017643
13. Monjezi N.H., Nobakhtian S., A filter proximal bundle method for nonsmooth nonconvex constrained optimization, J. Glob. Optim, 79, 1-37(2021)
https://doi.org/10.1007/s10898-020-00939-3
14. Monjezi N.H., Nobakhtian S., A new infeasible proximal bundle algorithm for nonsmooth nonconvex constrained optimization, Computational Optimization and Applications, 74, 443-480(2019)
https://doi.org/10.1007/s10589-019-00115-8
15. Kaplan A., Tichatschke R., Proximal point methods and nonconvex optimization, J. Glob. Optim, 13, 389-406(1998)
https://doi.org/10.1023/A:1008321423879
16. Kiwiel K.C., Restricted step and Levenberg-Marquardt techniques in proximal bundle methods for non-convex nondifferentiable optimization, SIAM J. Optim, 6, 227-249(1996)
https://doi.org/10.1137/0806013
17. Lu C., Deng Z., Jin Q., An eigenvalue decomposition based branch-and-bound algorithm for nonconvex quadratic programming problems with convex quadratic constraints, J. Glob. Optim, 67, 475-493(2017)
https://doi.org/10.1007/s10898-016-0436-2
18. Lu C., Deng Z., Zhou J., Guo X., A sensitive-eigenvector based global algorithm for quadratically constrained quadratic programming, J. Glob. Optim, 73, 371-388(2019)
https://doi.org/10.1007/s10898-018-0726-y
19. 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)
https://doi.org/10.1137/100793955
20. Martinet B., Régularisation d'inéquations variationelles par approximations successives, Revue francaise d'Informatique Et de Recherche operationnelle(RIRO), 4, 154-158(1970)
21. Matskani E., Sidiropoulos N.D., Luo Z.-Q., Tassiulas L., Convex approximation techniques for joint multiuser downlink beamforming and admission control, IEEE Transactions on Wireless Communications, 7, 2682-2693(2008)
https://doi.org/10.1109/TWC.2008.070104
22. Nikolova M., Tan P., Alternating structure-adapted proximal gradient decent for nonconvex nonsmooth block-regularized problems, SIAM J. Optim, 29, 2053-2078(2019)
https://doi.org/10.1137/17M1142624
23. Pardalos P.M., Vavasis S. A., Quadratic programming with one negative eigenvalue is NP-Hard, J. Glob. Optim, 1, 15-22(1991)
https://doi.org/10.1007/BF00120662
24. Peng Z., Chen J., Zhu W., A proximal alternating direction method of multipliers for a minimization problem with nonconvex constraints, J. Glob. Optim, 62, 711-728(2015)
https://doi.org/10.1007/s10898-015-0287-2
25. Rockafellar R.T., Monotone operators and the proximal point algorithm, SIAM J. Control Optim, 14, 877-898(1976)
https://doi.org/10.1137/0314056
26. Sun T., Jiang H., Cheng L., Global convergence of proximal iteratively reweighted algorithm, J Glob. Optim, 68, 815-826(2017)
https://doi.org/10.1007/s10898-017-0507-z
27. Vandenbussche D., Nemhauser G.L., A polyhedral study of nonconvex quadratic programs with box constraints, Math. Program, 102, 531-557(2005)
https://doi.org/10.1007/s10107-004-0549-0
28. Wu Z., Li M., General inertial proximal gradient method for a class of nonconvex nonsmooth optimization problems, Computational Optimization and Applications, 73, 129-158(2019)
https://doi.org/10.1007/s10589-019-00073-1

29. Zhang J., Luo Z., A proximal alternating direction method of multiplier for linearly constrained nonconvex minimization, SIAM J. Optim, 30, 2272-2302(2020) https://doi.org/10.1137/19M1242276