Bulletin of the Iranian Mathematical Society (2020) 46:865–892
https://doi.org/10.1007/s41980-019-00298-0

ORIGINAL PAPER

Two New Customized Proximal Point Algorithms Without Relaxation for Linearly Constrained Convex Optimization

Binqian Jiang[1] · Zheng Peng[2] · Kangkang Deng[2]

Received: 4 May 2018 / Revised: 31 August 2019 / Accepted: 14 September 2019 /
Published online: 23 October 2019

Abstract

In this paper, we propose a new customized proximal point algorithm for linearly constrained convex optimization problem, and further extend the proposed method to separable convex optimization problem. Unlike the existing customized proximal point algorithms, the proposed algorithms do not involve relaxation step, but still ensure the convergence. We obtain the particular iteration schemes and the unified variational inequality perspective. The global convergence and $\mathcal{O}(1/k)$-convergence rate of the proposed methods are investigated under some mild assumptions. Numerical experiments show that compared to some state-of-the-art methods, the proposed methods are effective.

Keywords Convex optimization · Separable convex optimization · Customized proximal point algorithm · Global convergence · $\mathcal{O}(1/k)$-convergence rate

Mathematics Subject Classification 90C25 · 90C30 · 65K10

Communicated by Majid Soleimani-damaneh.

This work was supported by the Natural Science Foundation of China with Grants 11571074 and 11726505.

✉ Zheng Peng
pzheng@fzu.edu.cn

Binqian Jiang
283736184@qq.com

Kangkang Deng
freedeng1208@gmail.com

[1] School of Economics and Management, Fuzhou University, Fuzhou 350108, China

[2] School of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China

1 Introduction

1.1 Proximal Point Algorithm

Proximal point algorithm (PPA for short) dates back to Moreau [1], which was extended by Rockafellar [2], Bertsekas and Tseng [3], and Kaplan and Tichatschke [4], etc. It was introduced to optimization community by Martinet [5]. The PPA possesses a robust convergence for very general problems in finite and infinite-dimensions (Kassay [6] and Glüer [7]), and is also the basis for the splitting methods (Spingarn [8] and Han [9]).

Consider a convex minimization problem with linear constraints of the form

$$\min\{\theta(x) | Ax = b, x \in \mathcal{X}\}, \tag{1.1}$$

where $\mathcal{X} \subset \mathbb{R}^n$ is a bounded closed convex and nonempty set, $\theta : \mathcal{X} \to \mathbb{R}$ is a convex (not necessarily smooth) function, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. The solution set of (1.1), denoted by $\mathcal{X}^*$, is assumed to be nonempty.

For solving problem (1.1), the primal-dual hybrid gradient (PDHG) algorithm proposed by Zhu and Chan [10] has the following iterative scheme:

$$\begin{cases} x^{k+1} = \arg\min_{x\in\mathcal{X}} \left\{\theta(x) + \frac{r}{2}\left\|(x - x^k) - \frac{1}{r}A^{\mathrm{T}}\lambda^k\right\|^2\right\}, \\ \lambda^{k+1} = \lambda^k - \frac{1}{t}(Ax^{k+1} - b), \end{cases} \tag{1.2}$$

where $\lambda \in \mathbb{R}^m$ is the Lagrange multiplier vector, $r, t > 0$ are iteration parameters. Let $w := (x, \lambda)$, and $\mathcal{W} := \mathcal{X} \times \mathbb{R}^m$. Then, by the first optimality condition of iteration (1.2), for $w^{k+1} \in \mathcal{W}$ we have that

$$\theta(x) - \theta(x^{k+1}) + (w - w^{k+1})^{\mathrm{T}}\Big[F(w^{k+1}) + P(w^{k+1} - w^k)\Big] \geq 0, \quad \forall\, w \in \mathcal{W}, \tag{1.3}$$

where

$$F(w) = \begin{pmatrix} -A^{\mathrm{T}}\lambda \\ Ax - b \end{pmatrix}, \quad P = \begin{pmatrix} rI_n & A^{\mathrm{T}} \\ 0_{m\times n} & tI_m \end{pmatrix},$$

and I_n denotes a $n \times n$-dimensional identity matrix, $0_{m\times n}$ is a $m \times n$-dimensional null matrix. However, He [11] showed that the PDHG algorithm could not be necessarily convergent, and gave a counter example, see Sect. 3 of [11].

He, Fu and Jiang [12] proposed a PPA using a linear proximal term (LPPA). For solving (1.1), the LPPA (Procedure 4.1 in [12]) first produces a predictor $\tilde{w}^k = (\tilde{x}^k, \tilde{\lambda}^k)$ via

$$\begin{cases} \tilde{x}^k = \arg\min_{x\in\mathcal{X}} \left\{\theta(x) + \frac{1}{2}\left\|A(x - x^k) - \lambda^k\right\|^2\right\}, \\ \tilde{\lambda}^k = \lambda^k - (A\tilde{x}^k - b). \end{cases} \tag{1.4}$$

By the first-order optimality conditions of (1.4), for $\tilde{w}^k \in \mathcal{W}$, we have

$$\theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^{\mathrm{T}}\Big[F(\tilde{w}^k) + P_1(\tilde{w}^k - w^k)\Big] \geq 0, \quad \forall\, w \in \mathcal{W}, \tag{1.5}$$

where

$$F(w) = \begin{pmatrix} -A^{\mathrm{T}}\lambda \\ Ax - b \end{pmatrix}, \quad P_1 = \begin{pmatrix} A^{\mathrm{T}}A & A^{\mathrm{T}} \\ -A & I_m \end{pmatrix}. \tag{1.6}$$

To ensure the convergence, the new iterate $w^{k+1} = (x^{k+1}, \lambda^{k+1})$ of LPPA is generated by a simple relaxation step

$$w^{k+1} = w^k - \rho_k P_1(w^k - \tilde{w}^k),$$

where $\rho_k = \tau\rho_k^*$ and

$$\rho_k^* = \frac{(w^k - \tilde{w}^k)^{\mathrm{T}} P_1(w^k - \tilde{w}^k)}{\|P_1(w^k - \tilde{w}^k)\|^2}, \ \tau \in [1, 2).$$

He, Yuan and Zhang [13] proposed a customized proximal point algorithm (CPPA) by using matrix P_2 instead of matrix P_1 in (1.5), where

$$P_2 = \begin{pmatrix} rI_n & -A^{\mathrm{T}} \\ -A & tI_m \end{pmatrix}.$$

To guarantee convergence, the matrix P_2 should be positive definite. As a consequence, it requires $rt > \rho(A^{\mathrm{T}}A)$, where $\rho(A^{\mathrm{T}}A)$ is the spectral radius of matrix $A^{\mathrm{T}}A$. The corresponding iterative scheme of CPPA first produces a predictor $\tilde{w}^k = (\tilde{x}^k, \tilde{\lambda}^k)$ via

$$\begin{cases} \tilde{\lambda}^k = \lambda^k - \frac{1}{t}(Ax^k - b), \\ \tilde{x}^k = \arg\min_{x\in\mathcal{X}} \left\{\theta(x) + \frac{r}{2}\left\|(x - x^k) - \frac{1}{r}A^{\mathrm{T}}(2\tilde{\lambda}^k - \lambda^k)\right\|^2\right\}, \end{cases} \tag{1.7}$$

and then takes a simple relaxation step

$$w^{k+1} = w^k - \gamma(w^k - \tilde{w}^k), \tag{1.8}$$

where $\gamma \in (0, 2)$ is a relaxation factor. Various numerical experiments show that $\gamma > 1$ provides the fast convergence. For distinction in this paper, the CPPA (1.7)-(1.8) with $\gamma = 1$ is called original-CPPA or CPPA for short, while the CPPA with $\gamma > 1$ is called relax-CPPA or rCPPA for short.

The augmented Lagrangian method of multipliers (ALM) is another efficient method for convex optimization, see Hestenes [14] and Powell [15]. Let $\beta > 0$ be a penalty parameter of ALM, the iterative scheme of the ALM to the problem (1.9) is

$$\begin{cases} x^{k+1} = \arg\min_{x\in\mathcal{X}} \left\{\theta(x) + \frac{\beta}{2}\left\|Ax - b - \frac{1}{\beta}\lambda^k\right\|^2\right\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} - b). \end{cases}$$

Rockafellar [16] showed that the ALM is exactly an application of PPA to the dual problem of (1.1), owning the compact inequality (1.3) where P is replaced by

$$P_3 = \begin{pmatrix} 0_{n\times n}, & 0_{n\times m} \\ 0_{m\times n} & \frac{1}{\beta}I_m \end{pmatrix}.$$

However, He, Yuan and Zhang [13] showed that the CPPA is more efficient than the ALM when the objective function $\theta(x)$ is simple in the sense that its resolvent operator of $\theta(x)$, defined by $(I + \frac{1}{r}\partial\theta)^{-1}$, has a close-form representation, where $\partial\theta$ is the subdifferential of θ.

1.2 Proximal Point Algorithm for Separable Problem

Consider a separable convex optimization problem of the form

$$\min \left\{\theta_1(x) + \theta_2(y) \middle| Ax + By = b, x \in \mathcal{X}, y \in \mathcal{Y}\right\}, \tag{1.9}$$

where $\mathcal{X} \subset \mathbb{R}^{n_1}$ and $\mathcal{Y} \subset \mathbb{R}^{n_2}$ are bounded closed and convex nonempty sets, $\theta_1 : \mathcal{X} \to \mathbb{R}$ and $\theta_2 : \mathcal{Y} \to \mathbb{R}$ are convex (not necessarily smooth) functions, $A \in \mathbb{R}^{m\times n_1}$, $B \in \mathbb{R}^{m\times n_2}$ and $b \in \mathbb{R}^m$. The solution set of (1.9), denoted by $\mathcal{X}^* \times \mathcal{Y}^*$, is assumed to be nonempty.

The augmented Lagrangian-based methods, especially some splitting forms including alternating direction method of multipliers (ADMM for short) and parallel splitting algorithm [17], are verified to be very efficient for problem (1.9). The iterative scheme of the ALM to the problem (1.9) is

$$\begin{cases} \lambda^{k+1} = \lambda^k - \beta(Ax^k + By^k - b), \\ (x^{k+1}, y^{k+1}) = \arg\min_{x\in\mathcal{X}, y\in\mathcal{Y}} \left\{\theta_1(x) + \theta_2(y) + \frac{\beta}{2}\left\|Ax + By - b - \frac{1}{\beta}\lambda^{k+1}\right\|^2\right\}. \end{cases} \tag{1.10}$$

In the second subproblem of scheme (1.10), the coupled variable (x, y) makes it intractable. As a splitting version of the ALM, the ADMM originally proposed by Glowinski [18] and Gabay [19], is as follows:

$$\begin{cases} x^{k+1} = \arg\min_{x\in\mathcal{X}} \left\{\theta_1(x) + \frac{\beta}{2}\left\|Ax + By^k - b - \frac{1}{\beta}\lambda^k\right\|^2\right\}, \\ y^{k+1} = \arg\min_{y\in\mathcal{Y}} \left\{\theta_2(y) + \frac{\beta}{2}\left\|Ax^{k+1} + By - b - \frac{1}{\beta}\lambda^k\right\|^2\right\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b). \end{cases}$$

The ADMM overcomes the drawback of ALM stated above by discoupling variable (x, y). Gabay [20] expressed the ADMM as essentially an application of the Douglas–Rachford splitting method [21]. Cai, Gu and He [22] provided a novel and simple PPA way to understand ADMM, and proposed a generalized ADMM (gADMM) which producing firstly a predictor $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)$ via

$$\begin{cases} \tilde{x}^k = \arg\min_{x\in\mathcal{X}} \left\{\theta_1(x) + \frac{\beta}{2}\left\|Ax + By^k - b - \frac{1}{\beta}\lambda^k\right\|^2\right\}, \\ \tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k - b), \\ \tilde{y}^k = \arg\min_{y\in\mathcal{Y}} \left\{\theta_2(y) + \frac{\beta}{2}\left\|A\tilde{x}^k + By - b - \frac{1}{\beta}\tilde{\lambda}^k\right\|^2\right\}. \end{cases} \tag{1.11}$$

Let $v = (y, \lambda)$ and $w := (x, v) \in \mathcal{W} = \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^m$, and then the new iterate $w^{k+1} = (x^{k+1}, v^{k+1})$ of the gADMM is generated by a relaxation step:

$$x^{k+1} = \tilde{x}^k, \qquad v^{k+1} = v^k - \gamma(v^k - \tilde{v}^k), \quad \gamma \in (0, 2).$$

The iterative scheme (1.11) can be recovered by a compact form: find $\tilde{w}^k \in \mathcal{W}$ such that

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^{\mathrm{T}} \Big\{ F(\tilde{w}^k) + P_4(\tilde{w}^k - w^k) \Big\} \geq 0, \quad \forall\, w \in \mathcal{W}, \quad (1.12)$$

where $u = (x, y), \quad w = (u, \lambda) \in \mathcal{W}$ and $\theta(u) = \theta_1(x) + \theta_2(y)$ and

$$u = \begin{pmatrix} x \\ y \end{pmatrix}, \quad w = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, \quad F(w) = \begin{pmatrix} -A^{\mathrm{T}}\lambda \\ -B^{\mathrm{T}}\lambda \\ Ax + By - b \end{pmatrix},$$

and the customized matrix P_4 is as follows:

$$P_4 = \begin{pmatrix} 0_{n_1 \times n_1} & 0_{n_1 \times n_2} & 0_{n_1 \times m} \\ 0_{n_2 \times n_1} & \beta B^{\mathrm{T}} B & -B^{\mathrm{T}} \\ 0_{m \times n_1} & -B & \frac{1}{\beta} I_m \end{pmatrix}.$$

The special structure of matrix P_4 implies that variable x may not be involved in the iteration of the gADMM. The following matrix

$$\bar{P}_4 = \begin{pmatrix} \beta B^{\mathrm{T}} B & -B^{\mathrm{T}} \\ -B & \frac{1}{\beta} I_m \end{pmatrix},$$

which removes zero-elements from P_4, is required to be symmetric and positive semidefinite to ensure the convergence of the gADMM. The gADMM was verified to be faster than ADMM because some approximate computation is permitted.

Let $P_4' \in \mathbb{R}^{n_1+n_2+m}$ be a matrix such that $P_4'(w^k - \tilde{w}^k)$ be a descent direction, He and Yuan [23,24] presented a general relaxation step as follows:

$$w^{k+1} = w^k - \gamma P_4'(w^k - \tilde{w}^k),$$

where γ is the steplength. Let $H = \bar{P}_4(P_4')^{-1}$ and $G = (\bar{P}_4)^{\mathrm{T}} + \bar{P}_4 - \gamma(P_4')^{\mathrm{T}} H (P_4')$. One can relax the conventional proposition on $\bar{P}_4$, but ensure symmetric and positive semidefinite property of G. In [23,24], He and Yuan established the worst-case convergence rate $\mathcal{O}(1/k)$ and non-ergodic convergence rate of the Douglas–Rachford alternating direction method of multipliers.

1.3 Contributions

From above analysis, there are many PPAs for problem (1.1) and (1.9).

The relaxation steps are very effective used in PPAs. For example, the relaxation step of CPPA accelerates the algorithm for problem (1.1), the relaxation step of LPPA and gADMM ensures the convergence for problem (1.1) and (1.9), respectively. However, the relaxation step might be unacceptable or even not permitted in some practical applications. These applications include real time control system, decision-making on flow line production, supply chain management and decision-making, and so on. For an instance, in the decision-making framework for supply chain management [25–27], each firm on the chain makes decision independently in a proper sequence, and the decision is immediately executed whenever it is given. Hence, all firms cannot adjust their decision at the end, correspondingly the relaxation step in the simulation method has not any actual meaning.

To adapt these real circumstances, we prefer to modify the prediction step rather than to take a relaxation step, and propose two novel customized proximal point algorithms without the relaxation step. The proposed methods are generalized versions of the CPPA proposed in [13,28] for both convex and separable convex problems, respectively. The global convergence and the $\mathcal{O}(1/k)$-convergence rate of the proposed methods will be established under some mild assumptions. We also verify that the new methods have a better numerical performance in practice comparing to some state-of-the-art methods within the relaxation step.

The rest of the paper is organized as follows. In Sect. 2, we propose two new customized proximal point algorithms without relaxation step for problems (1.1) and (1.9), respectively. The convergence of proposed methods will be proven in Sect. 2. In Sect. 3, some preliminary numerical results, comparing to the state-of-the-art methods, are presented to show the high efficiency of the new methods. Section 4 concludes this paper with some final remarks.

2 Two New Customized Proximal Point Algorithms

2.1 The Generalized Customized Proximal Point Algorithm

The Lagrangian function of problem (1.1) is

$$L(x, \lambda) = \theta(x) - \lambda^{\mathrm{T}}(Ax - b). \tag{2.1}$$

If $x^* \in \mathcal{X}$ is a solution of problem (1.1), then there exists $\lambda^* \in \mathbb{R}^m$ such that the pair (x^*, λ^*) is a saddle point of (2.1) satisfying

$$\begin{cases} L(x, \lambda^*) - L(x^*, \lambda^*) \geq 0, & \forall\, x \in \mathcal{X}, \\ L(x^*, \lambda^*) - L(x^*, \lambda) \geq 0, & \forall\, \lambda \in \mathbb{R}^m. \end{cases}$$

Consequently,

$$x^* = \arg\min_{x \in \mathcal{X}} L(x, \lambda^*), \quad \lambda^* = \arg\max_{\lambda \in \mathbb{R}^m} L(x^*, \lambda). \tag{2.2}$$

By Lemma 2.1 in [11], the first-order optimality conditions of problem (2.2) are as follows:

$$\begin{cases} \theta(x) - \theta(x^*) - (x - x^*)^{\mathrm{T}}(A^{\mathrm{T}}\lambda^*) \geq 0, & \forall\, x \in \mathcal{X}, \\ (\lambda - \lambda^*)^{\mathrm{T}}(Ax^* - b) \geq 0, & \forall\, \lambda \in \mathbb{R}^m. \end{cases}$$

Definition 2.1 The solution set of (1.1), denoted by $\mathcal{W}^* \subset \mathcal{W}$, consists of all $w^* \in \mathcal{W}$ satisfying

$$\theta(u) - \theta(u^*) + (w - w^*)^{\mathrm{T}} F(w^*) \geq 0, \ \ \forall\, w \in \mathcal{W}, \tag{2.3}$$

where

$$u = x, \ \ w = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \ \ F(w) = \begin{pmatrix} -A^{\mathrm{T}}\lambda \\ Ax - b \end{pmatrix}. \tag{2.4}$$

The generalized customized proximal point algorithm for problem (1.1) is proposed as follows.

Algorithm 2.2: Generalized customized proximal point algorithm, gCPPA

For a given $w^k \in \mathcal{W}$, the gCPPA produces the new iterate $w^{k+1} \in \mathcal{W}$ via solving

$$\begin{cases} \lambda^{k+1} = \lambda^k - \frac{\alpha}{t}(Ax^k - b), \\ x^{k+1} = \arg\min_{x \in \mathcal{X}} \left\{ \theta(x) + \frac{r}{2} \left\| (x - x^k) - \frac{1}{r} A^{\mathrm{T}}[(1+\alpha)\lambda^{k+1} - \alpha\lambda^k] \right\|^2 \right\}, \end{cases} \tag{2.5}$$

where $\alpha \in (0.5, 1)$.

By the first-order optimality conditions of iteration subproblem (2.5), we have

$$\begin{cases} \theta(x) - \theta(x^{k+1}) + (x - x^{k+1})^{\mathrm{T}} \Big\{ - A^{\mathrm{T}}[(1+\alpha)\lambda^{k+1} - \alpha\lambda^k] + r(x^{k+1} - x^k) \Big\} \geq 0, \\ (\lambda - \lambda^{k+1})^{\mathrm{T}} \left\{ \alpha(Ax^k - b) + t(\lambda^{k+1} - \lambda^k) \right\} = 0, \end{cases}$$

which implies that $\forall\, w \in \mathcal{W}$ the following variational inequality holds:

$$\begin{aligned} &\theta(u) - \theta(u^{k+1}) + (\alpha - 1)(\lambda - \lambda^{k+1})^{\mathrm{T}}(Ax^{k+1} - b) \\ &\quad + \begin{pmatrix} x - x^{k+1} \\ \lambda - \lambda^{k+1} \end{pmatrix}^{\mathrm{T}} \left[\begin{pmatrix} -A^{\mathrm{T}}\lambda^{k+1} \\ Ax^{k+1} - b \end{pmatrix} + \begin{pmatrix} rI_n & -\alpha A^{\mathrm{T}} \\ -\alpha A & tI_m \end{pmatrix} \begin{pmatrix} x^{k+1} - x^k \\ \lambda^{k+1} - \lambda^k \end{pmatrix} \right] \geq 0. \end{aligned}$$

In a compact form, we have

$$\begin{aligned} &\theta(u) - \theta(u^{k+1}) + \frac{t(\alpha - 1)}{\alpha}(\lambda - \lambda^{k+1})^{\mathrm{T}}(\lambda^{k+1} - \lambda^{k+2}) \\ &\quad + (w - w^{k+1})^{\mathrm{T}} \Big[F(w^{k+1}) + Q(w^{k+1} - w^k) \Big] \geq 0, \ \ \text{for any } w \in \mathcal{W}, \end{aligned} \tag{2.6}$$

where $F(w)$ is defined by (2.4), and

$$Q = \begin{pmatrix} rI_n & -\alpha A^{\mathrm{T}} \\ -\alpha A & tI_m \end{pmatrix}. \tag{2.7}$$

Lemma 2.3 *Let $w^* \in \mathcal{W}^*$ be a solution defined by (2.3), and $\{w^k\}$ be a sequence generated by the gCPPA method. Then we have*

$$(w^* - w^{k+1})^T Q(w^{k+1} - w^k) \geq \tfrac{t(1-\alpha)}{\alpha}(\lambda^* - \lambda^{k+1})^T(\lambda^{k+1} - \lambda^{k+2}). \tag{2.8}$$

Proof By setting $u = u^*,\quad w = w^*$ in (2.6), we get

$$\begin{aligned}&(w^* - w^{k+1})^{\mathrm{T}} Q(w^{k+1} - w^k)\\ &\quad\geq \theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^{\mathrm{T}} F(w^{k+1})\\ &\quad+\tfrac{t(1-\alpha)}{\alpha}(\lambda^* - \lambda^{k+1})^{\mathrm{T}}(\lambda^{k+1} - \lambda^{k+2}).\end{aligned} \tag{2.9}$$

By the monotonicity of $F(w)$,[1]. we have

$$(w^{k+1} - w^*)^{\mathrm{T}} F(w^{k+1}) \geq (w^{k+1} - w^*)^{\mathrm{T}} F(w^*). \tag{2.10}$$

Since $w^* \in \mathcal{W}^*$ is a solution, it follows

$$\theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^{\mathrm{T}} F(w^*) \geq 0. \tag{2.11}$$

Adding (2.10) and (2.11) yields

$$\theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^{\mathrm{T}} F(w^{k+1}) \geq 0. \tag{2.12}$$

Substituting (2.12) into (2.9), we obtain (2.8) and complete the proof. □

In the current subsection, let

$$M = rI_n - \frac{\alpha^2}{t} A^{\mathrm{T}} A. \tag{2.13}$$

It is easy to show that if $r, t > 0$ and $rt \geq \alpha^2 \rho(A^{\mathrm{T}} A)$, where $\alpha \in (0.5, 1)$, then Q and M are positive semidefinite. We use the notation $\|w\|_H^2 := w^{\mathrm{T}} H w \geq 0$ if H is symmetric and positive semidefinite.

Lemma 2.4 *For the sequence $\{w^k\}$ generated by the gCPPA method we have*

$$\begin{aligned}&\|x^k - x^{k+1}\|_M^2 + \frac{t(2\alpha - 1)}{\alpha}\|\lambda^{k+1} - \lambda^{k+2}\|^2\\ &\quad\leq \|w^* - w^k\|_Q^2 + \tfrac{t(1-\alpha)}{\alpha}\|\lambda^* - \lambda^{k+1}\|^2 - \Big[\|w^* - w^{k+1}\|_Q^2 + \tfrac{t(1-\alpha)}{\alpha}\|\lambda^* - \lambda^{k+2}\|^2\Big].\end{aligned} \tag{2.14}$$

[1] The mapping $F(w)$ is affine with a skew-symmetric matrix, and thus it is monotone; see He and Yuan [23]

Proof By the identity

$$(a-b)^{\mathrm{T}}Q(c-d)=\tfrac{1}{2}\big(\|a-d\|_Q^2-\|a-c\|_Q^2+\|c-b\|_Q^2-\|d-b\|_Q^2\big),$$

we have

$$\begin{aligned}&(w^*-w^{k+1})^{\mathrm{T}}Q(w^{k+1}-w^k)\\&=\frac{1}{2}\Big(\big\|w^*-w^k\big\|_Q^2-\big\|w^*-w^{k+1}\big\|_Q^2-\big\|w^k-w^{k+1}\big\|_Q^2\Big),\end{aligned}\tag{2.15}$$

and

$$\begin{aligned}&(\lambda^*-\lambda^{k+1})^{\mathrm{T}}(\lambda^{k+1}-\lambda^{k+2})\\&=\frac{1}{2}\Big(\|\lambda^*-\lambda^{k+2}\|^2-\|\lambda^*-\lambda^{k+1}\|^2-\|\lambda^{k+1}-\lambda^{k+2}\|^2\Big).\end{aligned}\tag{2.16}$$

Combining (2.15), (2.16) and (2.8), we obtain

$$\begin{aligned}&\big\|w^k-w^{k+1}\big\|_Q^2-\tfrac{t(1-\alpha)}{\alpha}\big\|\lambda^{k+1}-\lambda^{k+2}\big\|^2\\&\quad\le\big\|w^*-w^k\big\|_Q^2-\big\|w^*-w^{k+1}\big\|_Q^2+\tfrac{t(1-\alpha)}{\alpha}\Big(\big\|\lambda^*-\lambda^{k+1}\big\|^2-\big\|\lambda^*-\lambda^{k+2}\big\|^2\Big).\end{aligned}\tag{2.17}$$

By the definition of Q in (2.7), it follows

$$\begin{aligned}\big\|w^k-w^{k+1}\big\|_Q^2=r\big\|x^k-x^{k+1}\big\|^2-2\alpha(x^k-x^{k+1})^{\mathrm{T}}A^{\mathrm{T}}(\lambda^k-\lambda^{k+1})\\+t\big\|\lambda^k-\lambda^{k+1}\big\|^2.\end{aligned}\tag{2.18}$$

Applying λ-iteration of (2.5), we get

$$\begin{aligned}&-2\alpha(x^k-x^{k+1})^{\mathrm{T}}A^{\mathrm{T}}(\lambda^k-\lambda^{k+1})+t\|\lambda^k-\lambda^{k+1}\|^2\\&=t\Big\|\big[\tfrac{\alpha}{t}(Ax^k-b)-\tfrac{\alpha}{t}(Ax^{k+1}-b)\big]-(\lambda^k-\lambda^{k+1})\Big\|^2-\tfrac{\alpha^2}{t}\big\|x^k-x^{k+1}\big\|_{A^{\mathrm{T}}A}^2\\&=t\Big\|\big[(\lambda^k-\lambda^{k+1})-(\lambda^{k+1}-\lambda^{k+2})\big]-(\lambda^k-\lambda^{k+1})\Big\|^2-\tfrac{\alpha^2}{t}\big\|x^k-x^{k+1}\big\|_{A^{\mathrm{T}}A}^2\\&=t\big\|\lambda^{k+1}-\lambda^{k+2}\big\|^2-\tfrac{\alpha^2}{t}\big\|x^k-x^{k+1}\big\|_{A^{\mathrm{T}}A}^2.\end{aligned}\tag{2.19}$$

Combining (2.18) with (2.19), it follows

$$\big\|w^k-w^{k+1}\big\|_Q^2=\big\|x^k-x^{k+1}\big\|_M^2+t\big\|\lambda^{k+1}-\lambda^{k+2}\big\|^2.\tag{2.20}$$

Substituting (2.20) into (2.17), we obtain (2.14) immediately and complete the proof. □

Theorem 2.5 *Suppose* $\alpha \in (0.5, 1)$, $r, t > 0$ *and* $rt \geq \alpha^2 \rho(A^T A)$, *the sequence* $\{w^k\}$ *is generated by the gCPPA method. Then we have*

$$\lim_{k\to\infty} \|w^{k+1} - w^k\| = 0, \tag{2.21}$$

and $\{w^k\}$ *globally converges to an element of solution set* $\mathcal{W}^*$ *satisfying* (2.3).

Proof Since $\alpha \in (0.5, 1)$, $r, t > 0$ and $rt \geq \alpha^2 \rho(A^{\mathrm{T}} A)$, it holds that

$$\begin{aligned}&\left\|x^k - x^{k+1}\right\|_M^2 + \tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{k+1} - \lambda^{k+2}\|^2\\ &\quad\geq \left(r - \tfrac{\alpha^2\rho(A^{\mathrm{T}}A)}{t}\right)\|x^k - x^{k+1}\|^2 + \tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{k+1} - \lambda^{k+2}\|^2 \geq 0,\end{aligned} \tag{2.22}$$

where M is defined by (2.13). Summing (2.14) from 0 to $K > 1$, it yields

$$\begin{aligned}&\sum_{k=0}^{K}\left(\left\|x^k - x^{k+1}\right\|_M^2 + \tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{k+1} - \lambda^{k+2}\|^2\right)\\ &\quad\leq \left\|w^* - w^0\right\|_Q^2 + \tfrac{t(1-\alpha)}{\alpha}\|\lambda^* - \lambda^1\|^2 - \left[\left\|w^* - w^{K+1}\right\|_Q^2 + \tfrac{t(1-\alpha)}{\alpha}\|\lambda^* - \lambda^{K+2}\|^2\right].\end{aligned} \tag{2.23}$$

On one hand, we get from (2.22) and (2.23) that

$$\left\|w^* - w^0\right\|_Q^2 + \tfrac{t(1-\alpha)}{\alpha}\|\lambda^* - \lambda^1\|^2 \geq \left\|w^* - w^{K+1}\right\|_Q^2 + \tfrac{t(1-\alpha)}{\alpha}\|\lambda^* - \lambda^{K+2}\|^2 \tag{2.24}$$

holds for $\forall K > 1$. Since the objective function of problem (1.1) is a closed convex function, the solution set $\mathcal{W}^*$ is a bounded closed convex nonempty set.

On the other hand, by taking limits on the both side of (2.23), we get

$$\sum_{k=0}^{\infty}\left(\left\|x^k - x^{k+1}\right\|_M^2 + \tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{k+1} - \lambda^{k+2}\|^2\right) \leq d, \tag{2.25}$$

where $d := \left\|w^* - w^0\right\|_Q^2 + \tfrac{t(1-\alpha)}{\alpha}\|\lambda^* - \lambda^1\|^2$. Thus

$$\lim_{k\to\infty}\|x^k - x^{k+1}\|_M^2 = 0, \quad \lim_{k\to\infty}\|\lambda^{k+1} - \lambda^{k+2}\|^2 = 0,$$

which implies (2.21), i.e.,

$$\lim_{k\to\infty}\|x^k - x^{k+1}\|^2 = 0, \quad \lim_{k\to\infty}\|\lambda^k - \lambda^{k+1}\|^2 = 0, \tag{2.26}$$

The sequence $\{w^k\}$ has at least one cluster point. Let w^∞ be a cluster point of $\{w^k\}$, i.e., there exists a subsequence $\{w^{k_j}\}_{k_j\in\mathcal{K}}$ such that $w^\infty = \lim_{k_j\to\infty, k_j\in\mathcal{K}} w^{k_j}$. Then by taking limit on (2.6) as $k \to \infty$, and applying (2.26) we have

$$\theta(u) - \theta(u^{\infty}) + (w - w^{\infty})^{\mathrm{T}} F(w^{\infty}) \geq 0, \forall\, w \in \mathcal{W},$$

which implies that $w^{\infty} \in \mathcal{W}^*$ is a solution satisfying (2.3) and completes the proof. □

Theorem 2.6 *If the conditions of Theorem 2.5 hold, then the sequence $\{w^k\}$ generated by the gCPPA method has a worst-case convergence rate $\mathcal{O}(1/k)$.*

Proof Substituting $a = w^{k-1} - w^k$ and $b = w^k - w^{k+1}$ to the identity

$$\|a\|_Q^2 - \|b\|_Q^2 = 2a^{\mathrm{T}} Q(a - b) - \|a - b\|_Q^2,$$

we obtain

$$\begin{aligned} &\left\| w^{k-1} - w^k \right\|_Q^2 - \left\| w^k - w^{k+1} \right\|_Q^2 \\ &\quad = 2(w^{k-1} - w^k)^{\mathrm{T}} Q \Big\{ w^{k-1} - w^k - (w^k - w^{k+1}) \Big\} \\ &\quad - \Big\| w^{k-1} - w^k - (w^k - w^{k+1}) \Big\|_Q^2 . \end{aligned} \tag{2.27}$$

Letting $w := w^k$ in (2.6), we get

$$\begin{aligned} &\theta(u^k) - \theta(u^{k+1}) + \frac{t(\alpha - 1)}{\alpha} (\lambda^k - \lambda^{k+1})^{\mathrm{T}} (\lambda^{k+1} - \lambda^{k+2}) \\ &\quad + (w^k - w^{k+1})^{\mathrm{T}} \Big[F(w^{k+1}) + Q(w^{k+1} - w^k) \Big] \geq 0. \end{aligned} \tag{2.28}$$

Since (2.6) also holds at the $(k-1)^{th}$ iteration, we have

$$\begin{aligned} &\theta(u) - \theta(u^k) + \frac{t(\alpha - 1)}{\alpha} (\lambda - \lambda^k)^{\mathrm{T}} (\lambda^k - \lambda^{k+1}) \\ &\quad + (w - w^k)^{\mathrm{T}} \Big[F(w^k) + Q(w^k - w^{k-1}) \Big] \geq 0, \quad \forall\, w \in \mathcal{W}. \end{aligned}$$

Setting $w := w^{k+1}$ in the above inequality, we obtain

$$\begin{aligned} &\theta(u^{k+1}) - \theta(u^k) + \frac{t(\alpha - 1)}{\alpha} (\lambda^{k+1} - \lambda^k)^{\mathrm{T}} (\lambda^k - \lambda^{k+1}) \\ &\quad + (w^{k+1} - w^k)^{\mathrm{T}} \Big[F(w^k) + Q(w^k - w^{k-1}) \Big] \geq 0. \end{aligned} \tag{2.29}$$

Adding (2.28) and (2.29), it yields

$$\begin{aligned} &(w^k - w^{k+1})^{\mathrm{T}} Q \Big\{ w^{k-1} - w^k - (w^k - w^{k+1}) \Big\} \\ &\quad \geq (w^{k+1} - w^k)^{\mathrm{T}} \Big[F(w^{k+1}) - F(w^k) \Big] \\ &\quad + \tfrac{t(1-\alpha)}{\alpha} (\lambda^k - \lambda^{k+1})^{\mathrm{T}} \Big[\lambda^{k+1} - \lambda^{k+2} - (\lambda^k - \lambda^{k+1}) \Big]. \end{aligned}$$

By the monotonicity of $F(w)$, we get from the above inequality that

$$\begin{aligned}&(w^k - w^{k+1})^{\mathrm{T}} Q\Big\{w^{k-1} - w^k - (w^k - w^{k+1})\Big\}\\&\quad\geq \tfrac{t(1-\alpha)}{\alpha}(\lambda^k - \lambda^{k+1})^{\mathrm{T}}\Big[\lambda^{k+1} - \lambda^{k+2} - (\lambda^k - \lambda^{k+1})\Big].\end{aligned}\tag{2.30}$$

Adding $\big\|w^{k-1} - w^k - (w^k - w^{k+1})\big\|_Q^2$ to the both sides of (2.30), it follows

$$\begin{aligned}&(w^{k-1} - w^k)^{\mathrm{T}} Q\Big\{w^{k-1} - w^k - (w^k - w^{k+1})\Big\}\\&\quad\geq \Big\|w^{k-1} - w^k - (w^k - w^{k+1})\Big\|_Q^2\\&\quad+ \tfrac{t(1-\alpha)}{\alpha}(\lambda^k - \lambda^{k+1})^{\mathrm{T}}\Big[\lambda^{k+1} - \lambda^{k+2} - (\lambda^k - \lambda^{k+1})\Big].\end{aligned}\tag{2.31}$$

It is easy to derive that

$$\begin{aligned}&2(\lambda^k - \lambda^{k+1})^{\mathrm{T}}\Big[\lambda^{k+1} - \lambda^{k+2} - (\lambda^k - \lambda^{k+1})\Big]\\&\quad= -\Big\|\lambda^k - \lambda^{k+1} - (\lambda^{k+1} - \lambda^{k+2})\Big\|^2 + \|\lambda^{k+1} - \lambda^{k+2}\|^2 - \|\lambda^k - \lambda^{k+1}\|^2.\end{aligned}\tag{2.32}$$

Combining (2.27) with (2.31) and (2.32), we have

$$\begin{aligned}&\|w^{k-1} - w^k\|_Q^2 + \tfrac{t(1-\alpha)}{\alpha}\|\lambda^k - \lambda^{k+1}\|^2 - \Big(\|w^k - w^{k+1}\|_Q^2 + \tfrac{t(1-\alpha)}{\alpha}\|\lambda^{k+1} - \lambda^{k+2}\|^2\Big)\\&\quad\geq \Big\|w^{k-1} - w^k - (w^k - w^{k+1})\Big\|_Q^2 - \tfrac{t(1-\alpha)}{\alpha}\Big\|\lambda^k - \lambda^{k+1} - (\lambda^{k+1} - \lambda^{k+2})\Big\|^2.\end{aligned}\tag{2.33}$$

By the definition of Q in (2.7), it follows

$$\begin{aligned}\Big\|w^{k-1} - w^k - (w^k - w^{k+1})\Big\|_Q^2 &= r\Big\|x^{k-1} - x^k - (x^k - x^{k+1})\Big\|^2 + t\Big\|\lambda^{k-1} - \lambda^k - (\lambda^k - \lambda^{k+1})\Big\|^2\\&\quad - 2\alpha\Big[x^{k-1} - x^k - (x^k - x^{k+1})\Big]^{\mathrm{T}} A^{\mathrm{T}}\Big[\lambda^{k-1} - \lambda^k - (\lambda^k - \lambda^{k+1})\Big].\end{aligned}$$

By λ-iteration of (2.5), we get

$$\begin{aligned}&- 2\alpha\Big[x^{k-1} - x^k - (x^k - x^{k+1})\Big]^{\mathrm{T}} A^{\mathrm{T}}\Big[\lambda^{k-1} - \lambda^k - (\lambda^k - \lambda^{k+1})\Big]\\&+ t\Big\|\lambda^{k-1} - \lambda^k - (\lambda^k - \lambda^{k+1})\Big\|^2\\&\quad= t\Big\|\tfrac{\alpha}{t}A\big[x^{k-1} - x^k - (x^k - x^{k+1})\big] - \big[\lambda^{k-1} - \lambda^k - (\lambda^k - \lambda^{k+1})\big]\Big\|^2\end{aligned}$$

$$
\begin{aligned}
&- \tfrac{\alpha^2}{t}\left\|x^{k-1}-x^k-(x^k-x^{k+1})\right\|^2_{A^{\mathrm{T}}A}\\
&= t\left\|\lambda^k-\lambda^{k+1}-(\lambda^{k+1}-\lambda^{k+2})\right\|^2 - \tfrac{\alpha^2}{t}\left\|x^{k-1}-x^k-(x^k-x^{k+1})\right\|^2_{A^{\mathrm{T}}A},
\end{aligned}
$$

which implies

$$
\begin{aligned}
\left\|w^{k-1}-w^k-(w^k-w^{k+1})\right\|^2_Q &= \left\|x^{k-1}-x^k-(x^k-x^{k+1})\right\|^2_M\\
&\quad + t\left\|\lambda^k-\lambda^{k+1}-(\lambda^{k+1}-\lambda^{k+2})\right\|^2. \qquad (2.34)
\end{aligned}
$$

Substituting (2.20) and (2.34) into (2.33), we have

$$
\begin{aligned}
&\|x^{k-1}-x^k\|^2_M + \tfrac{t}{\alpha}\|\lambda^k-\lambda^{k+1}\|^2 - \left(\|x^k-x^{k+1}\|^2_M + \tfrac{t}{\alpha}\|\lambda^{k+1}-\lambda^{k+2}\|^2\right)\\
&\geq \left\|x^{k-1}-x^k-(x^k-x^{k+1})\right\|^2_M + \tfrac{t(2\alpha-1)}{\alpha}\left\|\lambda^k-\lambda^{k+1}-(\lambda^{k+1}-\lambda^{k+2})\right\|^2.
\end{aligned}
\qquad (2.35)
$$

Adding $\left(\frac{t(2\alpha-1)}{\alpha}-\frac{t}{\alpha}\right)\left[\|\lambda^k-\lambda^{k+1}\|^2-\|\lambda^{k+1}-\lambda^{k+2}\|^2\right]$ to the both side of (2.35), we obtain

$$
\begin{aligned}
&\|x^{k-1}-x^k\|^2_M + \tfrac{t(2\alpha-1)}{\alpha}\|\lambda^k-\lambda^{k+1}\|^2 - \left(\|x^k-x^{k+1}\|^2_M + \tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{k+1}-\lambda^{k+2}\|^2\right)\\
&\geq \left\|x^{k-1}-x^k-(x^k-x^{k+1})\right\|^2_M + \tfrac{t(2\alpha-1)}{\alpha}\left\|\lambda^k-\lambda^{k+1}-(\lambda^{k+1}-\lambda^{k+2})\right\|^2\\
&\quad -\tfrac{2t(1-\alpha)}{\alpha}\left[\|\lambda^k-\lambda^{k+1}\|^2-\|\lambda^{k+1}-\lambda^{k+2}\|^2\right]\\
&= \left\|x^{k-1}-x^k-(x^k-x^{k+1})\right\|^2_M + \tfrac{t}{\alpha}\left\|(2\alpha-1)(\lambda^k-\lambda^{k+1})-(\lambda^{k+1}-\lambda^{k+2})\right\|^2\\
&\quad +4(2-\alpha)t\|\lambda^k-\lambda^{k+1}\|^2.
\end{aligned}
$$

Recall that M is positive semidefinite and $\alpha \in (0.5, 1)$, it follows

$$
\|x^k-x^{k+1}\|^2_M + \tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{k+1}-\lambda^{k+2}\|^2 \leq \|x^{k-1}-x^k\|^2_M + \tfrac{t(2\alpha-1)}{\alpha}\|\lambda^k-\lambda^{k+1}\|^2. \qquad (2.36)
$$

Which implies $\left\{\|x^k-x^{k+1}\|^2_M + \frac{t(2\alpha-1)}{\alpha}\|\lambda^{k+1}-\lambda^{k+2}\|^2\right\}$ is monotonically nonincreasing whenever $k \geq 0$. Summing (2.36) from 1 to $K+1$, it yields

$$
\begin{aligned}
&(1+K)\left(\|x^K-x^{K+1}\|^2_M + \tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{K+1}-\lambda^{K+2}\|^2\right)\\
&\leq \sum_{k=1}^{K+1}\left(\|x^{k-1}-x^k\|^2_M + \tfrac{t(2\alpha-1)}{\alpha}\|\lambda^k-\lambda^{k+1}\|^2\right)
\end{aligned}
$$

$$= \sum_{k=0}^{K} \Big(\|x^k - x^{k+1}\|_M^2 + \tfrac{t(2\alpha-1)}{\alpha} \|\lambda^{k+1} - \lambda^{k+2}\|^2 \Big).$$

Combing with (2.23), (2.24) and (2.1), we have

$$(1+K)\Big(\|x^K - x^{K+1}\|_M^2 + \tfrac{t(2\alpha-1)}{\alpha} \|\lambda^{K+1} - \lambda^{K+2}\|^2 \Big) \le d.$$

Hence, for any $\epsilon > 0$ there exists K such that

$$\|x^K - x^{K+1}\|_M^2 + \tfrac{t(2\alpha-1)}{\alpha} \|\lambda^{K+1} - \lambda^{K+2}\|^2 \le \frac{d}{1+K} < \epsilon. \tag{2.37}$$

Let $\lfloor d/\epsilon \rfloor$ means the maximal integer not greater that d/ϵ. Then, inequality (2.37) implies that the gCPPA needs at most $\lfloor d/\epsilon \rfloor$ iterations to ensure that

$$\Big(r - \tfrac{\alpha^2 \rho(A^{\mathrm{T}}A)}{t} \Big) \|x^K - x^{K+1}\|^2 < \epsilon, \quad \tfrac{t(2\alpha-1)}{\alpha} \|\lambda^{K+1} - \lambda^{K+2}\|^2 < \epsilon,$$

which implies the sequence $\{w^k\}$ generated by the gCPPA method has a worst-case convergence rate $\mathcal{O}(1/k)$ in a non-ergodic sense. The proof of Theorem 2.6 is completed. □

2.2 Extended gCPPA to Separable Convex Minimization Problems

The Lagrangian function of problem (1.9) is

$$L(x, y, \lambda) = \theta_1(x) + \theta_2(y) - \lambda^{\mathrm{T}}(Ax + By - b). \tag{2.38}$$

If a pair of $(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$ is a solution of problem (1.9), then there exists $\lambda^* \in \mathbb{R}^m$ such that (x^*, y^*, λ^*) is a saddle point of (2.38) satisfying

$$\begin{cases} L(x, y^*, \lambda^*) - L(x^*, y^*, \lambda^*) \ge 0, & \forall\, x \in \mathcal{X}, \\ L(x^*, y, \lambda^*) - L(x^*, y^*, \lambda^*) \ge 0, & \forall\, y \in \mathcal{Y}, \\ L(x^*, y^*, \lambda^*) - L(x^*, y^*, \lambda) \ge 0, & \forall\, \lambda \in \mathbb{R}^m. \end{cases} \tag{2.39}$$

Consequently, by (2.39) it follows

$$\begin{aligned} x^* &= \arg\min_{x \in \mathcal{X}} L(x, y^*, \lambda^*), \quad y^* = \arg\min_{y \in \mathcal{Y}} L(x^*, y, \lambda^*), \\ \lambda^* &= \arg\max_{\lambda \in \mathbb{R}^m} L(x^*, y^*, \lambda). \end{aligned} \tag{2.40}$$

According to Lemma 2.1 in [11], we have the first-order optimality conditions of problem (2.40):

$$\begin{cases} \theta_1(x) - \theta_1(x^*) - (x - x^*)^{\mathrm{T}}(A^{\mathrm{T}}\lambda^*) \ge 0, & \forall\, x \in \mathcal{X}, \\ \theta_2(y) - \theta_2(y^*) - (y - y^*)^{\mathrm{T}}(B^{\mathrm{T}}\lambda^*) \ge 0, & \forall\, y \in \mathcal{Y}, \\ (\lambda - \lambda^*)^{\mathrm{T}}(Ax^* + By^* - b) \ge 0, & \forall\, \lambda \in \mathbb{R}^m. \end{cases} \tag{2.41}$$

By a compact form of (2.41), the solution set of problem (1.9), also denoted by $\mathcal{W}^* \subset \mathcal{W}$, consists of all $w^* \in \mathcal{W}$ satisfying (2.3), where $\mathcal{W} := \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^m$, $\theta(u) := \theta_1(x) + \theta_2(y)$ and

$$u := \begin{pmatrix} x \\ y \end{pmatrix}, \quad w := \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, \quad F(w) := \begin{pmatrix} -A^{\mathrm{T}}\lambda \\ -B^{\mathrm{T}}\lambda \\ Ax + By - b \end{pmatrix}. \tag{2.42}$$

We are in the position to propose the extended gCPPA for separable convex optimization problem (1.9).

Algorithm 2.7: An extension of the gCPPA to separable convex optimization, eCPPA

For a given w^k, the eCPPA produces the new iterate $w^{k+1} \in \mathcal{W}$ via solving:

$$\begin{cases} \lambda^{k+1} = \lambda^k - \frac{\alpha}{t}(Ax^k + By^k - b), \\ x^{k+1} = \arg\min\limits_{x \in \mathcal{X}} \left\{ \theta_1(x) + \frac{r}{2} \left\| (x - x^k) - \frac{1}{r} A^{\mathrm{T}}\big[(\alpha + 1)\lambda^{k+1} - \alpha\lambda^k\big] \right\|^2 \right\}, \\ y^{k+1} = \arg\min\limits_{y \in \mathcal{Y}} \left\{ \theta_2(y) + \frac{s}{2} \left\| (y - y^k) - \frac{1}{s} B^{\mathrm{T}}\big[(\alpha + 1)\lambda^{k+1} - \alpha\lambda^k\big] \right\|^2 \right\}, \end{cases} \tag{2.43}$$

where $\alpha \in (0.5, 1)$ is a constant.

By the first-order optimality conditions of iteration subproblem (2.43), we have that, for $w^{k+1} \in \mathcal{W}$ generated by the eCPPA and $\forall\, w \in \mathcal{W}$, we have

$$\begin{cases} \theta(x) - \theta(x^{k+1}) + (x - x^{k+1})^{\mathrm{T}} \Big\{ - A^{\mathrm{T}}[(1+\alpha)\lambda^{k+1} - \alpha\lambda^k] + r(x^{k+1} - x^k) \Big\} \ge 0, \\ \theta(y) - \theta(y^{k+1}) + (y - y^{k+1})^{\mathrm{T}} \Big\{ - B^{\mathrm{T}}[(1+\alpha)\lambda^{k+1} - \alpha\lambda^k] + s(y^{k+1} - y^k) \Big\} \ge 0, \\ (\lambda - \lambda^{k+1})^{\mathrm{T}} \Big\{ \alpha(Ax^k + By^k - b) + t(\lambda^{k+1} - \lambda^k) \Big\} = 0. \end{cases} \tag{2.44}$$

By notations (2.42), inequality (2.44) can be rewritten to the compact form (2.6) in which

$$Q = \begin{pmatrix} rI_{n_1} & 0_{n_1 \times n_2} & -\alpha A^{\mathrm{T}} \\ 0_{n_2 \times n_1} & sI_{n_2} & -\alpha B^{\mathrm{T}} \\ -\alpha A & -\alpha B & tI_m \end{pmatrix}. \tag{2.45}$$

Lemma 2.8 *Let $w^* \in \mathcal{W}^*$ be a solution satisfying (2.3), then we have*

$$(w^* - w^{k+1})^T Q(w^{k+1} - w^k) \geq \tfrac{t(1-\alpha)}{\alpha}(\lambda^* - \lambda^{k+1})^T(\lambda^{k+1} - \lambda^{k+2}). \quad (2.46)$$

Proof The mapping $F(w)$ is affine with a skew-symmetric matrix, and thus it is monotone. The rest of the proof is as same as that of Lemma 2.3. □

In this subsection, let

$$M = \begin{pmatrix} rI_{n_1} - \frac{\alpha^2 A^{\mathrm{T}} A}{t} & -\frac{\alpha^2 A^{\mathrm{T}} B}{t} \\ -\frac{\alpha^2 B^{\mathrm{T}} A}{t} & sI_{n_2} - \frac{\alpha^2 B^{\mathrm{T}} B}{t} \end{pmatrix}. \quad (2.47)$$

If $r, s, t > 0$ and $rt \geq 2\alpha^2 \rho(A^{\mathrm{T}} A),\ st \geq 2\alpha^2 \rho(B^{\mathrm{T}} B)$, where $\alpha \in (0.5, 1)$, it is easy to verify that Q and M are positive semidefinite.

Lemma 2.9 *For the sequence $\{w^k\}$ generated by the gCPPA method, we have*

$$\begin{aligned} &\|u^k - u^{k+1}\|_M^2 + \frac{t(2\alpha-1)}{\alpha}\|\lambda^{k+1} - \lambda^{k+2}\|^2 \\ &\quad \leq \|w^* - w^k\|_Q^2 + \tfrac{t(1-\alpha)}{\alpha}\|\lambda^* - \lambda^{k+1}\|^2 - \Big[\|w^* - w^{k+1}\|_Q^2 + \tfrac{t(1-\alpha)}{\alpha}\|\lambda^* - \lambda^{k+2}\|^2\Big]. \end{aligned} \quad (2.48)$$

Proof Similar to Lemma 2.4, we also get (2.17). By notation Q in (2.45), it holds

$$\begin{aligned} \|w^k - w^{k+1}\|_Q^2 = {}& r\|x^k - x^{k+1}\|^2 + s\|y^k - y^{k+1}\|^2 - 2\alpha(x^k - x^{k+1})^{\mathrm{T}} A^{\mathrm{T}}(\lambda^k - \lambda^{k+1}) \\ & -2\alpha(y^k - y^{k+1})^{\mathrm{T}} B^{\mathrm{T}}(\lambda^k - \lambda^{k+1}) + t\|\lambda^k - \lambda^{k+1}\|^2. \end{aligned}$$

From λ-iteration of (2.43), we get

$$\begin{aligned} &-2\alpha(x^k - x^{k+1})^{\mathrm{T}} A^{\mathrm{T}}(\lambda^k - \lambda^{k+1}) - 2\alpha(y^k - y^{k+1})^{\mathrm{T}} B^{\mathrm{T}}(\lambda^k - \lambda^{k+1}) + t\|\lambda^k - \lambda^{k+1}\|^2 \\ &= t\Big\|\big[\tfrac{\alpha}{t}(Ax^k + By^k - b) - \tfrac{\alpha}{t}(Ax^{k+1} + By^{k+1} - b)\big] - (\lambda^k - \lambda^{k+1})\Big\|^2 \\ &\quad - \tfrac{\alpha^2}{t}\Big\|A(x^k - x^{k+1}) + B(y^k - y^{k+1})\Big\|^2 \\ &= t\Big\|\big[(\lambda^k - \lambda^{k+1}) - (\lambda^{k+1} - \lambda^{k+2})\big] - (\lambda^k - \lambda^{k+1})\Big\|^2 \\ &- \tfrac{\alpha^2}{t}\Big\|A(x^k - x^{k+1}) + B(y^k - y^{k+1})\Big\|^2 \\ &= t\|\lambda^{k+1} - \lambda^{k+2}\|^2 - \tfrac{\alpha^2}{t}\Big\|A(x^k - x^{k+1}) + B(y^k - y^{k+1})\Big\|^2, \end{aligned}$$

which implies

$$\|w^k - w^{k+1}\|_Q^2 = \|u^k - u^{k+1}\|_M^2 + t\|\lambda^{k+1} - \lambda^{k+2}\|^2, \quad (2.49)$$

where

$$\begin{aligned}\left\|u^k-u^{k+1}\right\|_M^2 =\ & r\left\|x^k-x^{k+1}\right\|^2+s\left\|y^k-y^{k+1}\right\|^2\\ & -\tfrac{\alpha^2}{t}\left\|A(x^k-x^{k+1})+B(y^k-y^{k+1})\right\|^2.\end{aligned}$$

Substituting (2.49) into (2.17), we obtain (2.14) immediately and complete the proof. □

Theorem 2.10 *Suppose that $r, s, t > 0$ and $rt \geq 2\alpha^2\rho(A^TA)$, $st \geq 2\alpha^2\rho(B^TB)$, where $\alpha \in (0.5, 1)$. Then, for the sequence $\{w^k\}$ is generated by the eCPPA method, we have*

$$\lim_{k\to\infty}\|w^{k+1}-w^k\|=0, \tag{2.50}$$

and $\{w^k\}$ globally converges to an element of solution set $\mathcal{W}^*$ satisfying (2.41).

Proof Since $rt \geq 2\alpha^2\rho(A^{\mathrm{T}}A)$, $st \geq 2\alpha^2\rho(B^{\mathrm{T}}B)$, it follows

$$\begin{aligned}\left\|u^k-u^{k+1}\right\|_M^2 \geq & \left(r-\tfrac{2\alpha^2\rho(A^{\mathrm{T}}A)}{t}\right)\left\|x^{k+1}-x^{k+2}\right\|^2\\ & +\left(s-\tfrac{2\alpha^2\rho(B^{\mathrm{T}}B)}{t}\right)\left\|y^{k+1}-y^{k+2}\right\|^2 \geq 0.\end{aligned} \tag{2.51}$$

Notice that $t > 0$, $\alpha \in (0.5, 1)$, we get

$$\left\|u^k-u^{k+1}\right\|_M^2+\tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{k+1}-\lambda^{k+2}\|^2 \geq 0. \tag{2.52}$$

Summing (2.48) from 0 to $K > 1$, it yields

$$\begin{aligned}&\sum_{k=0}^{K}\left(\left\|u^k-u^{k+1}\right\|_M^2+\tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{k+1}-\lambda^{k+2}\|^2\right)\\ &\quad\leq \left\|w^*-w^0\right\|_Q^2+\tfrac{t(1-\alpha)}{\alpha}\|\lambda^*-\lambda^1\|^2-\left[\left\|w^*-w^{K+1}\right\|_Q^2+\tfrac{t(1-\alpha)}{\alpha}\|\lambda^*-\lambda^{K+2}\|^2\right].\end{aligned} \tag{2.53}$$

By (2.52) and (2.53), it is easy to derive (2.24) and

$$\sum_{k=0}^{\infty}\left(\left\|u^k-u^{k+1}\right\|_M^2+\tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{k+1}-\lambda^{k+2}\|^2\right) \leq d,$$

which implies that

$$\lim_{k\to\infty}\|x^k-x^{k+1}\|^2=0,\quad \lim_{k\to\infty}\|y^k-y^{k+1}\|^2=0,\quad \lim_{k\to\infty}\|\lambda^k-\lambda^{k+1}\|^2=0.$$

Hence, there exists a subsequence $\{w^{k_j}\}_{k_j\in\mathcal{K}}$ such that $w^\infty=\lim_{k_j\to\infty,k_j\in\mathcal{K}}w^{k_j}$ is a solution satisfying (2.3) and completes the proof. □

Theorem 2.11 *If the conditions of Theorem 2.10 hold, then the sequence $\{w^k\}$ generated by the eCPPA method has a worst-case convergence rate $\mathcal{O}(1/k)$ in a non-ergodic sense.*

Proof It holds obviously that

$$\left\|w^{k-1}-w^k-(w^k-w^{k+1})\right\|_Q^2=\left\|u^{k-1}-u^k-(u^k-u^{k+1})\right\|_M^2 \\ +t\left\|\lambda^k-\lambda^{k+1}-(\lambda^{k+1}-\lambda^{k+2})\right\|^2. \tag{2.54}$$

By (2.49) and (2.54), we have

$$\begin{aligned}&\|u^{k-1}-u^k\|_M^2+\tfrac{t(2\alpha-1)}{\alpha}\|\lambda^k-\lambda^{k+1}\|^2-\left(\|u^k-u^{k+1}\|_M^2+\tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{k+1}-\lambda^{k+2}\|^2\right)\\ &\geq\left\|u^{k-1}-u^k-(u^k-u^{k+1})\right\|_M^2+\tfrac{t}{\alpha}\left\|(2\alpha-1)(\lambda^k-\lambda^{k+1})-(\lambda^{k+1}-\lambda^{k+2})\right\|^2\\ &\quad+4(2-\alpha)t\|\lambda^k-\lambda^{k+1}\|^2.\end{aligned}$$

Then, for any $\epsilon>0$ there exists K such that

$$\|u^K-u^{K+1}\|_M^2+\tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{K+1}-\lambda^{K+2}\|^2\leq\frac{d}{1+K}<\epsilon. \tag{2.55}$$

By (2.51), inequality (2.55) implies that the eCPPA needs at most $\lfloor d/\epsilon\rfloor$ iterations to ensure that

$$\left(r-\tfrac{2\alpha^2\rho(A^{\mathrm{T}}A)}{t}\right)\left\|x^K-x^{K+1}\right\|^2<\epsilon,\ \left(s-\tfrac{2\alpha^2\rho(B^{\mathrm{T}}B)}{t}\right)\left\|y^K-y^{K+1}\right\|^2<\epsilon,\\ \tfrac{t(2\alpha-1)}{\alpha}\|\lambda^{K+1}-\lambda^{K+2}\|^2<\epsilon,$$

which implies that the sequence $\{w^k\}$ generated by the eCPPA method has a worst-case convergence rate $\mathcal{O}(1/k)$ in a non-ergodic sense. The proof of Theorem 2.11 is completed. □

3 Numerical Results

The section focuses on the numerical performance of the gCPPA and eCPPA comparing with some state-of-the-art methods. All methods are coded in MATLAB 2012b and run on an HP desktop computer with Intel(R) Core(TM) i5-6500 CPU 3.20GHz, RAM 8G. The termination criterion of gCPPA is set to $\max\left\{\|x^k-x^{k+1}\|,\|\lambda^k-\lambda^{k+1}\|\right\}\leq\varepsilon$, and that of eCPPA is set to $\max\left\{\|x^k-x^{k+1}\|,\|y^k-y^{k+1}\|,\|\lambda^k-\lambda^{k+1}\|\right\}\leq\varepsilon$, where $\varepsilon>0$ is a small real number. By the global convergence of the gCPPA and eCPPA methods, we have $\lim_{k\to\infty}\|w^k-w^{k+1}\|^2=0$, which means that for any given $\varepsilon>0$ the corresponding algorithm will stop in finite iterations.

To adjust parameters $r,\ s$ and t simultaneously, we introduce $\alpha_r,\ \alpha_t > 0$ satisfying $\alpha_r\alpha_t = \alpha^2$. Take $r = \alpha_r R,\ t > \alpha_t\rho(A^{\mathrm{T}}A)/R$ in gCPPA such that $rt \geq \alpha^2\rho(A^{\mathrm{T}}A)$, and take $r = \sqrt{2}\alpha_r R,\ s = \sqrt{2}\alpha_r R\rho(B^{\mathrm{T}}B)/\rho(A^{\mathrm{T}}A),\ t > \sqrt{2}\alpha_t\rho(A^{\mathrm{T}}A)/R$ in eCPPA such that $rt \geq 2\alpha^2\rho(A^{\mathrm{T}}A),\ st \geq 2\alpha^2\rho(B^{\mathrm{T}}B)$, where $R > 0$ is a selected constant.

Example 3.1 The first test problem is a correlation matrices calibrating problem selected from [11,29], which has the form

$$\min\left\{\frac{1}{2}\|X - C\|_F^2 \,\Big|\, \mathrm{diag}(X) = e,\ X \in S_+^n\right\}, \tag{3.1}$$

where $C \in \mathbb{R}^{n\times n}$ is a symmetrical matrix, $e = (1, 1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^{n\times 1}$ and $S_+^n = \{H \in \mathbb{R}^{n\times n} \mid H^{\mathrm{T}} = H, H \succeq 0\}$. Problem (3.1) can be reformulated to problem (1.1) with $\rho(A^{\mathrm{T}}A) = 1$, where A is a projection matrix for the linear constraint $\mathrm{diag}(X) = e$ (see Example 2 in [11]).

Let $z \in \mathbb{R}^n$ be the Lagrange multiplier to the linear equality constraint of problem (3.1).

We compare the performance of the gCPPA to the CPPA and rCPPA methods. For a given (X^k, z^k), the gCPPA produces the new iteration (X^{k+1}, z^{k+1}) as follows:

$$\begin{cases} z^{k+1} = z^k - \frac{\alpha}{t}(\mathrm{diag}(X^k) - e), \\ X^{k+1} = \arg\min_{X\in S_+^n}\left\{\frac{1}{2}\|X - C\|_F^2 + \frac{r}{2}\left\|(X - X^k) - \frac{1}{r}\mathrm{diag}[(1+\alpha)z^{k+1} - \alpha z^k]\right\|_F^2\right\}. \end{cases} \tag{3.2}$$

The X-subproblem of (3.2) is identical to

$$X^{k+1} = \arg\min_{X\in S_+^n}\left\{\frac{1}{2}\left\|X - \frac{1}{r+1}\{rX^k + \mathrm{diag}[(1+\alpha)z^{k+1} - \alpha z^k] + C\}\right\|_F^2\right\}.$$

It is inherent a projection operator via SVD decomposition. It takes the main computation load in each iteration. Similarly, the iteration schemes of CPPA and rCPPA for (3.1) can be written in the form of (1.7) and (1.8).

In our experiments, we take a random matrix $C = (c_{ij})_{n\times n}$ satisfying $c_{ij} \in (-1, 1)$ and $c_{ij} = c_{ji}$ for all i, j. For each given n, 20 random instances are tested. To be fair, the initial point X^0 is an $n \times n$ identity matrix and z^0 is n-dimensional zero vector. For the termination criterion of all methods, we set $\varepsilon = 10^{-5}$.

Let $R = 2.00$. The proximal parameters of the CPPA and rCPPA methods are set to $r = R$ and $t = 1.05/R = 0.53$, which satisfies $rt = 1.05 > \rho(AA^{\mathrm{T}})$. Set $\gamma = 1.50$ in the rCPPA method. The "gCPPA$^{(0.53)}$" means the gCPPA method with $\alpha_r = 0.70$, $\alpha_t = 0.40$, so $\alpha = \sqrt{\alpha_r\alpha_t} \approx 0.53$, $r = R\alpha_r = 1.40$ and $t = 1.05\alpha_t/R = 0.29$. The "gCPPA$^{(0.17)}$" means the gCPPA method with $\alpha_r = 0.30$, $\alpha_t = 0.10$, $r = R\alpha_r = 0.60$, so $\alpha = \sqrt{\alpha_r\alpha_t} \approx 0.17$ and $t = 1.05\alpha_t/R = 0.05$.

For easy comparisons, the results of three methods in terms of matrix dimension n, average iteration number (iter) and average cpu-time (cput in seconds) are put into Table 1, and displayed in Fig. 1.

Table 1 Numerical results of CPPA, rCPPA, gCPPA$^{(0.53)}$ and gCPPA$^{(0.17)}$

n	CPPA		rCPPA		gCPPA$^{(0.53)}$		gCPPA$^{(0.17)}$	
	Iter	cput	Iter	cput	Iter	cput	Iter	cput
100	31	0.15	23	0.11	26	0.12	19	0.09
200	34	0.97	25	0.70	29	0.82	21	0.63
300	37	2.19	28	1.65	31	1.83	22	1.35
400	38	3.93	27	2.78	32	3.34	23	2.44
500	40	8.18	28	5.70	33	6.88	24	5.09
600	41	14.90	28	10.12	34	12.44	22	8.31
700	42	22.97	29	15.79	35	18.85	23	12.68
800	43	32.99	30	23.01	36	27.45	24	18.72
900	44	36.79	30	24.87	37	30.94	25	21.25
1000	51	78.20	32	49.82	38	58.46	26	40.26

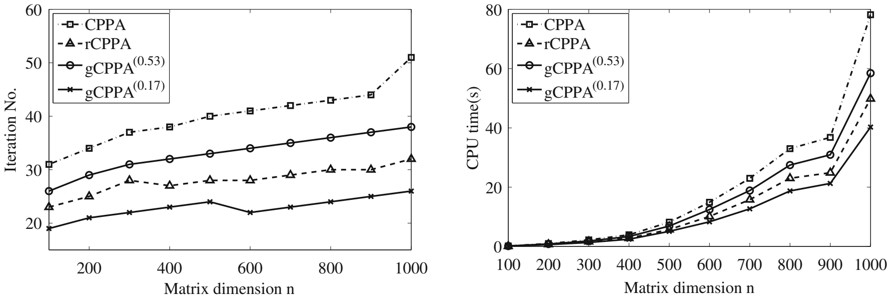

Fig. 1 Comparisons of CPPA, rCPPA, gCPPA$^{(0.53)}$ and gCPPA$^{(0.17)}$. Left: iteration number. Right: cpu time

One can conclude from Table 1 and Fig. 1 that: (1) the gCPPA$^{(0.53)}$ takes 80% iterations and time of the CPPA, but takes more iterations and time than the rCPPA; (2) the gCPPA$^{(0.17)}$ takes 60% iterations of the CPPA and 80% iterations of the rCPPA, and spends less cpu-time than the CPPA and rCPPA, especially for the high dimension problem. In other word, the gCPPA$^{(0.53)}$ and gCPPA$^{(0.17)}$ are better than CPPA, and gCPPA$^{(0.17)}$ has almost better numerical performance as the rCPPA with $\gamma = 1.50$. However, the convergence of gCPPA$^{(0.17)}$ is not established since $\alpha < 0.5$. It will be left to our future research.

Example 3.2 The second test problem is the matrix completion problem utilized in Tao, Yuan and He [30], which is as follows:

$$\min\left\{ \frac{1}{2}\|X - C\|_F^2 \;\middle|\; X \in S_+^n \cap S_{\mathrm{B}} \right\}, \tag{3.3}$$

where $S_{\mathrm{B}} = \{H \in \mathbb{R}^{n\times n} \mid H_{\mathrm{L}} \geq H \geq H_{\mathrm{U}}\}$.

By introducing an auxiliary variable Y such that $X - Y = 0$, problem (3.3) can be reformulated to a separable convex optimization problem of the form

$$\min\left\{ \frac{1}{2}\|X - C\|_F^2 + \frac{1}{2}\|Y - C\|_F^2 \;\middle|\; X = Y,\ X \in S_+^n,\ Y \in S_{\mathrm{B}} \right\}. \tag{3.4}$$

Obviously, problem (3.4) is a special case of problem (1.9) in which $\rho(A^{\mathrm{T}}A) = \rho(B^{\mathrm{T}}B) = 1$.

For comparisons, four methods are used for solving problem (3.4): classical ADMM (ADMM for short), gADMM, separable version of CPPA (i.e., Algorithm 2.7 with $\alpha = 1$, sCPPA) and eCPPA (Algorithm 2.7 with $0 < \alpha_0 < \alpha < 1$). Let $Z \in \mathbb{R}^{n\times n}$ be the Lagrange multiplier for linear constraint $X - Y = 0$. For a given (X^k, Y^k, Z^k), the eCPPA produces new iterate $(X^{k+1}, Y^{k+1}, Z^{k+1})$ via

$$\begin{cases} Z^{k+1} = Z^k - \frac{\alpha}{t}(X^k - Y^k), \\ X^{k+1} = \arg\min_{X\in S_+^n} \left\{ \frac{1}{2}\left\| X - \frac{1}{r+1}[rX^k + (1+\alpha)Z^{k+1} - \alpha Z^k + C] \right\|_F^2 \right\}, \\ Y^{k+1} = \arg\min_{Y\in S_B} \left\{ \frac{1}{2}\left\| Y - \frac{1}{s+1}[sY^k - (1+\alpha)Z^{k+1} + \alpha Z^k + C] \right\|_F^2 \right\}. \end{cases} \tag{3.5}$$

The X-subproblem of iteration (3.5) is inherent a projection operator via SVD decomposition, and it takes the main computation load in each iteration. The Y-subproblem of (3.5) is also a projection

$$Y^{k+1} = P_{S_B}\left\{ \frac{1}{s+1}\left[sY^k - (1+\alpha)Z^{k+1} + \alpha Z^k + C \right] \right\},$$

where P_{S_B} denotes the projection onto S_B under the Euclidean norm.

In the experiments, we take a random matrix $C = (c_{ij})_{n\times n}$ satisfying $c_{ij} \in (-1, 1)$ and $c_{ij} = c_{ji}$ for all i, j. For each given n, 20 random instances are tested. To be fair, X^0 and Y^0 are set to $n \times n$ identity matrice and z^0 is set to n-dimensional zero vector.

For termination criterion of all methods, we set $\varepsilon = 10^{-5}\|w^1 - w^0\|_\infty$.

The parameter settings of the used methods are stated as follows: The penalty parameter is set to $\beta = 1.80$ in both ADMM and gADMM, the relaxation parameter is set to $\gamma = 1.80$ in the gADMM. Let $R = 3.00$, the proximal parameters of the sCPPA are set to $r = s = \sqrt{2}R \approx 4.24$ and $t = 1.02\sqrt{2}/R \approx 0.48$, and those of the eCPPA are set to $r = s = \sqrt{2}\alpha_r R \approx 2.55$, $t = 1.02\sqrt{2}\alpha_t/R \approx 0.24$, where $\alpha_r = 0.60$ and $\alpha_t = 0.50$, and $\alpha = \sqrt{\alpha_r\alpha_t} \approx 0.55$.

We test all methods on 20 random instances for each fixed n. The average iteration number and the average cpu-time for problem (3.4) with different n are listed in Table 2 and displayed in Fig. 2.

From Table 2 and Fig. 2, one can conclude that the iteration number of eCPPA is about 30% of ADMM, and about 70% of gADMM and sCPPA. Moreover, eCPPA takes less cpu-time than others especially for the high-dimensional problems. Thus, eCPPA has the best performance for problem (3.4).

Table 2 Numerical results of ADMM, gADMM, sCPPA and eCPPA

n	ADMM		gADMM		sCPPA		eCPPA	
	Iter	cput	Iter	cput	Iter	cput	Iter	cput
100	120	0.65	60	0.34	65	0.35	48	0.26
200	127	4.19	64	2.10	74	2.21	52	1.62
300	146	10.14	74	5.13	80	5.06	56	3.68
400	145	17.54	73	9.34	76	8.91	58	6.78
500	153	29.19	77	14.79	78	14.34	60	11.27
600	169	52.99	84	26.52	80	24.25	62	19.20
700	179	81.60	89	40.67	82	35.66	63	27.98
800	162	107.26	80	53.36	84	53.18	64	42.06
900	180	160.57	89	79.51	86	74.71	66	58.29
1000	183	230.29	90	113.43	88	109.65	68	85.60

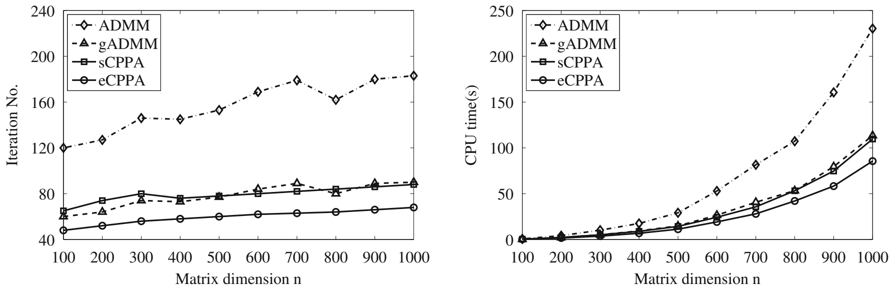

Fig. 2 Comparisons of ADMM, gADMM, sCPPA and eCPPA. Left: iteration number. Right: cpu time

Example 3.3 The third example selected from [13] focuses on the wavelet-based image inpainting and zooming problems. Let a two-dimensional image $x \in \mathbb{R}^{l_1 \times l_2}$ and $l = l_1 \cdot l_2$. Letting $\mathbf{vec}(\cdot)$ be the vectorization operator in the lexicographic order, $\mathbf{x} = \mathbf{vec}(x) \in \mathbb{R}^l$ is tackled by vectorizing x as an one-dimensional vector in the lexicographic order. Let $D \in \mathbb{R}^{l \times n}$ be a wavelet dictionary, each column of D is the elements of a wavelet frame. Commonly, the image $\mathbf{x}$ possesses a sparse representation under dictionary D, i.e., $\mathbf{x} = Dx$ where x is a sparse vector. The wavelet-based image processing considers recovering the real image $\mathbf{x}$ from an observation b which might have some missing pixels or convolutions. The model for wavelet-based image processing can be casted as

$$\min \left\{ \|\mathbf{x}\|_1 \middle| BD\mathbf{x} = b \right\}, \tag{3.6}$$

where $\|\mathbf{x}\|_1$ is to deduce a sparse representation under the wavelet dictionary, and B (also called mask) is a typically ill-conditioned matrix representation of convolution or downsampling operators. For the inpainting problem, matrix $B \in \mathbb{R}^{l \times l}$ in (3.6) is a diagonal matrix whose diagonal elements are either 0 or 1, where the locations of 0 correspond to missing pixels in the image and locations of 1 correspond to the pixels to be kept. For the image zooming problem, matrix $B \in \mathbb{R}^{m\times l}$ can be expressed as $B = SH$ where $S \in \mathbb{R}^{m\times l}$ is a downsampling matrix and $H \in \mathbb{R}^{l\times l}$ is a blurry matrix. Matrix H can be diagonalized by the discrete consine transform (DCT), i.e., $H = C^{-1}\Lambda C$ where C represents the DCT and Λ is a diagonal matrix whose diagonal entries are eigenvalues of H.

We compare the gCPPA method to the CPPA and rCPPA methods on 256 × 256 images of Peppers.png and Boat.png for the image inpainting and image zooming problems, respectively.

The dictionary D is chosen as the inverse discrete Haar wavelet transform with level 6. Below we give the detail of how the tested images are degraded.

- For the image inpainting problem, the original image Peppers is first blurred by the out-of-focus kernel with a radius of 7. Then 60% pixels of the blurred images are lost by implementing a mask operator S. The positions of missing pixels are located randomly.
- For the image zooming problem, the original image Boat is downsampled by a downsampling matrix S with a factor of 4. Then, the downsampled image is corrupted by a convolution whose kernel is generated by `fspecial(gaussian, 9,2.5)` of MATLAB.

Let z be the Lagrange multiplier. For a given $(\mathbf{x}^k, z^k)$, the gCPPA adopts the following iteration to obtain $(\mathbf{x}^{k+1}, z^{k+1})$:

$$\begin{cases} z^{k+1} = z^k - \frac{\alpha}{t}(BD\mathbf{x}^k - b), \\ \mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ \|\mathbf{x}\|_1 + \frac{r}{2}\left\|\mathbf{x} - \mathbf{x}^k - \frac{1}{r}BD\big[(\alpha+1)z^{k+1} - \alpha z^k\big]\right\|^2 \right\}. \end{cases}$$

On problem (3.6), the performance of the gCPPA (compared with CPPA and rCPPA) is tested. Similarly, the iteration scheme of the CPPA has the same iteration as the gCPPA with $\alpha = 1$, and the rCPPA can also be written in the form of (1.7) and (1.8). Since $DD^{\mathrm{T}} = I$, the blurry matrix H can be diagonalized by DCT and the binary matrix (both mask and downsampling matrices) S satisfies $\|S\| = 1$, we have $\|(BD)^{\mathrm{T}}(BD)\| = 1$ for the wavelet-based image inpainting and zooming problems. Therefore, the requirement $rt \geq \|(BD)^{\mathrm{T}}(BD)\|$ reduces to $rt \geq 1$ for the CPPA or rCPPA, and $rt > \alpha^2$ for the gCPPA. The parameters are described as below.

- For the image inpainting problem, we take initial point $(\mathbf{x}^0, \upsilon^0) = (D^{\mathrm{T}}(b), \mathbf{0})$, and $R = 0.60$, $r_0 = R = 0.60$, $t_0 = 1.02/R = 1.70$, $\gamma = 1.00$ for the CPPA and $\gamma = 1.90$ for the rCPPA. Take $\alpha_r = 2.00$, $\alpha_t = 0.13$, $\alpha = \sqrt{0.26} \approx 0.51$, $r = \alpha_r R = 1.20$, $t = 1.02\alpha_t/R \approx 0.22$ for the gCPPA;
- For the image zooming problem, we take initial point $(\mathbf{x}^0, \upsilon^0) = (\mathbf{0}, \mathbf{0})$, and $R = 0.55$, $r_0 = R = 0.55$, $t_0 = 1.02/R \approx 1.85$, $\gamma = 1.00$ for the CPPA and $\gamma = 1.20$ for the rCPPA. Take $\alpha_r = 1.00$, $\alpha_t = 0.26$, $a = \sqrt{0.26} \approx 0.51$, $r = \alpha_r R = 0.55$, $t = 1.02\alpha_t/R \approx 0.48$ for the gCPPA.

As usual, the quality of the reconstruction is measured by the signal-to-noise ratio (SNR) in decibel (dB)

$$\mathrm{SNR} := 20\log_{10}\frac{\|\mathbf{x}\|}{\|\bar{\mathbf{x}} - \mathbf{x}\|}, \tag{3.7}$$

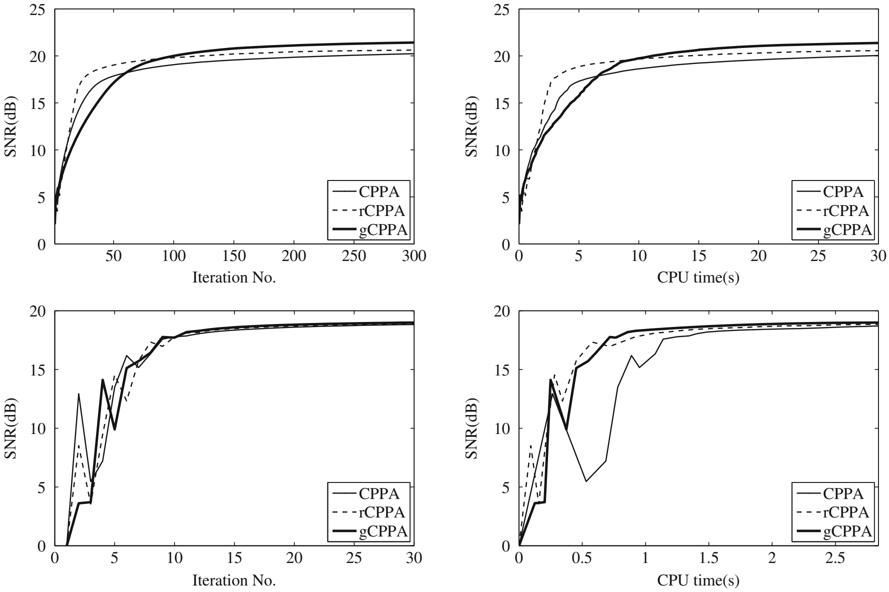

Fig. 3 Evolutions of SNR with respect to iterations (Iteration No.) and CPU time (*top row:* for image inpainting; *bottom row:* for image zooming)

where $\bar{\mathbf{x}}$ is a reconstructed image and $\mathbf{x}$ is a clean image.

For easy comparisons, the evolutions of SNR with respect to iterations and computing time for all tested methods are displayed in Fig. 3. It shows that the rCPPA is better than the CPPA, and the gCPPA converges to the nearly same solutions in inpainting and zooming problems as fast as the rCPPA.

The reconstructed images are displayed in Fig. 4 by executing 300 iterations in image inpainting (or 30 iterations in image zooming). It verifies our assertion: the gCPPA could be at least as efficient as the rCPPA on the inpainting and zooming problem. We have to emphasize that the gCPPA does not involve any relaxation step while the rCPPA has a relaxation step with $\gamma > 1$.

Example 3.4 The last example selected from [13] focuses on the total variation (TV) uniform noise removal model:

$$\min\left\{\|\nabla\mathbf{x}\|_1 \middle| \|H\mathbf{x}-\mathbf{x}^0\|_\infty \le \sigma\right\}, \tag{3.8}$$

where the two-dimensional image $x = (x_{i,j})_{l_1\times l_2} \in \mathbb{R}^{l_1\times l_2}$ and $l = l_1\cdot l_2$. The vectorized version of an image can be denoted by $\mathbf{x} = \mathbf{vec}(x) \in \mathbb{R}^l$ as Example 3.3, and $\|\nabla\mathbf{x}\|_1 := \sum_{i=1}^{l}(|\text{vec}(\nabla_h x)|_i + |\text{vec}(\nabla_v x)|_i)$ is the TV norm with $\nabla_h x = \big((\nabla_h x)_{i,j}\big)_{l_1\times l_2}$, $\nabla_v x = \big((\nabla_v x)_{i,j}\big)_{l_1\times l_2}$ where

$$(\nabla_h x)_{i,j} = \begin{cases} x_{i+1,j} - x_{i,j}, & i \neq l_1 \\ x_{1,j} - x_{l_1,j}, & i = l_1 \end{cases}, \quad (\nabla_v x)_{i,j} = \begin{cases} x_{i,j+1} - x_{i,j}, & j \neq l_2 \\ x_{i,1} - x_{i,l_2}, & j = l_2 \end{cases}.$$

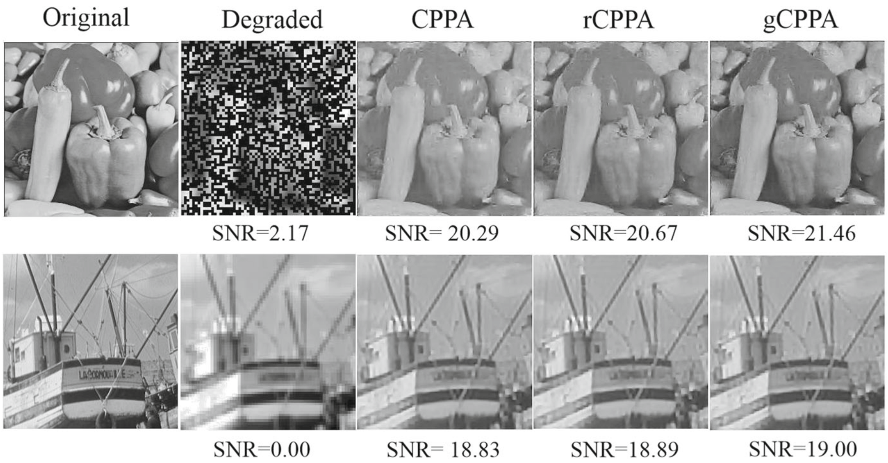

Fig. 4 Original images, degraded images and reconstructed images by CPPA, rCPPA and gCPPA (*top row:* for image inpainting; *bottom row:* for image zooming)

The vector $\mathbf{x}^0 \in \mathbb{R}^l$ is an observed image corrupted by a zero-mean uniform noise; $H \in \mathbb{R}^{l\times l}$ is the matrix representation of a blurry operator satisfying $\|HH^{\mathrm{T}}\| = 1$, σ is a parameter indicating the uniform noise level and $\|\mathbf{x}\|_\infty := \max_{1\le i\le l} \|\mathbf{x}_i\|$.

Let

$$A := \begin{pmatrix} H \\ -H \end{pmatrix}, \quad b := \begin{pmatrix} \mathbf{x}^0 - \sigma\mathbf{e} \\ -\mathbf{x}^0 - \sigma\mathbf{e} \end{pmatrix},$$

where $\mathbf{e} = (1, 1, \cdots, 1)^{\mathrm{T}} \in \mathbb{R}^l$. Model (3.8) can be reformulated to

$$\min\left\{\|\nabla\mathbf{x}\|_1 \;\middle|\; A\mathbf{x} \ge b\right\}. \tag{3.9}$$

We also compare the gCPPA to the CPPA and rCPPA methods on 256-by-256 images of Peppers.png and Boat.png. The clean images are degraded by either the Gaussian (fspecial(gaussian,9,2.5)”) or the out-of-focus (fspecial(disk,3)”) convolution.

Let $\mathbf{z}$ be the Lagrange multiplier for the linear inequality constraint $A\mathbf{x} \ge b$. The gCPPA produces the new iterate via:

$$\begin{cases} \mathbf{z}^{k+1} = P_{R_+}\left\{\mathbf{z}^k - \frac{\alpha}{t}(A\mathbf{x}^k - b)\right\}, \\ \mathbf{x}^{k+1} = \arg\min\left\{\|\nabla\mathbf{x}\|_1 + \frac{r}{2}\left\|\mathbf{x} - \mathbf{x}^k - \frac{1}{r}A^{\mathrm{T}}\left[(\alpha+1)\mathbf{z}^{k+1} - \alpha\mathbf{z}^k\right]\right\|^2 \;\middle|\; \mathbf{x} \in X\right\}. \end{cases}$$

The CPPA method also owns above iteration with $\alpha = 1$, and the rCPPA can be written in the form of (1.7) and (1.8). We take $R = 0.60$, $r_0 = R$, $t_0 = 1.02/R = 1.70$ and $\gamma = 1.80$ for the CPPA and rCPPA method, and take $\alpha_r = 7.00$, $\alpha_t = 0.04$, $\alpha = \sqrt{0.28} \approx 0.53$, $r = \alpha_r R = 4.20$, $t = 1.02\alpha_t/R \approx 0.07$ for the gCPPA. The initial points for all methods are taken as zeros and the stopping criterion is set to

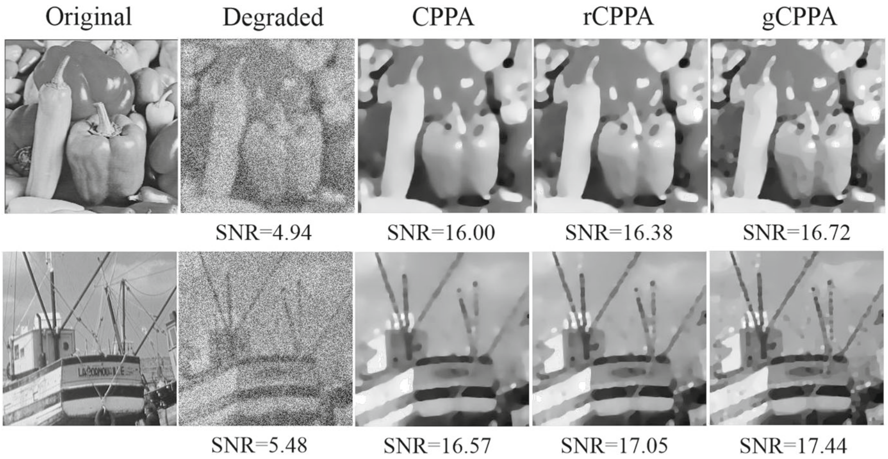

Fig. 5 Original images, degraded images with $\sigma = 0.5$ and reconstructed images by the CPPA, rCPPA, gCPPA (*top row:* the gaussian convolution; *bottom row:* the out-of-focus convolution)

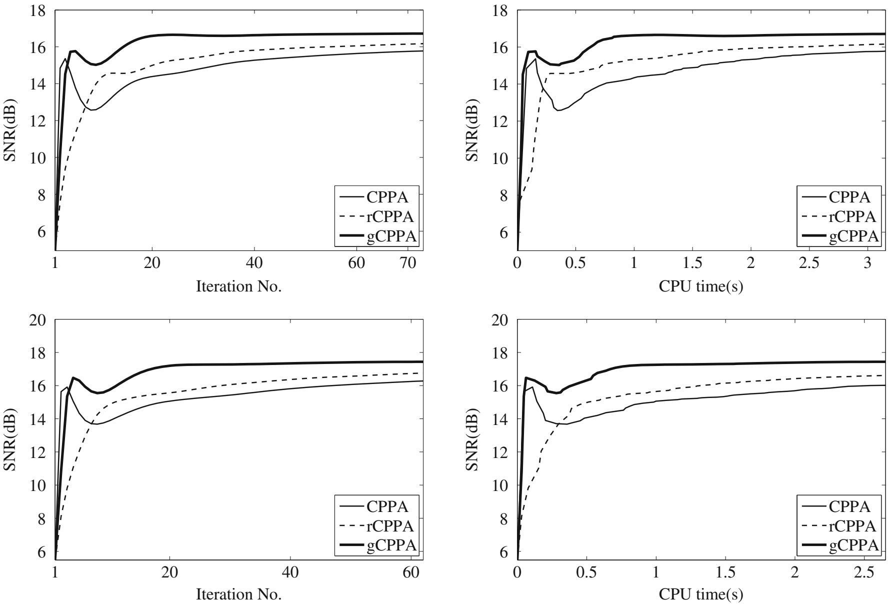

Fig. 6 Evolutions of SNR with respect to the number of iterations (Iteration No.) and computing time (CPU time) (*top row:* the gaussian convolution with $\sigma = 0.5$; *bottom row:* the out-of-focus convolution with $\sigma = 0.5$;)

$$Tol = \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|}{\|\mathbf{x}^k\|} < 10^{-6}. \tag{3.10}$$

Set $\sigma = 0.5$. For easy comparisons, the original images, degraded images and the restored images are displayed in Fig. 5.

Table 3 Results of the CPPA, rCPPA and gCPPA with different σ

Blur	σ	Images	CPPA	rCPPA	gCPPA
Gaussian	0.2	Peppers	72/3.39/16.75	87/4.17/17.13	58/2.53/17.57
		Boat	65/3.01/16.58	79/3.48/17.01	58/2.50/17.56
	0.5	Peppers	103/5.46/16.00	121/5.58/16.38	73/2.82/16.72
		Boat	83/3.78/15.85	101/4.40/16.33	71/2.92/16.79
Out-of-focus	0.2	Peppers	71/3.24/17.67	88/4.77/18.12	50/2.37/18.54
		Boat	65/3.51/17.53	82/3.59/18.07	51/2.06/18.64
	0.5	Peppers	94/4.38/16.65	108/4.93/17.01	64/2.96/17.27
		Boat	86/3.40/16.57	104/4.73/17.05	62/2.65/17.44

The evolutions of SNR with respect to iterations and computing time are displayed in Fig. 6. It shows that both the CPPA and gCPPA methods reach the higher SNR in a short time. However, the CPPA converges to the solutions worse than that of the rCPPA, while the gCPPA converges to better solutions than the rCPPA in TV uniform noise removal problem.

The numerical results on more experiments with different σ are put into Table 3 for comparisons.

In Table 3, each set of $\cdot \backslash \cdot \backslash \cdot$ represents the number of iterations, the computing time in seconds and the restored SNR when the stopping criterion (3.10) is met. From Table 3, one can conclude that: the rCPPA has better performance comparing with the CPPA in SNR, but worse in the number of iterations and cpu-time, while the gCPPA has better performance comparing with the CPPA in all terms including SNR, the number of iterations and cpu-time.

4 Conclusions

In this paper, we propose two customized proximal point algorithm by modifying the prediction step involving no relaxation step. We have proposed two new CPPAs (i.e., gCPPA and eCPPA), both do not involve any relaxation step but still keep the same convergence properties as the CPPA within some relaxation steps. Numerical results have demonstrated that the proposed methods are more effective than some state-of-the-art methods.

Acknowledgements The authors are very grateful to Doctor Wen-Xing Zhang at University of Electronic Science and Technology of China for his help on numerical experiments on image processing problems. This work was also supported by the Science and Technology on Communication Information Security Control Laboratory, Xiangtan University.

References

1. Moreau, J.J.: Proximité et dualité dans un espace hilbertien. Bull. Soc. Math. Fr. **93**, 273–299 (1965)
2. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM J. Control Optim. **14**(5), 877–898 (1976)

3. Bertsekas, D.P., Tseng, P.: Partial proximal minimization algorithms for convex programming. SIAM J. Optim. **4**(3), 551–572 (1993)
4. Kaplan, A., Tichatschke, R.: Proximal point methods and nonconvex optimization. J. Glob. Optim. **13**(4), 389–406 (1998)
5. Martinet, B.: Brève communication. Regularisation, d'iné quations variationelles par approximations succesives. ESAIM-Math. Model. Numer. Anal.-Model. Math. Anal. Numer **4**(R3), 154–158 (1970)
6. Kassay, G.: The proximal points algorithm for reflexive Banach spaces. Stud. Math. **30**, 9–17 (1985)
7. Güler, O.: New proximal point algorithms for convex minimization. SIAM J. Optim. **2**(4), 646–664 (1992)
8. Spingarn, J.E.: Partial inverse of a monotone operator. Appl. Math. Optim. **10**(10), 247–265 (1983)
9. Han, S.P.: A decomposition method and its application to convex programming. Math. Oper. Res. **14**(2), 237–248 (1989)
10. Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. UCLA CAM. Rep. **34**, 1–29 (2008)
11. He, B.S.: PPA-like contraction methods for convex optimization: a framework using variational inequality approach. J. Oper. Res. Soc. China **3**(4), 391–420 (2015)
12. He, B.S., Fu, X.L., Jiang, Z.K.: Proximal point algorithm using a linear proximal term. J. Optim. Theory Appl. **141**(2), 299–319 (2009)
13. He, B.S., Yuan, X.M., Zhang, W.: A customized proximal point algorithm for convex minimization with linear constraints. Comput. Optim. Appl. **56**(3), 559–572 (2013)
14. Hestenes, M.R.: Multiplier and gradient methods. J. Optim. Theory Appl. **4**(4), 303–320 (1969)
15. Powell, M.J.D.: A method for nonlinear constraints in minimization problems. Optimization **5**(6), 283–298 (1969)
16. Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Math. Oper. Res. **1**(2), 97–116 (1976)
17. He, B.S.: Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities. Comput. Optim. Appl. **42**(2), 195–212 (2009)
18. Glowinski, R., Marrocco, A.: Analyse numerique du champ magnetique d'un alternateur par elements finis et sur-relaxation ponctuelle non lineaire. Comput. Meth. Appl. Mech. Eng. **3**(1), 55–85 (1974)
19. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Comput. Math. Appl. **2**(1), 17–40 (1976)
20. Gabay, D.: Chapter IX: applications of the method of multipliers to variational inequalities. Stud. Math. Appl. **15**, 299–331 (1983)
21. Douglas, J., Rachford, H.H.: On the numerical solution of heat conduction problems in two and three space variables. Trans. Am. Math. Soc. **82**(2), 421–439 (1956)
22. Cai, X.J., Gu, G.Y., He, B.S., Yuan, X.M.: A proximal point algorithm revisit on the alternating direction method of multipliers. Sci. China Math. **56**(10), 2179–2186 (2013)
23. He, B.S., Yuan, X.M.: On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. SIAM J. Numer. Anal. **50**(2), 700–709 (2012)
24. He, B.S., Yuan, X.M.: On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers. Numer. Math. **130**(3), 567–577 (2014)
25. Sarkis, J.: A strategic decision framework for green supply chain management. J. Clean. Prod. **11**(4), 397–409 (2003)
26. Wu, Z., Pagell, M.: Balancing priorities: Decision-making in sustainable supply chain management. J. Oper. Manag. **29**(6), 577–590 (2011)
27. Yazdani, M., Zarate, P., Coulibaly, A., Zavadskas, E.K.: A group decision making support system in logistics and supply chain management. Expert Syst. Appl. **88**, 376–392 (2017)
28. Gu, G., He, B.S., Yuan, X.M.: Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: a unified approach. Comput. Optim. Appl. **59**, 135–161 (2014)
29. Cai, J.F., Candès, Emmanuel J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**(4), 1956–1982 (2008)
30. Tao, M., Yuan, X.M., He, B.S.: Solving a class of matrix minimization problems by linear variational inequality approaches. Linear Alg. Appl. **434**(11), 2343–2352 (2011)