General parameterized proximal point algorithm with applications in statistical learning [*]

Jianchao Bai[a†]   Jicheng Li[a]   Pingfan Dai[a,b]   Jiaofen Li[c]

[a] *School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, P.R. China*
[b] *Department of Information Engineering, Sanming University, Sanming 365004, P.R. China*
[c] *College of Mathematics and Computational Science, Guilin University of*
Electronic Technology, Guilin 541004, P.R. China

Abstract

In the literature, there are a few researches to design some parameters in the Proximal Point Algorithm (PPA), especially for the multi-objective convex optimizations. Introducing some parameters to PPA can make it more flexible and attractive. Mainly motivated by our recent work (Bai et al., A parameterized proximal point algorithm for separable convex optimization, Optim. Lett. (2017) doi: 10.1007/s11590-017-1195-9), in this paper we develop a general parameterized PPA with a relaxation step for solving the multi-block separable structured convex programming. By making use of the variational inequality and some mathematical identities, the global convergence and the worst-case $\mathcal{O}(1/t)$ convergence rate of the proposed algorithm are established. Preliminary numerical experiments on solving a sparse matrix minimization problem from statistical learning validate that our algorithm is more efficient than several state-of-the-art algorithms.

Keywords: Structured convex programming; Proximal point algorithm; Relaxation step; Complexity; Statistical learning
AMS subject classifications(2010): 65C60; 65Y20; 90C25

1 Introduction

Throughout this paper, let $\mathbb{R}(\mathbb{R}^+), \mathbb{R}^m, \mathbb{R}^{m \times n}$ be the set of real (positive) numbers, the set of m dimensional real column vectors and the set of $m \times n$ dimensional real matrices, respectively. The symbol $\|z\|_2$ denotes the Euclidean norm of $z \in \mathbb{R}^m$, which is defined by $\|z\|_2 = \sqrt{\langle z, z \rangle}$ with the standard inner product $\langle \cdot, \cdot \rangle$. For any symmetric positive definite matrix $G \in \mathbb{R}^{m \times m}$, $\|z\|_G = \sqrt{\langle z, Gz \rangle}$ represents the weighted G-norm of z. We also use $^\mathsf{T}$ and $\mathbf{I}$ to stand for the transpose of a vector/matrix and the identity matrix with proper dimensions, respectively.

[†] Corresponding author. E-mail address: bjc1987@163.com.

Consider the following multi-block separable structured convex optimization

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{p} f_i(x_i) \\
\text{s.t.} \quad & \sum_{i=1}^{p} A_i x_i = b, \ x_i \in \mathcal{X}_i,
\end{aligned} \tag{1.1}
$$

where $p > 1$ is a positive integer, $f_i(\cdot) : \mathbb{R}^{n_i} \to \mathbb{R}$ are closed convex functions (possibly nonsmooth); $A_i \in \mathbb{R}^{m \times n_i}, b \in \mathbb{R}^m$ are respectively given matrices and vector; all structured sets $\mathcal{X}_i \subset \mathbb{R}^{n_i} (i = 1, \cdots, p)$ are closed and convex. Throughout the discussions, we make the following assumption:

Assumption 1.1 *The solution set of the problem (1.1) is nonempty and all the matrices $A_i (i = 1, \cdots, p)$ have full column rank.*

Note that the first part of Assumption 1.1 is basic and the second part is necessary. For instance, when dealing with a minimization problem subject to the linearly equality constraint $Ax = b$ involving large-size coefficient matrix and variable, we can split

$$
A = [A_1, \cdots, A_p], \quad x = (x_1^{\mathsf{T}}, \cdots, x_p^{\mathsf{T}})^{\mathsf{T}}
$$

to reduce the dimensions of A and x, then in such case each A_i has full column rank. In fact, many practical application problems do contain at least two different variables (or can be transformed into an equivalent problem with at least two variables), e.g. the total-variational image deblurring problems [10, 19], the transformed joint sparse recovery problem [11], the sparse inverse covariance estimation problem [5], the low-rank and sparse problem [12, 18] and so forth.

The Proximal Point Algorithm (PPA), which was originally proposed to tackle the monotone operator inclusion problems [13, 14], is regarded as a powerful benchmark method for solving the convex problems like (1.1). As verified by Rockafellar [17], the well-known Augmented Lagrangian Method (ALM) for (1.1) with $p = 2$ was actually an application of PPA to its dual problem. Moreover, the classical Alternating Direction Method of Multipliers (ADMM) can be also treated as a special variant of PPA to the dual problem [4]. In the last several years, there has been a constantly increasing interest in developing the theories and applications of PPA. For example, He et al.[7] showed a customized application of the classical PPA to the model (1.1) with $p = 1$, where some image processing problems were carried out to show the efficiency of the method therein. Ma and Ni [15] proposed a parameterized PPA for (1.1) with $p = 1$, where both the basis pursuit problem and the matrix completion problem were tested in experiments to exam the numerical performance of their algorithm. Recently, Cai et al.[2] designed a PPA with a relaxation step for the model (1.1) with $p = 2$, whose global convergence and the worst-case sub-linear convergence rate were analyzed in detail. More recently, by introducing some parameters to the metric proximal matrix, an extended parameterized PPA based on [15] was developed for the two block separable convex programming [1], whose effectiveness and robustness was demonstrated by testing a sparse vector optimization problem in the statistical learning compared with two popular algorithms.

To the best of our knowledge, there are a few researches on the parameterized PPA for solving the multi-block model (1.1) with at least three variables. Based on such observation and motivated by our recent published work [1], the aims of this article are to design a general parameterized PPA with a relaxation step (**GR-PPA**) for solving (1.1) and to test some practical examples having more than two variables to investigate the performance of our GR-PPA. In the remaining parts, Section 2 shows the details of constructing the proposed GR-PPA and analyzing its convergence theories. In Section 3, some numerical experiments are carried out in terms of different parameters, tolerances and initial values. Finally, we conclude the paper in Section 4.

2 Main results

In this section, we first construct a parameterized proximal matrix to design a novel GR-PPA for (1.1), where its convergence is analyzed in detail afterwards. The whole convergence analysis is based on the variational inequality and uses some special techniques to simplify its proof.

2.1 Formation of GR-PPA

For any $\tau \in \mathbb{R}^+$, the Lagrangian function of (1.1) is constructed as

$$L(x_1, \cdots, x_p, \lambda) = \sum_{i=1}^{p} f_i(x_i) - \tau \left\langle \lambda, \sum_{i=1}^{p} A_i x_i - b \right\rangle, \tag{2.1}$$

where $\lambda \in \mathbb{R}^m$ denotes the Lagrange multiplier with respect to the equality constraint. Let $(x_1^*, \cdots, x_p^*, \lambda^*)$ be the saddle-point belonging to the solution set Ω^* of (1.1). Then, the following basic inequalities

$$L(x_1^*, \cdots, x_p^*, \lambda) \le L(x_1^*, \cdots, x_p^*, \lambda^*) \le L(x_1, \cdots, x_p, \lambda^*),$$

imply

$$\begin{cases} x_1^* = \arg \min_{x_1 \in \mathcal{X}_1} \left\{ f_1(x_1) - \tau \left\langle \lambda^*, A_1 x_1 \right\rangle \right\}, \\ \vdots \\ x_p^* = \arg \min_{x_p \in \mathcal{X}_p} \left\{ f_p(x_p) - \tau \left\langle \lambda^*, A_p x_p \right\rangle \right\}, \\ \lambda^* = \arg \max_{\lambda \in \mathbb{R}^m} -\tau \left\langle \lambda, \sum_{i=1}^{p} A_i x_i^* - b \right\rangle, \end{cases}$$

whose optimality conditions are derived as follows

$$\begin{cases} x_1^* \in \mathcal{X}_1, & f_1(x_1) - f_1(x_1^*) + \left\langle x_1 - x_1^*, -\tau A_1^{\mathsf{T}} \lambda^* \right\rangle \ge 0, & \forall x_1 \in \mathcal{X}_1, \\ \vdots \\ x_p^* \in \mathcal{X}_p, & f_p(x_p) - f_p(x_p^*) + \left\langle x_p - x_p^*, -\tau A_p^{\mathsf{T}} \lambda^* \right\rangle \ge 0, & \forall x_p \in \mathcal{X}_p, \\ \lambda^* \in \mathbb{R}^m, & \left\langle \lambda - \lambda^*, \tau(\sum_{i=1}^{p} A_i x_i^* - b) \right\rangle \ge 0, & \forall \lambda \in \mathbb{R}^m. \end{cases}$$

It is not hard to reformulate the above optimality conditions into a variational inequality:

$$w^* \in \Omega^*, \quad \mathrm{VI}(\phi, \mathcal{J}, \Omega): \ \phi(u) - \phi(u^*) + \left\langle w - w^*, \mathcal{J}(w^*) \right\rangle \ge 0, \ \forall w \in \Omega, \tag{2.2}$$

where

$$\phi(u) = \sum_{i=1}^{p} f_i(x_i), \quad \Omega = \mathcal{X}_1 \times \cdots \times \mathcal{X}_p \times \mathbb{R}^m,$$

$$u = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}, \quad w = \begin{pmatrix} x_1 \\ \vdots \\ x_p \\ \lambda \end{pmatrix} \text{ and } \mathcal{J}(w) = \tau \begin{pmatrix} -A_1^{\mathsf{T}} \lambda \\ \vdots \\ -A_p^{\mathsf{T}} \lambda \\ \sum_{i=1}^{p} A_i x_i - b \end{pmatrix}.$$

In the following, we also denote

$$u^k = \begin{pmatrix} x_1^k \\ x_2^k \\ \vdots \\ x_p^k \end{pmatrix}, \quad w^k = \begin{pmatrix} x_1^k \\ \vdots \\ x_p^k \\ \lambda^k \end{pmatrix} \text{ and } \mathcal{J}(w^k) = \tau \begin{pmatrix} -A_1^{\mathsf{T}} \lambda^k \\ \vdots \\ -A_p^{\mathsf{T}} \lambda^k \\ \sum_{i=1}^{p} A_i x_i^k - b \end{pmatrix}. \tag{2.3}$$

Clearly, we can get from the skew-symmetric property of the mapping $\mathcal{J}(w)$ that

$$\left\langle w^{k+1} - w, \mathcal{J}(w^{k+1}) \right\rangle = \left\langle w^{k+1} - w, \mathcal{J}(w) \right\rangle, \ \forall \ w, w^{k+1} \in \Omega. \tag{2.4}$$

It is well-known that the standard PPA with given iterate w^k reads the following unified framework:

$$w^{k+1} \in \Omega, \quad \phi(u) - \phi(u^{k+1}) + \left\langle w - w^{k+1}, \mathcal{J}(w^{k+1}) + G\left(w^{k+1} - w^k\right) \right\rangle \geq 0, \quad \forall \ w \in \Omega, \tag{2.5}$$

where G is a symmetric positive definite matrix (called the proximal matrix). Followed by such framework, a general parameterized proximal matrix will be designed for the new GR-PPA. Concisely, let the matrix G be of following block form

$$G = \begin{bmatrix} \left(\sigma_1 + \frac{\varepsilon^2-1}{s}\right) A_1^{\mathsf{T}} A_1 & & & & -\varepsilon A_1^{\mathsf{T}} \\ & \left(\sigma_2 + \frac{\tau^2-1}{s}\right) A_2^{\mathsf{T}} A_2 & & & -\tau A_2^{\mathsf{T}} \\ & & \ddots & & \vdots \\ & & & \left(\sigma_p + \frac{\tau^2-1}{s}\right) A_p^{\mathsf{T}} A_p & -\tau A_p^{\mathsf{T}} \\ \hline -\varepsilon A_1 & -\tau A_2 & \cdots & -\tau A_p & s\mathbf{I} \end{bmatrix}, \tag{2.6}$$

where $(\sigma_1, \cdots, \sigma_p, s)$ are parameters restricted into the domain

$$\mathcal{K} = \left\{ \sigma_1 > \frac{1 + (p-1)\tau|\varepsilon|}{s}, \sigma_i > \frac{1 + (p-2)\tau^2 + \tau|\varepsilon|}{s}, s > 0 \mid \varepsilon \in \mathbb{R}, \tau \in \mathbb{R}^+, i = 2, \cdots, p \right\}. \tag{2.7}$$

For the sake of analysis convenience, here and next we denote

$$\bar{\sigma}_i := \sigma_i + \frac{\tau^2 - 1}{s}, \ \forall \ i = 1, \cdots, p. \tag{2.8}$$

Lemma 2.1 *Let $\mathcal{K}$ be defined in (2.7) and the matrices $A_i(i = 1, 2, \cdots, p)$ have full column rank. Then, the matrix G in (2.6) is symmetric positive definite for any $(\sigma_1, \cdots, \sigma_p, s) \in \mathcal{K}$.*

Proof Notice that the matrix G is symmetric and can be decomposed as

$$G = D^{\mathsf{T}} G_0 D,$$

where $D = \mathrm{Diag}(A_1, \cdots, A_p, \mathbf{I})$ and

$$G_0 = \begin{bmatrix} \left(\sigma_1 + \frac{\varepsilon^2-1}{s}\right)\mathbf{I} & & & & -\varepsilon\mathbf{I} \\ & \left(\sigma_2 + \frac{\tau^2-1}{s}\right)\mathbf{I} & & & -\tau\mathbf{I} \\ & & \ddots & & \vdots \\ & & & \left(\sigma_p + \frac{\tau^2-1}{s}\right)\mathbf{I} & -\tau\mathbf{I} \\ \hline -\varepsilon\mathbf{I} & -\tau\mathbf{I} & \cdots & -\tau\mathbf{I} & s\mathbf{I} \end{bmatrix}. \tag{2.9}$$

Since all the matrices A_i are assumed to have full column rank, therefore, the matrix G is positive

definite if and only if G_0 is positive definite. Using the identity

$$
\begin{bmatrix}
\mathbf{I} & & & & \frac{\varepsilon}{s}\mathbf{I} \\
& \mathbf{I} & & & \frac{\tau}{s}\mathbf{I} \\
& & \ddots & & \vdots \\
& & & \mathbf{I} & \frac{\tau}{s}\mathbf{I} \\
& & & & \mathbf{I}
\end{bmatrix}
G_0
\begin{bmatrix}
\mathbf{I} & & & & \frac{\varepsilon}{s}\mathbf{I} \\
& \mathbf{I} & & & \frac{\tau}{s}\mathbf{I} \\
& & \ddots & & \vdots \\
& & & \mathbf{I} & \frac{\tau}{s}\mathbf{I} \\
& & & & \mathbf{I}
\end{bmatrix}^{\mathsf{T}}
$$

$$
=
\left[
\begin{array}{cccc|c}
\left(\sigma_1 - \frac{1}{s}\right)\mathbf{I} & -\frac{\varepsilon\tau}{s}\mathbf{I} & -\frac{\varepsilon\tau}{s_2}\mathbf{I} & \cdots & -\frac{\varepsilon\tau}{s}\mathbf{I} \\
-\frac{\tau\varepsilon}{s}\mathbf{I} & \left(\sigma_2 - \frac{1}{s}\right)\mathbf{I} & -\frac{\tau^2}{s}\mathbf{I} & \cdots & -\frac{\tau^2}{s}\mathbf{I} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-\frac{\tau\varepsilon}{s}\mathbf{I} & -\frac{\tau^2}{s}\mathbf{I} & -\frac{\tau^2}{s}\mathbf{I} & \cdots & \left(\sigma_p - \frac{1}{s}\right)\mathbf{I} \\
\hline
& & & & s\mathbf{I}
\end{array}
\right]
=: \widetilde{G}_0,
\tag{2.10}
$$

it is easy to demonstrate that G_0 is positive definite if and only if $\widetilde{G}_0$ is positive definite, which is guaranteed by the region $\mathcal{K}$ defined in (2.7) because in such case both the upper-left and lower-right block matrices of $\widetilde{G}_0$ are positive definite. $\square$

Remark 2.1 *According to the equation (3.7) in [8], the proximal matrix of their proposed algorithm can be decomposed as*

$$ G = P^{\mathsf{T}} \bar{G}_0 P $$

with $P = \mathrm{Diag}(\sqrt{\beta}A_1, \cdots, \sqrt{\beta}A_p, \mathbf{I}/\sqrt{\beta})$ and

$$
\bar{G}_0 =
\left[
\begin{array}{cccc|c}
\nu\mathbf{I} & -\mathbf{I} & \cdots & -\mathbf{I} & \mathbf{0} \\
-\mathbf{I} & \nu\mathbf{I} & \cdots & -\mathbf{I} & \mathbf{0} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
-\mathbf{I} & -\mathbf{I} & \cdots & \nu\mathbf{I} & \mathbf{0} \\
\hline
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I}
\end{array}
\right],
\tag{2.11}
$$

in which $\beta > 0$ is a penalty parameter with respect to the equality constraint of (1.1) and $\nu \geq m-1$ denotes the proximal parameter. Compared (2.10) to (2.11), if we set

$$ (s, \tau, \varepsilon) := (1, 1, 1) \quad \& \quad \sigma_i := \nu + 1, \quad \forall\, i = 1, 2, \cdots, p, $$

then it is clear that $\widetilde{G}_0 = \bar{G}_0$, which implies that the proximal matrix of [8] in essence is a special case of (2.6). That is, the method in [8] is a special case of our GR-PPA without relaxation step. A similar way can be used to analyze the proximal matrix in equation (8.6) in [6]. Such relations show that our parameterized proximal matrix is more flexible and general than some in the past.

In what follows, the concrete iterative principles of our GR-PPA are analyzed one by one. Substituting the matrix G into (2.5), we obtain

$$ \lambda^{k+1} \in \mathbb{R}^m, \quad \langle \lambda - \lambda^{k+1}, R_\lambda \rangle \geq 0, \quad \forall\, \lambda \in \mathbb{R}^m, $$

which is equivalent to

$$ 0 = R_\lambda := \tau\left(\sum_{i=1}^p A_i x_i^{k+1} - b\right) - \varepsilon A_1(x_1^{k+1} - x_1^k) - \tau\sum_{i=2}^p A_i(x_i^{k+1} - x_i^k) + s\left(\lambda^{k+1} - \lambda^k\right). $$

Then, it can be derived from the above equality that

$$ \lambda^{k+1} = \lambda^k - \frac{1}{s}\left[(\tau - \varepsilon)A_1 x_1^{k+1} + \varepsilon A_1 x_1^k + \tau\sum_{i=2}^p A_i x_i^k - b\tau\right]. \tag{2.12} $$

Combining (2.5)-(2.6) and (2.12), we have

$$x_1^{k+1} \in \mathcal{X}_1, \quad f_1(x_1) - f_1(x_1^{k+1}) + \langle x_1 - x_1^{k+1}, R_{x_1} \rangle \geq 0, \quad \forall\, x_1 \in \mathcal{X}_1, \tag{2.13}$$

where

$$
\begin{aligned}
R_{x_1} &= -\tau A_1^\mathsf{T} \lambda^{k+1} - \varepsilon A_1^\mathsf{T}(\lambda^{k+1} - \lambda^k) + \left(\sigma_1 + \tfrac{\varepsilon^2-1}{s}\right) A_1^\mathsf{T} A_1 (x_1^{k+1} - x_1^k) \\
&= \varepsilon A_1^\mathsf{T} \lambda^k - (\tau + \varepsilon) A_1^\mathsf{T} \left\{ \lambda^k - \tfrac{1}{s}\left[(\tau - \varepsilon) A_1 x_1^{k+1} + \varepsilon A_1 x_1^k + \tau \sum_{i=2}^{p} A_i x_i^k - b\tau \right] \right\} \\
&\quad + \left(\sigma_1 + \tfrac{\varepsilon^2-1}{s}\right) A_1^\mathsf{T} A_1 (x_1^{k+1} - x_1^k) \\
&= -\tau A_1^\mathsf{T} \lambda^k + \tfrac{\tau^2-\varepsilon^2}{s} A_1^\mathsf{T} A_1 x_1^{k+1} + \tfrac{(\tau+\varepsilon)\tau}{s} A_1^\mathsf{T} \left(\sum_{i=2}^{p} A_i x_i^k - b \right) + \left(\sigma_1 + \tfrac{\varepsilon^2-1}{s}\right) A_1^\mathsf{T} A_1 x_1^{k+1} \\
&\quad - \left(\sigma_1 + \tfrac{\varepsilon^2-1}{s}\right) A_1^\mathsf{T} A_1 x_1^k + \tfrac{(\tau+\varepsilon)\varepsilon}{s} A_1^\mathsf{T} A_1 x_1^k \\
&= -\tau A_1^\mathsf{T} \lambda^k + \left(\sigma_1 + \tfrac{\tau^2-1}{s}\right) A_1^\mathsf{T} A_1 x_1^{k+1} + \tfrac{(\tau+\varepsilon)\tau}{s} A_1^\mathsf{T} \left(\sum_{i=2}^{p} A_i x_i^k - b \right) \\
&\quad - \left[\sigma_1 + \tfrac{\tau^2-1-\tau(\tau+\varepsilon)}{s}\right] A_1^\mathsf{T} A_1 x_1^k \\
&= -\tau A_1^\mathsf{T} \bar{\lambda}^k + \bar{\sigma}_1 A_1^\mathsf{T} A_1 (x_1^{k+1} - x_1^k)
\end{aligned}
$$

with $\bar{\sigma}_1$ defined in (2.8) is positive by (2.7) and

$$\bar{\lambda}^k = \lambda^k - \frac{\tau + \varepsilon}{s}\left(\sum_{i=1}^{p} A_i x_i^k - b \right). \tag{2.14}$$

By (2.13) together with R_{x_1}, obviously, x_1^{k+1} is the solution of the following problem

$$
\begin{aligned}
x_1^{k+1} &= \arg\min_{x_1 \in \mathcal{X}_1} \left\{ f_1(x_1) - \langle A_1 x_1, \tau \bar{\lambda}^k \rangle + \frac{\bar{\sigma}_1}{2} \left\| A_1(x_1 - x_1^k) \right\|^2 \right\} \\
&= \arg\min_{x_1 \in \mathcal{X}_1} \left\{ f_1(x_1) + \frac{\bar{\sigma}_1}{2} \left\| A_1(x_1 - x_1^k) - \frac{\tau}{\bar{\sigma}_1} \bar{\lambda}^k \right\|^2 \right\}.
\end{aligned} \tag{2.15}
$$

Here we get from (2.12) and (2.14) that

$$
\begin{aligned}
\bar{\lambda}^{k+1} &= \lambda^{k+1} - \frac{\tau + \varepsilon}{s}\left(\sum_{i=1}^{p} A_i x_i^{k+1} - b \right) \\
&= \lambda^k - \frac{1}{s}\left[(\tau - \varepsilon) A_1 x_1^{k+1} + \varepsilon A_1 x_1^k + \tau \sum_{i=2}^{p} A_i x_i^k - b\tau \right] - \frac{\tau + \varepsilon}{s}\left(\sum_{i=1}^{p} A_i x_i^{k+1} - b \right) \\
&= \bar{\lambda}^k + \frac{\tau + \varepsilon}{s}\left(\sum_{i=1}^{p} A_i x_i^k - b \right) - \frac{\tau + \varepsilon}{s}\left(\sum_{i=1}^{p} A_i x_i^{k+1} - b \right) \\
&\quad - \frac{1}{s}\left[(\tau - \varepsilon) A_1 x_1^{k+1} + \varepsilon A_1 x_1^k + \tau \sum_{i=2}^{p} A_i x_i^k - b\tau + \tau A_1 x_1^k - \tau A_1 x_1^k \right] \\
&= \bar{\lambda}^k - \frac{\tau + \varepsilon}{s} \sum_{i=1}^{p} A_i(x_i^{k+1} - x_i^k) - \frac{1}{s}\left[(\tau - \varepsilon) A_1(x_1^{k+1} - x_1^k) + \tau \left(\sum_{i=1}^{p} A_i x_i^k - b \right) \right] \tag{2.16}
\end{aligned}
$$

Analogously, for $i = 2, 3, \cdots, p$, it follows from (2.5)-(2.6) and (2.12) that

$$x_i^{k+1} \in \mathcal{X}_i, \quad f_i(x_i) - f_i(x_i^{k+1}) + \langle x_i - x_i^{k+1}, R_{x_i} \rangle \geq 0, \quad \forall \, x_i \in \mathcal{X}_i, \tag{2.17}$$

where

$$
\begin{aligned}
R_{x_i} &= -\tau A_i^\mathsf{T} \lambda^{k+1} - \tau A_i^\mathsf{T}(\lambda^{k+1} - \lambda^k) + \left(\sigma_i + \tfrac{\tau^2 - 1}{s}\right) A_i^\mathsf{T} A_i(x_i^{k+1} - x_i^k) \\
&= \tau A_i^\mathsf{T} \lambda^k - 2\tau A_i^\mathsf{T} \left\{ \lambda^k - \tfrac{1}{s}\left[(\tau - \varepsilon)A_1 x_1^{k+1} + \varepsilon A_1 x_1^k + \tau \sum_{i=2}^{p} A_i x_i^k - b\tau \right] \right\} \\
&\quad + \left(\sigma_i + \tfrac{\tau^2 - 1}{s}\right) A_i^\mathsf{T} A_i(x_i^{k+1} - x_i^k) \\
&= -\tau A_i^\mathsf{T} \bar\lambda^{k+\frac{1}{2}} + \bar\sigma_i A_i^\mathsf{T} A_i(x_i^{k+1} - x_i^k),
\end{aligned}
$$

and

$$
\begin{aligned}
\bar\lambda^{k+\frac{1}{2}} &= \lambda^k - \frac{2}{s}\left[(\tau - \varepsilon)A_1 x_1^{k+1} + \varepsilon A_1 x_1^k + \tau \sum_{i=2}^{p} A_i x_i^k - b\tau\right] \\
&= \bar\lambda^k + \frac{\tau + \varepsilon}{s}\left(\sum_{i=1}^{p} A_i x_i^k - b\right) - \frac{2}{s}\left[(\tau - \varepsilon)A_1(x_1^{k+1} - x_1^k) + \tau\left(\sum_{i=1}^{p} A_i x_i^k - b\right)\right] \\
&= \bar\lambda^k + \frac{\tau + \varepsilon - 2\tau}{s}\left(\sum_{i=1}^{p} A_i x_i^k - b\right) - \frac{2(\tau - \varepsilon)}{s} A_1(x_1^{k+1} - x_1^k) \\
&= \bar\lambda^k - \frac{\tau - \varepsilon}{s}\left[2A_1(x_1^{k+1} - x_1^k) + \sum_{i=1}^{p} A_i x_i^k - b\right].
\end{aligned} \tag{2.18}
$$

Therefore, x_i^{k+1} is the solution of the following problem

$$
\begin{aligned}
x_i^{k+1} &= \arg\min_{x_i \in \mathcal{X}_i} \left\{ f_i(x_i) - \left\langle A_i x_i, \tau \bar\lambda^{k+\frac{1}{2}} \right\rangle + \frac{\bar\sigma_i}{2} \left\| A_i(x_i - x_i^k) \right\|^2 \right\} \\
&= \arg\min_{x_i \in \mathcal{X}_i} \left\{ f_i(x_i) + \frac{\bar\sigma_i}{2} \left\| A_i(x_i - x_i^k) - \frac{\tau}{\bar\sigma_i} \bar\lambda^{k+\frac{1}{2}} \right\|^2 \right\}, \quad i = 2, \cdots, p. \tag{2.19}
\end{aligned}
$$

Consequently, the algorithmic framework of GR-PPA is described in Algorithm 2.1, where we use $\widetilde{w}^k = (\widetilde{x}_1^k, \cdots, \widetilde{x}_p^k, \widetilde{\lambda}^k)$ to replace the original output of the x_i-subproblems in (2.15), (2.19) and Lagrange multipliers with given iterate $w^k = (x_1^k, \cdots, x_p^k, \bar\lambda^k)$, and we use $w^{k+1} = (x_1^{k+1}, \cdots, x_p^{k+1}, \bar\lambda^{k+1})$ to stand for the new iterate after adding a relaxation step.

Remark 2.2 *Noticing that the steps 4-9 in Algorithm 2.1 are actually the PPA updates whose proximal term is implicitly simplified into the quadratic term of each subproblem. Algorithm 2.1 is an extension of our method [1] for the two-block separable convex problem, but the domain of the parameters restricted into (2.7) is not a direct extension of the past. Besides, Algorithm 2.1 adopts a serial idea between x_1-subproblem and other subproblems, while the parallel idea is used among the x_i-subproblems $(i = 2, \cdots, p)$. From the relaxation step of Algorithm 2.1, we immediately have the following certain relationship*

$$w^{k+1} - w^k = \gamma(\widetilde{w}^k - w^k). \tag{2.20}$$

Algorithm 2.1 (GR-PPA for solving Problem (1.1))

---

1   Choose parameters $(\sigma_1, \cdots, \sigma_p, s) \in \mathcal{K}, \gamma \in (0, 2)$ and initialize $(x_1^0, \cdots, x_p^0, \lambda^0) \in \Omega$;

2    $r^0 = \sum\limits_{i=1}^{p} A_i x_i^0 - b; \ \bar{\lambda}^0 = \lambda^0 - \frac{\tau + \varepsilon}{s} r^0$ by (2.14);

3   **For** $k = 0, 1, \cdots$, if not converge, **do**

4        $\widetilde{x}_1^k = \arg \min\limits_{x_1 \in \mathcal{X}_1} \left\{ f_1(x_1) + \frac{\bar{\sigma}_1}{2} \left\| A_1(x_1 - x_1^k) - \frac{\tau}{\bar{\sigma}_1} \bar{\lambda}^k \right\|^2 \right\}$;

5        $r^k = \sum\limits_{i=1}^{p} A_i x_i^k - b; \ \Delta x_1^k = \widetilde{x}_1^k - x_1^k$;

6        $\bar{\lambda}^{k+\frac{1}{2}} = \bar{\lambda}^k - \frac{\tau - \varepsilon}{s} \left(2A_1 \Delta x_1^k + r^k \right)$;

7   Update the x_i-subproblem for $i = 2, \cdots, p$ by (2.19):

$$\begin{cases} \widetilde{x}_2^k = \arg \min\limits_{x_2 \in \mathcal{X}_2} \left\{ f_2(x_2) + \frac{\bar{\sigma}_2}{2} \left\| A_2(x_2 - x_2^k) - \frac{\tau}{\bar{\sigma}_2} \bar{\lambda}^{k+\frac{1}{2}} \right\|^2 \right\}; \\ \vdots \\ \widetilde{x}_p^k = \arg \min\limits_{x_p \in \mathcal{X}_p} \left\{ f_p(x_p) + \frac{\bar{\sigma}_p}{2} \left\| A_p(x_p - x_p^k) - \frac{\tau}{\bar{\sigma}_p} \bar{\lambda}^{k+\frac{1}{2}} \right\|^2 \right\}; \end{cases}$$

8        $\Delta x_i^k = \widetilde{x}_i^k - x_i^k, \ \forall \ i = 2, \cdots, p$;

9        $\widetilde{\lambda}^k = \bar{\lambda}^k - \frac{\tau + \varepsilon}{s} \sum\limits_{i=1}^{p} A_i \Delta x_i^k - \frac{1}{s} \left[(\tau - \varepsilon) A_1 \Delta x_1^k + r^k \tau \right]$ by (2.16);

10   Relaxation step:

$$\begin{pmatrix} x_1^{k+1} \\ \vdots \\ x_p^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^k \\ \vdots \\ x_p^k \end{pmatrix} + \gamma \begin{pmatrix} \Delta x_1^k \\ \vdots \\ \Delta x_p^k \end{pmatrix} \text{ and } \bar{\lambda}^{k+1} = \bar{\lambda}^k + \gamma(\widetilde{\lambda}^k - \bar{\lambda}^k).$$

2.2   Convergence analysis of GR-PPA

This subsection analyzes the global convergence and the ergodic convergence rate of Algorithm 2.1. First of all, we present an important lemma described as follows.

Lemma 2.2 *The sequence* $\{w^{k+1}\}$ *generated by Algorithm 2.1 satisfies*

$$\left\| w^{k+1} - w^* \right\|_G^2 \leq \left\| w^k - w^* \right\|_G^2 - \frac{2 - \gamma}{\gamma} \left\| w^k - w^{k+1} \right\|_G^2, \quad \forall \ w^* \in \Omega^*. \tag{2.21}$$

Proof By Algorithm 2.1 and (2.5), we have

$$\widetilde{w}^k \in \Omega, \quad \phi(u) - \phi(\widetilde{u}^k) + \left\langle w - \widetilde{w}^k, \mathcal{J}(\widetilde{w}^k) + G\left(\widetilde{w}^k - w^k\right) \right\rangle \geq 0, \quad \forall \ w \in \Omega, \tag{2.22}$$

which, by using (2.2) and (2.4) with setting $w = w^*$, leads to

$$\left\langle \widetilde{w}^k - w^*, G\left(w^k - \widetilde{w}^k\right) \right\rangle \geq 0. \tag{2.23}$$

Then, it follows from (2.20) and (2.23) that

$$
\begin{aligned}
\left\| w^k - w^* \right\|_G^2 - \left\| w^{k+1} - w^* \right\|_G^2 &= \left\| w^k - w^* \right\|_G^2 - \left\| w^k - w^* + w^{k+1} - w^k \right\|_G^2 \\
&= 2\gamma \left\langle w^k - w^*, G(w^k - \widetilde{w}^k) \right\rangle - \gamma^2 \left\| \widetilde{w}^k - w^k \right\|_G^2 \\
&= 2\gamma \left\langle w^k - \widetilde{w}^k + \widetilde{w}^k - w^*, G(w^k - \widetilde{w}^k) \right\rangle - \gamma^2 \left\| w^k - \widetilde{w}^k \right\|_G^2 \\
&= \gamma(2 - \gamma) \left\| w^k - \widetilde{w}^k \right\|_G^2 + 2\gamma \left\langle \widetilde{w}^k - w^*, G(w^k - \widetilde{w}^k) \right\rangle \\
&\geq \gamma(2 - \gamma) \left\| w^k - \widetilde{w}^k \right\|_G^2 = \frac{2 - \gamma}{\gamma} \left\| w^k - w^{k+1} \right\|_G^2,
\end{aligned}
$$

which immediately completes the proof. $\square$

Lemma 2.2 shows that the sequence $\{w^{k+1} - w^*\}$ is strictly contractive under the weighted G-norm. Moreover, the following global convergence theorem holds.

Theorem 2.1 *Let the parameters* $(\sigma_1, \cdots, \sigma_p, s) \in \mathcal{K}$ *and the sequence* $\{w^{k+1}\}$ *be generated by Algorithm 2.1. Then, under Assumption 1.1 there exists a point* $w^\infty \in \Omega^*$ *such that*

$$
\lim_{k \to \infty} w^{k+1} = w^\infty. \tag{2.24}
$$

Proof See the proof of Theorem 1 [1]. $\square$

In order to establish the convergence rate of Algorithm 2.1 in an ergodic sense, we first need to characterize the solution set of $VI(\phi, \mathcal{J}, \Omega)$, which had been given by e.g. [10] in the following:

Theorem 2.2 *The solution set of* $VI(\phi, \mathcal{J}, \Omega)$ *in (2.2) is convex and can be characterized as*

$$
\Omega^* = \bigcap_{w \in \Omega} \left\{ \widehat{w} \in \Omega \mid \phi(u) - \phi(\widehat{u}) + \langle w - \widehat{w}, \mathcal{J}(w) \rangle \geq 0 \right\}.
$$

Theorem 2.3 *For any* $(\sigma_1, \cdots, \sigma_p, s) \in \mathcal{K}$, *let*

$$
\boldsymbol{w}_t := \frac{1}{1+t} \sum_{k=0}^{t} \widetilde{w}^k \quad and \quad \boldsymbol{u}_t := \frac{1}{1+t} \sum_{k=0}^{t} \widetilde{u}^k,
$$

where $\{\widetilde{w}^k\}$ *is the iterative sequence of Algorithm 2.1. Then, under Assumption 1.1 we have*

$$
\phi(\boldsymbol{u}_t) - \phi(u) + \langle \boldsymbol{w}_t - w, \mathcal{J}(w) \rangle \leq \frac{1}{2\gamma(1+t)} \left\| w^0 - w \right\|_G^2, \ \forall \ w \in \Omega. \tag{2.25}
$$

Proof Combining (2.22) with (2.4), we get

$$
\phi(u) - \phi(\widetilde{u}^k) + \left\langle w - \widetilde{w}^k, \mathcal{J}(w) \right\rangle \geq \left\langle \widetilde{w}^k - w, G\left(\widetilde{w}^k - w^k \right) \right\rangle. \tag{2.26}
$$

Meanwhile, by (2.20) and the following identity

$$
2\langle a - b, G(c - d) \rangle = \|a - d\|_G^2 - \|a - c\|_G^2 + \|c - b\|_G^2 - \|d - b\|_G^2
$$

with substitutions $(\widetilde{w}^k, w, w^{k+1}, w^k) = (a, b, c, d)$, it follows that

$$
\begin{aligned}
\left\langle \widetilde{w}^k - w, G\left(\widetilde{w}^k - w^k \right) \right\rangle &= \frac{1}{\gamma} \left\langle \widetilde{w}^k - w, G\left(w^{k+1} - w^k \right) \right\rangle \\
&= \frac{1}{2\gamma} \left(\left\| \widetilde{w}^k - w^k \right\|_G^2 - \left\| \widetilde{w}^k - w^{k+1} \right\|_G^2 + \left\| w^{k+1} - w \right\|_G^2 - \left\| w^k - w \right\|_G^2 \right),
\end{aligned} \tag{2.27}
$$

where the first two terms

$$
\begin{aligned}
\left\| \widetilde{w}^k - w^k \right\|_G^2 - \left\| \widetilde{w}^k - w^{k+1} \right\|_G^2 &= \left\| \widetilde{w}^k - w^k \right\|_G^2 - \left\| w^k - \widetilde{w}^k + w^{k+1} - w^k \right\|_G^2 \\
&= \left\| \widetilde{w}^k - w^k \right\|_G^2 - \left\| w^k - \widetilde{w}^k + \gamma(\widetilde{w}^k - w^k) \right\|_G^2 \qquad (2.28) \\
&= \gamma(2 - \gamma) \left\| (\widetilde{w}^k - w^k) \right\|_G^2 \geq 0.
\end{aligned}
$$

Combining (2.26)-(2.28), we deduce

$$
\phi(u) - \phi(\widetilde{u}^k) + \langle w - \widetilde{w}^k, \mathcal{J}(w) \rangle + \frac{1}{2\gamma} \left\| w^k - w \right\|_G^2 \geq \frac{1}{2\gamma} \left\| w^{k+1} - w \right\|_G^2.
$$

Summing the above inequality over $k = 0, 1, \cdots, t$, we obtain

$$
(1+t)\phi(u) - \sum_{k=0}^{t} \phi(\widetilde{u}^k) + \left\langle (1+t)w - \sum_{k=0}^{t} \widetilde{w}^k, \mathcal{J}(w) \right\rangle + \frac{1}{2\gamma} \left\| w^0 - w \right\|_G^2 \geq 0,
$$

which by the definitions of $\mathbf{w}_t$ and $\mathbf{u}_t$ results in

$$
\frac{1}{1+t} \sum_{k=0}^{t} \phi(\widetilde{u}^k) - \phi(u) + \langle \mathbf{w}_t - w, \mathcal{J}(w) \rangle \leq \frac{1}{2\gamma(1+t)} \left\| w^0 - w \right\|_G^2. \qquad (2.29)
$$

Because of the convexity of the function $\phi(u)$ (since all f_i are assumed to be convex), the following inequality holds

$$
\phi(\mathbf{u}_t) \leq \frac{1}{1+t} \sum_{k=0}^{t} \phi(\widetilde{u}^k),
$$

which, by substituting it into (2.29), completes the whole proof. $\quad\square$

Remark 2.3 *Theorem 2.3 illustrates the worst-case $\mathcal{O}(1/t)$ convergence rate of Algorithm 2.1 in an ergodic sense. By the region $\gamma \in (0, 2)$ in Algorithm 2.1 and the inequality (2.25), one may choose a larger value γ approximating to 2 so that the value of right-hand of (2.25) is much smaller.*

3 Numerical experiments

In this section, we investigate the numerical performance of our proposed GR-PPA for solving a popular sparse matrix optimization problem. All experiments are simulated in MATLAB 7.14 (R2012a) on a lenovo-PC with Intel Core i5 processor (2.70GHz) and 4 GB memory.

3.1 Test problem

The Latent Variable Gaussian Graphical Model Selection (LVGGMS) problem [3, 16] arising in statistical learning is of the following form:

$$
\begin{aligned}
\min \quad & F(X, S, L) := \langle X, C \rangle - \log \det(X) + \nu \|S\|_1 + \mu tr(L) \\
\text{s.t.} \quad & X - S + L = \mathbf{0}, \ L \succeq \mathbf{0},
\end{aligned} \qquad (3.1)
$$

where $C \in \mathbb{R}^{n \times n}$ is a given covariance matrix obtained from the sample variables, ν and μ are given positive weighting factors, $tr(L)$ denotes the trace of the matrix L, $\|S\|_1 = \sum_{i,j=1}^{n} |S_{ij}|$ stands for the l_1-norm of the matrix S and S_{ij} means its ij-th entry.

Clearly, the LVGGMS problem (3.1) can be regarded as a special case of (1.1). And applying Algorithm 2.1 it is easy to write the three corresponding subproblems as the following

$$
\begin{cases}
\widetilde{X}^k = \arg \min\limits_{X \in \mathbb{R}^{n \times n}} \left\{ \langle X, C \rangle - \log \det(X) + \frac{\bar{\sigma}_1}{2} \left\| X - (X^k + \frac{\tau}{\bar{\sigma}_1} \bar{\lambda}^k) \right\|_F^2 \right\}, \\
\widetilde{S}^k = \arg \min\limits_{S \in \mathbb{R}^{n \times n}} \left\{ \nu \|S\|_1 + \frac{\bar{\sigma}_2}{2} \left\| S - (S^k - \frac{\tau}{\bar{\sigma}_2} \bar{\lambda}^{k+\frac{1}{2}}) \right\|_F^2 \right\}, \\
\widetilde{L}^k = \arg \min\limits_{L \succeq \mathbf{0}} \left\{ \mu tr(L) + \frac{\bar{\sigma}_3}{2} \left\| L - (L^k + \frac{\tau}{\bar{\sigma}_3} \bar{\lambda}^{k+\frac{1}{2}}) \right\|_F^2 \right\}.
\end{cases}
\tag{3.2}
$$

Observe that the above subproblems have closed formula solutions. According to the first-order optimality condition of the X-subproblem in (3.2), we derive

$$
\mathbf{0} = C - X^{-1} + \bar{\sigma}_1 \left(X - X^k - \frac{\tau}{\bar{\sigma}_1} \bar{\lambda}^k \right)
$$
$$
\iff \bar{\sigma}_1 X^2 + \left(C - \bar{\sigma}_1 X^k - \tau \bar{\lambda}^k \right) X - \mathbf{I} = \mathbf{0}.
$$

Then, by using the eigenvalue decomposition

$$
U \mathrm{Diag}(\rho) U^{\mathsf{T}} = C - \bar{\sigma}_1 X^k - \tau \bar{\lambda}^k,
$$

where $\mathrm{Diag}(\rho)$ is a diagonal matrix with diagonal entries $\rho_i (i = 1, \cdots, n)$, we get its explicit solution

$$
\widetilde{X}^k = U \mathrm{Diag}(\gamma) U^{\mathsf{T}},
\tag{3.3}
$$

in which $\mathrm{Diag}(\gamma)$ is the diagonal matrix with diagonal entries

$$
\gamma_i = \frac{-\rho_i + \sqrt{\rho_i^2 + 4\bar{\sigma}_1}}{2\bar{\sigma}_1}, \quad i = 1, 2, \cdots, n.
$$

Applying the soft shrinkage operator $\mathrm{Shrink}(\cdot, \cdot)$, see e.g.[18], the solution of the S-subproblem is

$$
\widetilde{S}^k = \mathrm{Shrink} \left(S^k - \frac{\tau}{\bar{\sigma}_2} \bar{\lambda}^{k+\frac{1}{2}}, \frac{\nu}{\bar{\sigma}_2} \right).
\tag{3.4}
$$

Besides, it is obvious that the L-subproblem in (3.2) is equivalent to

$$
L^{k+1} = \arg \min\limits_{L \succeq \mathbf{0}} \frac{\bar{\sigma}_3}{2} \left\| L - \widetilde{L} \right\|_F^2 = V \mathrm{Diag}(\max\{\widetilde{\rho}, \mathbf{0}\}) V^{\mathsf{T}},
\tag{3.5}
$$

where $\max\{\widetilde{\rho}, \mathbf{0}\}$ is taken component-wise and $V \mathrm{Diag}(\widetilde{\rho}) V^{\mathsf{T}}$ is the eigenvalue decomposition of the matrix

$$
\widetilde{L} = L^k + \frac{\tau \bar{\lambda}^{k+\frac{1}{2}} - \mu \mathbf{I}}{\bar{\sigma}_3}.
$$

3.2 Numerical results

In the following experiments, the parameters $(\nu, \mu) = (0.005, 0.05)$ in (3.1) come from [16] and the matrix C is randomly generated by the MATLAB codes of Boyd's homepage[1] with $n = 100$. The maximal number of iterations is set as 1000 and the following termination criterions are simultaneously used for all algorithms:

$$
\mathrm{IER(k)} \quad := \quad \max \left\{ \frac{\left\| X^k - X^{k-1} \right\|_F}{\|X^k\|_F}, \frac{\left\| S^k - S^{k-1} \right\|_F}{\|S^k\|_F}, \frac{\left\| L^k - L^{k-1} \right\|_F}{\|L^k\|_F} \right\} \leq \epsilon_1,
$$

$$
\mathrm{OER(k)} \quad := \quad \frac{|F(X^k, S^k, L^k) - F^*|}{|F^*|} \leq \epsilon_2,
$$

$$
\mathrm{CER(k)} \quad := \quad \frac{\left\| X^k - S^k + L^k \right\|_F}{\max\left\{1, \|X^k\|_F, \|S^k\|_F, \|L^k\|_F\right\}} \leq \epsilon_3,
$$

---

[1]http://web.stanford.edu/~boyd/papers/admm/covsel/covsel_example.html.

where $(\epsilon_1, \epsilon_2, \epsilon_3)$ are given tolerances, (X^k, S^k, L^k) are the k-th iterative values and F^* is the approximate optimal objective function value by running Algorithm 2.1 after 1000 iterations. The penalty parameter and the relaxation factor in all involved algorithms are set as $(\beta, \gamma) = (0.05, 1.8)$. We choose the feasible points $(X^0, S^0, L^0, \lambda^0) = (\mathbf{I}, \mathbf{4I}, \mathbf{3I}, \mathbf{0})$ as the initial starting iterates.

First of all, for the two-block classical *lasso* problem, as shown in [1] that Algorithm 2.1 could perform significantly better than ADMM and the customized relaxed PPA [6] when properly choosing the parameters, especially for solving large-size problems and high accurate solutions are required. Followed by the similar way of adjusting the parameters [1], we first would like to investigate the effects of the parameters $(\sigma_1, \sigma_2, \sigma_3, s)$ on Algorithm 2.1 for solving the three-block case (3.1). For this purpose, given (ε, τ) we randomly choose some values of one parameter and fix the remaining as known values, then it is clear from (2.7) that

$$s > 0, \ \sigma_1 > \frac{1 + 2\tau|\varepsilon|}{s}, \ \sigma_i > \frac{1 + \tau^2 + \tau|\varepsilon|}{s}, \ \forall i = 2, 3. \tag{3.6}$$

Here we can note that σ_1 and $\sigma_i (i = 2, 3)$ play the same role if taking $\varepsilon = \tau$, since in such case their regions are the same. Therefore, in the following we would take $\varepsilon = \tau = \frac{\sqrt{5}-1}{2}$ (the golden section ratio) to adjust the relative better value of e.g. σ_1 for an example. We also investigate the effect of the parameter s on Algorithm 2.1.

Under the tolerance $\epsilon_i = 10^{-8}$ $(i = 1, 2, 3)$, experimental results of adjusting the parameters σ_1 and s are respectively reported in Tables 1-2, where the parameter in Table 1 is restricted by $\sigma_1 > \frac{2+(\sqrt{5}-1)^2}{20} \approx 0.1764$ and the parameter in Table 2 is restricted by $s > \frac{2+(\sqrt{5}-1)^2}{0.178} \approx 9.9097$. The notations "IT", "CPU" denote the number of iterations and the CPU time in seconds, respectively. We can observe from Tables 1-2 that:

- For the parameters $(\sigma_1, \sigma_2, \sigma_3, s)$, the experimental results in each column of IER and CER are nearly the same when fixing any three parameters with one parameter changing;

- With the increase of the parameter σ_1 (s), both the iteration number and the CPU time tend to increase, which verifies the comments in the final part of [8]. That is, slow convergence would occur in terms of solving the proximal subproblem if the proximal parameter is set too large, which may significantly affect the whole computational efficiency of the algorithm;

- Both Tables 1-2 show that $(\sigma_1, \sigma_2, \sigma_3, s) = (0.178, 0.178, 0.178, 10)$ would be a relative reasonable choice for tackling the test problem (3.1) and we would use them as the default parameter setting for GR-PPA.

σ_1	IT	CPU	IER	OER	CER
0.178	160	4.82	4.30e-9	1.27e-12	6.97e-9
0.200	163	5.24	2.73e-9	1.69e-12	5.71e-9
0.350	168	5.44	2.43e-9	2.63e-12	7.75e-9
0.500	170	5.65	5.65e-9	1.15e-11	9.93e-9
0.600	182	5.75	3.06e-9	7.13e-11	8.58e-9
0.800	217	6.53	1.85e-9	1.39e-10	7.19e-9
0.900	236	7.36	1.09e-9	1.62e-10	7.21e-9
1.000	244	7.43	1.81e-9	1.93e-10	7.49e-9
3.000	318	10.01	9.72e-9	2.06e-10	8.07e-9
5.000	505	15.43	9.82e-9	2.85e-12	4.55e-9
8.000	775	23.98	9.85e-9	2.80e-12	4.56e-9
10.000	950	30.44	9.88e-9	2.40e-12	4.56e-9

Table 1: Results of GR-PPA with fixed parameters $(\sigma_2, \sigma_3, s) = (0.2, 0.2, 10)$.

s	IT	CPU	IER	OER	CER
10	141	4.20	6.05e-9	1.60e-12	7.80e-9
12	194	5.49	2.39e-9	2.41e-12	8.73e-9
15	257	7.34	2.42e-9	1.53e-12	9.92e-9
17	297	8.58	5.10e-9	2.45e-12	6.70e-9
20	363	10.42	6.55e-9	2.24e-12	4.47e-9
22	435	12.49	2.09e-9	1.97e-12	8.66e-9
25	487	14.03	4.62e-9	7.26e-13	5.19e-9
27	510	15.24	9.37e-9	1.74e-12	9.41e-9
30	585	16.93	6.94e-9	3.57e-12	7.10e-9
35	692	19.80	5.87e-9	5.75e-13	5.13e-9
40	782	22.22	6.54e-9	2.41e-12	9.37e-9
45	873	25.24	6.05e-9	2.61e-13	8.32e-9

Table 2: Results of GR-PPA with fixed parameters $(\sigma_1, \sigma_2, \sigma_3) = (0.178, 0.178, 0.178)$.

Secondly, we would like to use our GR-PPA (Algorithm 2.1) with the tuned parameters to compare with two state-of-the-art algorithms: the Proximal Jacobian decomposition of the ALM [8] ("PJALM") and the splitting algorithm of the ALM [9] ("HTY"). The default parameter $\mu = 2.01$ and $H = \beta\mathbf{I}$ are used for HTY [9], and the proximal parameter of PJALM [8] is fixed as 2 as suggested by the theory therein. Experimental results of these algorithms under different tolerances are reported in Table 3. Furthermore, with fixed tolerances $(\epsilon_1, \epsilon_2, \epsilon_3) = (10^{-12}, 10^{-14}, 10^{-12})$, convergence curves of the residuals IER and OER against the number of iterations by different algorithms with different starting iterates are respectively depicted in Figs. 1-2.

It is clear from Table 3 that under higher tolerances our GR-PPA performs significantly better than the others in terms of the number of iterations and the CPU time, although HTY could perform slightly better than GR-PPA under some lower tolerances. Besides, the comparative convergence curves depicted in Figs. 1-2 show that GR-PPA converges faster than the rest two algorithms for different starting feasible points, which illustrates that GR-PPA could performs well for any initial iterative values. Reported results of Table 1 and convergence curves in Figs. 1-2 demonstrate the effectiveness and robustness of the proposed algorithm.

4   Conclusion remarks

In this paper, we develop a relaxed parameterized PPA for solving a class of separable convex minimization problems. The global convergence and a worst-case sub-linear convergence rate of the algorithm are established. Numerical experiments on testing a sparse matrix minimization problem in statistical learning also verify that our proposed algorithm outperforms two popular algorithms when properly choosing the relaxation factor and the parameters in the proximal matrix.

Note that, in Sec.3, we only take a three-block problem of (1.1) for an example to investigate the performance of our algorithm, since its performance for the two-block *lasso* problem had been verified in [1] compared with the classical ADMM and the relaxed PPA [6]. For the problem with more than three variables, the parameters in GR-PPA need to adjust afresh via experiments. From the framework of PPA in (2.5), one may have other choices to construct a novel parameterized proximal matrix to develop the corresponding PPA only if it is symmetric positive definite. Besides, the proposed algorithm is applicable to the separable convex optimization problem where the coefficient matrices in the linear constraint have full column rank. Hence, whether there exists a PPA for the non-separable case with nonlinear constraints? In addition, we have tried our best to analyze the algorithmic convergence rate in the non-ergodic sense but without any results. These questions need further investigations in the future.
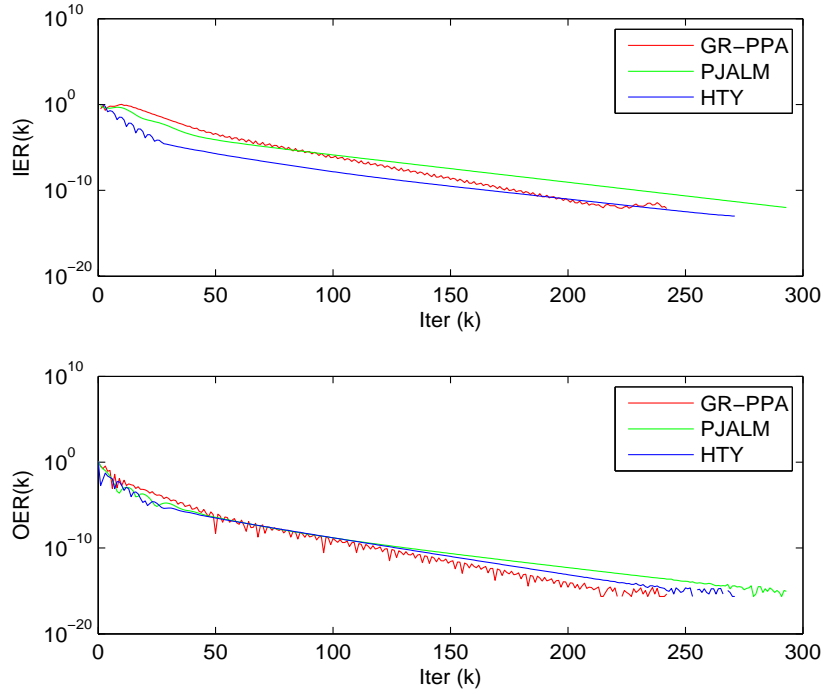
Fig. 1: Convergence curves of the residuals IER and OER by different algorithms with initial values $(X^0, S^0, L^0, \lambda^0) = (\mathbf{I}, 4\mathbf{I}, 3\mathbf{I}, \mathbf{0})$.
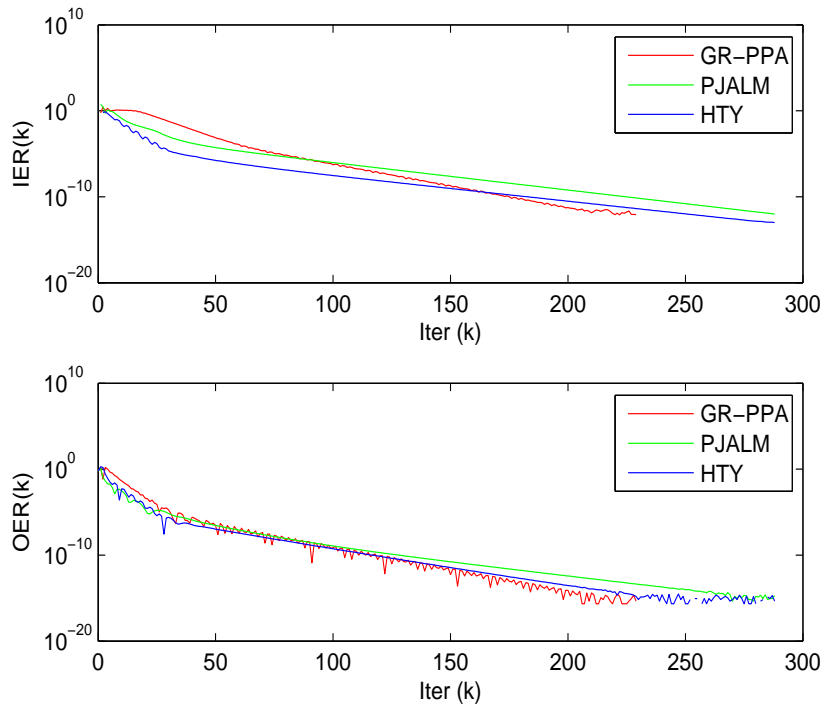

Fig. 2: Convergence curves of the residuals IER and OER by different algorithms with initial values $(X^0, S^0, L^0, \lambda^0) = (2\mathbf{I}, 4\mathbf{I}, 2\mathbf{I}, \mathbf{I}/2)$.

Acknowledgements

The authors wish to thank the Editor-in-Chief Prof. Choi-Hong Lai and the anonymous referees for providing their valuable suggestions, which have significantly improved the quality of our paper.

References

[1] J.C. Bai, H.C. Zhang, J.C. Li, A parameterized proximal point algorithm for separable convex optimization, Optim. Lett. (2017) doi: 10.1007/s11590-017-1195-9.

[2] X.J. Cai, G.Y. Gu, B.S. He, X.M. Yuan, A proximal point algorithm revisit on alternating direction method of multipliers, Sci. China Math. 56 (2013) 2179-2186.

[3] V. Chandrasekaran, P.A. Parrilo, A.S. Willsky, Latent variable graphical model selection via convex optimization, Ann. Stat. 40 (2012) 1935-1967.

[4] J. Eckstein, D.P. Bertsekas, On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators, Math. Program. 55 (1992) 293-318.

[5] J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso, Biostatistics, 9 (2008) 432-441.

[6] G.Y. Gu, B.S. He, X.M. Yuan, Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: a unified approach, Comput. Optim. Appl. 59 (2014) 135-161.

[7] B.S. He, X.M. Yuan, W.X. Zhang, A customized proximal point algorithm for convex minimization with linear constraints, Comput. Optim. Appl. 56 (2013) 559-572.

[8] B.S. He, H.K. Xu, X.M. Yuan, On the proximal Jacobian decomposition of ALM for multiple-block separable convex minimization problems and its relationship to ADMM, J. Sci. Comput. 66 (2016) 1204-1217.

[9] B.S. He, M. Tao, X.M. Yuan, A splitting method for separable convex programming, IMA J. Numer. Anal. 31 (2015) 394-426.

[10] B.S. He, F. Ma, X.M. Yuan, Convergence study on the symmetric version of ADMM with larger step sizes, SIAM J. Imaging Sci. 9 (2016) 1467-1501.

[11] A.P. Liao, X.B. Yang, J.X. Xie, Y. Lei, Analysis of convergence for the alternating direction method applied to joint sparse recovery, Appl. Math. Comput. 269 (2015) 548-557.

[12] Z.S. Liu, J.C. Li, G. Li, J.C. Bai, X.N. Liu, A new model for sparse and low-rank matrix decomposition, J. Appl. Anal. Comput. 7 (2017) 600-616.

[13] J.J. Moreau, Proximité et dualité dans un espace hilbertien, Bull. Soc. Math. Fr. 93 (1965) 273-299.

[14] B. Martinet, Regularisation, d'inéquations variationelles par approximations succesives, Rev. Fr. Inform. Rech. Oper. 4 (1970) 154-159.

[15] F. Ma, M.F. Ni, A class of customized proximal point algorithms for linearly constrained convex optimization, Comp. Appl. Math. (2016) doi:10.1007/s40314-016-0371-3.

[16] S.Q. Ma, Alternating proximal gradient method for convex minimization, J.Sci. Comput. 68 (2016) 546-572.

[17] R.T. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, Math. Oper. Res. 1 (1976) 97-116.

[18] M. Tao, X.M. Yuan, Recovering low-rank and sparse components of matrices from incomplete and noisy observations, SIAM J. Optim. 21 (2011) 57-81.

[19] B.X. Zhang, Z.B. Zhang, S. Wang, A simple primal-dual method for total variation image restoration, J. Vis. Commun. Image R. 38 (2016) 814-823.

Parameter	GR-PPA					PJALM					HTY				
	IT	CPU	IER	OER	CER	IT	CPU	IER	OER	CER	IT	CPU	IER	OER	CER
$\epsilon_1(\epsilon_2=10^{-10}/\epsilon_3=10^{-6})$															
10^{-4}	110	3.15	2.55e-7	2.89e-11	2.45e-7	132	3.55	1.28e-7	9.50e-11	5.59e-8	128	3.36	1.51e-9	9.26e-11	1.73e-8
10^{-7}	116	3.28	9.39e-8	7.73e-11	3.09e-7	136	3.67	9.53e-8	6.88e-11	4.17e-8	128	3.36	1.51e-9	9.26e-11	1.73e-8
10^{-12}	215	6.08	9.25e-13	4.44e-16	5.49e-12	293	7.63	9.96e-13	8.87e-16	4.35e-13	234	6.21	9.56e-13	2.44e-15	1.18e-11
$\epsilon_2(\epsilon_1=10^{-6}/\epsilon_3=10^{-6})$															
10^{-4}	105	2.72	4.17e-7	1.46e-10	6.58e-7	107	3.08	9.34e-7	9.54e-10	4.06e-7	79	2.03	1.03e-7	1.35e-8	9.20e-7
10^{-11}	124	3.54	5.11e-8	3.31e-12	5.86e-8	161	4.24	1.53e-8	9.76e-12	6.70e-9	151	4.01	2.83e-10	9.40e-12	3.41e-9
10^{-15}	214	6.15	1.36e-12	4.44e-16	1.71e-12	279	7.49	2.74e-12	2.22e-16	1.21e-12	243	6.48	5.20e-13	8.87e-16	6.42e-12
$\epsilon_3(\epsilon_1=10^{-6}/\epsilon_2=10^{-8})$															
10^{-8}	141	4.09	6.05e-9	1.60e-12	7.80e-9	156	5.12	2.20e-8	1.43e-11	9.65e-9	136	3.89	8.32e-10	4.17e-11	9.76e-9
10^{-11}	200	6.44	7.24e-12	2.44e-15	6.41e-12	251	7.19	2.12e-11	1.26e-14	9.33e-12	237	6.96	7.83e-13	3.10e-15	9.66e-12
10^{-12}	225	7.31	1.50e-12	6.66e-16	9.87e-13	282	8.65	2.22e-12	3.11e-15	6.70e-13	271	8.10	9.94e-14	2.22e-16	9.59e-13

Table 3: Comparative results of the problem (3.1) by different algorithms under different tolerances.