



On relaxation of some customized proximal point algorithms for convex minimization: from variational inequality perspective

Feng Ma¹

Received: 26 March 2017 / Published online: 2 April 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

The proximal point algorithm (PPA) is a fundamental method for convex programming. When applying the PPA to solve linearly constrained convex problems, we may prefer to choose an appropriate metric matrix to define the proximal regularization, so that the computational burden of the resulted PPA can be reduced, and sometimes even admit closed form or efficient solutions. This idea results in the so-called customized PPA (also known as preconditioned PPA), and it covers the linearized ALM, the primal-dual hybrid gradient algorithm, ADMM as special cases. Since each customized PPA owes special structures and has popular applications, it is interesting to ask whether we can design a simple relaxation strategy for these algorithms. In this paper we treat these customized PPA algorithms uniformly by a mixed variational inequality approach, and propose a new relaxation strategy for these customized PPA algorithms. Our idea is based on correcting the dual variables individually and does not rely on relaxing the primal variables. This is very different from previous works. From variational inequality perspective, we prove the global convergence and establish a worst-case convergence rate for these relaxed PPA algorithms. Finally, we demonstrate the performance improvements by some numerical results.

Keywords Convex minimization · Proximal point algorithm · Relaxation · Augmented Lagrangian method

1 Introduction

The problem concerned in this paper is the following convex minimization model

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

✉ Feng Ma
mafengnju@gmail.com

¹ High-Tech Institute of Xi'an, Xi'an 710025, Shaanxi, China

where $A \in \mathfrak{R}^{m \times n}$, $b \in \mathfrak{R}^m$, $\mathcal{X} \subset \mathfrak{R}^n$ is a closed convex set, $\theta : \mathfrak{R}^n \rightarrow \mathfrak{R}$ is a generic convex but not necessarily smooth function. Throughout, the solution set of (1.1) is assumed to be nonempty. Problems of this form arise in many applications such as compressed sensing, statistical estimation, machine learning, and image processing, see e.g., [2,8,17,33] for some examples.

We first reformulate (1.1) as a variational inequality form, so we can take a brief look at the algorithms to solve this problem. The Lagrangian function of the optimization problem (1.1) is

$$L(x, \lambda) = \theta(x) - \lambda^T (Ax - b),$$

which is defined on $\mathcal{X} \times \mathfrak{R}^m$. Let (x^*, λ^*) be a saddle point of the Lagrange function. Then we have

$$L_{\lambda \in \mathfrak{R}^m}(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L_{x \in \mathcal{X}}(x, \lambda^*).$$

By rearranging the above saddle point inequalities, we have the following optimal conditions:

$$\begin{cases} x^* \in \mathcal{X}, & \theta(x) - \theta(x^*) + (x - x^*)^T (-A^T \lambda^*) \geq 0, & \forall x \in \mathcal{X}, \\ \lambda^* \in \mathfrak{R}^m, & (\lambda - \lambda^*)^T (Ax^* - b) \geq 0, & \forall \lambda \in \mathfrak{R}^m. \end{cases} \quad (1.2)$$

More compactly, the inequalities (1.2) can be characterized as a monotone variational inequality (VI):

$$\text{VI}(\Omega, F, \theta) \quad w^* \in \Omega, \quad \theta(x) - \theta(x^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall w \in \Omega, \quad (1.3a)$$

where

$$w = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F(w) = \begin{pmatrix} -A^T \lambda \\ Ax - b \end{pmatrix} \quad \text{and} \quad \Omega = \mathcal{X} \times \mathfrak{R}^m. \quad (1.3b)$$

It is easily verified that $(w_1 - w_2)^T (F(w_1) - F(w_2)) \geq 0$ for any $w_1, w_2 \in \Omega$, thus the operator $F(\cdot)$ defined in (1.3b) is monotone. Since the solution set of (1.1) is assumed to be nonempty, the solution set of VI (1.3), denoted by Ω^* , is also nonempty.

As we have mentioned, our analysis will be conducted in the variational inequality context, so the VI reformulation (1.3) serves as the starting point for further analysis. The algorithms used to solve the model (1.1) amount to finding a solution point of VI (1.3). We now briefly revisit some relevant algorithms in VI form.

The classical proximal point algorithm (PPA) [29,31] is a popular method to solve the VI. Starting from an initial point w^0 , the PPA iterates as follows

$$\begin{aligned} w^{k+1} \in \Omega, \quad & \theta(x) - \theta(x^{k+1}) + (w - w^{k+1})^T \{F(w^{k+1}) \\ & + r(w^{k+1} - w^k)\} \geq 0, \quad \forall w \in \Omega, \end{aligned} \quad (1.4)$$

where $r > 0$ is a regularization parameter. The PPA is fundamental both theoretically and algorithmically in the area of the optimization. Indeed, the augmented Lagrangian method (ALM) [27,30] can be interpreted as applying the PPA to the dual of (1.1) [31]. We now verify the interpretation by revisiting the ALM (1.5) in the VI context. For (1.1), the ALM reads as

$$\begin{cases} x^{k+1} = \arg \min\{\theta(x) - (\lambda^k)^T(Ax - b) + \frac{\beta}{2}\|Ax - b\|^2 \mid x \in \mathcal{X}\}, & (1.5a) \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} - b), & (1.5b) \end{cases}$$

where λ is the Lagrange multiplier and $\beta > 0$ is a penalty parameter for the linear constraints. Note that the first-order optimality condition of the x -subproblem (1.5a) is

$$\theta(x) - \theta(x^{k+1}) + (x - x^{k+1})^T\{-A^T\lambda^k + \beta A^T(Ax^{k+1} - b)\} \geq 0, \forall x \in \mathcal{X}.$$

Substituting (1.5b) into the above equality yields

$$\theta(x) - \theta(x^{k+1}) + (x - x^{k+1})^T\{-A^T\lambda^{k+1}\} \geq 0, \forall x \in \mathcal{X}. \tag{1.6}$$

Next, (1.5b) can be written as

$$(\lambda - \lambda^{k+1})^T\{Ax^{k+1} - b + \frac{1}{\beta}(\lambda^{k+1} - \lambda^k)\} \geq 0, \forall \lambda \in \mathfrak{N}^m. \tag{1.7}$$

Combining (1.6) and (1.7), we have

$$\begin{aligned} &\theta(x) - \theta(x^{k+1}) + \begin{pmatrix} x - x^{k+1} \\ \lambda - \lambda^{k+1} \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\lambda^{k+1} \\ Ax^{k+1} - b \end{pmatrix} \right. \\ &\left. + \begin{pmatrix} 0 \\ \frac{1}{\beta}(\lambda^{k+1} - \lambda^k) \end{pmatrix} \right\} \geq 0, \quad \forall (x, \lambda) \in \Omega. \end{aligned} \tag{1.8}$$

By using the notations in (1.3), the compact form is

$$\begin{aligned} &w^{k+1} \in \Omega, \quad \theta(x) - \theta(x^{k+1}) + (w - w^{k+1})^T\{F(w^{k+1}) \\ &+ Q(w^{k+1} - w^k)\} \geq 0, \quad \forall w \in \Omega, \end{aligned} \tag{1.9a}$$

with

$$Q = \begin{pmatrix} 0 & \\ & \frac{1}{\beta}I_m \end{pmatrix}. \tag{1.9b}$$

Compared with the PPA scheme (1.4), it is easy to see that only the dual term is proximally regularized by the coefficient $\frac{1}{\beta}$. Thus, applying the PPA to the dual of (1.1), we get the ALM scheme (1.5).

ALM is a benchmark method that has been successfully used in many important areas (e.g., linear inverse problems in image processing [1,10]). At each iteration, the computational effort of ALM is dominated by evaluating the operator $(A^T A + \frac{1}{\beta} \partial \theta)^{-1}(\cdot)$ in \mathcal{X} . When $A^T A$ is not an identity matrix, or even large and dense, such evaluation usually does not have easy or closed form solutions.

On the other hand, we have encountered some concrete applications arising in sparse or low-rank optimization problems modeled by (1.1), in which the convex functions have some special structures to utilize. An important structure we often encountered in practice is that the proximal mapping related to the objective $\theta(\cdot)$ defined by

$$\text{Prox}_{\tau\theta}(a) := \operatorname{argmin} \left\{ \theta(x) + \frac{1}{2\tau} \|x - a\|^2 \right\}, \tag{1.10}$$

with parameter $\tau > 0$, can be computed easily. Here “easy” means a closed-form solution exists or efficient solvers are available. Examples of such structure exist in linear and quadratic programming, basis pursuit, nuclear norm minimization, and model fitting problems, etc. Let us briefly review two typical algorithms developed for employing this structure.

The first is the primal-dual hybrid gradient (PDHG) algorithm, which is used for total variation image restoration problems in [39], and subsequently improved by Chambolle and Pock in [5,6]. For solving (1.1), the PDHG iterates as

$$\begin{cases} x^{k+1} = \operatorname{argmin} \{ L(x, \lambda^k) + \frac{r}{2} \|x - x^k\|^2 \mid x \in \mathcal{X} \}, & (1.11a) \\ \bar{x}^k = x^{k+1} + (x^{k+1} - x^k), & (1.11b) \\ \lambda^{k+1} = \operatorname{argmax} \{ L(\bar{x}^k, \lambda) - \frac{s}{2} \|\lambda - \lambda^k\|^2 \}, & (1.11c) \end{cases}$$

where r and s are required to satisfy $r > 0, s > 0, rs > \|A^T A\|$.

Similar to the derivation of ALM, by using the notations in (1.3), the VI form of PDHG is

$$\begin{aligned} w^{k+1} \in \Omega, \quad & \theta(x) - \theta(x^{k+1}) + (w - w^{k+1})^T \{ F(w^{k+1}) \\ & + Q(w^{k+1} - w^k) \} \geq 0, \quad \forall w \in \Omega, \end{aligned} \tag{1.12a}$$

where

$$Q = \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix}. \tag{1.12b}$$

Since $rs > \|A^T A\|$, Q is symmetric positive definite, and can be used to define Q -norm by $\|w\|_Q := \sqrt{w^T Q w}$. With the VI characterization (1.12), we now can easily derive the contraction property of PDHG. Here we briefly discuss below.

Setting $w = w^*$ in (1.12), we have

$$(w^{k+1} - w^*)^T Q(w^k - w^{k+1}) \geq \theta(x^{k+1}) - \theta(x^*) + (w^{k+1} - w^*)^T F(w^{k+1}).$$

Recall $F(\cdot)$ is monotone (see (1.3b)), we have

$$\begin{aligned} \theta(x^{k+1}) - \theta(x^*) + (w^{k+1} - w^*)^T F(w^{k+1}) &\geq \theta(x^{k+1}) - \theta(x^*) \\ &+ (w^{k+1} - w^*)^T F(w^*) \geq 0. \end{aligned}$$

Then, we obtain

$$(w^{k+1} - w^*)^T Q(w^k - w^{k+1}) \geq 0.$$

Using the above inequality, we get

$$\begin{aligned} \|w^k - w^*\|_Q^2 &= \|(w^{k+1} - w^*) + (w^k - w^{k+1})\|_Q^2 \\ &\geq \|w^{k+1} - w^*\|_Q^2 + \|w^k - w^{k+1}\|_Q^2, \end{aligned}$$

and thus the sequence generated by PDHG (1.12) obeys

$$\|w^{k+1} - w^*\|_Q^2 \leq \|w^k - w^*\|_Q^2 - \|w^k - w^{k+1}\|_Q^2. \tag{1.13}$$

The above inequality implies that the sequence $\{w^k\}$ is strictly contractive to the solution set in Q -norm. Recall that the proximal parameter in the original PPA is a scalar r , means that all the coordinates of w are proximally regularized with equivalent weights. While for the VI form of PDHG (1.12), the proximal parameter is a positive definite matrix Q , and the coordinates of w are proximally regularized with different weights. Using this judicious choice of Q , the resulting subproblems are simple under the assumption (1.10) and the convergence is also guaranteed by noting that the distance to the solution point decreases at each step (see (1.13)). Further, we observe that the algorithm (1.12) owes a similar structure with the PPA (1.12). In this sense, it is essentially a customized application of PPA. The VI revisit (1.12) and PPA interpretation of PDHG is first shown in [24], and used in [7] to simplify the convergence analysis.

Since the PDHG (1.11) takes an algorithmic structure analogous to that of the generic PPA, the existing relaxation strategies for the PPA in the literature can be naturally used for these customized PPA algorithms. One popular relaxation scheme is to combine a computationally trivial relaxation step with the PPA [19]. Specially, let $\tilde{w}^k = (\tilde{x}^k, \tilde{\lambda}^k)$ be the output of the customized PPA. The new iterate w^{k+1} generated by the relaxation step in [19,20] is given by

VI framework of relaxed PPA for (1.3)

1. PPA step: generate \tilde{w}^k via solving

$$\begin{aligned} \tilde{w}^k \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) &\geq (w - \tilde{w}^k)^T Q(w^k - \tilde{w}^k), \\ \forall w \in \Omega, \end{aligned} \quad (1.14a)$$

where Q is a symmetric positive definite matrix.

2. Relaxation step: generate the new iterate w^{k+1} via

$$w^{k+1} = w^k - \gamma(w^k - \tilde{w}^k), \quad (1.14b)$$

where the relaxation factor $\gamma \in (0, 2)$.

As demonstrated in many cases, the relaxation step (1.14b) with $\gamma \in (1, 2)$ can speed up the convergence of the customized PPA, see, e.g. [23,24]. However, this relaxation strategy has some issues that would be overlooked: the point w^{k+1} is a combination of w^k and \tilde{w}^k . It may not be in the set Ω or owes the structure that \tilde{w}^k has. To address these limitations, one can first use the relaxation term (1.14b) in the iteration loops; when the specified stop criterion is achieved, one should perform an extra iteration without the relaxation step.

On the other hand, observe that each customized PPA owes special structure and has popular applications, it is interesting to ask whether there exists another simple relaxation strategy for these algorithms based on their structures and can also have good improvements. In this paper we treat these customized PPA algorithms uniformly by a mixed variational inequality approach, and propose a new relaxation strategy for these customized PPA algorithms. Now, we specify our relaxed customized PPA as follows

The new VI framework of relaxed PPA for (1.3)

1. PPA step: generate \tilde{w}^k via solving

$$\begin{aligned} \tilde{w}^k \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) &\geq (w - \tilde{w}^k)^T Q(w^k - \tilde{w}^k), \\ \forall w \in \Omega, \end{aligned} \quad (1.15a)$$

where Q is a symmetric positive definite matrix.

2. Relaxation step: set $x^{k+1} = \tilde{x}^k$, and generate the dual iterate λ^{k+1} via

$$\lambda^{k+1} = \lambda^k - \mathcal{A}(\gamma)(w^k - \tilde{w}^k), \quad (1.15b)$$

where $\gamma \in (0, 2)$ and $\mathcal{A}(\gamma)$ is a linear operator regarding of γ .

The paper is organized as follows. In Sect. 2, we introduce our relaxation steps for some customized PPA algorithms. In Sect. 3, we discuss the relaxation strategy for the ADMM. We then analyze the convergence of these methods in Sect. 4, and further establish their convergence rate in both the non-ergodic and ergodic sense in Sect. 5. After that, we present some numerical results to demonstrate the improved

performance of the relaxation strategy in Sect. 6. Finally, some conclusions are made in Sect. 7.

2 Relaxed customized PPA

In this section, we show how to relax the customized PPA via correcting the dual variable. To make our illustration more clear, the output variable w^{k+1} of the customized PPA is relabeled as \tilde{w}^k below.

Note that the relaxation falls into the category of prediction-correction methods. We use the prototype algorithm framework proposed in [22] to unify our analysis. This algorithm framework is also used in [26] for analyzing the PDHG.

Prediction-correction method for the VI problem (1.3):

[Prediction Step.] With given w^k , find a vector $\tilde{w}^k \in \Omega$ and a matrix Q satisfying

$$\begin{aligned} \tilde{w}^k \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) &\geq (w - \tilde{w}^k)^T Q(w^k - \tilde{w}^k), \\ \forall w \in \Omega, \end{aligned} \tag{2.1a}$$

where the matrix Q has the property: $Q^T + Q$ is positive definite.

[Correction Step.] Determine a nonsingular matrix M ; and generate the new iterate w^{k+1} via

$$w^{k+1} = w^k - M(w^k - \tilde{w}^k). \tag{2.1b}$$

Then, the convergence of the prototype algorithm (2.1) can be guaranteed if the following conditions are fulfilled.

Convergence conditions:

For the matrices Q and M in (2.1), the matrices

$$H := QM^{-1} \succ 0 \tag{2.2a}$$

and

$$G := Q^T + Q - M^T H M \succ 0. \tag{2.2b}$$

In the following, we show that each PPA with the correction step falls into the prototype algorithm (2.1). Then we demonstrate the conditions specified in (2.2) are fulfilled.

2.1 Relaxed PDHG

Recall that the PDHG in variational form is (1.15a) with metric

$$Q = \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix}. \tag{2.3}$$

Then we judiciously correct the dual variable via

$$\lambda^{k+1} = \lambda^k - \frac{\gamma - 1}{s} A(x^k - \tilde{x}^k) - \gamma(\lambda^k - \tilde{\lambda}^k), \tag{2.4}$$

where the relaxation factor $\gamma \in (0, 2)$. Since we always have $x^{k+1} = \tilde{x}^k$, using the notations in (1.3), we get the correction step as

$$w^{k+1} = w^k - M(w^k - \tilde{w}^k), \tag{2.5a}$$

where

$$M = \begin{pmatrix} I_n & 0 \\ \frac{\gamma-1}{s} A & \gamma I_m \end{pmatrix}. \tag{2.5b}$$

Now, we check the positive definiteness of H and G to verify whether the conditions in (2.2) are fulfilled. First, we need to deduce the expression of H and G . Recall M (see (2.5b)), we have

$$M^{-1} = \begin{pmatrix} I_n & 0 \\ -\frac{\gamma-1}{s\gamma} A & \frac{1}{\gamma} I_m \end{pmatrix}.$$

Then

$$\begin{aligned} H &= QM^{-1} = \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix} \begin{pmatrix} I_n & 0 \\ -\frac{\gamma-1}{s\gamma} A & \frac{1}{\gamma} I_m \end{pmatrix} \\ &= \begin{pmatrix} rI_n - \frac{\gamma-1}{s\gamma} A^T A & \frac{1}{\gamma} A^T \\ \frac{1}{\gamma} A & \frac{s}{\gamma} I_m \end{pmatrix}. \end{aligned} \tag{2.6}$$

According to the Schur complement, we have

$$H = \begin{pmatrix} I & \frac{1}{s} A^T \\ 0 & I \end{pmatrix} \begin{pmatrix} rI - \frac{1}{s} A^T A & 0 \\ 0 & \frac{s}{\gamma} I \end{pmatrix} \begin{pmatrix} I & 0 \\ \frac{1}{s} A & I \end{pmatrix}.$$

Since $rs > \|A^T A\|$, H is positive definite.

The matrix G is symmetric by $Q^T = Q$, we next deduce the expression of G .

$$\begin{aligned} G &= Q^T + Q - M^T H M = 2Q - M^T Q \\ &= 2 \cdot \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix} - \begin{pmatrix} I_n & \frac{\gamma-1}{s} A^T \\ 0 & \gamma I_m \end{pmatrix} \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix} \\ &= \begin{pmatrix} 2rI_n & 2A^T \\ 2A & 2sI_m \end{pmatrix} - \begin{pmatrix} rI_n + \frac{\gamma-1}{s} A^T A & \gamma A^T \\ \gamma A & \gamma s I_m \end{pmatrix} \\ &= \begin{pmatrix} rI_n - \frac{\gamma-1}{s} A^T A & (2-\gamma)A^T \\ (2-\gamma)A & (2-\gamma)sI_m \end{pmatrix}. \end{aligned} \tag{2.7}$$

According to the Schur complement,

$$G = \begin{pmatrix} I & \frac{1}{s}A^T \\ 0 & I \end{pmatrix} \begin{pmatrix} rI - \frac{1}{s}A^T A & 0 \\ 0 & (2 - \gamma)sI \end{pmatrix} \begin{pmatrix} I & 0 \\ \frac{1}{s}A & I \end{pmatrix}. \tag{2.8}$$

Since $rI > \frac{1}{s}A^T A$ and $0 < \gamma < 2$, G is positive definite. The relaxed PDHG scheme is a special case of the prototype algorithm (2.2).

The relaxation step (2.4) looks somehow complicated. However, combining PDHG and (2.4), the update steps of the relaxed PDHG can be simple. By straightforward manipulations, the PDHG can be rewritten as a prediction step, which gives

$$\begin{cases} \tilde{x}^k = \operatorname{argmin}\{\theta(x) + \frac{r}{2}\|x - x^k - \frac{1}{r}A^T \lambda^k\|^2 \mid x \in \mathcal{X}\}, & (2.9a) \\ \tilde{\lambda}^k = \lambda^k - \frac{1}{s}(A(2\tilde{x}^k - x^k) - b). & (2.9b) \end{cases}$$

Together with (2.4) and noting $x^{k+1} = \tilde{x}^k$, we obtain:

Relaxed PDHG for (1.1)

$$\begin{cases} x^{k+1} = \operatorname{argmin}\{\theta(x) + \frac{r}{2}\|x - x^k - \frac{1}{r}A^T \lambda^k\|^2 \mid x \in \mathcal{X}\}, & (2.10a) \\ \lambda^{k+1} = \lambda^k - \frac{1}{s}\{A((1 + \gamma)x^{k+1} - x^k) - \gamma b\}, & (2.10b) \end{cases}$$

where $r > 0, s > 0, rs > \|A^T A\|$, and $\gamma \in (0, 2)$.

2.2 Relaxed linearized ALM

In this section, we show that the linearized ALM (L-ALM) [35,38] is also a special case of customized PPA, and it can be relaxed by (1.15b). The main idea of L-ALM is to linearize the augmented Lagrangian function (1.5a) in the ALM via adding a quadratic term $\frac{1}{2}\|x - x^k\|_{(rI - \beta A^T A)}^2$. Ignoring some constant terms in the x -subproblem, the iterate scheme of L-ALM is

$$\begin{cases} x^{k+1} = \operatorname{argmin} \left\{ \theta(x) + \frac{r}{2}\|x - x^k - \frac{1}{r}A^T(\lambda^k - \beta(Ax^k - b))\|^2 \mid x \in \mathcal{X} \right\}, & (2.11a) \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} - b). & (2.11b) \end{cases}$$

Note that the quadratic term should be positive, we have $r > \beta\|A^T A\|$. Now, we revisit L-ALM (2.11) in VI form.

The first-order optimality condition of the x -subproblem of (2.11a) is

$$\theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T \{-A^T \lambda^k + \beta A^T (A\tilde{x}^k - b) + (rI - \beta A^T A)(\tilde{x}^k - x^k)\} \geq 0, \forall x \in \mathcal{X},$$

and it can be written as

$$\theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T \{-A^T \tilde{\lambda}^k + (rI - \beta A^T A)(\tilde{x}^k - x^k)\} \geq 0, \forall x \in \mathcal{X}. \tag{2.12}$$

Next, we rewrite $\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k - b)$ (see (2.11b)) as

$$(\lambda - \tilde{\lambda}^k)^T \left\{ A\tilde{x}^k - b + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) \right\} \geq 0, \forall \lambda \in \mathfrak{N}^m. \tag{2.13}$$

Combining (2.12) and (2.13), we have

$$\forall w \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T \left\{ \begin{pmatrix} -A^T \tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} + \begin{pmatrix} (rI_n - \beta A^T A)(\tilde{x}^k - x^k) \\ \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \geq 0. \tag{2.14}$$

Recall the notations in (1.3), the L-ALM (2.11) can thus be interpreted as a customized PPA (1.15a) with Q given by

$$Q = \begin{pmatrix} rI_n - \beta A^T A & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix}. \tag{2.15}$$

To accelerate the L-ALM, we suggest to relax the dual variable by

$$\lambda^{k+1} = \lambda^k - \gamma(\lambda^k - \tilde{\lambda}^k), \tag{2.16}$$

where the relaxation factor $\gamma \in (0, 2)$. Note that x^{k+1} always amounts to \tilde{x}^k , we have

$$w^{k+1} = w^k - M(w^k - \tilde{w}^k), \tag{2.17a}$$

where

$$M = \begin{pmatrix} I_n & 0 \\ 0 & \gamma I_m \end{pmatrix}. \tag{2.17b}$$

Next, we show that the condition (2.2) is fulfilled by the relaxed L-ALM scheme. We have

$$\begin{aligned} H &= QM^{-1} = \begin{pmatrix} rI_n - \beta A^T A & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} I_n & 0 \\ 0 & \frac{1}{\gamma} I_m \end{pmatrix} \\ &= \begin{pmatrix} rI_n - \beta A^T A & 0 \\ 0 & \frac{1}{\gamma\beta} I_m \end{pmatrix}, \end{aligned} \tag{2.18}$$

and

$$\begin{aligned}
 G &= Q^T + Q - M^T H M = 2Q - M^T Q \\
 &= \begin{pmatrix} 2(rI_n - \beta A^T A) & 0 \\ 0 & \frac{2}{\beta} I_m \end{pmatrix} - \begin{pmatrix} I_n & 0 \\ 0 & \gamma I_m \end{pmatrix} \begin{pmatrix} rI_n - \beta A^T A & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \\
 &= \begin{pmatrix} 2(rI_n - \beta A^T A) & 0 \\ 0 & \frac{2}{\beta} I_m \end{pmatrix} - \begin{pmatrix} rI_n - \beta A^T A & 0 \\ 0 & \frac{\gamma}{\beta} I_m \end{pmatrix} \tag{2.19} \\
 &= \begin{pmatrix} rI_n - \beta A^T A & 0 \\ 0 & \frac{2-\gamma}{\beta} I_m \end{pmatrix}.
 \end{aligned}$$

Under the condition that $r > 0, \beta > 0, r > \beta \|A^T A\|$ and the relaxation factor $\gamma \in (0, 2)$, the matrix H and G are both symmetric and positive definite. Thus, the relaxed L-ALM scheme is a special case of the prototype algorithm (2.2) and the conditions in (2.2) are fulfilled. We refer to [32] for the relaxed L-ALM with (1.14b).

Now, the relaxed L-ALM scheme is:

Relaxed L-ALM for (1.1)

$$\begin{cases} x^{k+1} = \operatorname{argmin}\{\theta(x) + \frac{r}{2}\|x - x^k - \frac{1}{r}A^T(\lambda^k - \beta(Ax^k - b))\|^2 \mid x \in \mathcal{X}\}, \\ \lambda^{k+1} = \lambda^k - \gamma\beta(Ax^{k+1} - b), \end{cases} \tag{2.20a}$$

$$\tag{2.20b}$$

where $r > 0, \beta > 0, r > \beta \|A^T A\|$, and $\gamma \in (0, 2)$.

3 Relaxed ADMM for separable case

Motivated by recent popular applications, we also discuss a separable case of (1.1), in which the objective function $\theta(x)$ can be decomposed as a sum of two individual convex functions without coupled variables:

$$\min\{\theta_1(x_1) + \theta_2(x_2) \mid A_1x_1 + A_2x_2 = b, x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2\}, \tag{3.1}$$

where $\theta_1 : \mathfrak{N}^{n_1} \rightarrow \mathfrak{R}, \theta_2 : \mathfrak{N}^{n_2} \rightarrow \mathfrak{R}$ are both closed convex (not necessarily smooth) functions; $A_1 \in \mathfrak{R}^{m \times n_1}, A_2 \in \mathfrak{R}^{m \times n_2}; \mathcal{X}_1 \subset \mathfrak{N}^{n_1}, \mathcal{X}_2 \subset \mathfrak{N}^{n_2}$ are closed convex sets.

The optimality condition of the problem (3.1) can be analogously written as the following variational inequality:

$$\text{VI}(\Omega, F, \theta) \quad w^* \in \Omega, \quad \theta(x) - \theta(x^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall w \in \Omega, \tag{3.2a}$$

where

$$\begin{aligned}
 x &= \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad w = \begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix}, \quad F(w) = \begin{pmatrix} -A_1^T \lambda \\ -A_2^T \lambda \\ A_1 x_1 + A_2 x_2 - b \end{pmatrix}, \\
 \theta(x) &= \theta_1(x_1) + \theta_2(x_2), \quad \text{and} \quad \Omega = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathfrak{R}^m.
 \end{aligned}
 \tag{3.2b}$$

Note that the mapping $F(w)$ defined in (3.2b) also satisfies $(w_1 - w_2)^T (F(w_1) - F(w_2)) \geq 0$ for any $w_1, w_2 \in \Omega$, thus it is monotone.

The classical alternating directions method of multipliers (ADMM) originates in [9,16] is suitable for solving (3.1). Starting with an initial iterate $(x_2^0, \lambda^0) \in \mathcal{X}_2 \times \mathfrak{R}^m$, the ADMM iterates via the scheme

$$\begin{cases}
 x_1^{k+1} = \arg \min \left\{ \theta_1(x_1) - (\lambda^k)^T A_1 x_1 + \frac{\beta}{2} \|A_1 x_1 + A_2 x_2^k - b\|^2 \mid x_1 \in \mathcal{X}_1 \right\}, & (3.3a)
 \end{cases}$$

$$\begin{cases}
 x_2^{k+1} = \arg \min \left\{ \theta_2(x_2) - (\lambda^k)^T A_2 x_2 + \frac{\beta}{2} \|A_1 x_1^{k+1} + A_2 x_2 - b\|^2 \mid x_2 \in \mathcal{X}_2 \right\}, & (3.3b)
 \end{cases}$$

$$\lambda^{k+1} = \lambda^k - \beta(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b). \tag{3.3c}$$

In [13], it was shown that ADMM is actually equivalent to the Douglas-Rachford splitting method (DRSM) applied to the dual of (3.1). We also refer to [21] for the primal application of DRSM. In this section, to make our analysis more general, we consider the proximal ADMM where each subproblem is regularized with an additional quadratic proximal term. Note that the original ADMM produces the new iterate in the forward order $x_1 \rightarrow x_2 \rightarrow \lambda$. In cyclical sense, it can also be written as the $x_1 \rightarrow \lambda \rightarrow x_2$ order, which gives

$$\begin{cases}
 x_1^{k+1} = \arg \min \left\{ \theta_1(x_1) - (\lambda^k)^T A_1 x_1 + \frac{\beta}{2} \|A_1 x_1 + A_2 x_2^k - b\|^2 + \|x_1 - x_1^k\|_{D_1}^2 \mid x_1 \in \mathcal{X}_1 \right\}, & (3.4a)
 \end{cases}$$

$$\lambda^{k+1} = \lambda^k - \beta(A_1 x_1^{k+1} + A_2 x_2^k - b), \tag{3.4b}$$

$$\begin{cases}
 x_2^{k+1} = \arg \min \left\{ \theta_2(x_2) - (\lambda^{k+1})^T A_2 x_2 + \frac{\beta}{2} \|A_1 x_1^{k+1} + A_2 x_2 - b\|^2 + \|x_2 - x_2^k\|_{D_2}^2 \mid x_2 \in \mathcal{X}_2 \right\}, & (3.4c)
 \end{cases}$$

where D_1 and D_2 are two positive definite matrices. Obviously, the proximal ADMM contains ADMM as a special case. When $D_1 = rI - \beta A_1^T A_1$ ($r > \beta \|A_1^T A_1\|$) or $D_2 = sI - \beta A_2^T A_2$ ($s > \beta \|A_2^T A_2\|$), the so-called split inexact Uzawa method

[37,38] is recovered. In this case, the resulting subproblem is in form of (1.10) (when $\mathcal{X}_i = \mathfrak{N}^{n_i}, i = 1, 2$), which is simple under the assumption (1.10). We refer the reader to see [11] for more different choices of D_1 and D_2 .

In our relaxation, the output of (3.4) serves as a predictor, we here relabel it as $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \tilde{\lambda}^k)$ as before. Now, we first follow [4] to revisit the proximal ADMM (3.4) in VI form.

Lemma 3.1 *For given $w^k = (x_1^k, x_2^k, \lambda^k)$, let \tilde{w}^k be generated by (3.4). Then, we have*

$$\begin{aligned} &\tilde{w}^k \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \\ &\geq (w - \tilde{w}^k)^T Q(w^k - \tilde{w}^k), \quad \forall w \in \Omega, \end{aligned} \tag{3.5a}$$

where

$$Q = \begin{pmatrix} D_1 & 0 & 0 \\ 0 & D_2 + \beta A_2^T A_2 & -A_2^T \\ 0 & -A_2 & \frac{1}{\beta} I_m \end{pmatrix}. \tag{3.5b}$$

Proof The optimality condition of the x_1 -subproblem in (3.4) is

$$\begin{aligned} &\tilde{x}_1^k \in \mathcal{X}_1, \quad \theta_1(x_1) - \theta_1(\tilde{x}_1^k) + (x_1 - \tilde{x}_1^k)^T \{-A_1^T \lambda^k + \beta A_1^T (A_1 \tilde{x}_1^k \\ &\quad + A_2 x_2^k - b) + D_1(\tilde{x}_1^k - x_1^k)\} \geq 0, \quad \forall x_1 \in \mathcal{X}_1, \end{aligned}$$

and it can be written as (using $\tilde{\lambda}^k$ generated by (3.4b))

$$\begin{aligned} &\tilde{x}_1^k \in \mathcal{X}_1, \quad \theta_1(x_1) - \theta_1(\tilde{x}_1^k) + (x_1 - \tilde{x}_1^k)^T (-A_1^T \tilde{\lambda}^k \\ &\quad + D_1(\tilde{x}_1^k - x_1^k)) \geq 0, \quad \forall x_1 \in \mathcal{X}_1. \end{aligned} \tag{3.6a}$$

The optimality condition of the x_2 -subproblem of (3.4) is

$$\begin{aligned} &\tilde{x}_2^k \in \mathcal{X}_2, \quad \theta_2(x_2) - \theta_2(\tilde{x}_2^k) + (x_2 - \tilde{x}_2^k)^T \{-A_2^T \tilde{\lambda}^k + \beta A_2^T (A_1 \tilde{x}_1^k \\ &\quad + A_2 \tilde{x}_2^k - b) + D_2(\tilde{x}_2^k - x_2^k)\} \geq 0, \quad \forall x_2 \in \mathcal{X}_2. \end{aligned}$$

Now, we treat the $\{\cdot\}$ term in the last inequality. Using $\beta(A_1 \tilde{x}_1^k + A_2 \tilde{x}_2^k - b) = -(\tilde{\lambda}^k - \lambda^k)$ (see (3.4b)), we obtain

$$\begin{aligned} &-A_2^T \tilde{\lambda}^k + \beta A_2^T (A_1 \tilde{x}_1^k + A_2 \tilde{x}_2^k - b) + D_2(\tilde{x}_2^k - x_2^k) \\ &= -A_2^T \tilde{\lambda}^k + \beta A_2^T A_2(\tilde{x}_2^k - x_2^k) - A_2^T(\tilde{\lambda}^k - \lambda^k) + D_2(\tilde{x}_2^k - x_2^k) \\ &= -A_2^T \tilde{\lambda}^k + (D_2 + \beta A_2^T A_2)(\tilde{x}_2^k - x_2^k) - A_2^T(\tilde{\lambda}^k - \lambda^k). \end{aligned}$$

Then the optimality condition of the x_2 -subproblem can be written as

$$\begin{aligned} &\tilde{x}_2^k \in \mathcal{X}_2, \quad \theta_2(x_2) - \theta_2(\tilde{x}_2^k) + (x_2 - \tilde{x}_2^k)^T \{-A_2^T \tilde{\lambda}^k \\ &\quad + (D_2 + \beta A_2^T A_2)(\tilde{x}_2^k - x_2^k) - A_2^T(\tilde{\lambda}^k - \lambda^k)\} \geq 0, \quad \forall x_2 \in \mathcal{X}_2. \end{aligned} \tag{3.6b}$$

From (3.4b), we have

$$(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k - b) - A_2(\tilde{x}_2^k - x_2^k) + (1/\beta)(\tilde{\lambda}^k - \lambda^k) = 0,$$

and it can be written as

$$\begin{aligned} \tilde{\lambda}^k \in \mathfrak{R}^m, \quad (\lambda - \tilde{\lambda}^k)^T \{ & (A_1\tilde{x}_1^k + A_2\tilde{x}_2^k - b) - A_2(\tilde{x}_2^k - x_2^k) \\ & + (1/\beta)(\tilde{\lambda}^k - \lambda^k) \} \geq 0, \quad \forall \lambda \in \mathfrak{R}^m. \end{aligned} \tag{3.6c}$$

Combining (3.6a), (3.6b) and (3.6c), and using the notations of (3.2b), the assertion of this lemma is proved. □

Remark 3.2 The matrix Q (3.5b) is symmetric positive definite. Thus, the proximal ADMM (3.4) is a customized PPA.

To accelerate the ADMM (3.4), we suggest to relax the dual variable by

$$\lambda^{k+1} = \lambda^k + (\gamma - 1)\beta A_2(x_2^k - \tilde{x}_2^k) - \gamma(\lambda^k - \tilde{\lambda}^k), \tag{3.7}$$

where the relaxation factor $\gamma \in (0, 2)$. Note that x_1^{k+1} always amounts to \tilde{x}_1^k and x_2^{k+1} always amounts to \tilde{x}_2^k , we have

$$w^{k+1} = w^k - M(w^k - \tilde{w}^k), \tag{3.8a}$$

where

$$M = \begin{pmatrix} I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & -(\gamma - 1)\beta A_2 & \gamma I_m \end{pmatrix}. \tag{3.8b}$$

Then, we check the positive definiteness of H and G to verify whether the condition (2.2) is fulfilled for the relaxed ADMM scheme. First, recall M (see (3.8b)), we have

$$M^{-1} = \begin{pmatrix} I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & \frac{\gamma-1}{\gamma}\beta A_2 & \frac{1}{\gamma}I_m \end{pmatrix},$$

and

$$\begin{aligned} H = QM^{-1} &= \begin{pmatrix} D_1 & 0 & 0 \\ 0 & D_2 + \beta A_2^T A_2 & -A_2^T \\ 0 & -A_2 & \frac{1}{\beta}I_m \end{pmatrix} \begin{pmatrix} I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & \frac{\gamma-1}{\gamma}\beta A_2 & \frac{1}{\gamma}I_m \end{pmatrix} \\ &= \begin{pmatrix} D_1 & 0 & 0 \\ 0 & D_2 + \frac{1}{\gamma}\beta A_2^T A_2 & -\frac{1}{\gamma}A_2^T \\ 0 & -\frac{1}{\gamma}A_2 & \frac{1}{\gamma\beta}I_m \end{pmatrix}. \end{aligned} \tag{3.9}$$

Thus H is positive definite. Next, we check the positive definiteness of G . We first deduce the expression of G :

$$\begin{aligned}
 G &= Q^T + Q - M^T H M = 2Q - M^T Q \\
 &= \begin{pmatrix} 2D_1 & 0 & 0 \\ 0 & 2(D_2 + \beta A_2^T A_2) & -2A_2^T \\ 0 & -2A_2 & \frac{2}{\beta} I_m \end{pmatrix} \\
 &\quad - \begin{pmatrix} I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & -(\gamma - 1)\beta A_2^T \\ 0 & 0 & \gamma I_m \end{pmatrix} \begin{pmatrix} D_1 & 0 & 0 \\ 0 & D_2 + \beta A_2^T A_2 & -A_2^T \\ 0 & -A_2 & \frac{1}{\beta} I_m \end{pmatrix} \\
 &= \begin{pmatrix} 2D_1 & 0 & 0 \\ 0 & 2(D_2 + \beta A_2^T A_2) & -2A_2^T \\ 0 & -2A_2 & \frac{2}{\beta} I_m \end{pmatrix} - \begin{pmatrix} D_1 & 0 & 0 \\ 0 & D_2 + \gamma\beta A_2^T A_2 & -\gamma A_2^T \\ 0 & -\gamma A_2 & \frac{\gamma}{\beta} I_m \end{pmatrix} \\
 &= \begin{pmatrix} D_1 & 0 & 0 \\ 0 & D_2 + (2 - \gamma)\beta A_2^T A_2 & -(2 - \gamma)A_2^T \\ 0 & -(2 - \gamma)A_2 & \frac{2 - \gamma}{\beta} I_m \end{pmatrix}.
 \end{aligned}
 \tag{3.10}$$

From (3.10), G is positive definite with $0 < \gamma < 2$. The relaxed ADMM scheme is thus a special case of the prototype algorithm (2.2). Note that

$$\begin{aligned}
 \lambda^{k+1} &= \tilde{\lambda}^k + (\gamma - 1)\beta A_2(x_2^k - \tilde{x}_2^k) - (\gamma - 1)(\lambda^k - \tilde{\lambda}^k) \\
 &= \tilde{\lambda}^k + (\gamma - 1)\beta A_2(x_2^k - \tilde{x}_2^k) - (\gamma - 1)\beta(A_1 \tilde{x}_1^k + A_2 x_2^k - b) \\
 &= \tilde{\lambda}^k - (\gamma - 1)\beta(A_1 \tilde{x}_1^k + A_2 \tilde{x}_2^k - b).
 \end{aligned}$$

Write (3.4) and (3.7) together in a compact form and recall $x_1^{k+1} = \tilde{x}_1^k$ and $x_2^{k+1} = \tilde{x}_2^k$, the relaxed ADMM is

Relaxed ADMM for (3.1)

$$\left\{ \begin{aligned}
 x_1^{k+1} &= \arg \min \{ \theta_1(x_1) - (\lambda^k)^T A_1 x_1 + \frac{\beta}{2} \|A_1 x_1 + A_2 x_2^k - b\|^2 + \|x_1 - x_1^k\|_{D_1}^2 \mid x_1 \in \mathcal{X}_1 \}, & (3.11a) \\
 \tilde{\lambda}^k &= \lambda^k - \beta(A_1 x_1^{k+1} + A_2 x_2^k - b), & (3.11b) \\
 x_2^{k+1} &= \arg \min \{ \theta_2(x_2) - (\tilde{\lambda}^k)^T A_2 x_2 + \frac{\beta}{2} \|A_1 x_1^{k+1} + A_2 x_2 - b\|^2 + \|x_2 - x_2^k\|_{D_2}^2 \mid x_2 \in \mathcal{X}_2 \}, & (3.11c) \\
 \lambda^{k+1} &= \tilde{\lambda}^k - (\gamma - 1)\beta(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b), & (3.11d)
 \end{aligned} \right.$$

where $\gamma \in (0, 2)$.

4 Global convergence

We have characterized the relaxed PDHG, relaxed L-ALM, relaxed ADMM as special cases of the Prototype Framework (2.1). Using the Prototype Framework (2.1), we can easily analyze their convergence property in a unified way.

Under the condition that H and G are positive definite, we now characterize the right hand side of (2.1a) in terms of $\|w - w^k\|_H$ and $\|w - w^{k+1}\|_H$. This is given in the following lemma.

Lemma 4.1 *Let \tilde{w}^k be generated by the step (2.1a) and $\{w^k\}$ be generated by (2.1b). We have*

$$\begin{aligned} \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) &\geq \frac{1}{2}(\|w - w^{k+1}\|_H^2 - \|w - w^k\|_H^2) \\ &+ \frac{1}{2}\|w^k - \tilde{w}^k\|_G^2, \quad \forall w \in \Omega, \end{aligned} \quad (4.1)$$

where H and G are defined in (2.2a) and (2.2b), respectively.

Proof Observe that $M(w^k - \tilde{w}^k) = (w^k - w^{k+1})$ (see (2.1b)) and $Q = HM$, (2.1a) can be written as

$$\theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq (w - \tilde{w}^k)^T H(w^k - w^{k+1}), \quad \forall w \in \Omega. \quad (4.2)$$

Applying the identity

$$(a - b)^T H(c - d) = \frac{1}{2}(\|a - d\|_H^2 - \|a - c\|_H^2) + \frac{1}{2}(\|c - b\|_H^2 - \|d - b\|_H^2),$$

with $a = w$, $b = \tilde{w}^k$, $c = w^k$, $d = w^{k+1}$ to the right hand side of (4.2), we have

$$\begin{aligned} (w - \tilde{w}^k)^T H(w^k - w^{k+1}) &= \frac{1}{2}(\|w - w^{k+1}\|_H^2 - \|w - w^k\|_H^2) \\ &+ \frac{1}{2}(\|w^k - \tilde{w}^k\|_H^2 - \|w^{k+1} - \tilde{w}^k\|_H^2). \end{aligned} \quad (4.3)$$

For the last term of (4.3), we have

$$\begin{aligned} &\|w^k - \tilde{w}^k\|_H^2 - \|w^{k+1} - \tilde{w}^k\|_H^2 \\ &= \|w^k - \tilde{w}^k\|_H^2 - \|(w^k - \tilde{w}^k) - (w^k - w^{k+1})\|_H^2 \\ &\stackrel{(2.1b)}{=} \|w^k - \tilde{w}^k\|_H^2 - \|(w^k - \tilde{w}^k) - M(w^k - \tilde{w}^k)\|_H^2 \\ &= 2(w^k - \tilde{w}^k)^T HM(w^k - \tilde{w}^k) - (w^k - \tilde{w}^k)^T M^T HM(w^k - \tilde{w}^k) \\ &\stackrel{(2.2)}{=} (w^k - \tilde{w}^k)^T (Q^T + Q - M^T HM)(w^k - \tilde{w}^k) \\ &\stackrel{(2.2b)}{=} (w^k - \tilde{w}^k)^T G(w^k - \tilde{w}^k). \end{aligned} \quad (4.4)$$

Combining (4.2), (4.3) and (4.4), the lemma is proved. \square

Theorem 4.2 *Let \tilde{w}^k be generated by the step (2.1a) and $\{w^k\}$ be generated by (2.1b). Then we have*

$$\|w^{k+1} - w^*\|_H^2 \leq \|w^k - w^*\|_H^2 - \|w^k - \tilde{w}^k\|_G^2, \quad \forall w^* \in \Omega^*. \tag{4.5}$$

Proof Setting $w = w^*$ in (4.1), we get

$$\begin{aligned} \|w^k - w^*\|_H^2 - \|w^{k+1} - w^*\|_H^2 &\geq \|w^k - \tilde{w}^k\|_G^2 \\ &\quad + 2\{\theta(\tilde{x}^k) - \theta(x^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k)\}. \end{aligned}$$

By using the optimality of w^* and the monotonicity of $F(w)$, we have

$$\theta(\tilde{x}^k) - \theta(x^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k) \geq \theta(\tilde{x}^k) - \theta(x^*) + (\tilde{w}^k - w^*)^T F(w^*) \geq 0,$$

and thus

$$\|w^k - w^*\|_H^2 - \|w^{k+1} - w^*\|_H^2 \geq \|w^k - \tilde{w}^k\|_G^2.$$

The assertion (4.5) follows directly. □

Since both H and G are positive definite, Theorem 4.2 implies that the sequence $\{w^k\}$ generated by (2.1) is strictly contractive with respect to the solution set Ω^* . Thus the global convergence of these relaxed PPA algorithms can be easily established. This is given by the following theorem.

Theorem 4.3 *Let $\{w^k\}$ be a sequence generated by (2.1). Then it globally converges to a point w^∞ which belongs to Ω^* .*

Proof From (4.5), we can see that $\{w^k\}$ is bounded, and

$$\lim_{k \rightarrow \infty} \|w^k - \tilde{w}^k\| = 0. \tag{4.6}$$

Thus, the sequence $\{\tilde{w}^k\}$ is also bounded. Let w^∞ be a cluster point of the sequence $\{\tilde{w}^k\}$. Since the sequence $\{\tilde{w}^k\}$ is bounded, it has a subsequence $\{\tilde{w}^{k_j}\}$ that converges to w^∞ . From (2.1a), we have

$$\tilde{w}^{k_j} \in \Omega, \quad \theta(x) - \theta(\tilde{x}^{k_j}) + (w - \tilde{w}^{k_j})^T F(\tilde{w}^{k_j}) \geq (w - \tilde{w}^{k_j})^T Q(w^{k_j} - \tilde{w}^{k_j}), \quad \forall w \in \Omega.$$

Using the continuity of $\theta(x)$ and $F(w)$, we have

$$w^\infty \in \Omega, \quad \theta(x) - \theta(x^\infty) + (w - w^\infty)^T F(w^\infty) \geq 0, \quad \forall w \in \Omega.$$

The above variational inequality implies that w^∞ is a solution point of $\text{VI}(\Omega; F; \theta)$. By using (4.6) and $\lim_{j \rightarrow \infty} \tilde{w}^{k_j} = w^\infty$, we get $\lim_{j \rightarrow \infty} w^{k_j} = w^\infty$. Recall (4.5), we have

$$\|w^{k+1} - w^\infty\|_H \leq \|w^k - w^\infty\|_H.$$

Thus the sequence $\{w^k\}$ converges to w^∞ . The proof is completed. □

5 Convergence rate

In this section, we analyze the iteration complexity in both the non-ergodic and ergodic sense for the sequence generated by these relaxed customized PPA algorithms. Note that by letting the matrix $M = I$, our analysis is also applicable for the original customized PPA.

5.1 Worse-case convergence rate in the non-ergodic sense

In this subsection, we establish the $\mathcal{O}(1/\sqrt{t})$ non-ergodic convergence rate of these relaxed PPA.

Lemma 5.1 *Let \tilde{w}^k be generated by the step (2.1a) and $\{w^k\}$ be generated by (2.1b). Then we have*

$$(\tilde{w}^k - \tilde{w}^{k+1})^T Q\{(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\} \geq 0. \tag{5.1}$$

Proof Setting $w = \tilde{w}^{k+1}$ in (2.1a), we get

$$\theta(\tilde{x}^{k+1}) - \theta(\tilde{x}^k) + (\tilde{w}^{k+1} - \tilde{w}^k)^T F(\tilde{w}^k) \geq (\tilde{w}^{k+1} - \tilde{w}^k)^T Q(w^k - \tilde{w}^k). \tag{5.2}$$

Notice that (2.1a) is also suitable for $k := k + 1$, thus

$$\theta(x) - \theta(\tilde{x}^{k+1}) + (w - \tilde{w}^{k+1})^T F(\tilde{w}^{k+1}) \geq (w - \tilde{w}^{k+1})^T Q(w^{k+1} - \tilde{w}^{k+1}), \quad \forall w \in \Omega. \tag{5.3}$$

Let $w = \tilde{w}^k$ in the above inequality, we get

$$\theta(\tilde{x}^k) - \theta(\tilde{x}^{k+1}) + (\tilde{w}^k - \tilde{w}^{k+1})^T F(\tilde{w}^{k+1}) \geq (\tilde{w}^k - \tilde{w}^{k+1})^T Q(w^{k+1} - \tilde{w}^{k+1}). \tag{5.4}$$

Summing (5.2) and (5.4) together and using the monotonicity of F , the assertion (5.1) is proved. □

Lemma 5.2 *Let \tilde{w}^k be generated by the step (2.1a) and $\{w^k\}$ be generated by (2.1b). Then we have*

$$(w^k - \tilde{w}^k)^T M^T H M\{(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\} \geq \|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_Q^2. \tag{5.5}$$

Proof Note that Q is positive definite. By adding the equation

$$\begin{aligned} & \{(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\}^T Q\{(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\} \\ &= \|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_Q^2 \end{aligned}$$

to the both sides of (5.1), we get

$$(w^k - w^{k+1})^T Q\{(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\} \geq \|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_Q^2. \tag{5.6}$$

Substituting the terms

$$M(w^k - \tilde{w}^k) = w^k - w^{k+1} \text{ and } Q = HM$$

into (5.6), we obtain

$$\begin{aligned} & (w^k - \tilde{w}^k)^T M^T HM\{(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\} \\ & \geq \|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_Q^2. \end{aligned}$$

This completes the proof. □

Theorem 5.3 *Let $\{w^k\}$ be a sequence generated by (2.1). Then we have*

$$\|w^{k+1} - w^{k+2}\|_H^2 \leq \|w^k - w^{k+1}\|_H^2, \tag{5.7}$$

where H is defined in (2.2a).

Proof Let $a = M(w^k - \tilde{w}^k)$ and $b = M(w^{k+1} - \tilde{w}^{k+1})$ be in the following identity

$$\|a\|_H^2 - \|b\|_H^2 = 2a^T H(a - b) - \|a - b\|_H^2,$$

we get

$$\begin{aligned} & \|M(w^k - \tilde{w}^k)\|_H^2 - \|M(w^{k+1} - \tilde{w}^{k+1})\|_H^2 \\ & = 2(w^k - \tilde{w}^k)^T M^T HM\{(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\} - \|(w^k - \tilde{w}^k) \\ & \quad - (w^{k+1} - \tilde{w}^{k+1})\|_{M^T HM}^2. \end{aligned} \tag{5.8}$$

Substitute (5.5) into the right hand side of above equality, we get

$$\begin{aligned} & \|M(w^k - \tilde{w}^k)\|_H^2 - \|M(w^{k+1} - \tilde{w}^{k+1})\|_H^2 \\ & \geq \|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_{(Q^T + Q - M^T HM)}^2. \end{aligned}$$

Note that

$$M(w^k - \tilde{w}^k) = w^k - w^{k+1} \text{ and } M(w^{k+1} - \tilde{w}^{k+1}) = w^{k+1} - w^{k+2}.$$

Combining with $G = Q^T + Q - M^T HM > 0$ (see (2.2b)), the assertion is proved. □

Theorem 5.3 implies that the sequences $\{\|w^k - w^{k+1}\|_H^2\}$ is monotonically non-increasing, which is crucial for our proof. Next, we need to establish the connection between $\|w^k - \tilde{w}^k\|_G^2$ and $\|w^k - w^{k+1}\|_H^2$.

Lemma 5.4 *For each relaxed customized PPA, there exists a constant $c := \frac{2-\gamma}{4}$, such that*

$$c\|w^k - w^{k+1}\|_H^2 < \|w^k - \tilde{w}^k\|_G^2. \tag{5.9}$$

Proof From $w^k - w^{k+1} = M(w^k - \tilde{w}^k)$ (see (2.1b)), we have

$$\|w^k - w^{k+1}\|_H^2 = \|w^k - \tilde{w}^k\|_{M^T H M}^2.$$

The remaining task is to find a constant c , such that

$$G \succ cM^T H M,$$

for each G, H pair emerged in these algorithms. Note that from (2.2b), we have $M^T H M < 2Q$. From (2.3) and (2.7), we have $G \succ \frac{2-\gamma}{2}Q$ for the relaxed PDHG. From (2.19) and (2.15), $G \succ \frac{2-\gamma}{2}Q$ for the relaxed L-ALM. From (3.5b) and (3.10), $G \succ \frac{2-\gamma}{2}Q$ for the relaxed ADMM. The assertion is thus proved. \square

Theorem 5.5 *Let $\{w^k\}$ be a sequence generated by (2.1). Then we have*

$$\|w^k - w^{k+1}\|_H^2 \leq \frac{1}{c(k+1)} \|w^0 - w^*\|_H^2, \quad \forall w^* \in \Omega^*, \tag{5.10}$$

where the matrix H is defined as (2.2a).

Proof Inserting (5.9) into (4.5), we get

$$\|w^{k+1} - w^*\|_H^2 \leq \|w^k - w^*\|_H^2 - c\|w^k - w^{k+1}\|_H^2, \quad \forall w^* \in \Omega^*.$$

Summing this inequality over $t = 0, 1, \dots, \infty$, we have

$$c \sum_{t=0}^{\infty} \|w^t - w^{t+1}\|_H^2 \leq \|w^0 - w^*\|_H^2, \quad \forall w^* \in \Omega^*. \tag{5.11}$$

Now observe that the sequences $\{\|w^k - w^{k+1}\|_H^2\}$ is monotonically nonincreasing in Theorem 5.3, we have

$$(k+1)\|w^k - w^{k+1}\|_H^2 \leq \sum_{t=0}^k \|w^t - w^{t+1}\|_H^2. \tag{5.12}$$

Using the above two inequalities (5.11) and (5.12), we obtain (5.10). \square

From (1.15a), we know that if $\|w^k - w^{k+1}\|_H = 0$, then w^k is the solution point of $\text{VI}(\Omega; F; \theta)$. This allows us to use $\|w^k - w^{k+1}\|_H$ as an error measurement in terms of the distance to the solution set of $\text{VI}(\Omega; F; \theta)$ for the t -th iteration of the algorithm. Let $d := \inf\{\|w^0 - w^*\|_H \mid w^* \in \Omega^*\}$. Then, for any given $\varepsilon > 0$,

Theorem 5.5 implies that these relaxed PPA needs at most $\lceil \frac{d^2}{c\varepsilon^2} \rceil$ iterations to ensure that $\|w^k - w^{k+1}\|_H \leq \varepsilon$. Therefore, a worse-case $\mathcal{O}(1/\sqrt{t})$ convergence rate in the non-ergodic sense for these relaxed PPA is established.

5.2 Worse-case convergence rate in ergodic sense

In this section, we establish $\mathcal{O}(1/t)$ worse-case convergence rate in the ergodic sense for these relaxed customized PPA algorithms. First, we introduce the following lemma, which is convenient for analyzing the ergodic iteration complexity of the algorithm. Its proof can be found in [24] or Theorem 2.3.5 in [15].

Theorem 5.6 *The solution set of $VI(\Omega; F; \theta)$ ((1.3), (3.2)) is closed and convex, and can be characterized via*

$$\Omega^* := \bigcap_{w \in \Omega} \{ \tilde{w} \in \Omega : \theta(x) - \theta(\tilde{x}) + (w - \tilde{w})^T F(w) \geq 0 \}.$$

Computing exactly a primal-dual solution w^* is impractical, but we can find an approximate solution $\tilde{w} \in \Omega$. With the Theorem 5.6, we define an ε -approximation solution of $VI(\Omega; F; \theta)$ as follows: Given an accuracy $\varepsilon > 0$, \tilde{w} is said to be an ε -approximation solution of $VI(\Omega; F; \theta)$, if it satisfies

$$\sup\{\theta(\tilde{x}) - \theta(x) + (\tilde{w} - w)^T F(w)\} \leq \varepsilon, \quad \forall w \in \mathcal{D},$$

where $\mathcal{D} \in \Omega$ is a defined compact set, e.g., $\mathcal{D} = \{w \mid \|w - \tilde{w}\| \leq 1\}$. Next, our purpose is to prove that we can find an ε -solution after t iterations.

Theorem 5.7 *Let \tilde{w}^k be generated by the step (2.1a) and $\{w^k\}$ be generated by (2.1b). For any integer number $t > 0$, define*

$$\tilde{w}_t = \frac{1}{t+1} \sum_{k=0}^t \tilde{w}^k, \tag{5.13}$$

then we have $\tilde{w}_t \in \Omega$ and

$$\theta(\tilde{x}_t) - \theta(x) + (\tilde{w}_t - w)^T F(w) \leq \frac{1}{2(t+1)} \|w - w_0\|_H^2, \quad \forall w \in \Omega, \tag{5.14}$$

where H is defined in (2.2a).

Proof First, because of $\tilde{w}^k \in \Omega$, it holds that $\tilde{w}_t \in \Omega$ for all integer t .

Next, by using the definitions of F (see (1.3b) and (3.2b)), we obtain,

$$(w - \tilde{w}^k)^T F(w) \geq (w - \tilde{w}^k)^T F(\tilde{w}^k).$$

Also, using the positive definiteness of G and (4.1), we get

$$\theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(w) + \frac{1}{2} \|w - w^k\|_H^2 \geq \frac{1}{2} \|w - w^{k+1}\|_H^2, \quad \forall w \in \Omega. \tag{5.15}$$

Summing inequality (5.15) from $k = 0$ to $k = t$, we derive

$$(t + 1)\theta(x) - \sum_{k=0}^t \theta(\tilde{x}^k) + ((t + 1)w - \sum_{k=0}^t \tilde{w}^k)^T F(w) + \frac{1}{2} \|w - w^0\|_H^2 \geq 0, \quad \forall w \in \Omega.$$

From the definition of \tilde{w}_t (5.13), we can write above inequality as

$$\frac{1}{t + 1} \sum_{k=0}^t \theta(\tilde{x}^k) - \theta(x) + (\tilde{w}_t - w)^T F(w) \leq \frac{1}{2(t + 1)} \|w - w^0\|_H^2, \quad \forall w \in \Omega. \tag{5.16}$$

Noticing the fact that $\theta(x)$ is convex, for \tilde{w}_t , we have

$$\theta(\tilde{x}_t) \leq \frac{1}{t + 1} \sum_{k=0}^t \theta(\tilde{x}^k). \tag{5.17}$$

Substituting (5.17) into (5.16), we obtain (5.14). □

Let $d = \sup \{\|w - w_0\|_H, w \in \mathcal{D}\}$. Theorem 5.7 implies that, after t iterations of (2.1), \tilde{w}_t defined in (5.13) satisfies

$$\sup_{w \in \mathcal{D}} \{\theta(\tilde{x}_t) - \theta(x) + (\tilde{w}_t - w)^T F(w)\} \leq \frac{d^2}{2(t + 1)}.$$

Thus, we obtained computational complexity estimates for the average of \tilde{w}^k , which is of order $\mathcal{O}(1/t)$ to get an approximation solution for VI(Ω, F, θ).

6 Numerical results

In this section, we report some numerical results of the proposed algorithms. Note that the numerical comparisons with related methods for the customized PPA have already been conducted in the literatures, see, e.g., [6,35,38]. Thus they are omitted in our tests and we focus on showing the advantages of the relaxed methods to the original customized PPA algorithms, and verify the improvements in the computational sense. All codes were written in Matlab 2015b and all experiments were conducted on a laptop with Intel Core (TM) CPU 2.40GHz and 8G memory.

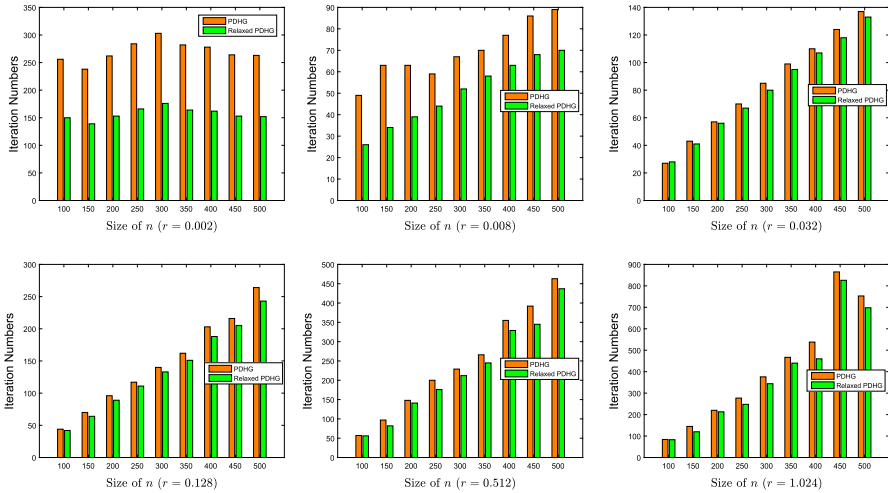


Fig. 1 Comparison results of PDHG and relaxed PDHG with different r ($r_a = 8, OS = 5$)

6.1 Matrix completion problem

The matrix completion problem consists of reconstructing an unknown low rank matrix from a given subset of observed entries [3]. Specifically, let $M \in \mathbb{R}^{m \times n}$ be a low rank matrix, Ω be a subset of the indices of entries $\{1, \dots, m\} \times \{1, \dots, n\}$. The convex model is

$$\min\{\|X\|_* \mid X_{ij} = M_{ij}, \forall \{i, j\} \in \Omega\}, \tag{6.1}$$

where $\|\cdot\|_*$ denotes the nuclear norm of a matrix. Note that the constraints can also be viewed as a projection equation $P_\Omega(X) = P_\Omega(M)$, thus we have $\|A^T A\| = 1$.

In our implementations, for the SVD decomposition, we resort to the widely recommended PROPACK package [28]. We generate the rank- r_a matrix M as a product $M_L M_R^T$, where M_L and M_R are independent $n \times r_a$ matrices who have i.i.d. Gaussian entries. The set of observations Ω is sampled uniformly at random among all sets of cardinality $|\Omega|$. The degrees of freedom of the matrix is $d_{r_a} = r_a(2n - r_a)$, then the oversampling factor $|\Omega|/d_{r_a}$ is the ratio of the number of samples to the degrees of freedom. The primal variable X^0 and the dual variable Λ^0 were all initialized as zeros(n). The stopping criteria is set to be

$$\frac{\|X^k - M\|_F}{\|M\|_F} \leq 10^{-4}. \tag{6.2}$$

For each instance, we randomly generated ten cases, so the results reported were averaged over ten runs.

Now we show how to choose the parameters. Note that the PDHG requires multiple stepsize parameters, and it seems impossible to give a best choice of step sizes. We already know from (1.11) that $1/r$ and $1/s$ essentially play the role of proximal step

Table 1 Performance comparison of PDHG, Relaxed PDHG with $r = 0.008$

$n \times n$ matrix				PDHG			Relaxed PDHG($\gamma = 1.7$)		
n	$rank(r_a)$	$ \Omega /d_{r_a}$	$ \Omega /n^2$	RelErr	No. it	Time	RelErr	No. it	Time
100	5	4	0.39	9.59e-05	139	0.30	9.99e-05	79	0.18
100	5	5	0.49	9.50e-05	79	0.20	9.62e-05	44	0.11
100	5	6	0.58	8.72e-05	51	0.13	9.28e-05	27	0.07
100	10	3	0.57	9.59e-05	87	0.22	9.43e-05	49	0.12
100	10	4	0.76	9.58e-05	43	0.11	7.72e-05	24	0.06
100	15	3	0.83	8.94e-05	42	0.11	8.95e-05	22	0.07
300	5	5	0.17	9.93e-05	94	1.06	9.09e-05	58	0.61
300	5	6	0.20	9.45e-05	73	0.61	7.64e-05	53	0.46
300	10	4	0.26	8.97e-05	51	0.84	8.70e-05	39	0.72
300	10	5	0.33	7.89e-05	39	0.57	8.79e-05	33	0.55
300	15	3	0.29	9.19e-05	58	1.11	8.29e-05	41	0.74
300	20	3	0.39	9.29e-05	41	0.78	9.35e-05	35	0.65
500	5	6	0.12	8.76e-05	93	3.27	9.23e-05	73	2.63
500	5	8	0.16	9.87e-05	73	1.74	9.45e-05	61	1.50
500	10	4	0.16	9.50e-05	73	2.37	9.97e-05	62	2.30
500	10	5	0.20	9.10e-05	58	2.52	9.77e-05	53	2.13
500	20	3	0.24	9.12e-05	61	3.30	8.99e-05	58	4.14
500	25	3	0.29	9.68e-05	54	3.51	9.19e-05	52	3.76
1000	10	6	0.12	9.78e-05	97	20.07	9.33e-05	95	25.37
1000	20	3	0.12	9.49e-05	116	63.47	9.96e-05	108	58.33
1000	20	6	0.24	9.23e-05	69	22.52	9.02e-05	66	22.86
1000	50	6	0.58	7.01e-05	34	11.66	9.78e-05	33	11.32
1000	50	3	0.29	9.64e-05	76	52.02	9.40e-05	69	49.17
1000	60	3	0.35	9.30e-05	68	46.74	9.34e-05	62	36.68

sizes, and it may be preferable to choose r and s such that $r \cdot s$ is close to the lower bound of $\|A^T A\|$ by considering the convergence condition $rs > \|A^T A\|$. Thus, in our experiments, we always set $s = 1.01/r$. However, the choice of r still lacks of theoretical analysis, and as far as we know, it is usually selected by numerical experiments. Since the condition varies for different problem size, or even different iteration step, it seems that adjusting the PDHG parameters adaptively at each iteration is a better way is to achieve faster convergence, see, e.g., [18] for a simple adaptive strategy. In this paper, our main goal is to illustrate the advantages of the relaxation steps, so the parameters are kept fixed in all our experiments. To choose a suitable value of r , We first run PDHG and relaxed PDHG ($\gamma = 1.7$) for several different values of r and record the resulting iteration numbers. In particular, r varies from 0.002 to 1.024. The results are reported in Fig. 1. We can see that the relaxed PDHG with $\gamma = 1.7$ took fewer iterations than the PDHG for almost all cases. We observe that both algorithms are sensitive to the choice of r . As a result, for different dimension of n , the proper

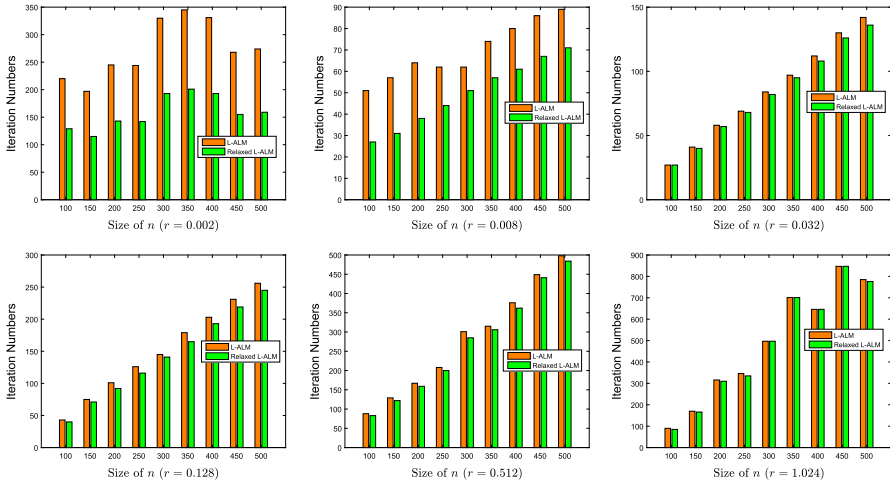


Fig. 2 Comparison results of L-ALM and relaxed L-ALM with different r ($r_a = 8, OS = 5$)

choice of r is different and there does not exist fixed and best parameters for PDHG throughout the implementation.

We further compare the two algorithms with specified values of the parameters. From Fig. 1, we see that when r is chosen in the interval $(0.008, 0.0032)$, the performance is efficient and robust. Here, we set the parameters r and s to be $r = 0.008$, $s = 1.01/r$, respectively. We point out that the values of r and s are determined based on the experiments and may be far from optimal. For each setting, we report the number of iterations (“No. it”), the relative error (“RelErr”), the time in Table 1, respectively. As shown in the figure, the relaxed PDHG took much fewer iterations and is much faster than the PDHG for the cases $n < 300$. When n becomes larger, the relaxed PDHG also needs less number of iterations than PDHG, although they both perform well. We remark that, we can further improve the performance of the relaxed PDHG by judiciously choosing a proper value of γ in the interval $(1, 2)$. But this is trivial and beyond the scope of this paper.

Next, we test the performance of L-ALM. Similarly with PDHG, the parameters r and β would be properly chosen so that r/β is close to the lower bound of $\|A^T A\|$. Thus, in our experiments, we always set $\beta = r/1.01$. Since the condition is similar to PDHG, we illustrate the performance with different values of r in Fig. 2. We can see that the relaxed L-ALM with $\gamma = 1.7$ took fewer iterations than the L-ALM for almost all cases. When $n < 500$ and $r \in (0.008, 0.032)$, the two algorithms cost less iterations and perform robust and efficient. When $r > 0.032$, the two algorithms require more iterations to achieve the same stopping criteria, and the relaxation step seems less effective to reduce the iteration numbers. We can note that the performance is remarkably similar to the PDHG. This phenomenon is not difficult to be understood: L-ALM can be explained as the linearized ADMM algorithm, and is closely related to the PDHG [14]. Here, we also set the parameters r and β to be $r = 0.008, \beta = r/1.01$, respectively. In Table 2, we report the performance of the relaxed L-ALM algorithm for different settings of the dimensions. We can see that the relaxed L-ALM is more

Table 2 Performance comparison of L-ALM, Relaxed L-ALM with $r = 0.008$

$n \times n$ matrix				L-ALM			Relaxed L-ALM($\gamma = 1.7$)		
n	$rank(r_a)$	$ \Omega /d_{r_a}$	$ \Omega /n^2$	RelErr	No. it	Time	RelErr	No. it	Time
100	5	4	0.39	9.63e-05	130	0.30	9.23e-05	75	0.17
100	5	5	0.49	9.75e-05	101	0.24	9.82e-05	57	0.13
100	5	6	0.58	9.78e-05	57	0.14	8.35e-05	32	0.08
100	10	3	0.57	9.98e-05	97	0.23	9.53e-05	55	0.13
100	10	4	0.76	8.91e-05	46	0.12	9.22e-05	25	0.07
100	15	3	0.83	8.86e-05	42	0.11	7.69e-05	22	0.06
300	5	5	0.17	9.83e-05	109	0.86	9.18e-05	61	0.52
300	5	6	0.20	9.12e-05	64	0.51	8.34e-05	52	0.42
300	10	4	0.26	9.02e-05	54	0.69	9.40e-05	39	0.48
300	10	5	0.33	7.85e-05	38	0.53	9.42e-05	33	0.45
300	15	3	0.29	9.87e-05	65	1.01	8.99e-05	40	0.56
300	20	3	0.39	9.91e-05	42	0.76	8.82e-05	34	0.61
500	5	6	0.12	9.90e-05	91	2.79	9.55e-05	72	2.22
500	5	8	0.16	9.82e-05	77	2.46	9.21e-05	64	2.06
500	10	4	0.16	9.98e-05	66	2.84	9.97e-05	63	3.01
500	10	5	0.20	8.42e-05	58	2.73	9.78e-05	54	1.87
500	20	3	0.24	9.77e-05	59	3.91	8.74e-05	57	3.81
500	25	3	0.29	8.95e-05	53	2.47	9.33e-05	51	2.94
1000	10	6	0.12	9.63e-05	96	19.75	9.43e-05	94	21.35
1000	20	3	0.12	9.66e-05	114	56.54	9.73e-05	107	67.97
1000	20	6	0.24	9.14e-05	67	22.08	9.66e-05	64	19.46
1000	50	6	0.58	9.25e-05	32	15.52	8.98e-05	30	14.92
1000	50	3	0.29	9.75e-05	77	59.54	9.87e-05	74	54.71
1000	60	3	0.35	9.10e-05	69	44.23	9.50e-05	66	49.97

effective than the L-ALM when $n < 300$. For large problem dimension, the results also demonstrate that the relaxation took less iterations, and can reduce 10% iterations numbers.

6.2 Robust principal component analysis

In this subsection, we test the robust principal component analysis (RPCA) which has the form

$$\begin{aligned}
 & \min_{L,S} \quad \|L\|_* + \tau \|S\|_1, \\
 & \text{s.t.} \quad L + S = M.
 \end{aligned}
 \tag{6.3}$$

where M is a known matrix. Note that the problem (6.3) is a standard model of (3.1), and thus can be solved by ADMM. We refer to [36] for the strategies to solve the

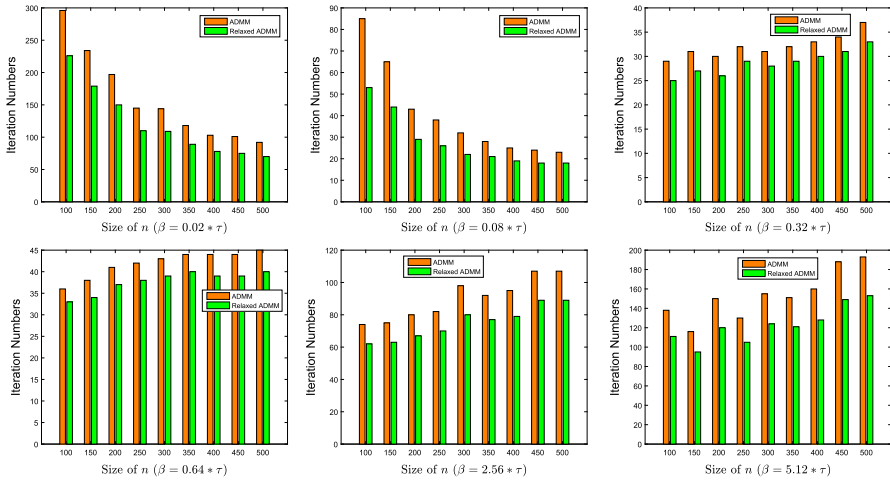


Fig. 3 Comparison results of ADMM and relaxed ADMM with different $\beta = k * \tau$

subproblems. We now demonstrate the efficiency of the relaxed ADMM for solving (6.3). In the experiments, we generated the data similar to [36]. More specially, we generate the rank- r_a matrix L^* as a product $L_L L_R^T$, where L_L and L_R are independent $n \times r_a$ matrices who have i.i.d. Gaussian entries. For S^* , we first generate a $n \times n$ zero matrix S , and let the numbers of the nonzero entries amount to the nearest integer of $sr \times n \times n$, where sr denotes the sparsity ratio. We then set the locations of the nonzero entries of the underlying sparse matrix S^* randomly, where the nonzero elements are generated by standard Gaussian distribution. We use $L^* + S^*$ to generate the matrix M . The model parameter τ is always set to be $1/\sqrt{n}$, and the stopping criteria is set to be

$$\max \left\{ \frac{\|L - L^*\|_F}{\|L^*\|_F}, \frac{\|S - S^*\|_F}{\|S^*\|_F} \right\} \leq 10^{-6}. \tag{6.4}$$

We also randomly generated ten cases for each instance, and the results reported were averaged over ten runs.

It is well-known that, the numerical performance of ADMM is highly dependent on the value of β , and how to determine the optimal value of β is a theoretically unknown question. To select a reasonably good value of β , we first conduct some experiments to evaluate the performance of ADMM with different values of β . In the experiments, we test a set of values $\beta = k * \tau$ with $k = 0.02, 0.08, \dots, 5.12$ on different dimensions from $n = 100$ to $n = 500$. The initial point is set to be zero. For simplicity, we always set $r_a = 0.1 * n$ and the sparsity ratio $sr = 0.1$. For the relaxed ADMM, the relaxation factor is set to be $\gamma = 1.3$. The number of iterations of ADMM and the relaxed ADMM with respect to different values of β ($\beta = k * \tau$) are plotted in Fig. 3, respectively. According to the plots, the ADMM methods are very sensitive to the value of β . For various dimensions of the problem, we can see that the optimal value of β could be different and there does not exist an optimal choice of β that is data-dependent. We can also find that the relaxed ADMM performs better than ADMM for most cases. In Fig. 3, we found that when $k \in (0.08, 0.32)$, i.e., $\beta \in (0.08\tau, 0.32\tau)$,

Table 3 Performance comparison of ADMM and relaxed ADMM with $\beta = 0.3 * \tau$

$n \times n$ matrix			ADMM			Relaxed ADMM		
n	$rank(r_a)$	sr	RelErr	No. it	Time	RelErr	No. it	Time
100	5	0.05	4.91e-07	19	0.06	5.67e-07	16	0.05
100	5	0.10	6.60e-07	21	0.07	4.23e-07	19	0.06
100	10	0.05	8.35e-07	23	0.07	5.00e-07	20	0.06
100	10	0.10	8.93e-07	26	0.08	7.76e-07	23	0.07
100	15	0.05	9.26e-07	30	0.09	5.51e-07	24	0.07
100	15	0.10	8.08e-07	34	0.10	7.93e-07	27	0.08
200	10	0.05	7.94e-07	19	0.22	9.77e-07	17	0.19
200	10	0.10	4.64e-07	25	0.27	8.11e-07	21	0.22
200	20	0.05	7.30e-07	24	0.27	9.15e-07	21	0.23
200	20	0.10	6.70e-07	32	0.33	6.46e-07	27	0.29
200	30	0.05	8.57e-07	29	0.31	8.85e-07	25	0.26
200	30	0.10	7.06e-07	36	0.39	8.75e-07	30	0.32
500	25	0.05	8.51e-07	19	1.57	7.47e-07	19	1.58
500	25	0.10	6.05e-07	25	2.03	7.89e-07	23	1.77
500	50	0.05	6.74e-07	26	2.04	8.67e-07	24	1.81
500	50	0.10	9.38e-07	33	2.48	8.48e-07	30	2.30
500	75	0.05	7.35e-07	32	2.45	9.68e-07	28	2.17
500	75	0.10	9.25e-07	42	3.23	8.74e-07	37	2.86
1000	50	0.05	4.58e-07	21	9.79	7.77e-07	20	9.31
1000	50	0.10	4.89e-07	27	12.85	5.35e-07	26	12.53
1000	100	0.05	6.83e-07	27	13.08	8.62e-07	26	12.01
1000	100	0.10	9.56e-07	38	17.96	9.20e-07	34	16.05
1000	150	0.05	6.95e-07	36	18.87	9.72e-07	32	16.65
1000	150	0.10	7.05e-07	48	24.35	8.89e-07	42	20.82
2000	100	0.05	9.41e-07	22	109.76	6.26e-07	22	106.19
2000	100	0.10	5.16e-07	30	139.17	8.84e-07	28	132.05
2000	200	0.05	8.04e-07	30	138.48	5.98e-07	29	128.76
2000	200	0.10	5.21e-07	44	205.96	9.45e-07	39	187.31
2000	300	0.05	5.06e-07	40	172.60	7.80e-07	36	153.15
2000	300	0.10	8.99e-07	57	247.06	7.50e-07	51	219.14

the ADMM performs more robust and efficient, and the two algorithms all cost less iterations compared with other choices. When $\beta > 0.32\tau$, both algorithms need more iterations for the same problem, and their performances are almost same.

Next, we report the experimental results of ADMM and the relaxed ADMM for more cases. For comparisons, we just set $\beta = 0.3 * \tau$ for the two methods, since it demonstrates a good performance and is robust for different dimensions. The results of these algorithms are reported in Table 3. From these tests, we can see that the relaxed ADMM requires less iterations and computing time than ADMM to produce the same

accurate solutions. For these tested instances, we can further improve the performance of relaxed ADMM by choosing the best relaxation factor. However, to simplify the demonstration, we used a fixed factor for all cases.

7 Conclusions

In the literature, it is well known that the proximal point algorithm (PPA) can be accelerated by using the relaxation strategy proposed in [19]. For many cases, the relaxed PPA with $\gamma \in [1.5, 1.8]$ can obtain a guaranteed improvement of the convergence. In this paper, different from most previous works, we suggest to relax the PPA by correcting the dual variables individually. For the PDHG proposed in [5,6], the linearized augmented Lagrangian method (L-ALM) and the alternating direction method of multipliers (ADMM), which are special cases of the customized PPA, we studied their relaxed schemes and observed the corresponding speedup. With the guidance of the variational inequality, we established the global convergence of these relaxed algorithms and the worst-case convergence rate in both ergodic and nonergodic sense. In conclusion, our results theoretically provide a new way of relaxing PPA algorithms.

It should be mentioned that this paper only discussed the customized PPA algorithms for the convex problems less than two blocks. In many real applications, the problems may be modeled in the form of (1.1) with the objective function consisting of multiple block components, and it is often more challenging to directly handle them. For the multi-block cases, we can also formulate them as variational inequality problems, and then design some customized PPA algorithms to solve them. In fact, the multi-block Jacobian ADMM [12,34] and its linearized version, the partial parallel ADMM [25] are all designed for this purpose and can be viewed as PPA algorithms. Consequently, the relaxation strategy proposed in this paper can be further extended to these customized PPA algorithms, and therefore, gives to the relaxed versions. On the other hand, the stepsize parameters of the PPA algorithms discussed in this paper can all be tuned at each iteration by some self-adaptive rules, and the resulting performance would be further improved. So it would be interesting to investigate the self-adaptive rule for the relaxation factor when these algorithms are dynamically adjusted. We leave these as potential directions of future work.

Acknowledgements The author is grateful to the associate editor and two anonymous reviewers for their valuable comments and suggestions that have helped improve the presentation of this paper greatly. This work was supported by the NSFC Grants 11701564 and 11871029.

References

1. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2**, 183–202 (2009)
2. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**, 1–122 (2010)
3. Cai, J.F., Candès, E.J., Shen, Z.W.: A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.* **20**, 1956–1982 (2010)

4. 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**, 2179–2186 (2013)
5. Chambolle, A., Pock, T.: Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 1762–1769 (2011)
6. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**, 120–145 (2011)
7. Chambolle, A., Pock, T.: On the ergodic convergence rates of a first-order primal-dual algorithm. *Math. Program.* **159**, 253–287 (2016)
8. Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numer.* **25**, 161–319 (2016)
9. Chan, T.F., Glowinski, R.: *Finite Element Approximation and Iterative Solution of a Class of Mildly Nonlinear Elliptic Equations*, Technical Report STAN-CS-78-674. Stanford University, Computer Science Department (1978)
10. Daubechies, I., Defrise, M., Mol, C.D.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.* **57**, 1413–1457 (2004)
11. Deng, W., Yin, W.T.: On the global and linear convergence of the generalized alternating direction method of multipliers. *J. Sci. Comput.* **66**, 889–916 (2016)
12. Deng, W., Lai, M.J., Peng, Z.M., Yin, W.T.: Parallel multi-block ADMM with $o(1/k)$ convergence. *J. Sci. Comput.* **71**, 712–736 (2017)
13. Eckstein, J., Bertsekas, D.P.: On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.* **55**, 293–318 (1992)
14. Esser, E.: *Primal Dual Algorithms for Convex Models and Applications to Image Restoration, Registration and Nonlocal Inpainting*. University of California, Los Angeles (2010)
15. Facchinei, F., Pang, J.S.: *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer, New York (2003)
16. Glowinski, R., Marrocco, A.: Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires. *Revue Fr. Autom. Inform. Rech. Opér. Anal. Numér.* **2**, 41–76 (1975)
17. Glowinski, R., Osher, S.J., Yin, W. (eds.): *Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer, New York (2016)
18. Goldstein, T., Li, M., Yuan, X.M.: Adaptive primal-dual splitting methods for statistical learning and image processing. In: *Advances in Neural Information Processing Systems*, pp. 2089–2097 (2015)
19. Gol’shtein, E.G., Tre’t’yakov, N.V.: Modified Lagrangians in convex programming and their generalizations. *Math. Program. Stud.* **10**, 86–97 (1979)
20. Gu, G.Y., 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)
21. Han, D.R., He, H.J., Yang, H., Yuan, X.M.: A customized Douglas–Rachford splitting algorithm for separable convex minimization with linear constraints. *Numer. Math.* **127**, 167–200 (2014)
22. He, B.S., Yuan, X.M.: A class of ADMM-based algorithms for three-block separable convex programming. *Comput. Optim. Appl.* **70**, 791–826 (2018)
23. He, B.S.: PPA-Like contraction methods for convex optimization: a framework using variational inequality approach. *J. Oper. Res. Soc. China* **3**, 391–420 (2015)
24. He, B.S., Yuan, X.M.: Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM J. Imaging Sci.* **5**, 119–149 (2012)
25. He, B.S., Yuan, X.M.: Block-wise alternating direction method of multipliers for multiple-block convex programming and beyond. *SMAI J. Comput. Math.* **1**, 145–174 (2015)
26. He, B.S., Ma, F., Yuan, X.M.: An algorithmic framework of generalized primal-dual hybrid gradient methods for saddle point problems. *J. Math. Imaging Vis.* **58**, 279–293 (2017)
27. Hestenes, M.R.: Multiplier and gradient methods. *J. Optim. Theory Appl.* **4**, 303–320 (1969)
28. Larsen, R.M.: *PROPACK Software for large and sparse SVD calculations*. Stanford University. <http://sun.stanford.edu/~rmunk/PROPACK/> (1969)
29. Martinet, B.: Regularisation, d’inéquations variationnelles par approximations successives. *Rev. Fr. d’Inform. Rech. Opér.* **4**, 154–159 (1970)
30. Powell, M.J.D.: A method for nonlinear constraints in minimization problems. In: *Fletcher, R. (ed.) Optimization*, pp. 283–298. Academic Press, New York (1969)

31. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.* **14**, 877–898 (1976)
32. Shen, Y., Wang, H.Y.: New augmented Lagrangian-based proximal point algorithm for convex optimization with equality constraints. *J. Optim. Theory Appl.* **171**, 251–261 (2016)
33. Sra, S., Nowozin, S., Wright, S.J.: *Optimization for Machine Learning*. MIT Press, Cambridge (2012)
34. Wang, X.F., Hong, M.Y., Ma, S.Q., Luo, Z.Q.: Solving multiple-block separable convex minimization problems using two-block alternating direction method of multipliers. *Pac. J. Optim.* **11**, 645–667 (2015)
35. Yang, J.F., Yuan, X.M.: Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Math. Comput.* **82**, 301–329 (2013)
36. Yuan, X.M., Yang, J.F.: Sparse and low-rank matrix decomposition via alternating direction methods. *Pac. J. Optim.* **9**, 167–180 (2013)
37. Zhang, X.Q., Burger, M., Bresson, X., Osher, S.: Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM J. Imaging Sci.* **3**, 253–276 (2010)
38. Zhang, X.Q., Burger, M., Osher, S.: A unified primal-dual algorithm framework based on Bregman iteration. *J. Sci. Comput.* **46**, 20–46 (2010)
39. Zhu, M., Chan, T.F.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration, CAM Report 08-34. UCLA, USA (2008)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.