ACCELERATED STOCHASTIC PEACEMAN-RACHFORD METHOD FOR EMPIRICAL RISK MINIMIZATION*

JIANCHAO BAI†, FENGMIAO BIAN‡, XIAOKAI CHANG§, AND LIN DU¶

Abstract. This work is devoted to studying an Accelerated Stochastic Peaceman-Rachford Splitting Method (AS-PRSM) for solving a family of structural empirical risk minimization problems. The objective function to be optimized is the sum of a possibly nonsmooth convex function and a finite-sum of smooth convex component functions. The smooth subproblem in AS-PRSM is solved by a stochastic gradient method using variance reduction technique and accelerated techniques, while the possibly nonsmooth subproblem is solved by introducing an indefinite proximal term to transform its solution into a proximity operator. By a proper choice for the involved parameters, we show that AS-PRSM converges in a sublinear convergence rate measured by the function value residual and constraint violation in the sense of expectation and ergodic. Preliminary experiments on testing the popular graph-guided fused lasso problem in machine learning and the 3D CT reconstruction problem in medical image processing show that the proposed AS-PRSM is very efficient.

Key words. empirical risk minimization, convex optimization, stochastic Peaceman-Rachford method, indefinite proximal term, complexity

AMS subject classifications. 65K10, 65Y20, 90C25

1. Introduction.

In this paper, we are interested in solving the following structural empirical risk minimization problem

$$\min_{\mathbf{x}\in\mathcal{X},\,\mathbf{y}\in\mathcal{Y}} f(\mathbf{x})+g(\mathbf{y}) \qquad \text{s.t. } A\mathbf{x}+B\mathbf{y}=\mathbf{b}, \tag{1.1}$$

where f is an average of a set of continuously differentiable functions, i.e.,

$$f(\mathbf{x})=\frac{1}{N}\sum_{j=1}^{N} f_j(\mathbf{x}), \tag{1.2}$$

the scalar N denotes the sample size and $f_j(\mathbf{x})$ corresponds to the empirical loss on the j-th sample data, such as $f_j(\mathbf{x})=\log\big(1+\exp(-b_j a_j^{\mathsf{T}}\mathbf{x})\big)$; $g:\mathcal{Y}\to\mathbb{R}\cup\{+\infty\}$ is a convex regularizer but possibly nonsmooth, for example, $g(\mathbf{y})=\mu\|\mathbf{y}\|_1$; $\mathcal{X}\subset\mathbb{R}^{n_1},\mathcal{Y}\subset\mathbb{R}^{n_2}$ are *simple* closed convex sets, that is, the projections onto $\mathcal{X}$ and $\mathcal{Y}$ can be calculated easily; $A\in\mathbb{R}^{n\times n_1},B\in\mathbb{R}^{n\times n_2}$ are given nonzero matrices, and $\mathbf{b}\in\mathbb{R}^n$ is a given vector. Here, the symbols $\mathbb{R}$, $\mathbb{R}^n$ and $\mathbb{R}^{n\times n_1}$ denote the sets of real numbers, n dimensional real column vectors, and $n\times n_1$ dimensional real matrices, respectively. Problems in the form of (1.1) arise in some practical applications of data processing, for example, artificial intelligence, machine learning and medical image processing, cf. [3, 5, 6, 22, 23, 29] to list a few.

*This research was supported by the National Natural Science Foundation of China under Grants 12001430, 11972292 and 12161053, the China Postdoctoral Science Foundation under Grant 2020M683545.

†✉ jianchaobai@nwpu.edu.cn, School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710129, China.

‡ Department of Mathematics, The Hong Kong University of Science and Technology, Hong Kong, China.

§✉ xkchang@lut.cn, School of Science, Lanzhou University of Technology, Lanzhou 730050, China.

¶ School of Mathematics and Statistics and the MIIT Key Laboratory of Dynamics and Control of Complex Systems, Northwestern Polytechnical University, Xi'an 710129, China.

The augmented Lagrangian function of (1.1) is given by

$$\mathcal{L}_\beta(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}) = \mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}) + \frac{\beta}{2}\|A\mathbf{x}+B\mathbf{y}-\mathbf{b}\|^2, \tag{1.3}$$

where $\boldsymbol{\lambda}\in\mathbb{R}^n$ and $\beta>0$ are respectively the primary and secondary penalty for the equality constraint, and $\mathcal{L}(\cdot)$ is the corresponding Lagrangian function:

$$\mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}) = f(\mathbf{x})+g(\mathbf{y})-\langle\boldsymbol{\lambda}, A\mathbf{x}+B\mathbf{y}-\mathbf{b}\rangle.$$

To solve the problem (1.1), one may consider the traditional Augmented Lagrangian Method (ALM) whose core subproblem is $\min_{(\mathbf{x},\mathbf{y})\in\mathcal{X}\times\mathcal{Y}}\mathcal{L}_\beta(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}^k)$. Generally speaking, it is not easy to solve this subproblem and usually needs an inner solver or linearization techniques, especially for the objective functions having special properties (e.g. smoothness, sparsity and low-rank) or involving big data. As a splitting version of ALM, the Alternating Direction Method of Multipliers (ADMM, [7, 14]) with unit stepsize for (1.1) reads

$$\begin{cases}\mathbf{x}^{k+1} = \arg\min\limits_{\mathbf{x}\in\mathcal{X}}\mathcal{L}_\beta(\mathbf{x},\mathbf{y}^k,\boldsymbol{\lambda}^k),\\ \mathbf{y}^{k+1} = \arg\min\limits_{\mathbf{y}\in\mathcal{Y}}\mathcal{L}_\beta(\mathbf{x}^{k+1},\mathbf{y},\boldsymbol{\lambda}^k),\\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k-\beta(A\mathbf{x}^{k+1}+B\mathbf{y}^{k+1}-\mathbf{b}).\end{cases} \tag{1.4}$$

An obvious feature of (1.4) is that each involved subproblem does not contain coupled variables and hence could be tackled efficiently by exploiting the structural properties of the objective functions. The ADMM (1.4) is in essence a serial algorithm with certain updating order $\mathbf{x}^{k+1}\to\mathbf{y}^{k+1}\to\boldsymbol{\lambda}^{k+1}$, and it optimizes over one block variable while fixing other variables per iteration. Moreover, followed by Theorem 4 [28, pp. 184-185] it is equivalent to another version with the order $\mathbf{y}^{k+1}\to\mathbf{x}^{k+1}\to\boldsymbol{\lambda}^{k+1}$. To make the utmost of previous iterations, an interesting algorithm called Generalized ADMM (G-ADMM, [11]) for solving (1.1) is

$$\begin{cases}\mathbf{x}^{k+1} = \arg\min\limits_{\mathbf{x}\in\mathcal{X}}\mathcal{L}_\beta(\mathbf{x},\mathbf{y}^k,\boldsymbol{\lambda}^k),\\ \mathbf{u}^{k+1} = sA\mathbf{x}^{k+1}+(1-s)(\mathbf{b}-B\mathbf{y}^k),\\ \mathbf{y}^{k+1} = \arg\min\limits_{\mathbf{y}\in\mathcal{Y}}g(\mathbf{y})-\langle\boldsymbol{\lambda}^k,B\mathbf{y}\rangle+\frac{\beta}{2}\left\|\mathbf{u}^{k+1}+B\mathbf{y}-\mathbf{b}\right\|^2,\\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k-\beta\left(\mathbf{u}^{k+1}+B\mathbf{y}^{k+1}-\mathbf{b}\right),\end{cases} \tag{1.5}$$

where $s\in(0,2)$ denotes a relaxation factor. Clearly, (1.5) with $s=1$ reduces to (1.4). Moreover, it was demonstrated in [11, 12] that the scheme (1.5) with $s\in(1,2)$ performs numerically better than other values. Recall the following particular Symmetric ADMM (SADMM):

$$\begin{cases}\mathbf{x}^{k+1} = \arg\min\limits_{\mathbf{x}\in\mathcal{X}}\mathcal{L}_\beta(\mathbf{x},\mathbf{y}^k,\boldsymbol{\lambda}^k),\\ \boldsymbol{\lambda}^{k+\frac{1}{2}} = \boldsymbol{\lambda}^k-\alpha\beta\left(A\mathbf{x}^{k+1}+B\mathbf{y}^k-\mathbf{b}\right),\\ \mathbf{y}^{k+1} = \arg\min\limits_{\mathbf{y}\in\mathcal{Y}}\mathcal{L}_\beta(\mathbf{x}^{k+1},\mathbf{y},\boldsymbol{\lambda}^{k+\frac{1}{2}}),\\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^{k+\frac{1}{2}}-\beta\left(A\mathbf{x}^{k+1}+B\mathbf{y}^{k+1}-\mathbf{b}\right),\end{cases}$$

where $\alpha \in (-1, 1)$. The above SADMM is a special case of the method in [19] with $s = 1$, and it is also equivalent to G-ADMM (1.5) by substituting $\boldsymbol{\lambda}^{k+\frac{1}{2}}$ into the last two iterations. The additional $\boldsymbol{\lambda}^{k+\frac{1}{2}}$ was designed with proper stepsize to balance the update of the dual variable. Since ADMM was often viewed as a splitting version of ALM, the dual variable may be updated after each primal update and with suitable stepsizes, that is, it could be updated symmetrically. Some researchers have demonstrated that SADMM with relatively larger stepsize performs better than the standard ADMM, see e.g. [2, 3, 19, 27]. For more variants of ADMM-type methods using larger stepsize of the dual variable, we refer to [1, 4, 16] and the references therein.

When B (or A) is not an identity matrix, a proximal term of the form

$$\frac{1}{2}\left\|\mathbf{y}-\mathbf{y}^k\right\|_{\mathcal{P}}^2 \quad \text{with} \quad \mathcal{P}=\mathcal{P}_1-\beta B^{\mathsf{T}}B \tag{1.6}$$

is often added to make the solution to the $\mathbf{y}$-subproblem become a proximity operator. In (1.6), P_1 is an arbitrarily positive semidefinite matrix and $\|\cdot\|_{\mathcal{P}}^2 := \langle\cdot, \mathcal{P}\cdot\rangle$ where the proximal matrix $\mathcal{P}$ is symmetric and usually takes $\mathcal{P} = \rho\mathbf{I} - \beta B^{\mathsf{T}}B$ with $\rho > \beta\|B^{\mathsf{T}}B\|$, see e.g. [17, 18]. Hereafter, the symbol $\|B^{\mathsf{T}}B\|$ denotes the spectral norm of $B^{\mathsf{T}}B$, that is, the square root of its largest eigenvalue. However, as said in [21] that "these methods with small value of ρ or with indefinite proximal term are preferred in practical implementation". Consequently, a number of researchers focused their attention on constructing different forms of indefinite proximal matrices. Based on the so-called Peaceman-Rachford splitting method [25], Gao and Ma [15] proposed and analyzed the convergence of the following positive-indefinite proximal symmetric ADMM (PID-SADMM):

$$\begin{cases} \mathbf{x}^{k+1} = \arg\min\limits_{\mathbf{x}\in\mathcal{X}} \mathcal{L}_\beta(\mathbf{x}, \mathbf{y}^k, \boldsymbol{\lambda}^k), \\ \boldsymbol{\lambda}^{k+\frac{1}{2}} = \boldsymbol{\lambda}^k - \alpha\beta\left(A\mathbf{x}^{k+1} + B\mathbf{y}^k - \mathbf{b}\right), \\ \mathbf{y}^{k+1} = \arg\min\limits_{\mathbf{y}\in\mathcal{Y}} \mathcal{L}_\beta(\mathbf{x}^{k+1}, \mathbf{y}, \boldsymbol{\lambda}^{k+\frac{1}{2}}) + \dfrac{1}{2}\left\|\mathbf{y}-\mathbf{y}^k\right\|_{\mathcal{P}}^2, \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^{k+\frac{1}{2}} - \beta\left(A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{b}\right), \end{cases}$$

where $\mathcal{P}$ reads the form of (1.6) but with $\mathcal{P}_1 = \tau\rho\mathbf{I}$ and $\tau \in \left[\frac{\alpha^2-\alpha+4}{\alpha^2-2\alpha+5}, 1\right)$ for any $\alpha \in (-1, 1)$. To relax the region of τ to accelerate the convergence of PID-SADMM numerically, Jiang et al. [21] reduced the lower bound of τ to $\frac{3+\alpha}{4}$. Later, by applying the relaxation iteration $\mathbf{u}^{k+1}$ in G-ADMM to PID-SADMM, Deng-Liu [10] constructed a generalized Peaceman-Rachford splitting method with substitution procedure, and they kept choosing $\mathcal{P}_1 = \tau\rho\mathbf{I}$ but $\tau \in (\frac{2+\alpha+s}{4}, 1)$, where $s \in (0, 2)$ is a relaxation factor satisfying $\alpha + s \in (0, 2)$. If ignoring both the indefinite proximal term and the substitution procedure, then the algorithm in [10] is equivalent to G-ADMM and SADMM. The aforementioned indefinite proximal matrix was also employed in the linearized symmetric ADMM [8] for solving multi-block separable convex optimization problems. Interested readers may refer to e.g. [9, 17, 18, 24, 26] for a comprehensive survey on using indefinite proximal term. Notice that the above types of indefinite proximal matrix are related to the stepsize of the dual variable, however, a new choice of indefinite proximal matrix with $\tau \in [-1, 1]$ is only related[1] to the set $\mathcal{Y}$ [3,

[1] More precisely, the assumption $\sup\{\|\mathbf{y}_1 - \mathbf{y}_2\| : \mathbf{y}_1, \mathbf{y}_2 \in \mathcal{Y}\} < \infty$ is needed.

Remark 4.1]. Since f in (1.2) is the average of N (big data) continuously differentiable functions, computing its full gradient at each iteration may be prohibitive, but its stochastic gradient can be exploited to develop a randomized gradient-type method to make the $\mathbf{x}$-subproblem being tackled inexactly and quickly.

Based on the above discussions and observations, we will develop an Accelerated Stochastic Peaceman-Rachford Splitting Method (**AS-PRSM**) for solving the structural empirical risk minimization problem (1.1). Our algorithm is similar to PID-SADMM but its $\mathbf{x}$-subproblem is solved by an accelerated stochastic gradient method, and both $\mathbf{y}$-subproblem and the dual variable employ a relaxation step. Especially, AS-PRSM can be regarded as a stochastic version of PID-SADMM when taking $s = 1$, and AS-PRSM is a multistep version of PID-SADMM if further $m_k = 1$. Notice that in Algorithm 2.1 the matrix $\mathcal{P}$ is not always positive definite because $\tau \in [\frac{2+\rho}{4}, 1)$ for $\rho \in (0, 2)$. To the best of our knowledge, it is perhaps the first time that a proximal indefinite term is exploited in a stochastic PRSM with convergence guaranteed.

In Section 2, we describe the proposed algorithm and give some preliminaries, including two assumptions on the problem and the variational characterization for both the saddle-point of the problem and the sequence generated by AS-PRSM respectively. Section 3 analyzes the convergence of AS-PRSM in the sense of expectation. Section 4 investigates the numerical performance of AS-PRSM by testing two popular examples from machine learning and medical image processing. Finally, we conclude the paper in Section 5. Some standard notations used throughout the context are collected at the end of this section. The bold $\mathbf{0}$ and $\mathbf{I}$ denote the zero matrix and the identity matrix with proper dimensions, respectively. For any symmetric matrices A, B having the same dimensions, we use $A \succeq (\succ) B$ to denote that $A - B$ is a positive semidefinite (definite) matrix. Without specific instructions, $\|\cdot\|$ is the Euclidean norm equipped with inner product $\langle\cdot,\cdot\rangle$, $\nabla f(\mathbf{x})$ denotes the gradient of f at $\mathbf{x}$, and $\mathbb{E}[\cdot]$ represents the mathematical expectation of a random variable. For convenience of analysis, let us denote

$$\mathbf{w} = \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \boldsymbol{\lambda} \end{pmatrix}, \qquad \mathcal{J}(\mathbf{w}) = \begin{pmatrix} -A^{\mathsf{T}}\boldsymbol{\lambda} \\ -B^{\mathsf{T}}\boldsymbol{\lambda} \\ A\mathbf{x} + B\mathbf{y} - \mathbf{b} \end{pmatrix}, \tag{1.7}$$

and

$$\mathbf{w}^k = \begin{pmatrix} \mathbf{u}^k \\ \boldsymbol{\lambda}^k \end{pmatrix} = \begin{pmatrix} \mathbf{x}^k \\ \mathbf{y}^k \\ \boldsymbol{\lambda}^k \end{pmatrix}, \quad \mathcal{J}(\mathbf{w}^k) = \begin{pmatrix} -A^{\mathsf{T}}\boldsymbol{\lambda}^k \\ -B^{\mathsf{T}}\boldsymbol{\lambda}^k \\ A\mathbf{x}^k + B\mathbf{y}^k - \mathbf{b} \end{pmatrix}. \tag{1.8}$$

We also define $F(\mathbf{u}) := f(\mathbf{x}) + g(\mathbf{y})$.

2. Algorithm and Preliminaries. Our proposed stochastic-type algorithm with the indefinite proximal term is described as the following Algorithm 2.1, whose main features are summarized as three aspects:

(i) The $\mathbf{x}$-subproblem is solved stochastically by the **xsub** routine, which enjoys the so-called momentum acceleration technique and allows the variance reduction technique to estimate $\mathbf{d}_t$. Users have the flexibility of choosing a zero mean random vector $\mathbf{e}_t$ to reduce the variance of a stochastic gradient. If the $\mathbf{x}$-subproblem is solved exactly: $\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \mathcal{L}_\beta(\mathbf{x}, \mathbf{y}^k, \boldsymbol{\lambda}^k)$ with $\alpha = 0$ and $\mathcal{P} = \mathbf{0}$, then Algorithm 2.1 reduces to G-ADMM (1.5), and moreover it reduces to the ADMM (1.4) when $s = 1$. If the deterministic gradient is used

Alg. 2.1: AS-PRSM with indefinite proximal term for solving the problem (1.1)

Parameters: $\beta > 0, \alpha \in (-1,1), s \in (0,2), \rho := \alpha + s \in (0,2), \mathcal{H} \succ \mathbf{0}$, $\mathcal{P} = \sigma\tau\mathbf{I} - \beta B^\mathsf{T}B$ `with` $\sigma > \beta\|B^\mathsf{T}B\|$ `and` $\tau \in [\frac{\rho+2}{4}, 1)$.
Initialization: $(\mathbf{x}^0, \mathbf{y}^0, \boldsymbol{\lambda}^0) \in \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^n, \breve{\mathbf{x}}^0 = \mathbf{x}^0$.
`For` $k = 0, 1, \ldots$
`Choose` $m_k > 0, \eta_k > 0$ `and` $\mathcal{M}_k$ `such that` $\mathcal{M}_k - \beta A^\mathsf{T}A \succeq \mathbf{0}$.
$\mathbf{h}^k := -A^\mathsf{T}\left[\boldsymbol{\lambda}^k - \beta(A\mathbf{x}^k + B\mathbf{y}^k - \mathbf{b})\right]$.
$(\mathbf{x}^{k+1}, \breve{\mathbf{x}}^{k+1}) = \mathbf{xsub}(\mathbf{x}^k, \breve{\mathbf{x}}^k, \mathbf{h}^k)$.
$\boldsymbol{\lambda}^{k+\frac{1}{2}} = \boldsymbol{\lambda}^k - \alpha\beta\left(A\mathbf{x}^{k+1} + B\mathbf{y}^k - \mathbf{b}\right)$.
$\mathbf{y}^{k+1} \in \arg\min_{\mathbf{y}\in\mathcal{Y}} \left\{ \begin{array}{l} g(\mathbf{y}) - \langle \boldsymbol{\lambda}^{k+\frac{1}{2}}, B\mathbf{y}\rangle + \frac{1}{2}\left\|\mathbf{y} - \mathbf{y}^k\right\|_{\mathcal{P}}^2 \\ +\frac{\beta}{2}\left\|sA\mathbf{x}^{k+1} + (1-s)(\mathbf{b} - B\mathbf{y}^k) + B\mathbf{y} - \mathbf{b}\right\|^2 \end{array} \right\}$.
$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^{k+\frac{1}{2}} - \beta\left[sA\mathbf{x}^{k+1} + (1-s)(\mathbf{b} - B\mathbf{y}^k) + B\mathbf{y}^{k+1} - \mathbf{b}\right]$.
`end`

$(\mathbf{x}^+, \breve{\mathbf{x}}^+) = \mathbf{xsub}(\mathbf{x}_1, \breve{\mathbf{x}}_1, \mathbf{h})$.
`For` $t = 1, 2, \ldots, m_k$
`Randomly select` $\xi_t \in \{1, 2, \ldots, N\}$ `with uniform probability.`
$\beta_t = 2/(t+1)$, $\gamma_t = 2/(t\eta_k)$, $\widehat{\mathbf{x}}_t = \beta_t\breve{\mathbf{x}}_t + (1-\beta_t)\mathbf{x}_t$.
$\mathbf{d}_t = \widehat{\mathbf{g}}_t + \mathbf{e}_t$, `where` $\widehat{\mathbf{g}}_t = \nabla f_{\xi_t}(\widehat{\mathbf{x}}_t)$ `and` $\mathbf{e}_t$ `is a random vector satisfying` $\mathbb{E}[\mathbf{e}_t] = \mathbf{0}$.
$\breve{\mathbf{x}}_{t+1} = \arg\min\left\{\langle \mathbf{d}_t + \mathbf{h}, \mathbf{x}\rangle + \frac{\gamma_t}{2}\|\mathbf{x} - \breve{\mathbf{x}}_t\|_{\mathcal{H}}^2 + \frac{1}{2}\left\|\mathbf{x} - \mathbf{x}^k\right\|_{\mathcal{M}_k}^2 : \mathbf{x} \in \mathcal{X}\right\}$.
$\mathbf{x}_{t+1} = \beta_t\breve{\mathbf{x}}_{t+1} + (1-\beta_t)\mathbf{x}_t$.
`end`
`Return` $(\mathbf{x}^+, \breve{\mathbf{x}}^+) = (\mathbf{x}_{m_k+1}, \breve{\mathbf{x}}_{m_k+1})$.

---

in stead of a stochastic gradient, then Algorithm 2.1 is a multi-step linearized ADMM. These discussions indicate that Algorithm 2.1 is more general and flexible than some of previous. Both $\mathcal{M}_k$ and $\mathcal{H}$ could be chosen as multiples of the identity matrix as explained in [5, Remark 4.2] and the experiments therein, so the $\breve{\mathbf{x}}_{t+1}$-subproblem is equivalent to a projection onto $\mathcal{X}$. This will make it much easier to be solved than the exact subproblem.

(ii) The $\mathbf{y}$-subproblem is solved inexactly and admits an indefinite proximal term to annihilate the involved quadratic term $\frac{\beta}{2}\|B\mathbf{y}\|^2$, so that the subproblem itself can be transformed into

$$\mathbf{y}^{k+1} = \mathbf{prox}_g^{\chi}(\bar{c}_y^k) := \arg\min_{\mathbf{y}\in\mathcal{Y}} g(\mathbf{y}) + \frac{\chi}{2}\left\|\mathbf{y} - \bar{c}_y^k\right\|^2,$$

where

$$\bar{c}_y^k = \mathbf{y}^k + \frac{B^\mathsf{T}\left[\boldsymbol{\lambda}^{k+\frac{1}{2}} - s\beta(A\mathbf{x}^{k+1} + B\mathbf{y}^k - \mathbf{b})\right]}{\chi} \quad \text{and} \quad \chi = \sigma\tau.$$

The proximity operator $\mathbf{prox}_g^{\chi}(\bar{c}_y^k)$ is well-defined since the involved objective function is strongly convex, which together with the convexity of $\mathcal{Y}$ indicates that there exists a uniquely global solution for the $\mathbf{y}$-subproblem. This fact

will directly annihilate the extra assumption (a2) in [5]. To the best of our knowledge, most of researchers focused on stochastic splitting methods allowing positive semidefinite proximal term, and there are few kinds of references on the stochastic PRSM using positive indefinite proximal term. Then, we fill this gap for theoretical interests.

(iii) Compared with some deterministic ADMM-type methods in e.g. [2, 8, 11, 12, 15, 19], convergence of Algorithm 2.1 does not need the full column/row assumption on both A and B. Moreover, by properly selecting the algorithm parameters, we establish the sublinear convergence rate of Algorithm 2.1 in the sense of expectation and ergodic. AS-PRSM will reduce to a linearized version of SADMM when $m_k = 1, N = 1$, while it reduces to a multi-step deterministic inexact SADMM when $m_k > 1, N = 1$. That's, the convergence results developed in this paper are applicable to these two cases. In addition, comparison experiments on testing practical examples show that Algorithm 2.1 performs slightly better than several existing algorithms to some extent.

Throughout the context, we make the following two assumptions where Assumption 2.2 will reduce to the traditional Lipschitz condition when $\mathcal{H}$ is the identity matrix.

ASSUMPTION 2.1. *The primal-dual solution set Ω^* of Problem (1.1) is nonempty.*

ASSUMPTION 2.2. *(Blockwise Lipschitzian) For any $\mathcal{H} \succ \mathbf{0}$, there exists a constant $\nu > 0$ such that the gradients ∇f_j satisfy the general Lipschitz condition*

$$\|\nabla f_j(\mathbf{x}_1) - \nabla f_j(\mathbf{x}_2)\|_{\mathcal{H}^{-1}} \leq \nu \|\mathbf{x}_1 - \mathbf{x}_2\|_{\mathcal{H}}$$

for every $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ and $j = 1, 2, \cdots, N$.

It is well-known that a point

$$\mathbf{w}^* := \begin{pmatrix} \mathbf{u}^* \\ \boldsymbol{\lambda}^* \end{pmatrix} = \begin{pmatrix} \mathbf{x}^* \\ \mathbf{y}^* \\ \boldsymbol{\lambda}^* \end{pmatrix} \in \Omega := \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^n$$

is called the saddle-point of $\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda})$ if the following inequalities

$$\mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}) \leq \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*) \leq \mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}^*)$$

hold for any $\mathbf{w} \in \Omega$, that is,

$$\left\{ \begin{array}{lcll} f(\mathbf{x}) - f(\mathbf{x}^*) & + & \left\langle \mathbf{x} - \mathbf{x}^*, -A^{\mathsf{T}}\boldsymbol{\lambda}^* \right\rangle & \geq & 0, \\ g(\mathbf{y}) - g(\mathbf{y}^*) & + & \left\langle \mathbf{y} - \mathbf{y}^*, -B^{\mathsf{T}}\boldsymbol{\lambda}^* \right\rangle & \geq & 0, \\ & & A\mathbf{x}^* + B\mathbf{y}^* - \mathbf{b} & = & \mathbf{0}. \end{array} \right.$$

By the previous notations, these inequalities can be rewritten as

$$F(\mathbf{u}) - F(\mathbf{u}^*) + \left\langle \mathbf{w} - \mathbf{w}^*, \mathcal{J}(\mathbf{w}^*) \right\rangle \geq 0, \quad \forall \mathbf{w} \in \Omega. \tag{2.1}$$

Since the affine mapping $\mathcal{J}(\cdot)$ defined in (1.7) is skew-symmetric,

$$\langle \mathbf{w} - \overline{\mathbf{w}}, \mathcal{J}(\mathbf{w}) - \mathcal{J}(\overline{\mathbf{w}}) \rangle \equiv 0, \quad \forall \mathbf{w}, \overline{\mathbf{w}} \in \Omega. \tag{2.2}$$

Consequently, (2.1) amounts to

$$F(\mathbf{u}) - F(\mathbf{u}^*) + \left\langle \mathbf{w} - \mathbf{w}^*, \mathcal{J}(\mathbf{w}) \right\rangle \geq 0, \quad \forall \mathbf{w} \in \Omega. \tag{2.3}$$

Similar to the above variational characterization for the saddle-point of (1.1), we will show that the iterates generated by Algorithm 2.1 satisfy an analogous inequality to (2.3) with the aid of the following notations

$$\widetilde{\boldsymbol{\lambda}}^k = \boldsymbol{\lambda}^k - \beta\left(A\mathbf{x}^{k+1} + B\mathbf{y}^k - \mathbf{b}\right), \quad \mathcal{D}_k = \mathcal{M}_k - \beta A^{\mathsf{T}}A, \tag{2.4}$$

$$\begin{aligned}\zeta^k = \frac{2}{m_k(m_k+1)}\Bigg[&\frac{1}{\eta_k}\left(\left\|\mathbf{x}-\breve{\mathbf{x}}^{k+1}\right\|_{\mathcal{H}}^2 - \left\|\mathbf{x}-\breve{\mathbf{x}}^{k}\right\|_{\mathcal{H}}^2\right) \\ &- \sum_{t=1}^{m_k} t\langle\boldsymbol{\delta}_t, \breve{\mathbf{x}}_t - \mathbf{x}\rangle - \frac{\eta_k}{4(1-\eta_k\nu)}\sum_{t=1}^{m_k} t^2\left\|\boldsymbol{\delta}_t\right\|_{\mathcal{H}^{-1}}^2\Bigg],\end{aligned} \tag{2.5}$$

where $\boldsymbol{\delta}_t = \nabla f(\widehat{\mathbf{x}}_t) - \mathbf{d}_t$ and $\mathbf{d}_t$ is given by the **xsub** routine of Algorithm 2.1.

LEMMA 2.1. *Suppose $\eta_k \in (0, \frac{1}{\nu})$ and Assumption 2.2 holds. Then, the iterates generated by Algorithm 2.1 satisfy $\widetilde{\mathbf{w}}^k \in \Omega$ and*

$$F(\mathbf{u}) - F(\widetilde{\mathbf{u}}^k) + \left\langle \mathbf{w} - \widetilde{\mathbf{w}}^k, \mathcal{J}(\mathbf{w})\right\rangle \geq \left\langle \mathbf{w} - \widetilde{\mathbf{w}}^k, H_k(\mathbf{w}^k - \mathbf{w}^{k+1})\right\rangle + \zeta^k \tag{2.6}$$

for all $\mathbf{w} \in \Omega$, where $\widetilde{\boldsymbol{\lambda}}^k$ and ζ^k are given by (2.4) and (2.5) respectively,

$$\widetilde{\mathbf{w}}^k = \begin{pmatrix} \widetilde{\mathbf{u}}^k \\ \widetilde{\boldsymbol{\lambda}}^k \end{pmatrix} = \begin{pmatrix} \widetilde{\mathbf{x}}^k \\ \widetilde{\mathbf{y}}^k \\ \widetilde{\boldsymbol{\lambda}}^k \end{pmatrix} = \begin{pmatrix} \mathbf{x}^{k+1} \\ \mathbf{y}^{k+1} \\ \widetilde{\boldsymbol{\lambda}}^k \end{pmatrix}, \; H_k = \begin{bmatrix} \mathcal{D}_k & & \\ & \chi\mathbf{I} + \frac{1-\rho}{\rho}\beta B^{\mathsf{T}}B & \frac{1-\rho}{\rho}B^{\mathsf{T}} \\ & \frac{1-\rho}{\rho}B & \frac{1}{\rho\beta}\mathbf{I} \end{bmatrix}. \tag{2.7}$$

Proof. Followed by the proof of [5, Lemma 3.2], we have by the **xsub** routine that $\mathbf{x}^{k+1} \in \mathcal{X}$ and

$$f(\mathbf{x}) - f(\mathbf{x}^{k+1}) + \left\langle \mathbf{x} - \mathbf{x}^{k+1}, -A^{\mathsf{T}}\widetilde{\boldsymbol{\lambda}}^k\right\rangle \geq \left\langle \mathbf{x} - \mathbf{x}^{k+1}, \mathcal{D}_k(\mathbf{x}^k - \mathbf{x}^{k+1})\right\rangle + \zeta^k \tag{2.8}$$

for all $\mathbf{x} \in \mathcal{X}$, where $\widetilde{\boldsymbol{\lambda}}^k$ and ζ^k are given by (2.4) and (2.5) respectively.

In addition, it follows from the optimality condition of $\mathbf{y}$-subproblem that

$$\mathbf{y}^{k+1} \in \mathcal{Y}, \quad g(\mathbf{y}) - g(\mathbf{y}^{k+1}) + \left\langle \mathbf{y} - \mathbf{y}^{k+1}, c_y^k\right\rangle \geq 0, \quad \forall \mathbf{y} \in \mathcal{Y}, \tag{2.9}$$

where

$$\begin{aligned} c_y^k &= -B^{\mathsf{T}}\boldsymbol{\lambda}^{k+\frac{1}{2}} + \beta B^{\mathsf{T}}\left[s(A\mathbf{x}^{k+1} + B\mathbf{y}^k - \mathbf{b}) + B(\mathbf{y}^{k+1} - \mathbf{y}^k)\right] + \mathcal{P}(\mathbf{y}^{k+1} - \mathbf{y}^k) \\ &= -B^{\mathsf{T}}\boldsymbol{\lambda}^{k+\frac{1}{2}} + sB^{\mathsf{T}}\left(\boldsymbol{\lambda}^k - \widetilde{\boldsymbol{\lambda}}^k\right) + \left[\mathcal{P} + \beta B^{\mathsf{T}}B\right]\left(\mathbf{y}^{k+1} - \mathbf{y}^k\right) \\ &= -B^{\mathsf{T}}\widetilde{\boldsymbol{\lambda}}^k + (\rho-1)B^{\mathsf{T}}\left(\boldsymbol{\lambda}^k - \widetilde{\boldsymbol{\lambda}}^k\right) + \chi\left(\mathbf{y}^{k+1} - \mathbf{y}^k\right), \end{aligned} \tag{2.10}$$

from which the last equality uses the notations $\rho := \alpha + s, \chi = \sigma\tau$ and

$$\boldsymbol{\lambda}^{k+\frac{1}{2}} = \boldsymbol{\lambda}^k - \alpha(\boldsymbol{\lambda}^k - \widetilde{\boldsymbol{\lambda}}^k) = \widetilde{\boldsymbol{\lambda}}^k + (\alpha-1)(\widetilde{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^k). \tag{2.11}$$

The notation of $\widetilde{\boldsymbol{\lambda}}^k$ gives

$$\frac{1}{\beta}(\boldsymbol{\lambda}^k - \widetilde{\boldsymbol{\lambda}}^k) = \left(A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{b}\right) + B(\mathbf{y}^k - \mathbf{y}^{k+1}) \tag{2.12}$$

that is equivalent to

$$\left\langle \boldsymbol{\lambda} - \widetilde{\boldsymbol{\lambda}}^k, Ax^{k+1} + By^{k+1} - \mathbf{b} \right\rangle = \left\langle \boldsymbol{\lambda} - \widetilde{\boldsymbol{\lambda}}^k, -B\left(\mathbf{y}^k - \mathbf{y}^{k+1}\right) + \frac{\boldsymbol{\lambda}^k - \widetilde{\boldsymbol{\lambda}}^k}{\beta} \right\rangle \quad (2.13)$$

for any $\boldsymbol{\lambda} \in \mathbb{R}^n$. Based on (2.2) and the notation $\widetilde{\mathbf{w}}^k$, combine the above inequalities (2.8), (2.9)-(2.10), (2.13) to obtain

$$F(\mathbf{u}) - F(\widetilde{\mathbf{u}}^k) + \left\langle \mathbf{w} - \widetilde{\mathbf{w}}^k, \mathcal{J}(\mathbf{w}) \right\rangle \geq \left\langle \mathbf{w} - \widetilde{\mathbf{w}}^k, Q_k(\mathbf{w}^k - \widetilde{\mathbf{w}}^k) \right\rangle + \zeta^k, \quad (2.14)$$

where

$$Q_k = \begin{bmatrix} \mathcal{D}_k & & \\ & \chi\mathbf{I} & (1-\rho)B^{\mathsf{T}} \\ & -B & \frac{1}{\beta}\mathbf{I} \end{bmatrix}.$$

By using the notations about $\widetilde{\boldsymbol{\lambda}}^k, \rho$ and the equality (2.11), we further have

$$\begin{aligned} \boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^{k+\frac{1}{2}} - \beta\left[s(Ax^{k+1} + By^k - \mathbf{b}) + B(\mathbf{y}^{k+1} - \mathbf{y}^k)\right] \\ &= \boldsymbol{\lambda}^k - \alpha(\boldsymbol{\lambda}^k - \widetilde{\boldsymbol{\lambda}}^k) - \left[s(\boldsymbol{\lambda}^k - \widetilde{\boldsymbol{\lambda}}^k) + \beta B(\widetilde{\mathbf{y}}^k - \mathbf{y}^k)\right] \\ &= \boldsymbol{\lambda}^k + \beta B(\mathbf{y}^k - \widetilde{\mathbf{y}}^k) - \rho(\boldsymbol{\lambda}^k - \widetilde{\boldsymbol{\lambda}}^k). \end{aligned}$$

Combine this equality with the notations about $\widetilde{\mathbf{x}}^k$ and $\widetilde{\mathbf{y}}^k$ to obtain

$$\mathbf{w}^{k+1} = \mathbf{w}^k - M(\mathbf{w}^k - \widetilde{\mathbf{w}}^k), \quad (2.15)$$

where

$$M = \begin{bmatrix} \mathbf{I} & & \\ & \mathbf{I} & \\ & -\beta B & \rho\mathbf{I} \end{bmatrix}.$$

Finally, the inequality (2.6) can be achieved by substituting (2.15) into (2.14) together with the relation $H_k = Q_k M^{-1}$ that is just the form in (2.7). □

3. Convergence Analysis. This section aims to analyze convergence of the proposed algorithm in detail. Firstly, by the region of σ and τ it holds

$$H_k \succeq \begin{bmatrix} \mathcal{D}_k & & \\ & (\frac{2+\rho}{4} + \frac{1-\rho}{\rho})\beta B^{\mathsf{T}}B & \frac{1-\rho}{\rho}B^{\mathsf{T}} \\ & \frac{1-\rho}{\rho}B & \frac{1}{\rho\beta}\mathbf{I} \end{bmatrix} := \overline{H}_k.$$

Observing that the matrix H_k is positive definite if $\overline{H}_k$ is positive definite. By the structure of $\overline{H}_k$, we only need to verify the positive definiteness of the following block matrix

$$\begin{bmatrix} (\frac{2+\rho}{4} + \frac{1-\rho}{\rho})\beta B^{\mathsf{T}}B & \frac{1-\rho}{\rho}B^{\mathsf{T}} \\ \frac{1-\rho}{\rho}B & \frac{1}{\rho\beta}\mathbf{I} \end{bmatrix} = \begin{bmatrix} \beta^{\frac{1}{2}}B & \\ & \beta^{-\frac{1}{2}}\mathbf{I} \end{bmatrix}^{\mathsf{T}} \overline{H}_k^L \begin{bmatrix} \beta^{\frac{1}{2}}B & \\ & \beta^{-\frac{1}{2}}\mathbf{I} \end{bmatrix},$$

where

$$\overline{H}_k^L = \begin{bmatrix} (\frac{2+\rho}{4} + \frac{1-\rho}{\rho})\mathbf{I} & \frac{1-\rho}{\rho}\mathbf{I} \\ \frac{1-\rho}{\rho}\mathbf{I} & \frac{1}{\rho}\mathbf{I} \end{bmatrix} = \frac{1}{\rho}\begin{bmatrix} (\frac{(2+\rho)\rho}{4} + 1 - \rho)\mathbf{I} & (1-\rho)\mathbf{I} \\ (1-\rho)\mathbf{I} & \mathbf{I} \end{bmatrix}$$

is positive definite if $\rho > 0$ and $\frac{2+\rho}{4} > \rho - 1$. Clearly, $\frac{2+\rho}{4} > \rho - 1$ is equivalent to $\rho < 2$, which holds obviously for the parameter restricted in Algorithm 2.1.

COROLLARY 3.1. *Suppose the conditions of Lemma 2.1 hold. Then, we have*

$$\begin{aligned} &F(\mathbf{u}) - F(\widetilde{\mathbf{u}}^k) + \left\langle \mathbf{w} - \widetilde{\mathbf{w}}^k, \mathcal{J}(\mathbf{w}) \right\rangle \\ &\geq \frac{1}{2}\left(\left\| \mathbf{w} - \mathbf{w}^{k+1} \right\|_{H_k}^2 - \left\| \mathbf{w} - \mathbf{w}^k \right\|_{H_k}^2 \right) + \frac{1}{2} \left\| \mathbf{w}^k - \widetilde{\mathbf{w}}^k \right\|_{G_k}^2 + \zeta^k \end{aligned} \tag{3.1}$$

for all $\mathbf{w} \in \Omega$, *where* ζ^k *is defined in (2.5) and* G_k *is given by (3.4).*

Proof. By the positive definiteness of H_k and the following identity

$$\left\langle a - b, H_k(c - d) \right\rangle = \frac{1}{2}\Big\{ \|a - d\|_{H_k}^2 - \|a - c\|_{H_k}^2 + \|c - b\|_{H_k}^2 - \|b - d\|_{H_k}^2 \Big\}$$

with specifications $a := \mathbf{w}, b := \widetilde{\mathbf{w}}^k, c := \mathbf{w}^k, d := \mathbf{w}^{k+1}$, we have

$$\begin{aligned} &F(\mathbf{u}) - F(\widetilde{\mathbf{u}}^k) + \left\langle \mathbf{w} - \widetilde{\mathbf{w}}^k, \mathcal{J}(\mathbf{w}) \right\rangle \geq \zeta^k + \frac{1}{2}\Big\{ \left\| \mathbf{w} - \mathbf{w}^{k+1} \right\|_{H_k}^2 \\ &- \left\| \mathbf{w} - \mathbf{w}^k \right\|_{H_k}^2 + \left\| \mathbf{w}^k - \widetilde{\mathbf{w}}^k \right\|_{H_k}^2 - \left\| \mathbf{w}^{k+1} - \widetilde{\mathbf{w}}^k \right\|_{H_k}^2 \Big\}. \end{aligned} \tag{3.2}$$

Here, it follows from the relationship in (2.15) that

$$\begin{aligned} &\left\| \mathbf{w}^k - \widetilde{\mathbf{w}}^k \right\|_{H_k}^2 - \left\| \mathbf{w}^{k+1} - \widetilde{\mathbf{w}}^k \right\|_{H_k}^2 \\ &= \left\| \mathbf{w}^k - \widetilde{\mathbf{w}}^k \right\|_{H_k}^2 - \left\| \mathbf{w}^k - \widetilde{\mathbf{w}}^k + \mathbf{w}^{k+1} - \mathbf{w}^k \right\|_{H_k}^2 \\ &= \left\| \mathbf{w}^k - \widetilde{\mathbf{w}}^k \right\|_{H_k}^2 - \left\| \mathbf{w}^k - \widetilde{\mathbf{w}}^k - M(\mathbf{w}^k - \widetilde{\mathbf{w}}^k) \right\|_{H_k}^2 \\ &= \left\| \mathbf{w}^k - \widetilde{\mathbf{w}}^k \right\|_{G_k}^2, \end{aligned} \tag{3.3}$$

where

$$\begin{aligned} G_k &= M^{\mathsf{T}} H_k + H_k M - M^{\mathsf{T}} H_k M \\ &= (Q_k)^{\mathsf{T}} + Q_k - M^{\mathsf{T}} Q_k = \begin{bmatrix} \mathcal{D}_k & & \\ & \mathcal{P} & \\ & & \frac{2-\rho}{\beta}\mathbf{I} \end{bmatrix}. \end{aligned} \tag{3.4}$$

Plugging (3.4) and (3.3) into (3.2), the proof is completed. □

Notice that the matrix G_k is not always positive definite due to the indefiniteness of $\mathcal{P}$. This makes it very challenging to establish the convergence of Algorithm 2.1 although the inequality (3.1) is similar to (2.3) with some additional terms. A feasible line to further build the convergence theories of Algorithm 2.1 is to estimate the lower bound of $\left\| \mathbf{w}^k - \widetilde{\mathbf{w}}^k \right\|_{G_k}^2$. First and foremost, for $\delta \in (0, 1)$, we have by the Cauchy-Schwarz inequality that

$$\begin{aligned} &\left(\mathbf{y}^k - \mathbf{y}^{k+1} \right)^{\mathsf{T}} B^{\mathsf{T}} (A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{b}) \\ &\geq -\frac{1}{4(1-\delta)} \left\| B(\mathbf{y}^k - \mathbf{y}^{k+1}) \right\|^2 - (1 - \delta) \left\| A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{b} \right\|^2. \end{aligned}$$

Let

$$\delta \in \left(0, \frac{4\mu - 1}{4\mu} \right) \subset (0, 1) \quad \text{with} \quad \mu > \frac{1}{4}. \tag{3.5}$$

Then, we have $\frac{1}{4(1-\delta)} < \frac{1}{4} + \mu\delta$ and consequently

$$\begin{aligned}&\left(\mathbf{y}^k - \mathbf{y}^{k+1}\right)^{\mathsf{T}} B^{\mathsf{T}}(A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{b})\\ \geq &-\left(\frac{1}{4} + \mu\delta\right)\left\|B(\mathbf{y}^k - \mathbf{y}^{k+1})\right\|^2 - (1-\delta)\left\|A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{b}\right\|^2. \end{aligned} \tag{3.6}$$

In addition, by the first-order optimality condition of the $\mathbf{y}$-subproblem, we can get another estimate on the term in the left-hand-side of (3.6):

LEMMA 3.2. *Let $\mathcal{P}_0 = \sigma\mathbf{I} - \beta B^{\mathsf{T}}B$ with $\sigma > \beta\|B^{\mathsf{T}}B\|$. Then, the iterates generated by Algorithm 2.1 satisfy*

$$\begin{aligned}&\left(\mathbf{y}^k - \mathbf{y}^{k+1}\right)^{\mathsf{T}} B^{\mathsf{T}}(A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{b}) \\ \geq &\frac{1-\tau}{2\rho}\left[\left\|B(\mathbf{y}^k - \mathbf{y}^{k+1})\right\|^2 - \left\|B(\mathbf{y}^k - \mathbf{y}^{k-1})\right\|^2\right]\\ &+\frac{\tau}{2\rho\beta}\left[\left\|\mathbf{y}^k - \mathbf{y}^{k+1}\right\|^2_{\mathcal{P}_0} - \left\|\mathbf{y}^k - \mathbf{y}^{k-1}\right\|^2_{\mathcal{P}_0}\right] + \frac{2(\tau-1)+1-\rho}{\rho}\left\|B(\mathbf{y}^k - \mathbf{y}^{k+1})\right\|^2. \end{aligned} \tag{3.7}$$

Proof. Followed by the inequality (2.9), it holds

$$g(\mathbf{y}) - g(\mathbf{y}^{k+1}) + \left(\mathbf{y} - \mathbf{y}^{k+1}\right)^{\mathsf{T}}\begin{bmatrix} -B^{\mathsf{T}}\boldsymbol{\lambda}^{k+\frac{1}{2}} + s\beta B^{\mathsf{T}}(A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{b}) \\ +(1-s)\beta B^{\mathsf{T}}B(\mathbf{y}^{k+1} - \mathbf{y}^k) + \mathcal{P}(\mathbf{y}^{k+1} - \mathbf{y}^k)\end{bmatrix} \geq 0 \tag{3.8}$$

and

$$g(\mathbf{y}) - g(\mathbf{y}^k) + \left(\mathbf{y} - \mathbf{y}^k\right)^{\mathsf{T}}\begin{bmatrix} -B^{\mathsf{T}}\boldsymbol{\lambda}^{k-\frac{1}{2}} + s\beta B^{\mathsf{T}}(A\mathbf{x}^k + B\mathbf{y}^k - \mathbf{b}) \\ +(1-s)\beta B^{\mathsf{T}}B(\mathbf{y}^k - \mathbf{y}^{k-1}) + \mathcal{P}(\mathbf{y}^k - \mathbf{y}^{k-1})\end{bmatrix} \geq 0 \tag{3.9}$$

for any $\mathbf{y} \in \mathcal{Y}$. Add the inequality (3.8) with $\mathbf{y} = \mathbf{y}^k$ to the inequality (3.9) with $\mathbf{y} = \mathbf{y}^{k+1}$ to obtain

$$\left(\mathbf{y}^k - \mathbf{y}^{k+1}\right)^{\mathsf{T}}\left\{\begin{array}{c} s\beta B^{\mathsf{T}}[(A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{b}) - (A\mathbf{x}^k + B\mathbf{y}^k - \mathbf{b})] \\ +(1-s)\beta B^{\mathsf{T}}B[(\mathbf{y}^{k+1} - \mathbf{y}^k) - (\mathbf{y}^k - \mathbf{y}^{k-1})] \\ +\mathcal{P}(\mathbf{y}^{k+1} - \mathbf{y}^k) - \mathcal{P}(\mathbf{y}^k - \mathbf{y}^{k-1}) + B^{\mathsf{T}}(\boldsymbol{\lambda}^{k-\frac{1}{2}} - \boldsymbol{\lambda}^{k+\frac{1}{2}})\end{array}\right\} \geq 0. \tag{3.10}$$

Notice that, the update of $\boldsymbol{\lambda}^{k+\frac{1}{2}}$ and the update of $\boldsymbol{\lambda}^{k+1}$ at previous iteration imply

$$\begin{aligned}&\boldsymbol{\lambda}^{k-\frac{1}{2}} - \boldsymbol{\lambda}^{k+\frac{1}{2}}\\ =\ &\alpha\beta\left(A\mathbf{x}^{k+1} + B\mathbf{y}^k - \mathbf{b}\right) + \beta\left[sA\mathbf{x}^k + (1-s)(\mathbf{b} - B\mathbf{y}^{k-1}) + B\mathbf{y}^k - \mathbf{b}\right]\\ =\ &\alpha\beta\left(A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{b}\right) + \alpha\beta B(\mathbf{y}^k - \mathbf{y}^{k+1})\\ &+\beta\left[s\left(A\mathbf{x}^k + B\mathbf{y}^k - \mathbf{b}\right) + (1-s)B(\mathbf{y}^k - \mathbf{y}^{k-1})\right].\end{aligned}$$

Substitute it into (3.10) with $\rho = \alpha + s$ and simple calculations to obtain

$$\left(\mathbf{y}^k - \mathbf{y}^{k+1}\right)^{\mathsf{T}}\left\{\begin{array}{c} \mathcal{P}(\mathbf{y}^{k+1} - \mathbf{y}^k) - \mathcal{P}(\mathbf{y}^k - \mathbf{y}^{k-1}) \\ +\rho\beta B^{\mathsf{T}}(A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{b}) + (1-\rho)\beta B^{\mathsf{T}}B(\mathbf{y}^{k+1} - \mathbf{y}^k)\end{array}\right\} \geq 0,$$

which, by the following certain relation

$$\mathcal{P} = \tau\mathcal{P}_0 - (1-\tau)\beta B^{\mathsf{T}}B, \tag{3.11}$$

implies

$$
\begin{aligned}
&\left(\mathbf{y}^k-\mathbf{y}^{k+1}\right)^{\mathsf{T}} B^{\mathsf{T}}(A\mathbf{x}^{k+1}+B\mathbf{y}^{k+1}-\mathbf{b})\\
\geq&\frac{1}{\rho\beta}\left(\mathbf{y}^k-\mathbf{y}^{k+1}\right)^{\mathsf{T}}\mathcal{P}[(\mathbf{y}^k-\mathbf{y}^{k+1})+(\mathbf{y}^k-\mathbf{y}^{k-1})]+\frac{1-\rho}{\rho}\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2\\
\geq&\frac{1}{\rho\beta}\left[\tau\left\|\mathbf{y}^k-\mathbf{y}^{k+1}\right\|_{\mathcal{P}_0}^2-(1-\tau)\beta\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2\right]\\
&-\frac{\tau}{2\rho\beta}\left[\left\|\mathbf{y}^k-\mathbf{y}^{k+1}\right\|_{\mathcal{P}_0}^2+\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|_{\mathcal{P}_0}^2\right]+\frac{1-\rho}{\rho}\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2\\
&-\frac{1-\tau}{2\rho}\left[\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2+\left\|B(\mathbf{y}^k-\mathbf{y}^{k-1})\right\|^2\right]\\
=&\frac{\tau}{2\rho\beta}\left[\left\|\mathbf{y}^k-\mathbf{y}^{k+1}\right\|_{\mathcal{P}_0}^2-\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|_{\mathcal{P}_0}^2\right]-\frac{2(1-\tau)+\rho-1}{\rho}\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2\\
&+\frac{1-\tau}{2\rho}\left[\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2-\left\|B(\mathbf{y}^k-\mathbf{y}^{k-1})\right\|^2\right].
\end{aligned}
$$

This completes the proof. □

In fact, for any $\tau\in[\frac{2+\rho}{4},1)$ it holds

$$
\begin{aligned}
\tau&\geq\frac{(2+\rho)(4-\rho)}{4(4-\rho)}=\frac{8+2\rho+\rho^2}{4(4-\rho)}=\frac{6\rho-8-\rho^2}{4(4-\rho)}+1\\
&=\frac{1}{4-\rho}\left(\frac{\rho(2-\rho)}{4}+\rho-2\right)+1,
\end{aligned}
$$

which implies

$$
\frac{2-\rho+(\tau-1)(4-\rho)}{\rho(2-\rho)}\geq\frac{1}{4}\quad\text{for any }\rho\in(0,2). \tag{3.12}
$$

Compared the last inequality to that in (3.5), we have the following estimation on the lower bound of $\left\|\mathbf{w}^k-\widetilde{\mathbf{w}}^k\right\|_{G_k}^2$.

LEMMA 3.3. *For any $\mu>\frac{2-\rho+(\tau-1)(4-\rho)}{\rho(2-\rho)}$, there exists a $\delta_0\in\left(0,\frac{4\mu-1}{4\mu}\right)$ such that the iterates generated by Algorithm 2.1 satisfy*

$$
\begin{aligned}
\left\|\mathbf{w}^k-\widetilde{\mathbf{w}}^k\right\|_{G_k}^2\geq&\left\|\mathbf{x}^k-\mathbf{x}^{k+1}\right\|_{\mathcal{D}_k}^2+\tau\left\|\mathbf{y}^k-\mathbf{y}^{k+1}\right\|_{\mathcal{P}_0}^2\\
&+\frac{2-\rho}{2\rho}\left[\tau\left\|\mathbf{y}^k-\mathbf{y}^{k+1}\right\|_{\mathcal{P}_0}^2+(1-\tau)\beta\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2\right]\\
&-\frac{2-\rho}{2\rho}\left[\tau\left\|\mathbf{y}^{k-1}-\mathbf{y}^k\right\|_{\mathcal{P}_0}^2+(1-\tau)\beta\left\|B(\mathbf{y}^{k-1}-\mathbf{y}^k)\right\|^2\right]\\
&+(2-\rho)\beta\left(c_0\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2+\delta_0\left\|A\mathbf{x}^{k+1}+B\mathbf{y}^{k+1}-\mathbf{b}\right\|^2\right),
\end{aligned}
\tag{3.13}
$$

where $c_0=\frac{1}{\rho}+\frac{(\tau-1)(4-\rho)}{\rho(2-\rho)}-\mu\delta-\frac{1}{4}\geq 0$.

Proof. Add the inequality (3.6) to (3.7) to obtain

$$
\begin{aligned}
&2\left(\mathbf{y}^k-\mathbf{y}^{k+1}\right)^{\mathsf{T}} B^{\mathsf{T}}(A\mathbf{x}^{k+1}+B\mathbf{y}^{k+1}-\mathbf{b}) \qquad (3.14)\\
\geq\ &\frac{1-\tau}{2\rho}\left[\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2-\left\|B(\mathbf{y}^k-\mathbf{y}^{k-1})\right\|^2\right]\\
&+\frac{\tau}{2\rho\beta}\left[\left\|\mathbf{y}^k-\mathbf{y}^{k+1}\right\|_{\mathcal{P}_0}^2-\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|_{\mathcal{P}_0}^2\right]\\
&-\left(\frac{2(1-\tau)-1}{\rho}+\mu\delta+\frac{5}{4}\right)\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2-(1-\delta)\left\|A\mathbf{x}^{k+1}+B\mathbf{y}^{k+1}-\mathbf{b}\right\|^2.
\end{aligned}
$$

Together with the previous (2.12) and (3.14), it holds by the block structure of G_k and the relation in (3.11) that

$$
\begin{aligned}
&\left\|\mathbf{w}^k-\widetilde{\mathbf{w}}^k\right\|_{G_k}^2 \qquad (3.15)\\
=\ &\left\|\mathbf{x}^k-\mathbf{x}^{k+1}\right\|_{\mathcal{D}_k}^2+\tau\left\|\mathbf{y}^k-\mathbf{y}^{k+1}\right\|_{\mathcal{P}_0}^2-(1-\tau)\beta\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2\\
&+(2-\rho)\beta\left\|\frac{1}{\beta}(\boldsymbol{\lambda}^k-\widetilde{\boldsymbol{\lambda}}^k)\right\|^2\\
=\ &\left\|\mathbf{x}^k-\mathbf{x}^{k+1}\right\|_{\mathcal{D}_k}^2+\tau\left\|\mathbf{y}^k-\mathbf{y}^{k+1}\right\|_{\mathcal{P}_0}^2+(1-\rho+\tau)\beta\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2\\
&+(2-\rho)\beta\left[\left\|A\mathbf{x}^{k+1}+B\mathbf{y}^{k+1}-\mathbf{b}\right\|^2+2\left(\mathbf{y}^k-\mathbf{y}^{k+1}\right)^{\mathsf{T}}B^{\mathsf{T}}(A\mathbf{x}^{k+1}+B\mathbf{y}^{k+1}-\mathbf{b})\right]\\
\geq\ &\left\|\mathbf{x}^k-\mathbf{x}^{k+1}\right\|_{\mathcal{D}_k}^2+\tau\left\|\mathbf{y}^k-\mathbf{y}^{k+1}\right\|_{\mathcal{P}_0}^2+(1-\rho+\tau)\beta\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2\\
&+(2-\rho)\beta\left\|A\mathbf{x}^{k+1}+B\mathbf{y}^{k+1}-\mathbf{b}\right\|^2+\frac{(2-\rho)\tau}{2\rho}\left[\left\|\mathbf{y}^k-\mathbf{y}^{k+1}\right\|_{\mathcal{P}_0}^2-\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|_{\mathcal{P}_0}^2\right]\\
&+\frac{(2-\rho)(1-\tau)\beta}{2\rho}\left[\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2-\left\|B(\mathbf{y}^k-\mathbf{y}^{k-1})\right\|^2\right]-(2-\rho)\beta\\
&\times\left[\left(\frac{2(1-\tau)-1}{\rho}+\mu\delta+\frac{5}{4}\right)\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2+(1-\delta)\left\|A\mathbf{x}^{k+1}+B\mathbf{y}^{k+1}-\mathbf{b}\right\|^2\right]\\
=\ &\left\|\mathbf{x}^k-\mathbf{x}^{k+1}\right\|_{\mathcal{D}_k}^2+\tau\left\|\mathbf{y}^k-\mathbf{y}^{k+1}\right\|_{\mathcal{P}_0}^2\\
&+\frac{2-\rho}{2\rho}\left[\tau\left\|\mathbf{y}^k-\mathbf{y}^{k+1}\right\|_{\mathcal{P}_0}^2+(1-\tau)\beta\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2\right]\\
&-\frac{2-\rho}{2\rho}\left[\tau\left\|\mathbf{y}^{k-1}-\mathbf{y}^k\right\|_{\mathcal{P}_0}^2+(1-\tau)\beta\left\|B(\mathbf{y}^{k-1}-\mathbf{y}^k)\right\|^2\right]\\
&+(2-\rho)\beta\left(c_0\left\|B(\mathbf{y}^k-\mathbf{y}^{k+1})\right\|^2+\delta\left\|A\mathbf{x}^{k+1}+B\mathbf{y}^{k+1}-\mathbf{b}\right\|^2\right)
\end{aligned}
$$

where $c_0=\frac{1}{\rho}+\frac{(\tau-1)(4-\rho)}{\rho(2-\rho)}-\mu\delta-\frac{1}{4}$ and clearly

$$
c_0\geq 0\Leftrightarrow\delta\leq\frac{1}{\mu}\left(\frac{1}{\rho}+\frac{(\tau-1)(4-\rho)}{\rho(2-\rho)}-\frac{1}{4}\right). \qquad (3.16)
$$

Notice that for any $\mu>\frac{2-\rho+(\tau-1)(4-\rho)}{\rho(2-\rho)}$(by (3.12) $\mu\geq\frac{1}{4}$) with $\tau\in[\frac{2+\rho}{4},1)$ we have

$$
\mu>\frac{1}{\rho}+\frac{(\tau-1)(4-\rho)}{\rho(2-\rho)}\Rightarrow\frac{1}{\mu}\left(\frac{1}{\rho}+\frac{(\tau-1)(4-\rho)}{\rho(2-\rho)}-\frac{1}{4}\right)<\frac{4\mu-1}{4\mu},
$$

which by (3.16) shows that δ_0 exists. This completes the proof. □

The following theorem establishes a basic convergence result of Algorithm 2.1.

THEOREM 3.4. *For any integers $\kappa, T > 0$, let*

$$\mathbf{w}_T = \frac{1}{T}\sum_{k=\kappa}^{\kappa+T} \widetilde{\mathbf{w}}^k \quad \textit{and} \quad \mathbf{u}_T = \frac{1}{T}\sum_{k=\kappa}^{\kappa+T} \widetilde{\mathbf{u}}^k.$$

Suppose that Assumptions 2.1-2.2, and the following conditions hold for all $k \in [\kappa, \kappa+T]$: (A1) $\eta_k \in (0, \frac{1}{2\nu}]$ and the sequence $\{\eta_k m_k(m_k+1)\}$ is nondecreasing; (A2) $\mathcal{D}_k \succeq \mathcal{D}_{k+1} \succeq \mathbf{0}$ and $\mathbb{E}(\|\boldsymbol{\delta}_t\|^2_{\mathcal{H}^{-1}}) \le c^2$ for some $c > 0$. Then, for any $\mathbf{w} \in \Omega$ we have

$$\begin{aligned} &\mathbb{E}\left[F(\mathbf{u}_T) - F(\mathbf{u}) + \langle \mathbf{w}_T - \mathbf{w}, \mathcal{J}(\mathbf{w})\rangle\right] \\ &\le \frac{1}{2T}\Bigg\{c^2 \sum_{k=\kappa}^{\kappa+T} \eta_k m_k + \frac{4}{m_\kappa(m_\kappa+1)\eta_\kappa}\|\mathbf{x} - \breve{\mathbf{x}}^\kappa\|^2_{\mathcal{H}} + \|\mathbf{w} - \mathbf{w}^\kappa\|^2_{H_\kappa} \\ &\quad + \frac{2-\rho}{\rho}\left[\tau\left\|\mathbf{y}^{\kappa-1} - \mathbf{y}^\kappa\right\|^2_{\mathcal{P}_0} + (1-\tau)\beta\left\|B(\mathbf{y}^{\kappa-1} - \mathbf{y}^\kappa)\right\|^2\right]\Bigg\}. \end{aligned} \tag{3.17}$$

Proof. By using the assumption $\mathcal{D}_k \succeq \mathcal{D}_{k+1} \succeq \mathbf{0}$, that is $H_k \succeq H_{k+1} \succ \mathbf{0}$, and by plugging the above inequality (3.13) into (3.1), it can be deduced that

$$\begin{aligned} &F(\widetilde{\mathbf{u}}^k) - F(\mathbf{u}) + \langle \widetilde{\mathbf{w}}^k - \mathbf{w}, \mathcal{J}(\mathbf{w})\rangle \\ &\le -\zeta^k + \frac{1}{2}\left\{\left\|\mathbf{w} - \mathbf{w}^k\right\|^2_{H_k} - \left\|\mathbf{w} - \mathbf{w}^{k+1}\right\|^2_{H_{k+1}}\right\} \\ &\quad + \frac{2-\rho}{2\rho}\left[\tau\left\|\mathbf{y}^{k-1} - \mathbf{y}^k\right\|^2_{\mathcal{P}_0} + (1-\tau)\beta\left\|B(\mathbf{y}^{k-1} - \mathbf{y}^k)\right\|^2\right] \\ &\quad - \frac{2-\rho}{2\rho}\left[\tau\left\|\mathbf{y}^{k} - \mathbf{y}^{k+1}\right\|^2_{\mathcal{P}_0} + (1-\tau)\beta\left\|B(\mathbf{y}^{k} - \mathbf{y}^{k+1})\right\|^2\right]. \end{aligned}$$

Summing the inequality over k between κ and $\kappa+T$ together with the definitions of $\mathbf{w}_T$ and $\mathbf{u}_T$, we have

$$\begin{aligned} &\sum_{k=\kappa}^{\kappa+T} F(\widetilde{\mathbf{u}}^k) - T[F(\mathbf{u}) + \langle \mathbf{w}_T - \mathbf{w}, \mathcal{J}(\mathbf{w})\rangle] \le \frac{1}{2}\|\mathbf{w} - \mathbf{w}^\kappa\|^2_{H_\kappa} \\ &- \sum_{k=\kappa}^{\kappa+T} \zeta^k + \frac{2-\rho}{2\rho}\left[\tau\left\|\mathbf{y}^{\kappa-1} - \mathbf{y}^\kappa\right\|^2_{\mathcal{P}_0} + (1-\tau)\beta\left\|B(\mathbf{y}^{\kappa-1} - \mathbf{y}^\kappa)\right\|^2\right]. \end{aligned}$$

This together with the convexity of the function F at the point $\mathbf{u}_T$, that is, $F(\mathbf{u}_T) \le \frac{1}{T}\sum_{k=\kappa}^{\kappa+T} F(\widetilde{\mathbf{u}}^k)$, implies

$$\begin{aligned} &F(\mathbf{u}_T) - F(\mathbf{u}) + \langle \mathbf{w}_T - \mathbf{w}, \mathcal{J}(\mathbf{w})\rangle \le \frac{1}{T}\Bigg\{\frac{1}{2}\|\mathbf{w} - \mathbf{w}^\kappa\|^2_{H_\kappa} - \\ &\sum_{k=\kappa}^{\kappa+T} \zeta^k + \frac{2-\rho}{2\rho}\left[\tau\left\|\mathbf{y}^{\kappa-1} - \mathbf{y}^\kappa\right\|^2_{\mathcal{P}_0} + (1-\tau)\beta\left\|B(\mathbf{y}^{\kappa-1} - \mathbf{y}^\kappa)\right\|^2\right]\Bigg\}. \end{aligned} \tag{3.18}$$

Now, we focus on the term $-\sum_{k=\kappa}^{\kappa+T} \zeta^k$ and estimate its expectation. Since $\{m_k(m_k+1)\eta_k\}$ is nondecreasing for $k \in [\kappa, \kappa+T]$ and $\mathcal{H} \succ \mathbf{0}$, we have

$$\sum_{k=\kappa}^{\kappa+T} \frac{2}{m_k(m_k+1)\eta_k}\left(\left\|\mathbf{x} - \breve{\mathbf{x}}^k\right\|^2_{\mathcal{H}} - \left\|\mathbf{x} - \breve{\mathbf{x}}^{k+1}\right\|^2_{\mathcal{H}}\right) \le \frac{2\left\|\mathbf{x} - \breve{\mathbf{x}}^\kappa\right\|^2_{\mathcal{H}}}{m_\kappa(m_\kappa+1)\eta_\kappa}.$$

The definition of $\boldsymbol{\delta}_t$ in Lemma 2.1 and the update of $\mathbf{d}_t$ in the **xsub** routine give

$$\boldsymbol{\delta}_t = \nabla f(\widehat{\mathbf{x}}_t) - \mathbf{d}_t = \nabla f(\widehat{\mathbf{x}}_t) - \nabla f_{\xi_t}(\widehat{\mathbf{x}}_t) - \mathbf{e}_t,$$

which only depends on the index ξ_t. Hence, $\mathbb{E}[\boldsymbol{\delta}_t] = \mathbf{0}$ because the random variable $\xi_t \in \{1, 2, \ldots, N\}$ is chosen with uniform probability such that $\mathbb{E}[\mathbf{e}_t] = \mathbf{0}$. In addition, the iterate $\breve{\mathbf{x}}_t$ depending on ξ_{t-1}, ξ_{t-2}, ... will lead to $\mathbb{E}\left[\langle \boldsymbol{\delta}_t, \breve{\mathbf{x}}_t - \mathbf{x}\rangle\right] = \mathbf{0}$. Together with the hypothesis $\mathbb{E}(\|\boldsymbol{\delta}_t\|^2_{\mathcal{H}^{-1}}) \leq c^2$, it can be deduced that

$$\mathbb{E}\left[\sum_{t=1}^{m_k} t^2 \|\boldsymbol{\delta}_t\|^2_{\mathcal{H}^{-1}}\right] \leq \frac{c^2 m_k(m_k+1)(2m_k+1)}{6} \leq \frac{c^2}{2} m_k^2(m_k+1)$$

due to $m_k \geq 1$. By the definition of ζ^k in (2.5), the above inequality and the assumption $\eta_k \leq 1/(2\nu)$, we have

$$-\mathbb{E}\left[\sum_{k=\kappa}^{\kappa+T} \zeta^k\right] \leq \frac{2}{m_\kappa(m_\kappa+1)\eta_\kappa} \|\mathbf{x} - \breve{\mathbf{x}}^\kappa\|^2_{\mathcal{H}} + \frac{c^2}{2} \sum_{k=\kappa}^{\kappa+T} \eta_k m_k. \tag{3.19}$$

Taking expectation on both sides of (3.18) and using (3.19), the proof is completed. □

Next, we analyze the convergence rate of Algorithm 2.1. To proceed, denote

$$\boldsymbol{\lambda}_T = \frac{1}{T}\sum_{k=\kappa}^{\kappa+T} \widetilde{\boldsymbol{\lambda}}^k, \quad \mathbf{x}_T = \frac{1}{T}\sum_{k=\kappa}^{\kappa+T} \widetilde{\mathbf{x}}^k \quad \text{and} \quad \mathbf{y}_T = \frac{1}{T}\sum_{k=\kappa}^{\kappa+T} \widetilde{\mathbf{y}}^k. \tag{3.20}$$

By a proper choice for η_k and m_k involved in (3.17) as the following:

$$\eta_k = \min\left\{\frac{c_1}{m_k(m_k+1)}, c_2\right\} \quad \text{and} \quad m_k = \max\left\{\lceil c_3 k^\varrho \rceil, m\right\}, \tag{3.21}$$

where $c_1, c_2, c_3 > 0$, $\varrho \geq 1$ are constants and $m > 0$ is a predetermined integer, it is easy to obtain from Theorem 3.4 that

$$\mathbb{E}\left[F(\mathbf{u}_T) - F(\mathbf{u}) + (\mathbf{w}_T - \mathbf{w})^{\mathsf{T}} \mathcal{J}(\mathbf{w})\right] \leq \mathcal{O}\left(\frac{1}{T}\Big(1 + \sum_{k=\kappa}^{\kappa+T} \frac{1}{k^\varrho}\Big)\right).$$

Actually, users could choose k large enough so that $m_k = \lceil c_3 k^\varrho \rceil$. As $k \to \infty$, we will have $m_k \to \infty$ and $\eta_k \to 0$. To ensure $\eta_k = \frac{c_1}{m_k(m_k+1)} \leq \frac{1}{2\nu}$ in the condition (A1), we can select a sufficiently large k if necessary. Because $\eta_k m_k(m_k+1) = c_1$ is a constant and $\eta_k \in (0, \frac{1}{2\nu}]$ for $k \geq \kappa$, (A1) is satisfied for this choice of k. Finally, we have the following convergence theorem by a similar proof to [5, Theorem 4.2].

THEOREM 3.5. *Suppose the conditions in Theorem* 3.4 *hold. Let η_k and m_k be selected as in (3.21). Then, for every $\mathbf{w}^* = (\mathbf{u}^*, \boldsymbol{\lambda}^*) \in \Omega^*$, we have*

$$\left|\mathbb{E}\big[F(\mathbf{u}_T) - F(\mathbf{u}^*)\big]\right| = E_\varrho(T) = \mathbb{E}\big[\,\|A\mathbf{x}_T + B\mathbf{y}_T - \mathbf{b}\|\,\big], \tag{3.22}$$

where $E_\varrho(T) = \mathcal{O}(1/T)$ for $\varrho > 1$ and $E_\varrho(T) = \mathcal{O}(T^{-1}\log T)$ for $\varrho = 1$.

The result (3.22) shows that by a proper choice for the parameters involved, Algorithm 2.1 could converge in expectation with the worst-case $\mathcal{O}(1/T)$ convergence

rate in terms of the function value residual and constraint violation, where T denotes the number of outer iterations. The constraint violation $\|A\mathbf{x}_T + B\mathbf{y}_T - \mathbf{b}\|$ is important for the problem of interest, especially when stochastic gradient information is utilized instead of a deterministic gradient. It's worth noting that Algorithm 2.1 directly uses an indefinite proximal term, while the convergence of the algorithm in [3] was ensured by requiring the boundness of $\mathcal{Y}$. This is an obvious difference of using indefinite proximal matrix under different conditions. Finally, Algorithm 2.1 could use an incremental sampling of the stochastic gradient with variance reduction, see the previous discussions in [5, Section 5].

4. Numerical Experiments. In this section, we investigate the performance of the proposed AS-PRSM for solving the so-called graph-guided fused lasso problem in machine learning and the 3D CT reconstruction problem in medical image processing. The first example is implemented in MATLAB R2018a and performed on a PC with an Intel i7-8700K CPU and 16GB memory, while the second example is run in MATLAB R2019a on a desktop equipped with *TITAN RTX GPU* with 4608 cores and 24GB memory.

4.1. Graph-Guided Fused Lasso Model. Consider the following equivalent reformulation of the so-called graph-guided fused lasso model:

$$\begin{array}{ll} \min\limits_{\mathbf{x},\mathbf{y}} & F(\mathbf{x},\mathbf{y}) := \frac{1}{N}\sum\limits_{j=1}^{N} f_j(\mathbf{x}) + \mu\|\mathbf{y}\|_1 \\ \text{s.t.} & A\mathbf{x} - \mathbf{y} = \mathbf{0}, \end{array} \tag{4.1}$$

where $f_j(\mathbf{x}) = \log\left(1 + \exp(-b_j a_j^{\mathsf{T}}\mathbf{x})\right)$ denotes the logistic loss function on the feature-label pair $(a_j, b_j) \in \mathbb{R}^l \times \{-1, 1\}$ for $j \in \{1, 2, \cdots, N\}$, N is the data size, $\mu > 0$ is a predetermined regularization parameter, $A = [\mathbf{G}; \mathbf{I}]$ and the matrix $\mathbf{G}$ is obtained by a sparse inverse covariance estimation. Problem (4.1) is clearly a special case of (1.1) with $B = -\mathbf{I}$, and Assumptions 2.1-2.2 hold. Since here $B = -\mathbf{I}$, the $\mathbf{y}$-subproblem will have a closed-form solution by taking $\mathcal{P} = \mathbf{0}$. Moreover, the subproblems in Algorithm 2.1 have the following closed-form solutions:

$$\left\{\begin{array}{lcl} \breve{\mathbf{x}}_{t+1} & = & \left[\gamma_t\mathcal{H} + \mathcal{M}_k\right]^{-1}\left[\gamma_t\mathcal{H}\breve{\mathbf{x}}_t + \mathcal{M}_k\mathbf{x}^k - \mathbf{d}_t - \mathbf{h}^k\right], \\ \mathbf{y}^{k+1} & = & \text{Shrink}\left(\frac{\mu}{\beta}, sA\mathbf{x}^{k+1} + (1-s)\mathbf{y}^k - \frac{\boldsymbol{\lambda}^{k+\frac{1}{2}}}{\beta}\right), \end{array}\right.$$

where $\text{Shrink}(\cdot,\cdot)$ denotes the well-known soft shrinkage operator and can be evaluated by the built-in MATLAB function "`wthresh`". In this experiment, the penalty parameter in AS-PRSM is fixed as $\beta = 1$, the matrix $\mathcal{M}_k$ is updated adaptively by the BB technique in [5, Remark 4.2] with initial values $\rho_0 = 1.5$ and $\mathcal{H} := 2 \times 10^{-5}\mathbf{I}$. According to (3.21) and our previous experience, we set $(c_1, c_2, c_3, \varrho) = (\frac{1}{\nu}, \frac{1}{2\nu}, 0.01, 1.001)$ and $m = 200$. The tuned parameters $(\alpha, s) = (-0.6, 1.6)$ will be used since this pair of values seems to perform relatively better than others for solving (4.1). The rest parameters and the vector $\mathbf{e}_t$ in AS-PRSM are chosen by the same way as in [5, Section 7.1]. The regularization parameter μ in (4.1) is fixed as 10^{-5}. We utilize similar notations as in [5], that is,

$$\text{Obj\_err} = \frac{|F(\mathbf{w}) - F^*|}{\max\{F^*, 1\}} \qquad \text{and} \qquad \text{Equ\_err} = \|A\mathbf{x} - \mathbf{y}\|$$

to denote the relative objective error and the constraint error, respectively, where F^* denotes the approximate optimal value obtained by running AS-ADMM [5] for more

than 10 minutes. Then, the following evaluation quantity

$$\text{Opt\_err} = \max(\text{Obj\_err}, \text{Equ\_err}),$$

against the CPU time elapsed is used to measure the performance of an algorithm.

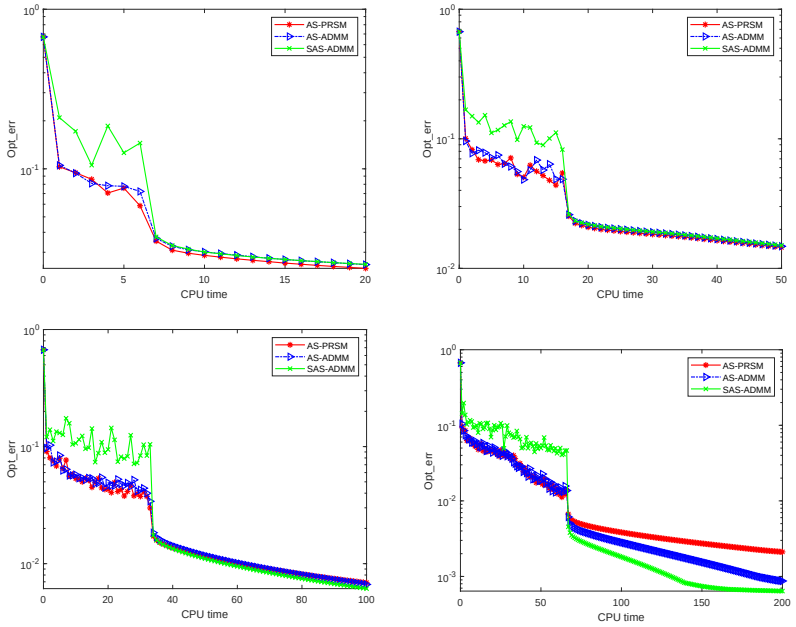


Fig. 4.1: Comparison of Opt_err vs CPU time for (4.1) and the *mnist* dataset

Under the same starting point $(\mathbf{x}^0, \mathbf{y}^0, \boldsymbol{\lambda}^0) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$, we compare AS-PRSM with our previous algorithms AS-ADMM [5] and SAS-ADMM [3] for solving the problem (4.1) on the public dataset *mnist* (including 11,791 samples and 784 features) that was downloaded from LIBSVM website. The codes of AS-ADMM and SAS-ADMM are available at https://github.com/bjc1987/bjc1987.github.io, where the involved parameter ϱ is modified as 1.001 in stead of 1.1. We make 20 successive runs for each algorithm under the CPU time budgets 20s, 50s, 100s, and 200s, respectively. Then we plot the average comparison results in Figure 4.1, where according to (3.20) we plot the error associated with the iterates over the first 1/3 of the total CPU time budget, followed by the error associated with the ergodic iterates over the last 2/3 of the budget. This coding technique was utilized according to the result (3.22) for the average iterate $\mathbf{u}_T$, see also [3, 5]. From the convergence curves in Figure 4.1, AS-PRSM is competitive to both AS-ADMM and SAS-ADMM if not much time (e.g. 20s, 50s, 100s) is allowed and it performs significantly better than SAS-ADMM at the beginning iterations. However, SAS-ADMM will perform better than the other two algorithms when high accuracy is required.

4.2. 3D CT Image Reconstruction. In this subsection, we apply AS-PRSM to resolve the 3D CT reconstruction problem in medical image processing. The observed data b is obtained by the Radon transform. The size of the reconstructed 3D image is $256 \times 256 \times 64$, the detector plane is 512×384 and the number of viewers is

668. As a result, the dimension of b is 131334144, that is, $N = 131334144$(big data). This problem can be modeled as the following minimization problem:

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{y}} F(\mathbf{x},\mathbf{y}) := \frac{1}{N}\sum_{j=1}^{N}(\mathcal{R}_j\mathbf{x} - b_j)^2 + \mu\|\mathbf{y}\|_1, \\ &\text{s.t.} \quad \nabla\mathbf{x} = \mathbf{y}, \end{aligned} \tag{4.2}$$

where $\mathcal{R}$ is the Radon transform generated by using the cone beam scanning geometry [13], ∇ is the discrete gradient operator and μ is the regularization parameter. It is easy to check that the problem (4.2) is a special case of (1.1) with $(A, B, \mathbf{b}) = (\nabla, -\mathbf{I}, \mathbf{0})$. Applying our Algorithm 2.1 to this problem, we have

$$\begin{cases} \breve{\mathbf{x}}_{t+1} &= \left[\gamma_t\sigma + \mathcal{M}_k\right]^{-1}\left[\gamma_t\sigma\breve{\mathbf{x}}_t + \mathcal{M}_k\mathbf{x}^k - \mathbf{d}_t - \mathbf{h}^k\right], \\ \mathbf{y}^{k+1} &= \text{Shrink}\left(\frac{\mu}{\sigma},\ \mathbf{y}^k - \frac{\lambda^{k+\frac{1}{2}} - s\beta(\nabla\mathbf{x}^{k+1} - \mathbf{y}^k)}{\sigma}\right). \end{cases}$$

In the forthcoming experiments, we set $(\alpha, \beta, \sigma, s) = (4\times10^{-2}, 8\times10^{-11}, 10^{-8}, 1.95)$ for Algorithm 2.1, the regularization parameter $\mu = 10^{-2}$ and the number of inner iteration $m_k = 40$. In [6], the authors used the stochastic ADMM combined with the SARAH gradient estimator (called SARAH-ADMM) as well as the SAGA gradient estimator (called SAGA-ADMM) to solve the 3D TV reconstruction problem based on L_0 norm (the number of non-zero elements). Since there is no convexity assumption for the objective function, the algorithm in [6] can also solve the problem (4.2). The authors in [20] have proposed a stochastic ADMM called STOC-ADMM, which is also applicable for solving (4.2). We compare AS-PRSM to four state-of-the-art algorithms: G-ADMM, STOC-ADMM, SAGA-ADMM and SARAH-ADMM. To guarantee the quality of the reconstructed image, we calculate the Peak Signal-to-Noise Ratio (PSNR) of the reconstructed image by the following way:

$$\text{PSNR} = 10\log_{10}\left(\frac{d_x\times d_y\times d_z}{\text{MSE}}\right) \quad \text{with} \quad \text{MSE} = \|I_m - \widetilde{I}_m\|^2, \tag{4.3}$$

where I_m denotes the original 3D image of dimension $d_x \times d_y \times d_z$ and $\widetilde{I}_m$ denotes the reconstructed 3D image.

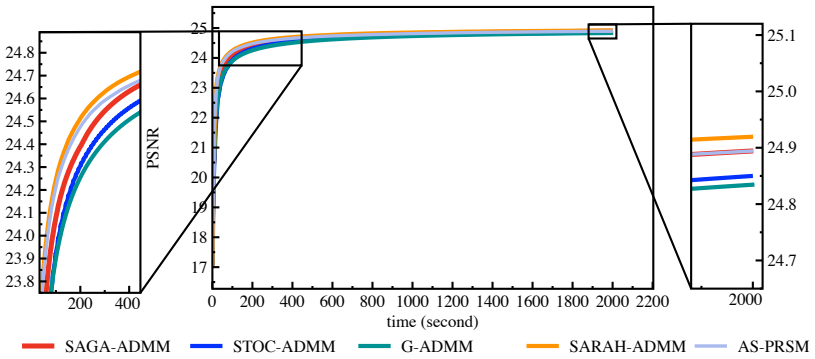


Fig. 4.2: PSNR results of five state-of-the-art algorithms.

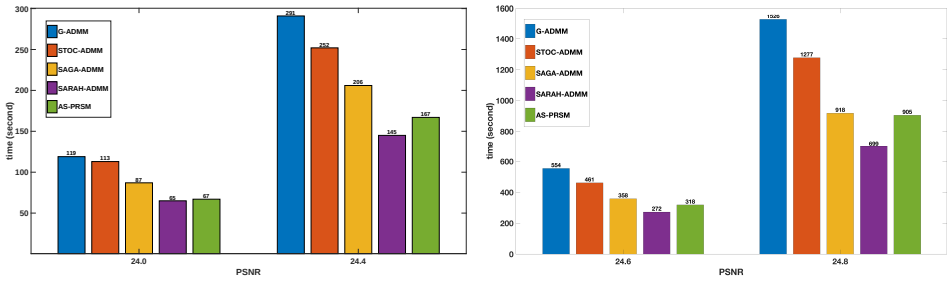
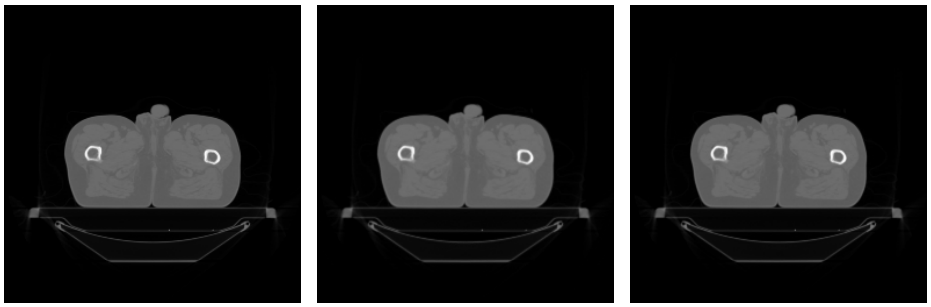
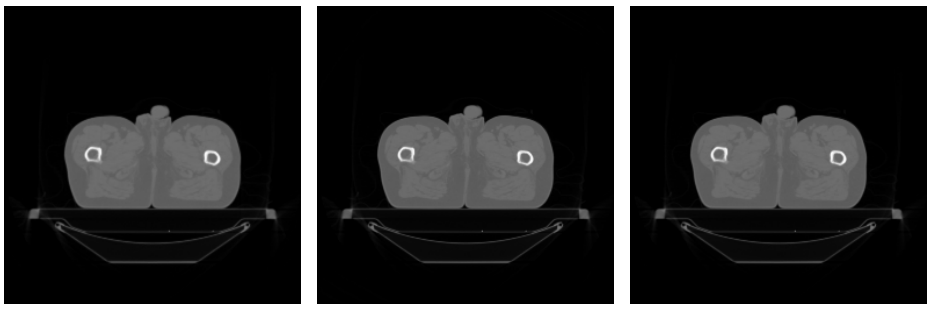


Fig. 4.3: The time required for each method to reach a different PSNR value.



(a) `Ground truth` (b) `G-ADMM= 48.02` (c) `SAGA-ADMM= 54.16`

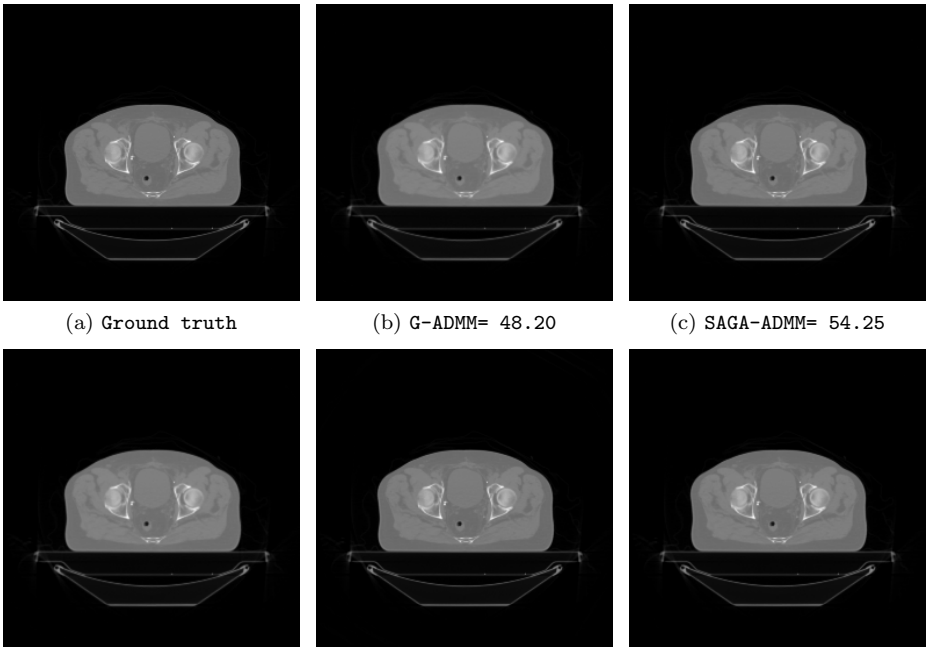


(d) `STOC-ADMM = 49.22` (e) `SARAH-ADMM= 56.31` (f) `AS-PRSM= 54.35`

Fig. 4.4: Final reconstruction image of different methods for the **15th** slice.

We report some experimental results in Figures 4.2, 4.3, 4.4 and 4.5. Figure 4.2 shows PSNR for each method running the same budget time. In Figure 4.3, we list the time required for each method to reach a specific PSNR value which takes 24.0, 24.4, 24.6 and 24.8, respectively. As can be seen from Figure 4.2, when the PSNR of the reconstructed 3D image is less than 24.65, the running time of AS-PRSM is basically similar to that of SARAH-ADMM, and is faster than that of SAGA-ADMM. However, when the required PSNR is greater than 24.65, both G-ADMM and STOC-ADMM are slower than AS-PRSM, while SAGA-ADMM and AS-PRSM take about the same amount of time, but SARAH-ADMM needs the least time. We can also find this conclusion from Figure 4.3. Figures 4.4 and 4.5 show the 15th and 45th slices of the reconstructed 3D image, respectively. We can see that the image recovered by

AS-PRSM is better than that of G-ADMM and STOC-ADMM, and similar to that of SAGA-ADMM which uses unbiased gradient estimator SAGA, but slightly worse than that of SARAH-ADMM which uses the biased gradient estimator SARAH.



(a) `Ground truth` (b) `G-ADMM= 48.20` (c) `SAGA-ADMM= 54.25`

(d) `STOC-ADMM = 49.29` (e) `SARAH-ADMM= 55.91` (f) `AS-PRSM= 54.45`

Fig. 4.5: Final reconstruction image of different methods for the **45th** slice.

5. Concluding Remarks. This paper considers the structural empirical risk minimization problem (1.1) which is a special two-block separable convex programming problem. An accelerated stochastic Peaceman-Rachford Splitting Method (PRSM), which combines both the traditional PRSM and stochastic gradient methods using variance reduction techniques, has been developed. The convergence and performance of the proposed method are established and verified respectively. Similar to our previous work [5], the ergodic convergence rate of AS-PRSM and its linear convergence rate can be established under the strong convexity assumption. However, can we show its linear convergence rate without making this strong assumption? In addition, the acceleration techniques are designed in the **xsub** routine, which does not achieve an acceleration result in theory. And the **xsub** routine uses an unbiased gradient estimator, can we use a biased gradient estimator such as SARAH to ensure convergence of AS-PRSM? These interesting questions may be further studied in the future work. From the convergence analysis of AS-PRSM, we believe that the **xsub** routine can be adopted in other first-order splitting-type methods. In particular, for the problem of (1.1) with $g(\mathbf{y}) = \frac{1}{N}\sum_{j=1}^{N} g_j(\mathbf{y})$, one may apply a similar routine to solve the $\mathbf{y}$-subproblem. This extended problem can be regarded as two different kinds of loss functions or different regularizer g_j for each loss f_j to prevent overfitting.

Acknowledgements The authors would like to thank the anonymous reviewers and the editor for providing valuable suggestions which have significantly improved the quality of the paper.

Author Contributions Jianchao Bai wrote the manuscript and established the theoretical results, Fengmiao Bian performed the second experiment and polished the manuscript, Xiaokai Chang and Lin Du helped perform the analysis with constructive discussions.

Conflict of interest The authors declare that they have no conflict of interest.

REFERENCES

[1] V. Adona and M. Goncalves, *An inexact version of the symmetric proximal ADMM for solving separable convex optimization*, arXiv:2006.02815, (2020)

[2] J. Bai, J. Li, F. Xu and H. Zhang, *Generalized symmetric ADMM for separable convex optimization*, Comput. Optim. Appl. 70 (2018) pp. 129-170.

[3] J. Bai, D. Han, H. Sun and H. Zhang, *Convergence on a symmetric accelerated stochastic ADMM with larger stepsizes*, CSIAM Trans. Appl. Math. 3 (2022) pp. 448-479.

[4] J. Bai, Y. Ma, H. Sun and M. Zhang, *Iteration complexity analysis of a partial LQP-based alternating direction method of multipliers*, Appl. Numer. Math. 165 (2021) pp. 500-518.

[5] J. Bai, W. Hager and H. Zhang, *An inexact accelerated stochastic ADMM for separable convex optimization*, Comput. Optim. Appl. 81 (2022) pp. 479-518.

[6] F. Bian, J. Liang and X. Zhang, *A stochastic alternating direction method of multipliers for non-smooth and non-convex optimization*, Inverse Problems, 37 (2021) 075009.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn. 3 (2010) pp. 1-122.

[8] X. Chang, J. Bai, D. Song and S. Liu, *Linearized symmetric multi-block ADMM with indefinite proximal regularization and optimal proximal parameter*, Calcolo, 57 (2020) pp. 1-36.

[9] J. Chen, Y. Wang, H. He and Y. Lv, *Convergence analysis of positive-indefinite proximal ADMM with a Glowinski's relaxation factor*, Numer. Algor. 83 (2020) pp. 1415-1440.

[10] Z. Deng and S. Liu, *Generalized Peaceman-Rachford splitting method with substitution for convex programming*, Optim. Lett. 14 (2020) pp. 1781-1802.

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

[12] E. Fang, B. He, H. Liu and X. M. Yuan, *Generalized alternating direction method of multipliers: new theoretical insights and applications*, Math. Prog. Comput. 7 (2015) pp. 149-187.

[13] H. Gao, *Fast parallel algorithms for the X-ray transform an its adjoint*, Medical Physics, 39 (2012) pp. 7110-7120.

[14] R. Glowinski and A. Marrocco, *Approximation paréléments finis d'rdre un et résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires*, Rev. Fr. Autom. Inform. Rech. Opér. Anal. Numér. 2 (1975) pp. 41-76.

[15] B. Gao and F. Ma, *Symmetric alternating direction method with indefinite proximal regularization for linearly constrained convex optimization*, J. Optim. Theory Appl. 176 (2018) pp. 178-204.

[16] Y. Gu, B. Jiang and D. Han, *A semi-proximal-based strictly contractive Peaceman-Rachford splitting method*, arXiv:1506.02221, (2015)

[17] B. He, F. Ma and X. Yuan, *Optimally linearizing the alternating direction method of multipliers for convex programming*, Comput. Optim. Appl. 75 (2020) pp. 361-388.

[18] B. He, F. Ma and X. Yuan, *Optimal proximal augmented Lagrangian method and its application to full Jacobian splitting for multi-block separable convex minimization problems*, IMA J. Numer. Anal. 40 (2020) pp. 1188-1216.

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

[20] F. Huang and S. Chen, *Mini-batch stochastic ADMMs for nonconvex nonsmooth optimization*, arXiv: 1802.03284v3, (2019)

[21] F. Jiang, Z. Wu and X. Cai, *Generalized ADMM with optimal indefinite proximal term for linearly constrained convex optimization*, J. Ind. Manag. Optim. 13(5) (2017) pp. 1-22.

[22] S. Kim and E. Xing, *Statistical estimation of correlated genome associations to a quantitative trait network*, PLOS Genetics, 5 (2009) pp. 1-18.

[23] H. Li, N. Liu, X. Ma, et al., *ADMM-based weight pruning for real-time deep learning acceleration on mobile devices*, Proceedings of the 2019 on Great Lakes Symposium on VLSI, (2019) pp. 501-506.

[24] M. Li and Z. Wu, *On the convergence rate of inexact majorized sGS ADMM with indefinite proximal terms for convex composite programming*, Asia Pac. J. Oper. Res. 38 (2021)

2050035.

[25] D. Peaceman and H. Rachford, *The numerical solution of parabolic and elliptic differential equations*, J. Soc. Ind. Appl. Math. 3 (1955) pp. 28-41.

[26] M. Tao, *Convergence study of indefinite proximal ADMM with a relaxation factor*, Comput. Optim. Appl. 77 (2020) pp. 91-123.

[27] Z. Wu and M. Li, *An LQP-based symmetric alternating direction method of multipliers with larger step sizes*, J. Oper. Res. Soc. China, 7 (2019) pp. 365-383.

[28] M. Yan and W. Yin, *Self equivalence of the alternating direction method of multipliers*, In R. Glowinski et al. Splitting Methods in Communication, Imaging, Science, and Engineering, Scientific Computation, In Chapter 5 (2016) pp. 165-194.

[29] Y. Zhu and X. Zhang, *Stochastic primal dual fixed point method for composite optimization*, J. Sci. Comput. 84 (2020) pp. 1-25.