Block-wise Alternating Direction Method of Multipliers with Gaussian Back Substitution for Multiple-block Convex Programming

Xiaoling Fu[1] **Bingsheng He**[2] **Xiangfeng Wang**[3] **Xiaoming Yuan** [4]

September 16, 2014

Abstract. We consider the linearly constrained convex minimization model with a separable objective function which is the sum of m functions without coupled variables, and discuss how to design an efficient algorithm based on the fundamental technique of splitting the augmented Lagrangian method (ALM). Our focus is the specific big-data scenario where m is huge. A pretreatment on the original data is to regroup the m functions in the objective and the corresponding m variables as t subgroups, where t is a handleable number (usually $t \geq 3$ but much smaller than m). To tackle the regrouped model with t blocks of functions and variables, some existing splitting methods in the literature are applicable. We concentrate on the application of the alternating direction method of multiplier with Gaussian back substitution (ADMM-GBS) whose efficiency and scalability have been well verified in the literature. The block-wise ADMM-GBS is thus resulted and named when the ADMM-GBS is applied to solve the t-block regrouped model. To alleviate the difficulty of the resulting ADMM-GBS subproblems, each of which may still require minimizing more than one function with coupled variables, we suggest further decomposing these subproblems but proximally regularizing these further decomposed subproblems to ensure the convergence. With this further decomposition, each of the resulting subproblems only requires handling one function in the original objective plus a simple quadratic term; it thus may be very easy for many concrete applications where the functions in the objective have some specific properties. Moreover, these further decomposed subproblems can be solved in parallel, making it possible to handle big-data by highly capable computing infrastructures. Consequently, a splitting version of the block-wise ADMM-GBS, is proposed for the particular big-data scenario. The implementation of this new algorithm is suitable for a centralized-distributed computing system, where the decomposed subproblems of each block can be computed in parallel by a distributed-computing infrastructure and the blocks are updated by a centralized-computing station. For the new algorithm, we prove its convergence and establish its worst-case convergence rate measured by the iteration complexity. Two refined versions of this new algorithm with iteratively calculated step sizes and linearized subproblems are also proposed, respectively.

Key Words: Convex programming, Alternating direction method of multipliers, Big data, Distributed computing, Centralized computing, Splitting methods, Convergence rate

[1]Institute of Systems Engineering, Southeast University, Nanjing, 210096, China. This author was supported by the NSFC grant 70901018. Email: fuxlnju@hotmail.com

[2]International Centre of Management Science and Engineering, and Department of Mathematics, Nanjing University, Nanjing, 210093, China. This author was supported by the NSFC Grant 91130007 and 11471156. Email: hebma@nju.edu.cn

[3]Shanghai Key Laboratory of Trustworthy Computing, Software Engineering Institute, East China Normal University, Shanghai 200062, China. Email: xfwang@sei.ecnu.edu.cn

[4]Corresponding author, Department of Mathematics, Hong Kong Baptist University, Hong Kong. This author was supported by a General Research Fund from Hong Kong Research Grants Council. Email: xmyuan@hkbu.edu.hk

1 Introduction

We consider a separable convex minimization problem with linear constraints and its objective function is the sum of more than one function without coupled variables:

$$\min\Big\{\sum_{i=1}^{m}\theta_i(x_i)\ \Big|\ \sum_{i=1}^{m}A_ix_i=b,\ x_i\in X_i, i=1,\cdots,m\Big\},\tag{1.1}$$

where $\theta_i:\mathbb{R}^{n_i}\to\mathbb{R}$ $(i=1,\cdots,m)$ are convex (not necessarily smooth) closed functions; $A_i\in\mathbb{R}^{\ell\times n_i}$, $b\in\mathbb{R}^{\ell}$, and $X_i\subseteq\mathbb{R}^{n_i}$ $(i=1,\cdots,m)$ are closed convex sets. The solution set of (1.1) is assumed to be nonempty throughout our discussions in this paper. We also assume that matrices $A_i^TA_i$ $(i=1,\ldots,m)$ are all nonsingular.

Our discussion is under the assumption that each function θ_i in the objective of (1.1) has some specific properties and it is worthwhile to take advantage of them in algorithmic design. One representative case, which has wide applications in many sparse- and/or low-rank-related fields, is when the proximal operator of θ_i given by

$$\arg\min_{x_i\in\mathbb{R}^{n_i}}\big\{\theta_i(x_i)+\frac{\tau}{2}\|x_i-p_i\|^2\big\}\tag{1.2}$$

has a closed-form representation for any given vector $p_i\in\mathbb{R}^{n_i}$ and scalar $\tau>0$. In (1.2), $\|\cdot\|$ denotes the standard l_2 norm. Thus, we do not discuss the case where the model (1.1) is treated as a whole and its separable structures are ignored in algorithmic design. Instead, we are interested in such an algorithm whose subproblems at each iteration are all of the same difficulty as (1.2) or at most as the one

$$\arg\min_{x_i\in\mathbb{R}^{n_i}}\big\{\theta_i(x_i)+\frac{\tau}{2}\|A_ix_i-a\|^2\big\}\tag{1.3}$$

with $a\in\mathbb{R}^{\ell}$. Note that when the proximal operator given in (1.2) has a closed-form representation, solving (1.3) is generally easy. For instance, the problem (1.3) can be iteratively solved by linearizing the quadratic term in (1.3) because the linearized subproblem reduces to the task of evaluating the proximal operator defined in (1.2). This is indeed an implementation of the forward-backward splitting method which originated in [28]. Therefore, to expose our main idea of algorithmic design with easier notation, we mainly focus on the discussion of designing an algorithm with subproblems in form of (1.3) and only briefly mention its advanced version with subproblems in form of (1.2).

The augmented Lagrangian method (ALM) in [23, 30] is the basis for a number of splitting methods in the literature for solving the model (1.1). Let the Lagrange function of (1.1) be

$$L^m(x_1,x_2,\ldots,x_m,\lambda)=\sum_{i=1}^{m}\theta_i(x_i)-\lambda^T(\sum_{i=1}^{m}A_ix_i-b),\tag{1.4}$$

with $\lambda\in\mathbb{R}^{\ell}$ the Lagrange multiplier and it be defined on $\Omega=X_1\times X_2\times\cdots\times X_m\times\mathbb{R}^{\ell}$. The augmented Lagrangian function is

$$\mathcal{L}_{\beta}^m(x_1,\ldots,x_m,\lambda)=L^m(x_1,\ldots,x_m,\lambda)+\frac{\beta}{2}\|\sum_{i=1}^{m}A_ix_i-b\|^2,\tag{1.5}$$

where $L^m(x_1,x_2,\ldots,x_m,\lambda)$ is given by (1.4) and $\beta>0$ is a penalty parameter with respect to the violation of the linear constraints in (1.1). If we treat the primal variables in model (1.1) as a whole and apply directly the ALM, then the resulting scheme is

$$\begin{cases}(x_1^{k+1},x_2^{k+1},\cdots,x_m^{k+1})=\arg\min\big\{\mathcal{L}_{\beta}^m(x_1,x_2,\cdots,x_m,\lambda^k)\ \big|\ x_i\in X_i,\ i=1,\cdots,m\big\},\\ \lambda^{k+1}=\lambda^k-\beta(\sum_{i=1}^{m}A_ix_i^{k+1}-b).\end{cases}\tag{1.6}$$

The minimization subproblem in (1.6) is clearly not efficient under the mentioned assumption that each θ_i has specific properties. Thus, when considering the model (1.1), the scheme (1.6) is only of conceptual sense. But it is the basis of a number of efficient methods in the literature whose common feature is to decompose the minimization subproblem in (1.6) appropriately and then to ensure the convergence with some additional steps if necessary. The most successful case is decomposing the minimization subproblem in (1.6) in Gauss-Seidel order for the special case of (1.1) with $m = 2$:

$$\begin{cases} x_1^{k+1} = \arg\min\{\mathcal{L}_\beta^2(x_1, x_2^k, \lambda^k) \mid x_1 \in X_1\}, \\ x_2^{k+1} = \arg\min\{\mathcal{L}_\beta^2(x_1^{k+1}, x_2, \lambda^k) \mid x_2 \in X_2\}, \\ \lambda^{k+1} = \lambda^k - \beta(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b). \end{cases} \tag{1.7}$$

This is the so-called alternating direction method of multiplier (ADMM) in [11] and it has found many efficient applications in a broad spectrum of application domains such as image processing, statistical learning, computer vision, network optimization, and so on. We refer to [2, 5, 10] for some review papers on the ADMM.

With the efficiency of ADMM, it is natural to consider directly extending the scheme (1.7) to the case of (1.1) with $m > 2$. The resulting direct extension of ADMM reads as

$$\begin{cases} x_1^{k+1} = \arg\min\{\mathcal{L}_\beta^m(x_1, x_2^k, \cdots, x_m^k, \lambda^k) \mid x_1 \in X_1\}, \\ \quad\vdots \\ x_i^{k+1} = \arg\min\{\mathcal{L}_\beta^m(x_1^{k+1}, \cdots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \cdots, x_m^k, \lambda^k) \mid x_i \in X_i\}, \\ \quad\vdots \\ x_m^{k+1} = \arg\min\{\mathcal{L}_\beta^m(x_1^{k+1}, \cdots, x_{m-1}^{k+1}, x_m, \lambda^k) \mid x_m \in X_m\}, \\ \lambda^{k+1} = \lambda^k - \beta(\sum_{i=1}^m A_i x_i^{k+1} - b). \end{cases} \tag{1.8}$$

Empirically, the direct extension of ADMM scheme (1.8) indeed works well for some applications, as shown in, e.g. [29, 31]. However, it was shown in [3] that theoretically the scheme (1.8) is not necessarily convergent. Hence, like the extreme case of treating (1.1) as a whole and applying no splitting at all to the ALM (1.6), this scheme (1.8) resulted by applying a full splitting to the ALM (1.6) does not work either.

In the literature, some surrogates with provable convergence and numerical performance competitive to (1.8) have been well studied. For examples, the schemes in [15, 16] treat the output of (1.8) as a predictor and suggest correcting it appropriately to ensure the convergence. These schemes are all in the prediction-correction framework. The scheme in [17] requires no correction step, but it slightly changes the order of updating the Lagrange multiplier and twists some of the subproblems appropriately to obtain the convergence. Accordingly, the $(x_2, \cdots, x_m)$-subproblems can be solved in parallel but they should be regularized by appropriate proximal terms with sufficiently large proximal coefficients. Moreover, the scheme in [24] suggests attaching a shrinking factor to the Lagrange multiplier updating step in (1.8). In [3], it was shown that it could be very hard to find such a factor to guarantee the convergence of the direct extension of the ADMM scheme (1.8).

In this paper, we focus on the particular case of (1.1) which arises from a big-data scenario; thus m is assumed to be huge. Under this big-data scenario with a huge m, a pretreatment on the original model (data) is usually implemented. For example, we can classify the original functions and the corresponding variables into t classes by identifying some common features or data-processing in particular applications. A more specific case is that t represents the number of features in a data-mining application of the abstract model (1.1). In general, $t \geq 2$ is a handleable number but it is much smaller than m. The general model (1.1) is thus treated as a separable

model with t blocks of functions and variables. For the r-th block ($r = 1, 2, \cdots, t$), let m_r be the number of variables in the r-th block and thus $\sum_{r=1}^{t} m_r = m$. That is, we consider regrouping the variables $x = (x_1, x_2, \cdots, x_m)$ and functions $(\theta_1, \theta_2, \cdots, \theta_m)$ in (1.1) as $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_t)$ with $\boldsymbol{x}_r = (x_{r_1}, x_{r_2}, \cdots, x_{r_{m_r}})$ and $(\vartheta_1(\boldsymbol{x}_1), \vartheta_2(\boldsymbol{x}_2), \cdots, \vartheta_t(\boldsymbol{x}_t))$ with $\vartheta_r(\boldsymbol{x}_r) = \sum_{j=1}^{m_r} \theta_{r_j}(x_{r_j})$, respectively; and furthermore, we define

$$\mathcal{A}_r = (A_{r_1}, \ldots, A_{r_{m_r}}), \qquad \mathcal{X}_r = \prod_{j=1}^{m_r} X_{r_j},\ r = 1, \ldots, t. \tag{1.9}$$

Then, the model (1.1) can be reformulated as the block-wise form

$$\min\Big\{\sum_{r=1}^{t} \vartheta_r(\boldsymbol{x}_r) \,\Big|\, \sum_{r=1}^{t} \mathcal{A}_r \boldsymbol{x}_r = b,\ \boldsymbol{x}_r \in \mathcal{X}_r,\ r = 1, \ldots, t\Big\}. \tag{1.10}$$

Reiterating the block-wise reformulation (1.10) may account for the application where each block of variables and functions represents a specific set of decision variables and cost functions in the same classification. Accordingly, the Lagrange function (1.4) can be written as the block-wise

$$L^t(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \lambda) = \sum_{r=1}^{t} \vartheta_r(\boldsymbol{x}_r) - \lambda^T(\textstyle\sum_{r=1}^{t} \mathcal{A}_r \boldsymbol{x}_r - b), \tag{1.11}$$

and thus the augmented Lagrangian function (1.5) as

$$\mathcal{L}_\beta^t(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \lambda) = L^t(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \lambda) + \frac{\beta}{2}\|\textstyle\sum_{r=1}^{t} \mathcal{A}_r \boldsymbol{x}_r - b\|^2. \tag{1.12}$$

When $t = 2$, the original ADMM scheme (1.7) can be applicable to the block-wise reformulation (1.10) and its iterative scheme reads as

$$\begin{cases} \boldsymbol{x}_1^{k+1} = \arg\min\big\{\mathcal{L}_\beta^2(\boldsymbol{x}_1, \boldsymbol{x}_2^k, \lambda^k) \,\big|\, \boldsymbol{x}_1 \in \mathcal{X}_1\big\}, \\ \boldsymbol{x}_2^{k+1} = \arg\min\big\{\mathcal{L}_\beta^2(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2, \lambda^k) \,\big|\, \boldsymbol{x}_2 \in \mathcal{X}_2\big\}, \\ \lambda^{k+1} = \lambda^k - \beta(\mathcal{A}_1 \boldsymbol{x}_1^{k+1} + \mathcal{A}_2 \boldsymbol{x}_2^{k+1} - b). \end{cases} \tag{1.13}$$

We refer to [22] for the discussion of how to further decomposing the subproblems in (1.13) and obtain solvable subproblems in form of (1.3).

In this paper, we focus on the case of $t \geq 3$ and discuss how to design implementable algorithms for the block-wise reformulation (1.10). Recall that the scheme (1.8) is not necessarily convergent. Thus, its block-wise extension to (1.10), which reads as

$$\begin{cases} \boldsymbol{x}_1^{k+1} = \arg\min\big\{\mathcal{L}_\beta^t(\boldsymbol{x}_1, \boldsymbol{x}_2^k, \cdots, \boldsymbol{x}_t^k, \lambda^k) \,\big|\, \boldsymbol{x}_1 \in \mathcal{X}_1\big\}, \\ \qquad \vdots \\ \boldsymbol{x}_r^{k+1} = \arg\min\big\{\mathcal{L}_\beta^t(\boldsymbol{x}_1^{k+1}, \cdots, \boldsymbol{x}_{r-1}^{k+1}, \boldsymbol{x}_r, \boldsymbol{x}_{r+1}^k, \cdots, \boldsymbol{x}_t^k, \lambda^k) \,\big|\, \boldsymbol{x}_r \in \mathcal{X}_r\big\}, \\ \qquad \vdots \\ \boldsymbol{x}_t^{k+1} = \arg\min\big\{\mathcal{L}_\beta^t(\boldsymbol{x}_1^{k+1}, \cdots, \boldsymbol{x}_{t-1}^{k+1}, \boldsymbol{x}_t, \lambda^k) \,\big|\, \boldsymbol{x}_t \in \mathcal{X}_t\big\}, \\ \lambda^{k+1} = \lambda^k - \beta(\textstyle\sum_{r=1}^{t} \mathcal{A}_r \boldsymbol{x}_r^{k+1} - b), \end{cases} \tag{1.14}$$

is not necessarily convergent, either; and it is important to investigate how to design implementable algorithms for (1.10) based on the scheme (1.14). In particular, since the efficiency and stability of the ADMM with a Gaussian back substitution (ADMM-GBS for short) in [15] has been well verified

for various applications such as image processing, statistical learning, SDP, etc., we focus on this method and further discuss how to extend it to the block-wise reformulation (1.10).

The rest of this paper is organized as follows. In Section 2, we review some known results and prove some preliminary propositions which are useful for further analysis. The new algorithm is presented in Section 3, followed by some remarks. Then, we prove the convergence for the new algorithm in Section 4; and establish its worst-case convergence rate in Section 5. In Section 6, we elucidate some special cases of the new algorithm and see its relationship to some existing schemes in the literature. We present a refined version for the new algorithm with an iteratively calculated step size in Section 7; and briefly mention its convergence analysis. In Section 8, we present a linearized version of the new algorithm proposed in Section 3, whose subproblems are in form of (1.2) rather than (1.3). In addition, two key results which essentially guarantee its convergence are established for this linearized version. Finally, we make some conclusions in Section 9.

2 Preliminaries

In this section, we summarize some results known in the literature and introduce some additional notations for the convenience of analysis later.

2.1 Variational Inequality Characterization

Let $\left(x_1^*, x_2^*, \ldots, x_m^*, \lambda^*\right)$ be a saddle point of the Lagrange function (1.4), it follows that

$$L^m_{\lambda\in\mathbb{R}^\ell}(x_1^*, x_2^*, \cdots, x_m^*, \lambda) \le L^m(x_1^*, x_2^*, \cdots, x_m^*, \lambda^*) \le L^m_{x_i\in\mathcal{X}_i\ (i=1,\ldots,m)}(x_1, x_2, \ldots, x_m, \lambda^*).$$

Then, finding a saddle point of $L^m(x_1, x_2, \ldots, x_m, \lambda)$ is equivalent to finding $(x_1^*, x_2^*, ..., x_m^*, \lambda^*) \in \Omega$ such that

$$\left\{\begin{array}{lll} x_1^*\in X_1, & \theta_1(x_1)-\theta_1(x_1^*)+(x_1-x_1^*)^T(-A_1^T\lambda^*)\ge 0, & \forall x_1\in X_1,\\ x_2^*\in X_2, & \theta_2(x_2)-\theta_2(x_2^*)+(x_2-x_2^*)^T(-A_2^T\lambda^*)\ge 0, & \forall x_2\in X_2,\\ & \vdots & \\ x_m^*\in X_m, & \theta_m(x_m)-\theta_m(x_m^*)+(x_m-x_m^*)^T(-A_m^T\lambda^*)\ge 0, & \forall x_m\in X_m,\\ \lambda^*\in\mathbb{R}^\ell, & (\lambda-\lambda^*)^T(\sum_{i=1}^m A_ix_i^*-b)\ge 0, & \forall\lambda\in\mathbb{R}^\ell. \end{array}\right. \tag{2.1}$$

We denote by Ω^* the set of all saddle points of $L^m(x_1, x_2, \ldots, x_m, \lambda)$. More compactly, (2.1) can be written as the following variational inequality:

$$\mathrm{VI}(\Omega, F, \theta)\quad \boldsymbol{w}^*\in\Omega,\ \ \vartheta(\boldsymbol{x})-\vartheta(\boldsymbol{x}^*)+(\boldsymbol{w}-\boldsymbol{w}^*)^TF(\boldsymbol{w}^*)\ge 0,\ \ \forall\,\boldsymbol{w}\in\Omega, \tag{2.2a}$$

where

$$\boldsymbol{x}=\begin{pmatrix} x_1\\ \vdots\\ x_m\end{pmatrix},\quad \boldsymbol{w}=\begin{pmatrix} x_1\\ \vdots\\ x_m\\ \lambda\end{pmatrix},\quad \vartheta(\boldsymbol{x})=\sum_{i=1}^m\theta_i(x_i),\quad F(\boldsymbol{w})=\begin{pmatrix} -A_1^T\lambda\\ \vdots\\ -A_m^T\lambda\\ \sum_{i=1}^m A_ix_i-b\end{pmatrix}. \tag{2.2b}$$

Using the mentioned block-wise notation, we can rewrite (2.1)-(2.2) respectively as

$$\left\{\begin{array}{lll} \boldsymbol{x}_1^*\in\mathcal{X}_1, & \vartheta_1(\boldsymbol{x}_1)-\vartheta_1(\boldsymbol{x}_1^*)+(\boldsymbol{x}_1-\boldsymbol{x}_1^*)^T(-\mathcal{A}_1^T\lambda^*)\ge 0, & \forall\boldsymbol{x}_1\in\mathcal{X}_1,\\ \boldsymbol{x}_2^*\in\mathcal{X}_2, & \vartheta_2(\boldsymbol{x}_2)-\vartheta_2(\boldsymbol{x}_2^*)+(\boldsymbol{x}_2-\boldsymbol{x}_2^*)^T(-\mathcal{A}_2^T\lambda^*)\ge 0, & \forall\boldsymbol{x}_2\in\mathcal{X}_2,\\ & \vdots & \\ \boldsymbol{x}_t^*\in\mathcal{X}_t, & \vartheta_t(\boldsymbol{x}_t)-\vartheta_t(\boldsymbol{x}_t^*)+(\boldsymbol{x}_t-\boldsymbol{x}_t^*)^T(-\mathcal{A}_m^T\lambda^*)\ge 0, & \forall\boldsymbol{x}_t\in\mathcal{X}_t,\\ \lambda^*\in\mathbb{R}^\ell, & (\lambda-\lambda^*)^T(\sum_{r=1}^t\mathcal{A}_r\boldsymbol{x}_r^*-b)\ge 0, & \forall\lambda\in\mathbb{R}^\ell, \end{array}\right. \tag{2.3}$$

and

$$\mathrm{VI}(\Omega, F, \theta) \quad \boldsymbol{w}^* \in \Omega, \ \ \vartheta(\boldsymbol{x}) - \vartheta(\boldsymbol{x}^*) + (\boldsymbol{w} - \boldsymbol{w}^*)^T F(\boldsymbol{w}^*) \geq 0, \ \ \forall\, \boldsymbol{w} \in \Omega, \tag{2.4a}$$

where

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_t \end{pmatrix}, \quad \boldsymbol{w} = \begin{pmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_t \\ \lambda \end{pmatrix}, \quad \vartheta(\boldsymbol{x}) = \sum_{r=1}^{t} \vartheta_r(\boldsymbol{x}_r), \quad F(\boldsymbol{w}) = \begin{pmatrix} -\mathcal{A}_1^T\lambda \\ \vdots \\ -\mathcal{A}_t^T\lambda \\ \sum_{r=1}^{t} \mathcal{A}_r\boldsymbol{x}_r - b \end{pmatrix}. \tag{2.4b}$$

2.2 Some Properties

Recall the matrices $\mathcal{A}_r$'s defined in (1.9). Then, for $\mathcal{A}_r$ and $\mathcal{A}_s$, we have

$$\mathcal{A}_r^T\mathcal{A}_s = \begin{pmatrix} A_{r_1}^T A_{s_1} & \cdots & \cdots & A_{r_1}^T A_{s_{m_s}} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ A_{r_{m_r}}^T A_{s_1} & \cdots & \cdots & A_{r_{m_r}}^T A_{s_{m_s}} \end{pmatrix}.$$

For these matrices $\mathcal{A}_r$'s, they have a useful property for further analysis. We summarize it the following lemma and omit its trivial proof.

Lemma 2.1. *For the matrix $\mathcal{A}_r$ defined in (1.9), if we define*

$$\mathit{diag}(\mathcal{A}_r^T\mathcal{A}_r) := \begin{pmatrix} A_{r_1}^T A_{r_1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_{r_{m_r}}^T A_{r_{m_r}} \end{pmatrix}, \tag{2.5}$$

then we have

$$m_r \cdot \mathit{diag}(\mathcal{A}_r^T\mathcal{A}_r) \succeq \mathcal{A}_r^T\mathcal{A}_r, \quad r = 1, \ldots, t. \tag{2.6}$$

Further more, we define

$$\tau_r \geq m_r - 1, \qquad \text{and} \qquad D_r = (\tau_r + 1)\mathrm{diag}(\mathcal{A}_r^T\mathcal{A}_r), \quad r = 1, \ldots, t. \tag{2.7}$$

3 The Block-wise ADMM with Gaussian Back Substitution

In this section, we consider how to extend the ADMM-GBS in [15] to the block-wise reformulation (1.10) of the original model (1.1) and propose a block-wise ADMM-GBS with solvable subproblems in form of (1.3). In particular, this block-wise ADMM-GBS turns out to be a unified scheme including the existing algorithms in [15, 17] as special cases.

3.1 The ADMM-GBS in [15]

First of all, let us recall the ADMM-GBS in [15] for the original model (1.1). As mentioned, the ADMM-GBS in [15] treats the output of the direct extension of the ADMM scheme (1.8) as a

predictor and corrects it via a Gaussian back substitution procedure to ensure the convergence. Its iterative scheme reads as

$$\begin{cases} \begin{cases} \bar{x}_1^{k+1} = \arg\min\{\mathcal{L}_\beta^m(x_1, x_2^k, \cdots, x_m^k, \lambda^k) \mid x_1 \in X_1\}, \\ \qquad \vdots \\ \bar{x}_i^{k+1} = \arg\min\{\mathcal{L}_\beta^m(\bar{x}_1^{k+1}, \cdots, \bar{x}_{i-1}^{k+1}, x_i, x_{i+1}^k, \cdots, x_m^k, \lambda^k) \mid x_i \in X_i\}, \\ \qquad \vdots \\ \bar{x}_m^{k+1} = \arg\min\{\mathcal{L}_\beta^m(\bar{x}_1^{k+1}, \cdots, \bar{x}_{m-1}^{k+1}, x_m, \lambda^k) \mid x_m \in X_m\}, \\ \bar{\lambda}^{k+1} = \lambda^k - \beta(\sum_{i=1}^m A_i \bar{x}_i^{k+1} - b), \end{cases} \\ \boldsymbol{v}^{k+1} = \boldsymbol{v}^k - \alpha P^{-1}(\boldsymbol{v}^k - \bar{\boldsymbol{v}}^{k+1}), \quad \alpha \in (0,1), \end{cases} \tag{3.1}$$

where $\mathcal{L}_\beta^m$ is defined in (1.5) and the matrix P is a block-wise upper triangular matrix defined as

$$P = \begin{pmatrix} I_{n_2} & (A_2^TA_2)^{-1}A_2^TA_3 & \cdots & (A_2^TA_2)^{-1}A_2^TA_m & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & (A_{m-1}^TA_{m-1})^{-1}A_{m-1}^TA_m & 0 \\ 0 & \cdots & 0 & I_{n_m} & 0 \\ 0 & \cdots & 0 & 0 & I_\ell \end{pmatrix}, \tag{3.2}$$

whose dimension is $(n_2 + \cdots + n_m + \ell)$. Note that in (3.1), $\boldsymbol{v}$ represents the collection of variables $(x_2^T, \cdots, x_m^T, \lambda^T)^T$ which are essentially required in the iteration. As mentioned in [2], the first variable x_1 is not required in the iteration and it is thus "intermediate" in the iteration. This is why in the scheme (3.1), the back substitution procedure is only implemented to $\boldsymbol{v}$ without x_1. Clearly, the last step in (3.1) can be written as

$$P(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k) = \alpha(\bar{\boldsymbol{v}}^{k+1} - \boldsymbol{v}^k).$$

Thus, with the block-wise upper triangular matrix P defined in (3.2), the entries of $\boldsymbol{v}^{k+1}$ can be updated in the order of $\lambda \to x_m \to \cdots x_2$, just like the standard Gaussian back substitution procedure for solving a system of liner equations.

For the ADMM-GBS (3.1), the ADMM splitting step (i.e., the x_i-subproblems in (1.8)) is mainly for yielding easier subproblems so that it becomes possible to exploit the properties of θ_i's individually. However, since yielding these easier subproblems is on the cost that the individual m x_i-subproblems in (3.1) is only an approximation of the ALM subproblem in (1.6) and thus the decomposed subproblems, even if all are solved exactly, are not necessarily accurate enough to provide a qualified input to update the Lagrange multiplier such that the convergence can be still ensured. This is an explanation of the failure of convergence for the direct extension of ADMM (1.8), see the counter example given in [3] showing the divergence of the direct extension of ADMM (1.8). The Gaussian back substitution step in (3.1) can thus be regarded as a correction step to compensate the inaccuracy resulted by the decomposition on the ALM and so as to ensure the contraction property for the iterative sequence to the solution set. With this contraction, the convergence of (3.1) can be established from the contraction method perspective.

3.2 Motivation

Since we consider the block-wise reformulation (1.10) with $t \geq 3$ blocks and the block-wise direct extension of ADMM scheme (1.14) is not necessarily convergent, the ADMM-GBS (3.1) can be extended to (1.10) and its convergence is ensured provided that all the resulting subproblems are solved exactly. The resulting block-wise ADMM-GBS reads as

$$\left\{\begin{array}{l} \left\{\begin{array}{l} \bar{\boldsymbol{x}}_1^{k+1} = \arg\min\{\mathcal{L}_\beta^t(\boldsymbol{x}_1, \boldsymbol{x}_2^k, \cdots, \boldsymbol{x}_t^k, \lambda^k) \mid \boldsymbol{x}_1 \in \mathcal{X}_1\}, \\ \qquad \vdots \\ \bar{\boldsymbol{x}}_r^{k+1} = \arg\min\{\mathcal{L}_\beta^t(\bar{\boldsymbol{x}}_1^{k+1}, \cdots, \bar{\boldsymbol{x}}_{r-1}^{k+1}, \boldsymbol{x}_r, \boldsymbol{x}_{r+1}^k, \cdots, \boldsymbol{x}_t^k, \lambda^k) \mid \boldsymbol{x}_r \in \mathcal{X}_r\}, \\ \qquad \vdots \\ \bar{\boldsymbol{x}}_t^{k+1} = \arg\min\{\mathcal{L}_\beta^t(\bar{\boldsymbol{x}}_1^{k+1}, \cdots, \bar{\boldsymbol{x}}_{t-1}^{k+1}, \boldsymbol{x}_t, \lambda^k) \mid \boldsymbol{x}_t \in \mathcal{X}_t\}, \\ \bar{\lambda}^{k+1} = \lambda^k - \beta(\sum_{r=1}^t \mathcal{A}_r \bar{\boldsymbol{x}}_r^{k+1} - b), \end{array}\right. \\ \boldsymbol{v}^{k+1} = \boldsymbol{v}^k - \alpha \mathcal{P}^{-1}(\boldsymbol{v}^k - \bar{\boldsymbol{v}}^{k+1}), \quad \alpha \in (0,1), \end{array}\right. \tag{3.3}$$

where $\mathcal{L}_\beta^t$ is defined in (1.12) and the matrix $\mathcal{P}$ in (3.3) is a block-wise upper triangular matrix similar as in (3.2), see (3.8). Note that this block-matrix $\mathcal{P}$ makes the output of (1.14) updated via a Gaussian back substitution procedure in block-wise form in the scheme (3.3).

For a general case, similar as (1.14), each of the minimization subproblems in (3.3) involves more than one function in its objective and the m_r variables are coupled by the quadratic term in (1.12). This may make it hard to solve these subproblems unless the special case $m_r = 1$. Recall that we only consider the case where each subproblem to be solved is in the form of (1.2) or (1.3). Thus, we suggest further decomposing the $\boldsymbol{x}_r$-subproblem in (3.3) as m_r smaller subproblems so that each function θ_i is treated individually. More specifically, the block-wise $\boldsymbol{x}_r$-subproblem in (3.3) is decomposed as the following m_r smaller subproblems:

$$\left\{\begin{array}{l} \bar{x}_{r_1}^{k+1} = \arg\min\{\mathcal{L}_\beta^t(\bar{\boldsymbol{x}}_1^{k+1}, \cdots, \bar{\boldsymbol{x}}_{r-1}^{k+1}, x_{r_1}, x_{r_2}^k, \cdots, x_{r_{m_r}}^k, \boldsymbol{x}_{r+1}^k, \cdots, \boldsymbol{x}_t^k, \lambda^k) \mid x_{r_1} \in X_{r_1}\}, \\ \qquad \vdots \\ \bar{x}_{r_j}^{k+1} = \arg\min\{\mathcal{L}_\beta^t(\bar{\boldsymbol{x}}_1^{k+1}, \cdots, \bar{\boldsymbol{x}}_{r-1}^{k+1}, x_{r_1}^k, \cdots, x_{r_{j-1}}^k, x_{r_j}, x_{r_{j+1}}^k, \cdots, x_{r_{m_r}}^k, \boldsymbol{x}_{r+1}^k, \cdots, \boldsymbol{x}_t^k, \lambda^k) \mid x_{r_j} \in X_{r_j}\}, \\ \qquad \vdots \\ \bar{x}_{r_{m_r}}^{k+1} = \arg\min\{\mathcal{L}_\beta^t(\bar{\boldsymbol{x}}_1^{k+1}, \cdots, \bar{\boldsymbol{x}}_{r-1}^{k+1}, x_{r_1}^k, \cdots, x_{r_{m_r-1}}^k, x_{r_{m_r}}, \boldsymbol{x}_{r+1}^k, \cdots, \boldsymbol{x}_t^k, \lambda^k) \mid x_{r_{m_r}} \in X_{r_{m_r}}\}. \end{array}\right. \tag{3.4}$$

Note that we only consider implementing the parallel decomposition to the $\boldsymbol{x}_r$-subproblem in (3.3). This makes it possible to implement parallel computation to tackle each block of subproblems by, e.g., a distributed-computing system. To summarize, the implementation of the new algorithm can be ordered as t main phases which are proceeded sequentially according to the block-wise ADMM-GBS scheme (3.3); and for the r-th phase, there are m_r subtasks in form of (1.3) which can be proceeded in parallel. This feature is useful for big-data scenarios where parallel computation is necessary.

It is also worthwhile to mentation that if the alternating decomposition is implemented to the $\boldsymbol{x}_r$-subproblem in (3.3), then the resulting scheme reduces to the original ADMM-GBS (3.1). Recall that the ADMM-GBS (3.1) requires solving all the decomposed subproblems in a completely sequential way. Hence, when the big-data scenario is consider where m is huge in (1.1), the waiting time resulted by the sequential computing might be too expensive if the ADMM-GBS (3.1) is directly used. We are thus interested in implementing the ADMM-GBS in the block-wise form (3.3) but further decomposing the block-wise subproblems in the parallel way of (3.4). In this way, the advantage of the ADMM-GBS such as its efficiency and stability is preserved among blocks while the parallel computation to tackle big-data scenarios is applicable within each block. This is the main motivation of the new algorithm to be proposed.

3.3 Further Remarks

We have emphasized the importance of parallel computation to tackle the big-data scenarios of the model (1.1). One may ask why not just implement the full parallel decomposition directly to the ALM (1.6) and thus obtain the following scheme whose m x_i-subproblems can be solved fully in parallel:

$$\begin{cases} x_1^{k+1} = \arg\min\{\mathcal{L}_\beta^m(x_1, x_2^k, \cdots, x_m^k, \lambda^k) \mid x_1 \in X_1\}, \\ \qquad \vdots \\ x_i^{k+1} = \arg\min\{\mathcal{L}_\beta^m(x_1^k, \cdots, x_{i-1}^k, x_i, x_{i+1}^k, \cdots, x_m^k, \lambda^k) \mid x_i \in X_i\}, \\ \qquad \vdots \\ x_m^{k+1} = \arg\min\{\mathcal{L}_\beta^m(x_1^k, \cdots, x_{m-1}^k, x_m, \lambda^k) \mid x_m \in X_m\}, \\ \lambda^{k+1} = \lambda^k - \beta(\sum_{i=1}^m A_i x_i^{k+1} - b). \end{cases} \tag{3.5}$$

In fact, the scheme (3.5) requires m work stations to realize the parallel computation. When m is huge for a big-data scenario, it might be too expensive to be practical. Moreover, from methodological point of view, as shown in [13], the scheme (3.5) is not necessarily convergent even for $m = 2$. Later, it was shown in [18] that the convergence of (3.5) can be guaranteed if all the x_i-subproblems are proximally regularized by certain proximal term

$$\begin{cases} x_1^{k+1} = \arg\min\{\mathcal{L}_\beta^m(x_1, x_2^k, \cdots, x_m^k, \lambda^k) + \frac{s\beta}{2}\|A_1(x_1 - x_1^k)\|^2 \mid x_1 \in X_1\}, \\ \qquad \vdots \\ x_i^{k+1} = \arg\min\{\mathcal{L}_\beta^m(x_1^k, \cdots, x_{i-1}^k, x_i, x_{i+1}^k, \cdots, x_m^k, \lambda^k) + \frac{s\beta}{2}\|A_i(x_i - x_i^k)\|^2 \mid x_i \in X_i\}, \\ \qquad \vdots \\ x_m^{k+1} = \arg\min\{\mathcal{L}_\beta^m(x_1^k, \cdots, x_{m-1}^k, x_m, \lambda^k) + \frac{s\beta}{2}\|A_m(x_m - x_m^k)\|^2 \mid x_m \in X_m\}, \\ \lambda^{k+1} = \lambda^k - \beta(\sum_{i=1}^m A_i x_i^{k+1} - b). \end{cases} \tag{3.6}$$

where the proximal parameter s is required to be greater than $m-1$. The x_i-subproblems in the scheme (3.6) are also eligible for parallel computation. But recall that we are considering the big-data scenarios where m is huge. Thus, the proximal terms in (3.6) with $s \geq m-1$ play a dominate role in the objective functions and the convergence is doomed to be extremely slow due to the huge value of $m-1$, though the convergence can be guaranteed asymptotically. Therefore, we do not expect that the existing schemes based on the technique of directly decomposing the ALM (1.6) in a parallel way are applicable for the big-data scenarios of (1.1) with a huge m. Note that in [12, 13], it was also suggested to correct the output of (3.5) by certain correction steps and the proximal terms are not needed to regularize the decomposed subproblems. But these schemes also require m work stations to realize the parallel computation.

3.4 The New Algorithm

Based on the previous discussion, we now propose the new algorithm which embeds the parallel computation (1.3) into the block-wise ADMM-GBS (3.3). As analyzed in [14, 22], if we replace the $\boldsymbol{x}_r$-subproblems in (3.3) directly by the further decomposed subproblems in (3.4), the convergence is not guaranteed. In fact, the proximity to the last iterate should be controlled when solving the further subproblems in (3.4). Therefore, we should embed not the subproblems in (3.4), but their

regularized counterparts:

$$x_{r_j}^{k+1} = \arg\min\left\{ \begin{array}{c} \mathcal{L}_\beta^t(\boldsymbol{x}_1^{k+1}, \cdots, \boldsymbol{x}_{r-1}^{k+1}, x_{r_1}^k, \cdots, x_{r_j-1}^k, x_{r_j}, x_{r_j+1}^k, \cdots, x_{r_{m_r}}^k, \boldsymbol{x}_{r+1}^k, \cdots, \boldsymbol{x}_t^k, \lambda^k) \\ +\frac{\tau_r\beta}{2}\|A_{r_j}(x_{r_j} - x_{r_j}^k)\|^2 \end{array} \middle| x_{r_j} \in X_{r_j} \right\} \tag{3.7}$$

with τ_r $(r = 1, \cdots, t)$ into the block-wise ADMM-GBS (3.3). By defining a matrix

$$\mathcal{P} = \begin{pmatrix} I & 0 & 0 & \cdots & 0 & 0 \\ 0 & I & D_2^{-1}\mathcal{A}_2^T\mathcal{A}_3 & \cdots & D_2^{-1}\mathcal{A}_2^T\mathcal{A}_t & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & D_{t-1}^{-1}\mathcal{A}_{t-1}^T\mathcal{A}_t & 0 \\ 0 & 0 & \cdots & 0 & I & 0 \\ 0 & 0 & \cdots & 0 & 0 & I_\ell \end{pmatrix}, \tag{3.8}$$

where D_r is defined in (2.7), we summarize the resulting algorithm as follows.

Algorithm 1: A splitting version of the block-wise ADMM-GBS (3.3) for (1.1)

Initialization: Specify a regrouping for the model (1.1) with determined values of t and m_r for $r = 1, 2, \cdots, t$. Choose constants τ_r such that $\tau_r \geq m_r - 1$ for $r = 1, \ldots, t$. Let $\mathcal{P}$ be defined in (3.8). With the given iterate $\boldsymbol{w}^k = (x_1^k, x_2^k, \cdots, x_m^k, \lambda^k) \in X_1 \times X_2 \times \cdots \times X_m \times \mathbb{R}^\ell$, the new iterate is generated by the following steps.

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{for } r = 1, 2, \ldots t, \text{ do:} \\ \quad \text{for } j = 1, \ldots m_r, \text{ parallel do:} \\ \quad\quad \bar{x}_{r_j}^{k+1} = \arg\min\left\{ \begin{array}{c} \mathcal{L}_\beta^t\big(\bar{\boldsymbol{x}}_1^{k+1}, \ldots, \bar{\boldsymbol{x}}_{r-1}^{k+1}, x_{r_1}^k, \ldots, x_{r_j-1}^k, x_{r_j}, x_{r_j+1}^k, \ldots, x_{r_{m_r}}^k, \\ \boldsymbol{x}_{r+1}^k, \ldots, \boldsymbol{x}_t^k, \lambda^k\big) + \frac{\tau_r\beta}{2}\|A_{r_j}(x_{r_j} - x_{r_j}^k)\|^2 \end{array} \middle| x_{r_j} \in X_{r_j} \right\}; \\ \quad \text{end.} \\ \text{end.} \\ \bar{\lambda}^{k+1} = \lambda^k - \beta\big(\sum_{r=1}^t \mathcal{A}_r\bar{\boldsymbol{x}}_r^{k+1} - b\big). \end{array} \right. \\ \mathcal{P}(\boldsymbol{w}^{k+1} - \boldsymbol{w}^k) = \alpha(\bar{\boldsymbol{w}}^{k+1} - \boldsymbol{w}^k), \quad \alpha \in (0, 1). \end{array} \right. \tag{3.9}$$

Remark 3.1. To implement the proposed algorithm (3.9), at most $\max\{m_1, \cdots, m_t\}$ work stations are needed. Also, the proximal parameters τ_r is only dependent on the number of variables m_r of the r-th group; they thus can be significantly smaller than $m - 1$ as required in (3.6). This feature thus can avoid slow convergence due to too large proximal coefficients. Certainly, when a specific application of the abstract model (1.1) is considered, the user can optimally determine the values of t and m_r for $r = 1, 2, \cdots, t$, so that the balance among the sequential and parallel computation is achieved and the optimal overall performance is achieved. But in this paper, we focus on the general methodology of algorithmic design for the generic case of (1.1), and do not discuss the specific regrouping strategies among variables which are case-dependent.

Remark 3.2. It is easy to see that at each iteration, the new algorithm (3.9) mainly requires solving m subproblems in form of (1.3). We use the proximal terms $\frac{\tau_r\beta}{2}\|A_{r_j}(x_{r_j} - x_{r_j}^k)\|^2$ to regularize the further decomposed subproblems in (3.9). But, just like the analysis in [21], we can instead

use the terms $\frac{\tau_r\beta}{2}\|x_{r_j} - x_{r_j}^k\|^2$, or more generally $\frac{\tau_r\beta}{2}\|x_{r_j} - x_{r_j}^k\|_G^2$ with a positive definite matrix G. Therefore, for the case where A_i is not the identity matrix while θ_i is simple in the sense that its proximal operator defined in (1.2) has a closed-form representation, then we can easily further consider linearizing the quadratic term in its corresponding subproblem in (3.9) and thus propose a linearized version of the algorithm (3.9). The corresponding analysis is not much different from our analysis to be presented. We thus will only briefly discuss the linearized version in Section 8; and mainly focus on the discussion for the scheme (3.9) for the purpose of exposing our main idea with easier notation.

4 Convergence

In this section, we prove the global convergence for the proposed algorithm (3.9).

4.1 Some Matrices

First of all, for the convenience of analysis, let us define some matrices and prove some useful properties for these matrices. Let

$$Q = \begin{pmatrix} \beta(D_1 - \mathcal{A}_1^T\mathcal{A}_1) & 0 & \cdots & \cdots & 0 & 0 \\ 0 & \beta D_2 & \ddots & & \vdots & \vdots \\ 0 & \beta\mathcal{A}_3^T\mathcal{A}_2 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \beta\mathcal{A}_t^T\mathcal{A}_2 & \cdots & \beta\mathcal{A}_t^T\mathcal{A}_{t-1} & \beta D_t & 0 \\ 0 & -\mathcal{A}_2 & \cdots & -\mathcal{A}_{t-1} & -\mathcal{A}_t & \frac{1}{\beta}I \end{pmatrix}, \tag{4.1}$$

where $\mathcal{A}_r$ and D_r are defined in (1.9) and (2.7), respectively.

In fact, the matrix Q in (4.1) can be written as the block-wise form

$$Q = \begin{pmatrix} \beta(D_1 - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & \beta\mathcal{Q}_e & 0 \\ 0 & -\mathcal{A} & \frac{1}{\beta}I \end{pmatrix}, \tag{4.2}$$

with

$$\mathcal{A} = (\mathcal{A}_2, \ldots, \mathcal{A}_t) \tag{4.3}$$

and

$$\mathcal{Q}_e = \begin{pmatrix} D_2 & 0 & \cdots & 0 \\ \mathcal{A}_3^T\mathcal{A}_2 & D_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \mathcal{A}_t^T\mathcal{A}_2 & \cdots & \mathcal{A}_t^T\mathcal{A}_{t-1} & D_t \end{pmatrix}. \tag{4.4}$$

Moreover, we use $\mathcal{D}_e$ to denote the diagonal part of $\mathcal{Q}_e$, *i.e.*,

$$\mathcal{D}_e = \begin{pmatrix} D_2 & 0 & \cdots & 0 \\ 0 & D_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & D_t \end{pmatrix}. \tag{4.5}$$

With the just defined matrices $\mathcal{A}$, $\mathcal{Q}_e$, and $\mathcal{D}_e$, we further define

$$M = \begin{pmatrix} I & 0 & 0 \\ 0 & \mathcal{Q}_e^{-T}\mathcal{D}_e & 0 \\ 0 & -\beta\mathcal{A} & I \end{pmatrix}. \tag{4.6}$$

These matrices will help us present the coming analysis more succinctly.

Indeed, proving the convergence for the proposed algorithm (3.9) crucially depends on some important properties of the just defined matrices. We summarize them in the following two lemmas.

Lemma 4.1. *For the matrices $\mathcal{A}$, $\mathcal{Q}_e$ and $\mathcal{D}_e$ which are defined in* (4.3), (4.4) *and* (4.5), *respectively, we have*

$$\mathcal{Q}_e^T + \mathcal{Q}_e \begin{cases} \succeq \mathcal{D}_e + \mathcal{A}^T\mathcal{A}, & \tau_r \geq m_r - 1,\ r = 1, \ldots, t; \\ \succ \mathcal{D}_e + \mathcal{A}^T\mathcal{A}, & \tau_r > m_r - 1,\ r = 1, \ldots, t. \end{cases} \tag{4.7}$$

Proof. Using the structure of the matrices $\mathcal{Q}_e$ and $\mathcal{D}_e$ (see (4.4) and (4.5)), we obtain

$$\mathcal{Q}_e^T + \mathcal{Q}_e = \mathcal{D}_e + \begin{pmatrix} D_2 & \mathcal{A}_2^T\mathcal{A}_3 & \cdots & \mathcal{A}_2^T\mathcal{A}_t \\ \mathcal{A}_3^T\mathcal{A}_2 & D_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathcal{A}_{t-1}^T\mathcal{A}_t \\ \mathcal{A}_t^T\mathcal{A}_2 & \cdots & \mathcal{A}_t^T\mathcal{A}_{t-1} & D_t \end{pmatrix}.$$

Since we choose $\tau_r \geq$ (resp. $>$) $m_r - 1$, it follows that

$$D_r = (\tau_r + 1)\mathrm{diag}(\mathcal{A}_r^T\mathcal{A}_r) \succeq (\text{Resp., } \succ)\mathcal{A}_r^T\mathcal{A}_r, \quad r = 1, \ldots, t,$$

and consequently,

$$\begin{pmatrix} D_2 & \mathcal{A}_2^T\mathcal{A}_3 & \cdots & \mathcal{A}_2^T\mathcal{A}_t \\ \mathcal{A}_3^T\mathcal{A}_2 & D_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathcal{A}_{t-1}^T\mathcal{A}_t \\ \mathcal{A}_t^T\mathcal{A}_2 & \cdots & \mathcal{A}_t^T\mathcal{A}_{t-1} & D_t \end{pmatrix} \succeq (\text{Resp., } \succ)\mathcal{A}^T\mathcal{A}.$$

The assertions (4.7) are followed immediately. □

Lemma 4.2. *For the matrices Q and M defined in* (4.1) *and* (4.6), *respectively, let*

$$H := QM^{-1} \tag{4.8a}$$

and

$$G := Q^T + Q - \alpha M^T H M. \tag{4.8b}$$

Then, we have the following conclusions.

1. *The matrix H defined in* (4.8a) *is symmetric and positive definite.*
2. *For the matrix G defined in* (4.8b), *we have*

$$G = Q^T + Q - \alpha M^T H M \begin{cases} \begin{cases} \succ 0, & \forall \alpha \in (0,1), \\ \succeq 0, & \alpha = 1, \end{cases} & \text{if} \quad \tau_r \geq m_r - 1,\ r = 1, \ldots, t; \\ \succ 0, \quad \forall\, \alpha \in (0,1], & \text{if} \quad \tau_r > m_r - 1,\ r = 1, \ldots, t. \end{cases} \tag{4.9}$$

Proof. First, we check the positive definiteness of the matrix H. For the matrix M defined in (4.6), we have

$$M^{-1} = \begin{pmatrix} I & 0 & 0 \\ 0 & \mathcal{D}_e^{-1}\mathcal{Q}_e^T & 0 \\ 0 & \beta\mathcal{A}\mathcal{D}_e^{-1}\mathcal{Q}_e^T & I \end{pmatrix}.$$

Thus, according to the definition of the matrix H (see (4.8a)), we conclude that

$$\begin{aligned} H &= QM^{-1} = \begin{pmatrix} \beta(D_1 - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & \beta\mathcal{Q}_e & 0 \\ 0 & -\mathcal{A} & \frac{1}{\beta}I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & \mathcal{D}_e^{-1}\mathcal{Q}_e^T & 0 \\ 0 & \beta\mathcal{A}\mathcal{D}_e^{-1}\mathcal{Q}_e^T & I \end{pmatrix} \\ &= \begin{pmatrix} \beta(D_1 - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & \beta\mathcal{Q}_e\mathcal{D}_e^{-1}\mathcal{Q}_e^T & 0 \\ 0 & 0 & \frac{1}{\beta}I \end{pmatrix}, \end{aligned}$$

is symmetric and positive definite.

Now, we turn to check the positive definiteness of the matrix G. Note that

$$Q^T+Q = \begin{pmatrix} 2\beta(D_1 - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & \beta(\mathcal{Q}_e^T + \mathcal{Q}_e) & -\mathcal{A}^T \\ 0 & -\mathcal{A} & \frac{2}{\beta}I \end{pmatrix} \overset{(4.7)}{\succeq} \begin{pmatrix} 2\beta(D_1 - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & \beta(\mathcal{D}_e + \mathcal{A}^T\mathcal{A}) & -\mathcal{A}^T \\ 0 & -\mathcal{A} & \frac{2}{\beta}I \end{pmatrix}$$

and

$$\begin{aligned} M^THM &= Q^TM = \begin{pmatrix} \beta(D_1 - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & \beta\mathcal{Q}_e^T & -\mathcal{A}^T \\ 0 & 0 & \frac{1}{\beta}I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & \mathcal{Q}_e^{-T}\mathcal{D}_e & 0 \\ 0 & -\beta\mathcal{A} & I \end{pmatrix} \\ &= \begin{pmatrix} \beta(D_1 - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & \beta(\mathcal{D}_e + \mathcal{A}^T\mathcal{A}) & -\mathcal{A}^T \\ 0 & -\mathcal{A} & \frac{1}{\beta}I \end{pmatrix}. \end{aligned} \tag{4.10}$$

From the definition of G (see (4.8b) and the two different cases of (4.7)), it follows that

$$\begin{aligned} G &= Q^T + Q - \alpha M^THM \\ &\begin{pmatrix} \succeq \\ \succ \end{pmatrix} \begin{pmatrix} (2-\alpha)\beta(D_1 - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + (1-\alpha) \begin{pmatrix} 0 & 0 & 0 \\ 0 & \beta(\mathcal{D}_e + \mathcal{A}^T\mathcal{A}) & -\mathcal{A}^T \\ 0 & -\mathcal{A} & \frac{1}{\beta}I \end{pmatrix} \\ &\succeq 0. \end{aligned}$$

The assertion (4.9) is proved. □

As we shall see, Lemmas 4.3 and 4.2 actually play a very important role in proving the convergence for the proposed algorithm (3.9).

4.2 A Prediction-Correction Reformulation of (3.9)

Now, with the matrices introduced in the last subsection, we can rewrite the proposed algorithm (3.9) as the following prediction-correction form.

Prediction. For the given $\boldsymbol{w}^k = (x_1^k, x_2^k \ldots, x_m^k, \lambda^k) = (\boldsymbol{x}_1^k, \ldots, \boldsymbol{x}_t^k, \lambda^k)$, generate the predictor $\tilde{\boldsymbol{w}}^k = (\tilde{x}_1^k, \tilde{x}_2^k \ldots, \tilde{x}_m^k, \tilde{\lambda}^k) = (\tilde{\boldsymbol{x}}_1^k, \ldots, \tilde{\boldsymbol{x}}_t^k, \tilde{\lambda}^k)$ by the following steps:

$$\left\{\begin{array}{l} \text{for } r = 1, 2, \ldots t, \text{ do:} \\ \quad \text{for } j = 1, \ldots m_r, \text{ parallel do:} \\ \qquad \tilde{x}_{r_j}^k = \arg\min\left\{ \begin{array}{c} \mathcal{L}_\beta^t\big(\tilde{\boldsymbol{x}}_1^k, \ldots, \tilde{\boldsymbol{x}}_{r-1}^k, x_{r_1}^k, \ldots, x_{r_{j-1}}^k, x_{r_j}, x_{r_{j+1}}^k, \ldots, x_{r_{m_r}}^k, \\ \boldsymbol{x}_{r+1}^k, \ldots, \boldsymbol{x}_t^k, \lambda^k\big) + \frac{\tau_r\beta}{2}\|A_{r_j}(x_{r_j} - x_{r_j}^k)\|^2 \end{array} \middle| \, x_{r_j} \in X_{r_j} \right\}; \\ \quad \text{end.} \\ \text{end.} \end{array}\right. \tag{4.11a}$$

Additionally, we define

$$\tilde{\lambda}^k = \lambda^k - \beta\big(\mathcal{A}_1\tilde{\boldsymbol{x}}_1^k + \textstyle\sum_{j=2}^t \mathcal{A}_j\boldsymbol{x}_j^k - b\big). \tag{4.11b}$$

Correction. The new iterate $\boldsymbol{w}^{k+1}$ is given by

$$\boldsymbol{w}^{k+1} = \boldsymbol{w}^k - \alpha M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k), \tag{4.12a}$$

where $\tilde{\boldsymbol{w}}^k$ is the predictor generated by (4.11), the matrix M is defined in (4.6) and

$$\alpha \in \left\{ \begin{array}{ll} (0,1), & \text{if} \quad \tau_r \geq m_r - 1, \ r = 1, \ldots, t; \\ (0,1], & \text{if} \quad \tau_r > m_r - 1, \ r = 1, \ldots, t. \end{array} \right. \tag{4.12b}$$

As mentioned in [22], we conduct the convergence analysis in the context of the prediction-correction form (4.11)-(4.12) because the proof of the convergence is essentially to prove the contraction property with respect to the solution set, while the progress of the proximity to the solution set is measured by the quantity $\|\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k\|_G^2$ where G is defined in (4.8b). Thus, it is convenient to explicitly analyze the predictor $\tilde{\boldsymbol{w}}^k$ and accordingly revisit the algorithm (3.9) from the prediction-correction perspective. The other reason is that this prediction-correction reformulation enables us to investigate the relationship between the proposed algorithm (3.9) and some existing schemes in the literature by a unified framework, as elaborated in Sections 6.1 and 6.2.

Let us take a closer look at the correction step (4.12). Recall that the matrix M defined in (4.6) and the matrices $\mathcal{Q}_e$, $\mathcal{D}_e$ in M are defined in (4.4) and (4.5), respectively. Moreover, using (4.3) and (4.11b), we can see that the correction step (4.12) consists of the following computations:

$$\left\{ \begin{array}{l} \boldsymbol{x}_1^{k+1} - \boldsymbol{x}_1^k = \alpha(\tilde{\boldsymbol{x}}_1^k - \boldsymbol{x}_1^k), \\ \mathcal{D}_e^{-1}\mathcal{Q}_e^T \begin{pmatrix} \boldsymbol{x}_2^{k+1} - \boldsymbol{x}_2^k \\ \vdots \\ \boldsymbol{x}_t^{k+1} - \boldsymbol{x}_t^k \end{pmatrix} = \alpha \begin{pmatrix} \tilde{\boldsymbol{x}}_2^k - \boldsymbol{x}_2^k \\ \vdots \\ \tilde{\boldsymbol{x}}_t^k - \boldsymbol{x}_t^k \end{pmatrix}, \\ \lambda^{k+1} = \lambda^k - \alpha\beta\big(\sum_{s=1}^t \mathcal{A}_s\tilde{\boldsymbol{x}}_s^k - b\big). \end{array} \right. \tag{4.13}$$

Notice that $\mathcal{D}_e^{-1}\mathcal{Q}_e^T$ is a block-wise upper-triangular matrix whose diagonal parts are identities. Thus, the block-wise variables $(\boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_t)$ are updated consecutively in the back substitution order:

$\boldsymbol{x}_t^{k+1} \to \boldsymbol{x}_{t-1}^{k+1} \to \cdots\cdots \to \boldsymbol{x}_2^{k+1}$. Recall that within each block variable, the further decomposed subproblems are eligible for parallel computation. Thus, The correction step (4.12) can be viewed as a Gaussian back substitution procedure to correct the output of (4.11).

Now, let us come back to the prediction step (4.11). In the following lemma, we analyze the optimality conditions of the $\tilde{x}_{r_j}$-subproblems in (4.11) and represent the predictor generated by (4.11) as a VI reformulation. This VI reformulation helps us better discern its difference from the VI characterization (2.4) of the original model (1.1); and thus clearly see how far the predictor $\tilde{w}^k$ is from a solution point. It also inspires the correction step (4.12).

Lemma 4.3. *Let $\tilde{\boldsymbol{x}}^k$ be generated by* (4.11a) *from the given vector $\boldsymbol{w}^k$ and $\tilde{\lambda}^k$ be defined by* (4.11b). *Then, the predictor $\tilde{\boldsymbol{w}}^k \in \Omega$ satisfies*

$$\tilde{\boldsymbol{w}}^k \in \Omega,\ \ \vartheta(\boldsymbol{x}) - \vartheta(\tilde{\boldsymbol{x}}^k) + (\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T F(\tilde{\boldsymbol{w}}^k) \geq (\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T Q(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k),\ \ \forall \boldsymbol{w} \in \Omega, \tag{4.14}$$

where Q is defined in (4.1).

Proof. Using the notation of the augmented Lagrangian function (see (1.5)), we observe the optimality condition of the x_{r_j}-subproblem in the r-th group of (4.11a) for $r = 1, \ldots, t$. Ignoring some constant terms in the objective function of the subproblems, we have

$$\begin{aligned}
\tilde{x}_{r_j}^k &= \arg\min\left\{ \begin{array}{c} \mathcal{L}_\beta^t\big(\tilde{\boldsymbol{x}}_1^k, \ldots, \tilde{\boldsymbol{x}}_{r-1}^k, x_{r_1}^k, \ldots, x_{r_{j-1}}^k, x_{r_j}, x_{r_{j+1}}^k, \ldots, x_{r_{m_r}}^k, \\ \boldsymbol{x}_{r+1}^k, \ldots, \boldsymbol{x}_t^k, \lambda^k\big) + \frac{\tau_r\beta}{2}\|A_{r_j}(x_{r_j} - x_{r_j}^k)\|^2 \end{array} \ \Big| \ x_{r_j} \in X_{r_j} \right\} \\
&\overset{(1.5)}{=} \arg\min\left\{ \begin{array}{c} \theta_{r_j}(x_{r_j}) - (\lambda^k)^T A_{r_j} x_{r_j} + \frac{\beta}{2}\|A_{r_j}(x_{r_j} - x_{r_j}^k) + \sum_{s=1}^{r-1} \mathcal{A}_s \tilde{\boldsymbol{x}}_s^k \\ + \sum_{s=r}^{t} \mathcal{A}_s \boldsymbol{x}_s^k - b\|^2 + \frac{\tau_r\beta}{2}\|A_{r_j}(x_{r_j} - x_{r_j}^k)\|^2 \end{array} \ \Big| \ x_{r_j} \in X_{r_j} \right\}.
\end{aligned}$$

The optimality condition of the above convex minimization problem is

$$\begin{aligned}
\tilde{x}_{r_j}^k \in X_{r_j},\ \ & \theta_{r_j}(x_{r_j}) - \theta_{r_j}(\tilde{x}_{r_j}^k) + (x_{r_j} - \tilde{x}_{r_j}^k)^T \big\{ -A_{r_j}^T \lambda^k \\
& + \beta A_{r_j}^T \big[\textstyle\sum_{s=1}^{r-1} \mathcal{A}_s \tilde{\boldsymbol{x}}_s^k + \sum_{s=r}^{t} \mathcal{A}_s \boldsymbol{x}_s^k - b \big] + (\tau_r + 1)\beta A_{r_j}^T A_{r_j}(\tilde{x}_{r_j}^k - x_{r_j}^k) \big\} \geq 0,\ \ \forall x_{r_j} \in X_{r_j}.
\end{aligned}$$

Using the definition of $\tilde{\lambda}^k$ (see (4.11b)), we have

$$\lambda^k = \tilde{\lambda}^k + \beta\big(\mathcal{A}_1 \tilde{\boldsymbol{x}}_1^k + \textstyle\sum_{s=2}^{t} \mathcal{A}_s \boldsymbol{x}_s^k - b\big).$$

Substituting it into the last inequality, we obtain

$$\begin{aligned}
\tilde{x}_{r_j}^k \in X_{r_j},\ \ & \theta_{r_j}(x_{r_j}) - \theta_{r_j}(\tilde{x}_{r_j}^k) + (x_{r_j} - \tilde{x}_{r_j}^k)^T \big\{ -A_{r_j}^T \tilde{\lambda}^k \\
& + \beta A_{r_j}^T \big[\textstyle\sum_{s=2}^{r-1} \mathcal{A}_s (\tilde{\boldsymbol{x}}_s^k - \boldsymbol{x}_s^k) \big] + (\tau_r + 1)\beta A_{r_j}^T A_{r_j}(\tilde{x}_{r_j}^k - x_{r_j}^k) \big\} \geq 0,\ \ \forall x_{r_j} \in X_{r_j}.
\end{aligned}$$

Applying this inequality for the cases of $j = 1, \ldots, m_r$, and summarizing the resulting inequalities, we get

$$\begin{aligned}
\tilde{\boldsymbol{x}}_r^k \in \mathcal{X}_r,\ \ & \vartheta_r(\boldsymbol{x}_r) - \vartheta_r(\tilde{\boldsymbol{x}}_r^k) + (\boldsymbol{x}_r - \tilde{\boldsymbol{x}}_r^k)^T \big\{ -\mathcal{A}_r^T \tilde{\lambda}^k \\
& + \beta \mathcal{A}_r^T \big[\textstyle\sum_{s=2}^{r-1} \mathcal{A}_s (\tilde{\boldsymbol{x}}_s^k - \boldsymbol{x}_s^k) \big] + (\tau_r + 1)\beta \mathrm{diag}(\mathcal{A}_r^T \mathcal{A}_r)(\tilde{\boldsymbol{x}}_r^k - \boldsymbol{x}_r^k) \big\} \geq 0,\ \forall \boldsymbol{x}_r \in \mathcal{X}_r.
\end{aligned} \tag{4.15}$$

Note that, for $r = 1$, (4.15) means that

$$\begin{aligned}
\tilde{\boldsymbol{x}}_1^k \in \mathcal{X}_1,\quad & \vartheta_1(\boldsymbol{x}_1) - \vartheta_1(\tilde{\boldsymbol{x}}_1^k) + (\boldsymbol{x}_1 - \tilde{\boldsymbol{x}}_1^k)^T \big\{ -\mathcal{A}_1^T \tilde{\lambda}^k \\
& - \beta \mathcal{A}_1^T \mathcal{A}_1 (\tilde{\boldsymbol{x}}_1^k - \boldsymbol{x}_1^k) + (\tau_1 + 1)\beta \mathrm{diag}(\mathcal{A}_1^T \mathcal{A}_1)(\tilde{\boldsymbol{x}}_1^k - \boldsymbol{x}_1^k) \big\} \geq 0, \quad \forall \boldsymbol{x}_1 \in \mathcal{X}_1,
\end{aligned}$$

Using the notation of matrix D_1 (see (2.7)), it can be written as

$$\tilde{\boldsymbol{x}}_1^k \in \mathcal{X}_1, \quad \vartheta_1(\boldsymbol{x}_1)-\vartheta_1(\tilde{\boldsymbol{x}}_1^k)+(\boldsymbol{x}_1-\tilde{\boldsymbol{x}}_1^k)^T\{-\mathcal{A}_1^T\tilde{\lambda}^k+\beta(D_1-\mathcal{A}_1^T\mathcal{A}_1)(\tilde{\boldsymbol{x}}_1^k-\boldsymbol{x}_1^k)\} \geq 0, \quad \forall \boldsymbol{x}_1 \in \mathcal{X}_1. \tag{4.16}$$

In addition, by using (4.11b), we have

$$(\sum_{r=1}^{t}\mathcal{A}_r\tilde{\boldsymbol{x}}_r^k - b) - \sum_{s=2}^{t}\mathcal{A}_s(\tilde{\boldsymbol{x}}_s^k - \boldsymbol{x}_s^k) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) = 0,$$

and it can be rewritten as

$$\tilde{\lambda}^k \in \mathbb{R}^\ell, \quad (\lambda - \tilde{\lambda}^k)^T\{(\sum_{r=1}^{t}\mathcal{A}_r\tilde{\boldsymbol{x}}_r^k - b) - \sum_{s=2}^{t}\mathcal{A}_s(\tilde{\boldsymbol{x}}_s^k - \boldsymbol{x}_s^k) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k)\} \geq 0, \quad \forall\, \lambda \in \mathbb{R}^\ell. \tag{4.17}$$

Combining (4.16), (4.15) ($r = 2, \ldots, t$) and (4.17) together and using the notations $F(\boldsymbol{w})$, Q and D_r (see (2.2), (4.1) and (2.7)), the assertion of this lemma is followed directly. □

Recall the VI reformulation (2.4a)-(2.4b) of the model (1.1). Lemma 4.3 thus indicates that the accuracy of the predictor $\tilde{\boldsymbol{w}}^k$ to a solution point $\boldsymbol{w}^*$ is measured by the quantity $\max\{(\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T Q(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) \mid \boldsymbol{w} \in \Omega\}$. This is also the reason we search for a better iterate at the correct step (4.12) along the direction $-(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)$ to further reduce the proximity and to guarantee that the whole sequence is monotonically closer to the solution set. With this strict contraction property, it becomes standard to prove the convergence from the contraction method perspective in [1].

4.3 An Illustrative Example of Lemma 4.3

For better understanding the proposed algorithm (3.9) and seeing the assertion in Lemma 4.3 more specifically, we consider the special case of (1.1) with $m = 6$:

$$\min\Big\{\sum_{i=1}^{6}\theta_i(x_i) \,\Big|\, \sum_{i=1}^{6}A_ix_i = b,\ x_i \in X_i,\ i = 1, 2, \cdots, 6\Big\};$$

and regroup the variables as

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \end{pmatrix} \quad \text{with} \quad \boldsymbol{x}_1 = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \boldsymbol{x}_2 = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} \quad \text{and} \quad \boldsymbol{x}_3 = \begin{pmatrix} x_5 \\ x_6 \end{pmatrix}. \tag{4.18a}$$

Therefore, $m_1 = m_2 = m_3 = 2$. Accordingly, we regroup

$$\mathcal{A}_1 = (A_1, A_2), \qquad \mathcal{A}_2 = (A_3, A_4), \qquad \mathcal{A}_3 = (A_5, A_6), \tag{4.18b}$$

and

$$\mathcal{X}_1 = X_1 \times X_2, \qquad \mathcal{X}_2 = X_3 \times X_4, \qquad \mathcal{X}_3 = X_5 \times X_6. \tag{4.18c}$$

The corresponding augmented Lagrangian function is

$$\mathcal{L}_\beta^6(x_1, x_2, x_3, x_4, x_5, x_6, \lambda) = \sum_{i=1}^{6}\theta_i(x_i) - \lambda^T(\sum_{i=1}^{6}A_ix_i - b) + \frac{\beta}{2}\Big\|\sum_{i=1}^{6}A_ix_i - b\Big\|^2. \tag{4.19}$$

With the given $\boldsymbol{w}^k = (x_1^k, x_2^k, x_3^k, x_4^k, x_5^k, x_6^k, \lambda^k)$, the prediction step (4.11) at the k-th iteration can be specified as

$$\begin{cases} \tilde{x}_1^k = \arg\min\big\{\mathcal{L}_\beta^6(x_1, x_2^k, x_3^k, x_4^k, x_5^k, x_6^k, \lambda^k) + \frac{\tau_1\beta}{2}\|A_1(x_1 - x_1^k)\|^2 \,\big|\, x_1 \in X_1\big\}, \\ \tilde{x}_2^k = \arg\min\big\{\mathcal{L}_\beta^6(x_1^k, x_2, x_3^k, x_4^k, x_5^k, x_6^k, \lambda^k) + \frac{\tau_1\beta}{2}\|A_2(x_2 - x_2^k)\|^2 \,\big|\, x_2 \in X_2\big\}; \end{cases} \tag{4.20a}$$

$$\begin{cases} \tilde{x}_3^k = \arg\min\{\mathcal{L}_\beta^6(\tilde{x}_1^k, \tilde{x}_2^k, x_3, x_4^k, x_5^k, x_6^k, \lambda^k) + \frac{\tau_2\beta}{2}\|A_3(x_3 - x_3^k)\|^2 \mid x_3 \in X_3\}, \\ \tilde{x}_4^k = \arg\min\{\mathcal{L}_\beta^6(\tilde{x}_1^k, \tilde{x}_2^k, x_3^k, x_4, x_5^k, x_6^k, \lambda^k) + \frac{\tau_2\beta}{2}\|A_4(x_4 - x_4^k)\|^2 \mid x_4 \in X_4\}; \end{cases} \tag{4.20b}$$

$$\begin{cases} \tilde{x}_5^k = \arg\min\{\mathcal{L}_\beta^6(\tilde{x}_1^k, \tilde{x}_2^k, \tilde{x}_3^k, \tilde{x}_4^k, x_5, x_6^k, \lambda^k) + \frac{\tau_3\beta}{2}\|A_5(x_5 - x_5^k)\|^2 \mid x_5 \in X_5\}, \\ \tilde{x}_6^k = \arg\min\{\mathcal{L}_\beta^6(\tilde{x}_1^k, \tilde{x}_2^k, \tilde{x}_3^k, \tilde{x}_4^k, x_5^k, x_6, \lambda^k) + \frac{\tau_3\beta}{2}\|A_6(x_6 - x_6^k)\|^2 \mid x_6 \in X_6\}; \end{cases} \tag{4.20c}$$

$$\tilde{\lambda}^k = \lambda^k - \beta\big(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k + \textstyle\sum_{j=3}^6 A_j x_j^k - b\big). \tag{4.20d}$$

By using (4.19), we derive the following optimal condition of the x_i-subproblems of (4.20a):

$$\begin{cases} \theta_1(x_1) - \theta_1(\tilde{x}_1^k) + (x_1 - \tilde{x}_1^k)^T\{-A_1^T\lambda^k + \beta A_1^T[A_1(\tilde{x}_1^k - x_1^k) + (\sum_{j=1}^6 A_j x_j^k - b)] + \tau_1\beta A_1^T A_1(\tilde{x}_1^k - x_1^k)\} \geq 0,\ \forall x_1 \in X_1; \\ \theta_2(x_2) - \theta_2(\tilde{x}_2^k) + (x_2 - \tilde{x}_2^k)^T\{-A_2^T\lambda^k + \beta A_2^T[A_2(\tilde{x}_2^k - x_2^k) + (\sum_{j=1}^6 A_j x_j^k - b)] + \tau_1\beta A_2^T A_2(\tilde{x}_2^k - x_2^k)\} \geq 0,\ \forall x_2 \in X_2. \end{cases}$$

Substituting $\lambda^k = \tilde{\lambda}^k + \beta\big(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k + \sum_{j=3}^6 A_j x_j^k - b\big)$ (see (4.20d)) into the last inequalities,

$$\begin{cases} \theta_1(x_1) - \theta_1(\tilde{x}_1^k) + (x_1 - \tilde{x}_1^k)^T\{-A_1^T\tilde{\lambda}^k - \beta A_1^T[A_1(\tilde{x}_1^k - x_1^k) + A_2(\tilde{x}_2^k - x_2^k)] + (\tau_1+1)\beta A_1^T A_1(\tilde{x}_1^k - x_1^k)\} \geq 0,\ \forall x_1 \in X_1; \\ \theta_2(x_2) - \theta_2(\tilde{x}_2^k) + (x_2 - \tilde{x}_2^k)^T\{-A_2^T\tilde{\lambda}^k - \beta A_2^T[A_1(\tilde{x}_1^k - x_1^k) + A_2(\tilde{x}_2^k - x_2^k)] + (\tau_1+1)\beta A_2^T A_2(\tilde{x}_2^k - x_2^k)\} \geq 0,\ \forall x_2 \in X_2. \end{cases}$$

Using the notations in (4.18), it can be written as

$$\vartheta(\boldsymbol{x}_1) - \vartheta(\tilde{\boldsymbol{x}}_1^k) + (\boldsymbol{x}_1 - \tilde{\boldsymbol{x}}_1^k)^T\{-\mathcal{A}_1^T\tilde{\lambda}^k - \beta\mathcal{A}_1^T\mathcal{A}_1(\tilde{\boldsymbol{x}}_1^k - \boldsymbol{x}_1^k) + (\tau_1+1)\beta\mathrm{diag}(\mathcal{A}_1^T\mathcal{A}_1)(\tilde{\boldsymbol{x}}_1^k - \boldsymbol{x}_1^k)\} \geq 0,\ \forall\, \boldsymbol{x}_1 \in \mathcal{X}_1. \tag{4.21}$$

For the $\boldsymbol{x}_2$-group,

$$\begin{cases} \theta_3(x_3) - \theta_3(\tilde{x}_3^k) + (x_3 - \tilde{x}_3^k)^T\{A_3^T[\beta(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k + \sum_{j=3}^6 A_j x_j^k - b) - \lambda^k] + (\tau_2+1)\beta A_3^T A_3(\tilde{x}_3^k - x_3^k)\} \geq 0,\ \forall x_3 \in X_3; \\ \theta_4(x_4) - \theta_4(\tilde{x}_4^k) + (x_4 - \tilde{x}_4^k)^T\{A_4^T[\beta(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k + \sum_{j=3}^6 A_j x_j^k - b) - \lambda^k] + (\tau_2+1)\beta A_4^T A_4(\tilde{x}_4^k - x_4^k)\} \geq 0,\ \forall x_4 \in X_4; \end{cases}$$

Substituting $\beta\big(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k + \sum_{j=3}^6 A_j x_j^k - b\big) - \lambda^k = -\tilde{\lambda}^k$ (see (4.20d)) into the last inequalities,

$$\begin{cases} \theta_3(x_3) - \theta_3(\tilde{x}_3^k) + (x_3 - \tilde{x}_3^k)^T\{-A_3^T\tilde{\lambda}^k + (\tau_2+1)\beta A_3^T A_3(\tilde{x}_3^k - x_3^k)\} \geq 0,\ \forall x_3 \in X_3; \\ \theta_4(x_4) - \theta_4(\tilde{x}_4^k) + (x_4 - \tilde{x}_4^k)^T\{-A_4^T\tilde{\lambda}^k + (\tau_2+1)\beta A_4^T A_4(\tilde{x}_4^k - x_4^k)\} \geq 0,\ \forall x_4 \in X_4; \end{cases}$$

Using the notation in (4.18), it can be rewritten as

$$\vartheta(\boldsymbol{x}_2) - \vartheta(\tilde{\boldsymbol{x}}_2^k) + (\boldsymbol{x}_2 - \tilde{\boldsymbol{x}}_2^k)^T\{-\mathcal{A}_2^T\tilde{\lambda}^k + (\tau_2+1)\beta\mathrm{diag}(\mathcal{A}_2^T\mathcal{A}_2)(\tilde{\boldsymbol{x}}_2^k - \boldsymbol{x}_2^k)\} \geq 0,\ \forall\, \boldsymbol{x}_2 \in \mathcal{X}_2. \tag{4.22}$$

For the $\boldsymbol{x}_3$-group,

$$\begin{cases} \theta_5(x_5) - \theta_5(\tilde{x}_5^k) + (x_5 - \tilde{x}_5^k)^T\{A_5^T[\beta(\sum_{i=1}^4 A_i\tilde{x}_i^k + A_5\tilde{x}_5^k + A_6 x_6^k - b) - \lambda^k] + \tau_3\beta A_5^T A_5(\tilde{x}_5^k - x_5^k)\} \geq 0,\ \forall x_5 \in X_5; \\ \theta_6(x_6) - \theta_6(\tilde{x}_6^k) + (x_6 - \tilde{x}_6^k)^T\{A_6^T[\beta(\sum_{i=1}^4 A_i\tilde{x}_i^k + A_5 x_5^k + A_6\tilde{x}_6^k - b) - \lambda^k] + \tau_3\beta A_6^T A_6(\tilde{x}_6^k - x_6^k)\} \geq 0,\ \forall x_6 \in X_6; \end{cases}$$

Substituting $\beta\big(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k + \sum_{j=3}^6 A_j x_j^k - b\big) - \lambda^k = -\tilde{\lambda}^k$ (see (4.20d)) into the last inequalities,

$$\begin{cases} \theta_5(x_5) - \theta_5(\tilde{x}_5^k) + (x_5 - \tilde{x}_5^k)^T\{-A_5^T\tilde{\lambda}^k + \beta A_5^T[A_3(\tilde{x}_3^k - x_3^k) + A_4(\tilde{x}_4^k - x_4^k)] + (\tau_3+1)\beta A_5^T A_5(\tilde{x}_5^k - x_5^k)\} \geq 0,\ \forall x_5 \in X_5; \\ \theta_6(x_6) - \theta_6(\tilde{x}_6^k) + (x_6 - \tilde{x}_6^k)^T\{-A_6^T\tilde{\lambda}^k + \beta A_6^T[A_3(\tilde{x}_3^k - x_3^k) + A_4(\tilde{x}_4^k - x_4^k)] + (\tau_3+1)\beta A_6^T A_6(\tilde{x}_6^k - x_6^k)\} \geq 0,\ \forall x_6 \in X_6; \end{cases}$$

Using the notation in (4.18), it can be rewritten as

$$\vartheta(\boldsymbol{x}_3) - \vartheta(\tilde{\boldsymbol{x}}_3^k) + (\boldsymbol{x}_3 - \tilde{\boldsymbol{x}}_3^k)^T\{-\mathcal{A}_3^T\tilde{\lambda}^k + \beta\mathcal{A}_3^T\mathcal{A}_2(\tilde{\boldsymbol{x}}_2^k - \boldsymbol{x}_2^k) + (\tau_3+1)\beta\mathrm{diag}(\mathcal{A}_3^T\mathcal{A}_3)(\tilde{\boldsymbol{x}}_3^k - \boldsymbol{x}_3^k)\} \geq 0,\ \forall\, \boldsymbol{x}_3 \in \mathcal{X}_3. \tag{4.23}$$

Using the notations in (4.18), we rewrite (4.20d) as

$$\tilde{\lambda}^k \in \mathbb{R}^\ell, \quad (\lambda - \tilde{\lambda}^k)^T\{(\sum_{r=1}^3 \mathcal{A}_r\tilde{\boldsymbol{x}}_r^k - b) - \mathcal{A}_2(\tilde{\boldsymbol{x}}_2^k - \boldsymbol{x}_2^k) - \mathcal{A}_3(\tilde{\boldsymbol{x}}_3^k - \boldsymbol{x}_3^k) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k)\} \geq 0, \quad \forall\, \lambda \in \mathbb{R}^\ell. \tag{4.24}$$

Combining (4.21), (4.22), (4.23) and (4.24) together, and using the VI (2.2), the predictor $\tilde{\boldsymbol{w}}^k \in \Omega$ satisfies (4.14) with the concrete matrix Q defined as

$$Q = \begin{pmatrix} (\tau_1+1)\beta\mathrm{diag}(\mathcal{A}_1^T\mathcal{A}_1) - \beta\mathcal{A}_1^T\mathcal{A}_1 & 0 & 0 & 0 \\ 0 & (\tau_2+1)\beta\mathrm{diag}(\mathcal{A}_2^T\mathcal{A}_2) & 0 & 0 \\ 0 & \beta\mathcal{A}_3^T\mathcal{A}_2 & (\tau_3+1)\beta\mathrm{diag}(\mathcal{A}_3^T\mathcal{A}_3) & 0 \\ 0 & -\mathcal{A}_2 & -\mathcal{A}_3 & \frac{1}{\beta}I \end{pmatrix}.$$

Therefore, for a given scenario of the abstract model (1.1) and when the regrouping strategy is determined, the matrix Q in (4.14) can be easily specified.

4.4 Convergence Proof

With the proved propositions, we are now ready to prove the convergence for the proposed algorithm (3.9). First of all, let us further analyze the term $(\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T Q(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)$ in the right-hand side of (4.14), which will help us show the strict contraction for the sequence $\{w^k\}$ generated by (3.9) with respect to the solution set Ω^*.

Theorem 4.4. *Let $\{\boldsymbol{w}^k\}$ be the sequence generated by the proposed algorithm (3.9). We have*

$$\begin{aligned} &\vartheta(\boldsymbol{x}) - \vartheta(\tilde{\boldsymbol{x}}^k) + (\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T F(\tilde{\boldsymbol{w}}^k) \\ &\quad \geq \frac{1}{2\alpha}\big(\|\boldsymbol{w} - \boldsymbol{w}^{k+1}\|_H^2 - \|\boldsymbol{w} - \boldsymbol{w}^k\|_H^2\big) + \frac{1}{2}\|\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k\|_G^2, \ \ \forall \boldsymbol{w} \in \Omega. \end{aligned} \tag{4.25}$$

Proof. First, it follows from (4.8a) that $Q = HM$. We thus have

$$(\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T Q(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) = \frac{1}{\alpha}(\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T H(\boldsymbol{w}^k - \boldsymbol{w}^{k+1}).$$

Together with (4.14), this identity means

$$\vartheta(\boldsymbol{x}) - \vartheta(\tilde{\boldsymbol{x}}^k) + (\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T F(\tilde{\boldsymbol{w}}^k) \geq \frac{1}{\alpha}(\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T H(\boldsymbol{w}^k - \boldsymbol{w}^{k+1}), \ \ \forall \boldsymbol{w} \in \Omega. \tag{4.26}$$

Applying the identity

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

to the therm $(\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T H(\boldsymbol{w}^k - \boldsymbol{w}^{k+1})$ in the right-hand side of (4.26) with

$$a = \boldsymbol{w}, \quad b = \tilde{\boldsymbol{w}}^k, \quad c = \boldsymbol{w}^k, \quad \text{and} \quad d = \boldsymbol{w}^{k+1},$$

we thus obtain

$$(\boldsymbol{w}-\tilde{\boldsymbol{w}}^k)^T H(\boldsymbol{w}^k-\boldsymbol{w}^{k+1}) = \frac{1}{2}\big(\|\boldsymbol{w}-\boldsymbol{w}^{k+1}\|_H^2 - \|\boldsymbol{w}-\boldsymbol{w}^k\|_H^2\big) + \frac{1}{2}\big(\|\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k\|_H^2 - \|\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^k\|_H^2\big). \tag{4.27}$$

For the last group term of (4.27), we have

$$\begin{aligned} &\|\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k\|_H^2 - \|\boldsymbol{w}^{k+1} - \tilde{\boldsymbol{w}}^k\|_H^2 \\ &\quad = \quad \|\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k\|_H^2 - \|(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) - (\boldsymbol{w}^k - \boldsymbol{w}^{k+1})\|_H^2 \\ &\quad \overset{(4.8a)}{=} \quad \|\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k\|_H^2 - \|(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) - \alpha M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)\|_H^2 \\ &\quad = \quad 2\alpha(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)^T HM(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) - \alpha^2(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)^T M^T HM(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) \\ &\quad = \quad \alpha(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)^T(Q^T + Q - \alpha M^T HM)(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) \\ &\quad \overset{(4.8b)}{=} \quad \alpha\|\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k\|_G^2. \end{aligned} \tag{4.28}$$

Substituting (4.27), (4.28) in (4.26), the assertion of this theorem is proved. □

Now, we are ready to show the strict contraction property of the sequence $\{\boldsymbol{w}^k\}$ generated by the proposed scheme (3.9).

Theorem 4.5. *Let $\{\boldsymbol{w}^k\}$ be the sequence generated by the proposed algorithm* (3.9)*. Then we have*

$$\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^*\|_H^2 \leq \|\boldsymbol{w}^k - \boldsymbol{w}^*\|_H^2 - \alpha\|\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k\|_G^2, \quad \forall \boldsymbol{w}^* \in \Omega^*. \tag{4.29}$$

Proof. Setting $\boldsymbol{w} = \boldsymbol{w}^*$ in (4.25), we get

$$\|\boldsymbol{w}^k - \boldsymbol{w}^*\|_H^2 - \|\boldsymbol{w}^{k+1} - \boldsymbol{w}^*\|_H^2 \geq \alpha\|\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k\|_G^2 + 2\alpha\{\vartheta(\tilde{\boldsymbol{x}}^k) - \vartheta(\boldsymbol{x}^*) + (\tilde{\boldsymbol{w}}^k - \boldsymbol{w}^*)^T F(\tilde{\boldsymbol{w}}^k)\}.$$

Using the optimality of $\boldsymbol{w}^*$ and the monotonicity of $F(\boldsymbol{w})$, we have

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

and thus

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

The assertion (4.29) follows directly. □

Finally, the convergence of $\{\boldsymbol{w}^k\}$ generated by the algorithm (3.9) can be proved easily. We summarize it in the following theorem.

Theorem 4.6. *The sequence $\{\boldsymbol{w}^k\}$ generated by the proposed algorithm (3.9) converges to a solution point of VI(Ω, F, θ).*

Proof. First, according to (4.29), it holds that $\{\boldsymbol{w}^k\}$ is bounded and

$$\lim_{k\to\infty} \|\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k\|_G = 0. \tag{4.30}$$

Thus, $\{\boldsymbol{w}^k\}$ (and $\{\tilde{\boldsymbol{w}}^k\}$) has a cluster point $\boldsymbol{w}^\infty$. Using $\boldsymbol{w}^\infty$ to start a new iteration, (4.14) becomes

$$\tilde{\boldsymbol{w}}^\infty \in \Omega, \quad \vartheta(\boldsymbol{x}) - \vartheta(\tilde{\boldsymbol{x}}^\infty) + (\boldsymbol{w} - \tilde{\boldsymbol{w}}^\infty)^T F(\tilde{\boldsymbol{w}}^\infty) \geq 0, \quad \forall \boldsymbol{w} \in \Omega,$$

and thus $\tilde{\boldsymbol{w}}^\infty$ is a solution of VI(Ω, F, θ). According to (4.29), the sequence $\{\boldsymbol{w}^k\}$ can not have another cluster point and it converges to $\tilde{\boldsymbol{w}}^\infty$. The proof is complete. □

5 Convergence Rate

In this section, we establish the $O(1/t)$ worst-case convergence rates measured by the iteration complexity in both the ergodic and nonergodic senses for the new algorithm (3.9), where t denotes the iteration counter. Recall the prediction-correction algorithm (4.11)-(4.12) is a reformulation of (3.9).

5.1 Convergence Rate in the Ergodic Sense

We first establish a worst-case $O(1/t)$ convergence rate for the scheme (3.9) in the ergodic sense. The proof is inspired by our earlier work in [19] for the ADMM (1.7).

For this convergence rate analysis, we need to recall a characterization of the solution set Ω^*, which is described in the following theorem. Its proof can be found in [6] (Theorem 2.3.5) or [19] (Theorem 2.1).

Theorem 5.1. *The solution set of VI(Ω, F, θ) is convex and it can be characterized as*

$$\Omega^* = \bigcap_{\boldsymbol{w}\in\Omega} \big\{\tilde{\boldsymbol{w}} \in \Omega : \big(\vartheta(\boldsymbol{x}) - \vartheta(\tilde{\boldsymbol{x}})\big) + (\boldsymbol{w} - \tilde{\boldsymbol{w}})^T F(\boldsymbol{w}) \geq 0\big\}. \tag{5.1}$$

Therefore, for given $\epsilon > 0$, $\tilde{\boldsymbol{w}} \in \Omega$ is called an ϵ-approximate solution of VI(Ω, F, θ) if it satisfies

$$\vartheta(\boldsymbol{x}) - \vartheta(\tilde{\boldsymbol{x}}) + (\boldsymbol{w} - \tilde{\boldsymbol{w}})^T F(\boldsymbol{w}) \geq -\epsilon, \ \ \forall\, \boldsymbol{w} \in \mathcal{D}_{(\tilde{\boldsymbol{w}})},$$

where

$$\mathcal{D}_{(\tilde{\boldsymbol{w}})} = \{\boldsymbol{w} \in \Omega \,|\, \|\boldsymbol{w} - \tilde{\boldsymbol{w}}\| \leq 1\}.$$

We refer the reader to [27] ((2.5) therein) for the definition of an ϵ-approximate solution using the above set.

In the following, we shall show that based on t iterations generated by the proposed algorithm (3.9), we can find $\tilde{\boldsymbol{w}} \in \Omega$ such that

$$\tilde{\boldsymbol{w}} \in \Omega \quad \text{and} \quad \sup_{\boldsymbol{w}\in\mathcal{D}_{(\tilde{\boldsymbol{w}})}} \big\{\vartheta(\tilde{\boldsymbol{x}}) - \vartheta(\boldsymbol{x}) + (\tilde{\boldsymbol{w}} - \boldsymbol{w})^T F(\boldsymbol{w})\big\} \leq \epsilon. \tag{5.2}$$

with $\epsilon = O(1/t)$. Theorem 4.4 is also the basis for the coming analysis about the worst-case convergence rate.

Note that it follows from the monotonicity of F that

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

Substituting it into (4.25), we obtain

$$\vartheta(\boldsymbol{x}) - \vartheta(\tilde{\boldsymbol{x}}^k) + (\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T F(\boldsymbol{w}) + \frac{1}{2\alpha}\|\boldsymbol{w} - \boldsymbol{w}^k\|_H^2 \geq \frac{1}{2\alpha}\|\boldsymbol{w} - \boldsymbol{w}^{k+1}\|_H^2, \ \forall \boldsymbol{w} \in \Omega. \tag{5.3}$$

Note that the above assertion hold whenever $G \succeq 0$.

Theorem 5.2. *Let $\{\boldsymbol{w}^k\}$ be generated by the proposed algorithm (3.9) and $\{\tilde{\boldsymbol{w}}^k\}$ be defined in* (4.11)*. For any integer $t > 0$, let $\tilde{\boldsymbol{w}}_t$ be defined as*

$$\tilde{\boldsymbol{w}}_t = \frac{1}{t+1}\sum_{k=0}^{t} \tilde{\boldsymbol{w}}^k. \tag{5.4}$$

Then, we have $\tilde{\boldsymbol{w}}_t \in \Omega$ and

$$\vartheta(\tilde{\boldsymbol{x}}_t) - \vartheta(\boldsymbol{x}) + (\tilde{\boldsymbol{w}}_t - \boldsymbol{w})^T F(\boldsymbol{w}) \leq \frac{1}{2\alpha(t+1)}\|\boldsymbol{w} - \boldsymbol{w}^0\|_H^2, \quad \forall \boldsymbol{w} \in \Omega. \tag{5.5}$$

Proof. First, it holds that $\tilde{\boldsymbol{w}}^k \in \Omega$ for all $k \geq 0$. Together with the convexity of $\mathcal{X}$ and $\mathbb{R}^\ell$, (5.4) implies that $\tilde{\boldsymbol{w}}_t \in \Omega$. Applying (5.3) to the cases with $k = 0, 1, \ldots, t$, and adding all the resulting inequalities together, we obtain

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

Use the notation of $\tilde{\boldsymbol{w}}_t$, it can be written as

$$\frac{1}{t+1}\sum_{k=0}^{t} \vartheta(\tilde{\boldsymbol{x}}^k) - \vartheta(\boldsymbol{x}) + (\tilde{\boldsymbol{w}}_t - \boldsymbol{w})^T F(\boldsymbol{w}) \leq \frac{1}{2\alpha(t+1)}\|\boldsymbol{w} - \boldsymbol{w}^0\|_H^2, \quad \forall \boldsymbol{w} \in \Omega. \tag{5.6}$$

Since $\vartheta(\boldsymbol{x})$ is convex and

$$\tilde{\boldsymbol{x}}_t = \frac{1}{t+1}\sum_{k=0}^{t}\tilde{\boldsymbol{x}}^k,$$

we have that

$$\vartheta(\tilde{\boldsymbol{x}}_t) \le \frac{1}{t+1}\sum_{k=0}^{t}\vartheta(\tilde{\boldsymbol{x}}^k).$$

Substituting it in (5.6), the assertion of this theorem follows directly. □

Recall (5.2). The conclusion (5.5) thus indicates that based on t iterations of the proposed algorithm (3.9), we can find an approximate solution of VI(Ω, F, θ) (i.e., $\tilde{\boldsymbol{w}}_t$ defined in (5.4)) with an accuracy of $O(1/t)$. That is, a worst-case $O(1/t)$ convergence rate is established for the proposed algorithm (3.9) in the ergodic sense.

5.2 Convergence Rate in a Nonergodic Sense

In this subsection, we establish a worst-case $O(1/t)$ convergence rate in a nonergodic sense for the proposed algorithm (3.9). Note that in general a worst-case nonergodic convergence rate is stronger than the ergodic convergence rate. The proof is inspired by our earlier work in [20] for the ADMM (1.7). We first need to prove the following lemma.

Lemma 5.3. *For the sequences $\{\boldsymbol{w}^k\}$ and $\{\tilde{\boldsymbol{w}}^k\}$ generated by the proposed prediction-correction algorithm* (4.11)-(4.12)*, we have*

$$(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)^T M^T H M\{(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)-(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})\} \ge \frac{1}{2\alpha}\|(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)-(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})\|^2_{(Q^T+Q)}. \quad (5.7)$$

Proof. First, set $\boldsymbol{w} = \tilde{\boldsymbol{w}}^{k+1}$ in (4.14), we have

$$\vartheta(\tilde{\boldsymbol{x}}^{k+1}) - \vartheta(\tilde{\boldsymbol{x}}^k) + (\tilde{\boldsymbol{w}}^{k+1}-\tilde{\boldsymbol{w}}^k)^T F(\tilde{\boldsymbol{w}}^k) \ge (\tilde{\boldsymbol{w}}^{k+1}-\tilde{\boldsymbol{w}}^k)^T Q(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k). \quad (5.8)$$

Note that (4.14) is also true for $k := k+1$. Thus, we have

$$\vartheta(\boldsymbol{x}) - \vartheta(\tilde{\boldsymbol{x}}^{k+1}) + (\boldsymbol{w}-\tilde{\boldsymbol{w}}^{k+1})^T F(\tilde{\boldsymbol{w}}^{k+1}) \ge (\boldsymbol{w}-\tilde{\boldsymbol{w}}^{k+1})^T Q(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1}),\ \forall\, \boldsymbol{w}\in\Omega.$$

Setting $\boldsymbol{w} = \tilde{\boldsymbol{w}}^k$ in the above inequality, we obtain

$$\vartheta(\tilde{\boldsymbol{x}}^k) - \vartheta(\tilde{\boldsymbol{x}}^{k+1}) + (\tilde{\boldsymbol{w}}^k-\tilde{\boldsymbol{w}}^{k+1})^T F(\tilde{\boldsymbol{w}}^{k+1}) \ge (\tilde{\boldsymbol{w}}^k-\tilde{\boldsymbol{w}}^{k+1}) Q(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1}). \quad (5.9)$$

Adding (5.8) and (5.9), and using the monotonicity of F, we get

$$(\tilde{\boldsymbol{w}}^k-\tilde{\boldsymbol{w}}^{k+1})^T Q\{(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)-(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})\} \ge 0. \quad (5.10)$$

Further, adding the term

$$\{(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)-(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})\}^T Q\{(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)-(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})\}$$

to both sides of (5.10), and using $\boldsymbol{w}^T Q\boldsymbol{w} = \frac{1}{2}\boldsymbol{w}^T(Q^T+Q)\boldsymbol{w}$, we obtain

$$(\boldsymbol{w}^k-\boldsymbol{w}^{k+1})^T Q\{(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)-(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})\} \ge \frac{1}{2}\|(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)-(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})\|^2_{(Q^T+Q)}.$$

Substituting $(\boldsymbol{w}^k-\boldsymbol{w}^{k+1}) = \alpha M(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)$ into the left-hand side of the last inequality and using $Q = HM$, we obtain (5.7) and the lemma is proved. □

In the following theorem, we prove a key inequality for establishing the worst-case $O(1/t)$ convergence rate in a nonergodic sense for the proposed algorithm (3.9).

Theorem 5.4. *For the sequences $\{\boldsymbol{w}^k\}$ and $\{\tilde{\boldsymbol{w}}^k\}$ generated by the proposed prediction-correction algorithm* (4.11)-(4.12)*, we have*

$$\|M(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})\|_H \le \|M(\boldsymbol{w}^{k}-\tilde{\boldsymbol{w}}^{k})\|_H, \quad \forall\, k>0, \tag{5.11}$$

where M and H are defined in (4.6) and (4.8a), respectively.

Proof. Setting $a = M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)$ and $b = M(\boldsymbol{w}^{k+1} - \tilde{\boldsymbol{w}}^{k+1})$ in the identity

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

we obtain

$$\begin{aligned}&\|M(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)\|_H^2 - \|M(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})\|_H^2\\ &= \ 2(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)^T M^T H M[(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)-(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})] - \|M[(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)-(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})]\|_H^2.\end{aligned}$$

Inserting (5.7) into the first term of the right-hand side of the last equality, we obtain

$$\begin{aligned}&\|M(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)\|_H^2 - \|M(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})\|_H^2\\ &\ge \ \frac{1}{\alpha}\|(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)-(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})\|^2_{(Q^T+Q)} - \|M[(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)-(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})]\|_H^2\\ &\overset{(4.8b)}{=} \ \frac{1}{\alpha}\|(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)-(\boldsymbol{w}^{k+1}-\tilde{\boldsymbol{w}}^{k+1})\|^2_{G} \ge 0,\end{aligned}$$

where the last inequality is because of the positive definiteness of the matrix $(Q^T+Q)-\alpha M^T H M \succeq 0$. The assertion (5.11) follows immediately. □

Note that it follows from $G \succ 0$ and Theorem 4.5 that there exists a constant $c_0 > 0$ such that

$$\|\boldsymbol{w}^{k+1}-\boldsymbol{w}^*\|_H^2 \le \|\boldsymbol{w}^{k}-\boldsymbol{w}^*\|_H^2 - c_0\|M(\boldsymbol{w}^k-\tilde{\boldsymbol{w}}^k)\|_H^2, \quad \forall \boldsymbol{w}^* \in \Omega^*.$$

Since $\alpha M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) = (\boldsymbol{w}^k - \boldsymbol{w}^{k+1})$, we have a constant $c > 0$ such that

$$\|\boldsymbol{w}^{k+1}-\boldsymbol{w}^*\|_H^2 \le \|\boldsymbol{w}^{k}-\boldsymbol{w}^*\|_H^2 - c\|\boldsymbol{w}^k-\boldsymbol{w}^{k+1}\|_H^2, \quad \forall \boldsymbol{w}^* \in \Omega^*. \tag{5.12}$$

Now, with (5.12) and (5.11), we are ready to establish a worst-case $O(1/t)$ convergence rate in a nonergodic sense for the proposed algorithm (3.9).

Theorem 5.5. *Let $\{\boldsymbol{w}^k\}$ be the sequence generated by the proposed algorithm (3.9). For any integer $t > 0$, we have*

$$\|\boldsymbol{w}^{t}-\boldsymbol{w}^{t+1}\|_H^2 \le \frac{1}{(t+1)c}\|\boldsymbol{w}^0-\boldsymbol{w}^*\|_H^2, \quad \forall \boldsymbol{w}^* \in \Omega^*, \tag{5.13}$$

with a constant $c > 0$.

Proof. First, it follows from (5.12) that

$$\sum_{k=0}^{\infty} c\|\boldsymbol{w}^k-\boldsymbol{w}^{k+1}\|_H^2 \le \|\boldsymbol{w}^0-\boldsymbol{w}^*\|_H^2, \quad \forall\, \boldsymbol{w}^* \in \Omega^*. \tag{5.14}$$

According to Theorem 5.4, the sequence $\{\|\boldsymbol{w}^k - \boldsymbol{w}^{k+1}\|_H^2\}$ is monotonically non-increasing. Therefore, we have

$$(t+1)\|\boldsymbol{w}^{t}-\boldsymbol{w}^{t+1}\|_H^2 \le \sum_{k=0}^{t} \|\boldsymbol{w}^k-\boldsymbol{w}^{k+1}\|_H^2. \tag{5.15}$$

The assertion (5.13) follows from (5.14) and (5.15) immediately. □

Let $d := \inf\{\|\boldsymbol{w}^0 - \boldsymbol{w}^*\|_H \,|\, \boldsymbol{w}^* \in \Omega^*\}$. Then, for any given $\epsilon > 0$, Theorem 5.5 shows that the proposed algorithm (3.9) needs at most $\lfloor d^2/c\epsilon \rfloor$ iterations to ensure that $\|\boldsymbol{w}^k - \boldsymbol{w}^{k+1}\|_H^2 \le \epsilon$. Recall (4.26) and $\alpha > 0$ is a constant. It indicates that $\boldsymbol{w}^k$ is a solution point of $\mathrm{VI}(\Omega, F, \theta)$ if $\|\boldsymbol{w}^k - \boldsymbol{w}^{k+1}\|_H^2 = 0$. A worst-case $O(1/t)$ convergence rate in a nonergodic sense is thus established for the proposed algorithm (3.9).

6 Some Special Cases

In this section, we discuss some special cases when a regrouping strategy for (1.1) is specified and demonstrate the new algorithm in some more specific contexts. In particular, we show that the existing algorithms in [15, 17] can both be recovered by regrouping the variables in (1.1) appropriately. Therefore, the convergence rate results established in Sections 5.1 and 5.2 are applicable to the methods in [15, 17]. This is a by-product of this paper.

In such special cases, we always consider the first group as x_1, thus we have

$$\boldsymbol{x}_1 = x_1, \qquad \text{and} \qquad m_1 = 1. \tag{6.1}$$

In addition, we take

$$\tau_1 = m_1 - 1 = 0.$$

Because $\boldsymbol{x}_1 = x_1$ and $\tau_1 = 0$, The first subproblem in the prediction step (4.11a) becomes

$$\tilde{\boldsymbol{x}}_1^k = \arg\min\big\{\mathcal{L}_\beta^t[\boldsymbol{x}_1, \boldsymbol{x}_2^k, \ldots, \boldsymbol{x}_t^k, \lambda^k] \;\big|\; \boldsymbol{x}_1 \in \mathcal{X}_1\big\}.$$

For this case, the prediction step (4.11) can be specified as follows.

Prediction. For given $\boldsymbol{v}^k = (x_2^k, \ldots, x_m^k, \lambda^k) = (\boldsymbol{x}_2^k, \ldots, \boldsymbol{x}_t^k, \lambda^k)$,

$$\left\{\begin{array}{l} \tilde{\boldsymbol{x}}_1^k = \arg\min\big\{\mathcal{L}_\beta^t[\boldsymbol{x}_1, \boldsymbol{x}_2^k, \ldots, \boldsymbol{x}_t^k, \lambda^k] \;\big|\; \boldsymbol{x}_1 \in \mathcal{X}_1\big\}; \\ \text{for } r = 2, \ldots t, \text{ do:} \\ \quad \text{for } j = 1, \ldots m_r, \text{ do:} \\ \qquad \tilde{x}_{r_j}^k = \arg\min\left\{\begin{array}{l} \mathcal{L}_\beta^t\big(\tilde{\boldsymbol{x}}_1^k, \ldots, \tilde{\boldsymbol{x}}_{r-1}^k, x_{r_1}^k, \ldots, x_{r_{j-1}}^k, x_{r_j}, x_{r_{j+1}}^k, \ldots, x_{r_{m_r}}^k, \\ \qquad \boldsymbol{x}_{r+1}^k, \ldots, \boldsymbol{x}_t^k, \lambda^k\big) + \frac{\tau_r\beta}{2}\|A_{r_j}(x_{r_j} - x_{r_j}^k)\|^2 \end{array} \;\middle|\; x_{r_j} \in X_{r_j}\right\}; \\ \quad \text{end.} \\ \text{end.} \end{array}\right. \tag{6.2a}$$

Additionally, we define

$$\tilde{\lambda}^k = \lambda^k - \beta\big(A_1\tilde{\boldsymbol{x}}_1^k + \sum_{r=2}^t \mathcal{A}_r\boldsymbol{x}_r^k - b\big). \tag{6.2b}$$

According to (6.2), because we choose $\boldsymbol{x}_1 = x_1$ and $\tau_1 = 0$, then $\boldsymbol{x}_1 = x_1$ is an intermediate variable and it is not needed in the iteration. In other words, to implement the proposed algorithm (3.9) with $\boldsymbol{x}_1 = x_1$, we only need $\boldsymbol{v}^k = (\boldsymbol{x}_2^k, \ldots, \boldsymbol{x}_t^k, \lambda^k)$. Moreover, note that $\beta(D_1 - \mathcal{A}_1^T\mathcal{A}_1)$ is the unique non-zero elements in the first row and first column of the matrix Q (see (4.1)). In this case,

$$D_1 = (\tau_1 + 1)\mathrm{diag}(\mathcal{A}_1^T\mathcal{A}_1) = \mathrm{diag}(\mathcal{A}_1^T\mathcal{A}_1)$$

and $(D_1 - \mathcal{A}_1^T\mathcal{A}_1)$ becomes the zero matrix. Accordingly, Lemma 4.3 is reduced to the following lemma (for convenience, we still use the same letters to denote the matrices).

Lemma 6.1. *Let $\tilde{\boldsymbol{x}}^k$ be generated by* (6.2a) *from the given vector $\boldsymbol{v}^k$ and $\tilde{\lambda}^k$ be defined by* (6.2b). *Then, the predictor $\tilde{\boldsymbol{w}}^k \in \Omega$ satisfies*

$$\tilde{\boldsymbol{w}}^k \in \Omega, \;\; \vartheta(\boldsymbol{x}) - \vartheta(\tilde{\boldsymbol{x}}^k) + (\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T F(\tilde{\boldsymbol{w}}^k) \geq (\boldsymbol{v} - \tilde{\boldsymbol{v}}^k)^T Q(\boldsymbol{v}^k - \tilde{\boldsymbol{v}}^k), \;\; \forall \boldsymbol{w} \in \Omega, \tag{6.3}$$

where Q is defined by

$$Q = \begin{pmatrix} \beta \mathcal{Q}_e & 0 \\ -\mathcal{A} & \frac{1}{\beta} I \end{pmatrix}, \tag{6.4}$$

$\mathcal{A}$ and $\mathcal{Q}_e$ are defined by (4.3) *and* (4.4)*, respectively.*

Note that the matrix Q in (6.4) can be generated by cutting off the first row and column of the matrix Q given in (4.1). This is because the first block-wise variable $\boldsymbol{x}_1$ is just an intermediate variable for the special case under our current consideration. Thus, the matrix Q originally given in (4.1) for the generic case of Algorithm 1 has all zeros in its first row and column for the special case where $\boldsymbol{x}_1 = x_1$; hence we only consider (6.4) for this special case. Likewise, with the specific Q in (6.4), we can also define the corresponding matrix H and G as in (4.8a) and (4.8b), respectively. Moreover, as shown in Lemma 4.2, the positive definiteness of these two matrices is crucial for proving the convergence of Algorithm 1 for the special case where $\boldsymbol{x}_1 = x_1$.

Moreover, the correction step (4.12) in the generic setting can be specified as follows.

Correction. The new iterate $\boldsymbol{v}^{k+1}$ is given by

$$\boldsymbol{v}^{k+1} = \boldsymbol{v}^k - \alpha M(\boldsymbol{v}^k - \tilde{\boldsymbol{v}}^k), \tag{6.5a}$$

where

$$M = \begin{pmatrix} \mathcal{Q}_e^{-T}\mathcal{D}_e & 0 \\ -\beta\mathcal{A} & I \end{pmatrix}, \qquad \alpha \in \begin{cases} (0,1), & \text{if} \quad \tau_r \geq m_r - 1,\ r = 2, \ldots, t; \\ (0,1], & \text{if} \quad \tau_r > m_r - 1,\ r = 2, \ldots, t, \end{cases} \tag{6.5b}$$

and $\tilde{\boldsymbol{v}}^k$ is the related sub-vector of the predictor $\tilde{\boldsymbol{w}}^k$ generated by (6.2).

The matrices $\mathcal{Q}_e$, $\mathcal{D}_e$ in (6.5b) are defined in (4.4) and (4.5), respectively. It follows from (6.5b) that

$$M = \begin{pmatrix} \mathcal{Q}_e^{-T}\mathcal{D}_e & 0 \\ -\beta\mathcal{A} & I \end{pmatrix} \quad \text{and} \quad \mathcal{A} = (A_2, A_3, \ldots, A_m).$$

Also, because of (6.2b), we have

$$\lambda^{k+1} = \lambda^k - \alpha\beta\big(\textstyle\sum_{j=1}^m A_j \tilde{x}_j^k - b\big). \tag{6.6a}$$

In addition, the variables $x_2, \cdots, x_m$ are updated by the back substitution procedure:

$$\mathcal{D}_e^{-1}\mathcal{Q}_e^T \begin{pmatrix} \boldsymbol{x}_2^{k+1} - \boldsymbol{x}_2^k \\ \vdots \\ \boldsymbol{x}_t^{k+1} - \boldsymbol{x}_t^k \end{pmatrix} = \alpha \begin{pmatrix} \tilde{\boldsymbol{x}}_2^k - \boldsymbol{x}_2^k \\ \vdots \\ \tilde{\boldsymbol{x}}_t^k - \boldsymbol{x}_t^k \end{pmatrix}. \tag{6.6b}$$

In the following, we show that both the methods in [15, 17] are special cases of the proposed algorithm (3.9) with $\boldsymbol{x}_1 = x_1$.

6.1 The ADMM-GBS in [15]

Let us consider the special regrouping strategy with $\boldsymbol{x}_i = x_i$ for $i = 1, \cdots, m$ for (1.1). That is, each block of variables only consists of one variable. For this case, we have

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_m \end{pmatrix}, \quad \text{where} \quad \boldsymbol{x}_i = x_i, \quad i = 1, \ldots, m. \tag{6.7}$$

Clearly, for this regrouping strategy, in the implementation of the proposed algorithm (3.9), we have

$$\tau_i = 0, \quad i = 1, 2, \ldots, m,$$

and thus the matrix $\mathcal{Q}_e$ (4.4) and $\mathcal{D}_e$ can be specified as

$$\mathcal{Q}_e = \begin{pmatrix} A_2^T A_2 & 0 & \cdots & 0 \\ A_3^T A_2 & A_3^T A_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ A_m^T A_2 & \cdots & A_m^T A_{m-1} & A_m^T A_m \end{pmatrix} \quad \text{and} \quad \mathcal{D}_e = \begin{pmatrix} A_2^T A_2 & 0 & \cdots & 0 \\ 0 & A_3^T A_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_m^T A_m \end{pmatrix},$$

respectively. According to (6.2), the prediction step (4.11) is reduced to

$$\begin{cases} \tilde{x}_1^k = \arg\min\{\mathcal{L}_\beta^m(x_1, x_2^k, x_3^k, \ldots, x_m^k, \lambda^k) \mid x_1 \in X_1\}; \\ \tilde{x}_2^k = \arg\min\{\mathcal{L}_\beta^m(\tilde{x}_1^k, x_2, x_3^k, \ldots, x_m^k, \lambda^k) \mid x_2 \in X_2\}; \\ \qquad \vdots \\ \tilde{x}_i^k = \arg\min\{\mathcal{L}_\beta^m(\tilde{x}_1^k, \ldots, \tilde{x}_{i-1}^k, x_i, x_{i+1}^k, \ldots, x_m^k, \lambda^k) \mid x_i \in X_i\}; \\ \qquad \vdots \\ \tilde{x}_m^k = \arg\min\{\mathcal{L}_\beta^m(\tilde{x}_1^k, \ldots, \tilde{x}_{m-1}^k, x_m, \lambda^k) \mid x_m \in X_m\}, \end{cases} \tag{6.8a}$$

and

$$\tilde{\lambda}^k = \lambda^k - \beta\big(A_1\tilde{x}_1^k + \sum_{j=2}^m A_j x_j^k - b\big). \tag{6.8b}$$

The new iterate $\boldsymbol{v}^{k+1}$ is given by (6.5). Since $\tau_i = m_i - 1 = 0$, the step size $\alpha \in (0, 1)$.

If we denote the output (6.8a) by $\bar{x}_1^{k+1}, \bar{x}_2^{k+1}, \ldots, \bar{x}_m^{k+1}$, namely,

$$\begin{cases} \bar{x}_1^{k+1} = \arg\min\{\mathcal{L}_\beta^m(x_1, x_2^k, x_3^k, \ldots, x_m^k, \lambda^k) \mid x_1 \in X_1\}; \\ \bar{x}_2^{k+1} = \arg\min\{\mathcal{L}_\beta^m(\bar{x}_1^{k+1}, x_2, x_3^k, \ldots, x_m^k, \lambda^k) \mid x_2 \in X_2\}; \\ \qquad \vdots \\ \bar{x}_i^{k+1} = \arg\min\{\mathcal{L}_\beta^m(\bar{x}_1^{k+1}, \ldots, \bar{x}_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_m^k, \lambda^k) \mid x_i \in X_i\}; \\ \qquad \vdots \\ \bar{x}_m^{k+1} = \arg\min\{\mathcal{L}_\beta^m(\bar{x}_1^{k+1}, \ldots, \bar{x}_{m-1}^{k+1}, x_m, \lambda^k) \mid x_m \in X_m\}, \end{cases} \tag{6.9a}$$

and set

$$\bar{\lambda}^{k+1} = \lambda^k - \beta\big(\sum_{j=1}^m A_j \bar{x}_j^{k+1} - b\big). \tag{6.9b}$$

The implementation of (6.6) becomes

$$\begin{cases} \mathcal{D}_e^{-1}\mathcal{Q}_e^T \begin{pmatrix} x_2^{k+1} - x_2^k \\ \vdots \\ x_m^{k+1} - x_m^k \end{pmatrix} = \alpha \begin{pmatrix} \bar{x}_2^{k+1} - x_2^k \\ \vdots \\ \bar{x}_m^{k+1} - x_m^k \end{pmatrix}, \\ \lambda^{k+1} - \lambda^k = \alpha(\bar{\lambda}^{k+1} - \lambda^k). \end{cases} \tag{6.10}$$

Note that for this special case, we have

$$\mathcal{D}_e^{-1}\mathcal{Q}_e^T = \begin{pmatrix} I_{n_2} & (A_2^TA_2)^{-1}A_2^TA_3 & \cdots & (A_2^TA_2)^{-1}A_2^TA_m \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & (A_{m-1}^TA_{m-1})^{-1}A_{m-1}^TA_m \\ 0 & \cdots & 0 & I_{n_m} \end{pmatrix}.$$

It is just the left-upper part of the matrix P (see (3.2)). Thus, the method (6.9)-(6.10) reduces to the ADMM-GBS in [15].

6.2 The Splitting Method in [17]

Then, we consider another regrouping for (1.1):

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix}, \quad \text{where} \quad \boldsymbol{x}_1 = x_1 \quad \text{and} \quad \boldsymbol{x}_2 = \begin{pmatrix} x_2 \\ \vdots \\ x_m \end{pmatrix}. \tag{6.11}$$

For this regrouping, we have

$$m_1 = 1 \qquad \text{and} \qquad m_2 = m-1.$$

Besides (6.1), for the implementation of the new algorithm (3.9), we have

$$\tau_2 = \tau > m-2 = m_2 - 1$$

and thus the matrix $\mathcal{Q}_e$ given in (4.4) is specified as

$$\mathcal{Q}_e = \mathcal{D}_e = \begin{pmatrix} (\tau+1)A_2^TA_2 & 0 & \cdots & 0 \\ 0 & (\tau+1)A_3^TA_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & (\tau+1)A_m^TA_m \end{pmatrix}. \tag{6.12}$$

Thus, according to (6.2), the prediction step (4.11) is reduced to

$$\begin{cases} \tilde{x}_1^k = \arg\min\{\mathcal{L}_\beta^2(x_1, x_2^k, x_3^k, \ldots, x_m^k, \lambda^k) \mid x_1 \in X_1\}; \\ \tilde{x}_i^k = \arg\min\left\{ \begin{array}{c} \mathcal{L}_\beta^2(\tilde{x}_1^k, x_2^k \ldots, x_{i-1}^k, x_i, x_{i+1}^k, \ldots, x_m^k, \lambda^k) \\ +\frac{\tau\beta}{2}\|A_i(x_i - x_i^k)\|^2 \end{array} \;\middle|\; x_i \in X_i \right\}, \quad i = 2, \ldots, m. \end{cases} \tag{6.13a}$$

and

$$\tilde{\lambda}^k = \lambda^k - \beta\big(A_1\tilde{x}_1^k + \sum_{j=2}^m A_j x_j^k - b\big). \tag{6.13b}$$

Since $\tau_2 = \tau > m_2 - 1 = m-2$, we take the step size $\alpha = 1$ in the correction step (6.5). The new iterate is given by

$$\boldsymbol{v}^{k+1} = \boldsymbol{v}^k - M(\boldsymbol{v}^k - \tilde{\boldsymbol{v}}^k).$$

Because $\boldsymbol{x}_2 = (x_2, \ldots, x_m)$, we have $\mathcal{Q}_e = \mathcal{D}_e$ (see (4.4) and (4.5)). Thus the matrix M in (6.5b) becomes

$$M = \begin{pmatrix} I & 0 \\ -\beta\mathcal{A} & I \end{pmatrix}.$$

Using $\mathcal{Q}_e = \mathcal{D}_e$ and $\alpha = 1$, the implementation of correction (6.6) is reduced to

$$\lambda^{k+1} = \lambda^k - \beta\big(\textstyle\sum_{j=1}^m A_j\tilde{x}_j^k - b\big). \tag{6.14a}$$

and

$$\begin{pmatrix} x_2^{k+1} \\ \vdots \\ x_m^{k+1} \end{pmatrix} = \begin{pmatrix} \tilde{x}_2^k \\ \vdots \\ \tilde{x}_m^k \end{pmatrix}. \tag{6.14b}$$

Therefore, the prediction-correction method consists of (6.13) and (6.14) can be directly represented by

$$\begin{cases} x_1^{k+1} = \arg\min\big\{\mathcal{L}_\beta^2(x_1, x_2^k, x_3^k, \ldots, x_m^k, \lambda^k) \;\big|\; x_1 \in X_1\big\}; \\ x_i^{k+1} = \arg\min\big\{\mathcal{L}_\beta^2(x_1^{k+1}, x_2^k \ldots, x_{i-1}^k, x_i, x_{i+1}^k, \ldots, x_m^k, \lambda^k) + \frac{\tau\beta}{2}\|A_i(x_i - x_i^k)\|^2\big|\; x_i \in X_i\big\}, \\ \qquad\qquad i = 2, \ldots, m. \\ \lambda^{k+1} = \lambda^k - \beta\big(\sum_{j=1}^m A_j x_j^{k+1} - b\big). \end{cases} \tag{6.15}$$

To clearly see the relationship between (6.15) and the method in [17], let us summarize a conclusion in the following lemma.

Lemma 6.2. *Let the augmented Lagrange function $\mathcal{L}_\beta^m(x_1, \ldots, x_m, \lambda)$ be defined in* (1.5)*. Then we have*

$$\begin{aligned} &\arg\min\Big\{\mathcal{L}_\beta^m(x_1^{k+1}, x_2^k \ldots, x_{i-1}^k, x_i, x_{i+1}^k, \ldots, x_m^k, \lambda^k) + \frac{\tau\beta}{2}\|A_i(x_i - x_i^k)\|^2\big|\; x_i \in X_i\Big\} \\ = \;& \arg\min\Big\{\theta_i(x_i) - (\lambda^{k+\frac{1}{2}})^T A_i x_i + \frac{(\tau+1)\beta}{2}\|A_i(x_i - x_i^k)\|^2\big|\; x_i \in X_i\Big\} \end{aligned} \tag{6.16}$$

where

$$\lambda^{k+\frac{1}{2}} = \lambda^k - \beta\big(A_1 x_1^{k+1} + \sum_{i=2}^m A_i x_i^k - b\big). \tag{6.17}$$

Proof. Let us observe the x_i-subproblems in the left-hand side of (6.16). Notice that

$$\begin{aligned} &\mathcal{L}_\beta^m(x_1^{k+1}, x_2^k \ldots, x_{i-1}^k, x_i, x_{i+1}^k, \ldots, x_m^k, \lambda^k) \\ = \;& \theta_i(x_i) - \theta_i(x_i^k) + \textstyle\sum_{j=1}^m \theta_j(x_j^k) - (\lambda^k)^T[A_1 x_1^{k+1} + A_i x_i + \sum_{j=2, j\neq i}^m A_j x_j^k - b] \\ & + \tfrac{\beta}{2}\|A_i(x_i - x_i^k) + A_1 x_1^{k+1} + \textstyle\sum_{j=2}^m A_j x_j^k - b\|^2. \end{aligned}$$

Ignoring some constant terms in the objective function of the minimization problem, we have

$$\begin{aligned} &\arg\min\big\{\mathcal{L}_\beta^m(x_1^{k+1}, x_2^k \ldots, x_{i-1}^k, x_i, x_{i+1}^k, \ldots, x_m^k, \lambda^k) + \tfrac{\tau\beta}{2}\|A_i(x_i - x_i^k)\|^2 | x_i \in X_i\big\} \\ = \;& \arg\min\left\{\begin{array}{c} \theta_i(x_i) - (\lambda^k)^T A_i x_i + \frac{\beta}{2}\|A_i(x_i - x_i^k) + A_1 x_1^{k+1} + \sum_{j=2}^m A_j x_j^k - b\|^2 \\ + \frac{\tau\beta}{2}\|A_i(x_i - x_i^k)\|^2 \end{array}\;\Big|\; x_i \in X_i\right\}. \end{aligned}$$

Thus, the optimality condition of the x_i-subproblem is

$$\begin{aligned} x_i^{k+1} \in X_i, \quad & \theta_i(x_i) - \theta_i(x_i^{k+1}) + (x_i - x_i^{k+1})^T\big\{-A_i^T\lambda^k + \\ & + \beta A_i^T\big[A_i(x_i^{k+1} - x_i^k) + (A_1 x_1^{k+1} + \textstyle\sum_{j=2}^m A_j x_j^k - b)\big] + \tau\beta A_i^T A_i(x_i^{k+1} - x_i^k)\big\} \geq 0, \quad \forall x_i \in X_i \end{aligned}$$

It follows from (6.17) that

$$\lambda^k = \lambda^{k+\frac{1}{2}} + \beta\big(A_1x_1^{k+1} + \sum_{j=2}^m A_jx_j^k - b\big).$$

Substituting this identity into the last inequality, we obtain

$$x_i^{k+1} \in X_i,\ \ \theta_i(x_i) - \theta_i(x_i^{k+1}) + (x_i - x_i^{k+1})^T\big\{-A_i^T\lambda^{k+\frac{1}{2}} + (1+\tau)\beta A_i^TA_i(x_i^{k+1} - x_i^k)\big\} \geq 0,\ \ \forall\, x_i \in X_i.$$

This is just the optimality condition of the x_i-subproblem of the right-hand side of (6.16). □

Thus, by setting $\mu = \tau + 1$, the scheme (6.15) can be represented as the following scheme:

$$\begin{cases} x_1^{k+1} = \arg\min\big\{\mathcal{L}_\beta^m(x_1, x_2^k, x_3^k, \ldots, x_m^k, \lambda^k) \ \big|\ x_1 \in X_1\big\}; \\ \lambda^{k+\frac{1}{2}} = \lambda^k - \beta\big(A_1x_1^{k+1} + \sum_{i=2}^m A_ix_i^k - b\big); \\ x_i^{k+1} = \arg\min\Big\{\theta_i(x_i) - (\lambda^{k+\frac{1}{2}})^TA_ix_i + \frac{\mu\beta}{2}\|A_i(x_i - x_i^k)\|^2\big|\ x_i \in X_i\Big\},\ i = 2, \ldots, m. \\ \lambda^{k+1} = \lambda^k - \beta\big(\sum_{j=1}^m A_jx_j^{k+1} - b\big). \end{cases} \tag{6.18}$$

This is just the method proposed in [17]. Recall that $\mu > m-1$ (since $\tau > m-2$) is the condition to ensure the convergence of the method in [17].

7 A Refined Version of Algorithm 1 with Calculated Step Size

Instead of taking the constant step size α in the correction step (4.12), we can refine the algorithm (3.9) by choosing a calculated step size α_k at each iteration. Recall the role of the correction step in the algorithm (3.9) is to ensure the strict contraction property of the sequence (see (4.29)). The main idea of refining the algorithm (3.9) is that we can find a better step size, which is iteration-dependent, for each iteration such that the proximity to the solution set can be further reduced. For the case where calculating the step size is not computationally expensive, this refined version can accelerate the convergence and the number of iteration can be reduced while the computation per iteration is just slightly increased. However, if the step size itself is computationally expensive, we still recommend the scheme (3.9) with a constant step size because for this case, the computation per iteration might be significantly increased thus the overall convergence might be slower even though the number of iteration might be smaller.

To see how to find a better step size to further reduce the proximity to the solution set, let us revisit Lemma 4.3. Indeed, setting $\boldsymbol{w} = \boldsymbol{w}^*$ in (4.14), we get

$$(\tilde{\boldsymbol{w}}^k - \boldsymbol{w}^*)^TQ(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) \geq \vartheta(\tilde{\boldsymbol{x}}^k) - \vartheta(\boldsymbol{x}^*) - (\tilde{\boldsymbol{w}}^k - \boldsymbol{w}^*)^TF(\tilde{\boldsymbol{w}}^k),\ \ \forall \boldsymbol{w}^* \in \Omega^*.$$

Using the monotonicity of F and (2.4), it follows that

$$(\tilde{\boldsymbol{w}}^k - \boldsymbol{w}^*)^TQ(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) \geq 0. \tag{7.1}$$

and consequently

$$(\boldsymbol{w}^k - \boldsymbol{w}^*)^TQ(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) \geq (\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)^TQ(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k), \quad \forall\, \boldsymbol{w}^* \in \Omega^*. \tag{7.2}$$

Because $Q = HM$, it follows that

$$\langle H(\boldsymbol{w}^k - \boldsymbol{w}^*), M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)\rangle \geq \frac{1}{2}\|\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k\|^2_{(Q^T+Q)}, \quad \forall \boldsymbol{w}^* \in \Omega^*.$$

This means that $M(\tilde{\boldsymbol{w}}^k - \boldsymbol{w}^k)$ is a descent direction of the distance function $\|\boldsymbol{w} - \boldsymbol{w}^*\|_H^2$ at the point $\boldsymbol{w}^k$, even if $\boldsymbol{w}^*$ is unknown. Along the direction $M(\tilde{\boldsymbol{w}}^k - \boldsymbol{w}^k)$ with well-chosen step size α, we can reduce the unknown distance function $\|\boldsymbol{w} - \boldsymbol{w}^*\|_H^2$. We define the step-size-dependent new iterate by

$$\boldsymbol{w}^{k+1}(\alpha) = \boldsymbol{w}^k - \alpha M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k), \tag{7.3}$$

and

$$p(\alpha) = \|\boldsymbol{w}^k - \boldsymbol{w}^*\|_H^2 - \|\boldsymbol{w}^{k+1}(\alpha) - \boldsymbol{w}^*\|_H^2. \tag{7.4}$$

By using $HM = Q$, we have

$$\begin{aligned} p(\alpha) &= \|\boldsymbol{w}^k - \boldsymbol{w}^*\|_H^2 - \|\boldsymbol{w}^{k+1}(\alpha) - \boldsymbol{w}^*\|_H^2 \\ &= \|\boldsymbol{w}^k - \boldsymbol{w}^*\|_H^2 - \|(\boldsymbol{w}^k - \boldsymbol{w}^*) - \alpha M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)\|_H^2 \\ &= 2\alpha(\boldsymbol{w}^k - \boldsymbol{w}^*)^T Q(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) - \alpha^2 \|M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)\|_H^2. \end{aligned}$$

Ideally we want to maximize the quadratic function $p(\alpha)$. However, it is impossible due to the lack of the unknown solution point $\boldsymbol{w}^*$. By using (7.2), we obtain

$$p(\alpha) \geq q(\alpha), \tag{7.5}$$

where

$$q(\alpha) = 2\alpha(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)^T Q(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) - \alpha^2 \|M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)\|_H^2. \tag{7.6}$$

We thus turn to the second best choice: Maximizing the quadratic function $q(\alpha)$ which is a lower bound of $p(\alpha)$. This promotes us to take the value of α as

$$\alpha_k^* = \frac{(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)^T Q(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)}{\|M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)\|_H^2} = \frac{(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)^T Q(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)}{(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)^T (M^T H M)(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)}. \tag{7.7}$$

We take $\alpha = \gamma \alpha_k^*$. According to (4.9), we have

$$Q^T + Q - M^T H M \succeq 0$$

and thus

$$\alpha_k^* \geq \frac{1}{2}. \tag{7.8}$$

Therefore, the iteration-dependent step size calculated by (7.7) is bounded away from 0.

Moreover, it worths to mention that it follows from (4.10) that

$$M^T H M = \begin{pmatrix} \beta(D_1 - \mathcal{A}_1^T \mathcal{A}_1) & 0 & 0 \\ 0 & \beta(\mathcal{D}_e + \mathcal{A}^T \mathcal{A}) & -\mathcal{A}^T \\ 0 & -\mathcal{A} & \frac{1}{\beta} I \end{pmatrix}.$$

Therefore, the denominator in (7.7) can be calculated directly based on the matrix defined above before implementing the Gaussian back substitution procedure and there is no need to calculate the inverse of any matrix for determining α_k.

So, the proposed algorithm (3.9) can be altered to a refined version where the constant step size α in (4.12b) is iteratively calculated by (7.7). The resulting refined version differs from the proposed algorithm (3.9) only in its correction step as shown below.

Correction step: The new iterate $\boldsymbol{w}^{k+1}$ is given by

$$\boldsymbol{w}^{k+1} = \boldsymbol{w}^k - \alpha_k M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k), \tag{7.9a}$$

where $\tilde{\boldsymbol{w}}^k$ is generated by the prediction step (4.11) and M is given by (4.6). The step size α_k is given by

$$\alpha_k = \gamma\alpha_k^*, \qquad \gamma \in (0,2) \qquad \text{and} \qquad \alpha_k^* = \frac{(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)^T Q(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)}{(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)^T (M^T H M)(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)}. \tag{7.9b}$$

Note that it follows from (7.6) and (7.7) that

$$\begin{aligned} q(\gamma\alpha_k^*) &= 2\gamma\alpha_k^*(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)^T Q(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k) - (\gamma\alpha_k^*)^2 \|M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)\|_H^2 \\ &= \gamma(2-\gamma)(\alpha_k^*)^2 \|M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)\|_H^2. \end{aligned} \tag{7.10}$$

The following theorem shows the strict contraction property of the sequence generated by the refined algorithm with the iteratively calculated step size (7.7). Its proof is similar as Theorem 4.5 and thus omitted.

Theorem 7.1. *Let $\{\boldsymbol{w}^k\}$ be the sequence generated by the refined algorithm of (3.9) with the iteratively calculated step size* (7.7)*. Then, it holds*

$$\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^*\|_H^2 \le \|\boldsymbol{w}^k - \boldsymbol{w}^*\|_H^2 - \frac{\gamma(2-\gamma)}{4}\|M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k)\|_H^2, \quad \forall\, \boldsymbol{w}^* \in \Omega^*. \tag{7.11}$$

Based on Theorem 7.1, the convergence and the convergence rates can all be established similar as the analysis in Sections 4 and 5. We omit them for succinctness.

8 A Linearized Splitting Block-wise ADMM with Gaussian Back Substitution

As analyzed, the x_{r_j}-subproblems (see (3.7)) in the proposed splitting version of block-wise ADMM-GBS (3.9) are in form of (1.3) and we can further alleviate them by linearizing their quadratic terms if these subproblems are still too hard for a particular application of the model (1.1). More specifically, recall the x_{r_j}-subproblem (3.7) in (3.9) and ignore some constant terms in its objective. Then, if we linearize its quadratic term, the resulting linearized subproblem becomes

$$\bar{x}_{r_j}^{k+1} = \arg\min\left\{ \begin{array}{c} \theta_{r_j}(x_{r_j}) - (\lambda^k)^T A_{r_j} x_{r_j} + (x_{r_j} - x_{r_j}^k)^T \\ \beta A_{r_j}^T \big(\sum_{s=1}^{r-1} \mathcal{A}_s \bar{\boldsymbol{x}}_s^{k+1} + \sum_{s=r}^{t} \mathcal{A}_s \boldsymbol{x}_s^k - b\big) + \frac{\nu_r \beta}{2}\|x_{r_j} - x_{r_j}^k\|^2, \end{array} \right\} \tag{8.1}$$

which is indeed in form of (1.2). Note that in (8.1), the constant $\nu_r > 0$ plays the role of controlling the proximity of the linearization, and it should be sufficiently large to ensure the accuracy of this linearized subproblem and finally the convergence. As well studied in the literature such as [19, 21, 32, 33, 34], we require

$$\nu_r > \rho(\mathcal{A}_r^T \mathcal{A}_r), \quad r = 1, \ldots, t, \tag{8.2}$$

where $\rho(\cdot)$ denotes the spectrum radius of a matrix.

Therefore, replacing the x_{r_j}-subproblems in (3.9) by their linearized counterparts given in (8.1), we can obtain a linearized version of the proposed splitting block-wise ADMM-GBS (3.9) whose x_{r_j}-subproblems are in form of (1.2).

8.1 Algorithm

For the algorithm in this section, we define the matrix

$$\mathcal{P}_L = \begin{pmatrix} I & 0 & 0 & 0 & 0 & 0 \\ 0 & I & \frac{1}{\nu_2}\mathcal{A}_2^T\mathcal{A}_3 & \cdots & \frac{1}{\nu_2}\mathcal{A}_2^T\mathcal{A}_t & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ & \vdots & \ddots & I & \frac{1}{\nu_{t-1}}\mathcal{A}_{t-1}^T\mathcal{A}_t & 0 \\ 0 & 0 & \cdots & 0 & I & 0 \\ 0 & 0 & \cdots & 0 & 0 & I_\ell \end{pmatrix}. \tag{8.3}$$

and summarize the linearized version of the scheme (3.9) as follows.

Algorithm 2: A linearized version of the splitting block-wise ADMM-GBS (3.9) for (1.1)

Initialization: Specify a regrouping for the model (1.1) with determined values of t and m_r for $r = 1, 2, \cdots, t$. Choose constants ν_r such that $\nu_r > \rho(\mathcal{A}_r^T\mathcal{A}_r)$ for $r = 1, \ldots, t$. Let $\mathcal{P}_L$ be defined in (8.3). With the given iterate $\boldsymbol{w}^k = (\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \cdots, \boldsymbol{x}_t^k, \lambda^k) \in \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_t \times \mathbb{R}^\ell$, the new iterate is generated by the following steps.

$$\begin{cases} \begin{cases} \text{for } r = 1, 2, \ldots t, \text{ do:} \\ \quad \text{for } j = 1, 2, \ldots, m_r, \text{ parallel do:} \\ \quad\quad \bar{x}_{r_j}^{k+1} = \arg\min\left\{ \begin{array}{c} \theta_{r_j}(x_{r_j}) - (\lambda^k)^T A_{r_j} x_{r_j} + (x_{r_j} - x_{r_j}^k)^T \\ \beta A_{r_j}^T\big(\sum_{s=1}^{r-1}\mathcal{A}_s\bar{\boldsymbol{x}}_s^{k+1} + \sum_{s=r}^{t}\mathcal{A}_s\boldsymbol{x}_s^k - b\big) + \frac{\nu_r\beta}{2}\|x_{r_j} - x_{r_j}^k\|^2 \end{array} \middle| x_{r_j} \in X_{r_j} \right\}; \\ \quad \text{end.} \\ \text{end.} \\ \bar{\lambda}^{k+1} = \lambda^k - \beta\big(\sum_{r=1}^{t}\mathcal{A}_r\bar{\boldsymbol{x}}_r^{k+1} - b\big). \end{cases} \\ \mathcal{P}_L(\boldsymbol{w}^{k+1} - \boldsymbol{w}^k) = (\bar{\boldsymbol{w}}^{k+1} - \boldsymbol{w}^k). \end{cases} \tag{8.4}$$

Remark 8.1. Just like the scheme (3.9), with the block-wise upper triangular matrix $\mathcal{P}_L$ defined in (3.2), the entries of $\boldsymbol{v}^{k+1}$ can be updated in the order of $\lambda \to x_m \to \cdots x_2$ by the Gaussian back substitution procedure when implementing (8.4). Moreover, the matrix $\mathcal{P}_L$ does not require computing any inverse of matrix, not like the matrix $\mathcal{P}$ defined in (3.8). Therefore, it is an easier substitution procedure compared with the one in (3.9). Meanwhile, theoretically it is required to estimate $\rho(\mathcal{A}_r^T\mathcal{A}_r)$ for $r = 1, 2, \cdots, t$, which might not be easy. This is the cost of alleviating the difficulty levels of subproblems from (1.3) to (1.2) for (8.4). Finally, it worths to mention that the requirements $\nu_r > \rho(\mathcal{A}_r^T\mathcal{A}_r)$ for $r = 1, 2, \cdots, t$ are sufficient conditions to ensure the convergence of the linearized version (8.4) and they represent conservative estimates on the parameters ν_r's. In implementation, usually we can choose smaller values for ν_r's which might not satisfy these sufficient conditions while can lead to better numerical performance.

Remark 8.2. In the scheme (8.4), we take the step size as 1 constantly in the Gaussian back substitution procedure. As Section 7, we can analogously discuss how to choose an iteratively calculated step size for the Gaussian back substitution step in (8.4). For succinctness, we omit it and refer to [21] for some useful analysis.

8.2 Convergence Analysis

In this subsection, we prove two important results for the proposed linearized version (8.4). Based on them, the convergence analysis including both the global convergence and the worst-case convergence rates can be established analogously as the analysis in Sections 4 and 5. As in Section 4, we need to rewrite the scheme (8.4) as a prediction-correction form for analysis. For this purpose, similarly as Section 4.1, we first write the matrix Q as the block-wise form

$$Q = \begin{pmatrix} \beta(\nu_1 I - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & \beta\mathcal{Q}_e & 0 \\ 0 & -\mathcal{A} & \frac{1}{\beta}I \end{pmatrix}, \tag{8.5}$$

with $\mathcal{A}$ defined in (4.3) and

$$\mathcal{Q}_e = \begin{pmatrix} \nu_2 I & 0 & \cdots & 0 \\ \mathcal{A}_3^T\mathcal{A}_2 & \nu_3 I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \mathcal{A}_t^T\mathcal{A}_2 & \cdots & \mathcal{A}_t^T\mathcal{A}_{t-1} & \nu_t I \end{pmatrix}. \tag{8.6}$$

Moreover, we use $\mathcal{D}_e = \mathrm{diag}(\nu_2 I, \nu_3 I, \ldots, \nu_t I)$ to denote the diagonal part of $\mathcal{Q}_e$. Using (8.2), we have

$$\mathcal{Q}_e^T + \mathcal{Q}_e \succ \mathcal{D}_e + \mathcal{A}^T\mathcal{A}. \tag{8.7}$$

With these matrices, we can rewrite the scheme (8.4) as follows.

Prediction. For the given $\boldsymbol{w}^k = (x_1^k, x_2^k \ldots, x_m^k, \lambda^k) = (\boldsymbol{x}_1^k, \ldots, \boldsymbol{x}_t^k, \lambda^k)$, generate the predictor $\tilde{\boldsymbol{w}}^k = (\tilde{x}_1^k, \tilde{x}_2^k \ldots, \tilde{x}_m^k, \tilde{\lambda}^k) = (\tilde{\boldsymbol{x}}_1^k, \ldots, \tilde{\boldsymbol{x}}_t^k, \tilde{\lambda}^k)$ by the following steps:

$$\left\{ \begin{array}{l} \text{for } r = 1, 2, \ldots t, \text{ do:} \\ \quad \text{for } j = 1, \ldots m_r, \text{ parallel do:} \\ \quad\quad \tilde{x}_{r_j}^k = \arg\min \left\{ \begin{array}{c} \theta_{r_j}(x_{r_j}) - (\lambda^k)^T A_{r_j} x_{r_j} + (x_{r_j} - x_{r_j}^k)^T \\ \beta A_{r_j}^T \big(\sum_{s=1}^{r-1} \mathcal{A}_s \tilde{\boldsymbol{x}}_s^k + \sum_{s=r}^{t} \mathcal{A}_s \boldsymbol{x}_s^k - b \big) + \frac{\nu_r \beta}{2} \| x_{r_j} - x_{r_j}^k \|^2 \end{array} \middle| \, x_{r_j} \in X_{r_j} \right\}; \\ \quad \text{end.} \\ \text{end.} \end{array} \right. \tag{8.8a}$$

Additionally, we define

$$\tilde{\lambda}^k = \lambda^k - \beta\big(\mathcal{A}_1 \tilde{\boldsymbol{x}}_1^k + \textstyle\sum_{j=2}^{t} \mathcal{A}_j \boldsymbol{x}_j^k - b\big). \tag{8.8b}$$

Correction. The new iterate $\boldsymbol{w}^{k+1}$ is given by

$$\boldsymbol{w}^{k+1} = \boldsymbol{w}^k - M(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k), \tag{8.9a}$$

where $\tilde{\boldsymbol{w}}^k$ is the predictor generated by (8.8) and

$$M = \begin{pmatrix} I & 0 & 0 \\ 0 & \mathcal{Q}_e^{-T}\mathcal{D}_e & 0 \\ 0 & -\beta\mathcal{A} & I \end{pmatrix}. \tag{8.9b}$$

Note that the matrix M in (8.9b) is the same form as the matric defined in (4.6). In the following, we prove a result similar as Lemma 4.3. This assertion enables us to discern the difference between the predictor $\tilde{\boldsymbol{w}}^k$ and a solution point $\boldsymbol{w}^*$.

Lemma 8.3. *Let $\tilde{\boldsymbol{x}}^k$ be generated by* (8.8a) *from the given vector $\boldsymbol{w}^k$ and $\tilde{\lambda}^k$ be defined by* (8.8b). *Then, the predictor $\tilde{\boldsymbol{w}}^k \in \Omega$ satisfies*

$$\tilde{\boldsymbol{w}}^k \in \Omega, \;\; \vartheta(\boldsymbol{x}) - \vartheta(\tilde{\boldsymbol{x}}^k) + (\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T F(\tilde{\boldsymbol{w}}^k) \geq (\boldsymbol{w} - \tilde{\boldsymbol{w}}^k)^T Q(\boldsymbol{w}^k - \tilde{\boldsymbol{w}}^k), \;\; \forall \boldsymbol{w} \in \Omega, \tag{8.10}$$

where

$$Q = \begin{pmatrix} \beta(\nu_1 I - \mathcal{A}_1^T \mathcal{A}_1) & 0 & \cdots & \cdots & 0 & 0 \\ 0 & \beta\nu_2 I & \ddots & & \vdots & \vdots \\ 0 & \beta\mathcal{A}_3^T \mathcal{A}_2 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \beta\mathcal{A}_t^T \mathcal{A}_2 & \cdots & \beta\mathcal{A}_t^T \mathcal{A}_{t-1} & \beta\nu_t I & 0 \\ 0 & -\mathcal{A}_2 & \cdots & -\mathcal{A}_{t-1} & -\mathcal{A}_t & \frac{1}{\beta} I \end{pmatrix}. \tag{8.11}$$

Proof. The optimality condition of the convex minimization problem (8.8a) is

$$\begin{aligned} \tilde{x}_{r_j}^k \in X_{r_j}, \;\; \theta_{r_j}(x_{r_j}) - \theta_{r_j}(\tilde{x}_{r_j}^k) + (x_{r_j} - \tilde{x}_{r_j}^k)^T \big\{ & -A_{r_j}^T \lambda^k \\ & + \beta A_{r_j}^T \big[\textstyle\sum_{s=1}^{r-1} \mathcal{A}_s \tilde{\boldsymbol{x}}_s^k + \sum_{s=r}^{t} \mathcal{A}_s \boldsymbol{x}_s^k - b \big] + \nu_r \beta (\tilde{x}_{r_j}^k - x_{r_j}^k) \big\} \geq 0, \;\; \forall x_{r_j} \in X_{r_j}. \end{aligned}$$

Using the definition of $\tilde{\lambda}^k$ (see (8.8b)), we have

$$\lambda^k = \tilde{\lambda}^k + \beta\big(\mathcal{A}_1 \tilde{\boldsymbol{x}}_1^k + \textstyle\sum_{s=2}^{t} \mathcal{A}_s \boldsymbol{x}_s^k - b\big).$$

Substituting it into the last inequality, we obtain

$$\begin{aligned} \tilde{x}_{r_j}^k \in X_{r_j}, \quad \theta_{r_j}(x_{r_j}) - \theta_{r_j}(\tilde{x}_{r_j}^k) + (x_{r_j} - \tilde{x}_{r_j}^k)^T \big\{ & -A_{r_j}^T \tilde{\lambda}^k \\ & + \beta A_{r_j}^T \big[\textstyle\sum_{s=2}^{r-1} \mathcal{A}_s (\tilde{\boldsymbol{x}}_s^k - \boldsymbol{x}_s^k) \big] + \nu_r \beta (\tilde{x}_{r_j}^k - x_{r_j}^k) \big\} \geq 0, \;\; \forall x_{r_j} \in X_{r_j}. \end{aligned}$$

Applying this inequality for the cases of $j = 1, \ldots, m_r$, and summarizing the resulting inequalities, we get

$$\begin{aligned} \tilde{\boldsymbol{x}}_r^k \in \mathcal{X}_r, \quad \vartheta_r(\boldsymbol{x}_r) - \vartheta_r(\tilde{\boldsymbol{x}}_r^k) + (\boldsymbol{x}_r - \tilde{\boldsymbol{x}}_r^k)^T \big\{ & -\mathcal{A}_r^T \tilde{\lambda}^k \\ & + \beta \mathcal{A}_r^T \big[\textstyle\sum_{s=2}^{r-1} \mathcal{A}_s (\tilde{\boldsymbol{x}}_s^k - \boldsymbol{x}_s^k) \big] + \nu_r \beta (\tilde{\boldsymbol{x}}_r^k - \boldsymbol{x}_r^k) \big\} \geq 0, \; \forall \boldsymbol{x}_r \in \mathcal{X}_r. \end{aligned} \tag{8.12}$$

Note that, for $r = 1$, (8.12) means that

$$\tilde{\boldsymbol{x}}_1^k \in \mathcal{X}_1, \;\; \vartheta_1(\boldsymbol{x}_1) - \vartheta_1(\tilde{\boldsymbol{x}}_1^k) + (\boldsymbol{x}_1 - \tilde{\boldsymbol{x}}_1^k)^T \big\{ -\mathcal{A}_1^T \tilde{\lambda}^k + \beta(\nu_1 I - \mathcal{A}_1^T \mathcal{A}_1)(\tilde{\boldsymbol{x}}_1^k - \boldsymbol{x}_1^k) \big\} \geq 0, \;\; \forall \boldsymbol{x}_1 \in \mathcal{X}_1. \tag{8.13}$$

In addition, by using (8.8b), we have

$$(\sum_{r=1}^{t} \mathcal{A}_r \tilde{\boldsymbol{x}}_r^k - b) - \sum_{s=2}^{t} \mathcal{A}_s (\tilde{\boldsymbol{x}}_s^k - \boldsymbol{x}_s^k) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) = 0,$$

and it can be rewritten as

$$\tilde{\lambda}^k \in \mathbb{R}^\ell, \quad (\lambda - \tilde{\lambda}^k)^T \big\{ (\sum_{r=1}^{t} \mathcal{A}_r \tilde{\boldsymbol{x}}_r^k - b) - \sum_{s=2}^{t} \mathcal{A}_s (\tilde{\boldsymbol{x}}_s^k - \boldsymbol{x}_s^k) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) \big\} \geq 0, \quad \forall\, \lambda \in \mathbb{R}^\ell. \tag{8.14}$$

Combining (8.13), (8.12) ($r = 2, \ldots, t$) and (8.14) together and using the notations $F(\boldsymbol{w})$, Q (see (2.2) and (8.11)), the assertion of this lemma is followed directly. □

Then, in the following lemma, we prove some assertions with respect to the matrices defined before.

Lemma 8.4. *For the matrices Q and M defined in* (8.11) *and* (8.9b)*, respectively, let*

$$H := QM^{-1} \tag{8.15a}$$

and

$$G := Q^T + Q - M^T H M. \tag{8.15b}$$

Then, both the matrices H and G are symmetric and positive definite.

Proof. First, we check the positive definiteness of the matrix H. For the matrix M defined in (8.9b), we have

$$M^{-1} = \begin{pmatrix} I & 0 & 0 \\ 0 & \mathcal{D}_e^{-1}\mathcal{Q}_e^T & 0 \\ 0 & \beta\mathcal{A}\mathcal{D}_e^{-1}\mathcal{Q}_e^T & I \end{pmatrix}.$$

Thus, according to the definition of the matrix H (see (8.15a)), we conclude that

$$H = QM^{-1} = \begin{pmatrix} \beta(\nu_1 I - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & \beta\mathcal{Q}_e\mathcal{D}_e^{-1}\mathcal{Q}_e^T & 0 \\ 0 & 0 & \frac{1}{\beta}I \end{pmatrix}$$

is symmetric and positive definite.

Now, we turn to check the positive definiteness of the matrix G. Note that

$$Q^T + Q = \begin{pmatrix} 2\beta(\nu_1 I - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & \beta(\mathcal{Q}_e^T + \mathcal{Q}_e) & -\mathcal{A}^T \\ 0 & -\mathcal{A} & \frac{2}{\beta}I \end{pmatrix} \tag{8.16}$$

and

$$M^T H M = Q^T M = \begin{pmatrix} \beta(\nu_1 I - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & \beta(\mathcal{D}_e + \mathcal{A}^T\mathcal{A}) & -\mathcal{A}^T \\ 0 & -\mathcal{A} & \frac{1}{\beta}I \end{pmatrix}. \tag{8.17}$$

Then, it follows from (8.16), (8.17) and (8.7)) that

$$\begin{aligned} G &= Q^T + Q - M^T H M \\ &= \begin{pmatrix} \beta(\nu_1 I - \mathcal{A}_1^T\mathcal{A}_1) & 0 & 0 \\ 0 & \beta(\mathcal{Q}_e^T + \mathcal{Q}_e) - \beta(\mathcal{D}_e + \mathcal{A}^T\mathcal{A}) & 0 \\ 0 & 0 & \frac{1}{\beta}I \end{pmatrix} \succ 0. \end{aligned}$$

The assertion of this lemma is proved. □

Based on Lemmas 8.3 and 8.4, and following the analysis in Sections 4.4 and 5, we can easily establish the convergence and worst-case convergence rate for the linearized version (8.4). We omit the detail for succinctness.

9 Conclusions

In this paper, we discuss how to develop an algorithm for the separable multiple-block convex minimization models with linear constraints and an objective function which is in the sum of m functions without coupled variables. We focus on the big-data scenario with a huge m, to which the existing splitting schemes in the literature seem not to be directly applicable. With the assumption that the variables and functions are regrouped as more than two blocks, we investigate how to apply the alternating direction method of multiplier with a Gaussian back substitution (ADMM-GBS) in [15] to the regrouped model which is still in a multiple-block form. The resulting block-wise ADMM-GBS, however, may involve hard subproblems. To yield solvable easier subproblems, we suggest embedding a parallel computation into the block-wise ADMM-GBS; and consequently propose a splitting version of the block-wise ADMM-GBS which is suitable for a distributed-centralized computing system. The global convergence and the worst-case convergence rates measured by the iteration complexity in both the ergodic and nonergodic senses are established for the new algorithm. Moreover, the new algorithm turns to include some existing schemes as special cases; thus a by-product of this paper is that the convergence rates for these existing schemes are also established. We also discuss how to refine the new scheme by choosing an iteratively calculated step size and further alleviating the resulting subproblems. Thus, two advanced versions with refined step sizes and linearized subproblems are proposed, respectively.

The proposed scheme is a basic scheme which can easily inspire specific algorithms when concrete applications of the abstract model under consideration are specified. For example, as mentioned, we can consider further linearizing the subproblems such that each subproblem is of the difficulty level of estimating a function's proximal operator. Also, in addition to the Gaussian back substitution, other correction steps in the literature (e.g., [12, 16, 13]) can be used. In [22], we focused on the case where the model (1.1) is regrouped as two groups and thus a block-wise version of the original ADMM (1.7) is applied. In this paper, we consider the case where the model (1.1) is regrouped as at least three groups and thus the direct extension of ADMM (1.8) is not necessarily convergent. Because of the significant difference between the cases of two and three blocks in ADMM-oriented schemes (see [3]), we regard this paper complementary to the most recent one [22] for using block-wise ADMM-based schemes for the multiple-block separable convex minimization model (1.1).

References

[1] E. Blum and W. Oettli, Mathematische Optimierung. Grundlagen und Verfahren. Ökonometrie und Unternehmensforschung, Springer-Verlag, Berlin-Heidelberg-New York, 1975.

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

[3] C. H. Chen, B. S. He, Y. Y. Ye and X. M. Yuan, *The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent*, Math. Program., under revision.

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

[5] J. Eckstein and W. Yao, *Augmented Lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results*, manuscript, 2012.

[6] F. Facchinei and J. S. Pang, Finite-Dimensional Variational Inequalities and Complementarity problems, Volume I, Springer Series in Operations Research, Springer-Verlag, 2003.

[7] D. Gabay, Applications of the method of multipliers to variational inequalities, *Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems*, edited by M. Fortin and R. Glowinski, North Holland, Amsterdam, The Netherlands, 1983, pp. 299–331.

[8] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[9] R. Glowinski, T. Kärkkäinen and K. Majava, *On the convergence of operator-splitting methods*, in Numerical Methods for Scienfic computing, Variational Problems and Applications, edited by Y. Kuznetsov, P. Neittanmaki and O. Pironneau, Barcelona, 2003.

[10] R. Glowinski, *On alternating directon methods of multipliers: a historical perspective*, Springer Proceedings of a Conference Dedicated to J. Periaux, to appear.

[11] R. Glowinski and A. Marrocco, *Approximation par éléments finis d'ordre un et résolution par pénalisation-dualité d'une classe de problèmes non linéaires*, R.A.I.R.O., R2 (1975), pp. 41-76.

[12] D. R. Han, X. M. Yuan and W. X. Zhang, *An augmented-Lagrangian-based parallel splitting method for separable convex programming with applications to image processing*, Math. Comput., 83 (2014), pp. 2263-2291.

[13] B. S. He, L. S. Hou and X. M. Yuan, *On full Jacobian decomposition of the augmented Lagrangian method for separable convex programming*, SIAM J. Optim., under revision.

[14] B. S. He, H. Liu, J. Lu, and X. M. Yuan, *Application of the strictly contractive Peaceman-Rachford splitting method to multi-block convex programming*, manuscript, 2014.

[15] B. S. He, M. Tao and X. M. Yuan, *Alternating direction method with Gaussian back substitution for separable convex programming*, SIAM J. Optim., 22 (2012), pp. 313-340.

[16] B. S. He, M. Tao and X. M. Yuan, *Convergence rate and iteration complexity on the alternating direction method of multipliers with a substitution procedure for separable convex programming*, Math. Oper. Res., under revision.

[17] B. S. He, M. Tao and X. M. Yuan, *A splitting method for separable convex programming*, IMA J. Numer. Anal., to appear.

[18] B. S. He, H. K. Xu and X. M. Yuan, *On the proximal Jacobian decomposition of ALM for multiple-block separable convex minimization problems and its relationship to ADMM*, manuscript, 2014.

[19] B. S. He and X. M. Yuan, *On the $O(1/n)$ convergence rate of the alternating direction method*, SIAM J. Numer. Anal., 50 (2012), pp. 700-709.

[20] B. S. He and X. M. Yuan, *On nonergodic convergence rate of Douglas-Rachford alternating direction method of multipliers*, manuscript, 2012.

[21] B. S. He and X. M. Yuan, *Linearized alternating direction method with Gaussian back substitution for separable convex programming*, Numerical Algebra, Control and Optimization, 3(2)(2013), pp. 247-260.

[22] B. S. He and X. M. Yuan, *Block-wise alternating direction method of multipliers for multiple-block convex programming and Beyond*, manuscript, 2014.

[23] M. R. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appli., 4(1969), pp. 303-320.

[24] M. Hong and Z. Q. Luo, *On the linear convergence of the alternating direction method of multipliers*, manuscript, August 2012.

[25] P. L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964-979.

[26] B. Martinet, *Regularision d'inéquations variationnelles par approximations successive*, Revue Francaise d'Automatique et Informatique Recherche Opérationnelle, 126 (1970), pp. 154-159.

[27] Y. E. Nesterov, *Gradient methods for minimizing composite objective function*, Math. Prog., Ser. B, 140 (2013), pp. 125-161.

[28] G. B. Passty, *Ergodic convergence to a zero of the sum of monotone operators in Hilbert space*, J. Math. Analy. Applic. 72 (1979), pp. 383-390.

[29] Y. G. Peng, A. Ganesh, J. Wright, W. L. Xu and Y. Ma, *Robust alignment by sparse and low-rank decomposition for linearly correlated images*, IEEE Tran. Pattern Anal. Mach. Intel., 34 (2012), pp. 2233-2246.

[30] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, In Optimization edited by R. Fletcher, pp. 283-298, Academic Press, New York, 1969.

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

[32] X. F. Wang and X. M. Yuan, *The linearized alternating direction method for Dantzig Selector*, SIAM J. Sci. Comput., 34 (5) (2012), pp. A2792 - A2811.

[33] J. F. Yang and X. M. Yuan, Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization, Math. Comput., 82 (281) (2013), pp. 301-329.

[34] X. Q. Zhang, M. Burger and S. Osher, *A Unified Primal-Dual Algorithm Framework Based on Bregman Iteration*. J. Sci. Comput., 46 (2011), pp. 20-46.