Convergence rate and iteration complexity on the alternating direction method of multipliers with a substitution procedure for separable convex programming*

Bingsheng He,† Min Tao‡ and Xiaoming Yuan§

September 18, 2012

Abstract. Recently, in [17] we have showed the first possibility of combining the Douglas-Rachford alternating direction method of multipliers (ADMM) with a Gaussian back substitution procedure for solving a convex minimization model with a general separable structure. This paper is a further study on theoretical aspects of this theme. We first derive a general algorithmic framework to combine ADMM with either a forward or backward substitution procedure. Then, we show that convergence of this framework can be easily proved from contraction perspective, and its local linear convergence rate is provable if certain standard error bound condition is assumed. Without such an error bound assumption, we can still estimate the worst-case iteration complexity for this framework in both ergodic and nonergodic senses.

Keywords. Convex programming, Alternating direction method of multipliers, Convergence rate, Iteration complexity, Contraction methods

1 Introduction

We consider a structured convex minimization model with linear constraints and a separable objective function:

$$\begin{array}{ll} \min & \sum_{i=1}^{m} \theta_i(x_i) \\ & \sum_{i=1}^{m} A_i x_i = b; \\ & x_i \in \mathcal{X}_i, \quad i = 1, \cdots, m; \end{array} \tag{1.1}$$

where $\theta_i : \Re^{n_i} \to \Re$ $(i = 1, \ldots, m)$ are closed proper convex functions and they are not necessarily smooth; $\mathcal{X}_i \subseteq \Re^{n_i}$ $(i = 1, \ldots, m)$ are closed convex sets; $A_i \in \Re^{l \times n_i}$ $(i = 1, \ldots, m)$ are given matrices with full column ranks; and $b \in \Re^l$ is a given vector. Throughout, the solution set of (1.1) is assumed to be nonempty. Our discussion focuses on the particular case of (1.1) where $m \geq 3$, see e.g. [26, 29, 30, 31, 34] for such applications.

Except for problems in very small dimension, it is not wise to treat (1.1) as a generic convex programming and ignore its favorable separable structure when we try to design efficient numerical

*The first version was released on March 04, 2012.

†Department of Mathematics, Nanjing University, Nanjing, 210093, China. This author was supported by the NSFC grants 10971095 and 91130007, and the MOEC fund 20110091110004. Email: hebma@nju.edu.cn

‡Department of Mathematics, Nanjing University, Nanjing, 210093, China. Email: taomin0903@gmail.com

§Corresponding author. Department of Mathematics, Hong Kong Baptist University, Hong Kong, China. This author was supported by the Hong Kong General Research Fund: HKBU 203311. Email: xmyuan@hkbu.edu.hk

algorithms for (1.1). One obvious reason is that each single θ_i could be simple, while the aggregation of all θ_i's is hard, to be minimized for many concrete applications of the abstract model (1.1). We are thus in favor of such an algorithm that can take advantage of the separable structure of (1.1) well, or more precisely, can exploit the properties of all θ_i's fully by treating these functions individually rather than aggregatively. In the literature, structure-exploited algorithms have been well studied, but only for the special case of (1.1) where $m = 2$. A fundamental method in this regard is the Douglas-Rachford alternating direction method of multipliers (ADMM for short) proposed in [14] (see also [11]). More specifically, for solving (1.1) with $m = 2$, the standard ADMM iterative scheme is

$$\begin{cases} x_1^{k+1} = \arg\min \left\{ \theta_1(x_1) - x_1^T A_1^T \lambda^k + \frac{1}{2}\|A_1 x_1 + A_2 x_2^k - b\|_H^2 \mid x_1 \in \mathcal{X}_1 \right\}, \\ x_2^{k+1} = \arg\min \left\{ \theta_2(x_2) - x_2^T A_2^T \lambda^k + \frac{1}{2}\|A_1 x_1^{k+1} + A_2 x_2 - b\|_H^2 \mid x_2 \in \mathcal{X}_2 \right\}, \\ \lambda^{k+1} = \lambda^k - H(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b), \end{cases} \tag{1.2}$$

where $\lambda^k \in \Re^l$ is the Lagrange multiplier and the penalty matrix $H \in \Re^{l \times l}$ is positive definite. In applications, we usually choose H as a diagonal matrix: $H = \beta I_{l \times l}$ with $\beta > 0$. The standard ADMM scheme (1.2) shows that ADMM blends the benefits of dual decomposition and augmented Lagrangian method [20, 27], and thus it is possible to exploit the properties of θ_i's individually. We refer to, e.g. [4, 6, 9, 10, 12, 13, 16, 21, 32], for some earlier articles in the areas of partial differential equations, convex programming and variational inequalities. Moreover, recently ADMM has found many interesting applications in a broad spectrum of areas such as imaging processing, statistical learning and engineering, see e.g. [2, 3, 5, 7, 18, 23, 24, 28, 30, 33] to mention a few. Essentially, the main reason ensuring ADMM's efficiency for these concrete applications (where the functions θ_i's are often in particular forms) is that the decomposed subproblems in (1.2) are often simple enough to have closed-form solutions or can be solved efficiently up to high precisions. In the review paper on ADMM [2], the authors complimented that "ADMM is at least comparable to very specialized algorithms (even in the serial setting), and in most cases, the simple ADMM algorithm will be efficient enough to be useful".

To take advantage of each θ_i's property individually, a natural idea for solving the general case of (1.1) with $m \geq 3$ is to extend the iterative scheme (1.2) straightforwardly. This yields the following scheme for (1.1) when $m \geq 3$:

$$\begin{cases} x_1^{k+1} = \arg\min \left\{ \theta_1(x_1) - x_1^T A_1^T \lambda^k + \frac{1}{2}\|A_1 x_1 + \sum_{j=2}^m A_j x_j^k - b\|_H^2 \mid x_1 \in \mathcal{X}_1 \right\}; \\ x_2^{k+1} = \arg\min \left\{ \theta_2(x_2) - x_2^T A_2^T \lambda^k + \frac{1}{2}\|A_1 x_1^{k+1} + A_2 x_2 + \sum_{j=3}^m A_j x_j^k - b\|_H^2 \mid x_2 \in \mathcal{X}_2 \right\}; \\ \quad \cdots \\ x_i^{k+1} = \arg\min \left\{ \theta_i(x_i) - x_i^T A_i^T \lambda^k + \frac{1}{2}\| \sum_{j=1}^{i-1} A_j x_j^{k+1} + A_i x_i + \sum_{j=i+1}^m A_j x_j^k - b\|_H^2 \mid x_i \in \mathcal{X}_i \right\}; \\ \quad \cdots \\ x_m^{k+1} = \arg\min \left\{ \theta_m(x_m) - x_m^T A_m^T \lambda^k + \frac{1}{2}\| \sum_{j=1}^{m-1} A_j x_j^{k+1} + A_m x_m - b\|_H^2 \mid x_m \in \mathcal{X}_m \right\}; \\ \lambda^{k+1} = \lambda^k - H(\sum_{j=1}^m A_j x_j^{k+1} - b). \end{cases} \tag{1.3}$$

Just as the original ADMM (1.2), the iterative scheme (1.3) can be easily derived by decomposing the augmented Lagrangian function of (1.1) in the Gauss-Seidel fashion. In (1.3), the variables x_i are minimized in alternating order and the decomposed subproblems are much easier than the original problem (1.1) since only one function θ_i is involved in each of these subproblems. Then, the step of updating the Lagrange multiplier coordinates all these solutions to local small subproblems to find a solution to a global large problem. Note that the direct extension of ADMM scheme (1.3) reduces to the augmented Lagrangian method [20, 27] when $m = 1$ and the standard ADMM (1.2) scheme when $m = 2$.

Without further assumptions, the convergence of (1.3) for the general case where $m \geq 3$ still remains open despite that its efficiency has been verified empirically in [26, 30]. In [15], the convergence of (1.3) is shown but under the conditions that all θ_i are strongly convex and the penalty parameter β (when $H = \beta I_{l\times l}$) should be chosen judiciously in a certain interval. When each convex function θ_i in (1.1) is of particular structure and the update of λ^{k+1} in (1.3) is required to adopt a new step size rather than $H = \beta I_{l\times l}$, i.e.,

$$\lambda^{k+1} = \lambda^k - \tau(\sum_{j=1}^{m} A_j x_j^{k+1} - b), \tag{1.4}$$

where $\tau > 0$ is sufficiently small to fulfill certain error bound, the resulting scheme was proved to be convergent in [22]. In addition, this scheme is shown to be locally linearly convergent in [22] if all θ_i are further assumed to be smooth. In fact, the scheme (1.3) but with the new update of λ (1.4) studied in [22] is a dual ascent method.

Since sometimes it is not easy to verify whether the step size τ in (1.4) is small enough, we stick to the direct extension of ADMM (1.3) where the step size for updating λ is taken as the same as the penalty parameter (thus it is not necessarily very small) and the function θ_i's in (1.3) are only assumed to be generic nonsmooth convex functions, and study in which way the convergence of (1.3) can be derived. In [17], we have shown that the resulting sequence is convergent if the output of (1.3) is further corrected by a Gaussian back substitution procedure. The numerical efficiency of the Gaussian back substitution procedure, together with its superiority to some other work based on (1.3), have been illustrated numerically in [17, 25].

The Gaussian back substitution procedure in [17] still requires to compute the inverses of $(A_i^T H A_i)$ for $i = 2, \cdots, m-1$ (see the equation (3.4) in [17]), which could be expensive for generic A_i's. This paper is a further theoretical study on the combination of the ADMM scheme (1.3) with a substitution procedure. In particular, we aim at proposing an algorithmic framework to combine an ADMM procedure with a substitution procedure, where the substitution procedure can be in either a forward (i.e., correcting the output of (1.3) in the order of $x_2^{k+1} \to x_3^{k+1} \to \cdots \to x_m^{k+1} \to \lambda^{k+1}$) or backward (i.e., correcting the output of (1.3) in the order of $\lambda^{k+1} \to x_m^{k+1} \to x_{m-1}^{k+1} \to \cdots \to x_2^{k+1}$) fashion. Two algorithms are thus derived, and they both reduce to the original ADMM in [11, 14] for the case where $m = 2$ in (1.1). The forward and backward substitution procedures require no computation of any matrix's inverse, and they are both much more inexpensive computationally than the Gaussian back substitution procedure in [17] (see Sections 5.1 and 5.2 for elaboration). Moreover, we show that the convergence of this algorithmic framework can be easily proved from contraction perspective, and its local linear convergence rate is provable if certain standard error bound condition is assumed. Without such an error bound assumption, we can still estimate the worst-case iteration complexity for the new algorithms in both ergodic and nonergodic senses.

The rest of the paper is organized as follows. In Section 2, we provide some preliminaries which are useful for further discussions and summarize some notations for the convenience of discussion. In Section 3, we propose an ADMM procedure inspired by (1.3) and prove an inequality for the sequence generated by this ADMM procedure. In Section 4, we elaborate on the motivation of developing an appropriate substitution procedure to be combined with the ADMM procedure proposed in Section 3. Then, in Section 5, we specify the framework of ADMM with a substitution procedure into two concrete algorithms, and establish their global convergence and convergence rate in a unified framework from contraction perspective. In Sections 6 and 7, we estimate the worst-case iteration complexity for the proposed algorithms in an ergodic and nonergodic sense, respectively. Finally, we make some conclusions in Section 8.

2 Preliminaries

In this section, we first provide some preliminaries which are useful for our further discussions and then summarize some notations to be used.

2.1 A variational characterization of (1.1)

We first reformulate (1.1) as a variational form, which is useful for our subsequent analysis of convergence and estimate of worst-case iteration complexity. By attaching the Lagrange multiplier $\lambda \in \Re^l$ to the corresponding linear constraints, the Lagrange function of (1.1) is:

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

and it is defined on

$$\mathcal{W} := \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m \times \Re^l.$$

Let $\big(x_1^*, x_2^*, \ldots, x_m^*, \lambda^*\big)$ be a saddle point of the Lagrange function, then we have

$$L_{\lambda\in\Re^l}(x_1^*, x_2^*, \cdots, x_m^*, \lambda) \le L(x_1^*, x_2^*, \cdots, x_m^*, \lambda^*) \le L_{\binom{x_i \in \mathcal{X}_i}{i=1,\ldots,m}}(x_1, x_2, \ldots, x_m, \lambda^*).$$

Thus, finding a saddle point of $L(x_1, x_2, \ldots, x_m, \lambda)$ is equivalent to finding a vector

$$w^* = (x_1^*, x_2^*, ..., x_m^*, \lambda^*) \in \mathcal{W},$$

such that

$$\left\{ \begin{array}{rcl} \theta_1(x_1) - \theta_1(x_1^*) & + & (x_1 - x_1^*)^T(-A_1^T\lambda^*) \ge 0, \quad \forall\, x_1 \in \mathcal{X}_1, \\ \theta_2(x_2) - \theta_2(x_2^*) & + & (x_2 - x_2^*)^T(-A_2^T\lambda^*) \ge 0, \quad \forall\, x_2 \in \mathcal{X}_2, \\ & \vdots & \\ \theta_m(x_m) - \theta_m(x_m^*) & + & (x_m - x_m^*)^T(-A_m^T\lambda^*) \ge 0, \quad \forall\, x_m \in \mathcal{X}_m, \\ & & (\lambda - \lambda^*)^T\{\sum_{i=1}^m A_i x_i^* - b\} \ge 0, \quad \forall \lambda \in \Re^l. \end{array} \right. \tag{2.1}$$

More compactly, (2.1) can be rewritten as the following variational inequality (VI):

$$\mathrm{VI}(\mathcal{W}, F, \theta): \qquad \theta(x) - \theta(x^*) + (w - w^*)^T F(w^*) \ge 0, \quad \forall\; w \in \mathcal{W}, \tag{2.2a}$$

where

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \quad \theta(x) = \sum_{i=1}^{m} \theta_i(x_i), \quad w = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ \lambda \end{pmatrix} \quad \text{and} \quad F(w) = \begin{pmatrix} -A_1^T\lambda \\ -A_2^T\lambda \\ \vdots \\ -A_m^T\lambda \\ \sum_{i=1}^m A_i x_i - b \end{pmatrix}. \tag{2.2b}$$

Note that the operator $F(w)$ defined in (2.2b) is monotone as it is affine with a skew-symmetric matrix.

Let $\mathcal{W}^*$ be the solution set of $\mathrm{VI}(\mathcal{W}, F, \theta)$. Since we have assumed that the solution set of (1.1) is nonempty, $\mathcal{W}^*$ is also nonempty.

2.2 A Characterization of $\mathcal{W}^*$

We then propose a characterization of $\mathcal{W}^*$, which is the basis of our discussion for the estimate of worst-case iteration complexity in Sections 6 and 7. We refer to Theorem 2.3.5 in [8] and Theorem 2.1 in [19] for the proof of the following theorem.

Theorem 2.1. *The solution set of VI($\mathcal{W}, F, \theta$) is convex and it can be characterized as*

$$\mathcal{W}^* = \bigcap_{w\in\mathcal{W}} \big\{\tilde{w} \in \mathcal{W} : \theta(x) - \theta(\tilde{x}) + (w - \tilde{w})^T F(w) \geq 0\big\}. \tag{2.3}$$

2.3 Some notations

In this subsection, we summarize some notations which will be used in later analysis. These notations serve the purpose of making our presentation of analysis more compact.

First, revisit the iterative scheme of the straightforward extension of ADMM (1.3). It is easy to notice that the variable x_1 is not involved in the implementation of (1.3), just like the original ADMM (1.2). In other words, the input to execute the iteration of (1.3) is only the sequence $\{x_2^k, \cdots, x_m^k, \lambda^k\}$. Therefore, following [2], we call x_1 an intermediate variable. It is thus natural to introduce the notations $v = (x_2, \cdots, x_m, \lambda)$ and $\mathcal{V} = \mathcal{X}_2 \times \cdots \times \mathcal{X}_m \times \Re^l$ to differentiate the variables which are truly involved in the iteration from intermediate variables. Obviously, v is a sub-vector of w defined in (2.2b). The notations v^k is thus clear from the context. Accordingly, we also use the notation

$$\mathcal{V}^* = \{(x_2^*, \dots, x_m^*, \lambda^*) \,|\, (x_1^*, x_2^*, \dots, x_m^*, \lambda^*) \in \mathcal{W}^*\}.$$

Taking a closer look at (1.3), we can easily find that what this scheme really requires as the input of iteration is $\{A_2 x_2^k, \cdots, A_m x_m^k, \lambda^k\}$. Thus, it is convenient to define

$$u = \begin{pmatrix} u_2 \\ \vdots \\ u_m \\ u_\lambda \end{pmatrix} = \begin{pmatrix} H^{1/2} A_2 x_2 \\ \vdots \\ H^{1/2} A_m x_m \\ H^{-1/2}\lambda \end{pmatrix}, \tag{2.4}$$

from which the notation $u^k = \{u_2^k, \dots, u_m^k, u_\lambda^k\}$ is also clear. For some concrete applications of (1.1) such as those in [17, 25], the matrices $A_2, \cdots A_m$ are all identity matrices. For these cases, u reduces to v if $H = I$.

Accordingly, we also use the notation

$$\mathcal{U}^* = \left\{ u^* = \begin{pmatrix} u_2^* \\ \vdots \\ u_m^* \\ u_\lambda^* \end{pmatrix} = \begin{pmatrix} H^{1/2} A_2 x_2^* \\ \vdots \\ H^{1/2} A_m x_m^* \\ H^{-1/2}\lambda^* \end{pmatrix} \;\middle|\; w^* = (x_1^*, \dots, x_m^*, \lambda^*) \in \mathcal{W}^* \right\}.$$

With these notations, the scheme (1.3) can be summarized as generating the new iteration w^{k+1} with the input u^k.

Second, we introduce several matrices in block-wise form. With the help of these matrices, our notation for theoretical analysis will be much easier and more compact. Recall that the variable x_1 is an intermediate variable for the scheme (1.3). We thus introduce the matrix

$$\mathcal{A} = \mathrm{diag}(H^{1/2} A_2, H^{1/2} A_3, \cdots, H^{1/2} A_m, H^{-1/2} I_l) \tag{2.5}$$

to collect all the coefficient matrices with respect to the variables in (1.1) except x_1 and the identity matrix (corresponding to the Lagrange multiplier). With the definition (2.5), we obviously can relate the variables v and u as

$$u = \mathcal{A}v. \tag{2.6}$$

We also define two block matrices as follows

$$\mathcal{S} = (I_l, I_l, \cdots, I_l, -I_l), \tag{2.7}$$

$$\mathcal{I} = \mathrm{diag}(I_l, I_l, \cdots, I_l, I_l). \tag{2.8}$$

Then, we define

$$\mathcal{P} = \begin{pmatrix} I_l & 0 & \cdots & \cdots & 0 \\ I_l & I_l & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ I_l & \cdots & I_l & I_l & 0 \\ 0_l & \cdots & 0_l & 0_l & I_l \end{pmatrix}, \qquad \mathcal{N} = \begin{pmatrix} I_l & 0 & \cdots & \cdots & 0 \\ 0_l & I_l & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0_l & \cdots & 0_l & I_l & 0 \\ -I_l & \cdots & -I_l & -I_l & I_l \end{pmatrix}, \tag{2.9}$$

and

$$\mathcal{L} = \begin{pmatrix} I_l & 0 & \cdots & \cdots & 0 \\ I_l & I_l & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ I_l & \cdots & I_l & I_l & 0 \\ -I_l & \cdots & -I_l & -I_l & I_l \end{pmatrix} \tag{2.10}$$

which will be used in the substitution procedures to be proposed. Note that $\mathcal{P}, \mathcal{N}$ and $\mathcal{L}$ are all well structured lower triangular matrices, and there are related in the following equation:

$$\mathcal{L} = \mathcal{P}\mathcal{N}. \tag{2.11}$$

In addition, in the case $m = 2$, we have $\mathcal{P} = \mathcal{I}$ and $\mathcal{N} = \mathcal{L}$.

Finally, two more matrices are useful:

$$\widetilde{\mathcal{M}} = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ A_2^T H^{1/2} & 0 & \cdots & \cdots & 0 \\ A_3^T H^{1/2} & A_3^T H^{1/2} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ A_m^T H^{1/2} & A_m^T H^{1/2} & \cdots & A_m^T H^{1/2} & 0 \\ H^{-1/2} & H^{-1/2} & \cdots & H^{-1/2} & H^{-1/2} \end{pmatrix}, \tag{2.12}$$

and

$$\mathcal{M} = \begin{pmatrix} A_2^T H^{1/2} & 0 & \cdots & \cdots & 0 \\ A_3^T H^{1/2} & A_3^T H^{1/2} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ A_m^T H^{1/2} & A_m^T H^{1/2} & \cdots & A_m^T H^{1/2} & 0 \\ H^{-1/2} & H^{-1/2} & \cdots & H^{-1/2} & H^{-1/2} \end{pmatrix}. \tag{2.13}$$

As we shall show, these two matrices are mainly used to represent the accumulated iterative information in a compact way and thus simplify our notation. In fact, $\widetilde{\mathcal{M}}$ is used as the associated matrix to left-multiply a vector w^k while $\mathcal{M}$ is the associated matrix to left-multiply a vector v^k. Note that the absence of the intermediate variable x_1^k in the iteration to be proposed also explains why the first row of $\widetilde{\mathcal{M}}$ is a zero vector. Nevertheless, despite that the matrix $\mathcal{M}$ can be obtained by easily removing the first row of $\widetilde{\mathcal{M}}$, we give the explicit expressions of these two relevant matrices separately as we need to use both of them later.

Some relationships among the matrices defined above are summarized in the following lemmas. We omit their proofs since they are elementary.

Lemma 2.2. *Let the matrices $\mathcal{S}$, $\mathcal{I}$ and $\mathcal{L}$ be defined in (2.7), (2.8) and (2.10), respectively. Then, we have*

$$\mathcal{L}^T + \mathcal{L} = \mathcal{I} + \mathcal{S}^T\mathcal{S}. \tag{2.14}$$

Lemma 2.3. *Let the matrices $\mathcal{A}$, $\mathcal{L}$ and $\mathcal{M}$ be defined in (2.5), (2.10) and (2.13), respectively. Then, we have*

$$\mathcal{M} = \mathcal{A}^T\mathcal{L}. \tag{2.15}$$

The identities (2.14) and (2.15) in Lemmas 2.2 and 2.3 will be used in our theoretical analysis.

3 An ADMM procedure based on (1.3)

We have mentioned that an efficient strategy to overcome the lack of convergence of the scheme (1.3) is to supplement a substitution procedure to the output of (1.3), as the work in [17]. In this section, we propose an ADMM procedure which is slightly different from the scheme (1.3) only in the update of Lagrange multiplier.

3.1 An ADMM procedure

Since the notations x_i^{k+1}'s will be used to denote the new iterate after a substitution procedure, we use $\tilde{x}_i^k$'s to denote the output of an ADMM procedure. Our ADMM procedure based on the scheme (1.3) is as follows.

$$\left\{\begin{array}{l}
\tilde{x}_1^k = \arg\min\Big\{\theta_1(x_1) - x_1^TA_1^T\lambda^k + \frac{1}{2}\|A_1x_1 + \sum_{j=2}^m A_jx_j^k - b\|_H^2 \;\big|\; x_1\in\mathcal{X}_1\Big\};\\
\tilde{x}_2^k = \arg\min\Big\{\theta_2(x_2) - x_2^TA_2^T\lambda^k + \frac{1}{2}\|A_1\tilde{x}_1^k + A_2x_2 + \sum_{j=3}^m A_jx_j^k - b\|_H^2 \big| x_2\in\mathcal{X}_2\Big\};\\
\qquad\cdots\\
\tilde{x}_i^k = \arg\min\Big\{\theta_i(x_i) - x_i^TA_i^T\lambda^k + \frac{1}{2}\|\sum_{j=1}^{i-1}A_j\tilde{x}_j^k + A_ix_i + \sum_{j=i+1}^m A_jx_j^k - b\|_H^2 \big| x_i\in\mathcal{X}_i\Big\};\\
\qquad\cdots\\
\tilde{x}_m^k = \arg\min\Big\{\theta_m(x_m) - x_m^TA_m^T\lambda^k + \frac{1}{2}\|\sum_{j=1}^{m-1}A_j\tilde{x}_j^k + A_mx_m - b\|_H^2 \;\big|\; x_m\in\mathcal{X}_m\Big\}.
\end{array}\right. \tag{3.1a}$$

$$\tilde{\lambda}^k = \lambda^k - H(A_1\tilde{x}_1^k + \sum_{j=2}^m A_jx_j^k - b). \tag{3.1b}$$

Remark 3.1. The scheme (3.1) differs from the extended ADMM scheme (1.3) (also the method in [17]) only in the way of updating the Lagrange multiplier, i.e., (3.1b), and all the essential subproblems dominating the computation (i.e., (3.1a)) at each iteration are the same as those in (1.3). For this reason, we still call the scheme (3.1) an ADMM procedure. The only difference in the update of Lagrange multiplier indeed provides us the possibility to ensure the convergence for the combination of (3.1) with a substitution procedure which is much less expensive than the Gaussian back substitution in [17]. In fact, as we shall show in Theorem 3.3, this particular update (3.1b) enables us to derive an estimate on the accuracy of $\tilde{w}^k$ in (3.2) which is more succinct than the inequality (4.3) in [17]. To see the relationship between the outputs of (3.1) and (1.3), we set

$$x_1^{k+1} = \tilde{x}_1^k, \quad x_2^{k+1} = \tilde{x}_2^k, \quad \ldots\ldots, \quad x_m^{k+1} = \tilde{x}_m^k$$

and

$$\lambda^{k+1} = \tilde{\lambda}^k + H\big(\sum_{j=2}^{m} A_j(x_j^k - \tilde{x}_j^k)\big),$$

then the output of (1.3) is recovered. More compactly, by using the notations u, v and $\mathcal{N}$, the outputs u^{k+1} generated by (1.3) and $\tilde{u}^k$ by (3.1) are related in the following way:

$$u^{k+1} = u^k - \mathcal{N}(u^k - \tilde{u}^k).$$

Remark 3.2. Taking a closer look at the iterative schemes (1.3) and (3.1), it is easy to find that at each iteration the input to implement these schemes is actually $(A_2x_2^k, \ldots, A_mx_m^k, \lambda^k)$.

3.2 An important inequality

In the following theorem, we prove an important inequality for the output of the ADMM procedure (3.1), which will be used often in our further discussions including both the analysis of the substitution procedure in Section 5 and the derivation of the worst-case iteration complexity in Sections 6 and 7.

Theorem 3.3. *Let $\tilde{w}^k$ be generated by the ADMM procedure* (3.1) *with given v^k and $u = \mathcal{A}v$. Then, we have*

$$\tilde{w}^k \in \mathcal{W}, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq (u - \tilde{u}^k)^T \mathcal{L}(u^k - \tilde{u}^k), \quad \forall\, w \in \mathcal{W}, \tag{3.2}$$

where $\mathcal{L}$ is defined in (2.10).

Proof. Since $\tilde{x}_i^k$ is the solution of (3.1a), for $i = 1, 2, \ldots, m$, we have

$$\tilde{x}_i^k \in \mathcal{X}_i, \;\; \theta_i(x_i) - \theta_i(\tilde{x}_i^k) + (x_i - \tilde{x}_i^k)^T \big\{A_i^T[H(\sum_{j=1}^{i} A\tilde{x}_j^k + \sum_{j=i+1}^{m} A_jx_j^k - b) - \lambda^k]\big\} \geq 0, \;\; \forall\, x_i \in \mathcal{X}_i.$$

Substituting $\tilde{\lambda}^k = \lambda^k - H(A_1\tilde{x}_1^k + \sum_{j=2}^m A_jx_j^k - b)$ (see (3.1b)) in the above inequality, we obtain

$$\tilde{x}_i^k \in \mathcal{X}_i, \;\; \theta_i(x_i) - \theta_i(\tilde{x}_i^k) + (x_i - \tilde{x}_i^k)^T \big\{-A_i^T\tilde{\lambda}^k + A_i^T H\big(\sum_{j=2}^{i}(A_j\tilde{x}_j^k - A_jx_j^k)\big)\big\} \geq 0, \;\; \forall\, x_i \in \mathcal{X}_i.$$

Summing the above inequality over $i = 1, \ldots, m$, we obtain $\tilde{w}^k \in \mathcal{W}$ and

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_i - \tilde{x}_i^k \\ \vdots \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A_1^T\tilde{\lambda}^k \\ -A_2^T\tilde{\lambda}^k \\ \vdots \\ -A_i^T\tilde{\lambda}^k \\ \vdots \\ -A_m^T\tilde{\lambda}^k \end{pmatrix} + \begin{pmatrix} 0 \\ A_2^T H\big(\sum_{j=2}^{2}(A_j\tilde{x}_j^k - A_jx_j^k)\big) \\ \vdots \\ A_i^T H\big(\sum_{j=2}^{i}(A_j\tilde{x}_j^k - A_jx_j^k)\big) \\ \vdots \\ A_m^T H\big(\sum_{j=2}^{m}(A_j\tilde{x}_j^k - A_jx_j^k)\big) \end{pmatrix} \right\} \geq 0, \forall\, w \in \mathcal{W}. \tag{3.3}$$

Because $A\tilde{x}_1^k + \sum_{j=2}^m A_j x_j^k - b + H^{-1}(\tilde{\lambda}^k - \lambda^k) = 0$ (see (3.1b) again), we have

$$(\sum_{j=1}^m A_j \tilde{x}_j^k - b) + H^{-1}(\tilde{\lambda}^k - \lambda^k) - \sum_{j=2}^m (A_j \tilde{x}_j^k - A_j x_j^k) = 0. \tag{3.4}$$

Combining (3.3) and (3.4) together, we get $\tilde{w}^k \in \mathcal{W}$ and

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_i - \tilde{x}_i^k \\ \vdots \\ x_m - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A_1^T \tilde{\lambda}^k \\ -A_2^T \tilde{\lambda}^k \\ \vdots \\ -A_i^T \tilde{\lambda}^k \\ \vdots \\ -A_m^T \tilde{\lambda}^k \\ \sum_{j=1}^m A_j \tilde{x}_j^k - b \end{pmatrix} + \begin{pmatrix} 0 \\ A_2^T H\big(\sum_{j=2}^2 (A_j \tilde{x}_j^k - A_j x_j^k)\big) \\ \vdots \\ A_i^T H\big(\sum_{j=2}^i (A_j \tilde{x}_j^k - A_j x_j^k)\big) \\ \vdots \\ A_m^T H\big(\sum_{j=2}^m (A_j \tilde{x}_j^k - A_j x_j^k)\big) \\ H^{-1}(\tilde{\lambda}^k - \lambda^k) - \sum_{j=2}^m (A_j \tilde{x}_j^k - A_j x_j^k) \end{pmatrix} \right\} \ge 0,$$

for all $w \in \mathcal{W}$. Using the notations of F (see (2.2b)), u (see (2.4)) and $\widetilde{\mathcal{M}}$ (see (2.12)), the above inequality can be rewritten into

$$\tilde{w}^k \in \mathcal{W}, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T \{F(\tilde{w}^k) - \widetilde{\mathcal{M}}\big(u^k - \tilde{u}^k\big)\} \ge 0, \quad \forall\, w \in \mathcal{W}. \tag{3.5}$$

Recall the definitions of $\widetilde{\mathcal{M}}$ and $\mathcal{M}$. We then have

$$(w - \tilde{w}^k)^T \widetilde{\mathcal{M}}\big(u^k - \tilde{u}^k\big) = (v - \tilde{v}^k)^T \mathcal{M}(u^k - \tilde{u}^k).$$

Because $\mathcal{M} = \mathcal{A}^T \mathcal{L}$ and $\mathcal{A}v = u$ (see (2.15) and (2.6)), we have

$$(w - \tilde{w}^k)^T \widetilde{\mathcal{M}}\big(u^k - \tilde{u}^k\big) = (u - \tilde{u}^k)^T \mathcal{L}\big(u^k - \tilde{u}^k\big).$$

The assertion (3.2) thus follows from (3.5) and the last equality immediately. □

4 The motivation of finding a substitution procedure: From the contraction perspective

As we have mentioned, the output of (3.1) needs to be further corrected to yield convergence and our strategy for the correction is to propose a substitution procedure whose computation is extremely easy. The idea inspiring the substitution strategy comes from the fact that the output of (3.1) can be corrected easily such that the corrected sequence is contractive to the set $\mathcal{U}^*$. Note that we follow the classical definition of a contractive sequence in [1]. Then, we show that the correction step on contraction purpose can be easily executed in either a forward or backward substitution fashion, because of the particular structure of the matrix $\mathcal{M}$. We would emphasize that Theorem 3.3 plays an important role for the coming analysis.

First, with the respective definitions of $\mathcal{A}$ and $\mathcal{L}$ in (2.5) and (2.10), we can prove two obvious propositions regarding the output of the ADMM procedure (3.1).

Proposition 4.1. *Let $\tilde{w}^k$ be generated by the ADMM procedure* (3.1) *with given v^k and $u = \mathcal{A}v$. Then, we have*

$$(u^k - u^*)^T \mathcal{L}(u^k - \tilde{u}^k) \ge (u^k - \tilde{u}^k)^T \mathcal{L}(u^k - \tilde{u}^k), \quad \forall\, u^* \in \mathcal{U}^*, \tag{4.1}$$

where $\mathcal{L}$ is defined in (2.10).

Proof. The proof is an immediate conclusion based on the assertion (3.2) and the monotonicity of F. In fact, for an arbitrarily fixed $w^* \in \mathcal{W}^*$, it follows from (3.2) that

$$(\tilde{u}^k - u^*)^T \mathcal{L}(u^k - \tilde{u}^k) \geq (\tilde{w}^k - w^*)^T F(\tilde{w}^k) + \theta(\tilde{x}^k) - \theta(x^*), \ \ \forall w^* \in \mathcal{W}^*.$$

Using the monotonicity of F and the optimality of w^*, we have

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

The above two inequalities imply that

$$(\tilde{u}^k - u^*)^T \mathcal{L}(u^k - \tilde{u}^k) \geq 0, \ \ \forall\, u^* \in \mathcal{U}^*,$$

and the assertion (4.1) follows from the last inequality immediately. □

Proposition 4.2. *Let $\tilde{w}^k$ be generated by the ADMM procedure* (3.1) *with given v^k and $u = \mathcal{A}v$. Then, we have*

$$(u^k - \tilde{u}^k)^T \mathcal{L}(u^k - \tilde{u}^k) = \frac{1}{2}\|u^k - \tilde{u}^k\|^2 + \frac{1}{2}\|\mathcal{S}(u^k - \tilde{u}^k)\|^2, \tag{4.2}$$

where $\mathcal{L}$ and $\mathcal{S}$ are defined in (2.10) *and* (2.7)*, respectively.*

Proof. First, we have

$$(u^k - \tilde{u}^k)^T \mathcal{L}(u^k - \tilde{u}^k) = \frac{1}{2}(u^k - \tilde{u}^k)^T (\mathcal{L}^T + \mathcal{L})(u^k - \tilde{u}^k).$$

Then, using the identity (2.14) in Lemma 2.2, the assertion (4.2) is proved immediately. □

Remark 4.3. The assertions (3.2) and (4.2) jointly imply that if $u^k - \tilde{u}^k = 0$, then we have

$$\tilde{w}^k \in \mathcal{W}, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq 0, \quad \forall\, w \in \mathcal{W},$$

which means that the output $\tilde{w}^k$ of the ADMM procedure (3.1) is a solution point of VI$(\mathcal{W}, F, \theta)$ because of Theorem 2.1. Therefore, the result (4.2) provides us an easy stopping criterion to terminate the iteration (3.1):

$$\|u^k - \tilde{u}^k\| \leq \epsilon,$$

where ϵ is the tolerance set by users.

Now, with Propositions 4.1 and 4.2, we have

$$(u^k - u^*)^T \mathcal{L}(u^k - \tilde{u}^k) \geq \frac{1}{2}\|u^k - \tilde{u}^k\|^2 + \frac{1}{2}\|\mathcal{S}(u^k - \tilde{u}^k)\|^2, \quad \forall\, u^* \in \mathcal{U}^*. \tag{4.3}$$

and it becomes apparent how to correct the output of (3.1) in the contraction way. More specifically, whenever $u^k - \tilde{u}^k \neq 0$, the assertion (4.2) shows the positivity of the term $(u^k - \tilde{u}^k)^T \mathcal{L}(u^k - \tilde{u}^k)$, and thus the assertion (4.1) indicates that the direction $-\mathcal{L}(u^k - \tilde{u}^k)$ is beneficial for reducing the proximity to the solution set $\mathcal{U}^*$ if the current iterate $\tilde{u}^k$ moves along this direction with an appropriate step size. That is, the spirit of contraction type methods (see [1]) is applicable. More explicitly, the new iterate u^{k+1} can be generated by

$$u^{k+1} = u^k - \alpha G^{-1} \mathcal{L}(u^k - \tilde{u}^k), \tag{4.4}$$

where G is an arbitrarily symmetric positive definite matrix with the same dimensionality as $\mathcal{L}$. Then, with an appropriate choice of the step size α, we can prove that the sequence $\{u^k\}$ generated by (4.4) is contractive with respect to the set $\mathcal{U}^*$ under the G-norm. In other words, the scheme (4.4) can be used to correct the output of (3.1) on contraction purpose.

To see why the scheme (4.4) yields a contractive sequence, we have the following easy fact:

$$\begin{aligned}
&\|u^k - u\|_G^2 - \|u^{k+1} - u\|_G^2 \\
&= \|u^k - u\|_G^2 - \|(u^k - u) - \alpha G^{-1}\mathcal{L}(u^k - \tilde{u}^k)\|_G^2 \\
&= 2\alpha(u^k - u)^T\mathcal{L}(u^k - \tilde{u}^k) - \alpha^2\|G^{-1}\mathcal{L}(u^k - \tilde{u}^k)\|_G^2. \qquad (4.5)
\end{aligned}$$

Set $u = u^*$ in (4.5), and use (4.1) and (4.2), we get

$$\begin{aligned}
&\|u^k - u^*\|_G^2 - \|u^{k+1} - u^*\|_G^2 \\
&\geq 2\alpha(u^k - \tilde{u}^k)^T\mathcal{L}(u^k - \tilde{u}^k) - \alpha^2\|G^{-1}\mathcal{L}(u^k - \tilde{u}^k)\|_G^2 \\
&= \alpha\big(\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2\big) - \alpha^2\|G^{-1}\mathcal{L}(u^k - \tilde{u}^k)\|_G^2, \ \ \forall u^* \in \mathcal{U}^*. \qquad (4.6)
\end{aligned}$$

Note that right-hand side of (4.6) is a quadratic function of α. In order to obtain the closest proximity to $\mathcal{U}^*$, we are in the desire to maximize this quadratic function and this promotes us to take the optimal value of α as

$$\alpha = \alpha_k := \frac{\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2}{2\|G^{-1}\mathcal{L}(u^k - \tilde{u}^k)\|_G^2}. \qquad (4.7)$$

With this choice of step size, it follows from (4.6) that

$$\|u^{k+1} - u^*\|_G^2 \leq \|u^k - u^*\|_G^2 - \frac{1}{2}\alpha_k\big(\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2\big), \ \forall u^* \in \mathcal{U}^*. \qquad (4.8)$$

Since $\alpha_k > 0$ whenever $u^k - \tilde{u}^k \neq 0$, (4.8) shows that the sequence $\{u^k\}$ generated by the scheme (4.4) is contractive to $\mathcal{U}^*$ if the step size is appropriately chosen.

Indeed, it is flexible to choose different positive definite matrices for G in the generic scheme (4.4) if our purpose is only to ensure that the sequence $\{u^k\}$ is contractive with respect to $\mathcal{U}^*$. In order to induce simple substitution procedures which are computationally inexpensive, two specific interesting choices are

$$G = \mathcal{I} \qquad \text{or} \qquad G = \mathcal{P}\mathcal{P}^T.$$

In fact, these two choices yield two efficient substitution procedures to correct the output of (3.1), as we elaborate below.

- If $G = \mathcal{I}$, then the scheme (4.4) reduces to

  $$u^{k+1} - u^k = \alpha\mathcal{L}(\tilde{u}^k - u^k), \qquad (4.9)$$

  and it can be rewritten as

  $$\mathcal{L}^{-1}(u^{k+1} - u^k) = \alpha(\tilde{u}^k - u^k). \qquad (4.10)$$

  Recall that the matrix $\mathcal{L}$ defined in (2.10) is a block unit lower triangular matrix and

  $$\mathcal{L}^{-1} = \begin{pmatrix} I_l & 0 & \cdots & \cdots & 0 \\ -I_l & I_l & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & -I_l & I_l & 0 \\ 0 & \cdots & 0 & I_l & I_l \end{pmatrix},$$

Thus, the implementation of (4.10) is essentially a forward substitution procedure where the new iterate u^{k+1} is yielded in the order of $u_2^{k+1} \to u_3^{k+1} \to \cdots \to u_m^{k+1} \to \lambda^{k+1}$. Overall, with the given u^k and the output $\tilde{u}^k$ of (3.1), the new iterative u^{k+1} can be generated via the forward substitution procedure (4.9) and the resulting sequence $\{u^k\}$ is contractive to the set $\mathcal{U}^*$ provided that the step size α is chosen appropriately.

- If $G = \mathcal{P}\mathcal{P}^T$ in (4.4), then the inequality (4.3) can be written as

$$\langle G(u^k - u^*), G^{-1}\mathcal{L}(u^k - \tilde{u}^k)\rangle \geq \frac{1}{2}\|u^k - \tilde{u}^k\|^2 + \frac{1}{2}\|\mathcal{S}(u^k - \tilde{u}^k)\|^2, \quad \forall\, u^* \in \mathcal{U}^*. \tag{4.11}$$

In this way, $G^{-1}\mathcal{L}(\tilde{u}^k - u^k)$ can be viewed as a descent direction of $\|u - u^*\|_G^2$ at the point u^k. By using $G = \mathcal{P}\mathcal{P}^T$ and (2.11), the scheme (4.4) can be rewritten as

$$\mathcal{P}^T(u^{k+1} - u^k) = \alpha\mathcal{P}^{-1}\mathcal{L}(\tilde{u}^k - u^k) = \alpha\mathcal{N}(\tilde{u}^k - u^k), \tag{4.12}$$

which is essentially a backward substitution procedure to yield the new iterate u^{k+1} in the order of $\lambda^{k+1} \to u_m^{k+1} \to u_{m-1}^{k+1} \to \cdots \to u_2^{k+1}$. Also, the resulting sequence $\{u^k\}$ is contractive with respect to the set $\mathcal{U}^*$ provided that the step size α is chosen appropriately.

Therefore, a unified framework of the combination of ADMM (see (3.1)) with a substitution procedure (see (4.4)) is proposed from the contraction perspective. In addition, two concrete substitution procedures (see (4.9) or (4.12)) which are computationally inexpensive are ready to be combined with the ADMM procedure (3.1) and these combinations lead to convergence again because of their contraction nature.

5 Alternating direction method of multipliers with a substitution procedure

In this section, we specify the choices of step size in (4.9) and (4.12), and thus derive two concrete algorithms by combining the ADMM procedure (3.1) with a substitution procedure. Then, we prove the global convergence of these two algorithms in a unified way, by following the analytic framework of contraction methods.

5.1 Alternating direction method of multipliers with a forward substitution

We first combine the ADMM procedure (3.1) with the forward substitution (4.9) for solving (1.1).

Algorithm 1: Alternating direction method with a forward substitution procedure.
Let $\gamma \in (0, 2)$, $\mathcal{A}$, $\mathcal{S}$ and $\mathcal{L}$ be defined in (2.5), (2.7) and (2.10), respectively. Start with u^0. With the given iterate u^k, the new iterate u^{k+1} is given as follows.
Step 1. **ADMM procedure (prediction step)**. Execute the scheme (3.1) to generate $\tilde{w}^k$ and thus $\tilde{v}^k$. Set $\tilde{u}^k = \mathcal{A}\tilde{v}^k$.
Step 2. **Forward substitution procedure (correction step)**. Generate the new iterate u^{k+1} via:

$$u^{k+1} - u^k = \alpha_k \mathcal{L}(\tilde{u}^k - u^k). \tag{5.1a}$$

where

$$\alpha_k = \gamma\alpha_k^F \quad \text{with} \quad \alpha_k^F = \frac{\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2}{2\|\mathcal{L}(u^k - \tilde{u}^k)\|^2}. \tag{5.1b}$$

Remark 5.1. The strategy of determining α_k^F in (5.1b) is obtained by taking $G = \mathcal{I}$ in (4.7), and as we have shown, it is for the purpose to ensure that the sequence $\{u^k\}$ is contractive to the set $\mathcal{U}^*$. The reason we attach the relaxation factor γ to α_k^F can be intuitively explained in the following way: we determine the step size α by maximizing the quadratic function of α in the right-hand side of (4.6), which is merely a lower bound of the true proximity progress. Thus, the optimal value of α calculated based on this quadratic function should be somehow relaxed to some extent, and this is the role of the relaxation factor γ. The restriction of γ into $(0, 2)$ is to ensure the contractive property of the iterative sequence $\{u^k\}$, see (5.8).

Recall that the proposed Algorithm 1 is gained by taking $G = \mathcal{I}$ in (4.4). Thus, the contraction of the sequence $\{u^k\}$ generated by Algorithm 1 is an immediate conclusion of the inequalities (4.6) and (4.8). For completeness, we summarize these results in the following theorem for the special case of $G = \mathcal{I}$. First, we emphasize that the forward substitution procedure with a undetermined step size α is

$$u^{k+1} = u^k + \alpha\mathcal{L}(\tilde{u}^k - u^k). \tag{5.2}$$

Theorem 5.2. *Let $\tilde{w}^k$ be generated by the ADMM procedure* (3.1) *with given u^k and $\tilde{u}^k = \mathcal{A}\tilde{v}^k$. If the new iterate u^{k+1} is updated by (5.2), then we have*

$$\|u^k - u^*\|^2 - \|u^{k+1} - u^*\|^2 \geq q^F(\alpha), \ \ \forall\, u^* \in \mathcal{U}^*, \tag{5.3}$$

where

$$q^F(\alpha) = \alpha\big(\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2\big) - \alpha^2\|\mathcal{L}(u^k - \tilde{u}^k)\|^2. \tag{5.4}$$

Proof. This is a special case of (4.6) by taking $G = \mathcal{I}$. □

The following corollary shows that the equality is achieved in (5.3) if u^* is replaced by the iterate $\tilde{u}^k$ generated by the ADMM procedure (3.1). This result will be used in Section 6 to estimate a worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity for Algorithm 1.

Corollary 5.3. *Let $\tilde{w}^k$ be generated by the ADMM procedure* (3.1) *with given u^k and $\tilde{u}^k = \mathcal{A}\tilde{v}^k$. If the new iterate u^{k+1} is updated by (5.2) and $q^F(\alpha)$ be defined in (5.4), then we have*

$$\|u^k - \tilde{u}^k\|^2 - \|u^{k+1} - \tilde{u}^k\|^2 = q^F(\alpha). \tag{5.5}$$

Proof. Setting $u = \tilde{u}^k$ in (4.5) with $G = \mathcal{I}$, we get

$$\|u^k - \tilde{u}^k\|^2 - \|u^{k+1} - \tilde{u}^k\|^2 = 2\alpha(u^k - \tilde{u}^k)^T\mathcal{L}(u^k - \tilde{u}^k) - \alpha^2\|\mathcal{L}(u^k - \tilde{u}^k)\|^2. \tag{5.6}$$

Using (4.2) and the definition $q^F(\alpha)$ in (5.4), the proof is complete. □

Now, we are ready to show that the sequence $\{u^k\}$ generated by the proposed Algorithm 1 is contractive with respect to the set $\mathcal{U}^*$, starting from the following lemma.

Lemma 5.4. *Let α_k^F be defined in (5.1b). Then we have*

$$\alpha_k^F \geq \frac{1}{2\|\mathcal{L}^T\mathcal{L}\|}. \tag{5.7}$$

Proof. It is trivial from (5.1b). □

Theorem 5.5. *Let the sequence $\{u^k\}$ be generated by the proposed Algorithm 1. Then we have*

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \frac{\gamma(2-\gamma)}{4\|\mathcal{L}^T\mathcal{L}\|}\|u^k - \tilde{u}^k\|^2, \quad \forall\, u^* \in \mathcal{U}^*. \tag{5.8}$$

Proof. Using (5.4), by a manipulation, we obtain

$$\begin{array}{rcl} q^F(\alpha_k) &=& \gamma\alpha_k^F\big(\|u^k-\tilde{u}^k\|^2+\|\mathcal{S}(u^k-\tilde{u}^k)\|^2\big)-\gamma^2\alpha_k^F(\alpha_k^F\|\mathcal{L}(u^k-\tilde{u}^k)\|^2) \\ &\overset{(5.1b)}{=}& \gamma(1-\frac{\gamma}{2})\alpha_k^F\big(\|u^k-\tilde{u}^k\|^2+\|\mathcal{S}(u^k-\tilde{u}^k)\|^2\big) \\ &\overset{(5.7)}{\geq}& \frac{\gamma(2-\gamma)}{4\|\mathcal{L}^T\mathcal{L}\|}\|u^k-\tilde{u}^k\|^2. \end{array}$$

The assertion (5.8) follows from (5.3) immediately. □

Remark 5.6. For the special case where $m=2$, if we determine the step size α_k in a simpler way: $\alpha_k \equiv 1$, then the forward substitution procedure (see (5.1))

$$u^{k+1}-u^k=\mathcal{L}(\tilde{u}^k-u^k)$$

for this special scenario becomes

$$\begin{pmatrix} H^{1/2}A_2(x_2^{k+1}-x_2^k) \\ H^{-1/2}(\lambda^{k+1}-\lambda^k) \end{pmatrix} = \begin{pmatrix} I_l & 0 \\ -I_l & I_l \end{pmatrix} \begin{pmatrix} H^{1/2}A_2(\tilde{x}_2^k-x_2^k) \\ H^{-1/2}(\tilde{\lambda}^k-\lambda^k) \end{pmatrix}. \tag{5.9}$$

In this case, we have (see (5.4))

$$\begin{array}{rcl} q^F(1) &=& \|u^k-\tilde{u}^k\|^2+\|\mathcal{S}(u^k-\tilde{u}^k)\|^2-\|\mathcal{L}(u^k-\tilde{u}^k)\|^2 \\ &=& \left\| \begin{matrix} H^{1/2}A_2(x_2^k-\tilde{x}_2^k) \\ H^{-1/2}(\lambda^k-\tilde{\lambda}^k) \end{matrix} \right\|^2 + \|H^{1/2}A_2(x_2^k-\tilde{x}_2^k)-H^{-1/2}(\lambda^k-\tilde{\lambda}^k)\|^2 \\ && -\left\| \begin{pmatrix} I_l & 0 \\ -I_l & I_l \end{pmatrix} \begin{pmatrix} H^{1/2}A_2(x_2^k-\tilde{x}_2^k) \\ H^{-1/2}(\lambda^k-\tilde{\lambda}^k) \end{pmatrix} \right\|^2 \\ &=& \|\lambda^k-\tilde{\lambda}^k\|_{H^{-1}}^2, \end{array} \tag{5.10}$$

and thus $q^F(1)\geq 0$. Combining (5.3) and the last equality, we obtain

$$\left\| \begin{matrix} H^{1/2}A_2(x_2^{k+1}-x_2^*) \\ H^{-1/2}(\lambda^{k+1}-\lambda^*) \end{matrix} \right\|^2 \leq \left\| \begin{matrix} H^{1/2}A_2(x_2^k-x_2^*) \\ H^{-1/2}(\lambda^k-\lambda^*) \end{matrix} \right\|^2 - \|\lambda^k-\tilde{\lambda}^k\|_{H^{-1}}^2, \quad \forall\, u^*\in\mathcal{U}^*.$$

Because

$$u^0=\begin{pmatrix} H^{1/2}A_2x_2^0 \\ H^{-1/2}\lambda^0 \end{pmatrix}, \quad \tilde{u}^k=\begin{pmatrix} H^{1/2}A_2\tilde{x}_2^k \\ H^{-1/2}\tilde{\lambda}^k \end{pmatrix} \quad \text{and} \quad A_2x_2^{k+1}=A_2\tilde{x}_2^k \ \text{(see (5.9))},$$

combining the definition of $\tilde{\lambda}^k$ (see (3.1b)), the method is equivalent to

$$\left\{ \begin{array}{rcl} x^{k+1} &=& \tilde{x}^k, \\ \lambda^{k+1} &=& \lambda^k-H(A_1x_1^{k+1}+A_2x_2^{k+1}-b), \end{array} \right.$$

which is the ADMM scheme (1.2). Therefore, for the case $m=2$, Algorithm 1 with $\alpha_k\equiv 1$ is exactly the standard ADMM. Based on the above analysis, the proposed Algorithm 1 can be regarded as an extension of the standard ADMM (1.2) from the special case of (1.1) with $m=2$ to the general case of (1.1) with a generic m.

The computation of the step size α_k^F in (5.1b) is already inexpensive thanks to the simplicity of $\mathcal{S}$ and $\mathcal{L}$ (see (2.7) and (2.10)). Moreover, in (5.7) we have shown a unified lower bound of this step size for any m. This bound is certainly in conservative nature as it applies to the generic case with an arbitrary m. For a specific given m, one may be more interested in a bound more accurate than this unified bound. Also, one may ask the question if the step size α_k can be fixed as a constant so that the computation of the forward substitution procedure (5.1a) can be further alleviated. The answers to these question are crucially dependent on the possibility of ensuring the positivity of $q^F(\alpha)$. In the remark above, we has shown that α_k could be fixed as 1 for the case where $m = 2$ in (1.1).

According to (5.4),

$$q^F(\alpha) \geq 0 \quad \Longleftrightarrow \quad \|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2 \geq \alpha \|\mathcal{L}(u^k - \tilde{u}^k)\|^2.$$

Hence, we need only to take α such that

$$(\mathcal{I} + \mathcal{S}^T\mathcal{S}) - \alpha \mathcal{L}^T\mathcal{L} \succ 0,$$

in order to ensure $q^F(\alpha) \geq 0$ (thus the contraction and convergence). For any fixed $m \geq 3$, we define

$$\alpha_*^F = \sup\{\alpha \,|\, (\mathcal{I} + \mathcal{S}^T\mathcal{S}) - \alpha \mathcal{L}^T\mathcal{L} \succeq 0\}. \tag{5.11}$$

Theorem 5.7. *Let $\{u^k\}$ be the sequence generated by the proposed Algorithm 1 with a constant step size $\alpha \in (0, \alpha_*^F)$, where α_*^F is defined in* (5.11)*. Then we have*

$$q^F(\alpha) \geq \alpha(\alpha_*^F - \alpha)\|\mathcal{L}(u^k - \tilde{u}^k)\|^2, \tag{5.12}$$

and consequently

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \alpha(\alpha_*^F - \alpha)\|\mathcal{L}(u^k - \tilde{u}^k)\|^2, \quad \forall\, u^* \in \mathcal{U}^*. \tag{5.13}$$

Proof. Using (5.4), we have

$$q^F(\alpha) = \alpha\big((\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2) - \alpha\|\mathcal{L}(u^k - \tilde{u}^k)\|^2)\big).$$

According to (5.11), we have

$$(\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2) \geq \alpha_*^F \|\mathcal{L}(u^k - \tilde{u}^k)\|^2$$

and thus (5.12) is proved. The assertion (5.13) follows from (5.3) and (5.12) immediately. □

We take the case where $m = 3$ as an example for detailed explanation. Since

$$\mathcal{I} + \mathcal{S}^T\mathcal{S} = \begin{pmatrix} 2I & I & -I \\ I & 2I & -I \\ -I & -I & 2I \end{pmatrix} \qquad \text{and} \qquad \mathcal{L}^T\mathcal{L} = \begin{pmatrix} 3I & 2I & -I \\ 2I & 2I & -I \\ -I & -I & I \end{pmatrix},$$

we investigate the determinate of $[(\mathcal{I} + \mathcal{S}^T\mathcal{S}) - \alpha\mathcal{L}^T\mathcal{L}]$,

$$\Delta(\alpha) := \begin{vmatrix} 2-3\alpha & 1-2\alpha & \alpha-1 \\ 1-2\alpha & 2-2\alpha & \alpha-1 \\ \alpha-1 & \alpha-1 & 2-\alpha \end{vmatrix}. \tag{5.14}$$

By a careful manipulation, we have

$$\begin{aligned}
\Delta(\alpha) &= -2(\alpha-1)(\alpha-2)(3\alpha-2) + \{(2(1-2\alpha) + (2\alpha-2) + (3\alpha-2))\}(\alpha-1)^2 \\
&\quad +(\alpha-2)(2\alpha-1)^2 \\
&= -2(\alpha-1)(\alpha-2)(3\alpha-2) + (\alpha-2)(\alpha-1)^2 + (\alpha-2)(2\alpha-1)^2 \\
&= -(\alpha-2)\{2(\alpha-1)(3\alpha-2) - (\alpha-1)^2 - (2\alpha-1)^2\} \\
&= -(\alpha-2)(\alpha^2 - 4\alpha + 2) \\
&= -\big(\alpha - (2-\sqrt{2})\big)\big(\alpha-2\big)\big(\alpha - (2+\sqrt{2})\big),
\end{aligned}$$

therefore, it follows that

$$\Delta(\alpha) \ge 0, \quad \forall\, \alpha \in \big(0, (2-\sqrt{2})\big].$$

In addition, for such $\alpha \in \big(0, (2-\sqrt{2})\big]$, it is easy to check that

$$2-3\alpha > 0 \quad \text{and} \quad \begin{vmatrix} 2-3\alpha & 1-2\alpha \\ 1-2\alpha & 2-2\alpha \end{vmatrix} > 0.$$

Thus,

$$(\mathcal{I} + \mathcal{S}^T\mathcal{S}) - \alpha\mathcal{L}^T\mathcal{L} \succeq 0, \quad \forall\, \alpha \in \big(0, (2-\sqrt{2})\big].$$

For $m = 3$, $\alpha_*^F = 2-\sqrt{2}$. Therefore, by using the forward substitution, in (5.1a), we can take fixed $\alpha_k \equiv \alpha \in \big(0, (2-\sqrt{2})\big)$ for the case $m = 3$ in (1.1).

5.2 Alternating direction method of multipliers with a backward substitution

In this subsection, we combine the ADMM procedure (3.1) with the backward substitution (4.12) for solving (1.1).

Algorithm 2: Alternating direction method of multipliers with a backward substitution.

Let $\gamma \in (0,2)$, $\mathcal{A}$, $\mathcal{S}$ and $\mathcal{L}$ be defined in (2.5), (2.7) and (2.10), respectively. Start with u^0. With the given iterate u^k, the new iterate u^{k+1} is given as follows.

Step 1. **ADMM procedure (prediction step)**. Execute the scheme (3.1) to generate $\tilde{w}^k$ and thus $\tilde{v}^k$. Set $\tilde{u}^k = \mathcal{A}\tilde{v}^k$.

Step 2. **Backward substitution procedure (correction step)**. Generate the new iterate u^{k+1} via:

$$\mathcal{P}^T(u^{k+1} - u^k) = \alpha_k \mathcal{N}(\tilde{u}^k - u^k). \tag{5.15a}$$

where

$$\alpha_k = \gamma\alpha_k^B \ \text{ with } \ \alpha_k^B = \frac{\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2}{2\|\mathcal{N}(u^k - \tilde{u}^k)\|^2}. \tag{5.15b}$$

Remark 5.8. The method proposed in [17] also combines an ADMM procedure (more precisely, (1.3)) with a backward substitution procedure. But, the Gaussian back substitution procedure in [17] requires to compute the inverse matrices of $A_i^THA_i$ for $i = 2, \cdots, m$, which is more demanding than (5.15a) computationally.

To show how to choose the step size in the backward substitution procedure (5.15a), we first look at a backward substitution procedure whose step size α is undetermined and then investigate how to seek an appropriate value for α on contraction purpose.

Theorem 5.9. *Let $\tilde{w}^k$ be generated by the ADMM procedure* (3.1) *with given u^k and $\tilde{u}^k = \mathcal{A}\tilde{v}^k$. If the new iterate u^{k+1} is updated by (5.15a), then we have*

$$\|u^k - u^*\|_G^2 - \|u^{k+1} - u^*\|_G^2 \ge q^B(\alpha), \ \ \forall\, u^* \in \mathcal{U}^*, \tag{5.16}$$

where $G = \mathcal{P}\mathcal{P}^T$, $\mathcal{P}$ is defined in (2.10) and

$$q^B(\alpha) = \alpha\big(\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2\big) - \alpha^2\|\mathcal{N}(u^k - \tilde{u}^k)\|^2. \tag{5.17}$$

Proof. Recall that (5.15a) is a special case of (4.4) by taking $G = \mathcal{P}\mathcal{P}^T$. It follows from (4.6) that

$$\begin{array}{l} \|u^k - u^*\|_G^2 - \|u^{k+1} - u^*\|_G^2 \\ \quad \geq \alpha\big(\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2\big) - \alpha^2\|\mathcal{P}^{-1}\mathcal{L}(u^k - \tilde{u}^k)\|^2, \ \forall u^* \in \mathcal{U}^*. \end{array} \tag{5.18}$$

Since $\mathcal{P}^{-1}\mathcal{L} = \mathcal{N}$ (see (2.11)), (5.16)-(5.17) is an immediate conclusion of (5.18). □

Accordingly, the inequality (5.16) suggests us to choose a value of α such that $q^B(\alpha)$ defined in (5.17) is maximized, i.e.,

$$\alpha_k^B = \frac{\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2}{2\|\mathcal{N}(u^k - \tilde{u}^k)\|^2}.$$

Note that we can also take $G = \mathcal{P}\mathcal{P}^T$ in (4.7) and obtain the choice of α_k^B as above. Similar as the forward substitution procedure (5.1a), we attach a relaxation factor $\gamma \in (0, 2)$ to α_k^B and thus choose the step size as

$$\alpha_k = \gamma\alpha_k^B$$

for the backward substitution procedure (5.15a).

Proposition 5.10. *For the $q^B(\alpha)$ defined in* (5.17)*, we have*

$$q^B(\alpha) \geq \alpha\|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2, \quad \forall\, \alpha \in (0, 1] \tag{5.19}$$

and

$$q^B(\alpha) \geq \alpha(1 - \alpha)\|\mathcal{N}(u^k - \tilde{u}^k)\|^2, \quad \forall\, \alpha \in (0, 1] \tag{5.20}$$

Proof. By using the structure of $\mathcal{N}$, u and $\mathcal{S}$ (see (2.9), (2.4) and (2.7)), we have

$$\|\mathcal{N}(u^k - \tilde{u}^k)\|^2 = \|u^k - \tilde{u}^k\|^2 - \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2. \tag{5.21}$$

According to the definition of $q^B(\alpha)$, for any $\alpha \in (0, 1]$, then we have

$$\begin{array}{rcl} q^B(\alpha) & = & \alpha\big\{(\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2) - \alpha\|\mathcal{N}(u^k - \tilde{u}^k)\|^2\big\} \\ \text{(use (5.21))} & \geq & \alpha\big\{(\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2) - \|\mathcal{N}(u^k - \tilde{u}^k)\|^2\big\} \\ & = & \alpha\|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2. \end{array}$$

By using

$$\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2 \geq \|\mathcal{N}(u^k - \tilde{u}^k)\|^2, \tag{5.22}$$

for any $\alpha \in (0, 1]$, we have

$$\begin{array}{rcl} q^B(\alpha) & = & \alpha\big\{(\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2) - \alpha\|\mathcal{N}(u^k - \tilde{u}^k)\|^2\big\} \\ & \geq & \alpha(1 - \alpha)\|\mathcal{N}(u^k - \tilde{u}^k)\|^2. \end{array}$$

The assertions are proved. □

Moreover, to further alleviate the computation of the substitution procedure, the proposed backward substitution procedure (5.15a) is eligible to simply take a constant

$$\alpha_k \equiv \mu \in (0, 1)$$

as the step size. This is in fact a byproduct of (5.21). From (5.21), we have

$$\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2 \geq \|\mathcal{N}(u^k - \tilde{u}^k)\|^2 \qquad \text{and} \qquad \alpha_k^B \geq \frac{1}{2}. \tag{5.23}$$

Therefore, we have $\gamma \cdot \alpha_k^B \geq \gamma \cdot \frac{1}{2} \in (0, 1)$. If we take a constant $\mu \in (0, 1)$ as the step size for the backward substitution procedure (5.15a), instead of of (5.15b), the convergence is still ensured.

The following corollary shows that the equality is achieved in (5.16) if u^* is replaced by the vector $\tilde{u}^k$ generated by the ADMM procedure (3.1). This result will be used in Section 6 for proving a worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity for Algorithm 2.

Corollary 5.11. *Let $\tilde{w}^k$ be generated by the ADMM procedure* (3.1) *with given u^k and $\tilde{u}^k = \mathcal{A}\tilde{v}^k$. If the new iterate u^{k+1} is updated by (5.15a) and $q^B(\alpha)$ be defined in (5.17), then we have*

$$\|u^k - \tilde{u}^k\|_G^2 - \|u^{k+1} - \tilde{u}^k\|_G^2 = q^B(\alpha). \tag{5.24}$$

Proof. Setting $u = \tilde{u}^k$ in (4.5) with $G = \mathcal{P}\mathcal{P}^T$ and $\mathcal{P}^{-1}\mathcal{L} = \mathcal{N}$, we get

$$\|u^k - \tilde{u}^k\|_G^2 - \|u^{k+1} - \tilde{u}^k\|_G^2 = 2\alpha(u^k - \tilde{u}^k)^T\mathcal{L}(u^k - \tilde{u}^k) - \alpha^2\|\mathcal{N}(u^k - \tilde{u}^k)\|^2. \tag{5.25}$$

Using (4.2) and the definition of $q^B(\alpha)$ in (5.17), the assertion (5.25) is obtained. □

Now, we can show that the sequence $\{u^k\}$ generated by the proposed Algorithm 2 is contractive with respect to the set $\mathcal{U}^*$ under the $(\mathcal{P}\mathcal{P}^T)$-norm. Based on this fact, the reason why γ should be restricted into the interval $(0, 2)$ is also clear.

Theorem 5.12. *Let the sequence $\{u^k\}$ be generated by the proposed Algorithm 2. Then we have*

$$\|u^{k+1} - u^*\|_G^2 \le \|u^k - u^*\|_G^2 - \frac{\gamma(2-\gamma)}{4}\|u^k - \tilde{u}^k\|^2, \quad \forall\, u^* \in \mathcal{U}^*. \tag{5.26}$$

Proof. Since $\alpha_k = \gamma\alpha_k^B$, it follows from (5.17) and $\alpha_k^B \ge \frac{1}{2}$ that

$$\begin{aligned} q^B(\alpha_k) &= \gamma\alpha_k^B\big(\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2\big) - \gamma^2\alpha_k^B(\alpha_k^B\|\mathcal{N}(u^k - \tilde{u}^k)\|^2) \\ &\overset{(5.15b)}{=} \gamma(1 - \frac{\gamma}{2})\alpha_k^B\big(\|u^k - \tilde{u}^k\|^2 + \|\mathcal{S}(u^k - \tilde{u}^k)\|^2\big) \\ &\ge \frac{\gamma(2-\gamma)}{4}\|u^k - \tilde{u}^k\|^2. \end{aligned}$$

The assertion follows from the above inequality and (5.16) directly. □

Remark 5.13. When $m = 2$ in (1.1), $\mathcal{P} = \mathcal{I}$, in addition, we have $\mathcal{L} = \mathcal{N}$, then (see (2.9) and (2.10)) and thus $q^F(\alpha) = q^B(\alpha)$ (see (5.4) and (5.17)). In Subsection 5.1 we have demonstrated that if the step-size $\alpha \equiv 1$, the forward substitution procedure becomes

$$u^{k+1} - u^k = \mathcal{L}(\tilde{u}^k - u^k),$$

and thus Algorithm 1 reduces to the standard ADMM scheme (1.2). Similarly, the proposed backward substitution procedure with step-size $\alpha \equiv 1$ reduces to

$$\mathcal{P}^T(u^{k+1} - u^k) = \mathcal{N}(\tilde{u}^k - u^k).$$

Because $\mathcal{P}$ is the identity matrix and $\mathcal{L} = \mathcal{N}$, Algorithm 2 also reduces to the standard ADMM scheme (1.2). In short, for the case $m = 2$ in (1.1), the proposed forward and backward substitutions are identical, and Algorithms 1 and 2 both reduce to the standard ADMM scheme (1.2). Our proposed algorithms are thus able to recover the standard ADMM scheme (1.2).

5.3 Convergence analysis

In the last subsections, we have shown that both the sequences generated by Algorithms 1 and 2 are contractive with respective to the set $\mathcal{U}^*$. This fact thus enables us to establish the global convergence for both algorithms simultaneously based on the analytic framework of contraction methods in [1]. Moreover, based on this analytic framework of contraction methods, the local linear convergence of Algorithms 1 and 2 can be derived immediately provided that certain error bounds (e.g., some analogous to those in [22]) are assumed to be satisfied. As we have mentioned, the input of the proposed algorithms at each iteration is u^k. Thus, in the following analysis we investigate the convergence for the sequence $\{u^k\}$.

In fact, the inequalities (5.8) and (5.26) which show the contraction of the sequences generated by Algorithms 1 and 2 can be unified as

$$\|u^{k+1} - u^*\|_G^2 \le \|u^k - u^*\|_G^2 - c \cdot \|u^k - \tilde{u}^k\|^2, \quad \forall\, u^* \in \mathcal{U}^*, \tag{5.27}$$

where $c > 0$ is certain constant. More precisely, $G = \mathcal{I}$ and $c = \frac{\gamma(2-\gamma)}{4\|\mathcal{L}^T\mathcal{L}\|}$ for Algorithm 1; and $G = \mathcal{P}\mathcal{P}^T$ and $c = \frac{\gamma(2-\gamma)}{4}$ for Algorithm 2.

Theorem 5.14. *[Global convergence] Let the sequence $\{u^k\}$ be generated by the proposed framework of ADMM with a substitution (either Algorithm 1 or Algorithm 2). Then, there exists $u^\infty \in \mathcal{U}^*$ such that*

$$\lim_{k\to\infty} u^k = u^\infty,$$

and the corresponding vector w^∞ is a solution point of VI$(\mathcal{W}, F, \theta)$.

Proof. First, for an arbitrarily fixed $u^* \in \mathcal{U}^*$, it follows from (5.27) that the sequence $\{u^k\}$ is bounded. Summing the inequality (5.27) over $k = 0, 1, \ldots$, we obtain

$$\sum_{k=0}^{\infty} c \cdot \|u^k - \tilde{u}^k\|^2 \le \|u^0 - u^*\|_G^2,$$

and thus

$$\lim_{k\to\infty} \|u^k - \tilde{u}^k\|^2 = 0. \tag{5.28}$$

Therefore, the sequence $\{\tilde{u}^k\}$ is also bounded. Let u^∞ be a cluster point of $\{\tilde{u}^k\}$ and w^∞ is the related vector to u^∞. Recall that all A_i's are assumed to be full column rank. By using (2.15), the assertion of Theorem 3.3 can be written as

$$\tilde{w}^k \in \mathcal{W}, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \ge (v - \tilde{v}^k)^T \mathcal{A}^T \mathcal{L}(u^k - \tilde{u}^k), \quad \forall\, w \in \mathcal{W}.$$

Since $\lim_{k\to\infty} \|u^k - \tilde{u}^k\|^2 = 0$, it follows that

$$\lim_{k\to\infty} \{\theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k)\} \ge 0, \quad \forall\, w \in \mathcal{W}. \tag{5.29}$$

Therefore, w^∞ is a solution point of VI$(\mathcal{W}, F, \theta)$. By using (5.27), we have

$$\|u^{k+1} - u^\infty\|_G^2 \le \|u^k - u^\infty\|_G^2 - c \cdot \|u^k - \tilde{u}^k\|^2,$$

and thus $\lim_{k\to\infty} u^k = u^\infty$. The theorem is proved. □

Now, we can show the local linear convergence of Algorithms 1 and 2 immediately under some error bound assumptions similar as those in [22].

Theorem 5.15. *[Local linear convergence rate] Let the sequence $\{u^k\}$ be generated by the proposed framework of ADMM with a substitution (either Algorithm 1 or Algorithm 2). Assume that there is a constant $\tau > 0$ such that*

$$\|u^k - u^*\| \le \tau \|u^k - \tilde{u}^k\|, \; u^* \in \mathcal{U}^*. \tag{5.30}$$

Then, $\{u^k\}$ converges to u^* on a linear rate.

Proof. Since $\{\|u^k - u^*\|_G\}$ is Fejér monotone with respect to $\mathcal{U}^*$, the sequence $\{u^k\}$ is bounded. Then, the local linear convergence rate of $\{u^k\}$ is an immediate assertion based on the assumption (5.30) and the fact (5.27). □

In the next two sections, our purpose is to show that when the error bound assumption in Theorem 5.15 is not assumed, we still can estimate iteration complexity for Algorithms 1 and 2. Thus, some worst-case estimate on the convergence rate of Algorithms 1 and 2 can be derived. We address this concern in both ergodic and nonergodic senses.

6 Worst-case iteration complexity in an ergodic sense

In this section, we show a worst-case iteration complexity for the proposed algorithms in an ergodic sense. More specifically, our goal is to show that Algorithms 1 and 2 need at most $\boldsymbol{O(1/\epsilon)}$ iterations to find $\tilde{w} \in \mathcal{W}$, an approximate solution of VI$(\mathcal{W}, F, \theta)$ with the accuracy ϵ in the sense that

$$\theta(\tilde{x}) - \theta(x) + (\tilde{w} - w)^T F(w) \le \epsilon, \quad \forall w \in \mathcal{W}. \tag{6.1}$$

Recall that it is reasonable to use (6.1) to measure the accuracy of $\tilde{w}$ to a solution point of VI$(\mathcal{W}, F, \theta)$, because of the characterization (2.3) in Theorem 2.1.

We first present a simple lemma which will be used in the coming analysis. Since the proof is trivial, we omit it.

Lemma 6.1. *Let $G \in \Re^{n\times n}$ be symmetric and positive semi-definite. Then, it holds that*

$$(a-b)^T G(c-d) = \frac{1}{2}\big(\|a-d\|_G^2 - \|a-c\|_G^2\big) + \frac{1}{2}\big(\|c-b\|_G^2 - \|d-b\|_G^2\big), \quad \forall\, a,b,c,d \in \Re^n. \tag{6.2}$$

Then, we delineate analyze the iteration complexity for Algorithms 1 and 2 individually in the following subsections.

6.1 $\boldsymbol{O(1/\epsilon)}$ iteration complexity for Algorithm 1 in an ergodic sense

Recall (6.1). In order to estimate the accuracy of $\tilde{w}$, in the following theorem we first find a upper bound of the term $\theta(\tilde{x}) - \theta(x) + (\tilde{w} - w)^T F(w)$ for the sequence generated by Algorithm 1. Note that the following result (6.3) is valid for any step size $\alpha > 0$ in (5.1a).

Theorem 6.2. *Let the sequence $\{u^k\}$ be generated by (5.2), i.e., Algorithm 1 with a unspecified step size $\alpha > 0$, and $q^F(\alpha)$ be defined in* (5.4)*. Then, we have*

$$\theta(\tilde{x}^k) - \theta(x) + (\tilde{w}^k - w)^T F(w) \le \frac{1}{2\alpha}\big(\|u-u^k\|^2 - \|u-u^{k+1}\|^2 - q^F(\alpha)\big), \ \forall w \in \mathcal{W}. \tag{6.3}$$

Proof. The assertions (6.3) can be obtained based on the following facts.

(1). Using Theorem 3.3 and the fact $(w-\tilde{w}^k)^T F(w) = (w-\tilde{w}^k)^T F(\tilde{w}^k)$, we have

$$\alpha\big\{\big(\theta(x) - \theta(\tilde{x}^k)\big) + (w-\tilde{w}^k)^T F(w)\big\} \ge \alpha(u-\tilde{u}^k)^T \mathcal{L}(u^k - \tilde{u}^k), \quad \forall w \in \mathcal{W}. \tag{6.4a}$$

(2). Since $\alpha\mathcal{L}(u^k - \tilde{u}^k) = (u^k - u^{k+1})$ for any $\alpha > 0$ in (5.1a), for the right hand side of (6.4a), we have

$$\alpha(u-\tilde{u}^k)^T \mathcal{L}(u^k - \tilde{u}^k) = (u-\tilde{u}^k)^T (u^k - u^{k+1}). \tag{6.4b}$$

(3). Using Lemma 6.1, we have

$$\begin{aligned}&\big(u-\tilde{u}^k\big)^T\big(u^k - u^{k+1}\big)\\ &= \frac{1}{2}\big(\|u-u^{k+1}\|^2 - \|u-u^k\|^2\big) + \frac{1}{2}\big(\|u^k-\tilde{u}^k\|^2 - \|u^{k+1}-\tilde{u}^k\|^2\big).\end{aligned} \tag{6.4c}$$

(4). Recall Corollary 5.3. We have

$$\frac{1}{2}\big(\|u^k-\tilde{u}^k\|^2 - \|u^{k+1}-\tilde{u}^k\|^2\big) = \frac{1}{2}q^F(\alpha). \tag{6.4d}$$

Combining these assertions, we obtain the assertion (6.3). □

Recall that for the proposed Algorithm 1, the step size α is taken as $\alpha = \gamma\alpha_k^F$ with $\gamma \in (0,2)$ and α_k^F is given by (5.1b). Moreover, (5.1b) and (5.4) imply that $q^F(\gamma\alpha_k^F) \geq 0$. Thus, we have the following corollary of Theorem 6.2.

Corollary 6.3. *Let the sequence $\{u^k\}$ be generated by the proposed Algorithm 1. Then, we have*

$$\theta(\tilde{x}^k) - \theta(x) + (\tilde{w}^k - w)^T F(w) \leq \frac{1}{2\gamma\alpha_k^F}\big(\|u-u^k\|^2 - \|u-u^{k+1}\|^2\big), \ \forall w \in \mathcal{W}. \tag{6.5}$$

Now, we are ready to show a worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity for Algorithm 1 in an ergodic sense.

Theorem 6.4. *Let $\{u^k\}$ be the sequence generated by Algorithm 1. For any integer number $t > 0$, let*

$$\tilde{w}_t := \frac{1}{\Upsilon_t^F}\sum_{k=0}^{t} \alpha_k^F \tilde{w}^k \quad \textit{with} \ \ \Upsilon_t^F = \sum_{k=0}^{t}\alpha_k^F. \tag{6.6}$$

Then, we have $\tilde{w}_t \in \mathcal{W}$ and

$$\theta(\tilde{x}_t) - \theta(x) + (\tilde{w}_t - w)^T F(w) \leq \frac{\|\mathcal{L}^T\mathcal{L}\|}{\gamma(t+1)}\|u-u^0\|^2, \quad \forall w \in \mathcal{W}, \tag{6.7}$$

i.e., $\tilde{w}_t$ is a solution point of VI$(\mathcal{W}, F, \theta)$ with the accuracy of $\boldsymbol{O(1/t)}$.

Proof. Summing the inequality (6.5) over $k = 0, 1, \ldots, t$, we obtain

$$\Big(\Upsilon_t^F\theta(x) - \sum_{k=0}^{t}\alpha_k^F\theta(\tilde{x}^k)\Big) + \Big(\Upsilon_t^F w - \sum_{k=0}^{t}\alpha_k^F\tilde{w}^k\Big)^T F(w) + \frac{1}{2\gamma}\|u-u^0\|^2 \geq 0, \quad \forall w \in \mathcal{W}.$$

It follows that

$$\Big(\frac{1}{\Upsilon_t^F}\sum_{k=0}^{t}\alpha_k^F\theta(\tilde{x}^k) - \theta(x)\Big) + \Big(\frac{1}{\Upsilon_t^F}\sum_{k=0}^{t}\alpha_k^F\tilde{w}^k - w\Big)^T F(w) \leq \frac{1}{2\gamma\Upsilon_t^F}\|u-u^0\|^2, \quad \forall w \in \mathcal{W}. \tag{6.8}$$

Since $\tilde{x}_t = \frac{1}{\Upsilon_t^F}\sum_{k=0}^{t}\alpha_k^F\tilde{x}^t$ is a convex combination of the vectors $(\tilde{x}^0, \tilde{x}^1, \ldots, \tilde{x}^k)$ and $\theta(x)$ is convex, we have

$$\theta(\tilde{x}_t) \leq \frac{1}{\Upsilon_t^F}\sum_{k=0}^{t}\alpha_k^F\theta(\tilde{x}^k).$$

Substituting it in (6.8), we have

$$\theta(\tilde{x}_t) - \theta(x) + (\tilde{w}_t - w)^T F(w) \leq \frac{1}{2\gamma\Upsilon_t^F}\|u-u^0\|^2, \quad \forall w \in \mathcal{W}, \tag{6.9}$$

Recall that we have shown that $\alpha_k^F \geq \|\mathcal{L}^T\mathcal{L}\|/2$ in Theorem 5.5. Then, it follows from (6.6) that

$$\Upsilon_t^F \geq \frac{(t+1)}{2\|\mathcal{L}^T\mathcal{L}\|}.$$

Finally, substituting it in (6.9), we obtain the assertion (6.7). □

The assertion (6.7) in Theorem 6.4 thus clearly shows a worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity for Algorithm 1 in an ergodic sense.

Remark 6.5. We have shown that Algorithm 1 reduces to the original ADMM (1.2) when $m = 2$ and $\alpha_k \equiv 1$. For this special case, we have $q^F(1) \geq 0$ (see (5.10)). Thus, Corollary 6.3 and Theorem 6.4 are valid. In addition, since $\alpha_k \equiv 1$, it follows from (6.9) that $\Upsilon_t = t + 1$ and thus

$$\theta(\tilde{x}_t) - \theta(x) + (\tilde{w}_t - w)^T F(w) \leq \frac{1}{2(t+1)} \|u - u^0\|^2, \quad \forall w \in \mathcal{W},$$

which indicates a worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity for the original ADMM (1.2). Thus, the $\boldsymbol{O(1/\epsilon)}$ iteration complexity of the original ADMM (1.2) proved in [19] can be regarded as a special case of Theorem 6.4 .

Remark 6.6. It is easy to see that $\alpha \in (0, \alpha_*^F]$ still ensures the positivity of $q^F(\alpha) \geq 0$ (see (5.12)). In this case, based on (6.3), the inequality (6.5) becomes

$$\theta(\tilde{x}^k) - \theta(x) + (\tilde{w}^k - w)^T F(w) \leq \frac{1}{2\alpha}\big(\|u - u^k\|^2 - \|u - u^{k+1}\|^2\big), \ \forall w \in \mathcal{W}. \tag{6.10}$$

Then, for $\tilde{w}_t := \frac{1}{t+1}\sum_{k=0}^{t} \tilde{w}^k \in \mathcal{W}$, we have

$$\theta(\tilde{x}_t) - \theta(x) + (\tilde{w}_t - w)^T F(w) \leq \frac{1}{2\alpha(t+1)} \|u - u^0\|^2, \quad \forall w \in \mathcal{W}. \tag{6.11}$$

In other words, the proved worst-case $O(1/\epsilon)$ iteration complexity holds for Algorithm 1 with a constant step size in $\alpha \in (0, \alpha_*^F]$. Although it saves computation by taking a constant step size, we do have found some applications where the step size chosen by (5.1b) is capable of accelerating convergence of Algorithm 1. To implement Algorithm 1, our recommendation is to use a constant step size if the computation of (5.1b) is inexpensive, or compute it via (5.1b) otherwise.

6.2 $\boldsymbol{O(1/\epsilon)}$ iteration complexity for Algorithm 2 in an ergodic sense

The framework of establishing a worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity for Algorithm 2 is analogous to that for Algorithm 1. Again, we first find a upper bound of the term $\theta(\tilde{x}) - \theta(x) + (\tilde{w} - w)^T F(w)$ in order to estimate the iteration complexity of Algorithm 2.

Theorem 6.7. *Let the sequence $\{u^k\}$ be generated by (5.15a), i.e., Algorithm 2 with a unspecified step size $\alpha > 0$. Let $q^B(\alpha)$ be defined in* (5.17) *and $G = \mathcal{P}\mathcal{P}^T$ where $\mathcal{L}$ be given by (2.10). Then, we have*

$$\theta(\tilde{x}^k) - \theta(x) + (\tilde{w}^k - w)^T F(w) \leq \frac{1}{2\alpha}\big(\|u - u^k\|_G^2 - \|u - u^{k+1}\|_G^2 - q^B(\alpha)\big), \ \forall w \in \mathcal{W}. \tag{6.12}$$

Proof. The assertion (6.12) is based on the following facts.

(1). Using Theorem 3.3 and the monotonicity of F, we have

$$\alpha\big\{\big(\theta(x) - \theta(\tilde{x}^k)\big) + (w - \tilde{w}^k)^T F(w)\big\} \geq \alpha(u - \tilde{u}^k)^T \mathcal{L}(u^k - \tilde{u}^k), \quad \forall w \in \mathcal{W}. \tag{6.13a}$$

(2). Using $\alpha\mathcal{L}(u^k - \tilde{u}^k) = \mathcal{P}\mathcal{P}^T(u^k - u^{k+1})$ (see (5.15a)) and $G = \mathcal{P}\mathcal{P}^T$, we have

$$\alpha(u - \tilde{u}^k)^T \mathcal{L}(u^k - \tilde{u}^k) = (u - \tilde{u}^k)^T G(u^k - u^{k+1}). \tag{6.13b}$$

(3). Using Lemma 6.1, we have

$$\begin{aligned}&\big(u - \tilde{u}^k\big)^T G(u^k - u^{k+1}) \\ &= \frac{1}{2}\big(\|u - u^{k+1}\|_G^2 - \|u - u^k\|_G^2\big) + \frac{1}{2}\big(\|u^k - \tilde{u}^k\|_G^2 - \|u^{k+1} - \tilde{u}^k\|_G^2\big).\end{aligned} \tag{6.13c}$$

(4). Recall Corollary 5.11. We have

$$\frac{1}{2}\big(\|u^k-\tilde{u}^k\|_G^2-\|u^{k+1}-\tilde{u}^k\|_G^2\big)=\frac{1}{2}q^B(\alpha). \tag{6.13d}$$

Combining the assertions in (6.13) together, the assertion (6.12) is proved. □

Recall that the step size $\alpha_k=\gamma\alpha_k^B$ is taken as $\gamma\in(0,2)$ for the proposed Algorithm 2. Moreover, (5.17) implies that $q^B(\alpha)\ge 0$. Thus, we have the following corollary of Theorem 6.7.

Corollary 6.8. *Let the sequence $\{u^k\}$ be generated by the proposed Algorithm 2. Then, we have*

$$\theta(\tilde{x}^k)-\theta(x)+(\tilde{w}^k-w)^TF(w)\le\frac{1}{2\gamma\alpha_k^B}\big(\|u-u^k\|_G^2-\|u-u^{k+1}\|_G^2\big),\ \forall w\in\mathcal{W}. \tag{6.14}$$

Now, we are ready to show a worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity for Algorithm 2 in an ergodic sense.

Theorem 6.9. *Let $\{u^k\}$ be the sequence generated by Algorithm 2. For any integer number $t>0$, let*

$$\tilde{w}_t:=\frac{1}{\Upsilon_t^B}\sum_{k=0}^{t}\alpha_k^B\tilde{w}^k\quad \textit{with}\ \ \Upsilon_t^B=\sum_{k=0}^{t}\alpha_k^B. \tag{6.15}$$

Then, we have $\tilde{w}_t\in\mathcal{W}$ and

$$\theta(\tilde{x}_t)-\theta(x)+(\tilde{w}_t-w)^TF(w)\le\frac{1}{\gamma(t+1)}\|u-u^0\|_G^2,\quad\forall w\in\mathcal{W}, \tag{6.16}$$

i.e., $\tilde{w}_t$ is a solution point of VI$(\mathcal{W},F,\theta)$ with the accuracy of $\boldsymbol{O(1/t)}$.

Proof. Summing the inequality (6.14) over $k=0,1,\ldots,t$, we obtain

$$\Big(\Upsilon_t^B\theta(x)-\sum_{k=0}^{t}\alpha_k^B\theta(\tilde{x}^k)\Big)+\Big(\Upsilon_t^Bw-\sum_{k=0}^{t}\alpha_k^B\tilde{w}^k\Big)^TF(w)+\frac{1}{2\gamma}\|u-u^0\|_G^2\ge 0,\quad\forall w\in\mathcal{W}.$$

It follows that

$$\Big(\frac{1}{\Upsilon_t^B}\sum_{k=0}^{t}\alpha_k^B\theta(\tilde{x}^k)-\theta(x)\Big)+\Big(\frac{1}{\Upsilon_t^B}\sum_{k=0}^{t}\alpha_k^B\tilde{w}^k-w\Big)^TF(w)\le\frac{1}{2\gamma\Upsilon_t^B}\|u-u^0\|_G^2,\quad\forall w\in\mathcal{W}. \tag{6.17}$$

Since $\theta(x)$ is convex, we have

$$\theta(\tilde{x}_t)\le\frac{1}{\Upsilon_t^B}\sum_{k=0}^{t}\alpha_k^B\theta(\tilde{x}^k).$$

Recall that we have shown that $\alpha_k^B\ge 1/2$ in Theorem 5.12. Then, it follows from (6.15) that

$$\Upsilon_t^B\ge\frac{(t+1)}{2}.$$

Substituting it in (6.17), the assertion (6.16) follows directly. □

The assertion (6.16) in Theorem 6.9 thus clearly shows a worst-case $O(1/\epsilon)$ iteration complexity for Algorithm 2 in an ergodic sense.

Remark 6.10. It is easy to see that $\alpha = 1$ still ensures the positivity of $q^B(\alpha) \geq 0$ (see (5.19)). In this case, based on (6.12), the inequality (6.14) becomes

$$\theta(\tilde{x}^k) - \theta(x) + (\tilde{w}^k - w)^T F(w) \leq \frac{1}{2}\big(\|u - u^k\|_G^2 - \|u - u^{k+1}\|_G^2\big), \ \forall w \in \mathcal{W}. \tag{6.18}$$

Then, for $\tilde{w}_t := \frac{1}{t+1}\sum_{k=0}^{t} \tilde{w}^k \in \mathcal{W}$, we have

$$\theta(\tilde{x}_t) - \theta(x) + (\tilde{w}_t - w)^T F(w) \leq \frac{1}{2(t+1)}\|u - u^0\|_G^2, \quad \forall w \in \mathcal{W}. \tag{6.19}$$

In other words, the proved worst-case $O(1/\epsilon)$ iteration complexity holds for Algorithm 2 with a constant step size in $\alpha \in (0, 1]$. Although it saves computation by taking a constant step size in $(0, 1]$, we also have observed some applications where the step size chosen by (5.15b) is capable of accelerating convergence of Algorithm 2. Thus, similarly as Algorithm 1, to implement Algorithm 2, our recommendation is to use a constant step size if the computation of (5.15b) is inexpensive or compute it via (5.15b) otherwise.

7 $O(1/\epsilon)$ worst-case iteration complexity in a nonergodic sense

In Section 6, a worst-case $O(1/\epsilon)$ iteration complexity is established for Algorithms 1 and 2 in an ergodic sense: $\tilde{w}_t$ is a convex combination of all the previous vectors $\{\tilde{w}^0, \tilde{w}^1, \ldots, \tilde{w}^t\}$. One may ask if we can establish the same iteration complexity in a nonergodic sense, i.e, directly for the sequence $\{u^k\}$ generated by the proposed algorithms. This section answers this question affirmatively.

Our basis of the analysis in this section is the fact that the assertion (3.2) in Theorem 3.3 can be rewritten as

$$\tilde{w}^k \in \mathcal{W}, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) + (u - \tilde{u}^k)^T \mathcal{L}(\tilde{u}^k - u^k) \geq 0, \quad \forall\, w \in \mathcal{W}.$$

Since $\mathcal{L}$ is a nonsingular matrix, the above inequality means that $\tilde{w}^k$ is a solution point in $\mathcal{W}^*$ if $\|u^k - \tilde{u}^k\|^2 = 0$. Thus, we can view $\|u^k - \tilde{u}^k\|^2$ as a residual or an error bound to measure the accuracy of $\tilde{w}^k$ to a solution point in $\mathcal{W}^*$. In this section, we will show that after t iterations of the proposed Algorithms 1 and 2, we can ensure that

$$\min_{0\leq k\leq t}\{\|u^k - \tilde{u}^k\|^2\} \leq \epsilon \qquad \text{or} \qquad \|u^t - \tilde{u}^t\|^2 \leq \epsilon,$$

where $\epsilon = O(1/t)$. Thus, a worst-case $O(1/\epsilon)$ iteration complexity is established in a nonergodic sense for Algorithms 1 and 2.

In Section 6, we have shown that the step size for either the forward or backward substitution step (i.e., (5.1a) or (5.15a)) can be simply taken as a constant in certain interval. This could be useful to further accelerate Algorithms 1 and 2 for the cases where the computation of step size chosen by (5.1b) or (5.15b) is expensive. To show the $O(1/\epsilon)$ worst-case iteration complexity in a nonergodic sense for Algorithms 1 and 2, we consider the cases with a constant or chosen step size, respectively.

7.1 The case with a chosen step size

We first establish a worst-case $O(1/\epsilon)$ iteration complexity in a nonergodic sense for Algorithms 1 and 2 when their substitution step sizes are taken as (5.1b) and (5.15b)), respectively.

For both algorithms, from (5.27) we have

$$c\sum_{k=0}^{\infty}\|u^k-\tilde{u}^k\|^2 \le \|u^0-u^*\|_G^2, \quad \forall\, u^*\in\mathcal{U}^*,$$

where $c=\frac{\gamma(2-\gamma)}{4\|\mathcal{L}^T\mathcal{L}\|}$ and $G=\mathcal{I}$ for Algorithm 1; and $c=\frac{\gamma(2-\gamma)}{4}$ and $G=\mathcal{P}\mathcal{P}^T$ for Algorithm 2. Thus, for any integer $t>0$, we obtain

$$c\sum_{k=0}^{t}\|u^k-\tilde{u}^k\|^2 \le \|u^0-u^*\|_G^2, \quad \forall\, u^*\in\mathcal{U}^*,$$

and consequently, it follows that

$$\min_{0\le k\le t}\{\|u^k-\tilde{u}^k\|^2\} \le \frac{1}{c(t+1)}\|u^0-u^*\|_G^2, \quad \forall\, u^*\in\mathcal{U}^*. \tag{7.1}$$

Recall $\mathcal{W}^*$ is convex and closed under our assumptions (see Theorem 2.3.5 in [8]). Let

$$d := \inf\{\|u^0-u^*\|_G^2 \,|\, u^*\in\mathcal{U}^*\}.$$

For any given $\epsilon>0$, the inequality (7.1) indicates that Algorithms 1 and 2 require at most

$$\lfloor d/c\epsilon \rfloor$$

iterations to fulfill the requirement $\|u^k-\tilde{u}^k\|^2\le\epsilon$. Thus, a worst-case $O(1/\epsilon)$ worst-case iteration complexity is established for Algorithms 1 and 2 in a nonergodic sense.

7.2 The case with a constant step size

Then, we establish a worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity in a nonergodic sense for Algorithms 1 and 2 when their substitution step sizes are taken as certain constants. We first show a lemma where an inequality regarding the output of the ADMM procedure (3.1) is proved.

Lemma 7.1. *Let $\tilde{w}^k$ be generated by the ADMM procedure* (3.1) *with given u^k and $\tilde{u}^k=\mathcal{A}\tilde{v}^k$. Then we have*

$$(u^k-u^{k+1})^T\mathcal{L}\big((u^k-\tilde{u}^k)-(u^{k+1}-\tilde{u}^{k+1})\big) \ge \frac{1}{2}\|(u^k-\tilde{u}^k)-(u^{k+1}-\tilde{u}^{k+1})\|_{(\mathcal{L}^T+\mathcal{L})}^2. \tag{7.2}$$

Proof. First, it follows from (3.2) that

$$\tilde{w}^k\in\Omega, \quad \theta(x)-\theta(\tilde{x}^k)+(w-\tilde{w}^k)^TF(\tilde{w}^k) \ge (u-\tilde{u}^k)\mathcal{L}(u^k-\tilde{u}^k), \quad \forall\, w\in\Omega. \tag{7.3}$$

This inequality is also true for $k:=k+1$, and thus we have

$$\tilde{w}^{k+1}\in\Omega, \quad \theta(x)-\theta(\tilde{x}^{k+1})+(w-\tilde{w}^{k+1})^TF(\tilde{w}^{k+1}) \ge (u-\tilde{u}^{k+1})\mathcal{L}(u^{k+1}-\tilde{u}^{k+1}), \;\; \forall\, w\in\Omega. \tag{7.4}$$

Setting $w=\tilde{w}^{k+1}$ and $w=\tilde{w}^k$ in (7.3) and (7.4), respectively, and then adding these two resulting inequalities, we obtain

$$(\tilde{u}^k-\tilde{u}^{k+1})^T\mathcal{L}\big((u^k-\tilde{u}^k)-(u^{k+1}-\tilde{u}^{k+1})\big) \ge (\tilde{w}^k-\tilde{w}^{k+1})^T(F(\tilde{w}^k)-F(\tilde{w}^{k+1})).$$

Using the monotonicity of F, we have

$$(\tilde{u}^k-\tilde{u}^{k+1})^T\mathcal{L}\big((u^k-\tilde{u}^k)-(u^{k+1}-\tilde{u}^{k+1})\big) \ge 0. \tag{7.5}$$

Adding the term

$$\big((u^k-\tilde{u}^k)-(u^{k+1}-\tilde{u}^{k+1})\big)^T\mathcal{L}\big((u^k-\tilde{u}^k)-(u^{k+1}-\tilde{u}^{k+1})\big)$$

to both sides of (7.5) and by a simple manipulation, we get (7.2) and the lemma is proved. □

Then we analyze the worst-case iteration complexity in a nonergodic sense for Algorithms 1 and 2, respectively, when their substitution step sizes are taken as certain constants.

7.2.1 Algorithm 1 with a constant step size

Recall that the substitution step size of Algorithm 1 can be taken as a constant in $\alpha \in (0, \alpha_*^F)$ where α_*^F is defined in (5.11). We first show that the sequence $\{\|\mathcal{L}(u^k - \tilde{u}^k)\|^2\}$ is monotonically non-increasing and derive the worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity.

Lemma 7.2. *Let $\{u^k\}$ be generated by Algorithm 1 with $\alpha_k \equiv \alpha \in (0, \alpha_*^F)$. Then we have*

$$\|\mathcal{L}(u^{k+1} - \tilde{u}^{k+1})\|^2 \le \|\mathcal{L}(u^k - \tilde{u}^k)\|^2, \quad \forall\, k \ge 0. \tag{7.6}$$

Proof. By using (5.15a), we have

$$u^k - u^{k+1} = \alpha \mathcal{L}(u^k - \tilde{u}^k),$$

and substituting it in (7.2), it follows that

$$(u^k - \tilde{u}^k)^T \mathcal{L}^T \mathcal{L}\big((u^k - \tilde{u}^k) - (u^{k+1} - \tilde{u}^{k+1})\big) \ge \frac{1}{2\alpha}\|(u^k - \tilde{u}^k) - (u^{k+1} - \tilde{u}^{k+1})\|_{(\mathcal{L}^\mathcal{T} + \mathcal{L})}. \tag{7.7}$$

Setting $a = \mathcal{L}(u^k - \tilde{u}^k)$ and $b = \mathcal{L}(u^{k+1} - \tilde{u}^{k+1})$ in the identity

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

and using the inequality (7.7) and the relation (2.14), we obtain

$$\begin{aligned}
&\|\mathcal{L}(u^k - \tilde{u}^k)\|^2 - \|\mathcal{L}(u^{k+1} - \tilde{u}^{k+1})\|^2 \\
&\ge \frac{1}{\alpha}\|(u^k - \tilde{u}^k) - (u^{k+1} - \tilde{u}^{k+1})\|_{(\mathcal{L}^\mathcal{T} + \mathcal{L})} - \|\mathcal{L}(u^k - \tilde{u}^k) - \mathcal{L}(u^{k+1} - \tilde{u}^{k+1})\|^2 \\
&= \frac{1}{\alpha}\|(u^k - \tilde{u}^k) - (u^{k+1} - \tilde{u}^{k+1})\|^2_{((\mathcal{I} + \mathcal{S}^T\mathcal{S}) - \alpha\mathcal{L}^T\mathcal{L})}.
\end{aligned} \tag{7.8}$$

Because $\alpha \in (0, \alpha_*^F)$ and thus $(\mathcal{I} + \mathcal{S}^T\mathcal{S} - \alpha\mathcal{L}^T\mathcal{L}) \succ 0$ (see (5.11)), the right-hand side of (7.8) is non-negative and the lemma is proved. □

Now, we are ready to estimate a worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity for Algorithm 1 with a constant substitution step size in a nonergodic sense.

Theorem 7.3. *Let $\{u^t\}$ be the sequence generated by the proposed Algorithm 1 with $\alpha_k \equiv \alpha \in (0, \alpha_*^F)$. Then, we have*

$$\|\mathcal{L}(u^t - \tilde{u}^t)\|^2 \le \frac{1}{\alpha(\alpha_*^F - \alpha)(t+1)}\|u^0 - u^*\|^2, \tag{7.9}$$

$\mathcal{L}$ is defined in (2.10).

Proof. First, it follows from Theorem 5.7 (see (5.13)) that

$$\alpha(\alpha_*^F - \alpha)\sum_{k=0}^{\infty}\|\mathcal{L}(u^k - \tilde{u}^k)\|^2 \le \|u^0 - u^*\|^2, \quad \forall u^* \in \mathcal{U}^*. \tag{7.10}$$

Since Lemma 7.2 shows that the sequence $\{\|\mathcal{L}(u^k - \tilde{u}^k)\|^2\}$ is monotonically non-increasing, we have

$$(t+1)\|\mathcal{L}(u^t - \tilde{u}^t)\|^2 \le \sum_{k=0}^{t}\|\mathcal{L}(u^k - \tilde{u}^k)\|^2, \tag{7.11}$$

which implies the assertion (7.9) immediately. □

Recall $\mathcal{W}^*$ is convex and closed under our assumptions (see Theorem 2.3.5 in [8]). Let

$$d_1 := \inf\{\|u^0 - u^*\|^2 \mid u^* \in \mathcal{U}^*\}.$$

For any given $\epsilon > 0$, Theorem 7.3 indicates that Algorithm 1 with $\alpha_k \equiv \alpha \in (0, \alpha_*^F)$ requires at most $\lfloor d_1/\alpha(\alpha_*^F - \alpha)\epsilon \rfloor$ iterations to ensure that $\|\mathcal{L}(u^k - \tilde{u}^k)\|^2 \le \epsilon$. A worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity is thus established for Algorithm 1 with $\alpha_k \equiv \alpha \in (0, \alpha_*^F)$ in a nonergodic sense.

7.2.2 Algorithm 2 with a constant step size

Recall that the the substitution step size of Algorithm 2 can be taken as a constant in $\alpha \in (0,1)$. First, it follows from (5.16) and (5.20) that

$$\|u^{k+1} - u^*\|_G^2 \le \|u^k - u^*\|_G^2 - \alpha(1-\alpha)\|\mathcal{N}(u^k - \tilde{u}^k)\|^2, \ \ \forall\, u^* \in \mathcal{U}^*. \tag{7.12}$$

Then, we need to show that the sequence $\{\|\mathcal{N}(u^k - \tilde{u}^k)\|^2\}$ is monotonically non-increasing before derive the worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity.

Lemma 7.4. *Let $\{u^k\}$ be generated by Algorithm 2 with $\alpha_k \equiv \mu \in (0,1)$. Then we have*

$$\|\mathcal{N}(u^{k+1} - \tilde{u}^{k+1})\|^2 \le \|\mathcal{N}(u^k - \tilde{u}^k)\|^2, \quad \forall\, k \ge 0. \tag{7.13}$$

Proof. By using (5.15a), we have

$$u^k - u^{k+1} = \alpha \mathcal{P}^{-T}\mathcal{N}(u^k - \tilde{u}^k),$$

and substituting it in (7.2) and use $\mathcal{N} = \mathcal{P}^{-1}\mathcal{L}$, it follows that

$$(u^k - \tilde{u}^k)^T\mathcal{N}^T\mathcal{N}\big((u^k - \tilde{u}^k) - (u^{k+1} - \tilde{u}^{k+1})\big) \ge \frac{1}{2\alpha}\|(u^k - \tilde{u}^k) - (u^{k+1} - \tilde{u}^{k+1})\|_{(\mathcal{L}^T+\mathcal{L})}. \tag{7.14}$$

Setting $a = \mathcal{N}(u^k - \tilde{u}^k)$ and $b = \mathcal{N}(u^{k+1} - \tilde{u}^{k+1})$ in the identity

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

and using the inequality (7.14), we obtain

$$\begin{aligned} &\|\mathcal{N}(u^k - \tilde{u}^k)\|^2 - \|\mathcal{N}(u^{k+1} - \tilde{u}^{k+1})\|^2 \\ \ge\ & \frac{1}{\alpha}\|(u^k - \tilde{u}^k) - (u^{k+1} - \tilde{u}^{k+1})\|_{(\mathcal{L}^T+\mathcal{L})} - \|\mathcal{N}(u^k - \tilde{u}^k) - \mathcal{N}(u^{k+1} - \tilde{u}^{k+1})\|^2 \\ \ge\ & \|(u^k - \tilde{u}^k) - (u^{k+1} - \tilde{u}^{k+1})\|_{(\mathcal{L}^T+\mathcal{L})} - \|(u^k - \tilde{u}^k) - (u^{k+1} - \tilde{u}^{k+1})\|^2_{(\mathcal{N}^T\mathcal{N})}. \end{aligned} \tag{7.15}$$

The last inequality is due to $\alpha \in (0,1]$. Because (see (2.9) and (2.10) for the definition of the matrices $\mathcal{L}$ and $\mathcal{N}$)

$$\mathcal{L}^T + \mathcal{L} = \begin{pmatrix} 2I_l & I_l & \cdots & I_l & -I_l \\ I_l & 2I_l & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & I_l & -I_l \\ I_l & \cdots & I_l & 2I_l & -I_l \\ -I_l & \cdots & -I_l & -I_l & 2I_l \end{pmatrix} \quad \text{and} \quad \mathcal{N}^T\mathcal{N} = \begin{pmatrix} 2I_l & I_l & \cdots & I_l & -I_l \\ I_l & 2I_l & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & I_l & -I_l \\ I_l & \cdots & I_l & 2I_l & -I_l \\ -I_l & \cdots & -I_l & -I_l & I_l \end{pmatrix},$$

the right-hand side of (7.15) is non-negative and the lemma is proved. □

Now, we are ready to estimate a worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity for Algorithm 2 with a constant substitution step size in a nonergodic sense.

Theorem 7.5. *Let $\{u^t\}$ be the sequence generated by the proposed Algorithm 2 with $\alpha_k \equiv \mu \in (0,1)$. Then, we have*

$$\|\mathcal{N}(u^t - \tilde{u}^t)\|^2 \le \frac{1}{\alpha(1-\alpha)(t+1)}\|u^0 - u^*\|_G^2. \tag{7.16}$$

where $G = \mathcal{P}\mathcal{P}^T$ and $\mathcal{P}$ is defined in (2.10).

Proof. First, it follows from (7.12) that

$$\alpha(1-\alpha)\sum_{k=0}^{\infty}\|\mathcal{N}(u^k-\tilde{u}^k)\|^2 \le \|u^0-u^*\|_G^2, \quad \forall u^*\in\mathcal{U}^*. \tag{7.17}$$

Since Lemma 7.4 shows that the sequence $\{\|\mathcal{N}(u^k-\tilde{u}^k)\|^2\}$ is monotonically non-increasing, we have

$$(t+1)\|\mathcal{N}(u^t-\tilde{u}^t)\|^2 \le \sum_{k=0}^{t}\|\mathcal{N}(u^k-\tilde{u}^k)\|^2, \tag{7.18}$$

which implies the assertion (7.16) immediately. □

Recall $\mathcal{W}^*$ is convex and closed under our assumptions (see Theorem 2.3.5 in [8]). Let

$$d_2 := \inf\{\|u^0-u^*\|_G^2 \,|\, u^*\in\mathcal{U}^*\}.$$

For any given $\epsilon>0$, Theorem 7.5 indicates that Algorithm 2 with $\alpha_k\equiv\mu\in(0,1)$ requires at most $\lfloor d_2/\mu(1-\mu)\epsilon\rfloor$ iterations to ensure that $\|\mathcal{N}(u^k-\tilde{u}^k)\|^2\le\epsilon$. A worst-case $\boldsymbol{O(1/\epsilon)}$ iteration complexity is thus established for Algorithm 2 with $\alpha_k\equiv\mu\in(0,1)$ in a nonergodic sense.

8 Conclusions

In this paper, we further study the combination of the Douglas-Rachford alternating direction method of multipliers (ADMM) with a substitution procedure for solving a convex minimization model with a general separable structure in the objective function. The proposed forward and backward substitution procedures are computationally inexpensive and thus numerically implementable. The resulting algorithms are thus competitive to the direct extension of ADMM (1.3) whose convergence remains open under mild assumptions. The proposed algorithms share the feature that they both reduce to the original ADMM for the case of (1.1) where $m=2$. We prove the global convergence of the proposed algorithms under the analytic framework of contraction methods, and derive their local linear convergence rate when some error bound assumptions are further assumed. For general settings without additional assumptions, we also estimate the worst-case iteration complexity for both of the algorithms in both ergodic and nonergodic senses.

Since there are enough impressive illustrations on the numerical efficiency of ADMM in the literature, including the efficiency of ADMM with a Gaussian back substitution procedure in [17, 25], in this paper we focus only on theoretical perspectives of the combination of ADMM with a substitution procedure and include no numerical results for succinctness purpose. We do have some numerical results which are available upon request, but we choose not to include them due to page limitation and the theoretical emphasis of this paper.

References

[1] E. Blum and W. Oettli, Mathematische Optimierung, Econometrics and Operations Research XX, Springer Verlag, 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*, Found. Trends Machine Learning, 3(1) (2010), pp. 1-122.

[3] R. H. Chan, J. F. Yang and X. M. Yuan, *Alternating direction method for image inpainting in wavelet domain*, SIAM J. Imaging Sci., 4 (2011), pp. 807-826.

[4] T. F. Chan and R. Glowinski, *Finite element approximation and iterative solution of a class of mildly non-linear elliptic equations*, technical report, Stanford University, 1978.

[5] C. H. Chen, B. S. He and X. M. Yuan, *Matrix completion via alternating direction method*, IMA J. Numer. Anal., 32 (2012) 227-245.

[6] 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.

[7] E. Esser, *Applications of Lagrangian-based alternating direction methods and connections to split Bregman*, UCLA CAM Report, 2009.

[8] F. Facchinei and J. S. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, Vol. I. Springer Series in Operations Research, Springer Verlag, New York, 2003.

[9] M. Fortin and R. Glowinski, Augmented Lagrangian Methods: Applications to the Numerical Solutions of Boundary Value Problems, Stud. Math. Appl. 15, North-Holland, Amsterdam, 1983.

[10] M. Fukushima, *Application of the alternating direction method of multipliers to separable convex programming problems*, Comput. Optim. Appli., 2 (1992), pp. 93-111.

[11] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appli., 2 (1976), pp. 17-40.

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

[13] R. Glowinski and P. Le Tallec, Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics, SIAM Studies in Applied Mathematics, Philadelphia, PA, 1989.

[14] 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.

[15] D. R. Han and X. M. Yuan, *A note on the alternating direction method of multipliers*, J. Optim. Theory Appli., to appear.

[16] B. S. He, L. Z. Liao, D. R. Han and H. Yang, *A new inexact alternating directions method for monontone variational inequalities*, Math. Program., 92 (2002), pp. 103-118.

[17] 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.

[18] B. S. He, M. H. Xu and X. M. Yuan, *Solving large-scale least squares covariance matrix problems by alternating direction methods*, SIAM J. Matrix Anal. Appli., 32(1) (2011), pp. 136-152.

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

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

[21] S. Kontogiorgis and R. R. Meyer, *A variable-penalty alternating directions method for convex optimization*, Math. Program., 83 (1998), pp. 29-53.

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

[23] M. K. Ng, P.A. Weiss and X.M. Yuan, *Solving constrained total-variation problems via alternating direction methods*, SIAM J. Sci. Comput., 32(5) (2010), pp. 2710-2736.

[24] M. K. Ng, F. Wang and X. M. Yuan, *Inexact alternating direction methods for image recovery*, SIAM J. Sci. Comput., 33(2011), pp. 1643-1668.

[25] M. K. Ng, X. M. Yuan and W. X. Zhang, *On variational image decomposition model for blurred images with missing pixel values*, IEEE Trans. Imaging Sci., to appear.

[26] 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 Trans. PAMI, to appear.

[27] 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.

[28] J. Sun and S. Zhang, *A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs*, European J. Oper. Res., 207 (2010), pp. 1210-1220.

[29] A. Ruszczyński, *Parallel decomposition of multistage stochastic programming problems*, Math. Program., 58(1993), pp.201–228.

[30] M. Tao and X. M. Yuan, *Recovering low-rank and sparse components of matrices from incomplete and noisy observations*, SIAM J. Optim, 21(1) (2011), pp.57-81.

[31] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu and K. Knight, *Sparsity and smoothness via the fused lasso*, J. Royal Statist. Soc., 67 (2005), pp. 91-108.

[32] P. Tseng, *Applications of a splitting algorithm to decomposition in convex programming and variational inequalities*, SIAM J. Con. Optim., 29 (1991), pp. 119-138.

[33] S. Zhang, J. Ang and J. Sun, *An alternating direction method for solving convex nonlinear semidefinite programming problem*, Optim., to appear.

[34] Z. Zhou, X. Li, J. Wright, E. J. Candes and Y. Ma, *Stable principal component pursuit*, Proceedings IEEE Intern. Sympos. Inform. Theory. 2010.