SIAM J. OPTIM.
Vol. 25, No. 4, pp. 2274–2312




# ON FULL JACOBIAN DECOMPOSITION OF THE AUGMENTED LAGRANGIAN METHOD FOR SEPARABLE CONVEX PROGRAMMING*

BINGSHENG HE†, LIUSHENG HOU‡, AND XIAOMING YUAN§

**Abstract.** The augmented Lagrangian method (ALM) is a benchmark for solving a convex minimization model with linear constraints. We consider the special case where the objective is the sum of $m$ functions without coupled variables. For solving this separable convex minimization model, it is usually required to decompose the ALM subproblem at each iteration into $m$ smaller subproblems, each of which only involves one function in the original objective. Easier subproblems capable of taking full advantage of the functions' properties individually could thus be generated. In this paper, we focus on the case where full Jacobian decomposition is applied to ALM subproblems, i.e., all the decomposed ALM subproblems are eligible for parallel computation at each iteration. For the first time, we show, by an example, that the ALM with full Jacobian decomposition could be divergent. To guarantee the convergence, we suggest combining a relaxation step and the output of the ALM with full Jacobian decomposition. A novel analysis is presented to illustrate how to choose refined step sizes for this relaxation step. Accordingly, a new splitting version of the ALM with full Jacobian decomposition is proposed. We derive the worst-case $O(1/k)$ convergence rate measured by the iteration complexity (where $k$ represents the iteration counter) in both the ergodic and nonergodic senses for the new algorithm. Finally, some numerical results are reported to show the efficiency of the new algorithm.

**Key words.** convex programming, augmented Lagrangian method, Jacobian decomposition, contraction methods, convergence rate, operator splitting methods

**AMS subject classifications.** 90C25, 90C06, 65K05

**DOI.** 10.1137/130922793

## 1. Introduction.

A canonical optimization model is the convex minimization problem with linear constraints:

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

where $A \in \Re^{l\times n}$, $b \in \Re^{l}$, $\mathcal{X} \subseteq \Re^{n}$ is a closed convex set, $\theta : \Re^n \to \Re$ is a convex function (could be nonsmooth). To solve (1.1), the augmented Lagrangian method (ALM) in [23, 33] turns out to be a benchmark in both theoretical and algorithmic aspects. Starting from $\lambda^0 \in \Re^l$, the ALM generates a sequence $\{(x^k, \lambda^k)\}$ via the following scheme:

$$\begin{cases} x^{k+1} = \arg\min_{x\in\mathcal{X}} L_A(x, \lambda^k), \\ \lambda^{k+1} = \lambda^k - H(Ax^{k+1} - b), \end{cases} \tag{1.2}$$

*Received by the editors May 29, 2013; accepted for publication (in revised form) August 21, 2015; published electronically November 17, 2015.
http://www.siam.org/journals/siopt/25-4/92279.html

†Department of Mathematics, South University of Science and Technology of China, and Department of Mathematics, Nanjing University, China (hebma@nju.edu.cn). This author was supported by the NSFC grant 11471156.
‡School of Mathematics and Information Technology, Nanjing Xiaozhuang University, Nanjing, 211117, China (houlsheng@gmail.com). This author was supported by the NSFC grant 11471156.
§Department of Mathematics, Hong Kong Baptist University, Hong Kong, China (xmyuan@hkbu.edu.hk). This author was supported by the General Research Fund from Hong Kong Research Grants Council: HKBU203613.

where

$$L_A(x,\lambda) = \theta(x) - \lambda^T(Ax-b) + \frac{1}{2}\|Ax-b\|_H^2$$

denotes the augmented Lagrangian function of (1.1); $\lambda \in \Re^l$ is the Lagrange multiplier and $H \in \Re^{l\times l}$ is a positive definite matrix playing the role of a penalty parameter (in applications, we usually choose $H$ as a scalar matrix: $H = \beta I_l$ with $\beta > 0$ and $I_l$ is the identity matrix in $\Re^{l\times l}$). Note that here and after, $\|x\|_H := \sqrt{x^T H x}$, where $x$ and the positive definite matrix $H$ have appropriate dimensions. In [34], it was analyzed that the ALM is indeed an application of the proximal point algorithm (PPA) in [28] to the dual of (1.1). Throughout our discussion, the penalty matrix $H$ is assumed to be fixed.

When specific applications of (1.1) are considered, the abstract model (1.1) can often be specified with favorable structures. One typical example is the case where the objective function can be expressed as the sum of $m$ ($m \geq 2$) functions without coupled variables, each function referring to a particular objective of modeling. We thus consider the following special form of the canonical convex minimization model (1.1):

$$\begin{array}{ll} \min & \sum_{i=1}^m \theta_i(x_i), \\ & \sum_{i=1}^m A_i x_i = b, \\ & x_i \in \mathcal{X}_i,\ i = 1,\ldots,m, \end{array} \tag{1.3}$$

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; $b \in \Re^l$ is a given vector; and $\sum_{i=1}^m n_i = n$. Note that the variable $x$ is also partitioned into $m$ subvectors, i.e., $x = (x_1, x_2 \ldots, x_m)$, each $x_i \in \Re^{n_i}$ can be explained as the decision variable of the $i$th objective $\theta_i$. The coefficient matrix $A$ is partitioned accordingly as $(A_1, A_2, \ldots, A_m)$ in (1.3). Throughout, the solution set of (1.3) is assumed to be nonempty.

Let the Lagrangian function of (1.3) be

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

and the augmented Lagrangian function of (1.3) be

$$L_A(x_1,\ldots,x_m,\lambda) = L(x_1,\ldots,x_m,\lambda) + \frac{1}{2}\left\|\sum_{i=1}^m A_i x_i - b\right\|_H^2$$

with $\lambda \in \Re^l$ the Lagrange multiplier and $H \in \Re^{l\times l}$ the penalty matrix. Applying the generic ALM scheme (1.2) straightforwardly to the well-structured form (1.3), the iterative scheme is

$$\left\{ \begin{array}{l} (x_1^{k+1},\ldots,x_m^{k+1}) = \arg\min\left\{L_A(x_1,\ldots,x_m,\lambda^k) \,\middle|\, x_i \in \mathcal{X}_i,\ i=1,\ldots,m\right\}, \\ \qquad\qquad \lambda^{k+1} = \lambda^k - H(\sum_{i=1}^m A_i x_i^{k+1} - b). \end{array} \right. \tag{1.5}$$

This is an exact execution of the ALM; thus the sequence generated by (1.5) has the known convergence of the ALM such as those in [23, 33, 35]. But, for the separable case with $m \geq 2$, the implementation of (1.5) may have the difficulty that all the subvectors $x_i$'s are required to be solved simultaneously and all $\theta_i$ are considered aggregately. Even though each $\theta_i$ is simple in the sense that the resolvent operator of $\partial\theta_i$ has a closed-form expression (e.g., $\theta_i(x_i) = \|x_i\|_1$ or $\frac{1}{2}\|x_i\|^2$), the $x$-subproblem in (1.5) might not be easy. Therefore, the straightforward implementation (1.5) of the ALM could be inefficient for the particularly structured model (1.3). One strategy for effectively taking advantage of $\theta_i$'s properties individually is to decompose the $x$-subproblem in (1.5) into $m$ smaller ones. Accordingly, the objective function of the $i$th decomposed subproblem involves only $\theta_i(x_i)$ and a simple quadratic term. This treatment thus results in subproblems that are easy enough to have closed-form solutions for many applications arising in diverse areas such as image processing, statistical learning, and compressive sensing. Therefore, splitting versions of the ALM have received wide attention for solving the separable convex programming model (1.3).

A fundamental splitting version of the ALM is the Douglas–Rachford alternating direction method of multipliers (ADMM for short) proposed in [13] (see also [10]) for the special case of (1.3) with $m = 2$. At each iteration, the ADMM splits the ALM subproblem into two smaller subproblems in Gauss–Seidel order, and generates the next iterate $(x_1^{k+1}, x_2^{k+1}, \lambda^{k+1})$ via the following scheme:

$$(1.6)\quad \begin{cases} x_1^{k+1} = \arg\min\{\theta_1(x_1) - x_1^T A_1^T \lambda^k + \frac{1}{2}\|A_1 x_1 + A_2 x_2^k - b\|_H^2 \mid x_1 \in \mathcal{X}_1\}, \\ x_2^{k+1} = \arg\min\{\theta_2(x_2) - x_2^T A_2^T \lambda^k + \frac{1}{2}\|A_1 x_1^{k+1} + A_2 x_2 - b\|_H^2 \mid x_2 \in \mathcal{X}_2\}, \\ \lambda^{k+1} = \lambda^k - H(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b). \end{cases}$$

We refer to, e.g., [6, 7, 9, 11, 12, 14, 19, 27, 37], for some earlier articles in the areas of partial differential equations, convex programming, and variational inequalities. In the review paper on the ADMM [2], the authors commented 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." One may immediately want to extend the idea of (1.6) to the generic case of (1.3) with $m \geq 3$, and propose the following splitting version of ALM with full Gauss–Seidel decomposition:

(1.7)

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

Despite the efficiency of (1.7) having been verified empirically in various contexts (e.g., [32, 36]), it was recently shown in [3] that the scheme (1.7) is not necessarily convergent. We refer to [3, 15, 20, 24] for some techniques to ensure the convergence of (1.7) under some additional assumptions.

In addition to (1.7), an equally important splitting version for solving (1.3) is the ALM with full Jacobian decomposition whose decomposed subproblems are as follows:

$$(1.8)\quad \begin{cases} x_1^{k+1} = \arg\min\{\theta_1(x_1) - x_1^T A_1^T \lambda^k + \frac{1}{2}\|A_1 x_1 + \sum_{j=2}^m A_j x_j^k - b\|_H^2 \mid x_1 \in \mathcal{X}_1\}, \\ x_2^{k+1} = \arg\min\{\theta_2(x_2) - x_2^T A_2^T \lambda^k \\ \qquad\qquad + \frac{1}{2}\|A_1 x_1^k + A_2 x_2 + \sum_{j=3}^m A_j x_j^k - b\|_H^2 \mid x_2 \in \mathcal{X}_2\}, \\ \cdots\cdots \\ x_i^{k+1} = \arg\min\{\theta_i(x_i) - x_i^T A_i^T \lambda^k \\ \qquad\qquad + \frac{1}{2}\| \sum_{j=1}^{i-1} A_j x_j^k + A_i x_i + \sum_{j=i+1}^m A_j x_j^k - b\|_H^2 \mid x_i \in \mathcal{X}_i\}, \\ \cdots\cdots \\ x_m^{k+1} = \arg\min\{\theta_m(x_m) - x_m^T A_m^T \lambda^k + \frac{1}{2}\| \sum_{j=1}^{m-1} A_j x_j^k + A_m x_m - b\|_H^2 \mid x_m \in \mathcal{X}_m\}, \\ \lambda^{k+1} = \lambda^k - H(\sum_{j=1}^m A_j x_j^{k+1} - b). \end{cases}$$

Different from (1.7), the splitting version of ALM with full Jacobian decomposition (1.8) enjoys the feature that all the $x_i$-subproblems can be solved in parallel, and this is an important feature when large- or huge-scale data are under consideration and when parallel computing infrastructures are available. Given the divergence of the Gauss–Seidel splitting scheme (1.7), it seems natural to conjecture that the Jacobian splitting scheme (1.8) should not be convergent as it is an even less accurate approximation to the ALM step (1.5) than (1.7). We will give an example to show that this conjecture is indeed true; see the appendix. The output of (1.8) thus cannot be used as the next iterate directly. This is the first contribution of this paper.

To tackle the divergence of (1.8), one strategy is combining the output of (1.8) with an underrelaxation step. In [16, 18, 25], some such steps were proposed for the special cases of $m = 2$, $m = 3$, and $m \geq 3$, respectively. In [16] (for the case $m \geq 3$) and [18] (for the case $m = 3$), it was suggested to further adjust the output of (1.8) via the step

$$(1.9)\qquad w^{k+1} = w^k - \alpha(w^k - \tilde{w}^k),$$

where $\alpha > 0$ is a chosen step size, $w^k = (x_1^k, x_2^k, \ldots, x_m^k, \lambda^k)$ and $\tilde{w}^k$ denotes the output of (1.8) with the input $w^k$. The step (1.9) is indeed very simple; we thus stick to this scheme to investigate how to combine an underrelaxation step with the splitting ALM step (1.8) to ensure convergence. Intuitively, we can understand the underrelaxation step (1.9) in this way: Since the output $\tilde{w}^k$ of (1.8) is a Jacobian decomposition of the real ALM step (1.5) and it might be too inaccurate to be the new iterate (especially when $m$ is large), we compensate for this loss of accuracy by combining the last iterate $w^k$ with $\tilde{w}^k$ approximately (i.e., seeking an appropriate step size $\alpha$). Technically, as Theorem 4.7 shows in section 4, this underrelaxation step with an appropriate step size $\alpha$ can ensure the strict contraction of the iterative sequence and thus the global convergence becomes provable by following the standard analytic framework of contraction methods in [1]. With the given moving direction $(\tilde{w}^k - w^k)$, the emphasis of designing (1.9) is thus to refine the step size $\alpha$ for (1.9) or, more specifically, to enlarge the range of possible step sizes. In [18], for the case of (1.3) with $m = 3$, it was shown that the upper bound of the range of step size is $2 - \sqrt{3}$; and in [16], for the case of (1.3) with $m \geq 3$, the upper bound is $1/(3m+1)$. We shall show that the upper bound of the range of step sizes in (1.9) can be enlarged to $2(1-\sqrt{\frac{m}{m+1}})$, i.e., $\alpha \in (0, 2(1-\sqrt{\frac{m}{m+1}}))$, to ensure the convergence of the combination of (1.8) with (1.9). In particular, because it holds that

$$\frac{1}{m+1} < 2\left(1 - \sqrt{\tfrac{m}{m+1}}\right)$$

for any integer $m > 0$, we can simply take a constant step size in (1.9) as

$$w^{k+1} = w^k - \frac{1}{m+1}(w^k - \tilde{w}^k); \tag{1.10}$$

see Remark 3.3 for details. With a constant step size, the underrelaxation step (1.10) is extremely easy to implement and the additional computation is negligible in comparison with the splitting ALM step (1.8). Note that in (1.10) the constant step size is nearly three times larger than the lower bound $1/(3m+1)$ derived in [16]. With this refined step size in (1.9), a new splitting version of ALM with full Jacobian decomposition is thus derived. This is the second contribution of this paper.

Our third contribution is to establish the worst-case $O(1/k)$ convergence rate measured by the iteration complexity (where $k$ represents the iteration counter) in both the ergodic and nonergodic senses for the new splitting version of ALM with full Jacobian decomposition. Note that we follow the work [29, 30] and many others, where a worst-case $O(1/k)$ convergence rate measured by the iteration complexity means the accuracy to a solution under certain criteria is of the order $O(1/k)$ after $k$ iterations of an iterative scheme or, equivalently, it requires at most $O(1/\epsilon)$ iterations to achieve an approximate solution with an accuracy of $\epsilon$. This line of analysis is mainly motivated by our recent work of convergence analysis for the ADMM in [21, 22].

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 notation for the convenience of discussion. In section 3, we propose two algorithms based on the new splitting version of ALM with full Jacobian decomposition. Then, we prove the global convergence for the algorithms in section 4 by using the technique of contraction methods in [1]. The rationale of choosing a refined step size in (1.9) is also explained in this section. In sections 5 and 6, we establish the worst-case $O(1/k)$ convergence rate for Algorithms 1 and 2 in the ergodic and nonergodic senses, respectively. In section 7, we report some numerical results to show the efficiency of the proposed algorithms. Finally, we draw some conclusions in section 8.

## 2. Preliminaries.

In this section, we summarize some preliminaries which are useful for further discussions and then give some notation to be used.

### 2.1. A variational characterization of (1.3).

We first reformulate (1.3) as a variational form, which is useful when we establish the global convergence and worst-case convergence rates for the proposed splitting version of ALM with full Jacobian decomposition.

Recall that $L(x_1, x_2, \ldots, x_m, \lambda)$ defined in (1.4) is the Lagrange function of (1.3). Let $(x_1^*, x_2^*, \ldots, x_m^*, \lambda^*)$ be a saddle point of $L(x_1, x_2, \ldots, x_m, \lambda)$. Then, for any $\lambda \in \Re^l$ and $x_i \in \mathcal{X}_i$ $(i = 1, \ldots, m)$, we have

$$L(x_1^*, x_2^*, \ldots, x_m^*, \lambda) \leq L(x_1^*, x_2^*, \ldots, x_m^*, \lambda^*) \leq L(x_1, x_2, \ldots, x_m, \lambda^*).$$

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

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

such that

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

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

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

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

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

Note that the operator $F(w)$ defined in (2.2b) is monotone because it is affine with a skew-symmetric matrix. Since we have assumed that the solution set of (1.3) is nonempty, the solution set of $\text{VI}(\mathcal{W}, F, \theta)$, denoted by $\mathcal{W}^*$, is also nonempty.

**2.2. A characterization of $\mathcal{W}^*$.** We recall a characterization of $\mathcal{W}^*$, which is the basis of our discussion for establishing the worst-case convergence rate in the ergodic sense in section 5. We refer to Theorem 2.3.5 in [8] and Theorem 2.1 in [21] 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*

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

According to Theorem 2.1, for a given $\epsilon > 0$, we say $\bar{w} \in \mathcal{W}$ is an $\epsilon$-approximate solution when

$$\sup_{w \in D_{\mathcal{W}}(\bar{w})} \left\{\theta(\bar{x}) - \theta(x) + (\bar{w} - w)^T F(w)\right\} \leq \epsilon, \tag{2.4}$$

where

$$D_{\mathcal{W}}(\bar{w}) = \left\{w \in \mathcal{W} \mid \|w - \bar{w}\| \leq 1\right\}. \tag{2.5}$$

We refer to [31] for a similar definition of the $\epsilon$-approximate solution.

**2.3. Some matrices.** To present our analysis with succinct notation, we need to define some symmetric matrices. More specifically, let

$$S = \begin{pmatrix} 0 & -A_1^T H A_2 & \cdots & -A_1^T H A_m & 0 \\ -A_2^T H A_1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & -A_{m-1}^T H A_m & \vdots \\ -A_m^T H A_1 & \cdots & -A_m^T H A_{m-1} & 0 & 0 \\ 0 & \cdots & \cdots & 0 & H^{-1} \end{pmatrix} \tag{2.6}$$

and

$$G = \begin{pmatrix} 2A_1^T H A_1 & A_1^T H A_2 & \cdots & A_1^T H A_m & 0 \\ A_2^T H A_1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & A_{m-1}^T H A_m & \vdots \\ A_m^T H A_1 & \cdots & A_m^T H A_{m-1} & 2A_m^T H A_m & 0 \\ 0 & \cdots & \cdots & 0 & H^{-1} \end{pmatrix}. \tag{2.7}$$

In addition, we let

$$P = (A_1, A_2, \ldots, A_m, 0)^T H (A_1, A_2, \ldots, A_m, 0), \tag{2.8}$$

and thus have

$$G = S + 2P. \tag{2.9}$$

*Remark* 2.2. The matrix $M$ in [18] (see (4.5) on page 206 of [18]) is just the matrix $G$ here with $m = 3$. In addition, if $A_1, \ldots, A_m$ are full column rank matrices, then $G$ is positive definite.

**2.4. A proposition.** The following proposition can be proved by elementary techniques, and it will be used in later analysis.

PROPOSITION 2.3. *For any scalars $a_1 \ge a_2 \ge 0$, $b_1 \ge b_2 \ge 0$ with $a_2 + b_1 > 0$, we have*

$$\frac{a_1 + b_2}{a_1 + b_1} \ge \frac{a_2 + b_2}{a_2 + b_1}. \tag{2.10}$$

*For any nonzero vectors $p, q \in \Re^l$, positive definite matrix $H \in \Re^{l \times l}$, and $\tau > 0$, it holds that*

$$\frac{\tau \|p\|_H^2 + 2p^T q + \|q\|_{H^{-1}}^2}{\tau \|p\|_H^2 + \|q\|_{H^{-1}}^2} \ge 1 - \sqrt{\frac{1}{\tau}}. \tag{2.11}$$

*Proof.* The first assertion is trivial. Note that $H$ is positive definite; by a manipulation, we get

$$\begin{aligned}
&\frac{\tau\|p\|_H^2 + 2p^Tq + \|q\|_{H^{-1}}^2}{\tau\|p\|_H^2 + \|q\|_{H^{-1}}^2} \\
&\quad = \frac{\big(1-\tau^{-\frac{1}{2}}\big)(\tau\|p\|_H^2 + \|q\|_{H^{-1}}^2) + (\tau^{\frac{1}{2}}\|p\|_H^2 + 2p^Tq + \tau^{-\frac{1}{2}}\|q\|_{H^{-1}}^2)}{\tau\|p\|_H^2 + \|q\|_{H^{-1}}^2} \\
&\quad = (1-\tau^{-\frac{1}{2}}) \ + \ \frac{\tau^{-\frac{1}{2}}\|\tau^{\frac{1}{2}}p + H^{-1}q\|_H^2}{\tau\|p\|_H^2 + \|q\|_{H^{-1}}^2},
\end{aligned}$$

and the proof is complete. ∎

## 3. Algorithms.

In the introduction, we have explained that the new splitting version of ALM with full Jacobian decomposition is a combination of the splitting step (1.8) with the underrelaxation step (1.9). In this section, we delineate the details of choosing the step size $\alpha$ in (1.9) and derive two concrete algorithms. One has dynamically updated step sizes and the other has a constant step size.

### 3.1. Algorithm 1 with dynamically updated step sizes.

We first show that the step size in (1.9) can be chosen judiciously at each iteration and it may be updated dynamically. Recall that the output of (1.8) needs to be further adjusted. We thus relabel it as $\tilde{w}^k := (\tilde{x}_1^k, \tilde{x}_2^k, \dots, \tilde{x}_m^k, \tilde{\lambda}^k)$. That is, we can rewrite the splitting ALM step (1.8) as

(3.1)

$$\begin{cases}
\tilde{x}_1^k = \arg\min\big\{\theta_1(x_1) - x_1^T A_1^T\lambda^k + \frac{1}{2}\|A_1x_1 + \sum_{j=2}^m A_jx_j^k - b\|_H^2 \ \big| \ x_1 \in \mathcal{X}_1\big\}, \\
\tilde{x}_2^k = \arg\min\big\{\theta_2(x_2) - x_2^T A_2^T\lambda^k + \frac{1}{2}\|A_1x_1^k + A_2x_2 + \sum_{j=3}^m A_jx_j^k - b\|_H^2 \big| \ x_2 \in \mathcal{X}_2\big\}, \\
\cdots\cdots \\
\tilde{x}_i^k = \arg\min\big\{\theta_i(x_i) - x_i^T A_i^T\lambda^k \\
\qquad\qquad + \frac{1}{2}\|\sum_{j=1}^{i-1}A_jx_j^k + A_ix_i + \sum_{j=i+1}^m A_jx_j^k - b\|_H^2 \big| x_i \in \mathcal{X}_i\big\}, \\
\cdots\cdots \\
\tilde{x}_m^k = \arg\min\big\{\theta_m(x_m) - x_m^T A_m^T\lambda^k + \frac{1}{2}\|\sum_{j=1}^{m-1} A_jx_j^k + A_mx_m - b\|_H^2 \ \big| \ x_m \in \mathcal{X}_m\big\}, \\
\tilde{\lambda}^k = \lambda^k - H(\sum_{j=1}^m A_j\tilde{x}_j^k - b).
\end{cases}$$

*Remark* 3.1. We will show that the specific strategy determining $\alpha_k^*$ in (3.2b) comes from the purpose of maximizing a certain quadratic function which is beneficial for making more progress toward proximity to the solution set $\mathcal{W}^*$ (or, more intuitively, making the iterative sequence more "contractive"). This is a standard technique for contraction-type methods. The parameter $\gamma$ is a relaxation factor, and its restriction $\gamma \in (0, 2)$ is also for the purpose of ensuring the contraction of the iterative sequence (see (4.21) in Theorem 4.7).

*Remark* 3.2. The strategy of choosing the step size $\alpha_k$ in (3.2b) can be regarded as an extension of that in [18] for the special case where $m = 3$. More specifically, in section 4, we will prove that $\alpha_k^*$ defined in (3.2b) is uniformly lower bounded by $(1 - \sqrt{\frac{m}{m+1}})$ for all $k$'s. Note that

$$2\left(1 - \sqrt{\tfrac{m}{m+1}}\right) = 2 - \sqrt{3}$$

holds when $m = 3$. Our lower bound for general $m$ includes as a special case the upper bound of the range of step size derived in [18] for the special case of $m = 3$. As we

Algorithm 1. A splitting version of ALM with full Jacobian decomposition and dynamically updated step sizes.

Step 1: Generate $\tilde{w}^k$ via (3.1).
Step 2: Adjust $\tilde{w}^k$ and generate the new iterate $w^{k+1}$ via

$$w^{k+1} = w^k - \alpha_k(w^k - \tilde{w}^k), \tag{3.2a}$$

where

$$\alpha_k = \gamma\alpha_k^*, \qquad \alpha_k^* = \frac{\varphi(w^k, \tilde{w}^k)}{\|w^k - \tilde{w}^k\|_G^2}, \qquad \gamma \in (0, 2), \tag{3.2b}$$

$G$ is defined in (2.7), and

$$\varphi(w^k, \tilde{w}^k) = \|w^k - \tilde{w}^k\|_G^2 + 2(\lambda^k - \tilde{\lambda}^k)^T \left(\sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k)\right). \tag{3.2c}$$

Algorithm 2. A splitting version of ALM with full Jacobian decomposition and a constant step size.

Step 1: Generate $\tilde{w}^k$ via (3.1).
Step 2: Adjust $\tilde{w}^k$ and generate the new iterate $w^{k+1}$ via

$$w^{k+1} = w^k - \alpha(w^k - \tilde{w}^k), \tag{3.3a}$$

where

$$\alpha = \gamma\left(1 - \sqrt{\tfrac{m}{m+1}}\right) \qquad \text{and} \qquad \gamma \in (0, 2). \tag{3.3b}$$

have mentioned, the matrix $G$ defined in (2.7) reduces to the matrix $M$ in [18] when $m = 3$. Also, the function $\varphi(w^k, \tilde{w}^k)$ defined in (3.2c) reduces to the function in [18] (see (4.13)) when $m = 3$.

**3.2. Algorithm 2 with a constant step size.** For Algorithm 1, the step size $\alpha_k$ is calculated by (3.2b) and it is updated at each iteration. The advantage of doing so is that some beneficial step size at each iteration could be found towards the purpose of maximizing the contraction of the sequence. At the same time, this chosen step size requires additional computation and it might be computationally demanding (e.g., some large-scale cases where large matrix variables are considered). We are thus also interested in the case where the step size of the underrelaxation step is fixed as a constant throughout the iteration. This can be done by choosing a uniform lower bound of the sequence $\{\alpha_k\}$ determined in (3.2b) as the constant step size.

As we have mentioned, we will prove later that $\alpha_k^*$ defined in (3.2b) satisfies $\alpha_k^* \geq (1 - \sqrt{\frac{m}{m+1}})$ for all $k$'s. We can thus take $1 - \sqrt{\frac{m}{m+1}}$ as a constant step size and a splitting version of ALM with full Jacobian decomposition and a constant step size is ready to be presented. This treatment is certainly more conservative than the strategy of dynamically updating the step size in Algorithm 1 and thus it is expected to require more iterations to achieve the same level of solution accuracy. But it enjoys cheaper computation at each iteration. Thus it is not conclusive which one is more preferable because it really depends on the specific problem setting of (1.3).

*Remark* 3.3. Note it holds that

$$1 - \sqrt{\frac{m}{m+1}} = \frac{1 - \frac{m}{m+1}}{1 + \sqrt{\frac{m}{m+1}}} \quad > \quad \frac{1 - \frac{m}{m+1}}{2} = \frac{1}{2(m+1)} \tag{3.4}$$

for any integer $m > 0$. The constant step size defined in (3.3b) thus satisfies

$$\alpha > \frac{\gamma}{2(m+1)}. \tag{3.5}$$

If $\gamma$ is taken as $(1 + \sqrt{\frac{m}{m+1}}\,) \in (0, 2)$ in (3.3b), then we have $\alpha = \frac{1}{m+1}$. This means the underrelaxation step (3.3) reduces to

$$w^{k+1} = w^k - \frac{1}{m+1}(w^k - \tilde{w}^k).$$

**4. Global convergence.** In this section, we prove the global convergence for Algorithms 1 and 2. As we have mentioned, the proof follows the standard analytic framework of contraction methods in [1] (see also [17]).

We first try to quantity the difference between the output $\tilde{w}^k$ of the splitting step (3.1) and a solution point in $\mathcal{W}^*$ by means of the characterization of $\mathcal{W}^*$ in (2.3). The result is shown in the following lemma.

LEMMA 4.1. *Let $\tilde{w}^k$ be the output of the splitting step* (3.1) *with given $w^k$. Then, we have*

$$\tilde{w}^k \in \mathcal{W}, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T \{F(\tilde{w}^k) + S(\tilde{w}^k - w^k)\} \geq 0 \quad \forall\, w \in \mathcal{W}, \tag{4.1}$$

*where $S$ is defined in* (2.6).

*Proof.* It follows from (3.1) that for $i = 1, 2, \ldots, m$, it holds that

$$\begin{aligned} &\tilde{x}_i^k \in \mathcal{X}_i, \ \ \theta_i(x_i) - \theta_i(\tilde{x}_i^k) \\ &\qquad + (x_i - \tilde{x}_i^k)^T \left\{ -A_i^T \lambda^k + A_i^T H \left( A\tilde{x}_i^k + \sum_{j=1,\, j\neq i}^{m} A_j x_j^k - b \right) \right\} \geq 0 \ \ \forall\, x_i \in \mathcal{X}_i. \end{aligned} \tag{4.2}$$

Substituting $\tilde{\lambda}^k = \lambda^k - H(\sum_{j=1}^m A_j \tilde{x}_j^k - b)$ (see also (3.1)) into the above inequality, we obtain

$$\begin{aligned} &\tilde{x}_i^k \in \mathcal{X}_i, \ \theta_i(x_i) - \theta_i(\tilde{x}_i^k) \\ &\qquad + (x_i - \tilde{x}_i^k)^T \left\{ -A_i^T \tilde{\lambda}^k - A_i^T H \left( \sum_{j=1, j\neq i}^{m} A_j(\tilde{x}_j^k - x_j^k) \right) \right\} \geq 0 \ \ \forall\, x_i \in \mathcal{X}_i. \end{aligned} \tag{4.3}$$

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

$$\begin{aligned} &\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x_1 - \tilde{x}_1^k \\ \vdots \\ x_i - \tilde{x}_i^k \\ \vdots \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A_1^T \tilde{\lambda}^k \\ \vdots \\ -A_i^T \tilde{\lambda}^k \\ \vdots \\ -A_m^T \tilde{\lambda}^k \end{pmatrix} - \begin{pmatrix} A_1^T H\big(\sum_{j=2}^m A_j(\tilde{x}_j^k - x_j^k)\big) \\ \vdots \\ A_i^T H\big(\sum_{j=1,j\neq i}^m A_j(\tilde{x}_j^k - x_j^k)\big) \\ \vdots \\ A_m^T H\big(\sum_{j=1}^{m-1} A_j(\tilde{x}_j^k - x_j^k)\big) \end{pmatrix} \right\} \\ &\quad \geq 0 \ \ \forall\, w \in \mathcal{W}. \end{aligned} \tag{4.4}$$

The last equation in (3.1) can be rewritten as

$$\left(\sum_{j=1}^{m} A_j \tilde{x}_j^k - b\right) + H^{-1}(\tilde{\lambda}^k - \lambda^k) = 0$$

and in variational form

$$\tilde{\lambda}^k \in \Re^l, \quad (\lambda - \tilde{\lambda}^k)^T \left\{ \left(\sum_{j=1}^{m} A_j \tilde{x}_j^k - b\right) + H^{-1}(\tilde{\lambda}^k - \lambda^k) \right\} \geq 0 \quad \forall\, \lambda \in \Re^l. \tag{4.5}$$

Combining (4.4) and (4.5), we get $\tilde{w}^k \in \mathcal{W}$ and

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

for all $w \in \mathcal{W}$. Using the notation of $F$ (see (2.2b)) and $S$ (see (2.6)), the above inequality can be rewritten as

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

The assertion (4.1) thus is proved. ∎

Recall the VI characterization (2.2) of the optimization problem (1.3). Then, the assertion (4.1) inspires us to investigate the term $(\tilde{w}^k - w^*)^T S(w^k - \tilde{w}^k)$.

LEMMA 4.2. *Let $\tilde{w}^k$ be the output of the splitting step* (3.1) *with given $w^k$. Then, we have*

$$(\tilde{w}^k - w^*)^T S(w^k - \tilde{w}^k) \geq 0 \quad \forall\, w^* \in \mathcal{W}^*, \tag{4.6}$$

*where $S$ is defined in* (2.6).

*Proof.* The proof is an immediate conclusion based on the assertion (4.1) and the monotonicity of $F$. In fact, for an arbitrarily fixed $w^* \in \mathcal{W}^*$, it follows from (4.1) that

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

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 the assertion (4.6) is true. ∎

LEMMA 4.3. *Let $\tilde{w}^k$ be the output of the splitting step* (3.1) *with given $w^k$. Then, we have*

$$(w^k - w^*)^T G(w^k - \tilde{w}^k) \geq \varphi(w^k, \tilde{w}^k) \quad \forall\, w^* \in \mathcal{W}^*, \tag{4.7}$$

*where $G$ is defined in* (2.7) *and $\varphi(w^k, \tilde{w}^k)$ is defined in* (3.2c).

*Proof.* Since $G = S + 2P$ (see (2.9)), we first show that

$$(\tilde{w}^k - w^*)^T P(w^k - \tilde{w}^k) = (\lambda^k - \tilde{\lambda}^k)^T \left( \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) \right) \quad \forall\, w^* \in \mathcal{W}^*. \tag{4.8}$$

Because $P = (A_1, A_2, \dots, A_m, 0)^T H(A_1, A_2, \dots, A_m, 0)$ (see (2.8)), we have

$$(\tilde{w}^k - w^*)^T P(w^k - \tilde{w}^k) = \left( \sum_{i=1}^m A_i(\tilde{x}_i^k - x_i^*) \right)^T H \left( \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) \right).$$

By using

$$\sum_{i=1}^m A_i x_i^* = b \qquad \text{and} \qquad H \left( \sum_{i=1}^m A_i \tilde{x}_i^k - b) \right) = \lambda^k - \tilde{\lambda}^k \quad \text{(see (3.1))},$$

we get

$$\left( \sum_{i=1}^m A_i(\tilde{x}_i^k - x_i^*) \right)^T H \left( \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) \right) = (\lambda^k - \tilde{\lambda}^k)^T \left( \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) \right).$$

The assertion (4.8) follows from the above equations directly. Adding

$$(\tilde{w}^k - w^*)^T (2P)(w^k - \tilde{w}^k) = 2(\lambda^k - \tilde{\lambda}^k)^T \left( \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) \right)$$

to both sides of (4.6) and using $G = S + 2P$, we get

$$(\tilde{w}^k - w^*)^T G(w^k - \tilde{w}^k) \geq 2(\lambda^k - \tilde{\lambda}^k)^T \left( \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) \right).$$

The assertion (4.7) follows from the above inequality and the definition of $\varphi(w^k, \tilde{w}^k)$ directly. ∎

Following the analytic framework of convergence analysis for contraction methods in [1, 17], we now need to prove that

$$\varphi(w^k, \tilde{w}^k) \geq \delta \|w^k - \tilde{w}^k\|_G^2$$

for a certain constant $\delta > 0$ which is only dependent on $m$. We show this fact in Lemma 4.4 by using Proposition 2.3.

LEMMA 4.4. *Let $\tilde{w}^k$ be the output of the splitting step* (3.1) *with given $w^k$. Then, we have*

$$\varphi(w^k, \tilde{w}^k) \geq \left( 1 - \sqrt{\frac{m}{m+1}} \right) \|w^k - \tilde{w}^k\|_G^2. \tag{4.9}$$

*Proof.* Using the notation of $G$ (see (2.7)), we get

$$\|w^k - \tilde{w}^k\|_G^2 = \sum_{i=1}^m \|A_i(x_i^k - \tilde{x}_i^k)\|_H^2 + \left\| \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) \right\|_H^2 + \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2.$$

Substituting it into the expression of $\varphi(w^k, \tilde{w}^k)$ (see (3.2c)) we obtain

$$\varphi(w^k, \tilde{w}^k) = \sum_{i=1}^m \|A_i(x_i^k - \tilde{x}_i^k)\|_H^2 + \left\| \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) + H^{-1}(\lambda^k - \tilde{\lambda}^k) \right\|_H^2 .$$

Therefore, we have

$$\frac{\varphi(w^k, \tilde{w}^k)}{\|w^k - \tilde{w}^k\|_G^2} = \frac{\sum_{i=1}^m \|A_i(x_i^k - \tilde{x}_i^k)\|_H^2 + \|\sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) + H^{-1}(\lambda^k - \tilde{\lambda}^k)\|_H^2}{\sum_{i=1}^m \|A_i(x_i^k - \tilde{x}_i^k)\|_H^2 + \|\sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k)\|_H^2 + \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2}. \tag{4.10}$$

Note that we need only to prove the assertion with the assumption

$$\left\| \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) \right\|_H^2 + \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2 \geq \left\| \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) + H^{-1}(\lambda^k - \tilde{\lambda}^k) \right\|_H^2 ,$$

otherwise $\varphi(w^k, \tilde{w}^k) \geq \|w^k - \tilde{w}^k\|_G^2$ and (4.9) is true. By using

$$a_1 = \sum_{i=1}^m \|A_i(x_i^k - \tilde{x}_i^k)\|_H^2, \qquad b_1 = \left\| \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) \right\|_H^2 + \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2,$$

and

$$b_2 = \left\| \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) + H^{-1}(\lambda^k - \tilde{\lambda}^k) \right\|_H^2$$

in (4.10), we get

$$\frac{\varphi(w^k, \tilde{w}^k)}{\|w^k - \tilde{w}^k\|_G^2} = \frac{a_1 + b_2}{a_1 + b_1}. \tag{4.11}$$

We denote

$$a_2 = \frac{1}{m} \left\| \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) \right\|_H^2 .$$

Thus, we have $a_1 \geq a_2 \geq 0$. Then, using (4.11) and (2.10), we obtain

$$\begin{aligned} \frac{\varphi(w^k, \tilde{w}^k)}{\|w^k - \tilde{w}^k\|_G^2} &\geq \frac{a_2 + b_2}{a_2 + b_1} \\ &= \frac{\frac{1}{m}\|\sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k)\|_H^2 + \|\sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) + H^{-1}(\lambda^k - \tilde{\lambda}^k)\|_H^2}{\frac{1}{m}\|\sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k)\|_H^2 + \|\sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k)\|_H^2 + \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2} \end{aligned} \tag{4.12}$$

and, consequently,

$$\frac{\varphi(w^k, \tilde{w}^k)}{\|w^k - \tilde{w}^k\|_G^2} \geq \frac{\frac{m+1}{m}\|\sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k)\|_H^2 + 2(\lambda^k - \tilde{\lambda}^k)^T\big(\sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k)\big) + \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2}{\frac{m+1}{m}\|\sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k)\|_H^2 + \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2}.$$

To the right-hand side of the last inequality, by setting $\tau = \frac{m+1}{m}$, $p = \sum_{i=1}^{m} A_i(x_i^k - \tilde{x}_i^k)$, and $q = \lambda^k - \tilde{\lambda}^k$, and using (2.11), we obtain

$$\frac{\varphi(w^k, \tilde{w}^k)}{\|w^k - \tilde{w}^k\|_G^2} \geq 1 - \sqrt{\frac{m}{m+1}},$$

and thus the assertion (4.9) is proved. ∎

*Remark* 4.5. When $m = 3$, the assertion (4.9) reduces to

$$\varphi(w^k, \tilde{w}^k) \geq \frac{2 - \sqrt{3}}{2} \|w^k - \tilde{w}^k\|_G^2.$$

Since the matrix $G$ defined in (2.7) reduces to the matrix $M$ in [18], the relation (4.15) in [18] is a special result of the assertion (4.9) with $m = 3$. In other words, Lemma 4.4 includes Lemma 4.1 in [18] as a special case.

Now, combining the results of Lemmas 4.3 and 4.4, we have

$$(4.13) \qquad (w^k - w^*)^T G(w^k - \tilde{w}^k) \geq \left(1 - \sqrt{\frac{m}{m+1}}\right) \|w^k - \tilde{w}^k\|_G^2 \quad \forall w^* \in \mathcal{W}^*.$$

This means, $\tilde{w}^k - w^k$ is a descent direction of the distance function $\|w - w^*\|_G^2$ at the point $w^k$, even if $w^*$ is unknown. Along the direction $\tilde{w}^k - w^k$, by choosing a suitable step size $\alpha$, we can reduce the unknown distance function $\|w - w^*\|_G^2$. In order to explain how to determine the step size $\alpha_k$ in (3.2a) (resp., in (3.3a)), we define the step-size-dependent new iterate by

$$(4.14) \qquad w^{k+1}(\alpha) = w^k - \alpha(w^k - \tilde{w}^k).$$

LEMMA 4.6. *Let* $\tilde{w}^k$ *be the output of the splitting step* (3.1) *with given* $w^k$ *and* $w^{k+1}(\alpha)$ *be given by* (4.14). *Then we have*

$$(4.15) \qquad \vartheta(\alpha) \geq q(\alpha),$$

*where*

$$(4.16) \qquad \vartheta(\alpha) = \|w^k - w^*\|_G^2 - \|w^{k+1}(\alpha) - w^*\|_G^2$$

*and*

$$(4.17) \qquad q(\alpha) = 2\alpha\varphi(w^k, \tilde{w}^k) - \alpha^2 \|w^k - \tilde{w}^k\|_G^2.$$

*Proof.* By using (4.7) and the definition of $q(\alpha)$, we get

$$\begin{aligned}
\vartheta(\alpha) &= \|w^k - w^*\|_G^2 - \|w^{k+1}(\alpha) - w^*\|_G^2 \\
&= \|w^k - w^*\|_G^2 - \|(w^k - w^*) - \alpha(w^k - \tilde{w}^k)\|_G^2 \\
&= 2\alpha(w^k - w^*)^T G(w^k - \tilde{w}^k) - \alpha^2 \|w^k - \tilde{w}^k\|_G^2 \\
&\geq 2\alpha\varphi(w^k, \tilde{w}^k) - \alpha^2 \|w^k - \tilde{w}^k\|_G^2 \\
&= q(\alpha).
\end{aligned}$$

The lemma is proved. ∎

Ideally we want to maximize $\vartheta(\alpha)$. However, it is impossible due to the lack of the unknown solution point $w^*$. We thus turn to the second best choice: maximizing the quadratic function $q(\alpha)$ which is a lower bound of $\vartheta(\alpha)$. This promotes us to take the value of $\alpha$ as

$$\alpha_k^* = \frac{\varphi(w^k, \tilde{w}^k)}{\|w^k - \tilde{w}^k\|_G^2}. \tag{4.18}$$

According to (4.9), $\alpha_k^*$ is positive and

$$\alpha_k^* \geq 1 - \sqrt{\frac{m}{m+1}} \quad \forall\, k \geq 0. \tag{4.19}$$

It follows from (3.4) that $\alpha_k^* > \frac{1}{2(m+1)}$ and

$$\alpha_k = \gamma \alpha_k^* > \frac{\gamma}{2(m+1)}. \tag{4.20}$$

The "optimal" step size in the underrelaxation step (3.2) is bounded away from zero and only dependent on $m$. By using the step (3.3), we need only to chose a constant $\alpha$ to guarantee $q(\alpha) > 0$ in each iteration.

Now, we are at the stage to prove the global convergence of Algorithms 1 and 2. The following theorem is the main theorem regarding convergence.

THEOREM 4.7. *Let $\{w^k\}$ be the sequence generated by either Algorithm 1 or Algorithm 2 with an arbitrary initial iterate $w^0$. Then, it holds that*

$$\|w^{k+1} - w^*\|_G^2 \leq \|w^k - w^*\|_G^2 - \gamma(2-\gamma)\left(1 - \sqrt{\frac{m}{m+1}}\right)^2 \|w^k - \tilde{w}^k\|_G^2 \quad \forall\, w^* \in \mathcal{W}^*. \tag{4.21}$$

*Proof.* For any step size $\alpha > 0$ in the underrelaxation step (3.2a) of Algorithm 1 (resp., (3.3a) of Algorithm 2), according to Lemma 4.6, we have that

$$\|w^{k+1} - w^*\|_G^2 \leq \|w^k - w^*\|_G^2 - q(\alpha) \quad \forall w^* \in \mathcal{W}^*. \tag{4.22}$$

For Algorithm 1, $\alpha = \gamma\alpha_k^*$. Then, it follows from (4.17) and (4.18) that

$$q(\gamma\alpha_k^*) = 2\gamma\alpha_k^*\varphi(w^k, \tilde{w}^k) - (\gamma\alpha_k^*)^2\|w^k - \tilde{w}^k\|_G^2 = \gamma(2-\gamma)(\alpha_k^*)^2\|w^k - \tilde{w}^k\|_G^2. \tag{4.23}$$

Using the fact (4.19) in (4.23), we obtain

$$q(\gamma\alpha_k^*) \geq \gamma(2-\gamma)\left(1 - \sqrt{\frac{m}{m+1}}\right)^2 \|w^k - \tilde{w}^k\|_G^2,$$

and the first assertion (4.21) is proved. For Algorithm 2, $\alpha = \gamma(1 - \sqrt{\frac{m}{m+1}})$. Substituting it into (4.17), we get

$$q(\alpha) = 2\gamma\left(1 - \sqrt{\tfrac{m}{m+1}}\right)\varphi(w^k, \tilde{w}^k) - \gamma^2\left(1 - \sqrt{\tfrac{m}{m+1}}\right)^2 \|w^k - \tilde{w}^k\|_G^2.$$

Using the fact (4.9) we obtain

$$q(\alpha) \geq \gamma(2-\gamma)\left(1 - \sqrt{\frac{m}{m+1}}\right)^2 \|w^k - \tilde{w}^k\|_G^2, \tag{4.24}$$

and the assertion (4.21) is proved. The proof is complete. ∎

The assertion (4.21) still involves $\tilde{w}^k$. We can easily remove it and refine (4.21) as a recursive inequality between two consecutive iterates.

COROLLARY 4.8. *Let $\{w^k\}$ be the sequence generated by either Algorithm* 1 *or Algorithm* 2 *with an arbitrary initial iterate $w^0$. Then, it holds that*

$$\|w^{k+1} - w^*\|_G^2 \le \|w^k - w^*\|_G^2 - \frac{2-\gamma}{\gamma}\|w^k - w^{k+1}\|_G^2 \quad \forall\, w^* \in \mathcal{W}^*. \tag{4.25}$$

*Proof.* For Algorithm 1, it follows from (3.2a) that

$$\alpha_k^*(w^k - \tilde{w}^k) = \frac{1}{\gamma}(w^k - w^{k+1}),$$

and thus the assertion follows from (4.15)–(4.17) and (4.23) immediately. For Algorithm 2, it follows from (3.3a) that

$$\alpha(w^k - \tilde{w}^k) = (w^k - w^{k+1})$$

and

$$\alpha = \gamma\left(1 - \sqrt{\frac{m}{m+1}}\right).$$

Thus, the assertion (4.25) follows from (4.15)–(4.17) and (4.24) immediately. ∎

By using the following notation

$$y_i = A_i x_i,\ i = 1, \dots, m, \qquad v = (y_1, y_2, \dots, y_m, \lambda),$$
$$\mathcal{V}^* = \{(A_1x_1^*, A_2x_2^*, \dots, A_mx_m^*, \lambda^*) \,|\, (x_1^*, x_2^*, \dots, x_m^*, \lambda^*) \in \mathcal{W}^*\},$$

the convergence of Algorithm 1 or 2 can be shown by either $w^k \to w^*$ with $w^* \in \mathcal{W}^*$ or $v^k \to v^*$ with $v^* \in \mathcal{V}^*$ under different assumptions. In the following theorem, we only list the sketch of the proof and omit the detail.

THEOREM 4.9. *Let $\{w^k\}$ be the sequence generated by either Algorithm* 1 *or* 2 *with an arbitrary initial iterate $w^0$.*

1. *If all $A_i$, $i = 1, \dots, m$, in* (1.3) *are assumed to be full column rank, then $\{w^k\}$ converges to a point $w^*$ which is a solution point of VI$(\mathcal{W}, F, \theta)$.*
2. *Otherwise, the sequence $\{v^k\}$ converges to a point $v^*$ in $\mathcal{V}^*$.*

*Proof.* 1. We prove the first assertion. If all $A_i$, $i = 1, \dots, m$, are full column rank matrices, then the matrix $G$ defined in (2.7) is positive definite. It follows from (4.21) that the sequence $\{w^k\}$ is bounded and thus it has at least one cluster point, say $w^*$. Let $w^{k_j}$ be the subsequence converging to $w^*$. An immediate conclusion of (4.21) is that $\|\tilde{w}^{k_j} - w^{k_j}\| \to 0$. Then, taking the limit in (4.1) for $\{w^{k_j}\}$, we have that

$$\theta(x) - \theta(x^*) + (w - w^*)^T(F(w^*)) \ge 0 \ \ \forall w \in \mathcal{W},$$

which means, by the definition, that $w^*$ is a solution point of VI$(\mathcal{W}, F, \theta)$. Moreover, it follows from (4.21) trivially that the sequence $\{w^k\}$ cannot have two cluster points. Thus, the sequence $\{w^k\}$ converges to a point $w^*$ in $\mathcal{W}^*$. For the second assertion, it follows from (4.25) that

$$\|v^{k+1} - v^*\|_{\mathcal{H}}^2 \le \|v^k - v^*\|_{\mathcal{H}}^2 - \frac{2-\gamma}{\gamma}\|v^k - v^{k+1}\|_{\mathcal{H}}^2 \quad \forall\, v^* \in \mathcal{V}^*, \tag{4.26}$$

where

$$\mathcal{H} = \begin{pmatrix} 2H & H & \cdots & H & 0 \\ H & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & H & \vdots \\ H & \cdots & H & 2H & 0 \\ 0 & \cdots & \cdots & 0 & H^{-1} \end{pmatrix}_{(m+1)\times(m+1)}.$$

Obviously, the matrix $\mathcal{H}$ defined above is positive definite because $H$ is assumed to be positive definite. Hence, the sequence $\{v^k\}$ converges to a point $v^*$ in $\mathcal{V}^*$. The proof is completed. ∎

**5. Convergence rate in the ergodic sense.** In this section, we establish the worst-case $O(1/k)$ convergence rate in the ergodic sense for Algorithms 1 and 2. The technique of analysis is motivated by our recent result in [21].

More specifically, our goal is to show that Algorithm 1 or 2 needs at most $\lfloor \boldsymbol{O(1/\epsilon)} \rfloor$ iterations to find $\bar{w} \in \mathcal{W}$, an approximate solution of VI$(\mathcal{W}, F, \theta)$ with an accuracy of $\epsilon$ in the sense that

$$\theta(\bar{x}) - \theta(x) + (\bar{w} - w)^T F(w) \le \epsilon \quad \forall\, w \in D_{\mathcal{W}}(\bar{w}), \tag{5.1}$$

where $D_{\mathcal{W}}(\bar{w})$ is defined in (2.5). Recall that it is reasonable to use (5.1) to measure the accuracy of $\bar{w}$ to a solution point of VI$(\mathcal{W}, F, \theta)$, because of the characterization (2.3) in Theorem 2.1.

First of all, we define a new sequence $\bar{w}^k = (\bar{x}_1^k, \bar{x}_2^k, \ldots, \bar{x}_m^k, \bar{\lambda}^k)$ by

$$\begin{pmatrix} \bar{x}_1^k \\ \bar{x}_2^k \\ \vdots \\ \bar{x}_m^k \end{pmatrix} = \begin{pmatrix} \tilde{x}_1^k \\ \tilde{x}_2^k \\ \vdots \\ \tilde{x}_m^k \end{pmatrix} \quad \text{and} \quad \bar{\lambda}^k = \tilde{\lambda}^k + 2H \sum_{j=1}^m A_j(\tilde{x}_j^k - x_j^k), \tag{5.2}$$

where $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ is generated by the splitting step (3.1). This is to be used in the convergence rate analysis. Note that for $\bar{w}^k$ and $\tilde{w}^k$, only their $\lambda$-parts are different. By using $\tilde{x}_i = \bar{x}_i$ $(i = 1, \ldots, m)$ and (5.2), we have

$$\lambda^k - \bar{\lambda}^k = (\lambda^k - \tilde{\lambda}^k) + 2H \sum_{j=1}^m A_j(x_j^k - \tilde{x}_j^k). \tag{5.3}$$

In addition, we define the following two matrices:

$$Q = \begin{pmatrix} 2A_1^T H A_1 & A_1^T H A_2 & \cdots & A_1^T H A_m & 0 \\ A_2^T H A_1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & A_{m-1}^T H A_m & 0 \\ A_m^T H A_1 & \cdots & A_m^T H A_{m-1} & 2A_m^T H A_m & 0 \\ -2A_1 & \cdots & -2A_{m-1} & -2A_m & H^{-1} \end{pmatrix} \tag{5.4}$$

and

$$L = \begin{pmatrix} I_{n_1} & 0 & \cdots & 0 & 0 \\ 0 & I_{n_2} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & I_{n_m} & 0 \\ 2HA_1 & \cdots & 2HA_{m-1} & 2HA_m & I_l \end{pmatrix}. \tag{5.5}$$

Note that

$$\frac{Q^T+Q}{2} = \begin{pmatrix} 2A_1^T HA_1 & A_1^T HA_2 & \cdots & A_1^T HA_m & -A_1^T \\ A_2^T HA_1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & A_{m-1}^T HA_m & -A_{m-1}^T \\ A_m^T HA_1 & \cdots & A_m^T HA_{m-1} & 2A_m^T HA_m & -A_m^T \\ -A_1 & \cdots & -A_{m-1} & -A_m & H^{-1} \end{pmatrix}$$

$$= \mathcal{A}^T \begin{pmatrix} 2I_l & I_l & \cdots & I_l & -I_l \\ I_l & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & I_l & -I_l \\ I_l & \cdots & I_l & 2I_l & -I_l \\ -I_l & \cdots & -I_l & -I_l & I_l \end{pmatrix}_{(m+1)\times(m+1)} \mathcal{A}, \tag{5.6}$$

where

$$\mathcal{A} = \mathrm{diag}(H^{1/2}A_1, \ldots, H^{1/2}A_m, H^{-1/2}).$$

Thus, the matrix $Q^T + Q$ is positive semidefinite. In fact, $Q^T + Q$ is positive definite when all the matrices $A_i$'s in (1.3) are full column rank.

To establish the convergence rate, we need to use the relations of the matrices $Q$, $L$, and $G$, and the vectors $(\bar{w}^k - w^k)$ and $(\tilde{w}^k - w^k)$. The assertion in Lemma 5.1 follows from the definitions directly, and thus the proof is omitted.

Lemma 5.1. *For the above defined matrices $Q$ and $L$, we have*

$$QL = G, \tag{5.7}$$

*where $G$ is defined in* (2.7).

Lemma 5.2. *Let $\tilde{w}^k$ be the output of the splitting step* (3.1) *with given $w^k$ and the vector $\bar{w}^k$ be defined by* (5.2). *Then we have*

$$w^k - \bar{w}^k = L(w^k - \tilde{w}^k), \tag{5.8}$$

*where matrix $L$ is defined in* (5.5).

*Proof.* It follows directly from (5.2), (5.3), and the definition of the matrix $L$. ∎

As will be shown later in Theorem 5.6, we find the $\bar{w} \in \mathcal{W}$ satisfying (5.26) based on the sequence $\{\bar{w}^k\}$. Now, we translate the assertion (4.1) in Lemma 4.1 into the form of $\bar{w}^k$.

LEMMA 5.3. *Let $\tilde{w}^k$ be the output of the splitting step* (3.1) *with given $w^k$ and the vector $\bar{w}^k$ be defined by* (5.2). *Then, we have*

$$\bar{w}^k \in \mathcal{W}, \quad \theta(x) - \theta(\bar{x}^k) + (w - \bar{w}^k)^T \{F(\bar{w}^k) + Q(\bar{w}^k - w^k)\} \geq 0 \quad \forall\, w \in \mathcal{W}, \tag{5.9}$$

*where $Q$ is defined in* (5.4).

*Proof.* By using (5.2), we have $\tilde{x}_i = \bar{x}_i,\ \ i = 1, \ldots, m$, and

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

Substituting it into the variational inequality (4.3), we get

$$\begin{aligned} &\bar{x}_i^k \in \mathcal{X}_i,\ \theta_i(x_i) - \theta_i(\bar{x}_i^k) \\ &\quad + (x_i - \bar{x}_i^k)^T \left\{ -A_i^T \bar{\lambda}^k + A_i^T H \left( \sum_{j=1}^{m} A_j(\bar{x}_j^k - x_j^k) \right) + A_i^T H A_i(\bar{x}_i^k - x_i^k) \right\} \geq 0 \end{aligned} \tag{5.10}$$

for all $x_i \in \mathcal{X}_i$. Summing the above inequality over $i = 1, \ldots, m$, we obtain $\bar{w}^k \in \mathcal{W}$ and

$$\begin{aligned} \theta(x) - \theta(\bar{x}^k) + \begin{pmatrix} x_1 - \bar{x}_1^k \\ \vdots \\ x_i - \bar{x}_i^k \\ \vdots \\ x_m - \bar{x}_m^k \end{pmatrix}^T & \left\{ \begin{pmatrix} -A_1^T \bar{\lambda}^k \\ \vdots \\ -A_i^T \bar{\lambda}^k \\ \vdots \\ -A_m^T \bar{\lambda}^k \end{pmatrix} + \begin{pmatrix} A_1^T H\left(\sum_{j=1}^m A_j(\bar{x}_j^k - x_j^k)\right) \\ \vdots \\ A_i^T H\left(\sum_{j=1}^m A_j(\bar{x}_j^k - x_j^k)\right) \\ \vdots \\ A_m^T H\left(\sum_{j=1}^m A_j(\bar{x}_j^k - x_j^k)\right) \end{pmatrix} \right. \\ & \left. + \begin{pmatrix} A_1^T H A_1(\bar{x}_1^k - x_1^k) \\ \vdots \\ A_i^T H A_i(\bar{x}_i^k - x_i^k) \\ \vdots \\ A_m^T H A_m(\bar{x}_m^k - x_m^k) \end{pmatrix} \right\} \geq 0 \end{aligned} \tag{5.11}$$

for all $x \in \mathcal{X}$. Moreover, because (see (5.2))

$$\bar{\lambda}^k - \tilde{\lambda}^k - 2H \sum_{j=1}^{m} A_j(\bar{x}_j^k - x_j^k) = 0$$

and (due to (3.1) using $\tilde{x}_i = \bar{x}_i,\ \ i = 1, \ldots, m$)

$$\tilde{\lambda}^k = \lambda^k - H \left( \sum_{j=1}^{m} A_j \bar{x}_j^k - b \right),$$

we have

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

The above equation can be written in variational form

$$\bar{\lambda}^k \in \Re^l, \quad (\lambda - \bar{\lambda}^k)^T \left\{ \left(\sum_{j=1}^{m} A_j\bar{x}_j^k - b\right) - 2\sum_{j=1}^{m} A_j(\bar{x}_j^k - x_j^k) + H^{-1}(\bar{\lambda}^k - \lambda^k) \right\} \geq 0 \quad \forall\, \lambda \in \Re^l. \tag{5.12}$$

Combining (5.11) and (5.12), using the notation of $F$ (see (2.2b)) and $Q$ (see (5.4)), we get a compact form

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

The assertion (5.9) thus is proved. ▯

The assertion of the next lemma will be used in the proof of Lemma 5.5 which is essential for establishing the worst-case $O(1/k)$ convergence rate in the ergodic sense.

LEMMA 5.4. *Let $\tilde{w}^k$ be the output of the splitting step* (3.1) *with given $w^k$ and the vector $\bar{w}^k$ be defined by* (5.2)*. Then we have*

$$\|w^k - \bar{w}^k\|_G^2 - \|w^{k+1} - \bar{w}^k\|_G^2 > 0. \tag{5.13}$$

*Proof*. By using $w^{k+1} = w^k - \alpha(w^k - \tilde{w}^k)$, we obtain

$$\begin{aligned}\|w^k - \bar{w}^k\|_G^2 - \|w^{k+1} - \bar{w}^k\|_G^2 &= \|w^k - \bar{w}^k\|_G^2 - \|w^k - \bar{w}^k - \alpha(w^k - \tilde{w}^k)\|_G^2 \\ &= 2\alpha(w^k - \bar{w}^k)^T G(w^k - \tilde{w}^k) - \alpha^2\|w^k - \tilde{w}^k\|_G^2.\end{aligned}$$

Since $w^k - \bar{w}^k = L(w^k - \tilde{w}^k)$ (see (5.8)), from the above equation it follows that

$$\|w^k - \bar{w}^k\|_G^2 - \|w^{k+1} - \bar{w}^k\|_G^2 = 2\alpha(w^k - \tilde{w}^k)^T L^T G(w^k - \tilde{w}^k) - \alpha^2\|w^k - \tilde{w}^k\|_G^2. \tag{5.14}$$

By a manipulation (see $L$ in (5.5) and $G$ in (2.7)), we have

$$L^T G = \begin{pmatrix} 2A_1^T H A_1 & A_1^T H A_2 & \cdots & A_1^T H A_m & 2A_1^T \\ A_2^T H A_1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & A_{m-1}^T H A_m & 2A_{m-1}^T \\ A_m^T H A_1 & \cdots & A_m^T H A_{m-1} & 2A_m^T H A_m & 2A_m^T \\ 0 & \cdots & \cdots & 0 & H^{-1} \end{pmatrix}. \tag{5.15}$$

And thus

$$\begin{aligned}(w^k - \tilde{w}^k)^T L^T G(w^k - \tilde{w}^k) &= \|w^k - \tilde{w}^k\|_G^2 + 2(\lambda^k - \tilde{\lambda}^k)^T \left(\sum_{i=1}^{m} A_i(x_i^k - \tilde{x}_i^k)\right) \\ &= \varphi(w^k, \tilde{w}^k) \qquad (\text{see } \varphi(w^k, \tilde{w}^k) \text{ in (3.2c)}).\end{aligned} \tag{5.16}$$

Substituting (5.16) into (5.14) and using (4.17), we get

$$\|w^k - \bar{w}^k\|_G^2 - \|w^{k+1} - \bar{w}^k\|_G^2 = 2\alpha\varphi(w^k, \tilde{w}^k) - \alpha^2\|w^k - \tilde{w}^k\|_G^2 = q(\alpha).$$

By using each update form, we always have $q(\alpha_k) > 0$ and thus (5.13) is proved. ∎

To establish the worst-case $O(1/k)$ convergence rate in the ergodic sense, we need to prove one more lemma.

LEMMA 5.5. *Let $\tilde{w}^k$ be the output of the splitting step* (3.1) *with given $w^k$ and the vector $\bar{w}^k$ be defined by* (5.2). *Then, we have*

$$\alpha_k\big\{\theta(x)-\theta(\bar{x}^k)+(w-\bar{w}^k)^T F(w)\big\} \geq \frac{1}{2}\big(\|w-w^{k+1}\|_G^2-\|w-w^k\|_G^2\big) \ \ \forall w \in \mathcal{W}. \tag{5.17}$$

*Proof.* The assertions (5.17) can be obtained based on the following facts:

(1) Using Lemma 5.3 and the fact $(w - \bar{w}^k)^T F(w) \geq (w - \bar{w}^k)^T F(\bar{w}^k)$, we have

$$\alpha_k\big\{\big(\theta(x) - \theta(\bar{x}^k)\big) + (w - \bar{w}^k)^T F(w)\big\} \geq \alpha_k (w - \bar{w}^k)^T Q(w^k - \bar{w}^k) \quad \forall w \in \mathcal{W}. \tag{5.18}$$

(2) For the right-hand side of (5.18), using Lemmas 5.1 and 5.2, we have

$$(w^k - \bar{w}^k) = L(w^k - \tilde{w}^k) \qquad \text{and} \qquad QL = G.$$

Together with $\alpha_k(w^k - \tilde{w}^k) = (w^k - w^{k+1})$, we obtain

$$\alpha_k(w - \bar{w}^k)^T Q(w^k - \bar{w}^k) = (w - \bar{w}^k)^T G(w^k - w^{k+1}). \tag{5.19}$$

(3) Set $a = w$, $h = \bar{w}^k$ $g = w^k$, and $r = w^{k+1}$ in the identity

$$(a-h)^T G(g-r) = \frac{1}{2}\big(\|a-r\|_G^2 - \|a-g\|_G^2\big) + \frac{1}{2}\big(\|g-h\|_G^2 - \|r-h\|_G^2\big);$$

the right-hand side of (5.19) becomes

$$\begin{aligned} &\big(w - \bar{w}^k\big)^T G\big(w^k - w^{k+1}\big) \\ &\quad = \frac{1}{2}\big(\|w - w^{k+1}\|_G^2 - \|w - w^k\|_G^2\big) \\ &\qquad + \frac{1}{2}\big(\|w^k - \bar{w}^k\|_G^2 - \|w^{k+1} - \bar{w}^k\|_G^2\big). \end{aligned} \tag{5.20}$$

Combining (5.18), (5.19), and (5.20), we obtain

$$\begin{aligned} &\alpha_k\big\{\big(\theta(x) - \theta(\bar{x}^k)\big) + (w - \bar{w}^k)^T F(w)\big\} \\ &\quad \geq \frac{1}{2}\big(\|w - w^{k+1}\|_G^2 - \|w - w^k\|_G^2\big) + \frac{1}{2}\big(\|w^k - \bar{w}^k\|_G^2 - \|w^{k+1} - \bar{w}^k\|_G^2\big). \end{aligned} \tag{5.21}$$

The assertion (5.17) follows from (5.21) and (5.13) immediately. The proof is complete. ∎

Now, we are ready to show a worst-case $O(1/k)$ convergence rate in the ergodic sense (more precisely, in the uniformly weighted average sense) for the proposed algorithms.

THEOREM 5.6. *Let $\{w^k\}$ be the sequence generated by Algorithm* 1 *or* 2, *and the accompanying sequence $\{\bar{w}^k\}$ be defined by* (5.2). *For any integer $k > 0$, let*

$$\bar{w}_k := \frac{1}{\Upsilon_k}\sum_{i=0}^{k} \alpha_i \bar{w}^i \qquad \textit{with} \qquad \Upsilon_k = \sum_{i=0}^{k} \alpha_i. \tag{5.22}$$

*Then, we have* $\bar{w}_k \in \mathcal{W}$ *and*

$$\theta(\bar{x}_k) - \theta(x) + (\bar{w}_k - w)^T F(w) \le \frac{m+1}{\gamma(k+1)} \|w - w^0\|_G^2 \quad \forall w \in \mathcal{W}. \tag{5.23}$$

*Proof.* Note that the the inequality (5.17) holds for $i = 0, 1, \dots, k$. Summarizing these inequalities, we obtain

$$\left(\Upsilon_k \theta(x) - \sum_{i=0}^{k} \alpha_i \theta(\bar{x}^i)\right) + \left(\Upsilon_k w - \sum_{i=0}^{k} \alpha_i \bar{w}^i\right)^T F(w) \ge -\frac{1}{2}\|w - w^0\|_G^2 \quad \forall w \in \mathcal{W},$$

which implies that

$$\left(\frac{1}{\Upsilon_k}\sum_{i=0}^{k} \alpha_i \theta(\bar{x}^i) - \theta(x)\right) + \left(\frac{1}{\Upsilon_k}\sum_{i=0}^{k} \alpha_i \bar{w}^i - w\right)^T F(w) \le \frac{1}{2\Upsilon_k}\|w - w^0\|_G^2 \quad \forall w \in \mathcal{W}. \tag{5.24}$$

Since $\bar{x}_k := \frac{1}{\Upsilon_k}\sum_{i=0}^{k} \alpha_i \bar{x}^i$ is a convex combination of the vectors $(\bar{x}^0, \bar{x}^1, \dots, \bar{x}^k)$ and $\theta(x)$ is convex, we have

$$\theta(\bar{x}_k) \le \frac{1}{\Upsilon_k}\sum_{i=0}^{k} \alpha_i \theta(\bar{x}^i).$$

Substituting it into (5.24), we obtain

$$\theta(\bar{x}_k) - \theta(x) + (\bar{w}_k - w)^T F(w) \le \frac{1}{2\Upsilon_k}\|w - w^0\|_G^2 \quad \forall w \in \mathcal{W}. \tag{5.25}$$

Recall that we have shown (see (3.5) and (4.20)) that

$$\alpha_i \ge \frac{\gamma}{2(m+1)}$$

holds for any integer $i$. Using this fact in (5.22), we get

$$\Upsilon_k \ge (k+1)\frac{\gamma}{2(m+1)}$$

and thus

$$\frac{1}{\Upsilon_k} \le \frac{2(m+1)}{\gamma(k+1)}.$$

Substituting it into (5.25), we obtain the assertion (5.23). The proof is complete. ∎

For given substantial compact set $D_{\mathcal{W}}(\bar{w}_k) \subset \mathcal{W}$, we define

$$d = \sup\{\|w - w^0\|_G^2 \mid w \in D_{\mathcal{W}}(\bar{w}_k)\},$$

where $w^0$ is the initial point. Based on Theorem 5.6, after $k$ iterations of Algorithm 1 or 2, we can find $\bar{w}_k \in \mathcal{W}$ such that

$$\sup_{\forall w \in D_{\mathcal{W}}(\bar{w}_k)} \left\{\theta(\bar{x}_k) - \theta(x)\big) + (\bar{w}_k - w)^T F(w)\right\} \le \frac{1}{k+1}\left(\frac{(m+1)d}{\gamma}\right). \tag{5.26}$$

Recall (5.26). The proposed Algorithm 1 or 2 is able to generate an approximate solution (i.e., $\bar{w}_t$) with the accuracy $O(1/k)$ after $k$ iterations. That is, a worst-case $O(1/k)$ convergence rate in the ergodic (uniformly weighted average) sense is established for Algorithms 1 and 2.

**6. Convergence rate in a nonergodic sense.** In section 5, a worst-case $O(1/k)$ convergence rate in the ergodic sense is established for Algorithms 1 and 2. One may ask if we can establish the same convergence rate in some nonergodic sense, i.e, directly for the sequence $\{w^k\}$ generated by the proposed algorithms. This section answers this question affirmatively. The technique of analysis is motivated by our work [22]. As stated in [22], a necessary fact for conducting this analysis is that the quantity $\|w^k - w^{k+1}\|_G^2$ can be used to measure the accuracy of the iterate $w^{k+1}$ to a solution point of VI$(\mathcal{W}, F, \theta)$ (see the VI characterization (2.2) and Lemma 4.1), and it is reasonable to seek an upper bound of $\|w^k - w^{k+1}\|_G^2$ in term of the quantity $O(1/k)$ to investigate the worst-case convergence rate for the proposed algorithms.

Recall we have shown that the sequence generated by either Algorithm 1 or 2 is strictly contractive with respect to the set $\mathcal{W}^*$ (see (4.21)):

$$\|w^{k+1} - w^*\|_G^2 \le \|w^k - w^*\|_G^2 - \tau\|w^k - \tilde{w}^k\|_G^2 \ \ \forall\, w^* \in \mathcal{W}^*, \tag{6.1}$$

where

$$\tau = \gamma(2-\gamma)\left(1 - \sqrt{\frac{m}{m+1}}\right)^2 > 0 \tag{6.2}$$

with $\gamma \in (0,2)$. To establish the worst-case $O(1/k)$ convergence rate in a nonergodic sense, we need to show that the sequence $\{\|w^k - \tilde{w}^k\|_G\}$ is monotonically nonincreasing. The basis of the analysis in this section is the assertion of Lemma 5.3.

LEMMA 6.1. *Let $\{w^k\}$ be the sequence generated by Algorithm* 1 *or* 2, *and the accompanying sequence $\{\bar{w}^k\}$ be defined by* (5.2). *Then we have*

$$(w^k - w^{k+1})^T Q\big((w^k - \bar{w}^k) - (w^{k+1} - \bar{w}^{k+1})\big) \ge \frac{1}{2}\|(w^k - \bar{w}^k) - (w^{k+1} - \bar{w}^{k+1})\|_{(Q^T+Q)}^2, \tag{6.3}$$

*where $Q$ is defined in* (5.4).

*Proof.* First, it follows from (5.9) that

$$\bar{w}^k \in \mathcal{W}, \quad \theta(x) - \theta(\bar{x}^k) + (w - \bar{w}^k)^T F(\bar{w}^k) \ge (w - \bar{w}^k)^T Q(w^k - \bar{w}^k) \quad \forall\, w \in \mathcal{W}. \tag{6.4}$$

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

$$\begin{aligned} \bar{w}^{k+1} \in \mathcal{W}, \quad & \theta(x) - \theta(\bar{x}^{k+1}) + (w - \bar{w}^{k+1})^T F(\bar{w}^{k+1}) \\ & \qquad \ge (w - \bar{w}^{k+1})^T Q(w^{k+1} - \bar{w}^{k+1}) \ \ \forall\, w \in \mathcal{W}. \end{aligned} \tag{6.5}$$

Setting $w = \bar{w}^{k+1}$ and $w = \bar{w}^k$ in (6.4) and (6.5), respectively, and then adding these two resulting inequalities, we obtain

$$(\bar{w}^k - \bar{w}^{k+1})^T Q\big((w^k - \bar{w}^k) - (w^{k+1} - \bar{w}^{k+1})\big) \ge (\bar{w}^k - \bar{w}^{k+1})^T (F(\bar{w}^k) - F(\bar{w}^{k+1})).$$

Using the monotonicity of $F$, we have

$$(\bar{w}^k - \bar{w}^{k+1})^T Q\big((w^k - \bar{w}^k) - (w^{k+1} - \bar{w}^{k+1})\big) \ge 0. \tag{6.6}$$

Adding the identity

$$\begin{aligned} &\big((w^k - \bar{w}^k) - (w^{k+1} - \bar{w}^{k+1})\big)^T Q\big((w^k - \bar{w}^k) - (w^{k+1} - \bar{w}^{k+1})\big) \\ &\quad = \frac{1}{2}\|(w^k - \bar{w}^k) - (w^{k+1} - \bar{w}^{k+1})\|_{(Q^T+Q)}^2 \end{aligned}$$

to both sides of (6.6) and by a simple manipulation, we get (6.3) and the lemma is proved. ∎

Using the assertion (6.3) in Lemma 6.1, it is now possible to make an estimate for the difference $\|w^k - \tilde{w}^k\|_G^2 - \|w^{k+1} - \tilde{w}^{k+1}\|_G^2$.

LEMMA 6.2. *Let $\{w^k\}$ be the sequence generated by Algorithm 1 or 2 with a constant step size $\alpha_k \equiv \alpha > 0$. Then we have*

$$\begin{aligned}\|w^k - \tilde{w}^k\|_G^2 &- \|w^{k+1} - \tilde{w}^{k+1}\|_G^2 \\ &\geq \frac{1}{\alpha}\|L[(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})]\|^2_{(Q^T+Q)} - \|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_G^2.\end{aligned} \tag{6.7}$$

*Proof.* By using $(w^k - \bar{w}^k) = L(w^k - \tilde{w}^k)$ (see (5.8)) in (6.3), we get

$$\begin{aligned}&2(w^k - w^{k+1})^T QL\big((w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\big) \\ &\quad \geq \|L[(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})]\|^2_{(Q^T+Q)}.\end{aligned} \tag{6.8}$$

Using the relations (see (3.3) and (5.7))

$$w^k - w^{k+1} = \alpha(w^k - \tilde{w}^k) \qquad \text{and} \qquad QL = G$$

in (6.8), we have that

$$2(w^k-\tilde{w}^k)^T G\big((w^k-\tilde{w}^k)-(w^{k+1}-\tilde{w}^{k+1})\big) \geq \frac{1}{\alpha}\|L[(w^k-\tilde{w}^k)-(w^{k+1}-\tilde{w}^{k+1})]\|^2_{(Q^T+Q)}. \tag{6.9}$$

On the other hand, by setting $h = (w^k - \tilde{w}^k)$ and $g = (w^{k+1} - \tilde{w}^{k+1})$ in the identity

$$\|h\|_G^2 - \|g\|_G^2 = 2h^T G(h - g) - \|h - g\|_G^2,$$

we have

$$\begin{aligned}\|w^k - \tilde{w}^k\|_G^2 &- \|w^{k+1} - \tilde{w}^{k+1}\|_G^2 \\ &= 2(w^k - \tilde{w}^k)^T G\big((w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\big) - \|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_G^2.\end{aligned} \tag{6.10}$$

Substituting (6.9) into the right-hand side of (6.10), the assertion (6.7) is proved. ∎

In order to show the monotonicity of $\{\|w^k - \tilde{w}^k\|_G\}$, we need only to show the right-hand side of (6.7) is nonnegative. Thus, we prove the following lemma.

LEMMA 6.3. *For the given matrices $G$, $Q$, $L$ and any constant $\alpha \leq 2(1-\sqrt{\frac{m}{m+1}})$, we have*

$$L^T(Q^T + Q)L - \alpha G \succeq 0. \tag{6.11}$$

*Proof.* Since $QL = G$ (see (5.7)) and $G$ is symmetric, we have

$$L^T(Q^T + Q)L = GL + L^T G.$$

Using the above equation and the expression of $L^TG$ (see (5.15)), it yields

(6.12)
$$L^T(Q^T+Q)L = 2\begin{pmatrix} 2A_1^THA_1 & A_1^THA_2 & \cdots & A_1^THA_m & A_1^T \\ A_2^THA_1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & A_{m-1}^THA_m & A_{m-1}^T \\ A_m^THA_1 & \cdots & A_m^THA_{m-1} & 2A_m^THA_m & A_m^T \\ A_1 & \cdots & A_{m-1} & A_m & H^{-1} \end{pmatrix}.$$

Furthermore, using the notation

$$\mathcal{A} = \mathrm{diag}(H^{1/2}A_1, \dots, H^{1/2}A_m, H^{-1/2})$$

and the expression of $G$ (see (2.7)), we have

(6.13)
$$\begin{aligned} &L^T(Q^T+Q)L - \alpha G \\ &= \mathcal{A}^T\left\{ 2\begin{pmatrix} 2I_l & I_l & \cdots & I_l & I_l \\ I_l & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & I_l & \vdots \\ I_l & \cdots & I_l & 2I_l & I_l \\ I_l & \cdots & \cdots & I_l & I_l \end{pmatrix} \right. \\ &\qquad \left. - \alpha \begin{pmatrix} 2I_l & I_l & \cdots & I_l & 0 \\ I_l & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & I_l & \vdots \\ I_l & \cdots & I_l & 2I_l & 0 \\ 0 & \cdots & \cdots & 0 & I_l \end{pmatrix} \right\}\mathcal{A}_{(m+1)\times(m+1)}. \end{aligned}$$

In this way, in order to show (6.11), we need only to prove that the $(m+1)\times(m+1)$ symmetric matrix

$$T = 2\begin{pmatrix} I_m + ee^T & e \\ e^T & 1 \end{pmatrix} - \alpha \begin{pmatrix} I_m + ee^T & 0 \\ 0 & 1 \end{pmatrix} \succeq 0,$$

where $e$ is an $m$-vector whose each element equals 1. Let $Tz = \nu z$, where $\nu$ is the eigenvalue of $T$ and $z$ is the related eigenvector. In the following we show that all the eigenvalues of $T$ are nonnegative. Note that

(6.14)
$$T = \begin{pmatrix} (2-\alpha)(I_m + ee^T) & 2e \\ 2e^T & 2-\alpha \end{pmatrix}.$$

Without loss of generality, we assume that the eigenvectors of $T$ have the form

$$z = \begin{pmatrix} y \\ 0 \end{pmatrix} \qquad \text{or} \qquad z = \begin{pmatrix} y \\ 1 \end{pmatrix},$$

where $y \in \Re^m$. In the first case, $z^T = (y^T, 0)$, it follows from $Tz = \nu z$ and (6.14) that

$$\begin{cases} (2-\alpha)y + (2-\alpha)(e^T y)e = \nu y, \\ e^T y = 0. \end{cases} \quad \Rightarrow \quad \begin{cases} (2-\alpha)y = \nu y, \\ e^T y = 0. \end{cases}$$

Therefore, we have $(m-1)$ linear independent vectors, $y^i,\ i = 1, \ldots, m-1$, in the orthogonal subspace to $e$, and

$$z^i = \begin{pmatrix} y^i \\ 0 \end{pmatrix}, \quad i = 1, \ldots, m-1,$$

are eigenvectors of $T$ and the related eigenvalues are

$$\nu_1 = \nu_2 = \cdots = \nu_{m-1} = (2-\alpha) > 0 \quad \left(\text{due to } 0 < \alpha \le 2\left(1 - \sqrt{\tfrac{m}{m+1}}\right)\right).$$

In the second case, $z^T = (y^T, 1)$, from $Tz = \nu z$ and (6.14) we have

$$\begin{cases} (2-\alpha)y + ((2-\alpha)e^T y + 2)e = \nu y, \\ 2e^T y + (2-\alpha) = \nu. \end{cases} \tag{6.15}$$

Left multiplying the first equation of (6.15) by $e^T$ and then using the second equation of (6.15) and $e^T e = m$, we derive that

$$\nu^2 - (m+2)(2-\alpha)\nu + [(m+1)(2-\alpha)^2 - 4m] = 0.$$

The remaining two eigenvalues of $T$ are the roots of the above equation and thus

$$\nu(T) = \frac{(m+2)(2-\alpha) \pm \sqrt{(m+2)^2(2-\alpha)^2 - 4[(m+1)(2-\alpha)^2 - 4m]}}{2}.$$

Since $\alpha \le 2\big(1 - \sqrt{\tfrac{m}{m+1}}\big)$, we have

$$0 \le 4[(m+1)(2-\alpha)^2 - 4m] \le (m+2)^2(2-\alpha)^2,$$

and thus $\nu_{\min}(T) \ge 0$. All the eigenvalues of $T$ are nonnegative and the lemma is proved. ∎

Therefore, the monotonicity of $\{\|w^k - \tilde{w}^k\|_G\}$ is a straightforward consequence of Lemmas 6.2 and 6.3. Now, we are ready to estimate the worst-case $O(1/k)$ convergence rate in a nonergodic sense.

THEOREM 6.4. *Let $\{w^k\}$ be the sequence generated by Algorithm 1 or 2 with the requirement on step size*

$$\alpha_k \le 2\left(1 - \sqrt{\tfrac{m}{m+1}}\right) \tag{6.16}$$

*for all k's. Then we have*

$$\left\|w^k - w^{k+1}\right\|_G^2 \le \frac{4}{\gamma(2-\gamma)(k+1)} \left\|w^0 - w^*\right\|_G^2 \quad \forall w^* \in \mathcal{W}^*. \tag{6.17}$$

*Proof.* First, we remark that the requirement (6.16) is satisfied for Algorithm 1 if its step size determined in (3.2b) is now set as

$$\alpha_k = \min\left\{\gamma\alpha_k^*, 2\left(1-\sqrt{\tfrac{m}{m+1}}\right)\right\},$$

and it always holds for Algorithm 2 (see (3.3b)). Then, it follows from (6.1) that

$$\tau\sum_{i=0}^{\infty}\|w^i-\tilde{w}^i\|_G^2 \le \|w^0-w^*\|_G^2 \quad \forall\, w^*\in\mathcal{W}^*. \tag{6.18}$$

Using (6.16), it follows from Lemmas 6.2 and 6.3 that the sequence $\{\|w^k-\tilde{w}^k\|_G^2\}$ is monotonically nonincreasing. Therefore, we have

$$(k+1)\|w^k-\tilde{w}^k\|_G^2 \le \sum_{i=0}^{k}\|w^i-\tilde{w}^i\|_G^2. \tag{6.19}$$

It follows from (6.18) and (6.19) that

$$\left\|w^k-\tilde{w}^k\right\|_G^2 \le \frac{1}{\tau(k+1)}\left\|w^0-w^*\right\|_G^2 \quad \forall w^*\in W^*.$$

Since $w^k-w^{k+1}=\alpha_k(w^k-\tilde{w}^k)$ and $\alpha_k\le 2\left(1-\sqrt{\frac{m}{m+1}}\right)$, we have

$$\left\|w^k-w^{k+1}\right\|_G^2 \le \frac{\alpha^2}{\tau(k+1)}\left\|w^0-w^*\right\|_G^2 \le \frac{4}{\gamma(2-\gamma)(k+1)}\left\|w^0-w^*\right\|_G^2 \quad \forall w^*\in W^*.$$

The assertion (6.17) is proved, The proof is complete. ∎

Notice that $\mathcal{W}^*$ is convex and closed. Let $d:=\sup\{\|w^0-w^*\|_G^2 \mid w^*\in\mathcal{W}^*\}$. Then, after $(k+1)$ iterations of Algorithm 1 or 2, we have $\|w^k-w^{k+1}\|_G^2\le\frac{4d}{\gamma(2-\gamma)}\cdot\frac{1}{k+1}=O(1/k)$. Since $w^{k+1}$ is a solution point of VI$(\mathcal{W},F,\theta)$ if $\|w^k-w^{k+1}\|_G^2=0$, the worst-case $O(1/k)$ convergence rate in a nonergodic sense for Algorithms 1 and 2 is established in Theorem 6.4.

**7. Numerical results.** In this section, we report some numerical results to show the efficiency of the proposed algorithms. In the literature, splitting versions of ALM with full Jacobian decomposition have been well tested by various examples; see, e.g., [16, 25] and especially in [16] for a number of applications in image processing. The efficiency of the proposed new splitting version of ALM with a refined step size thus can be easily demonstrated by this type of example. Here, we further illustrate the efficiency of this type of algorithm from a different perspective. Note that an extreme case of (1.3) is a linear programming (LP) model where $n_i=1$ for all $i$, each $\theta_i$ is the product of a constant in $\Re^1$ with $x_i$, and each $A_i$ is a column. We thus can artificially treat a linear program as a very special case of (1.3) and solve it by the proposed splitting versions of ALM with full Jacobian decomposition. We will test two different LP cases: first, an assignment problem which can be expressed as a structured linear program but with specific coefficient matrices in its constraints. In fact, these matrices result in extremely easy subproblems when the proposed scheme is applied. Second, we test a series of LP models whose coefficient matrices are ill-conditioned. For these LP models, the resulting subproblems when the proposed algorithms are applied are general and they do not have any specific structure as the mentioned assignment problem. Moreover, we test an $l_1$-norm model which has a wide range of applications in areas such as compressive sensing, signal processing, information science, and so on.

We would mention that our purpose of testing these examples is verifying the fact that the combination of a relaxation step with the full Jacobian decomposition of the ALM, where step size of the relaxation step is refined and the Jacobian decomposition extent could be huge, works and, numerically, tackling a separable convex minimization model with many functions in its objective. Thus, our algorithmic-design theory of ensuring the convergence by a very simple relaxation step even when the ALM is splitted many times can be well verified. We do not advocate that the proposed algorithms can beat the state-of-the-art in LP literature, even though they also work well for the tested applications. Indeed, for the assignment problem, the proposed algorithms are even competitive with the well-commercialized specific LP solver "IBM ILOG CPLEX Optimization Studio" (CPLEX). But the new algorithms are proposed in a generic setting and they are applicable for nonlinear and nonsmooth problems, not just for LP.

Our code was written by MATLAB R2014a and all our experiments were performed on a desktop with the Windows 7 system, Intel(R) Core(TM)2 Quad CPU processor (2.66GHz), and 4 GB memory.

**7.1. Implementation of Algorithms 1 and 2 in an LP context.** We first show the implementation of the proposed algorithms to linear programs. Let us consider the following LP model with box constraints:

$$\begin{array}{rl} \max & c^T x \\ \text{s.t.} & Ax = b, \\ & x^l \le x \le x^u, \end{array} \tag{7.1}$$

where $A \in \Re^{l\times m}$, $c \in \Re^m$, $b \in \Re^l$ $(l < m)$, $x^l \in \Re^m$, and $x^u \in \Re^m$ with $x_i^l \le x_i^u$ for $i = 1, 2, \dots, m$. This LP model (7.1) is a special case of (1.3), where $n_i \equiv 1$, $\theta_i(x_i) = -c_i x_i$, $A_i \in \Re^l$ for $i = 1, 2, \dots, m$, and $\mathcal{X}_i = [x_i^l, x_i^u]$.

To implement the proposed algorithms, we set $H = \beta I_l$ throughout, where $\beta > 0$ is a scalar and $I_l$ is the identity matrix with dimensionality $l$. We first look at the $i$th subproblem of the splitting step (3.1), which is a 1-dimensional minimization problem. For solving (7.1), the $x_i$-subproblem in (3.1) reduces to

$$\tilde{x}_i^k = \arg\min \left\{ -c_i x_i - (\lambda^k)^T A_i x_i + \frac{\beta}{2} \left\| \sum_{j=1}^{i-1} A_j x_j^k + A_i x_i + \sum_{j=i+1}^{m} A_j x_j^k - b \right\|^2 \ \Big|\ x_i \in \mathcal{X}_i \right\},$$

which, by its first-order optimality condition, can be further expressed as

$$\tilde{x}_i^k \in \mathcal{X}_i, \quad (x_i - \tilde{x}_i^k) \left\{ -c_i - A_i^T \lambda^k + A_i^T \beta \left( A_i \tilde{x}_i^k + \sum_{j=1,\, j\neq i}^{m} A_j x_j^k - b \right) \right\} \ge 0 \quad \forall\, x_i \in \mathcal{X}_i.$$

Recall $A_i \in \Re^l$ for $i = 1, 2, \ldots, m$. Thus, $\beta A_i^T A_i$ is a positive scalar and the above inequality can also be written as

$$\tilde{x}_i^k \in \mathcal{X}_i, \ \ (x_i - \tilde{x}_i^k)\left\{(\tilde{x}_i^k - x_i^k) + \frac{1}{A_i^T A_i}\left[-c_i/\beta + A_i^T\left(\left(\sum_{j=1}^m A_j x_j^k - b\right) - \lambda^k/\beta\right)\right]\right\} \geq 0 \ \ \forall x_i \in \mathcal{X}_i,$$

which can be characterized by a projection equation:

$$\tilde{x}_i^k = P_{\mathcal{X}_i}\left\{x_i^k - \frac{1}{A_i^T A_i}\left[-c_i/\beta + A_i^T\left(\left(\sum_{j=1}^m A_j x_j^k - b\right) - \lambda^k/\beta\right)\right]\right\}, \quad i = 1, \ldots, m,$$

where $P_{\mathcal{X}_i}$ denotes the projection under Euclidean norm onto $\mathcal{X}_i$. Note $\mathcal{X}_i = [x_i^l, x_i^u]$ is a given interval. We thus have

$$P_{\mathcal{X}_i}(\xi) = \min\{\max\{\xi, x_i^l\}, x_i^u\} \ \ \forall \xi \in \Re.$$

Let $D = \mathrm{diag}(A^T A)$. Then, based on the analysis above, all the $x_i$-subproblems can be written compactly as

$$\tilde{x}^k = P_{\mathcal{X}}\{x^k - D^{-1}[-c/\beta + A^T((Ax^k - b) - \lambda^k/\beta)]\}, \tag{7.2}$$

where $x = (x_1, x_2, \ldots, x_m)$ and $A = (A_1, A_2, \ldots, A_m)$. In addition, the $\lambda$-subproblem in (3.1) is specified as

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k - b). \tag{7.3}$$

To compute the step size of the relaxation step (3.2b) in Algorithm 1, we have

$$\|w^k - \tilde{w}^k\|_G^2 = \beta(\|A * \mathrm{diag}(x^k - \tilde{x}^k)\|_F^2 + \|A(x^k - \tilde{x}^k)\|^2) + \|\lambda^k - \tilde{\lambda}^k\|^2/\beta \tag{7.4a}$$

and

$$\varphi(w^k, \tilde{w}^k) = \|w^k - \tilde{w}^k\|_G^2 + 2(\lambda^k - \tilde{\lambda}^k)^T A(x^k - \tilde{x}^k), \tag{7.4b}$$

where "diag" denotes the MATLAB script. Thus, it can be easily computed. For the constant step size of the relaxation step (3.3b) in Algorithm 2, it is also easy.

**7.2. Application to the assignment problem.** We cite from Wikipedia, "The assignment problem is one of the fundamental combinatorial optimization problems in the branch of optimization or operations research in mathematics. It consists of finding a maximum weight matching in a weighted bipartite graph."

**7.2.1. Specifications.** Let us consider the LP model

$$\begin{array}{ll} \max & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}, \\ & \sum_{j=1}^n x_{ij} = 1, \ \ i = 1, \ldots, n, \\ & \sum_{i=1}^n x_{ij} = 1, \ \ j = 1, \ldots, n, \\ & 0 \leq x_{ij} \leq 1. \end{array} \tag{7.5}$$

The classical assignment problem is recovered if we replace $0 \le x_{ij} \le 1$ by $x_{ij} \in \{0,1\}$ in (7.5). This is a special case of (7.1), and thus (1.3) with $l = 2n$ and $m = n^2$. We choose this model for simulation because it is a well-structured problem (see the matrix $A$ in (7.6)). Thus, splitting versions of ALM are easily implementable because the subproblems can be solved trivially.

For simulation, we use the standard MATLAB scripts "`rand('state',0)}`" and "`C=rand(n,n)*10`" to produce the assignment matrix $C$ for (7.5),

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix}.$$

Thus, entries of $C$ are randomly generated in the interval $(0, 10)$.

To cast (7.5) into the setting of (7.1), we have an $n^2$-dimensional vector

$$x^T = (x_{11}, x_{12}, \ldots, x_{1n}, x_{21}, x_{22}, \ldots, x_{2n}, \ldots, x_{n1}, x_{n2}, \ldots, x_{nn}).$$

Moreover, the matrix $A$ and the vector $b$ in (7.1) are specified as

$$A = \begin{pmatrix} 1 & 1 & \cdots & 1 & & & & & & & & & \\ & & & & 1 & 1 & \cdots & 1 & & & & & \\ & & & & & & & & \cdots\cdots\cdots & & & & \\ & & & & & & & & & 1 & 1 & \cdots & 1 \\ 1 & & & & 1 & & & & & 1 & & & \\ & 1 & & & & 1 & & & & & 1 & & \\ & & \ddots & & & & \ddots & & \cdots\cdots\cdots & & & \ddots & \\ & & & 1 & & & & 1 & & & & & 1 \end{pmatrix}_{2n \times n^2} \quad \text{and}$$

$$b = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{7.6}$$

Thus, the Lagrangian multiplier $\lambda$ is a $2n$-vector $\lambda = (y, z)$, and $\operatorname{diag}(A^T A) = 2I_{n^2}$. For convenience, we regroup the $n^2$-vector $x$ in the matrix form

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{pmatrix},$$

then the $2n$-vector $(Ax - b)$ is given by

$$Ax - b = \begin{pmatrix} Xe - e \\ X^T e - e \end{pmatrix},$$

where $e$ is an $n$-vector whose each element equals 1.

For given $x^k$ and $\lambda^k = (y^k, z^k)$, the $n^2$-vector $A^T((Ax^k - b) - \lambda^k/\beta)$ of (7.2) is also regrouped in an $n \times n$ matrix

$$p^k * e^T + e * (q^k)^T,$$

where the $n$-vectors $p^k$ and $q^k$ are given by

$$p^k = (X^k)e - e - y^k/\beta \qquad \text{and} \qquad q^k = (X^k)^T e - e - z^k/\beta.$$

Finally, the predictor (7.2) is computed by

$$\tilde{X}^k = P_{\mathcal{X}}\left[X^k - \frac{1}{2}\left(-C/\beta + p^k * e^T + e * (q^k)^T\right)\right]$$

and

$$\begin{pmatrix} \tilde{y}^k \\ \tilde{z}^k \end{pmatrix} = \begin{pmatrix} y^k \\ z^k \end{pmatrix} - \beta \begin{pmatrix} (\tilde{X}^k)e - e \\ (\tilde{X}^k)^T e - e \end{pmatrix},$$

where $P_{\mathcal{X}}(\xi)$ is elementwise given by $\max\{\min\{\xi, 1\}, 0\}$.

To calculate the relaxation step (3.2b) in Algorithm 1, we need to calculate $\|w^k - \tilde{w}^k\|_G^2$ and $\varphi(w^k, \tilde{w}^k)$, which are available, respectively, by

$$\|w^k - \tilde{w}^k\|_G^2 = \beta\left(2\|X^k - \tilde{X}^k\|_F^2 + \left\|\begin{matrix} (X^k - \tilde{X}^k)e \\ (X^k - \tilde{X}^k)^T e \end{matrix}\right\|^2\right) + \frac{1}{\beta}\left\|\begin{matrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \end{matrix}\right\|^2, \tag{7.7a}$$

and

$$\varphi(w^k, \tilde{w}^k) = \|w^k - \tilde{w}^k\|_G^2 + 2\big(e^T(X^k - \tilde{X}^k)^T(y^k - \tilde{y}^k) + e^T(X^k - \tilde{X}^k)(z^k - \tilde{z}^k)\big). \tag{7.7b}$$

**7.2.2. Numerical results.** Now, we report some numerical results when Algorithms 1 and 2 are applied to solve the model (7.5). Throughout, we choose $\beta = 5/n$ (recall $H = \beta I_l$) and the initial iterate is taken as 0. The stopping criterion is

$$\max\{\|x^k - \tilde{x}^k\|_\infty, \|A\tilde{x}^k - b\|_\infty\} \le 10^{-8}, \tag{7.8}$$

and we take the final $\tilde{w}^k$ as the output solution.

We first use some small-scale cases of (7.5) to compare Algorithms 1 and 2. More specifically, we test the cases where $n = 3,\ 5$, and 10 in (7.5). Recall $l = 2n$ and $m = n^2$ in the setting of (7.1). For Algorithm 1, we fix $\gamma = 1$ and thus $\alpha_k = \alpha_k^*$. Recall Algorithm 1 requires us to choose an "optimal" step size in the sense of maximizing the quadratic function defined in (4.17) at each iteration, while Algorithm 2 simply chooses a constant step size for all iterations. According to (3.3), the constant step size for Algorithm 2 should be in the interval $(0, 2(1 - n\sqrt{1/(n^2+1)}\,))$. Nevertheless, this choice is too conservative and it can hardly be good enough to result in fast convergence. For example, $2(1 - n\sqrt{1/(n^2+1)}\,) \approx 0.0099$ when $n = 10$, which is extremely small. We thus test other constant step sizes including the $n$-dependent value $1/(n^2+1)$ and some more aggressive $n$-independent values (even though these values already exceed the theoretical upper bound of the range of step sizes). All the number of iterations of Algorithms 1 and 2 with different step sizes are reported in Table 1.

Data in Table 1 show that Algorithm 1 significantly outperforms Algorithm 2 with a constant step size determined by (3.3b). As we have mentioned, this is mainly because an optimal step size is sought judiciously for Algorithm 1 at each iteration and the cost of finding this step size is low for the assignment problem (7.5). Algorithm 2 with conservative constant step size converges very slowly, despite its proven theoretical convergence. On the other hand, Algorithm 2 with some selective excessive step sizes could be fast even though the convergence cannot be established rigorously

TABLE 1
*Comparison of Algorithms* 1 *and* 2 *for small-scale cases of* (7.5).

| $n$ | Algorithm 1 | Algorithm 2 | | Algorithm 2 with excessive step size | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\alpha_k = \alpha_k^*$ | $\alpha = \frac{1}{n^2+1}$ | $\alpha = 2\left(1-\sqrt{\frac{n^2}{n^2+1}}\right)$ | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.9 | 1.1 | 1.2 |
| 3 | 11 | 250 | 243 | 121 | 78 | 56 | 42 | 33 | 24 | 16 | 27 | – |
| 5 | 25 | 682 | 675 | 127 | 82 | 55 | 46 | 39 | – | – | – | – |
| 10 | 28 | 2929 | 2922 | 128 | 83 | 59 | 46 | 39 | – | – | – | – |

"–" means not convergent.

TABLE 2
*Comparison of Algorithm* 1 *and CPLEX for medium-scale cases of* (7.5).

| $n$ | Algorithm 1 with $\gamma = 1$ | | CPLEX | Optimal Objective |
|---|---|---|---|---|
| | No. It | CPU Sec | CPU Sec | Value=Trace($C^T X$) |
| 50 | 184 | 0.045 | 0.094 | 485.782539 |
| 100 | 211 | 0.132 | 0.188 | 985.870693 |
| 200 | 233 | 0.590 | 0.765 | 1983.976390 |

with such an excessive step size. This discrepancy is in fact quite common in optimization. But there is no general rule of how to choose an appropriate excessive step size for Algorithm 2. Last, we would reiterate that Algorithm 2 is still useful when the calculation of the iteration-dependant optimal step size $\alpha_k^*$ is too computationally expensive for Algorithm 1.

Then we compare the proposed algorithms with CPLEX for solving some medium-scale cases of the assignment problem (7.5). We test the cases where $n = 50$, 100, and 200. Note for these cases, the values $2\big(1 - n\sqrt{1/(n^2+1)}\,\big)$ are extremely tiny (e.g., it is $3.9988e-04$ when $n = 50$). Thus Algorithm 2 is by no way efficient with such a tiny step size. We thus only compare Algorithm 1 with CPLEX. We report the comparison of Algorithm 1 and CPLEX in Table 2. Note that we compare the number of iterations and computing time (in seconds) when these two methods achieve the same optimal objective function value. According to Table 2, Algorithm 1 is even faster than CPLEX for solving medium-scale cases of (7.5).

For large-scale cases of (7.5), the proposed algorithms are usually slower than CPLEX. For example, when $n = 300$, Algorithm 1 requires about 1.6 times more computation time than CPLEX to achieve the same optimal objective function value 2984.198323. One reason is that the exact ALM step (1.5) is decomposed into too many subproblems ($n^2$ ones), making the loss of accuracy too large. In any case, CPLEX is a well-commercialized package that is particularly efficient for linear programs while our proposed algorithms are for the generic setting of (1.3). Its superiority to CPLEX for small- and medium-scale cases of (7.5) is interesting.

**7.3. Application to ill-conditioned linear programs.** In this subsection, we test some cases of the LP model (7.1) whose coefficient matrices are ill-conditioned and further show the efficiency of Algorithm 1.

**7.3.1. Dataset.** We first generate an ill-conditioned coefficient matrix $A$ for the LP model (7.1). Three matrices $U$, $V$, and $\Sigma$ are generated as following:

$$\begin{aligned} &u = 10\cdot \text{rand}(l,1) - 5, && U = I - 2\frac{uu^T}{u^T u},\\ &v = 10\cdot \text{rand}(m,1) - 5, && V = I - 2\frac{vv^T}{v^T v}, \qquad (7.9)\\ &\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_l)^T, && \Sigma = [\text{diag}(\sigma+\tau)\ \text{zeros}(l, m-l)], \end{aligned}$$

where $\sigma_i = \cos(\frac{i\pi}{l+1}) + 1$ for $i = 1, \ldots, l$, and $\tau$ is a constant. Then, we generate $A := U\Sigma V \in \Re^{l\times m}$. It is easy to see that the condition number of $A$, denoted by $\kappa$, is given by $\frac{\sigma_1+\tau}{\sigma_l+\tau}$. Thus, we can generate ill-conditioned cases of $A$ easily. For example, if we let $\tau = \frac{\sigma_1-10^6\sigma_l}{10^6-1}, \frac{\sigma_1-10^8\sigma_l}{10^8-1}, \frac{\sigma_1-10^{10}\sigma_l}{10^{10}-1}, \frac{\sigma_1-10^{12}\sigma_l}{10^{12}-1}$, then the condition numbers of $A$ are $\kappa = 10^6$, $10^8$, $10^{10}$, $10^{12}$, respectively. Next, we generate two vectors $x^l \in \Re^m$ and $x^u \in \Re^m$ by "`x^l=zeros(m,1)`" and "`x^u=5+5*rand(m,1)`," respectively, as the box constraints in (7.1). Let us choose

$$
\begin{aligned}
&x_i^* = x_i^l, \quad i = 1, 3, \ldots, 2\left\lceil \frac{m}{2} \right\rceil - 1,\\
&x_i^* = x_i^u, \quad i = 2, 4, \ldots, 2\left\lceil \frac{m-1}{2} \right\rceil,\\
&\lambda^* = 4 \cdot \mathrm{rand}(l, 1) - 2,
\end{aligned}
\tag{7.10}
$$

where $\lceil a \rceil$ is the smallest integer larger than or equal to $a$. Moreover, let us generate a vector $p \in \Re^m$ by the following procedure:

```
q=5*rand(m,1)-2.5,
p(1:2:m)=max(q(1:2:m),0)+0.05*rand(size(q(1:2:m))),
p(2:2:m)=min(q(2:2:m),0)-0.05*rand(size(q(2:2:m))).
```

Obviously, we have $p_i \geq 0$ for $i = 1, 3, \ldots, 2\left\lceil \frac{m}{2} \right\rceil - 1$ and $p_i \leq 0$ otherwise. Now, we generate $c \in \Re^m$ by

$$c = A^T\lambda^* + p$$

and set $b = Ax^* \in \Re^l$. Then, it is easy to see that the generated $(A, x^*, \lambda^*, c, b)$ satisfies

$$(x - x^*)^T(c - A^T\lambda^*) \geq 0, \quad \forall\, x \in [x^l, x^u], \tag{7.11}$$

$$Ax^* = b, \tag{7.12}$$

which is exactly the optimality condition of the model (7.1). Therefore, we can generate a specific case of the model (7.1) whose solution is known and whose coefficient matrix $A$ can be ill-conditioned with an assigned condition number.

**7.3.2. Numerical results.** We use the stopping criterion

$$\max\{\|x^k - \widetilde{x}^k\|_\infty, \|A\widetilde{x}^k - b\|_\infty\} < 10^{-6}$$

and choose $\gamma = 1$ and $\beta = 10/\sqrt{m}$ when implementing Algorithm 1.

In Table 3, we test some cases of $l$, $m$, and $\kappa$ for (7.1) and report the results of the proposed Algorithm 1. Since the dataset is generated randomly, for each fixed combination of $l$, $m$, and $\kappa$, we run Algorithm 1 ten times and report the average performance. For all the tested cases, the solutions obtained by Algorithm 1 are exactly the same as what we generated. Thus, we only report the number of iterations ("No. It"), computing time in seconds ("CPU Sec"), and the objective function value ("$c^Tx^*$"). The data in Table 3 show that the proposed Algorithm 1 is efficient even for ill-conditioned cases of (7.1). In particular, it works well for quite challenging cases where there are lots of constraints and the dimension of the variable is high. This observation further supports our philosophy in algorithmic design of ensuring the convergence of splitting ALM by a simple relaxation step.

Table 3
*Algorithm 1 for ill-conditioned LP model (7.1).*

| $l$ | $m$ | $\kappa$ | No. It | CPU Sec | $c^T x^*$ |
|---|---|---|---|---|---|
| 10 | 25 | $10^6$ | 133.5 | 0.019 | −63.702126 |
| | | $10^8$ | 145.5 | 0.019 | −63.735452 |
| | | $10^{10}$ | 119.2 | 0.020 | −75.329298 |
| | | $10^{12}$ | 149.0 | 0.023 | −67.049817 |
| 50 | 500 | $10^6$ | 52.3 | 0.807 | −1236.612600 |
| | | $10^8$ | 44.1 | 0.580 | −1183.861435 |
| | | $10^{10}$ | 53.3 | 0.920 | −1210.642820 |
| | | $10^{12}$ | 38.3 | 0.616 | −1185.817668 |
| 200 | 1000 | $10^6$ | 95.7 | 9.765 | −2493.092632 |
| | | $10^8$ | 67.5 | 6.931 | −2401.944498 |
| | | $10^{10}$ | 78.0 | 7.906 | −2456.893622 |
| | | $10^{12}$ | 97.9 | 9.892 | −2490.443992 |
| 500 | 2000 | $10^6$ | 122.1 | 87.068 | −4902.155818 |
| | | $10^8$ | 110.1 | 80.676 | −4863.458678 |
| | | $10^{10}$ | 100.0 | 76.595 | −4818.817975 |
| | | $10^{12}$ | 106.7 | 79.734 | −4754.975873 |
| 1000 | 5000 | $10^6$ | 118.9 | 767.535 | −12262.167578 |
| | | $10^8$ | 116.0 | 722.327 | −12172.577610 |
| | | $10^{10}$ | 142.9 | 863.860 | −12245.499874 |
| | | $10^{12}$ | 141.8 | 869.286 | −12150.665808 |

**7.4. Application to an $l_1$ norm model.** Last, we test the model

$$\begin{aligned} &\min \ \ \|x\|_1 \\ &\text{s.t.} \ \ Ax = b, \\ &\qquad x^l \le x \le x^u, \end{aligned} \tag{7.13}$$

where $A \in \Re^{l\times m}$, $b \in \Re^l$, $x^l \in \Re^m$, and $x^u \in \Re^m$ with $x_i^l \le x_i^u$ for $i = 1, 2, \ldots, m$, and $\|x\|_1 = \sum_{i=1}^m |x_i|$. We assume $l < m$ throughout. This model can be explained as seeking a sparse solution of the underdetermined system linear equations $Ax = b$, subject to additional bound constraints. In practice, the $l_1$ norm model (7.13) is a special case of (1.3), where $n_i \equiv 1$, $\theta_i(x_i) = |x_i|$, $A_i \in \Re^l$ for $i = 1, 2, \ldots, m$, and $\mathcal{X}_i = [x_i^l, x_i^u]$.

We apply Algorithm 1 and test two cases of (7.13). First, $x^l = -\mathbf{1}$, and $x^u = \mathbf{1}$, where $\mathbf{1}$ stands for a vector in $\Re^m$ whose elements are all 1. Second, $x^l = -\infty$ and $x^u = +\infty$ and $l \ll m$. In this case, the model (7.13) reduces to the well-known basis pursuit problem (see, e.g., [4, 5]), which has a wide range of applications in areas such as compressing sensing, signal processing, data science, information science, and so on. Note that conceptually the model (7.13) can be reformulated as a special case of (7.1) by introducing auxiliary variables. But LP-based solvers are not efficient for (7.13) because of the enlargement of dimensionality; and there is a rich set of literature to discuss how to model it directly. Here, our approach, regarding it as a special case of (1.3) and applying the proposed splitting version of the ALM, is new and efficient and has to be tested.

**7.4.1. Specification.** Let us elaborate on the resulting subproblems (3.1) when the model (7.13) is treated as a special case of (1.3) and the proposed Algorithm 1 is applied. We also choose $H = \beta \cdot I$ with $\beta > 0$ for implementing Algorithm 1.

Clearly the $i$th subproblem in the splitting step (3.1) is the 1-dimensional minimization problem

$$\tilde{x}_i^k = \arg\min \left\{ |x_i| - x_i^T A_i^T \lambda^k + \frac{\beta}{2} \left\| \sum_{j=1}^{i-1} A_j x_j^k + A_i x_i + \sum_{j=i+1}^{m} A_j x_j^k - b \right\|^2 \Big| \; x_i \in [x_i^l, x_i^u] \right\},$$

which can be rewritten as

(7.14)
$$\tilde{x}_i^k = \arg\min \left\{ \frac{1}{\beta A_i^T A_i} |x_i| + \frac{1}{2} \left( x_i - \left( x_i^k - \frac{A_i^T (c - \lambda^k/\beta)}{A_i^T A_i} \right) \right)^2 \Big| \; x_i \in [x_i^l, x_i^u] \right\}.$$

The subproblem (7.14) can be further written as

$$\tilde{x}_i^k = \arg\min \left\{ \mu_i |x_i| + \frac{1}{2}(x_i - \rho_i)^2 \Big| \; x_i \in [x_i^l, x_i^u] \right\}$$

with

$$\mu_i := \frac{1}{\beta A_i^T A_i} \quad \text{and} \quad \rho_i := x_i^k - \frac{A_i^T (c - \lambda^k/\beta)}{A_i^T A_i}.$$

Thus, its closed-form solution is given by

$$\tilde{x}_i^k = \begin{cases} P_{[x_i^l, x_i^u]}\{\rho_i - \mu_i\} & \text{if } \; x_i^l \geq 0, \\ P_{[x_i^l, x_i^u]}\{\rho_i - P_{[-\mu_i, \mu_i]}\{\rho_i\}\} & \text{if } \; x_i^l < 0 < x_i^u, \\ P_{[x_i^l, x_i^u]}\{\rho_i + \mu_i\} & \text{if } \; x_i^u \leq 0, \\ \rho_i - P_{[-\mu_i, \mu_i]}\{\rho_i\} & \text{if } \; x_i^l = -\infty, \quad x_i^u = +\infty, \end{cases}$$

where $P_{[x_i^l, x_i^u]}$ denotes the projection onto $[x_i^l, x_i^u]$ under the Euclidean norm.

To generate a dataset for (7.13), we first generate the entries of $A$ randomly by the Gaussian distribution with mean 0 and standard deviation 1. As in [26], each row of $A$ is normalized as a vector with a length of 1. Specifically, the procedure of generating $A$ is "`randn('state',0)`" and "`A=normr(randn(l,m))`." As mentioned, the primary purpose of the model (7.13) is seeking a sparse solution of the system $Ax = b$. We thus generate a sparse vector $x^f \in \Re^m$ by the following procedure:

```
rand('state',0),   index=randperm(m),
x^f=zeros(m,1),     x^f(index(1:s))=sign(randn(s,1)).
```

This means $x^f$ has only $s$ nonzero entries and their values are 1 or $-1$. Finally, the vector $b$ is chosen as $Ax^f$.

**7.4.2. Numerical results.** The stopping condition is still

$$\max\{\|x^k - \widetilde{x}^k\|_\infty, \|A\widetilde{x}^k - b\|_\infty\} < 10^{-6},$$

and the parameters are still chosen as $\gamma = 1$ and $\beta = 10/\sqrt{m}$ when implementing Algorithm 1.

Table 4
*Algorithm* 1 *for* (7.13) *with* $x^l = -\mathbf{1}, x^u = \mathbf{1}$.

| $l$ | $m$ | s | No. It | CPU Sec |
|---|---|---|---|---|
| 10 | 25 | 2 | 43 | 0.031 |
| 20 | 50 | 5 | 12 | 0.031 |
| 50 | 100 | 10 | 13 | 0.062 |
| 100 | 300 | 15 | 18 | 0.124 |
| 200 | 500 | 20 | 14 | 0.390 |
| 500 | 1000 | 30 | 14 | 2.464 |
| 1000 | 2000 | 50 | 15 | 17.862 |
| 2000 | 5000 | 100 | 31 | 403.730 |

Table 5
*Algorithm* 1 *for* (7.13) *with* $x^l = -\infty, x^u = +\infty$.

| $l$ | $m$ | s | No. It | CPU Sec |
|---|---|---|---|---|
| 10 | 100 | 1 | 33 | 0.140 |
| 20 | 200 | 2 | 40 | 0.125 |
| 50 | 500 | 5 | 50 | 0.795 |
| 100 | 1000 | 10 | 93 | 4.617 |
| 200 | 2000 | 20 | 138 | 37.908 |
| 500 | 5000 | 50 | 222 | 784.154 |

We test some situations of (7.13) when $l$, $m$, and $s$ vary. Recall that the exact solution of (7.13) is unknown. For all the cases we have tested with $x^l = -\mathbf{1}$ and $x^u = \mathbf{1}$, we have $\|Ax^k - b\|_\infty = 0$ and the objective function value $\|x^k\|_1$ is exactly the same as the value of the corresponding $s$ when the stopping criterion is satisfied. We thus do not report their values (all are 0 and $s$), and only report the iteration numbers and computing time in Table 4. Note that the iterates are projected onto the box constraints $[-\mathbf{1}, \mathbf{1}]$. Thus, the objective function values are exactly the number of nonzero entries, i.e., the values of $s$. For all the cases we have tested with $x^l = -\infty$ and $x^u = +\infty$, we have $\|Ax^k - b\|_\infty \leq 10^{-5}$ and $|\|x^k\|_1 - s| \leq 10^{-5}$. We thus only report the iteration numbers and computing time in Table 5. The data in Tables 4 and 5 also clearly show the efficiency of Algorithm 1 for the $l_1$-norm model (7.13).

**8. Conclusions.** We consider embedding a full Jacobian decomposition into the ALM for solving a convex minimization model with linear constraints and an objective function in the form of the sum of $m$ functions without coupled variables. We find an example showing that the straightforward splitting version of ALM with full Jacobian decomposition could be divergent. We propose to adjust the output of the splitting version of ALM with full Jacobian decomposition by a relaxation step. Furthermore, we show that the range of the step size of the relaxation step in existing methods for special $m$ can be significantly enlarged for generic $m$. Two algorithms with different strategies of step size are thus derived. The refined splitting version of ALM with full Jacobian decomposition is then proved to have the worst-case $O(1/k)$ convergence rate, in both the ergodic and nonergodic senses. We finally report some numerical results to show the efficiency of the proposed algorithms. In particular, the algorithms designed in a generic setting have comparable performance with the well-developed specific software "IBM ILOG CPLEX Optimization Studio." This prompts the promising possibility of further optimizing the implementation of the proposed algorithms (e.g., coding in C++ with more proficiency) and then finally a publicized or commercialized version.

**Appendix A. An example showing (1.8)'s divergence.** In the appendix we show by a simple linear equation that the straightforward splitting version of ALM (1.8) with full Jacobian decomposition is divergent.

We consider the linear equation

$$x_1 + x_2 = 0,$$

which is a special problem of the LP model (7.1) with $c = 0$, $x^l = -\infty$, $x^u = +\infty$, $l = 1$, $m = 2$, $A_1 = A_2 = 1$, and $b = 0$. For this linear equation, the VI characterization (2.1) reduces to

$$\begin{cases} (x_1 - x_1^*)^T(-A_1^T\lambda^*) \geq 0 & \forall\, x_1 \in \mathcal{X}_1, \\ (x_2 - x_2^*)^T(-A_2^T\lambda^*) \geq 0 & \forall\, x_2 \in \mathcal{X}_2, \\ (\lambda - \lambda^*)^T\big(x_1^* + x_2^*\big) \geq 0 & \forall \lambda \in \Re. \end{cases}$$

Since $\mathcal{X}_1 = \mathcal{X}_2 = \Re$, the above VI is a system of linear equations

$$\begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = 0,$$

and its solution set is

$$\mathcal{W}^* = \{(x_1^*, x_2^*, \lambda^*) \,|\, x_1^* + x_2^* = 0, \lambda^* = 0\}.$$

To apply (1.8), we take $H = 1$. Recall we have

$$A = (1, 1), \qquad b = 0, \qquad \text{and} \qquad \text{diag}(A^TA) = I_2.$$

According to (7.2), the predictor $\tilde{w}^k = (\tilde{x}^k, \tilde{\lambda}^k)$ is given by

$$\tilde{x}^k = x^k - A^T\big(Ax^k - \lambda^k\big) \tag{A.1a}$$

and

$$\tilde{\lambda}^k = \lambda^k - A\tilde{x}^k.$$

Substituting (A.1a) into the above equation and using $AA^T = 2$, we get

$$\tilde{\lambda}^k = Ax^k - \lambda^k. \tag{A.1b}$$

Putting (A.1a) and (A.1b) together, using $A^T = \left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$ and $A^TA = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right)$, the predictor form (A.1) becomes

$$\begin{pmatrix} \tilde{x}_1^k \\ \tilde{x}_2^k \\ \tilde{\lambda}^k \end{pmatrix} = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} x_1^k \\ x_2^k \\ \lambda^k \end{pmatrix}. \tag{A.2}$$

If we directly take the predictor as the new iterate and begin with $(x_1^0, x_2^0, \lambda^0) = (0, 0, 1)$, then according to (A.2), we have

$$\begin{pmatrix} x_1^1 \\ x_2^1 \\ \lambda^1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}, \quad \begin{pmatrix} x_1^2 \\ x_2^2 \\ \lambda^2 \end{pmatrix} = \begin{pmatrix} -2 \\ -2 \\ 3 \end{pmatrix},$$
$$\begin{pmatrix} x_1^3 \\ x_2^3 \\ \lambda^3 \end{pmatrix} = \begin{pmatrix} 5 \\ 5 \\ -7 \end{pmatrix}, \quad \begin{pmatrix} x_1^4 \\ x_2^4 \\ \lambda^4 \end{pmatrix} = \begin{pmatrix} -12 \\ -12 \\ 17 \end{pmatrix}, \ldots.$$

Using the formula (A.2) and $w^{k+1} = \tilde{w}^k$, we obtain by induction that

$$x_1^k = x_2^k, \qquad \mathrm{Sign}(x_1^k) = -\mathrm{Sign}(\lambda^k), \qquad |x_1^k| \geq k, \quad \text{and} \quad |\lambda^k| \geq k \quad \forall\, k \geq 1.$$

The sequence $\{w^k\}$ thus does not converge to any solution point in the set

$$w^* = \{(x_1^*, x_2^*, 0)\,|\, x_1^* = -x_2^*\}.$$

In other words, the splitting version of ALM (1.8) with full Jacobian decomposition is divergent and a certain relaxation step is a must to ensure its convergence.

**Acknowledgment.** The authors would like to thank Dr. Haiming Song for his help in completing some parts of the numerical experiments.

REFERENCES

[1] E. Blum and W. Oettli, *Mathematische Optimierung, Grundlagen und Verfahren*, Ökonometrie Unternehmensforschung 20, Springer-Verlag, Berlin, 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 Mach. Learn., 3 (2010), pp. 1–122.

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

[4] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM Rev., 43 (2001), pp. 129–159.

[5] D. Donoho, *Compressed sensing*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1289–1306.

[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] J. Eckstein and M. Fukushima, *Some reformulation and applications of the alternating direction method of multipliers*, in Large Scale Optimization: State of the Art, W. W. Hager, D. W. Hearn, and P. M. Pardalos, eds., Kluwer Academic, Dordrecht, 1994, pp. 115–134.

[8] F. Facchinei and J. S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer Ser. Oper. Res. 1, Springer Verlag, New York, 2003.

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

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

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

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

[13] R. Glowinski and A. Marrocco, *Approximation par éléments finis d'ordre un et résolution par pénalisation-dualité d'une classe de problèmes non linéaires*, M2AN Math. Model. Numer. Anal., (1975), pp. 41–76.

[14] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM Stud. Appl. Numer. Math., SIAM, Philadelphia, 1989.

[15] D. R. Han and X. M. Yuan, *A note on the alternating direction method of multipliers*, J. Optim. Theory Appl., 155 (2012), pp. 227–238.

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

[17] B. S. He, *A class of projection and contraction methods for monotone variational inequalities*, Appl. Math. Optim., 35 (1997), pp. 69–76.

[18] B. S. He, *Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities*, Comput. Optim. Appl., 42 (2009), pp. 195–212.

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

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

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

[22] B. S. He and X. M. Yuan, *On nonergodic convergence rate of Douglas-Rachford alternating direction method of multipliers*, Numer. Math., 130 (2015), pp. 567–577.

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

[24] M. Hong and Z. Q. Luo, *On the Linear Convergence of the Alternating Direction Method of Multipliers*, preprint, arXiv:1208.3922, 2012.

[25] Z. K. Jiang and X. M. Yuan, *New parallel descent-like method for solving a class of variational inequalities*, J. Optim. Theory Appl., 145 (2010), pp. 311–323.

[26] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, *An interior-point method for large-scale $l_1$-regularized least squares*, IEEE J. Sel. Top. Signal Process., 1 (2007), pp. 606–617.

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

[28] B. Martinet, *Regularisation, d'inéquations variationelles par approximations succesives*, M2AN Math. Model. Numer. Anal., 4 (1970), pp. 154–159.

[29] A. S. Nemirovsky and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Wiley Ser. Discrete Math., Wiley, New York, 1983.

[30] Y. E. Nesterov, *A method for solving the convex programming problem with convergence rate* $O(1/k^2)$, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543–547.

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

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

[33] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, New York, 1969, pp. 283–298.

[34] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

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

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

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