On full Jacobian decomposition of the augmented Lagrangian method for separable convex programming

Bingsheng He[1] **Liusheng Hou** [2] and **Xiaoming Yuan**[3]

Revised on March 21, 2014

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 an under-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 under-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 a nonergodic senses for the new algorithm. Finally, an assignment problem is tested to illustrate the efficiency of the new algorithm.

Keywords: Convex programming, augmented Lagrangian method, Jacobian decomposition, contraction methods, convergence rate, operator splitting methods, assignment problem

---

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 [21, 30] 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

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

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, $\|x\|_H := \sqrt{x^T Hx}$ where x and the positive definite matrix H have appropriate

---

[1]International Centre of Management Science and Engineering, and Department of Mathematics, Nanjing University, Nanjing, 210093, China. This author was supported by the NSFC Grant 91130007 and the MOEC fund 20110091110004. Email: hebma@nju.edu.cn

[2]Department of Mathematics, Nanjing University, Nanjing, 210093, China. Email: houlsheng@gmail.com

[3]Department of Mathematics, Hong Kong Baptist University, Hong Kong, China. This author was supported by the General Research Fund from Hong Kong Research Grants Council: HKBU203712. Email: xmyuan@hkbu.edu.hk

dimensions. In [31], it was analyzed that the ALM is indeed an application of the proximal point algorithm (PPA) in [25] 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 as concrete forms 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):

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

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 sub-vectors, i.e., $x = (x_1, x_2 \cdots, x_m)$, each $x_i \in \Re^{n_i}$ can be explained as the decision variable of the i-th objective θ_i. The coefficient matrix A is partitioned accordingly as $(A_1, A_2, \cdots, A_m)$ in (1.3). Throughout, the solution set of (1.3) is assumed to be nonempty.

Let the Lagrange function of (1.3) be

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

and the augmented Lagrange function of (1.3) be

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

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

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

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 [21, 30, 32]. But, for the separable case with $m \geq 2$, the implementation of (1.5) may have the difficulty that all the sub-vectors x_i's are required to solve simultaneously and all θ_i's are considered aggregately. Even though each θ_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 θ_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 diversifying 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 [11] (see also [8]) 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:

$$\begin{cases} x_1^{k+1} &= \arg\min\{\theta_1(x_1) - x_1^T A_1^T \lambda^k + \frac{1}{2}\|A_1x_1 + A_2x_2^k - b\|_H^2 \mid x_1 \in \mathcal{X}_1\}, \\ x_2^{k+1} &= \arg\min\{\theta_2(x_2) - x_2^T A_2^T \lambda^k + \frac{1}{2}\|A_1x_1^{k+1} + A_2x_2 - b\|_H^2 \mid x_2 \in \mathcal{X}_2\}, \\ \lambda^{k+1} &= \lambda^k - H(A_1x_1^{k+1} + A_2x_2^{k+1} - b). \end{cases} \tag{1.6}$$

We refer to, e.g. [4, 5, 7, 9, 10, 12, 17, 24, 34], 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 complimented that "ADMM is at least comparable to very specialized algorithms (even in the serial setting), and in most cases, the simple ADMM algorithm will be efficient enough to be useful". 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:

$$\begin{cases} x_1^{k+1} = \arg\min\{\theta_1(x_1) - x_1^T A_1^T \lambda^k + \frac{1}{2}\|A_1x_1 + \sum_{j=2}^m A_jx_j^k - b\|_H^2 \mid x_1 \in \mathcal{X}_1\}; \\ x_2^{k+1} = \arg\min\{\theta_2(x_2) - x_2^T A_2^T \lambda^k + \frac{1}{2}\|A_1x_1^{k+1} + A_2x_2 + \sum_{j=3}^m A_jx_j^k - b\|_H^2 \mid x_2 \in \mathcal{X}_2\}; \\ \cdots\cdots \\ x_i^{k+1} = \arg\min\{\theta_i(x_i) - x_i^T A_i^T \lambda^k + \frac{1}{2}\| \sum_{j=1}^{i-1} A_jx_j^{k+1} + A_ix_i + \sum_{j=i+1}^m A_jx_j^k - b\|_H^2 \mid x_i \in \mathcal{X}_i\}; \\ \cdots\cdots \\ x_m^{k+1} = \arg\min\{\theta_m(x_m) - x_m^T A_m^T \lambda^k + \frac{1}{2}\| \sum_{j=1}^{m-1} A_jx_j^{k+1} + A_mx_m - b\|_H^2 \mid x_m \in \mathcal{X}_m\}; \\ \lambda^{k+1} = \lambda^k - H(\sum_{j=1}^m A_jx_j^{k+1} - b). \end{cases} \tag{1.7}$$

Despite that the efficiency of (1.7) has been verified empirically in various contexts (e.g. [29, 33]), it was recently shown in [3] that the scheme (1.7) is not necessarily convergent. We refer to [3, 13, 18, 22] 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:

$$\begin{cases} x_1^{k+1} = \arg\min\{\theta_1(x_1) - x_1^T A_1^T \lambda^k + \frac{1}{2}\|A_1x_1 + \sum_{j=2}^m A_jx_j^k - b\|_H^2 \mid x_1 \in \mathcal{X}_1\}; \\ x_2^{k+1} = \arg\min\{\theta_2(x_2) - x_2^T A_2^T \lambda^k + \frac{1}{2}\|A_1x_1^{k} + A_2x_2 + \sum_{j=3}^m A_jx_j^k - b\|_H^2 \mid x_2 \in \mathcal{X}_2\}; \\ \cdots\cdots \\ x_i^{k+1} = \arg\min\{\theta_i(x_i) - x_i^T A_i^T \lambda^k + \frac{1}{2}\| \sum_{j=1}^{i-1} A_jx_j^{k} + A_ix_i + \sum_{j=i+1}^m A_jx_j^k - b\|_H^2 \mid x_i \in \mathcal{X}_i\}; \\ \cdots\cdots \\ x_m^{k+1} = \arg\min\{\theta_m(x_m) - x_m^T A_m^T \lambda^k + \frac{1}{2}\| \sum_{j=1}^{m-1} A_jx_j^{k} + A_mx_m - b\|_H^2 \mid x_m \in \mathcal{X}_m\}; \\ \lambda^{k+1} = \lambda^k - H(\sum_{j=1}^m A_jx_j^{k+1} - b). \end{cases} \tag{1.8}$$

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 is 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 even a 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 can not 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 to combine the output of (1.8) with an under-relaxation step. In [14, 16, 23], some such steps were proposed for the special cases of $m = 2$, $m = 3$

and $m \geq 3$, respectively. In [14] (for the case $m \geq 3$) and [16] (for the case $m = 3$), it was suggested to further adjust the output of (1.8) via the step

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

where $\alpha > 0$ is a chosen step size, $w^k = (x_1^k, x_2^k, \cdots, x_m^k, \lambda^k)$ 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 this loss of accuracy by combining the last iterate w^k with $\tilde{w}^k$ approximately (i.e., seeking an appropriate step size α). Technically, as Theorem 4.7 shows in Section 4, this underrelaxation step with an appropriate step size α can ensure the strict contraction of the iterative sequence and thus the global convergence becomes provable by following 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 α for (1.9); or more specifically, to enlarge the range of possible step size. In [16], 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 [14], 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 size in (1.9) can be enlarged to $2\big(1 - \sqrt{\frac{m}{m+1}}\big)$, i.e., $\alpha \in \big(0, 2\big(1 - \sqrt{\frac{m}{m+1}}\,\big)\big)$ ensures the convergence of the combination of (1.8) with (1.9). In particular, because it holds

$$\frac{1}{m+1} < 2\big(1 - \sqrt{\tfrac{m}{m+1}}\,\big)$$

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 [14]. 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 a nonergodic senses for the new splitting version of ALM with full Jacobian decomposition. Note that we follow the work [26, 27] 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 ϵ. This line of analysis is mainly motivated by our recent work of convergence analysis for the ADMM in [19, 20].

The rest of the paper is organized as follows. In Section 2, we provide some preliminaries which are useful for further discussions and summarize some notations for the convenience of discussion. In Section 3, we proposed 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 a nonergodic senses, respectively. In Section 7, we test the efficiency of the proposed algorithms for solving the assignment problem, and compare them with CPLEX. Finally, we make some conclusions in Section 8.

2 Preliminaries

In this section, we summarize some preliminaries which are useful for further discussions and then give some notations 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 $L(x_1, x_2, \cdots, x_m, \lambda)$ defined in (1.4) is the Lagrange function of (1.3). Let $\big(x_1^*, x_2^*, \ldots, x_m^*, \lambda^*\big)$ be a saddle point of $L(x_1, x_2, \cdots, x_m, \lambda)$. Then, we have

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

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

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

such that

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

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

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

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

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

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 $\mathrm{VI}(\mathcal{W}, F, \theta)$, 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 [6] and Theorem 2.1 in [19] for the proof of the following theorem.

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

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

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

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

where

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

We refer to [28] for similar definition of the ϵ-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

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

and thus have

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

Remark 2.2 *The matrix M in [16] (see (4.5) in Page 206 of [16]) is just the matrix G here with $m = 3$. In addition, if $A_1, \dots, A_m$ are full column rank matrices, then G is positive definite.*

2.4 A proposition

The following proposition can be proved by elementary technique, 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 $a_2 + b_2 > 0$, 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 non-zero 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

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

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 detail of choosing the step size α 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 is updated dynamically. Recall the output of (1.8) needs to be further adjusted. We thus relabel it as $\tilde{w}^k := (\tilde{x}_1^k, \tilde{x}_2^k, \cdots, \tilde{x}_m^k, \tilde{\lambda}^k)$. That is, we can rewrite the splitting ALM step (1.8) as

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

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

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

Remark 3.1 *We will show that the specific strategy determining α_k^* in (3.2b) comes from the purpose of maximizing certain quadratic function which is beneficial for making more progress of 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 γ 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 α_k in (3.2b) can be regarded as an extension of that in [16] for the special case where $m = 3$. More specifically, in Section 4, we will prove that α_k^* defined in (3.2b) is uniformly lower bounded by $\big(1 - \sqrt{\frac{m}{m+1}}\,\big)$ for all k's. Note that*

$$2\big(1 - \sqrt{\tfrac{m}{m+1}}\,\big) = 2 - \sqrt{3}$$

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 [16] for the special case of $m = 3$. As we have mentioned, the matrix G defined in (2.7) reduces to the matrix M in [16] when $m = 3$. Also, the function $\varphi(w^k, \tilde{w}^k)$ defined in (3.2c) reduces to the function in [16] (see (4.13)) when $m = 3$.

3.2 Algorithm 2 with a constant step size

For Algorithm 1, the step size α_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 α_k^* defined in (3.2b) satisfies $\alpha_k^* \geq \big(1 - \sqrt{\frac{m}{m+1}}\,\big)$ 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).

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

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

Remark 3.3 *Note it holds*

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

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 γ is taken as $\big(1+\sqrt{\frac{m}{m+1}}\,\big) \in (0,2)$ 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 [15]).

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

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

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

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

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

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

The last equation in (3.1) can be rewritten as

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

and in variational form

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

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

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

for all $w \in \mathcal{W}$. Using the notations 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

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

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

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

By using

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

we get

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

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

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

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

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

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, 15], we now need to prove that

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

for 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*

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

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

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

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

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

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

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

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

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

and

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

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

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

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 [16], the relation (4.15) in [16] is a special result of the assertion (4.9) with $m = 3$. In other words, Lemma 4.4 include Lemma 4.1 in [16] as a special case.

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

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

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 α, we can reduce the unknown distance function $\|w - w^*\|_G^2$. In order to explain how to determine the step size α_k in (3.2a) (resp. in (3.3a)), we define the step-size-dependent new iterate by

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

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*

$$\vartheta(\alpha) \geq q(\alpha), \tag{4.15}$$

where

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

and

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

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 α as

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

According to (4.9), α_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 α 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*

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

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

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

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

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

Using the fact (4.9) we obtain

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

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*

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

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). For Algorithm 2, it follows from (3.3a) that

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

and

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

Thus, the assertion (4.25) follows from (4.15)-(4.17) and (4.24) immediately. □

By using the following notations

$$y_i = A_i x_i,\ i = 1, \ldots, m, \qquad v = (y_1, y_2, \ldots, y_m, \lambda)$$

$$\mathcal{V}^* = \{(A_1x_1^*, A_2x_2^*, \ldots, A_mx_m^*, \lambda^*) \,|\, (x_1^*, x_2^*, \ldots, x_m^*, \lambda^*) \in \mathcal{W}^*\},$$

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 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, $i = 1, \ldots, m$ 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 first prove the first assertion. If all A_i, $i = 1, \ldots, m$ are full column rank matrices, then the matrix G defined in (2.7) is positive definite. Following the standard analytic framework of contraction methods in [1, 15], the assertion (4.21) or (4.25) immediately indicates that $\{w^k\}$ 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 [19].

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 ϵ in the sense that

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

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

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

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 λ-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:

$$Q = \begin{pmatrix} 2A_1^T HA_1 & A_1^T HA_2 & \cdots & A_1^T HA_m & 0 \\ A_2^T HA_1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & A_{m-1}^T HA_m & 0 \\ A_m^T HA_1 & \cdots & A_m^T HA_{m-1} & 2A_m^T HA_m & 0 \\ -2A_1 & \cdots & -2A_{m-1} & -2A_m & H^{-1} \end{pmatrix}, \tag{5.4}$$

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

$$\begin{aligned} \frac{Q^T + Q}{2} &= \begin{pmatrix} 2A_1^T HA_1 & A_1^T HA_2 & \cdots & A_1^T HA_m & -A_1^T \\ A_2^T HA_1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & A_{m-1}^T HA_m & -A_{m-1}^T \\ A_m^T HA_1 & \cdots & A_m^T HA_{m-1} & 2A_m^T HA_m & -A_m^T \\ -A_1 & \cdots & -A_{m-1} & -A_m & H^{-1} \end{pmatrix} \\ &= \mathcal{A}^T \begin{pmatrix} 2I_l & I_l & \cdots & I_l & -I_l \\ I_l & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & I_l & -I_l \\ I_l & \cdots & I_l & 2I_l & -I_l \\ -I_l & \cdots & -I_l & -I_l & I_l \end{pmatrix}_{(m+1)\times(m+1)} \mathcal{A}, \end{aligned} \tag{5.6}$$

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 semi-definite. 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 in 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

$$\bar{x}_i^k \in \mathcal{X}_i,\; \theta_i(x_i) - \theta_i(\bar{x}_i^k) + (x_i - \bar{x}_i^k)^T \big\{ -A_i^T \bar{\lambda}^k + A_i^T H \big(\sum_{j=1}^m A_j(\bar{x}_j^k - x_j^k) \big) + A_i^T H A_i (\bar{x}_i^k - x_i^k) \big\} \geq 0, \tag{5.10}$$

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

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

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 $\tilde{x}_i = \bar{x}_i, \;\; i = 1, \ldots, m$)

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

we have

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

The above equation can be written in variational form

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

Combining (5.11) and (5.12) together, using the notations 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*

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

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 follows that

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

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

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

and thus

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

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 $(w-\bar{w}^k)^T F(w)=(w-\bar{w}^k)^T F(\bar{w}^k)$, 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 Lemma 5.1 and Lemma 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 $a=w$, $b=\bar{w}^k$ $c=w^k$ and $d=w^{k+1}$ in the identity

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

the right-hand side of (5.19) becomes

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

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

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

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 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

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

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

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

which implies that

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

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

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

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

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

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 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 [20]. As stated in [20], 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 Algorithm 2.

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

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

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 non-increasing. 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

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

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

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

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*

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

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

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

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

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

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

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

we have

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

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 $\alpha \leq 2\big(1 - \sqrt{\frac{m}{m+1}}\big)$, 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 $L^T G$ (see (5.15)), it yields

$$L^T(Q^T + Q)L = 2\begin{pmatrix} 2A_1^T HA_1 & A_1^T HA_2 & \cdots & A_1^T HA_m & A_1^T \\ A_2^T HA_1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & A_{m-1}^T HA_m & A_{m-1}^T \\ A_m^T HA_1 & \cdots & A_m^T HA_{m-1} & 2A_m^T HA_m & A_m^T \\ A_1 & \cdots & A_{m-1} & A_m & H^{-1} \end{pmatrix}. \tag{6.12}$$

Furthermore, using the notation

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

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

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

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 ν is the eigenvalue of T and z is the related eigenvector. In the following we show that all the eigenvalue of T are non-negative. Note that

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

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

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

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

$$\left\{ \begin{array}{l} (2-\alpha)y + (2-\alpha)(e^Ty)e = \nu y, \\ e^Ty = 0. \end{array} \right. \qquad \Rightarrow \qquad \left\{ \begin{array}{l} (2-\alpha)y = \nu y, \\ e^Ty = 0. \end{array} \right.$$

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

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

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

$$\left\{ \begin{array}{l} (2-\alpha)y + ((2-\alpha)e^Ty + 2)e = \nu y, \\ 2e^Ty + (2-\alpha) = \nu. \end{array} \right. \tag{6.15}$$

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

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

The rest 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 $\alpha \leq 2\big(1 - \sqrt{\frac{m}{m+1}}\,\big)$, we have

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

and thus $\nu_{\min}(T) \geq 0$. All the eigenvalues of T are non-negative and the lemma is proved. □

Therefore, the monotonicity of $\{\|w^k - \tilde{w}^k\|_G\}$ is a straightforward consequence of Lemma 6.2 and Lemma 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*

$$\alpha_k \leq 2\big(1 - \sqrt{\tfrac{m}{m+1}}\,\big) \tag{6.16}$$

for all k's. Then we have

$$\|w^k - w^{k+1}\|_G^2 \leq \frac{1}{\tau(k+1)}\|w^0 - w^*\|_G^2, \quad \forall w^* \in \mathcal{W}^*, \tag{6.17}$$

where τ is a constant defined in (6.2).

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

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

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

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

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

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

The assertion (6.17) follows from (6.18) and (6.19) immediately. The proof is complete. □

Notice that $\mathcal{W}^*$ is convex and closed. Let $d := \sup\{\|w^0 - w^*\|_G^2 \,|\, w^* \in \mathcal{W}^*\}$. Then, after $(k+1)$ iterations of Algorithm 1 or 2, we have $\|w^k - w^{k+1}\|_G^2 \leq \frac{d}{\tau} \cdot \frac{1}{k+1} = O(1/k)$. Since w^{k+1} is a solution of $\mathrm{VI}(\mathcal{W}, F, \theta)$ 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

As we have mentioned, we shall apply the proposed algorithms to solve an assignment problem which can be formulated as a linear programming (LP) model and thus a special case of (1.3). By comparison with the very efficient LP solver "CPLEX", we demonstrate the efficiency of our proposed algorithms numerically.

Note that splitting versions of ALM with full Jacobian decomposition have been well tested in the literature [14, 23], especially in [14] by 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. In this paper, we further illustrate the efficiency of the new splitting version of ALM with full Jacobian decomposition, but from a different perspective. Note that an extreme case of (1.3) is the LP model where $n_i = 1$ for all i's and each θ_i is a linear function defined on $\Re^1$. We thus can artificially treat a LP as a very special case of (1.3) and apply the proposed splitting version of ALM with full Jacobian decomposition. We shall focus on the assignment problem which is a structured LP with specific coefficient matrices in its constraint (In fact, these matrices result in extremely easy subproblems when the proposed scheme is applied). We will show that when the new splitting version of ALM with full Jacobian decomposition developed in generic setting is applied to solve the particular LP problem, it is even competitive with the well commercialized specific LP solver "IBM ILOG CPLEX Optimization Studio" (CPLEX for short). The efficiency of new splitting versions of ALM with full Jacobian decomposition is thus further demonstrated.

7.1 Implementation of Algorithms 1 and 2 in LP context

We first show the implementation of the proposed algorithms to LPs. As we have mentioned, instead of aiming at the specific LPs, the proposed algorithms are proposed for the general setting of (1.3) and they are applicable to many other cases such as the image processing applications already tested in [14]. But we want to show that they are still efficient even when they are used to solve some LPs,

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 $i = 1, 2, \cdots, m$. 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 $i = 1, 2, \cdots, m$, and $\mathcal{X}_i = [x_i^l, x_i^u]$.

To implement the proposed algorithms, we throughout set $H = \beta I_l$ 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

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

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

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

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

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

which can be characterized by a projection equation:

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

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

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

where $x = (x_1, x_2, \cdots, x_m)$ and $A = (A_1, A_2, \cdots, A_m)$. In addition, the λ-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 under-relaxation step (3.2b) in Algorithm 1, we have

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

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 under-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". 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 $0 \leq x_{ij} \leq 1$ 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 script

```
rand('state',0);        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

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

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

$$A = \begin{pmatrix} 1 & 1 & \cdots & 1 & & & & & & & & & \\ & & & & 1 & 1 & \cdots & 1 & & & & & \\ & & & & & & & & \cdots\cdots\cdots & & & & \\ & & & & & & & & & 1 & 1 & \cdots & 1 \\ 1 & & & & 1 & & & & & 1 & & & \\ & 1 & & & & 1 & & & & & 1 & & \\ & & \ddots & & & & \ddots & & \cdots\cdots\cdots & & & \ddots & \\ & & & 1 & & & & 1 & & & & & 1 \end{pmatrix}_{2n \times n^2} \quad \text{and} \quad b = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{7.6}$$

Thus, the Lagrangian multiplier λ is an $2n$-vector $\lambda = (y, z)$, and $\text{diag}(A^T A) = 2I_{n^2}$. 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) are 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

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

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 element-wise given by $\max\{\min\{\xi, 1\}, 0\}$.

To calculate the under-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

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

and

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

7.3 Numerical results

Now, we report some numerical results when Algorithms 1 and 2 are applied to solve the model (7.5). Our code was written by Matlab and all the numerical experiments were conducted on a laptop computer with a 2.53GHz processor and 4GB memory.

Throughout, we choose $\beta = 5/n$ (recall $H = \beta I_l$) and the initial iterate is taken as 0. The stopping criterion is

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

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 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 $\big(0, 2\big(1 - n\sqrt{1/(n^2+1)}\,\big)\big)$. Nevertheless, this choice is too conservative and it can hardly be good to result in fast convergence. For example, $2\big(1 - n\sqrt{1/(n^2+1)} \approx 0.0099$ 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 values n-independent values (even though these values exceed already the theoretical upper bound of the range of step size). All the number of iterations of Algorithms 1 and 2 with different step sizes are reported in Table 1.

Data in Table 1 shows 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 proved 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 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 α_k^* is too computationally expensive for Algorithm 1.

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

	Algorithm 1	Algorithm 2		Algorithm 2 with excessive step size								
n	$\alpha_k = \alpha_k^*$	$\alpha = \frac{1}{n^2+1}$	$\alpha = 2(1 - \sqrt{\frac{n^2}{n^2+1}})$	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

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 $2\big(1 - n\sqrt{1/(n^2+1)}$ 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 2 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 much. In any case, CPLEX is a well commercialized package that is particularly efficient for LPs 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.

Table 2. comparison of Algorithm 1 and CPLEX for medium-scale cases of (7.5)

n	Algorithm 1 with $\gamma = 1$ No. It	Algorithm 1 with $\gamma = 1$ CPU Sec.	CPLEX CPU Sec.	Optimal Objective 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

8 Conclusions

We consider embedding a full Jacobian decomposition into the augmented Lagrangian method (ALM) for solving a convex minimization model with linear constraints and an objective function in form of the sum of m functions without coupled variables. We find an example showing that the s-traightforward 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 an under-relaxation step. Furthermore, we show that the range of the step size of the under-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 a nonergodic senses. We finally test an assignment problem and show that the proposed algorithms designed in generic setting are even competitive 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 towards a publicized or commercialized version.

Appendix: 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

$$\tilde{x}^k = x^k - A^T\big(Ax^k - \lambda^k\big), \tag{8.9a}$$

and

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

Substituting (8.9a) into the above equation and using $AA^T = 2$, we get

$$\tilde{\lambda}^k = Ax^k - \lambda^k. \tag{8.9b}$$

Putting (8.9a) and (8.9b) together, using $A^T = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $A^TA = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, the predictor form (8.9) becomes

$$\begin{pmatrix} \tilde{x}_1^k \\ \tilde{x}_2^k \\ \tilde{\lambda}^k \end{pmatrix} = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} x_1^k \\ x_2^k \\ \lambda^k \end{pmatrix}. \tag{8.10}$$

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 (8.10), we have

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

Using the formula (8.10) and $w^{k+1} = \tilde{w}^k$, we obtain by induction that

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

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 certain under-relaxation step is a must to ensure its convergence.

References

[1] E. Blum and W. Oettli. *Mathematische Optimierung.* Springer-Verlag, Berlin, 1975. Grundlagen und Verfahren, Mit einem Anhang "Bibliographie zur Nichtlinearer Programmierung", Ökonometrie und Unternehmensforschung, No. XX.

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

[3] 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*, manuscript, 2013.

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

[5] J. Eckstein and M. Fukushima, Some reformulation and applications of the alternating direction method of multipliers, *Large Scale Optimization: State of the Art*, W. W. Hager *et al* eds., Kluwer Academic Publishers, pp. 115-134, 1994.

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

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

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

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

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

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

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

[13] D. R. Han and X. M. Yuan, *A note on the alternating direction method of multipliers*, J. Optim. Theory Appl., 155 (1) (2012), pp. 227-238.

[14] 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., to appear.

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

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

[17] 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.

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

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

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

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

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

[23] 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.

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

[25] B. Martinet, *Regularisation, d'inéquations variationelles par approximations succesives*, Rev. Francaise d'Inform. Recherche Oper., 4 (1970), pp. 154-159.

[26] A. S. Nemirovsky and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons, New York, 1983.

[27] 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.

[28] Y. E. Nesterov, *Gradient methods for minimizing composite objective function*, Math. Prog., Ser. B, 140 (2013), pp. 125C161.

[29] Y. G. Peng, A. Ganesh, J. Wright, W. L. Xu and Y. Ma, *Robust alignment by sparse and low-rank decomposition for linearly correlated images*, IEEE Con. Comput. Vision Patt. Recog. (CVPR), pp. 763-770, 2010.

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

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

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

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

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