Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 10, No. 2, pp. 359–384, April 2014

SOME PARALLEL SPLITTING METHODS FOR SEPARABLE CONVEX PROGRAMMING WITH THE $O(\frac{1}{t})$ CONVERGENCE RATE

Min Tao*

Abstract: This paper considers the separable convex programming problem with linking linear constraints of which objective function is formed as the sum of m individual functions without overlapping variables. As the special case with $m = 2$ has been well studied in the literature, we are particularly interested in the general case with $m \geq 3$. With the philosophy of exploring the structure of the involved function components individually, we develop some splitting type methods by decomposing the original problem into m smaller and potentially easier ones at each iteration. Our new methods distinguish from the existing methods mainly in that all the decomposed subproblems are completely tailored for simultaneous computation. Moreover, we show that the convergence rate of the proposed methods is $O(\frac{1}{t})$ in an ergodic sense. Finally, we apply the new parallel splitting methods to solve the matrix decomposition problem and video surveillance problem. Extensive numerical results illustrate that our new methods are competitive with state-of-the-art methods.

Key words: *convex programming, separable structure, splitting methods, parallel computation*

Mathematics Subject Classification: *90C25, 90C33, 65K05*

1 Introduction

Consider the following separable convex programming problem with linking linear constraints:

$$\min\left\{\sum_{i=1}^{m}\theta_i(x_i)\ \middle|\ \sum_{i=1}^{m}A_ix_i = b;\ x_i \in \mathcal{X}_i \subseteq \mathbb{R}^{n_i},\ i = 1, 2, \ldots, m\right\}, \tag{1.1}$$

where $\theta_i : \mathbb{R}^{n_i} \to \mathbb{R}$ $(i = 1, 2, \ldots, m)$ are closed proper convex functions and they are not necessarily smooth; $A_i \in \mathbb{R}^{l\times n_i}$ $(i = 1, 2, \ldots, m)$; $\mathcal{X}_i$ $(i = 1, 2, \ldots, m)$ are closed convex sets; $b \in \mathbb{R}^l$ and $\sum_{i=1}^m n_i = n$. Throughout, we assume that the solution set of (1.1) is nonempty.

Problem (1.1) arises in many applications. To mention a few of applications of (1.1) with $m \geq 3$, the fused Lasso problem in [20] and the robust batch image alignment problem in [16] are both in the form of (1.1) with $m = 3$; the constrained total-variation superresolution image reconstruction problem [1] can be easily reformulated into (1.1) with $m = 4$; the multistage stochastic programming problem in [14] is the case of (1.1) with $m = 8$. As most of these applications involve large-scale computation, the task of developing some algorithms for solving (1.1) has attracted much attention lately.

*This author was supported by the Natural Science Foundation of China NSFC-11301280.

The special case of (1.1) with $m = 2$ has been well studied in the literature, and the most influential methods for this special case might be the alternating direction method of multiplier (ADMM) [11, 12, 13] and the proximal-based decomposition method (PCPM) [4]. Compared to the rich literature for the case $m = 2$ [11, 12, 13, 4], the available work on the general case of (1.1) with $m \geq 3$ is significantly limited despite that the general case captures an even broader spectrum of applications in various fields.

Although the classical augmented Lagrangian method (ALM) proposed in [9, 17] can be applied to solve (1.1), a direct application of ALM will treat (1.1) as a general convex programming problem, and all the individual variables x_i's are dealt with in a single minimization problem. Thus, the nice separable structure in the objective function which might be very beneficial for algorithmic design is completely ignored. With the aim of taking advantage of the separable structure in the objective function, a predominant strategy is to decompose the augmented Lagrangian function of (1.1) into m subproblems such that the ith subproblem only involves $\theta_i(x_i)$. An instant idea is to extend the ADMM for (1.1) with $m = 2$ to the general case $m \geq 3$, and it yields the following ADMM-like splitting scheme:

$$\begin{cases} x_1^{k+1} = \arg\min \left\{\theta_1(x_1) - (\lambda^k)^\top p_1^k(x_1) + \frac{1}{2}\|p_1^k(x_1)\|_H^2 \;\middle|\; x_1 \in \mathcal{X}_1\right\}; \\ \quad \cdots\cdots \quad \cdots\cdots \\ x_i^{k+1} = \arg\min \left\{\theta_i(x_i) - (\lambda^k)^\top p_i^k(x_i) + \frac{1}{2}\|p_i^k(x_i)\|_H^2 \;\middle|\; x_i \in \mathcal{X}_i\right\}; \\ \quad \cdots\cdots \quad \cdots\cdots \\ x_m^{k+1} = \arg\min \left\{\theta_m(x_m) - (\lambda^k)^\top p_m^k(x_m) + \frac{1}{2}\|p_m^k(x_m)\|_H^2 \;\middle|\; x_m \in \mathcal{X}_m\right\}; \\ \lambda^{k+1} = \lambda^k - H(\sum_{j=1}^m A_j x_j^{k+1} - b), \end{cases} \tag{1.2}$$

where

$$p_i^k(x_i) = \sum_{j=1}^{i-1} A_j x_j^{k+1} + A_i x_i + \sum_{j=i+1}^{m} A_j x_j^k - b, \quad i = 1, \ldots, m,$$

and superscript $\top$ is the transpose operator, and $H \succ 0$ is a penalty matrix. The benefit of introducing matrix penalty instead of constant penalty is to permit a flexible strategy of enforcing different level of penalty. However, the convergence of the direct extension of ADMM (1.2) is still open. Recently, the lack of convergence of (1.2) has inspired some efforts in the prediction-correction fashion for solving (1.1), whose main idea is to generate the new iterate via correcting the output of (1.2), see e.g. [7].

Note that the subproblems arising in (1.2) need to be solved sequentially, i.e., in order to compute x_i^{k+1}, one must first compute $x_1^{k+1}, x_2^{k+1}, \ldots, x_{i-1}^{k+1}$. This creates an obstacle to parallelization. With the advent of the big data era, an important and interesting problem is to develop parallel splitting methods for solving (1.1) where the resulting subproblems are tailored completely for parallel computation. By decomposing the corresponding augmented Lagrangian function of (1.1) in a parallel manner, we can easily derive the following procedure for obtaining the x_i^{k+1}'s:

$$\begin{cases} x_1^{k+1} = \arg\min \left\{\theta_1(x_1) - (\lambda^k)^\top q_1^k(x_1) + \frac{\beta}{2}\|q_1^k(x_1)\|^2 \;\middle|\; x_1 \in \mathcal{X}_1\right\}; \\ \quad \cdots\cdots \quad \cdots\cdots \\ x_i^{k+1} = \arg\min \left\{\theta_i(x_i) - (\lambda^k)^\top q_i^k(x_i) + \frac{\beta}{2}\|q_i^k(x_i)\|^2 \;\middle|\; x_i \in \mathcal{X}_i\right\}; \\ \quad \cdots\cdots \quad \cdots\cdots \\ x_m^{k+1} = \arg\min \left\{\theta_m(x_m) - (\lambda^k)^\top q_m^k(x_m) + \frac{\beta}{2}\|q_m^k(x_m)\|^2 \;\middle|\; x_m \in \mathcal{X}_m\right\}; \\ \lambda^{k+1} = \lambda^k - \beta(\sum_{j=1}^m A_j x_j^{k+1} - b), \end{cases} \tag{1.3}$$

where

$$q_i^k(x_i) = \sum_{j=1, j\neq i}^{m} A_j x_j^k + A_i x_i - b. \quad i = 1, \ldots, m.$$

However, compared to the adequate literature of alternating splitting augmented-Lagrangian-based methods for (1.1) (especially when $m = 2$), numerical development on parallel splitting augmented-Lagrangian-based methods for (1.1), even for $m = 2$, is in its infancy. For existing parallel splitting augmented-Lagrangian-based methods for (1.1), we refer to [4, 6, 19] for the case $m = 2, 3$. More recently, in [5], a parallel splitting method for (1.1) with general m was developed in the prediction-correction fashion. This parallel splitting method differs from the prediction-correction method in [7], in that the predictor is generated by the parallel-oriented scheme (1.3) rather than the alternating-oriented scheme (1.2), and that the structure of correction step is much easier than that of [7]. Note that all the splitting methods in [7, 6, 19, 5] require correction step to ensure convergence. The correction step, however, may cause critical difficulties for some concrete applications of (1.1). As shown in the numerical example of recovering sparse and low-rank matrices with incomplete and noisy observation in [18], the correction step may ruin the low-rank characteristic of recovered components obtained by the prediction step and end up with high-rank iterates after implementing some correction steps. Motivated by these applications, a natural question thus arises: Is it possible to develop splitting methods for solving the general case of (1.1) without any correction step, while the decomposed subproblems are completely tailored for parallel computation?

As our first contribution, we answer this question affirmatively by proposing four parallel splitting methods for solving (1.1) that do not require a correction step. Note that the method in [8] for (1.1) also requires no correction step, but its resulting subproblems at each iteration are not eligible for completely parallel computation. Our second contribution is to establish the global convergence and to show the $O(\frac{1}{t})$ convergence rate in an ergodic sense of the proposed methods. The results provide some theoretical justification of the witnessed empirical efficiency.

The rest of this paper is organized as follows. In Section 2, we provide some useful preliminaries for later analysis. In Section 3, four distinct versions of parallel splitting methods are presented. In Section 4, the convergence analysis for these methods are provided. Then, we show that the convergence rate is $O(\frac{1}{t})$ in an ergodic sense in Section 5. In order to verify the efficiency of our proposed methods, we report the numerical performance for the matrix decomposition and video surveillance problems. Numerical comparisons with some existing efficient methods are reported in Section 6. Finally, some conclusions are drawn in Section 7.

2 Preliminaries

In this section, we summarize some basic definitions and related properties that will be used later in our analysis.

$\boxed{2.1}$ Variational Characterization

Let $\mathcal{W} := \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m \times \mathbb{R}^l$. By deriving its optimality condition, it is easy to see that (1.1) is equivalent to finding $w^* = (x_1^*, x_2^*, ..., x_m^*, \lambda^*) \in \mathcal{W}$ such that

$$\begin{cases} \theta_1(x_1) - \theta_1(x_1^*) + (x_1 - x_1^*)^\top(-A_1^\top \lambda^*) \geq 0, \\ \theta_2(x_2) - \theta_2(x_2^*) + (x_2 - x_2^*)^\top(-A_2^\top \lambda^*) \geq 0, \\ \quad\cdots\cdots \qquad\qquad \cdots\cdots \\ \theta_m(x_m) - \theta_m(x_m^*) + (x_m - x_m^*)^\top(-A_m^\top \lambda^*) \geq 0, \\ \sum_{i=1}^m A_i x_i^* - b = 0, \end{cases} \qquad \forall\, w = (x_1, x_2, \ldots, x_m, \lambda) \in \mathcal{W},$$

or, in a more compact form:

$$\mathrm{VI}(\mathcal{W}, F, \theta) \qquad \theta(u) - \theta(u^*) + (w - w^*)^\top F(w^*) \geq 0, \quad \forall\, w \in \mathcal{W}, \tag{2.1a}$$

where

$$u = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \quad \theta(u) = \sum_{i=1}^m \theta_i(x_i), \quad w = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ \lambda \end{pmatrix} \quad \text{and} \quad F(w) = \begin{pmatrix} -A_1^\top \lambda \\ -A_2^\top \lambda \\ \vdots \\ -A_m^\top \lambda \\ \sum_{i=1}^m A_i x_i - b \end{pmatrix}. \tag{2.1b}$$

Note that u collects all the primal variables in (1.1) and it is a sub-vector of w. We have the following lemma regarding $F(w)$ defined above. We omit its proof since it is trivial.

Lemma 2.1. *The mapping $F(w)$ defined in (2.1b) satisfies*

$$(w' - w)^\top (F(w') - F(w)) = 0, \quad \forall\, w', w \in \mathbb{R}^{n+l}.$$

Under the assumption that the solution set of (1.1) is nonempty, the solution set of $\mathrm{VI}(\mathcal{W}, F, \theta)$, which denotes by $\mathcal{W}^*$, is also nonempty and convex (see Theorem 2.3.5 in [10]). Moreover, the following theorem provides a description of $\mathcal{W}^*$, and it is inspired by Theorem 2.3.5 in [10].

Theorem 2.2. *The solution set of VI$(\mathcal{W}, F, \theta)$ is convex and it can be characterized as*

$$\mathcal{W}^* = \bigcap_{w \in \mathcal{W}} \big\{\tilde{w} \in \mathcal{W} : \theta(u) - \theta(\tilde{u}) + (w - \tilde{w})^\top F(w) \geq 0\big\}. \tag{2.2}$$

Based on Theorem 2.2, we present the definition of *ϵ-approximate solution* in the following.

Definition 2.3. $\tilde{w} \in \mathcal{W}$ is an ϵ-approximate solution of $\mathrm{VI}(\mathcal{W}, F, \theta)$ if it satisfies

$$\theta(\tilde{u}) - \theta(u) + (\tilde{w} - w)^\top F(w) \leq \epsilon, \forall w \in \mathcal{W}.$$

The identity summarized in the following lemma is useful in the convergence analysis. We omit the proof which is very elementary.

Lemma 2.4. *Let $D \in \mathbb{R}^{n \times n}$ be symmetric and positive definite. Then, it holds that*

$$(a - b)^\top D(c - d) = \frac{1}{2}\big(\|a - d\|_D^2 - \|a - c\|_D^2\big) + \frac{1}{2}\big(\|c - b\|_D^2 - \|d - b\|_D^2\big) \qquad \forall\, a, b, c, d \in \mathbb{R}^n,$$

where $\|x\|_D^2$ represents $x^\top Dx$ for any vector $x \in \mathbb{R}^n$.

$\boxed{2.2}$ Some Notations

We define some matrices which will simplify our notations significantly in the later analysis. For $m \geq 3$, a block diagonal matrix is defined as

$$G_r = \text{diag}\{r_1 A_1^\top H A_1, r_2 A_2^\top H A_2, \ldots, r_m A_m^\top H A_m, H^{-1}\}, \tag{2.3}$$

where $r = (r_1, \ldots, r_m) \in \mathbb{R}_+^m$. Note that the matrix G_r defined in (2.3) is positive definite under the assumption that A_i's $(i = 1, \ldots, m)$ are of full column rank and H is positive definite. We use e to denote the vector with all entries equal to 1. More specifically,

$$G_e = \text{diag}\{A_1^\top H A_1, A_2^\top H A_2, \ldots, A_m^\top H A_m, H^{-1}\}. \tag{2.4}$$

Four more matrices will be applied in the coming analysis:

$$M = \begin{pmatrix} r_1 A_1^\top H A_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & r_m A_m^\top H A_m & 0 \\ -A_1 & \cdots & -A_m & H^{-1} \end{pmatrix}_{p \times p},$$

$$Q = \begin{pmatrix} I & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & I & 0 \\ -HA_1 & \cdots & -HA_m & I \end{pmatrix}_{p \times p}, \tag{2.5}$$

and

$$N = \begin{pmatrix} A_1^\top H A_1 & 0 & \cdots & A_1^\top \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & A_m^\top H A_m & A_m^\top \\ 0 & \cdots & 0 & \eta H^{-1} \end{pmatrix}_{p \times p},$$

$$N^{-\top} G_e = \begin{pmatrix} I & 0 & \cdots & 0 \\ 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{\eta} H A_1 & \cdots & -\frac{1}{\eta} H A_m & \frac{1}{\eta} I \end{pmatrix}_{p \times p}, \tag{2.6}$$

where η is a positive parameter and $p = n + l$. The matrices A_i $(i = 1, \ldots, m)$ are assumed to be of full column rank. Thus, all the matrices $A_i^\top H A_i$'s are nonsingular. The matrix N defined in (2.6) is also nonsingular block upper-triangular, and hence $N^{-\top} G_e$ is well defined. Note the following relation between these matrices:

$$M = G_r \cdot Q, \qquad G_e = N^\top \cdot (N^{-\top} G_e). \tag{2.7}$$

Finally, we summarize some facts regarding the matrices defined in (2.3) - (2.6) in the following.

Lemma 2.5. *Let the matrix* $\Xi \in \mathbb{R}^{(m+1) \times (m+1)}$ *be defined as*

$$\Xi = \begin{pmatrix} r_1 - 1 & -1 & \cdots & -1 & 0 \\ -1 & r_2 - 1 & \cdots & -1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & -1 & \cdots & r_m - 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Then Ξ is positive definite if and only if

$$\sum_{i=1}^{m} \frac{1}{r_i} < 1. \tag{2.8}$$

Proof. Note the following identity:

$$\Xi = \mathcal{U} \cdot \Lambda \cdot \mathcal{U}^\top,$$

where

$$\mathcal{U} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \quad \text{and} \quad \Lambda = \begin{pmatrix} r_1 & 0 & \cdots & 0 & -1 \\ 0 & r_2 & \cdots & 0 & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & r_m & -1 \\ -1 & -1 & \cdots & -1 & 1 \end{pmatrix}.$$

Obviously, the ith $(i = 1, \ldots, m)$ order sequential minor of matrix Λ are positive. And the $(m+1)$th order sequential minor is equal to

$$\det(\Lambda) = (\prod_{i=1}^{m} r_i)(1 - \sum_{i=1}^{m} \frac{1}{r_i}).$$

Since $r_i > 0$ $(i = 1, \ldots, m)$, $\det(\Lambda) > 0$ if and only if $(1 - \sum_{i=1}^{m} \frac{1}{r_i}) > 0$. Therefore, Λ is positive definite if and only if $\sum_{i=1}^{m} \frac{1}{r_i} < 1$, i.e., condition (2.8) holds. Because the matrix $\mathcal{U}$ is nonsingular, the matrix Ξ is positive definite if and only if the matrix Λ is positive definite. Hence, the positive definiteness of the matrix Ξ is ensured by the condition (2.8). □

Remark 2.6. If $r_i \equiv r$ $(i = 1, \ldots, m)$, the condition (2.8) is reduced to $r > m$.

Lemma 2.7. *Let the matrices G_e, M, Q and N be defined in (2.4), (2.5) and (2.6), respectively. Then,*
1) *The matrix*

$$S_1 = M + M^\top - Q^\top G_r Q \tag{2.9}$$

is positive definite if and only if the condition (2.8) holds;
2) *The matrix*

$$S_2 = N + N^\top - G_e \tag{2.10}$$

is positive definite if and only if $\eta > \frac{m+1}{2}$

Proof. For the assertion 1), we notice that

$$\begin{aligned} S_1 &= M + M^\top - Q^\top G_r Q \\ &= \operatorname{diag}\{r_1 A_1^\top H A_1, r_2 A_2^\top H A_2, \ldots, r_m A_m^\top H A_m, H^{-1}\} - \begin{pmatrix} A_1^\top \\ \vdots \\ A_m^\top \\ 0 \end{pmatrix} H(A_1, \ldots, A_m, 0) \end{aligned}$$

$$
\begin{aligned}
&= \mathcal{A}^\top \cdot \begin{pmatrix} r_1 I_l & 0_l & \cdots & 0_l & 0_l \\ 0_l & r_2 I_l & \cdots & 0 & 0_l \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_l & 0_l & \cdots & r_m I_l & 0_l \\ 0_l & 0_l & \cdots & 0_l & I_l \end{pmatrix} \cdot \mathcal{A} - \mathcal{A}^\top \cdot \begin{pmatrix} I_l & I_l & \cdots & I_l & 0_l \\ I_l & I_l & \cdots & I_l & 0_l \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ I_l & I_l & \cdots & I_l & 0_l \\ 0_l & 0_l & \cdots & 0_l & 0_l \end{pmatrix} \cdot \mathcal{A} \\
&= \mathcal{A}^\top \cdot \begin{pmatrix} (r_1-1)I_l & -I_l & \cdots & -I_l & 0_l \\ -I_l & (r_2-1)I_l & \cdots & -I_l & 0_l \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -I_l & -I_l & \cdots & -I_l & 0_l \\ 0_l & 0_l & \cdots & 0_l & I_l \end{pmatrix} \cdot \mathcal{A} \\
&= \mathcal{A}^\top \cdot (\Xi \otimes I_l) \cdot \mathcal{A},
\end{aligned}
$$

where $\mathcal{A} = \mathrm{diag}\{H^{1/2}A_1, \ldots, H^{1/2}A_m, H^{-1/2}\}$. The matrix $\mathcal{A}$ is of full column rank, and $\Xi \otimes I_l$ has the same eigenvalues as the matrix Ξ. Thus, the matrix S_1 is positive definite if and only if the matrix Ξ is positive definite. In view of Lemma 2.5, the first assertion follows immediately.

For the assertion 2), we note that

$$
\begin{aligned}
S_2 &= N + N^\top - G_e \\
&= \begin{pmatrix} A_1^\top H A_1 & 0 & \cdots & 0 & A_1^\top \\ 0 & A_2^\top H A_2 & \ddots & 0 & A_2^\top \\ \vdots & \dots & \ddots & \vdots & \vdots \\ 0 & \cdots & \dots & A_m^\top H A_m & A_m^\top \\ A_1 & A_2 & \cdots & A_m & (2\eta-1)H^{-1} \end{pmatrix} \\
&= \mathcal{A}^\top \cdot (\Upsilon \otimes I_l) \cdot \mathcal{A},
\end{aligned}
$$

where

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

Thus, the positive definiteness of S_2 is equivalent to the positive definiteness of the matrix Υ, and it is equivalent to $\eta > (m+1)/2$. □

3 The New Parallel Methods for Solving (1.1)

In this section, we present four different versions of parallel splitting methods for (1.1) and prove some elementary properties for the coming convergence analysis.

3.1 Algorithm 1

Let us revisit the classical ALM, so that the relationship of the proposed algorithm with ALM will be clear later. When the classical ALM is applied to (1.1), it yields the following scheme:

$$\left\{\begin{array}{l}\begin{pmatrix} x_1^{k+1} \\ \vdots \\ x_m^{k+1}\end{pmatrix} = \arg\min_{\{x_i\in\mathcal{X}_i, i=1,\dots,m\}}\{\sum_{i=1}^m \theta_i(x_i) - \langle\lambda^k, \sum_{i=1}^m A_i x_i\rangle \\ \qquad\qquad\qquad +\frac{1}{2}\|\sum_{i=1}^m A_i x_i - b\|_H^2\}, \\ \lambda^{k+1} = \lambda^k - H(\sum_{j=1}^m A_j x_j^{k+1} - b).\end{array}\right. \tag{3.1}$$

Note that the subproblem involving $(x_1, x_2, \dots, x_m)$ in (3.1) requires an inner minimization to obtain an approximate solution. Hence, the direct application of ALM may ignore the nice separable structure of problem (1.1). However, it is possible to utilize the properties of each θ_i individually when we apply the classical proximal point algorithm (PPA) to regularize the subproblem involving $(x_1, x_2, \dots, x_m)$, i.e.,

$$\left\{\begin{array}{l}\begin{pmatrix} x_1^{k+1} \\ \vdots \\ x_m^{k+1}\end{pmatrix} = \arg\min_{\{x_i\in\mathcal{X}_i, i=1,\dots,m\}}\left\{\sum_{i=1}^m \theta_i(x_i) - \langle\lambda^k, \sum_{i=1}^m A_i x_i\rangle\right. \\ \qquad\left. +\frac{1}{2}\|\sum_{i=1}^m A_i x_i - b\|_H^2 + \frac{1}{2}\left\|\begin{pmatrix} x_1 \\ \vdots \\ x_m\end{pmatrix} - \begin{pmatrix} x_1^k \\ \vdots \\ x_m^k\end{pmatrix}\right\|_R^2\right\}, \\ \lambda^{k+1} = \lambda^k - H(\sum_{j=1}^m A_j x_j^{k+1} - b),\end{array}\right. \tag{3.2}$$

where $R \succ 0$ is a matrix to be determined later. Usually, the above scheme is referred to as the proximal-like augmented Lagrangian method (PL-ALM). By choosing matrix R appropriately, the first subproblem in (3.2) could be decomposed into m dependent subproblems without overlapping variables. Indeed, note that

$$\begin{aligned}\tfrac{1}{2}\|\sum_{i=1}^m A_i x_i - b\|_H^2 = \tfrac{1}{2}\|\sum_{i=1}^m A_i x_i^k - b\|_H^2 + \left\langle \begin{pmatrix} A_1^\top \\ \vdots \\ A_m^\top\end{pmatrix} H(\sum_{i=1}^m A_i x_i^k - b), \begin{pmatrix} x_1 - x_1^k \\ \vdots \\ x_m - x_m^k\end{pmatrix}\right\rangle \\ +\tfrac{1}{2}\left\|\begin{pmatrix} x_1 - x_1^k \\ \vdots \\ x_m - x_m^k\end{pmatrix}\right\|_\Upsilon^2,\end{aligned}$$

where

$$\Upsilon = \begin{pmatrix} A_1^\top \\ \vdots \\ A_m^\top\end{pmatrix} H(A_1, \dots, A_m). \tag{3.3}$$

By setting $R = G_r - \Upsilon$ (G_r defined in (2.3) and $r = (r_1, \dots, r_m)$ is chosen to satisfy the condition $R \succ 0$) in the first subproblem of (3.2), we can derive the following procedure:

$$\begin{cases} \begin{pmatrix} x_1^{k+1} \\ \vdots \\ x_m^{k+1} \end{pmatrix} = \arg\min_{\{x_i \in \mathcal{X}_i, i=1,\dots,m\}} \big\{ \sum_{i=1}^m \theta_i(x_i) - \langle \lambda^k, \sum_{i=1}^m A_i x_i \rangle \\ \quad + \left\langle \begin{pmatrix} A_1^\top \\ \vdots \\ A_m^\top \end{pmatrix} H(\sum_{i=1}^m A_i x_i^k - b), \begin{pmatrix} x_1 - x_1^k \\ \vdots \\ x_m - x_m^k \end{pmatrix} \right\rangle + \frac{1}{2} \left\| \begin{pmatrix} x_1 - x_1^k \\ \vdots \\ x_m - x_m^k \end{pmatrix} \right\|_{G_r}^2 \right\}, \\ \lambda^{k+1} = \lambda^k - H(\sum_{j=1}^m A_j x_j^{k+1} - b). \end{cases}$$

In view of the definition (2.3) of the matrix G_r, the above scheme is equivalent to

$$\begin{cases} x_1^{k+1} = \arg\min_{x_1 \in \mathcal{X}_1} \big\{ \theta_1(x_1) - \langle (\lambda^k - H(\sum_{i=1}^m A_i x_i^k - b)), A_1 x_1 \rangle \\ \qquad + \frac{r_1}{2} \left\| A_1(x_1 - x_1^k) \right\|_H^2 \big\}, \\ \qquad \cdots \quad \cdots \quad \cdots \quad \cdots \\ x_m^{k+1} = \arg\min_{x_m \in \mathcal{X}_m} \big\{ \theta_m(x_m) - \langle (\lambda^k - H(\sum_{i=1}^m A_i x_i^k - b)), A_m x_m \rangle \\ \qquad + \frac{r_m}{2} \left\| A_m(x_m - x_m^k) \right\|_H^2 \big\}, \\ \lambda^{k+1} = \lambda^k - H(\sum_{j=1}^m A_j x_j^{k+1} - b). \end{cases}$$

In this way, the subproblem involving $(x_1, x_2, \dots, x_m)$ in (3.1) is decomposed into m independent subproblems and thus each subproblem can be solved in a parallel manner. The resulted method, denoted as Algorithm 1a, for (1.1) generates the new iterate $w^{k+1} = (x_1^{k+1}, x_2^{k+1}, \dots, x_m^{k+1}, \lambda^{k+1})$ as follows.

Algorithm 1a: The $(k+1)$th iteration of the new parallel splitting method
Step 1. Update multiplier:

$$\hat{\lambda}^k = \lambda^k - H\big(\sum_{i=1}^m A_i x_i^k - b\big). \tag{3.4}$$

Step 2. Solve the following m subproblems (in parallel):

$$x_i^{k+1} := \text{argmin}\{\theta_i(x_i) - (\hat{\lambda}^k)^\top A_i x_i + \frac{r_i}{2} \|A_i(x_i - x_i^k)\|_H^2 \mid x_i \in \mathcal{X}_i\}, i = 1, 2, \dots, m. \tag{3.5}$$

Step 3. Update multiplier

$$\lambda^{k+1} = \lambda^k - H\big(\sum_{i=1}^m A_i x_i^{k+1} - b\big).$$

Remark 3.1. Algorithm 1a requires no correction step and all the subproblems in (3.5) are tailored for parallel computation. Compared to (1.2) and (1.3), each subproblem in (3.5) involves one function component $\theta_i(x_i)$ only and it is possible to take advantage of the structure of $\theta_i(x_i)$. In addition, if A_i is of full column rank, then the ith subproblem in (3.5) is strongly convex and thus a unique solution is guaranteed.

As observed in (3.5), the efficiency of Algorithm 1a heavily depends upon the efficient solvability of the subproblem involving x_i $(i = 1, \ldots, m)$, i.e.

$$\min_{x_i \in \mathcal{X}_i} \theta_i(x_i) + \frac{a_i}{2}\|A_i x_i - y\|^2, i = 1, \ldots, m, \tag{3.6}$$

where the positive scalar a_i and vector y are given. When the linear operator A_i is not the identity, the above subproblem may not preserve a closed-form solution. In fact, another splitting parallel method can be developed assuming that each subproblem (3.6) preserves a closed-form solution with $A_i = I$. The resulted method, denoted as Algorithm 1b, is also derived from PL-ALM (3.2) by setting the matrix R to be $diag\{\delta_1 I_{n_1}, \ldots, \delta_m I_{n_m}\} - \Upsilon$, where Υ is defined in (3.3). Note that Algorithm 1b inherits the advantage of Algorithm 1a, i.e., it can be implemented in a parallel manner. We indicate that the difference between Algorithm 1a and Algorithm 1b is the different setting of the matrix R in (3.2). However, Algorithm 1b will be more flexible to deal with the more general matrices A_i and more widely used in practice than Algorithm 1a when the given θ_i $(i = 1, \ldots, m)$ are of certain special structures. The proximal parameters δ_i's in Algorithm 1b are assumed to satisfy

$$\sum_{i=1}^{m} \frac{\rho(A_i^\top H A_i)}{\delta_i} < 1,$$

where $\rho(\cdot)$ denotes the spectral radius. Then, Algorithm 1b for (1.1) generates the new iterate $w^{k+1} = (x_1^{k+1}, x_2^{k+1}, \ldots, x_m^{k+1}, \lambda^{k+1})$ as follows.

Algorithm 1b: The $(k+1)$th iteration of the new parallel splitting method

Step 1. Update multiplier:

$$\hat{\lambda}^k = \lambda^k - H\big(\textstyle\sum_{i=1}^m A_i x_i^k - b\big),$$

Step 2. Solve the following m subproblems (in parallel):

$$x_i^{k+1} := \text{argmin}\{\theta_i(x_i) - (\hat{\lambda}^k)^\top A_i x_i + \frac{\delta_i}{2}\|x_i - x_i^k\|^2 \mid x_i \in \mathcal{X}_i\}, \quad i = 1, 2, \ldots, m. \tag{3.7}$$

Step 3. Update multiplier

$$\lambda^{k+1} = \lambda^k - H\big(\textstyle\sum_{i=1}^m A_i x_i^{k+1} - b\big).$$

Remark 3.2. Algorithm 1b can be viewed as an extension of the proximal-based decomposition method (PCPM) for (1.1) with $m = 2$ to $m \geq 3$. The PCPM generates the new iterate with the given $(x_1^k, x_2^k, \lambda^k)$ as follows:

$$\begin{cases} \hat{\lambda}^k = \lambda^k - \beta(A_1 x_1^k + A_2 x_2^k); \\ x_1^{k+1} = \arg\min_{x_1 \in \mathcal{X}_1}\{f_1(x_1) - \langle \hat{\lambda}^k, A_1 x_1 \rangle + \frac{1}{2\beta}\|x_1 - x_1^k\|^2\}; \\ x_2^{k+1} = \arg\min_{x_2 \in \mathcal{X}_2}\{f_2(x_2) - \langle \hat{\lambda}^k, A_2 x_2 \rangle + \frac{1}{2\beta}\|x_2 - x_2^k\|^2\}; \\ \lambda^{k+1} = \lambda^k - \beta(A_1 x_1^{k+1} + A_2 x_2^{k+1}), \end{cases}$$

where $0 < \epsilon \leq \beta \leq \min(\frac{1-\epsilon}{2\|A_1\|}, \frac{1-\epsilon}{2\|A_2\|})$.

The convergence analysis for Algorithm 1b can be developed in a similar way to Algorithm 1a. Hence, we focus on Algorithm 1a and omit the details for Algorithm 1b for succinctness. For convenience of further analysis, we use the sequence $\{w^k\}$ generated by the proposed Algorithm 1a to construct an auxiliary sequence via

$$\hat{w}^k = \begin{pmatrix} \hat{x}_1^k \\ \hat{x}_2^k \\ \vdots \\ \hat{x}_m^k \\ \hat{\lambda}^k \end{pmatrix} = \begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ \vdots \\ x_m^{k+1} \\ \lambda^k - H\big(\sum_{i=1}^m A_i x_i^k - b\big) \end{pmatrix}. \tag{3.8}$$

With the notation Q defined in (2.5), we can get

$$w^{k+1} = w^k - Q(w^k - \hat{w}^k). \tag{3.9}$$

Recall the characterization of $\mathcal{W}^*$ in (2.2), the following theorem reflects the discrepancy of $\hat{w}^k$ from a solution point in $\mathcal{W}^*$.

Theorem 3.3. *Let $\{w^k\}$ be generated by Algorithm 1a and $\{\hat{w}^k\}$ be given in (3.8). Then, we have*

$$\hat{w}^k \in \mathcal{W},\ \theta(u) - \theta(\hat{u}^k) + (w - \hat{w}^k)^\top F(\hat{w}^k) + (w - \hat{w}^k)^\top G_r Q(\hat{w}^k - w^k) \geq 0,\ \forall\, w \in \mathcal{W}, \tag{3.10}$$

where G_r and Q are defined in (2.3) and (2.5), respectively.

Proof. First, it follows from (3.4) and the fact $x_i^{k+1} = \hat{x}_i^k$ $(i = 1, \dots, m)$ that

$$(\lambda - \hat{\lambda}^k)^\top\{(A_1\hat{x}_1^k + \cdots + A_m\hat{x}_m^k - b) - \sum_{i=1}^m A_i(\hat{x}_i^k - x_i^k) + H^{-1}(\hat{\lambda}^k - \lambda^k)\} \geq 0.$$

On the other hand, according to the optimality condition of the x_i-subproblem

$$\hat{x}_i^k \in \mathcal{X}_i,\ \theta(x_i) - \theta(\hat{x}_i^k) + (x_i - \hat{x}_i^k)^\top(-A_i^\top\hat{\lambda}^k) + (x_i - \hat{x}_i^k)^\top r_i A_i^\top H A_i(\hat{x}_i^k - x_i^k) \geq 0,\ \forall x_i \in \mathcal{X}_i.$$

Consequently, it follows from the above two inequalities that

$$\begin{cases} \theta(x_1) - \theta(\hat{x}_1^k) + (x_1 - \hat{x}_1^k)^\top(-A_1^\top\hat{\lambda}^k) + (x_1 - \hat{x}_1^k)^\top r_1 A_1^\top H A_1(\hat{x}_1^k - x_1^k) \geq 0, \\ \theta(x_2) - \theta(\hat{x}_2^k) + (x_2 - \hat{x}_2^k)^\top(-A_2^\top\hat{\lambda}^k) + (x_2 - \hat{x}_2^k)^\top r_2 A_2^\top H A_2(\hat{x}_2^k - x_2^k) \geq 0, \\ \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ \theta(x_m) - \theta(\hat{x}_m^k) + (x_m - \hat{x}_m^k)^\top(-A_m^\top\hat{\lambda}^k) + (x_m - \hat{x}_m^k)^\top r_m A_m^\top H A_m(\hat{x}_m^k - x_m^k) \geq 0, \\ (\lambda - \hat{\lambda}^k)^\top\big\{A_1\hat{x}_1^k + A_2\hat{x}_2^k + \cdots + A_m\hat{x}_m^k - b - H^{-1}(\lambda^k - \hat{\lambda}^k) - \sum_{i=1}^m A_i(\hat{x}_i^k - x_i^k)\big\} \geq 0. \\ \hfill \forall w \in \mathcal{W}. \end{cases}$$

Adding all these inequalities together and using the definitions of F in (2.1b), and M in (2.5), it leads that

$$\hat{w}^k \in \mathcal{W}, \quad \theta(u) - \theta(\hat{u}^k) + (w - \hat{w}^k)^\top F(\hat{w}^k) + (w - \hat{w}^k)^\top M(\hat{w}^k - w^k) \geq 0, \quad \forall\, w \in \mathcal{W}.$$

Recall that $M = G_r Q$ (see (2.7)), the assertion (3.10) follows immediately. □

Lemma 3.4. *Let $\{w^k\}$ be generated by Algorithm 1a and $\{\hat{w}^k\}$ be given in (3.8). Then, we have*

$$\|w^k - \hat{w}^k\|_{G_r}^2 - \|w^{k+1} - \hat{w}^k\|_{G_r}^2 = \|w^k - \hat{w}^k\|_{S_1}^2,$$

where G_r and S_1 are defined in (2.3) and (2.9), respectively.

Proof. First, combining (3.9), we have

$$\begin{aligned}\|w^k - w\|_{G_r}^2 - \|w^{k+1} - w\|_{G_r}^2 &= \|w^k - w\|_{G_r}^2 - \|w^k - Q(w^k - \hat{w}^k) - w\|_{G_r}^2 \\ &= 2(w^k - w)^\top G_r Q(w^k - \hat{w}^k) - \|Q(w^k - \hat{w}^k)\|_{G_r}^2 \\ &= 2(w^k - w)^\top M(w^k - \hat{w}^k) - \|Q(w^k - \hat{w}^k)\|_{G_r}^2. \quad (3.11)\end{aligned}$$

The last equality follows from the fact $M = G_r Q$. Then, setting $w = \hat{w}^k$ in (3.11), we have

$$\begin{aligned}\|w^k - \hat{w}^k\|_{G_r}^2 - \|w^{k+1} - \hat{w}^k\|_{G_r}^2 &= 2(w^k - \hat{w}^k)^\top M(w^k - \hat{w}^k) - \|Q(w^k - \hat{w}^k)\|_{G_r}^2 \\ &= (w^k - \hat{w}^k)^\top (M + M^\top)(w^k - \hat{w}^k) - \|Q(w^k - \hat{w}^k)\|_{G_r}^2 \\ &= \|w^k - \hat{w}^k\|_{(M+M^\top - Q^\top G_r Q)}^2 \\ &= \|w^k - \hat{w}^k\|_{S_1}^2. \quad (3.12)\end{aligned}$$

The last equality follows from the definition of matrix S_1 (see (2.9)). Thus the assertion follows directly. □

Theorem 3.5. *Let $\{w^k\}$ be generated by Algorithm 1a and $\{\hat{w}^k\}$ be given in (3.8). Then, we have*

$$\begin{aligned}\theta(u) - \theta(\hat{u}^k) + (w - \hat{w}^k)^\top F(\hat{w}^k) + \frac{1}{2}(\|w - w^k\|_{G_r}^2 - \|w - w^{k+1}\|_{G_r}^2) \\ \geq \frac{1}{2}\|\hat{w}^k - w^k\|_{S_1}^2, \ \forall\, w \in \mathcal{W}, \quad (3.13)\end{aligned}$$

where G_r and S_1 are defined in (2.3) and (2.9), respectively.

Proof. First, by using the relationship in (3.9), it follows from (3.10) that

$$\hat{w}^k \in \mathcal{W}, \ \theta(u) - \theta(\hat{u}^k) + (w - \hat{w}^k)^\top F(\hat{w}^k) + (w - \hat{w}^k)^\top G_r(w^{k+1} - w^k) \geq 0, \ \forall\, w \in \mathcal{W}. \quad (3.14)$$

In view of Lemma 2.4, we have

$$\begin{aligned}(w - \hat{w}^k)^\top G_r(w^{k+1} - w^k) =& \frac{1}{2}\big(\|w - w^k\|_{G_r}^2 - \|w - w^{k+1}\|_{G_r}^2\big) \\ &+ \frac{1}{2}\big(\|\hat{w}^k - w^{k+1}\|_{G_r}^2 - \|\hat{w}^k - w^k\|_{G_r}^2\big).\end{aligned}$$

Substituting the above identity into (3.14), we have

$$\begin{aligned}\hat{w}^k \in \mathcal{W}, \quad & \theta(u) - \theta(\hat{u}^k) + (w - \hat{w}^k)^\top F(\hat{w}^k) + \frac{1}{2}\big(\|w - w^k\|_{G_r}^2 - \|w - w^{k+1}\|_{G_r}^2\big) \\ & \geq \frac{1}{2}\big(\|\hat{w}^k - w^k\|_{G_r}^2 - \|\hat{w}^k - w^{k+1}\|_{G_r}^2\big) \\ & = \frac{1}{2}\|w^k - \hat{w}^k\|_{S_1}^2, \quad \forall\, w \in \mathcal{W}.\end{aligned}$$

The last equality follows from (3.12). Hence, the assertion (3.13) follows immediately. □

3.2 Algorithm 2

In this section, we present another two parallel splitting methods for (1.1). In order to alleviate the difficulty of setting m parameters in Algorithm 1a, we propose another parallel algorithm, denoted as Algorithm 2a, by setting $r_i = 1$ $(i = 1, \ldots, m)$ in Algorithm 1a. However, another parameter η in the multiplier updating (see (3.16)) step is introduced. As specified in Algorithm 1a, H is also a positive definite matrix. The stepsize should satisfy $\eta > \frac{m+1}{2}$. Then, the resulted parallel splitting method for (1.1) generates the new iterate $w^{k+1} = (x_1^{k+1}, x_2^{k+1}, \ldots, x_m^{k+1}, \lambda^{k+1})$ as follows.

Algorithm 2a: The $(k+1)$th iteration of the new parallel splitting method

Step 1.1 Solve the following m subproblems (in parallel):

$$\tilde{x}_i^k := \operatorname{argmin}\{\theta_i(x_i) - (\lambda^k)^\top A_i x_i + \frac{1}{2}\|A_i(x_i - x_i^k)\|_H^2 \mid x_i \in \mathcal{X}_i\}, \;\; i = 1, 2, \ldots, m, \tag{3.15}$$

Step 1.2 Update multiplier

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{\eta} H(\sum_{i=1}^m A_i \tilde{x}_i^k - b). \tag{3.16}$$

Step 2. Generate new iterate

$$w^{k+1} = w^k - \alpha \cdot N^{-T} G_e (w^k - \tilde{w}^k), \tag{3.17}$$

where $\alpha \in (0, 1]$ and the matrix $N^{-T}G_e$ is defined in (2.6).

Remark 3.6. If we set stepsize $\alpha = 1$ in correction step (3.17), then

$$w^{k+1} = w^k - N^{-T} G_e (w^k - \tilde{w}^k). \tag{3.18}$$

Recall the definition of matrix $N^{-T}G_e$ (see (2.6)), thus $x_i^{k+1} = \tilde{x}_i^k$ $(i = 1, \ldots, m)$, and

$$\lambda^{k+1} = \lambda^k - \big[\frac{1}{\eta}(\lambda^k - \tilde{\lambda}^k) - \frac{1}{\eta} H \textstyle\sum_{i=1}^m A_i (x_i^k - x_i^{k+1})\big].$$

Substitute (3.16) into the above equality, we obtain the following formula to update multiplier λ^{k+1} without computing $\tilde{\lambda}^k$:

$$\lambda^{k+1} = \lambda^k - \frac{1}{\eta^2} H(\sum_{i=1}^m A_i x_i^{k+1} - b) + \frac{1}{\eta} H \sum_{i=1}^m A_i (x_i^k - x_i^{k+1}).$$

Remark 3.7. The subproblems of $\tilde{x}_i^k$'s are also solved in a parallel style.

Remark 3.8. In contrast to Algorithm 1a, the proximal parameters r_i $(i = 1, \ldots, m)$ are replaced with a unique parameter η in Algorithm 2a, which is assigned to a more relaxed range.

Analogous to Algorithm 1b, Algorithm 2b is developed in the context of dealing with more general linear operator A_i in contrast to Algorithm 2a. The involved proximal parameters μ_i's in Algorithm 2b are assumed to satisfy the following conditions:

$$\mu_i \geq \rho(A_i^\top H A_i), \; i = 1, \ldots, m.$$

Then, the resulted Algorithm 2b for (1.1) generates the new iterate $w^{k+1} = (x_1^{k+1}, x_2^{k+1}, \dots, x_m^{k+1}, \lambda^{k+1})$ as follows.

Algorithm 2b: The $(k+1)$th iteration of the new parallel splitting method
Step 1.1 Solve the following m subproblems (in parallel):

$$\tilde{x}_i^k := \arg\min\{\theta_i(x_i) - (\lambda^k)^\top A_i x_i + \frac{\mu_i}{2}\|x_i - x_i^k\|^2 \mid x_i \in \mathcal{X}_i\}, \quad i = 1, 2, \dots, m,$$

Step 1.2 Update multiplier

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{\eta} H(\sum_{i=1}^m A_i \tilde{x}_i^k - b).$$

Step 2. Generate new iterate

$$w^{k+1} = w^k - \alpha \cdot N^{-T} G_e (w^k - \tilde{w}^k),$$

where $\alpha \in (0, 1]$ and the matrix $N^{-T} G_e$ is defined in (2.6).

The convergence analysis of Algorithm 2b can also be established by following the way adopted by Algorithm 2a. For succinctness, we omit the routine analysis for Algorithm 2b in the following. In the following analysis, the sequence $\{\tilde{w}^k\}$ generated by the proposed Algorithm 2a will be involved.

Lemma 3.9. *Let the sequences $\{w^k\}$ and $\{\tilde{w}^k\}$ be generated by Algorithm 2a. Then, we have*

$$\tilde{w}^k \in \mathcal{W}, \quad \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^\top F(\tilde{w}^k) + (w - \tilde{w}^k)^\top N(\tilde{w}^k - w^k) \geq 0, \quad \forall\, w \in \mathcal{W}, \tag{3.19}$$

where N is defined in (2.6).

Proof. First, it follows from (3.16) that

$$(\lambda - \tilde{\lambda}^k)^\top \{(A_1 \tilde{x}_1^k + \ldots + A_m \tilde{x}_m^k - b) + H^{-1}(\tilde{\lambda}^k - \lambda^k)\} \geq 0.$$

On the other hand, according to the optimality condition of the x_i-subproblem of (3.15)

$$\theta(x_i) - \theta(\tilde{x}_i^k) + (x_i - \tilde{x}_i^k)^\top (-A_i^\top \tilde{\lambda}^k) + (x_i - \tilde{x}_i^k)^\top A_i^\top [H A_i (\tilde{x}_i^k - x_i^k) + (\tilde{\lambda}^k - \lambda^k)] \geq 0.$$

Consequently, it follows from the above two inequalities:

$$\begin{cases} \theta(x_1) - \theta(\tilde{x}_1^k) + (x_1 - \tilde{x}_1^k)^\top (-A_1^\top \tilde{\lambda}^k) + (x_1 - \tilde{x}_1^k)^\top A_1^\top [H A_1 (\tilde{x}_1^k - x_1^k) + (\tilde{\lambda}^k - \lambda^k)] \geq 0, \\ \theta(x_2) - \theta(\tilde{x}_2^k) + (x_2 - \tilde{x}_2^k)^\top (-A_2^\top \tilde{\lambda}^k) + (x_2 - \tilde{x}_2^k)^\top A_2^\top [H A_2 (\tilde{x}_2^k - x_2^k) + (\tilde{\lambda}^k - \lambda^k)] \geq 0, \\ \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ \theta(x_m) - \theta(\tilde{x}_m^k) + (x_m - \tilde{x}_m^k)^\top (-A_m^\top \tilde{\lambda}^k) + (x_m - \tilde{x}_m^k)^\top A_m^\top [H A_m (\tilde{x}_m^k - x_m^k) + (\tilde{\lambda}^k - \lambda^k)] \\ \geq 0, \\ (\lambda - \tilde{\lambda}^k)^\top \big\{A_1 \tilde{x}_1^k + A_2 \tilde{x}_2^k + \cdots + A_m \tilde{x}_m^k - b - \eta H^{-1}(\lambda^k - \tilde{\lambda}^k)\big\} \geq 0, \end{cases}$$
$$\forall w \in \mathcal{W}.$$

Adding all these inequalities together and using the definitions of F in (2.1b), and N in (2.6), the assertion (3.19) follows immediately. □

Lemma 3.10. *Let the sequences $\{w^k\}$ and $\{\tilde{w}^k\}$ be generated by Algorithm 2a. Then, we have*

$$\|w^k - \tilde{w}^k\|_P^2 - \|w^{k+1} - \tilde{w}^k\|_P^2 = \alpha(1-\alpha)\|w^k - \tilde{w}^k\|_{G_e}^2 + \alpha\|w^k - \tilde{w}^k\|_{S_2}, \tag{3.20}$$

where

$$P = NG_e^{-1}N^\top \tag{3.21}$$

is a symmetric and positive definite matrix.

Proof. First, from (3.18) and (3.21), we have

$$w^{k+1} = w^k - \alpha P^{-1}N(w^k - \tilde{w}^k), \quad \alpha \in (0,1]. \tag{3.22}$$

Consequently,

$$\begin{aligned}
&\|w^k - w\|_P^2 - \|w^{k+1} - w\|_P^2 \\
&\quad= \|w^k - w\|_P^2 - \|w^k - \alpha P^{-1}N(w^k - \tilde{w}^k) - w\|_P^2 \\
&\quad= 2\alpha(w^k - w)^\top N(w^k - \tilde{w}^k) - \alpha^2\|P^{-1}N(w^k - \tilde{w}^k)\|_P^2.
\end{aligned} \tag{3.23}$$

Then, taking $w = \tilde{w}^k$ in (3.23), we have

$$\begin{aligned}
&\|w^k - \tilde{w}^k\|_P^2 - \|w^{k+1} - \tilde{w}^k\|_P^2 \\
&\quad= \alpha(w^k - \tilde{w}^k)^\top(N + N^\top)(w^k - \tilde{w}^k) - \alpha^2\|w^k - \tilde{w}^k\|_{N^\top P^{-1}N}^2 \\
&\quad= \alpha(w^k - \tilde{w}^k)^\top(N + N^\top)(w^k - \tilde{w}^k) - \alpha^2\|w^k - \tilde{w}^k\|_{N^\top P^{-1}N}^2 \\
&\quad= \alpha(w^k - \tilde{w})^\top(G_e + S_2)(w^k - \tilde{w}^k) - \alpha^2\|w^k - \tilde{w}^k\|_{G_e}^2 \\
&\quad= \alpha(1-\alpha)\|w^k - \tilde{w}^k\|_{G_e}^2 + \alpha\|w^k - \tilde{w}^k\|_{S_2}
\end{aligned}$$

The third equality follows from the definition of matrix S_2 (see (2.10)) and the fact $G_e = N^\top P^{-1}N$ (see (3.21)). □

Theorem 3.11. *Let the sequences $\{w^k\}$ and $\{\tilde{w}^k\}$ be generated by Algorithm 2a. Then, we have*

$$\begin{aligned}
&\alpha\Big(\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^\top F(\tilde{w}^k)\Big) + \frac{1}{2}(\|w - w^k\|_P^2 - \|w - w^{k+1}\|_P^2) \\
&\qquad\qquad \geq \frac{1}{2}\{\alpha(1-\alpha)\|w^k - \tilde{w}^k\|_{G_e}^2 + \alpha\|w^k - \tilde{w}^k\|_{S_2}\}, \ \forall\, w \in \mathcal{W},
\end{aligned} \tag{3.24}$$

where the stepsize $\alpha > 0$.

Proof. First, by using the relationship in (3.22), we get

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

Substituting it into (3.19), we obtain

$$\tilde{w}^k \in \mathcal{W},\ \alpha\Big(\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^\top F(\tilde{w}^k)\Big) + (w - \tilde{w}^k)^\top P(w^{k+1} - w^k) \geq 0, \ \forall\, w \in \mathcal{W}. \tag{3.25}$$

In view of Lemma 2.4, we have

$$(w-\tilde{w}^k)^\top P(w^{k+1}-w^k) = \frac{1}{2}\big(\|w-w^k\|_P^2 - \|w-w^{k+1}\|_P^2\big) + \frac{1}{2}\big(\|\tilde{w}^k-w^{k+1}\|_P^2 - \|\tilde{w}^k-w^k\|_P^2\big).$$

Substituting the above identity into (3.25), we have

$$\begin{aligned}\tilde{w}^k\in\mathcal{W},\ \alpha\big(\theta(u)-\theta(\tilde{u}^k)+(w-\tilde{w}^k)^\top F(\tilde{w}^k)\big)+\frac{1}{2}\big(\|w-w^k\|_P^2-\|w-w^{k+1}\|_P^2\big)\\ \ge \frac{1}{2}\big(\|\tilde{w}^k-w^k\|_P^2-\|\tilde{w}^k-w^{k+1}\|_P^2\big),\\ = \frac{1}{2}\{\alpha(1-\alpha)\|w^k-\tilde{w}^k\|_{G_e}^2+\alpha\|w^k-\tilde{w}^k\|_{S_2}\}.\end{aligned}$$

The last equality follows from (3.20). Hence, the assertion (3.24) follows immediately. □

$\boxed{4}$ Convergence Analysis

The following theorem implies the fact that the sequence $\{w^k\}$ generated by Algorithm 1a is Fejèr monotone with respect to the solution set of (1.1). Hence, the convergence of Algorithm 1a can be easily derived.

Theorem 4.1. *Let $\{w^k\}$ be the sequence generated by Algorithm 1a and $\{\hat{w}^k\}$ be given in (3.8). Then, for any $w^*\in\mathcal{W}^*$, we have*

$$\|w^{k+1}-w^*\|_{G_r}^2 \le \|w^k-w^*\|_{G_r}^2 - \|w^k-\hat{w}^k\|_{S_1}^2,\quad \forall\, w^*\in\mathcal{W}^*, \tag{4.1}$$

where G_r and S_1 are defined in (2.3) and (2.9), respectively.

Proof. By setting $w=w^*$ in (3.13), we obtain that

$$\|w^{k+1}-w^*\|_{G_r}^2 \le \|w^k-w^*\|_{G_r}^2 - \|w^k-\hat{w}^k\|_{s_1}^2 - \{\theta(\hat{u}^k)-\theta(u^*)+(\hat{w}^k-w^*)^\top F(\hat{w}^k)\}.$$

Setting $u=\hat{u}^k$ in (2.1a), we get

$$0\le\theta(\hat{u}^k)-\theta(u^*)+(\hat{w}^k-w^*)^\top F(\hat{w}^k).$$

Adding the above two inequalities, the assertion (4.1) follows immediately. □

Now, we are ready to derive the convergence of the proposed Algorithm 1a in the following theorem.

Theorem 4.2. *Assume that the parameters r_i's satisfy (2.8). Then the sequence $\{w^k\}$ generated by Algorithm 1a converges to a solution point of VI($\mathcal{W},F,\theta$).*

Proof. First, it follows from inequality (4.1) that the sequence $\{w^k\}$ is bounded. Moreover,

$$\sum_{k=0}^{\infty}\|w^k-\hat{w}^k\|_{S_1}^2 \le \|w^0-w^*\|_{G_r}^2,\quad \forall\, w^*\in\mathcal{W}^*.$$

In view of the conclusion 2) of Lemma 2.7, the positive definiteness of S_1 is ensured by the assumption $\sum_{i=1}^m\frac{1}{r_i}<1$. Thus, the above inequality implies that

$$\lim_{k\to\infty}\|x_i^k-\hat{x}_i^k\|=0,\ \ i=1,\dots,m,\quad\text{and}\quad \lim_{k\to\infty}\|\lambda^k-\hat{\lambda}^k\|=0.$$

Hence,

$$\lim_{k\to\infty} \|w^k - \hat{w}^k\| = 0. \tag{4.2}$$

Hence, the sequence $\{\hat{w}^k\}$ is also bounded, and has at least one cluster point.

Let w^∞ be a cluster point of the sequence $\{\hat{w}^k\}$ and $\{\hat{w}^{k_j}\}$ be a subsequence converging to w^∞. It follows from Theorem 3.3 that

$$\hat{w}^k \in \mathcal{W}, \quad \theta(u) - \theta(\hat{u}^k) + (w - \hat{w}^k)^\top F(\hat{w}^k) + (w - \hat{w}^k)^\top G_r Q(\hat{w}^k - w^k) \geq 0, \quad \forall\, w \in \mathcal{W}.$$

Consequently,

$$\hat{w}^k \in \mathcal{W}, \quad \theta(u) - \theta(\hat{u}^{k_j}) + (w - \hat{w}^{k_j})^\top F(\hat{w}^{k_j}) + (w - \hat{w}^{k_j})^\top G_r Q(\hat{w}^{k_j} - w^{k_j}) \geq 0, \quad \forall\, w \in \mathcal{W}.$$

Taking the limit over j in the above inequality, and considering the continuousness of a convex function in its domain and (4.2), we have

$$\theta(u) - \theta(u^\infty) + (w - w^\infty)^\top F(w^\infty) \geq 0, \forall\, w \in \mathcal{W}.$$

According to Theorem 2.2, w^∞ is a solution of VI$(\mathcal{W}, F, \theta)$. Finally, from (4.2),

$$w^{k_j} \to w^\infty.$$

Hence, (4.1) implies the sequence $\{w^k\}$ has the only cluster point $\{w^\infty\}$. □

Analogous to Theorem 4.1, the following theorem implies the fact that the sequence $\{w^k\}$ generated by Algorithm 2a is Fejèr monotone with respect to the solution set of (1.1).

Theorem 4.3. *Let the sequences $\{w^k\}$ and $\{\tilde{w}^k\}$ be generated by Algorithm 2a. Then, for any $w^* \in \mathcal{W}^*$, we have*

$$\|w^{k+1} - w^*\|_P^2 \leq \|w^k - w^*\|_P^2 - \alpha(1-\alpha)\|w^k - \tilde{w}^k\|_{G_e}^2 - \alpha\|w^k - \tilde{w}^k\|_{S_2}^2, \quad \forall\, w^* \in \mathcal{W}^*. \tag{4.3}$$

Proof. By setting $w = w^*$ in (3.24), we obtain that

$$\begin{aligned} \|w^{k+1} - w^*\|_P^2 \leq \|w^k - w^*\|_P^2 - \{\alpha(1-\alpha)\|w^k - \tilde{w}^k\|_{G_e}^2 + \alpha\|w^k - \tilde{w}^k\|_{S_2}\} \\ -\alpha(\theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^\top F(\tilde{w}^k)), \ \forall\, w \in \mathcal{W}. \end{aligned}$$

Note that

$$0 \leq \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^\top F(\tilde{w}^k).$$

Adding the above two inequalities, the assertion (4.3) follows immediately. □

Finally, we are in the stage to derive the convergence of the proposed Algorithm 2a in the following theorem.

Theorem 4.4. *Assume that $\eta > \frac{m+1}{2}$. Then, the sequence $\{w^k\}$ generated by Algorithm 2a converges to a solution point of VI$(\mathcal{W}, F, \theta)$.*

Proof. The proof is similar to Theorem 4.2, and thus is omitted. □

$\boxed{5}$ Convergence Rate

In this section, our purpose is to show that after t iterations of the proposed algorithms, we can find a $\tilde{w} \in \mathcal{W}$ such that (2.3) is satisfied with $\epsilon \sim O(1/t)$. Thus the $O(1/t)$ convergence rate of the methods is established. In the following, we delineate the convergence rate for Algorithms 1a and 2a individually in two subsections.

$\boxed{5.1}$ The $O(\frac{1}{t})$ Convergence Rate for Algorithm 1a

Corollary 5.1. *Let the sequence $\{w^k\}$ be generated by Algorithm 1a and $\{\hat{w}^k\}$ be defined in (3.8). Then, we have*

$$\theta(u) - \theta(\hat{u}^k) + (w - \hat{w}^k)^\top F(w) + \frac{1}{2}(\|w - w^k\|^2_{G_r} - \|w - w^{k+1}\|^2_{G_r}) \geq 0, \ \forall\, w \in \mathcal{W}. \tag{5.1}$$

Proof. From inequality (3.13), we get

$$\theta(u) - \theta(\hat{u}^k) + (w - \hat{w}^k)^\top F(\hat{w}^k) + \frac{1}{2}(\|w - w^k\|^2_{G_r} - \|w - w^{k+1}\|^2_{G_r}) \geq 0.$$

Hence, by using the monotonicity of operator F, we obtain

$$\theta(u) - \theta(\hat{u}^k) + (w - \hat{w}^k)^\top F(w) + \frac{1}{2}(\|w - w^k\|^2_{G_r} - \|w - w^{k+1}\|^2_{G_r}) \geq 0, \quad \forall\, w \in \mathcal{W}.$$

The conclusion is verified. □

Theorem 5.2. *Let $\{w^k\}$ be the sequence generated by Algorithm 1a and $\{\hat{w}^k\}$ be given in (3.8). For any integer $t > 0$, let*

$$\hat{w}_t := \frac{1}{t+1}\sum_{k=0}^{T} \hat{w}^k. \tag{5.2}$$

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

$$\big(\theta(\hat{u}_t) - \theta(u)\big) + (\hat{w}_t - w)^\top F(w) \leq \frac{1}{2(t+1)}\|w - w^0\|^2_{G_r} \quad \forall\, w \in \mathcal{W}.$$

For any given compact set $\mathcal{D} \subset \mathcal{W}$, let $d_1 := \sup\{\|w - w^0\|^2_{G_r} \mid w \in \mathcal{D}\}$. Then, after t iterations of Algorithm 1a, we find a certain point $\hat{w}_t$ that satisfies

$$\sup_{w\in\mathcal{D}}\{\theta(\hat{u}_t) - \theta(u) + (\hat{w}_t - w)^\top F(w)\} \leq \frac{d_1}{2(t+1)},$$

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

Proof. First, because $\hat{x}_i^k = x_i^{k+1}$ $(i = 1, \dots, m)$, it holds that $\hat{w}^k \in \mathcal{W}$ for all $k \geq 0$. Thus, together with convexity of $\mathcal{X}_i$ $(i = 1, \dots, m)$, the definition in (5.2) implies that $\hat{w}_t \in \mathcal{W}$. Second, summing the inequalities (5.1) over $k = 0, 1, \dots, t$, we obtain

$$(t+1)\theta(u) - \sum_{k=0}^{T}\theta(\hat{u}^k) + \big((t+1)w - \sum_{k=0}^{T}\hat{w}^k)^\top F(w) + \frac{1}{2}\|w - w^0\|^2_{G_r} \geq 0, \quad \forall\, w \in \mathcal{W}.$$

Combining the notation of $\hat{w}_t$, it can be written as

$$\frac{1}{t+1}\sum_{k=0}^{T}\theta(\hat{u}^k)-\theta(u)+(\hat{w}_t-w)^\top F(w)\le\frac{1}{2(t+1)}\|w-w^0\|_{G_r}^2,\quad\forall\, w\in\mathcal{W}. \tag{5.3}$$

Since $\theta(u)$ is convex and

$$\hat{u}_t=\frac{1}{t+1}\sum_{k=0}^{T}\hat{u}^k,$$

we have $\theta(\hat{u}_t)\le\frac{1}{t+1}\sum_{k=0}^{T}\theta(\hat{u}^k)$. Substituting it in inequality (5.3), the assertion of this theorem follows directly. □

Theorem 5.2 implies that for any given $\epsilon>0$, after at most

$$t=\lceil\frac{d_1}{2\epsilon}-1\rceil$$

iterations, we have

$$\theta(\hat{u}_t)+(\hat{w}_t-w)^\top F(w)\le\epsilon,\forall w\in\mathcal{D}.$$

Thus, the $O(\frac{1}{t})$ convergence rate of Algorithm 1a is established in an ergodic sense.

5.2 The $O(\frac{1}{t})$ Convergence Rate for Algorithm 2a

Now we start to prove some properties of the sequence $\{\tilde{w}^k\}$ generated by Algorithm 2a.

Corollary 5.3. *Let the sequences $\{w^k\}$ and $\{\tilde{w}^k\}$ be generated by Algorithm 2a. Then, we have*

$$\alpha(\theta(u)-\theta(\tilde{u}^k)+(w-\tilde{w}^k)^\top F(w))+\frac{1}{2}\big(\|w-w^k\|_P^2-\|w-w^{k+1}\|_P^2\big)\ge 0,\ \forall\, w\in\mathcal{W},$$

where $\alpha\in(0,1]$.

Proof. The proof is similar to Corollary 5.1, thus is omitted. □

Theorem 5.4. *Let the sequences $\{w^k\}$ and $\{\tilde{w}^k\}$ be generated by Algorithm 2a. For any integer $t>0$, let*

$$\tilde{w}_t:=\frac{1}{t+1}\sum_{k=0}^{T}\tilde{w}^k.$$

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

$$\big(\theta(\tilde{u}_t)-\theta(u)\big)+(\tilde{w}_t-w)^\top F(w)\le\frac{1}{2\alpha(t+1)}\|w-w^0\|_P^2\quad\forall w\ \in\mathcal{W}.$$

For any given compact set $\mathcal{D}\subset\mathcal{W}$, let $d_2:=\sup\{\|w-w^0\|_P^2\mid w\in\mathcal{D}\}$. Then, after t iterations of Algorithm 2a, we find a certain point $\tilde{w}_t$ that satisfies

$$\sup_{w\in\mathcal{D}}\{\theta(\tilde{u}_t)-\theta(u)+(\tilde{w}_t-w)^\top F(w)\}\le\frac{d_2}{2\alpha(t+1)},$$

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

Proof. The proof is analogous to Theorem 5.2, thus is omitted. □

In a similar way, we can establish the $O(\frac{1}{t})$ convergence rate of Algorithm 2a in an ergodic sense.

6 Numerical Implementation

In this section, we apply the proposed algorithms to solve some problem arising in matrix decomposition and compare them with some existing splitting methods. Through numerical comparison, we show the efficiency of the proposed algorithms. All the codes were written in MATLAB 7.12 (R2011a) and were run on a ThinkPad notebook with the Intel Core i5-2140M CPU at 2.3 GHz and 4 GB of memory.

We focus on the specific model of recovering low-rank and sparse components of matrices from incomplete and noisy observations, which was recently proposed in [18] based on the pioneering work [2, 3]. By comparing the proposed algorithms with the methods in [5] (denoted by HYZ), and VASALM [18], the advantages of the proposed methods will be evident.

6.1 Synthetic Simulations

We consider the following constrained model for the matrix decomposition problem [18]:

$$\begin{array}{rl} \min_{A,E} & \|A\|_* + \tau\|E\|_1 \\ \text{s.t.} & \|P_\Omega(C - A - E)\|_F \le \delta, \end{array} \tag{6.1}$$

where $C \in \mathbb{R}^{m\times n}$ is the given matrix (data) and $\|\cdot\|_*$ is the nuclear norm (i.e., sum of the singular values) while the $\|\cdot\|_1$ represents the sum of absolute values of all entries. Ω is a subset of the index set of entries $\{1, 2, \dots, m\} \times \{1, 2, \dots, n\}$ representing the observable entries; $P_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ denotes the incomplete observation information and it is summarized by the orthogonal projection onto the span of matrices vanishing outside of Ω so that the ijth entry of $P_\Omega(X)$ is X_{ij} if $(i, j) \in \Omega$ and zero otherwise; δ is related to the Gaussian noise level and $\|\cdot\|_F$ is the standard Frobenius norm. By defining $M = P_\Omega(C)$, the inequality constraint in (6.1) can be rewritten as

$$\|P_\Omega(M - A - E)\|_F \le \delta.$$

Let $\mathbf{B} := \{Z \in \mathbb{R}^{m\times n} | \|P_\Omega(Z)\|_F \le \delta\}$ and introduce an auxiliary variable Z. Then, it is easy to see that (6.1) can be reformulated as

$$\begin{array}{rl} \min_{A,E,Z} & \|A\|_* + \tau\|E\|_1 \\ s.t. & A + E + Z = M, \\ & Z \in \mathbf{B}, \end{array} \tag{6.2}$$

see [18] for more details. Note (6.2) is a concrete application of (1.1) with $m = 3$, except that the vector variables and coefficients in (1.1) are replaced by matrix variables and linear operators in matrix spaces, respectively. As we have mentioned, the proposed methods and theoretical analysis can be trivially extended to this extended model. More specifically, (6.2) can be explained as a special case of (1.1) with the specification $\theta_1(A) = \|A\|_*$, $\theta_2(E) = \tau\|E\|_1$ and $\theta_3(Z) = \iota_{\mathbf{B}}(Z)$ where $\iota_{\mathbf{B}}(\cdot)$ represents the indicator function defined on the closed convex set $\mathbf{B}$; A_1, A_2 and A_3 in (1.1) are all the identity mapping and $b := M$. In all these methods, we set the penalty matrix $H = \beta I$ for a fair comparison, where I denotes identity matrix.

For solving model (6.2), Algorithm 1a is equivalent to Algorithm 1b by setting $\delta_i = \beta r_i$ $(i = 1, \ldots, m)$. Moreover, Algorithm 2a is identical to Algorithm 2b with the choice of $\mu_i = \beta I$ $(i = 1, \ldots, m)$ in Algorithm 2b. Therefore, in the coming comparison, we unify Algorithm 1a (1b) as Algorithm 1, and so does Algorithm 2. As shown in [18], the subproblems in each proposed algorithm are all simple enough to have closed-form solutions.

For completeness, in the following we illustrate each subproblem of Algorithm 1 in detail. Note that for the resulting $(A, E, Z)-$ subproblems are implemented in a parallel manner. By setting $r_i \equiv r$ $(i = 1, 2, 3)$, each subproblem of Algorithm 1 is given by the following:

- Update the multiplier $\hat{\Lambda}^k = \Lambda^k - \beta(A^k + E^k + Z^k - M)$.
- Solve the Z-subproblem and obtain its solution Z^{k+1} via

  $$Z_{ij}^{k+1} = \begin{cases} N_{ij}^k, & \text{if } (i,j) \notin \Omega, \\ \frac{\min\{\|P_\Omega(N^k)\|_F, \delta\}}{\|P_\Omega(N^k)\|_F} N_{ij}^k & \text{if } (i,j) \in \Omega, \end{cases}$$

  where $N^k = \frac{1}{\beta r}\hat{\Lambda}^k + Z^k$.
- Solve the E-subproblem and obtain its solution E^{k+1} via

  $$E^{k+1} = \mathcal{S}_{\frac{\tau}{\beta r}}(E^k + \frac{\hat{\Lambda}^k}{\beta r}),$$

  where the operator $\mathcal{S}_c : \mathcal{R}^{m\times n} \to \mathcal{R}^{m\times n}$ is the shrinkage operator defined by

  $$(\mathcal{S}_c(T))_{ij} := \text{sign}(T_{ij}) \cdot \max\{|T_{ij}| - c, 0\}, \quad 1 \le i \le m, \ 1 \le j \le n,$$

  with sign$(\cdot)$ being the sign function, $c > 0$ and the matrix $T \in \mathcal{R}^{m\times n}$.
- Solve the A-subproblem and obtain its solution A^{k+1} via

  $$A^{k+1} = \mathcal{D}_{\frac{1}{\beta r}}(\hat{\Lambda}^k/\beta r + A^k).$$

  Here, for $c > 0$, the operator $\mathcal{D} : \mathcal{R}^{m\times n} \to \mathcal{R}^{m\times n}$ is defined by

  $$\mathcal{D}_c(T) := U\text{diag}\left(\mathcal{S}_c(\Sigma)\right) V^\top,$$

  where $U\Sigma V^\top$ is the singular value decomposition (SVD) of matrix T.

As in [18], we let $C = A^* + E^* + Z^*$ be the data matrix, where A^*, E^* and Z^* are the low-rank, the sparse components and the Gaussian noise, respectively. We generate A^* by $A^* = LR^\top$, where L and R are independent $m \times r$, and $n \times r$ matrices respectively whose elements are i.i.d. Gaussian random variables with zero mean and unit variance. Hence, the rank of A^* is r. The index of observed entries, i.e. Ω, is determined at random. The support $\Gamma \subset \Omega$ of the impulsive noise E^* (sparse but large) is chosen uniformly at random, and the non-zero entries of E^* are i.i.d. uniformly in the interval $[-500, 500]$. Let `sr`, `spr` and `rr` represent the ratios of sample (observed) entries (i.e., $|\Omega|/(m \cdot n)$, where the symbol $|\Omega|$ denotes the cardinality of Ω), the number of non-zero entries of E (i.e., $\|E\|_0/(m \cdot n)$), and the rank of A^* (i.e., r/m), respectively.

In our experiments, we choose $m = n = 500, 1000$, $\texttt{sr} = 0.8$, and set $\tau = 1/\sqrt{m}$ in (6.1). Our numerical experiments focus on the special case $\sigma = 0$ (σ denotes Gaussian noise level). The model parameter δ is therefore chosen as 0. As in [18], we stop HYZ by the stopping criterion

$$\text{RelChg} := \frac{\|(A^{k+1}, E^{k+1}, Z^{k+1}) - (A^k, E^k, Z^k)\|_F}{\|(A^k, E^k, Z^k)\|_F + 1} \le Tol, \tag{6.3}$$

where $Tol = 1e-4$. Then, we run other methods until they achieve a more accurate solution than HYZ in terms of the relative error of the low rank and the sparse components. For other individual parameters required by these methods, we choose

$$\beta = \begin{cases} 0.08\frac{|\Omega|}{\|P_\Omega(C)\|_1}, & \text{if } \texttt{spr} = 0.05; \\ 0.15\frac{|\Omega|}{\|P_\Omega(C)\|_1}, & \text{if } \texttt{spr} = 0.1, \end{cases}$$

for both Algorithms 1 and 2, $r_i \equiv r = 3$ in Algorithm 1, and set $\alpha = 1$ and $\eta = 2.01$ in Algorithm 2. In all the tested scenarios, the initial iterate is $(A^0, E^0, Z^0) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$. We use the recommended setting for VASALM and set $\gamma = 1.7$ in HYZ for achieving the best performance. In our experiments, we executed the singular value decomposition (SVD) by using PROPACK [15] to compute those singular values that are larger than a particular threshold and their corresponding singular vectors in A-involved subproblem. We denote by $(\hat{A}, \hat{E})$ the iterate when the stopping criterion is satisfied.

Table 1 shows that the numerical results of Algorithms 1 and 2, HYZ and VASALM. More specifically, we report the relative error of the recovered sparse component ($ErrsSP := \frac{\|\hat{E}-E^*\|_F}{\|E^*\|_F}$), the relative error of the recovered low-rank component ($ErrsLR := \frac{\|\hat{A}-A^*\|_F}{\|A^*\|_F}$), the computing time in seconds ("Time (s)") and the number of singular value decompositions required by the A-related subproblems ("#SVD"). The computing time is recorded by considering the possibility of parallel implementation, and so we include only the time of the most time-consuming subproblem at each iteration for Algorithm 1 and 2.

According to the data in Table 1, as we expect, all these methods can exactly recover the low rank matrix and sparse components from corruption and missing observations. Algorithm 2 behaves almost in the same manner as Algorithm 1 in terms of the computation cost and the solution accuracy. Compared with HYZ, the proposed methods are more attractive. Both of them achieve a more accurate solution than HYZ while keeping computational time low. Note that HYZ requires a correction step to ensure convergence. These correction steps may ruin the low-rank characteristic. Hence, HYZ ends up with high-rank iterates after implementing some correction steps. The inferior performance of HYZ mainly comes from two terms: First, the correction step can only be realized after prediction step, hence the method cannot be implemented in a parallel style completely. On the other hand, to compute stepsize is always time consuming. However, VASALM cannot be implemented in a parallel way while the new algorithms could.

To see the comparison clearly, we focus on the particular case where $m = n = 500$, $\texttt{spr} = 0.05$, $\texttt{rr} = 0.05$, $\texttt{sr} = 0.8$ and $\sigma = 0$; and visualize the iterative processes of different methods in Figure 1. More specifically, we plot the evolutions of the rank of the recovered low rank part, and the relative error $ErrsSP$ and $ErrsLR$ with respect to CPU time. Figure 1 shows that the rank of iterates generated by HYZ changes radically according to iterations at the first stage; while the rank of iterates generated by any of Algorithms 1, 2 and VASALM is much more stable, i.e., not sensitive to the iterations. Therefore, the low-rank feature is well preserved by Algorithm 1, 2 and VASALM; and this advantage is very suitable for the application of some popular packages for partial SVD such as PROPACK.

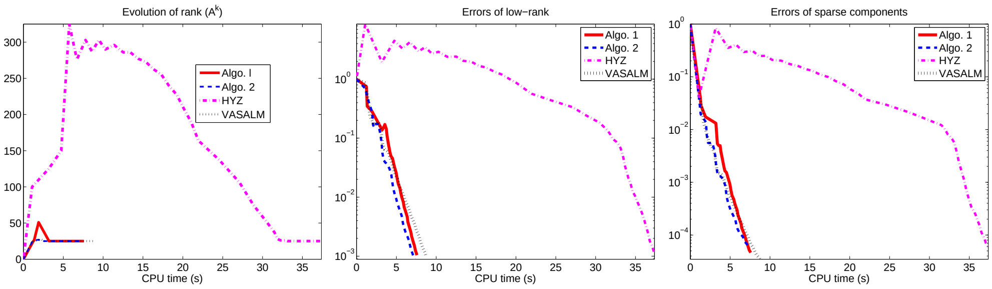

Figure 1: Evolution of rank (left), the relative error of the low-rank (middle) and the sparse components (right) for Algorithm 1, 2, HYZ and VASALM.

Table 1: Recovery results of VASALM, Algorithms 1 and 2, HYZ for (6.1) with sr=80% and $\sigma = 0$

n	rr	spr	$\frac{\|\hat{E}-E^*\|_F}{\|E^*\|_F}(\times 10^{-5})$				$\frac{\|\hat{A}-A^*\|_F}{\|A^*\|_F}(\times 10^{-3})$				#SVD				Time (s)			
			VA	Alg1	Alg2	HYZ	VA	Alg1	Alg2	HYZ	VA	Alg1	Alg2	HYZ	VA	Alg1	Alg2	HYZ
500	0.05	0.05	3.48	4.61	6.27	6.30	0.991	1.05	0.888	1.09	39	35	33	50	9.8	8.2	7.6	39.4
		0.1	6.81	5.58	12.4	14.5	1.47	1.60	1.46	1.64	36	36	33	54	8.2	12.8	8.0	39.9
	0.1	0.05	14.5	13.0	18.9	19.0	2.56	2.69	2.62	2.82	33	37	42	74	13.2	27.2	11.6	44.2
		0.1	5.02	6.15	9.20	9.65	0.941	0.966	0.927	1.04	44	40	37	57	16.9	12.9	10.2	40.5
1000	0.05	0.05	3.69	5.22	6.46	7.30	1.05	1.03	1.05	1.09	49	36	37	52	100.0	31.2	36.2	355.6
		0.1	4.38	5.86	7.83	11.4	1.26	1.27	1.29	1.32	46	40	37	51	98.2	61.7	37.3	384.3
	0.1	0.05	4.60	6.75	7.96	8.39	0.893	0.909	0.904	0.920	62	46	47	65	219.4	121.3	98.8	382.1
		0.1	5.60	7.79	10.2	l4.1	1.09	1.16	1.14	1.17	59	48	46	63	258.2	212.5	128.7	406.2

Alg1, Alg2 represent Algorithm 1 and 2, respectively.
VA stands for VASALM.

On the other hand, the errors generated by HYZ's iterations change more radically than Algorithms 1 and 2. In conclusion, Algorithms 1 and 2 converge faster than HYZ, and even competitive with VASALM in terms of computing time.

6.2 Background Extraction on a Noisy Video with Missing Data

We investigate the application of (6.1): extracting background from surveillance video with missing and noisy data. To understand this concrete application, we first introduce some preliminary background of this application and refer the readers to, e.g. [2], for more details. More specifically, video consists of a sequence of frames, and mathematically it is a natural candidate for low-rank modeling due to high correlation between frames. Each frame consists of foreground and background. Since the background of video needs to be flexible enough to accommodate changes in the scene, it is natural to model it as approximately low rank. Foreground objects, such as cars or pedestrians, occupy a fraction of the image pixels and hence can be treated as sparse errors. The basic task in video surveillance is to separate the foreground from background. In our experiments, we test the airport video downloadable at the website [†], which is a sequence of 150 grayscale frames of size 144×176 taken in an airport. The data matrix C in model (6.1) is formed by stacking each frame into a column and $C \in \mathbb{R}^{20480\times 150}$. The tested video has 30% missing pixels. The index of observed entries, i.e., Ω, is determined randomly by the MATLAB built-in function `randperm`. The Gaussian noise is generated with a zero mean and its standard deviation is $\sigma = 10^{-3}$. Therefore, we take model parameter $\delta = \sqrt{m + \sqrt{8m}\sigma}$. We use Algorithms 1,

[†] `http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html`

2 and HYZ, VASALM to extract moving objects from these corrupted videos. The setting of each algorithm is the same as Section 6.1 except $\beta = 0.01\frac{|\Omega|}{\|P_\Omega(C)\|_1}$. We stop HYZ using a more relaxed termination criterion, i.e., set $Tol = \sigma$ in (6.3) as implemented in Section 6.1. Then, other algorithms continue to run until they achieve a much better solution than HYZ in the term of the relative residual. The recovery results from Algorithms 1, 2 and HYZ and VASALM are visually similar. Figure 2 shows the results via Algorithm 1 for the 50th, 100th and 125th frames of corrupted video, and recovered background and foreground in the first, second and third columns, respectively. The recovery numerical results of each algorithm are displayed in Table 2, including the number of iterations (It.), relative residual (RelRes.), rank of background (rank($\hat{A}$)) and computation time in seconds (Time(s)), and the number of singular value decompositions (#SVD). As shown in Table 2, the recovered rank of Algorithms 1, 2 and VASALM are the same, and much lower than HYZ. Moreover, the CPU time of Algorithms 1 and 2 are significantly less than those for HYZ, and is very comparable to VASALM. Therefore, our proposed methods may open up a new way to handle video surveillance.

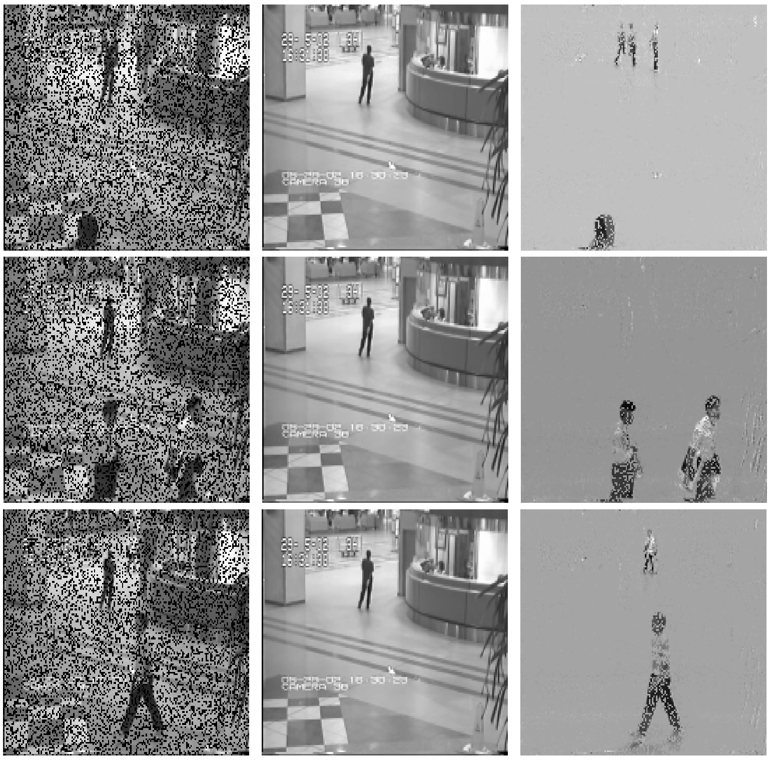

Figure 2: Background extraction from a noisy video with missing data

7 Conclusions

For solving the separable convex programming problem with linking linear constraints and its objective function is formed as the sum of m individual functions without overlapping variables, we present four distinct splitting algorithms whose m decomposed subproblems

Table 2: Recovery results for background extraction

	It.	RelRes.	rank($\hat{A}$)	Time (s)	#SVD
Algo. 1	100	1.86e-3	32	212.2	101
Algo. 2	129	1.88e-3	32	242.7	130
VA	100	1.87e-3	32	266.6	100
HYZ	182	1.89e-3	35	510.4	182

are completely tailored for parallel computation. Moreover, the new methods require no correction step. The efficiency of the new algorithms are illustrated numerically by some concrete applications arising in the area of matrix optimization and video processing.

Acknowledgements

I would like to thank Dr. Anthony Man-Cho So and Dr. Qing Ling for reading this paper carefully and giving many helpful comments.

References

[1] N. Bose and K. Boo, High-resolution image reconstruction with multisensors. *Int. J. Imag. Syst. Tech.* 9 (1998) 294–304.

[2] E.J. Candès, X. Li, Y. Ma and J. Wright, Robust principal component analysis, *J. ACM.* 58 (2011) 1–37.

[3] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo and A.S. Willskyc, Rank-sparsity incoherence for matrix decomposition, *SIAM J. Optim.* 21 (2011) 572–596.

[4] G. Chen and M. Teboulle, A proximal based decomposition method for convex minimization problems, *Math. Program.* 64 (1994) 81–101.

[5] D.R. Han, X.M. Yuan and W.X. Zhang, An augmented-Lagrangian-based parallel splitting method for linearly constrained separate convex programming with applications to image processing, *Math. Comp.*, to appear.

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

[7] B.S. He, M. Tao, M.H. Xu. and X.M. Yuan, An alternating directions based contraction method for generally separable linearly constrained convex programming problems, *Optimization*, to appear.

[8] B.S. He, M. Tao and X.M. Yuan, A splitting method for separate convex programming with linking linear constraints, Submitted (2010).

[9] M. Hestenes, Multiplier and gradient methods, *J. Opti. Theory Appli.* 4 (1969) 303–320.

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

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

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

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

[14] K.C. Kiwiel, C.H. Rosa and A. Ruszczyński, Proximal decomposition via alternating linearization, *SIAM J. Opti.* 9 (1999) 668–689.

[15] R.M. Larsen, Lanczos bidiagonalization with partial reorthogonalization. Department of Computer Science, Aarhus University, Technical report, DAIMI PB-357, code available at http://soi.stanfor.edu/ rmunk/PROPACK/ (1998)

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

[17] M. Powell, A method for nonlinear constraints in minimization problems, in *Reformulation*: *Optimization*, R. Fletcher (eds.), 1969, pp. 283–298.

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

[19] M. Tao and X.M. Yuan, An inexact parallel splitting augmented Lagrangian methods for monotone variational inequalities with separable structures, *Comput. Optim. Appl.* 52 (2012) 439–461.

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

Manuscript received 10 July 2013
revised 10 October 22 2013
accepted for publication 22 October 2013

Min Tao
Department of Mathematics, Nanjing University
Nanjing, Jiangsu, China
E-mail address: taom@nju.edu.cn