

A Generalized Matrix Splitting Algorithm

Ganzhao Yuan, Wei-Shi Zheng, Li Shen, Bernard Ghanem

Abstract—Composite function minimization captures a wide spectrum of applications in both computer vision and machine learning. It includes bound constrained optimization, ℓ_1 norm regularized optimization, and ℓ_0 norm regularized optimization as special cases. This paper proposes and analyzes a new Generalized Matrix Splitting Algorithm (GMSA) for minimizing composite functions. It can be viewed as a generalization of the classical Gauss-Seidel method and the Successive Over-Relaxation method for solving linear systems in the literature. Our algorithm is derived from a novel triangle operator mapping, which can be computed exactly using a new generalized Gaussian elimination procedure. We establish the global convergence, convergence rate, and iteration complexity of GMSA for convex problems. In addition, we also discuss several important extensions of GMSA. Finally, we validate the performance of our proposed method on three particular applications: nonnegative matrix factorization, ℓ_0 norm regularized sparse coding, and ℓ_1 norm regularized Dantzig selector problem. Extensive experiments show that our method achieves state-of-the-art performance in term of both efficiency and efficacy.

Index Terms—Matrix Splitting Algorithm, Nonsmooth Optimization, Convex Optimization, Convergence Analysis.



1 INTRODUCTION

In this paper, we focus on the following composite function minimization problem:

$$\min_{\mathbf{x}} f(\mathbf{x}) \triangleq q(\mathbf{x}) + h(\mathbf{x}); \quad q(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{x}^T \mathbf{b} \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a symmetric positive semidefinite matrix, and $h(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}$ is a piecewise separable function (i.e. $h(\mathbf{x}) = \sum_{i=1}^n h_i(\mathbf{x}_i)$) but not necessarily convex. Typical examples of $h(\mathbf{x})$ include the bound constrained function and the ℓ_0 and ℓ_1 norm functions. We assume that $f(\mathbf{x})$ is bounded below for any feasible solution \mathbf{x} .

The optimization in (1) is flexible enough to model a variety of applications of interest in both computer vision and machine learning, including compressive sensing [9], nonnegative matrix factorization [11], [20], [22], sparse coding [1], [2], [21], [35], support vector machine [15], logistic regression [47], subspace clustering [10], to name a few. Although we only focus on quadratic function $q(\cdot)$, our method can be extended to handle

general non-quadratic composite functions by considering a Newton approximation of the objective [42], [50] and to solve general linear constrained problems by using its associated augmented Lagrangian function of the problem [12], [13].

The most popular method for solving problem (1) is perhaps the proximal gradient method [3], [31]. It considers a fixed-point proximal iterative procedure $\mathbf{x}^{k+1} = \text{prox}_{\gamma h}(\mathbf{x}^k - \gamma \nabla q(\mathbf{x}^k))$ based on the current gradient $\nabla q(\mathbf{x}^k)$. Here the proximal operator $\text{prox}_{\tilde{h}}(\mathbf{a}) = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{a}\|_2^2 + \tilde{h}(\mathbf{x})$ can often be evaluated analytically, and $\gamma = 1/L$ is the step size with L being the local (or global) Lipschitz constant. It is guaranteed that the objective decreases at a rate of $\mathcal{O}(L/k)$, where k is the iteration number. The accelerated proximal gradient method can further boost the rate to $\mathcal{O}(L/k^2)$. Tighter estimates of the local Lipschitz constant leads to better convergence rate, but it scarifies extra computation overhead to compute L . Our method is also a fixed-point iterative method, but it does not rely on a sparse eigenvalue solver or line search backtracking to compute such a Lipschitz constant, and it can exploit the specified structure of the quadratic Hessian matrix \mathbf{A} .

Our proposed method is essentially a generalization of the classical Gauss-Seidel (GS) method and Successive Over-Relaxation (SOR) method [8], [37]. In numerical linear algebra, the Gauss-Seidel method, also known as the successive displacement method, is a fast iterative method for solving a linear system of equations. It works by solving a sequence of triangular matrix equations. The method of SOR is a variant of the GS method and it often leads to faster convergence. Similar iterative methods for solving linear systems include the Jacobi method and symmetric SOR. Our proposed method can solve versatile composite function minimization problems, while inheriting the efficiency of modern linear algebra techniques.

Our method is closely related to coordinate gradient descent and its variants such as randomized coordinate descent [15], [34], cyclic coordinate descent [39], block coordinate descent [4], [14], [30], randomized block coordinate descent [26], [36], accelerated randomized coordinate descent [23], [25], [30] and others [18], [24], [52]. However, all these work are based on gradient-descent type iterations and a constant Lipschitz step size. They work by

- Ganzhao Yuan is with the School of Data and Computer Science, Sun Yat-sen University (SYSU), Guangzhou, Guangdong 510275, China, and also with Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Beijing 221143, China. E-mail: yuan-ganzhao@gmail.com.
- Wei-Shi Zheng is with the School of Data and Computer Science, Sun Yat-sen University (SYSU), Guangzhou, Guangdong 510275, China, and also with Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Beijing 221143, China. E-mail: wszheng@ieee.org.
- Li Shen (mathshenli@gmail.com) is with Tencent AI Lab, Shenzhen, China.
- Bernard Ghanem (bernard.ghanem@kaust.edu.sa) is with Visual Computing Center, King Abdullah University of Science and Technology (KAUST), Saudi Arabia.

Manuscript received 01 JAN. 2018; revised 01 JAN. 2018; accepted 01 JAN. 2018. Date of publication 01 JAN. 2018; date of current version 0.0000. Recommended for acceptance by XXX XXX. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 00.0000/JOURNAL.2018.0000000

solving a first-order majorization/surrogate function via closed form updates. Their algorithm design and convergence result cannot be applied here. In contrast, our method does not rely on computing the Lipschitz constant step size, yet it adopts a triangle matrix factorization strategy, where the triangle subproblem can be solved by an alternating cyclic coordinate strategy.

We are aware that matrix splitting algorithm has been considered to solve symmetric linear complementarity problems [17], [27], [28] and second-order cone complementarity problems [53] in the literature. However, we focus on minimizing a general separable nonsmooth composite function which is different from theirs. In addition, our algorithm is derived from a novel triangle operator mapping, which can be computed exactly using a new Gaussian elimination procedure. It is worthwhile to mention that matrix splitting has been extended to operator splitting recently to solve multi-term nonsmooth convex composite optimization problems [38].

Contributions. The contributions of this paper are three-fold.

(i) We propose a new Generalized Matrix Splitting Algorithm (GMSA) for composite function minimization (See Section 2). Our method is derived from a novel triangle proximal operator (See Subsection 2.1). We establish the global convergence, convergence rate, and iteration complexity of GMSA for convex problems (See Subsection 2.2). (ii) We present several important extensions of GMSA (see Section 3), including a new correction strategy to achieve point-wise monotone/contraction property for the proposed method (See Subsection 3.1), using Richardson extrapolation technique to further accelerating GMSA (See Subsection 3.2), extending GMSA to solve nonconvex problems with global convergence guarantee (See Subsection 3.3), adapting GMSA to minimize non-quadratic functions (See Subsection 3.4), and incorporating GMSA into the general optimization framework of Alternating Direction Method of Multipliers (ADMM) (See Subsection 3.5). (iii) Our extensive experiments on nonnegative matrix factorization, sparse coding and Danzig selectors have shown that GMSA achieves state-of-the-art performance in term of both efficiency and efficacy (See Section 4). A preliminary version of this paper appeared in [49].

Notation. We use lowercase and uppercase boldfaced letters to denote real vectors and matrices, respectively. The Euclidean inner product between \mathbf{x} and \mathbf{y} is denoted by $\langle \mathbf{x}, \mathbf{y} \rangle$ or $\mathbf{x}^T \mathbf{y}$. We denote $\|\mathbf{x}\| = \|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$, $\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\mathbf{x}^T \mathbf{A} \mathbf{x}}$, and $\|\mathbf{C}\|$ as the spectral norm (*i.e.* the largest singular value) of \mathbf{C} . We denote the i^{th} element of vector \mathbf{x} as x_i and the $(i, j)^{\text{th}}$ element of matrix \mathbf{C} as $C_{i,j}$. $\text{diag}(\mathbf{D}) \in \mathbb{R}^n$ is a column vector formed from the main diagonal of $\mathbf{D} \in \mathbb{R}^{n \times n}$. $\mathbf{C} \succeq 0$ and $\mathbf{C} \succ 0$ indicate that the matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ is positive semidefinite and positive definite, respectively. Here \mathbf{C} is not necessarily symmetric¹. We denote \mathbf{D} as a diagonal matrix of \mathbf{A} and \mathbf{L} as a strictly lower triangle matrix of \mathbf{A} ². Thus, we have $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{L}^T$. Throughout this paper, \mathbf{x}^k denotes the value of \mathbf{x} at k -th iteration if $\mathbf{x} \in \mathbb{R}^n$ is a variable, and x^k denotes the k -th power of x if $x \in \mathbb{R}$ is a constant scalar. We use $\nabla q(\mathbf{x})$ and $\partial h(\mathbf{x})$ to denote the gradient of $q(\cdot)$ and the sub-gradient of $h(\cdot)$ in \mathbf{x} , respectively. Finally, we use \mathbf{x}^* to denote any solution of the optimal solution set of (1).

1. $\mathbf{C} \succeq 0 \Leftrightarrow \forall \mathbf{x}, \mathbf{x}^T \mathbf{C} \mathbf{x} \geq 0 \Leftrightarrow \forall \mathbf{x}, \frac{1}{2} \mathbf{x}^T (\mathbf{C} + \mathbf{C}^T) \mathbf{x} \geq 0$

2. For example, when $n = 3$, \mathbf{D} and \mathbf{L} take the following form:

$$\mathbf{D} = \begin{bmatrix} \mathbf{A}_{1,1} & 0 & 0 \\ 0 & \mathbf{A}_{2,2} & 0 \\ 0 & 0 & \mathbf{A}_{3,3} \end{bmatrix}, \mathbf{L} = \begin{bmatrix} 0 & 0 & 0 \\ \mathbf{A}_{2,1} & 0 & 0 \\ \mathbf{A}_{3,1} & \mathbf{A}_{3,2} & 0 \end{bmatrix}.$$

For simplicity, we denote:

$$\begin{aligned} \mathbf{r}^k &\triangleq \mathbf{x}^k - \mathbf{x}^*, \mathbf{d}^k \triangleq \mathbf{x}^{k+1} - \mathbf{x}^k \\ u^k &\triangleq f(\mathbf{x}^k) - f(\mathbf{x}^*), f^k \triangleq f(\mathbf{x}^k), f^* \triangleq f(\mathbf{x}^*) \end{aligned}$$

2 PROPOSED ALGORITHM

This section presents our proposed Generalized Matrix Splitting Algorithm (GMSA) for solving (1). Throughout this section, we assume that $h(\mathbf{x})$ is convex and postpone the discussion for nonconvex $h(\mathbf{x})$ in Section 3.3.

Our solution algorithm is derived from a fixed-point iterative method based on the first-order optimal condition of (1). It is not hard to validate that a solution \mathbf{x} is the optimal solution of (1) if and only if \mathbf{x} satisfies the following nonlinear equation (“ \triangleq ” means define):

$$\mathbf{0} \in \partial f(\mathbf{x}) = \nabla q(\mathbf{x}) + \partial h(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b} + \partial h(\mathbf{x}) \quad (2)$$

In numerical analysis, a point \mathbf{x} is called a fixed point if it satisfies the equation $\mathbf{x} \in \mathcal{T}(\mathbf{x})$, for some operator $\mathcal{T}(\cdot)$. By converting the transcendental equation $\mathbf{0} \in \partial f(\mathbf{x})$ algebraically into the form $\mathbf{x} \in \mathcal{T}(\mathbf{x})$, we obtain the following iterative scheme with recursive relation:

$$\mathbf{x}^{k+1} \in \mathcal{T}(\mathbf{x}^k), \quad k = 0, 1, 2, \dots \quad (3)$$

We now discuss how to adapt our algorithm into the iterative scheme in (3). First, we split the matrix \mathbf{A} in (2) using the following strategy:

$$\mathbf{A} = \underbrace{\mathbf{L} + \frac{1}{\omega} \mathbf{D} + \epsilon \mathbf{I}}_{\mathbf{B}} + \underbrace{\mathbf{L}^T + \frac{\omega-1}{\omega} \mathbf{D} - \epsilon \mathbf{I}}_{\mathbf{C}} \quad (4)$$

Here, $\omega \in (0, 2)$ is a relaxation parameter and $\epsilon \in [0, \infty)$ is a parameter for strong convexity that enforces $\text{diag}(\mathbf{B}) > \mathbf{0}$. These parameters are specified by the user beforehand. Using these notations, we obtain the following optimality condition which is equivalent to (2):

$$-\mathbf{C}\mathbf{x} - \mathbf{b} \in (\mathbf{B} + \partial h)(\mathbf{x})$$

Then, we have the following equivalent fixed-point equation:

$$\mathbf{x} \in \mathcal{T}(\mathbf{x}; \mathbf{A}, \mathbf{b}, h) \triangleq (\mathbf{B} + \partial h)^{-1}(-\mathbf{C}\mathbf{x} - \mathbf{b}) \quad (5)$$

For notation simplicity, we denote $\mathcal{T}(\mathbf{x}; \mathbf{A}, \mathbf{b}, h)$ as $\mathcal{T}(\mathbf{x})$ since $\{\mathbf{A}, \mathbf{b}, h\}$ can be known from the context.

We name \mathcal{T} as the triangle proximal operator, which is novel in this work³. Due to the triangle property of the matrix \mathbf{B} and the element-wise separable structure of $h(\cdot)$, the triangle proximal operator $\mathcal{T}(\mathbf{x})$ in (5) can be computed exactly and analytically, by a generalized Gaussian elimination procedure (discussed later in Section 2.1). Our generalized matrix splitting algorithm iteratively applies $\mathbf{x}^{k+1} \leftarrow \mathcal{T}(\mathbf{x}^k)$ until convergence. We summarize our algorithm in Algorithm 1.

In what follows, we show how to compute $\mathcal{T}(\mathbf{x})$ in (5) in Section 2.1, and then we study the convergence properties of Algorithm 1 in Section 2.2.

3. This is in contrast with Moreau’s proximal operator [33]: $\text{prox}_h(\mathbf{a}) = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{a}\|_2^2 + h(\mathbf{x}) = (\mathbf{I} + \partial h)^{-1}(\mathbf{a})$, where the mapping $(\mathbf{I} + \partial h)^{-1}$ is called the resolvent of the subdifferential operator ∂h .

Algorithm 1 GMSA: A Generalized Matrix Splitting Algorithm for Solving the Composite Function Minimization Problem in (1)

- 1: Choose suitable parameters $\{\omega, \epsilon\}$. Initialize \mathbf{x}^0 , $k = 0$.
- 2: while not converge
- 3: $\mathbf{x}^{k+1} = \mathcal{T}(\mathbf{x}^k)$ (Solve (6) by Algorithm 2)
- 4: $k = k + 1$
- 5: end while
- 6: Output \mathbf{x}^{k+1}

2.1 Computing the Triangle Proximal Operator

We now present how to compute the triangle proximal operator in (5), which is based on a new generalized Gaussian elimination procedure. Notice that (5) seeks a solution $\mathbf{z}^* \triangleq \mathcal{T}(\mathbf{x}^k)$ that satisfies the following nonlinear system:

$$\mathbf{0} \in \mathbf{B}\mathbf{z}^* + \mathbf{u} + \partial h(\mathbf{z}^*), \text{ where } \mathbf{u} = \mathbf{b} + \mathbf{C}\mathbf{x}^k \quad (6)$$

By taking advantage of the triangular form of \mathbf{B} and the element-wise/decomposable structure of $h(\cdot)$, the elements of \mathbf{z}^* can be computed sequentially using forward substitution. Specifically, the above equation can be written as a system of nonlinear equations:

$$\mathbf{0} \in \begin{bmatrix} \mathbf{B}_{1,1} & 0 & 0 & 0 & 0 \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ \mathbf{B}_{n-1,1} & \mathbf{B}_{n-1,2} & \cdots & \mathbf{B}_{n-1,n-1} & 0 \\ \mathbf{B}_{n,1} & \mathbf{B}_{n,2} & \cdots & \mathbf{B}_{n,n-1} & \mathbf{B}_{n,n} \end{bmatrix} \begin{bmatrix} \mathbf{z}_1^* \\ \mathbf{z}_2^* \\ \vdots \\ \mathbf{z}_{n-1}^* \\ \mathbf{z}_n^* \end{bmatrix} + \mathbf{u} + \partial h(\mathbf{z}^*)$$

If \mathbf{z}^* satisfies the equations above, it must solve the following one-dimensional subproblems:

$$\mathbf{0} \in \mathbf{B}_{j,j}\mathbf{z}_j^* + \mathbf{w}_j + \partial h_j(\mathbf{z}_j^*), \forall j = 1, 2, \dots, n, \\ \mathbf{w}_j = \mathbf{u}_j + \sum_{i=1}^{j-1} \mathbf{B}_{j,i}\mathbf{z}_i^*$$

This is equivalent to solving the following one-dimensional problem for all $j = 1, 2, \dots, n$:

$$\mathbf{z}_j^* = t^* \triangleq \arg \min_t \frac{1}{2} \mathbf{B}_{j,j} t^2 + \mathbf{w}_j t + h_j(t) \quad (7)$$

Note that the computation of \mathbf{z}^* uses only the elements of \mathbf{z}^* that have already been computed and a successive displacement strategy is applied to find \mathbf{z}^* .

We remark that the one-dimensional subproblem in (7) often admits a closed form solution for many problems of interest. For example, when $h_j(t) = I_{[lb, ub]}(t)$, $\forall j = 1, 2, \dots, n$ with $I_{[lb, ub]}(t)$ denoting an indicator function on the box constraint $lb \leq t \leq ub$, the optimal solution can be computed as: $t^* = \min(ub, \max(lb, -\mathbf{w}_j/\mathbf{B}_{j,j}))$; when $h_j(t) = \lambda|t|$, $\forall j = 1, 2, \dots, n$ (i.e. in the case of the ℓ_1 norm), the optimal solution can be computed as: $t^* = -\max(0, |\mathbf{w}_j/\mathbf{B}_{j,j}| - \lambda/\mathbf{B}_{j,j}) \cdot \text{sign}(\mathbf{w}_j/\mathbf{B}_{j,j})$.

Our generalized Gaussian elimination procedure for computing $\mathcal{T}(\mathbf{x}^k)$ is summarized in Algorithm 2. Note that its computational complexity is $\mathcal{O}(n^2)$, which is the same as computing a matrix-vector product.

2.2 Convergence Analysis

In what follows, we present our convergence analysis for Algorithm 1. The following lemma characterizes the optimality of the triangle proximal operator $\mathcal{T}(\mathbf{x})$ for any \mathbf{x} .

Algorithm 2 A Generalized Gaussian Elimination Procedure for Computing the Triangle Proximal Operator $\mathcal{T}(\mathbf{x}^k)$.

- 1: Input \mathbf{x}^k
- 2: Initialization: compute $\mathbf{u} = \mathbf{b} + \mathbf{C}\mathbf{x}^k$
- 3: $\mathbf{x}_1 = \arg \min_t \frac{1}{2} \mathbf{B}_{1,1} t^2 + (\mathbf{u}_1)t + h_1(t)$
- 4: $\mathbf{x}_2 = \arg \min_t \frac{1}{2} \mathbf{B}_{2,2} t^2 + (\mathbf{u}_2 + \mathbf{B}_{2,1}\mathbf{x}_1)t + h_2(t)$
- 5: $\mathbf{x}_3 = \arg \min_t \frac{1}{2} \mathbf{B}_{3,3} t^2 + (\mathbf{u}_3 + \mathbf{B}_{3,1}\mathbf{x}_1 + \mathbf{B}_{3,2}\mathbf{x}_2)t + h_3(t)$
- 6: ...
- 7: $\mathbf{x}_n = \arg \min_t \frac{1}{2} \mathbf{B}_{n,n} t^2 + (\mathbf{u}_n + \sum_{i=1}^{n-1} \mathbf{B}_{n,i}\mathbf{x}_i)t + h_n(t)$
- 8: Collect $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n)^T$ as \mathbf{x}^{k+1} and Output \mathbf{x}^{k+1}

Lemma 1. For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, it holds that:

$$(i) \mathbf{0} \in \nabla q(\mathcal{T}(\mathbf{x})) + \partial h(\mathcal{T}(\mathbf{x})) + \mathbf{C}(\mathbf{x} - \mathcal{T}(\mathbf{x})) \quad (8)$$

$$(ii) h(\mathcal{T}(\mathbf{x})) - h(\mathbf{y}) + \langle \nabla q(\mathcal{T}(\mathbf{x})), \mathcal{T}(\mathbf{x}) - \mathbf{y} \rangle \leq \langle \mathbf{C}(\mathcal{T}(\mathbf{x}) - \mathbf{x}), \mathcal{T}(\mathbf{x}) - \mathbf{y} \rangle \quad (9)$$

$$(iii) f(\mathcal{T}(\mathbf{x})) - f(\mathbf{y}) \leq \langle \mathbf{C}(\mathcal{T}(\mathbf{x}) - \mathbf{x}), \mathcal{T}(\mathbf{x}) - \mathbf{y} \rangle - \frac{1}{2} \|\mathcal{T}(\mathbf{x}) - \mathbf{y}\|_{\mathbf{A}}^2 \quad (10)$$

Proof. (i) Using the optimality of $\mathcal{T}(\mathbf{x})$ in (6), we derive the following results:

$$\begin{aligned} \mathbf{0} &\in \mathbf{B}\mathcal{T}(\mathbf{x}) + \partial h(\mathcal{T}(\mathbf{x})) + \mathbf{b} + \mathbf{C}\mathbf{x} \\ &\stackrel{(a)}{\Rightarrow} \mathbf{0} \in \mathbf{A}\mathcal{T}(\mathbf{x}) + \partial h(\mathcal{T}(\mathbf{x})) + \mathbf{b} + \mathbf{C}(\mathbf{x} - \mathcal{T}(\mathbf{x})) \\ &\stackrel{(b)}{\Rightarrow} \mathbf{0} \in \nabla q(\mathcal{T}(\mathbf{x})) + \partial h(\mathcal{T}(\mathbf{x})) + \mathbf{C}(\mathbf{x} - \mathcal{T}(\mathbf{x})) \end{aligned}$$

where step (a) uses $\mathbf{B} = \mathbf{A} - \mathbf{C}$ and step (b) uses the definition of $\nabla q(\cdot)$ in (2).

(ii) Since $h(\cdot)$ is convex, we have:

$$\forall \mathbf{s}, \mathbf{z}, h(\mathbf{s}) - h(\mathbf{z}) \leq \langle \mathbf{o}, \mathbf{s} - \mathbf{z} \rangle, \forall \mathbf{o} \in \partial h(\mathbf{s}). \quad (11)$$

We define $\mathbf{s} \triangleq \mathcal{T}(\mathbf{x})$ and derive the following inequalities:

$$\begin{aligned} h(\mathbf{s}) - h(\mathbf{y}) &\stackrel{(a)}{\leq} \langle \mathbf{o}, \mathbf{s} - \mathbf{y} \rangle, \forall \mathbf{o} \in \partial h(\mathbf{s}) \\ &\stackrel{(b)}{\leq} \langle -\nabla q(\mathbf{s}) - \mathbf{C}(\mathbf{x} - \mathbf{s}), \mathbf{s} - \mathbf{y} \rangle \end{aligned}$$

where step (a) uses (11) with $\mathbf{z} = \mathbf{y}$; step (b) uses (8).

(iii) Since $q(\cdot)$ is a quadratic function, we have:

$$\forall \mathbf{s}, \mathbf{z}, q(\mathbf{s}) - q(\mathbf{z}) = \langle \nabla q(\mathbf{s}), \mathbf{s} - \mathbf{z} \rangle - \frac{1}{2} \|\mathbf{s} - \mathbf{z}\|_{\mathbf{A}}^2 \quad (12)$$

We define $\mathbf{s} \triangleq \mathcal{T}(\mathbf{x})$ and derive the following results:

$$\begin{aligned} f(\mathbf{s}) - f(\mathbf{y}) &\stackrel{(a)}{=} h(\mathbf{s}) - h(\mathbf{y}) + q(\mathbf{s}) - q(\mathbf{y}) \\ &\stackrel{(b)}{=} h(\mathbf{s}) - h(\mathbf{y}) + \langle \nabla q(\mathbf{s}), \mathbf{s} - \mathbf{y} \rangle - \frac{1}{2} \|\mathbf{s} - \mathbf{y}\|_{\mathbf{A}}^2 \\ &\stackrel{(c)}{\leq} \langle \mathbf{C}(\mathbf{s} - \mathbf{x}), \mathbf{s} - \mathbf{y} \rangle - \frac{1}{2} \|\mathbf{s} - \mathbf{y}\|_{\mathbf{A}}^2 \end{aligned}$$

where step (a) uses the definition of $f(\cdot)$; step (b) uses (12) with $\mathbf{z} = \mathbf{y}$; step (c) uses (9). \square

Remarks. Both (8) and (9) can be used to characterize the optimality of (1). Recall that we have the following sufficient and necessary conditions for the optimal solution: \mathbf{x}^* is the optimal solution $\Leftrightarrow \mathbf{0} \in \nabla q(\mathcal{T}(\mathbf{x}^*)) + \partial h(\mathcal{T}(\mathbf{x}^*)) \Leftrightarrow \langle \nabla q(\mathcal{T}(\mathbf{x}^*)), \mathcal{T}(\mathbf{x}^*) - \mathbf{y} \rangle + h(\mathcal{T}(\mathbf{x}^*)) - h(\mathbf{y}) \leq 0, \forall \mathbf{y}$.

When $\mathbf{x} = \mathcal{T}(\mathbf{x})$ occurs, (8) and (9) coincide with the optimal condition and one can conclude that \mathbf{x} is the optimal solution.

Theorem 1. (Proof of Global Convergence) We define $\delta \triangleq 2\epsilon + \frac{2-\omega}{\omega} \min(\text{diag}(\mathbf{D}))$ and let $\{\omega, \epsilon\}$ be chosen such that $\delta \in (0, \infty)$. Algorithm 1 is globally convergent.

Proof. (i) First, the following results hold for all $\mathbf{z} \in \mathbb{R}^n$:

$$\begin{aligned} \mathbf{z}^T(\mathbf{A} - 2\mathbf{C})\mathbf{z} &= \mathbf{z}^T(\mathbf{B} - \mathbf{C})\mathbf{z} \\ &= \mathbf{z}^T(\mathbf{L} - \mathbf{L}^T + \frac{2-\omega}{\omega}\mathbf{D} + 2\epsilon\mathbf{I})\mathbf{z} \\ &= \mathbf{z}^T(2\epsilon\mathbf{I} + \frac{2-\omega}{\omega}\mathbf{D})\mathbf{z} \geq \delta\|\mathbf{z}\|_2^2 \end{aligned} \quad (13)$$

where we have used the definition of \mathbf{A} and \mathbf{C} , and the fact that $\mathbf{z}^T\mathbf{L}\mathbf{z} = (\mathbf{z}^T\mathbf{L}\mathbf{z})^T = \mathbf{z}^T\mathbf{L}^T\mathbf{z}$, $\forall \mathbf{z}$.

We invoke (10) in Lemma 1 with $\mathbf{x} = \mathbf{x}^k$, $\mathbf{y} = \mathbf{x}^k$ and combine the inequality in (13) to obtain:

$$f^{k+1} - f^k \leq -\frac{1}{2}\langle \mathbf{d}^k, (\mathbf{A} - 2\mathbf{C})\mathbf{d}^k \rangle \leq -\frac{\delta}{2}\|\mathbf{d}^k\|_2^2 \quad (14)$$

(ii) Second, summing (14) over $i = 0, \dots, k-1$, we have:

$$\begin{aligned} \frac{\delta}{2} \sum_{i=0}^{k-1} \|\mathbf{d}^i\|_2^2 &\leq f^0 - f^k \stackrel{(a)}{\leq} f^0 - f^* \\ \Rightarrow \frac{\delta}{2} \min_{i=0, \dots, k-1} \|\mathbf{d}^i\|_2^2 &\leq (f^0 - f^*)/k \end{aligned}$$

where step (a) uses the fact that $f^* \leq f^k$. Note that f^* is bounded below. As $k \rightarrow \infty$, we have $\mathbf{d}^k \triangleq \mathbf{x}^{k+1} - \mathbf{x}^k \rightarrow \mathbf{0}$, which implies the convergence of the algorithm. By using (8) in Lemma 1 with $\mathbf{x} = \mathbf{x}^k$, we obtain: $\nabla q(\mathbf{x}^{k+1}) + \partial h(\mathbf{x}^{k+1}) \ni -\mathbf{C}(\mathbf{x}^k - \mathbf{x}^{k+1}) \rightarrow \mathbf{0}$. The fact that $\nabla q(\mathbf{x}^{k+1}) + \partial h(\mathbf{x}^{k+1}) \ni \mathbf{0}$ implies that \mathbf{x}^{k+1} is the global optimal solution of the convex problem.

Note that guaranteeing $\delta \in (0, \infty)$ can be achieved by simply choosing $\omega \in (0, 2)$ and setting ϵ to a small number. \square

Remarks. (i) When $h(\cdot)$ is empty and $\epsilon = 0$, Algorithm 1 reduces to the classical Gauss-Seidel method ($\omega = 1$) and Successive Over-Relaxation method ($\omega \neq 1$). (ii) When \mathbf{A} contains zeros in its diagonal entries, one needs to set ϵ to a strictly positive number. This is to guarantee the strong convexity of the one dimensional subproblem and a bounded solution for any $h(\cdot)$ in (7). The introduction of the parameter ϵ is novel in this paper and it removes the assumption that \mathbf{A} is strictly positive-definite or strictly diagonally dominant, which is used in the classical result of GS and SOR method [8], [37].

We now prove the convergence rate of Algorithm 1. We make the following assumption, which characterizes the relations between $\mathcal{T}(\mathbf{x})$ and \mathbf{x}^* for any \mathbf{x} .

Assumption 1. If \mathbf{x} is not the optimum of (1), there exists a constant $\eta \in (0, \infty)$ such that $\|\mathbf{x} - \mathbf{x}^*\| \leq \eta\|\mathbf{x} - \mathcal{T}(\mathbf{x})\|$.

Remarks. Assumption 1 is similar to the classical local proximal error bound assumption in the literature [29], [41], [42], [51], and it is mild. Firstly, if \mathbf{x} is not the optimum, we have $\mathbf{x} \neq \mathcal{T}(\mathbf{x})$. This is because when $\mathbf{x} = \mathcal{T}(\mathbf{x})$, we have $\mathbf{0} = -\mathbf{C}(\mathbf{x} - \mathcal{T}(\mathbf{x})) \in \nabla q(\mathcal{T}(\mathbf{x})) + \partial h(\mathcal{T}(\mathbf{x}))$ (refer to the optimal condition of $\mathcal{T}(\mathbf{x})$ in (8)), which contradicts with the condition that \mathbf{x} is not the optimal solution. Secondly, by the boundedness of \mathbf{x} and \mathbf{x}^* , there exists a sufficiently large constant $\eta \in (0, \infty)$ such that $\|\mathbf{x} - \mathbf{x}^*\| \leq \eta\|\mathbf{x} - \mathcal{T}(\mathbf{x})\|$.

We now prove the convergence rate of Algorithm 1.

Theorem 2. (Proof of Convergence Rate) We define $\delta \triangleq 2\epsilon + \frac{2-\omega}{\omega} \min(\text{diag}(\mathbf{D}))$ and let $\{\omega, \epsilon\}$ be chosen such that $\delta \in (0, \infty)$. Assuming that \mathbf{x}^k is bound for all k , we have:

$$f(\mathbf{x}^k) - f(\mathbf{x}^*) \leq \left(\frac{C_1}{1+C_1}\right)^k [f(\mathbf{x}^0) - f(\mathbf{x}^*)], \quad (15)$$

$$\|\mathbf{x}^k - \mathbf{x}^{k+1}\|_2^2 \leq \frac{2}{\delta} \left(\frac{C_1}{1+C_1}\right)^k [f(\mathbf{x}^0) - f(\mathbf{x}^*)]. \quad (16)$$

where $C_1 \triangleq 2\|\mathbf{B}\|/\eta/\delta - 1$.

Proof. By invoking the assumption 1 with $\mathbf{x} = \mathbf{x}^k$, we obtain:

$$\|\mathbf{x}^k - \mathbf{x}^*\| \leq \eta\|\mathbf{x}^k - \mathcal{T}(\mathbf{x}^k)\| \Rightarrow \|\mathbf{r}^k\| \leq \eta\|\mathbf{d}^k\| \quad (17)$$

We derive the following inequalities:

$$\begin{aligned} f^{k+1} - f^* &\stackrel{(a)}{\leq} \langle \mathbf{r}^{k+1}, \mathbf{C}\mathbf{d}^k \rangle - \frac{1}{2}\langle \mathbf{r}^{k+1}, \mathbf{A}\mathbf{r}^{k+1} \rangle \\ &\stackrel{(b)}{=} \langle \mathbf{r}^k, (\mathbf{C} - \mathbf{A})\mathbf{d}^k \rangle - \frac{1}{2}\|\mathbf{r}^k\|_{\mathbf{A}}^2 + \frac{1}{2}\|\mathbf{d}^k\|_{2\mathbf{C}-\mathbf{A}}^2 \\ &\stackrel{(c)}{\leq} -\langle \mathbf{r}^k, \mathbf{B}\mathbf{d}^k \rangle + 0 - \frac{\delta}{2}\|\mathbf{d}^k\|_2^2 \\ &\stackrel{(d)}{\leq} \|\mathbf{r}^k\| \|\mathbf{B}\| \|\mathbf{d}^k\| - \frac{\delta}{2}\|\mathbf{d}^k\|_2^2 \stackrel{(e)}{\leq} (\eta\|\mathbf{B}\| - \frac{\delta}{2})\|\mathbf{d}^k\|_2^2 \\ &\stackrel{(f)}{\leq} (\eta\|\mathbf{B}\| - \frac{\delta}{2})\frac{2}{\delta}(f^k - f^{k+1}) \stackrel{(g)}{=} C_1(f^k - f^{k+1}) \end{aligned} \quad (18)$$

where step (a) uses (10) in Lemma 1 with $\mathbf{x} = \mathbf{x}^*$, $\mathbf{y} = \mathbf{x}^k$; step (b) uses the fact that $\mathbf{r}^{k+1} = \mathbf{r}^k + \mathbf{d}^k$ and $\mathbf{A} = \mathbf{B} + \mathbf{C}$; step (c) uses $\mathbf{A} \succeq \mathbf{0}$ and the inequality that $\mathbf{A} - 2\mathbf{C} \succeq \delta\mathbf{I}$ which is due to (13); step (d) uses the Cauchy-Schwarz inequality $\langle \mathbf{x}, \mathbf{y} \rangle \leq \|\mathbf{x}\| \|\mathbf{y}\|$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and the norm inequality $\|\mathbf{B}\mathbf{x}\| \leq \|\mathbf{B}\| \|\mathbf{x}\|$, $\forall \mathbf{x} \in \mathbb{R}^n$; step (e) uses (17); step (f) uses the descent condition in (14); step (g) uses the definition of C_1 .

Rearranging the last inequality in (19), we have $f^{k+1} - f^* \leq C_1(f^k - f^{k+1}) = C_1(f^k - f^*) - C_1(f^{k+1} - f^*) \Rightarrow (1 + C_1)[f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*)] \leq C_1[f(\mathbf{x}^k) - f(\mathbf{x}^*)]$, leading to: $\frac{f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*)}{f(\mathbf{x}^k) - f(\mathbf{x}^*)} \leq \frac{C_1}{1+C_1} < 1$. Solving this recursive formulation, we obtain (15). In other words, the sequence $\{f(\mathbf{x}^k)\}_{k=0}^{\infty}$ converges to $f(\mathbf{x}^*)$ linearly in the quotient sense. Using (14), we derive the following inequalities: $\|\mathbf{x}^k - \mathbf{x}^{k+1}\|_2^2 \leq \frac{2(f^k - f^{k+1})}{\delta} \leq \frac{2(f^k - f^*)}{\delta}$. Therefore, we obtain (16). \square

The following lemma is useful in our proof of iteration complexity.

Lemma 2. Suppose a nonnegative sequence $\{u^k\}_{k=0}^{\infty}$ satisfies $u^{k+1} \leq -2C + 2C\sqrt{1 + \frac{u^k}{C}}$ for some constant $C > 0$. It holds that: $u^{k+1} \leq \frac{\max(8C, \sqrt{4Cu^0})}{k+1}$.

Proof. The proof of this lemma can be obtained by mathematical induction. We denote $\chi \triangleq \max(8C, \sqrt{4Cu^0})$. (i) When $k = 0$, we have $u^1 \leq -2C + 2C\sqrt{1 + \frac{u^0}{C}} \leq -2C + 2C(1 + \sqrt{\frac{u^0}{C}}) = 2\sqrt{Cu^0} \leq \frac{\chi}{k+1}$. (ii) When $k \geq 1$, we assume that $u^k \leq \frac{\chi}{k}$ holds. We derive the following results: $k \geq 1 \Rightarrow \frac{k+1}{k} \leq 2 \stackrel{(a)}{\Rightarrow} 4C\frac{k+1}{k} \leq 8C \leq \chi \stackrel{(b)}{\Rightarrow} \frac{4C}{k(k+1)} = 4C(\frac{1}{k} - \frac{1}{k+1}) \leq \frac{\chi}{(k+1)^2} \Rightarrow \frac{4C}{k} \leq \frac{4C}{k+1} + \frac{\chi}{(k+1)^2} \Rightarrow 4C^2(1 + \frac{\chi}{kC}) \leq 4C^2 + \frac{4\chi C}{k+1} + \frac{\chi^2}{(k+1)^2} \Rightarrow 2C\sqrt{1 + \frac{\chi}{kC}} \leq 2C + \frac{\chi}{k+1} \Rightarrow -2C + 2C\sqrt{1 + \frac{\chi}{kC}} \leq \frac{\chi}{k+1}$

$\stackrel{(c)}{\Rightarrow} -2C + 2C\sqrt{1 + \frac{u^k}{C}} \leq \frac{\chi}{k+1} \Rightarrow u^{k+1} \leq \frac{\chi}{k+1}$. Here, step (a) uses $8C \leq \chi$; step (b) uses the inequality that $\frac{1}{k(k+1)} = \frac{1}{k} - \frac{1}{k+1}$; step (c) uses $u^k \leq \frac{\chi}{k}$.

□

We now prove the iteration complexity of Algorithm 1.

Theorem 3. (Proof of Iteration Complexity) We define $\delta \triangleq 2\epsilon + \frac{2-\omega}{\omega} \min(\text{diag}(\mathbf{D}))$ and let $\{\omega, \epsilon\}$ be chosen such that $\delta \in (0, \infty)$. Assuming that $\|\mathbf{x}^k\| \leq R$ for all k , we have:

$$f^k - f^* \leq \begin{cases} u^0 \left(\frac{2C_3}{2C_3+1} \right)^k, & \text{if } \sqrt{f^k - f^{k+1}} \geq \frac{C_2}{C_3}, \forall k \leq \check{k} \\ \frac{C_4}{k}, & \text{if } \sqrt{f^k - f^{k+1}} < \frac{C_2}{C_3}, \text{ else} \end{cases}$$

where $C_2 \triangleq 2R\|\mathbf{C}\|\sqrt{2/\delta}$, $C_3 \triangleq \frac{2}{\delta}\|\mathbf{C}\|$, $C_4 \triangleq \max(8C_2^2, \sqrt{4C_2^2}u^0)$, and k is some unknown iteration index.

Proof. We have the following inequalities:

$$\begin{aligned} & u^{k+1} \\ & \stackrel{(a)}{\leq} \langle \mathbf{r}^{k+1}, \mathbf{C}\mathbf{d}^k \rangle - \frac{1}{2} \langle \mathbf{r}^{k+1}, \mathbf{A}\mathbf{r}^{k+1} \rangle \\ & \stackrel{(b)}{\leq} \langle \mathbf{r}^k + \mathbf{d}^k, \mathbf{C}\mathbf{d}^k \rangle + 0 \\ & \stackrel{(d)}{\leq} \|\mathbf{r}^k\| \cdot \|\mathbf{C}\| \cdot \|\mathbf{d}^k\| + \|\mathbf{C}\| \cdot \|\mathbf{d}^k\|_2^2 \\ & \stackrel{(d)}{\leq} 2R\|\mathbf{C}\| \cdot \|\mathbf{d}^k\| + \|\mathbf{C}\| \cdot \|\mathbf{d}^k\|_2^2 \\ & \stackrel{(e)}{\leq} 2R\|\mathbf{C}\| \cdot \sqrt{\frac{2}{\delta}(u^k - u^{k+1})} + \|\mathbf{C}\| \cdot \frac{2}{\delta} \cdot (u^k - u^{k+1}) \\ & \stackrel{(f)}{=} C_2\sqrt{u^k - u^{k+1}} + C_3(u^k - u^{k+1}) \end{aligned} \quad (20)$$

where step (a) uses (18); step (b) uses the fact that $\mathbf{r}^{k+1} = \mathbf{r}^k + \mathbf{d}^k$, $\mathbf{A} \succeq \mathbf{0}$; step (c) uses the Cauchy-Schwarz inequality and the norm inequality; step (d) uses the fact that $\|\mathbf{r}^k\|_2 \leq \|\mathbf{x}^k\|_2 + \|\mathbf{x}^*\|_2 \leq 2R$; step (e) uses (14); step (f) uses the definition of C_2 and C_3 .

Now we consider the two cases for the recursion formula in (20): (i) $\sqrt{u^k - u^{k+1}} \geq \frac{C_2}{C_3}$ for some $k \leq \check{k}$ (ii) $\sqrt{u^k - u^{k+1}} \leq \frac{C_2}{C_3}$ for some $k > \check{k}$. In case (i), (20) implies that we have $u^{k+1} \leq 2C_3(u^k - u^{k+1})$ and rearranging terms gives: $u^{k+1} \leq \frac{2C_3}{2C_3+1}u^k$. Thus, we have: $u^{k+1} \leq \left(\frac{2C_3}{2C_3+1}\right)^{k+1}u^0$. We now focus on case (ii). When $\sqrt{u^k - u^{k+1}} \leq \frac{C_2}{C_3}$, (20) implies that we have $u^{k+1} \leq 2C_2\sqrt{u^k - u^{k+1}}$ and rearranging terms yields: $\frac{(u^{k+1})^2}{4C_2^2} + u^{k+1} \leq u^k$. Solving this quadratic inequality, we have: $u^{k+1} \leq -2C_2^2 + 2C_2^2\sqrt{1 + \frac{1}{C_2^2}u^k}$; solving this recursive formulation by Lemma 2, we obtain $u^{k+1} \leq \frac{C_4}{k+1}$.

□

Remarks. The convergence result in Theorem 3 is weaker than that in Theorem 2, however, it does not rely on Assumption 1 and the unknown constant η .

We now derive the convergence rate when $q(\cdot)$ is strongly convex.

Theorem 4. (Proof of Convergence Rate when $q(\cdot)$ is Strongly Convex) We define $\delta \triangleq 2\epsilon + \frac{2-\omega}{\omega} \min(\text{diag}(\mathbf{D}))$ and let $\{\omega, \epsilon\}$ be chosen such that $\delta \in (0, \infty)$. Assuming that $q(\mathbf{x})$ is strongly convex with respect to \mathbf{x} such that $\mathbf{A} \succeq \sigma\mathbf{I}$ with $\sigma > 0$, we have:

$$f(\mathbf{x}^k) - f(\mathbf{x}^*) \leq \left(\frac{C_5}{1+C_5}\right)^k [f(\mathbf{x}^0) - f(\mathbf{x}^*)], \quad (21)$$

$$\|\mathbf{x}^k - \mathbf{x}^*\|_2^2 \leq \frac{8\|\mathbf{C}\|^2}{\sigma^2\delta} \left(\frac{C_5}{1+C_5}\right)^{k-1} [f(\mathbf{x}^0) - f(\mathbf{x}^*)]. \quad (22)$$

where $C_5 \triangleq \|\mathbf{C}\|^2/(\delta\sigma)$.

Proof. By using (10) in Lemma 1 with $\mathbf{x} = \mathbf{x}^k$, $\mathbf{y} = \mathbf{x}^*$, we derive the following inequalities:

$$\begin{aligned} & f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*) \\ & \leq \langle \mathbf{C}(\mathbf{x}^{k+1} - \mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^* \rangle - \frac{1}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|_{\mathbf{A}}^2 \\ & \leq \langle \mathbf{C}(\mathbf{x}^{k+1} - \mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^* \rangle - \frac{\sigma}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 \end{aligned} \quad (23)$$

We notice that the right-hand side in (23) is concave. Maximizing over \mathbf{x}^* , we obtain:

$$\begin{aligned} & \sigma(\mathbf{x}^* - \mathbf{x}^{k+1}) + \mathbf{C}(\mathbf{x}^{k+1} - \mathbf{x}^k) = 0 \\ & \Rightarrow \mathbf{x}^* = \mathbf{x}^{k+1} - \mathbf{C}(\mathbf{x}^{k+1} - \mathbf{x}^k)/\sigma \end{aligned} \quad (24)$$

Putting (24) into (23), we further derive the following inequalities:

$$\begin{aligned} f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*) & \leq \frac{\|\mathbf{C}(\mathbf{x}^{k+1} - \mathbf{x}^k)\|_2^2}{2\sigma} \stackrel{(a)}{\leq} \frac{\|\mathbf{C}\|^2 \cdot \|\mathbf{x}^k - \mathbf{x}^{k+1}\|_2^2}{2\sigma} \\ & \stackrel{(b)}{\leq} \frac{\|\mathbf{C}\|^2 \cdot [f(\mathbf{x}^k) - f(\mathbf{x}^{k+1})]}{\delta\sigma} \stackrel{(c)}{=} C_5[f(\mathbf{x}^k) - f(\mathbf{x}^{k+1})] \end{aligned}$$

where step (a) uses the norm inequality $\|\mathbf{C}\mathbf{x}\| \leq \|\mathbf{C}\| \cdot \|\mathbf{x}\|$; step (b) uses (14); step (c) uses the definition of C_5 . Finally, we obtain: $\frac{f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*)}{f(\mathbf{x}^k) - f(\mathbf{x}^*)} \leq \frac{C_5}{1+C_5}$. Solving the recursive formulation, we obtain the result in (21).

Using the similar strategy for deriving (16), we have:

$$\|\mathbf{x}^k - \mathbf{x}^{k+1}\|_2^2 \leq \frac{2}{\delta} \left(\frac{C_5}{1+C_5}\right)^k [f(\mathbf{x}^0) - f(\mathbf{x}^*)] \quad (25)$$

Finally, we derive the following inequalities:

$$\begin{aligned} & \frac{\sigma}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 \\ & \stackrel{(a)}{\leq} \langle \mathbf{C}(\mathbf{x}^{k+1} - \mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^* \rangle + f(\mathbf{x}^*) - f(\mathbf{x}^{k+1}) \\ & \stackrel{(b)}{\leq} \langle \mathbf{C}(\mathbf{x}^{k+1} - \mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^* \rangle + 0 \\ & \stackrel{(c)}{\leq} \|\mathbf{C}\| \cdot \|\mathbf{x}^{k+1} - \mathbf{x}^*\| \cdot \|\mathbf{x}^{k+1} - \mathbf{x}^k\| \end{aligned}$$

where step (a) uses (23); step (b) uses the fact that $f(\mathbf{x}^*) \leq f(\mathbf{x}^{k+1})$; step (c) uses the norm inequality. Therefore, we obtain:

$$\frac{\sigma}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2 \leq \|\mathbf{C}\| \|\mathbf{x}^{k+1} - \mathbf{x}^k\|$$

Combining with (25), we obtain (22).

□

Remarks. Thanks to the strong convexity of $q(\cdot)$, we can characterize the convergence rate for both $\|\mathbf{x}^k - \mathbf{x}^*\|$ and $\|\mathbf{x}^k - \mathbf{x}^{k+1}\|$ in Theorem 4 without using the assumption 1 and the unknown constant η . Therefore, the convergence result in Theorem 4 is stronger than that in Theorem 2.

3 EXTENSIONS

This section discusses several extensions of our proposed generalized matrix splitting algorithm.

Algorithm 3 GMSA-C: Generalized Matrix Splitting Algorithm with Correction Strategy for Solving (1).

- 1: Choose suitable parameters $\{\omega, \epsilon\}$. Initialize \mathbf{x}^0 .
- 2: for $k = 0, 1, 2, \dots$
- 3: $\mathbf{y}^k = \mathcal{T}(\mathbf{x}^k)$
- 4: Choose a suitable parameter α^k (e.g. $\alpha^k = \frac{2\epsilon + \frac{2-\omega}{\omega} \min(\text{diag}(\mathbf{D}))}{\|\mathbf{B}^T \mathbf{B}\|}$ or $\alpha^k = \frac{\|\mathbf{y}^k - \mathbf{x}^k\|_{2\mathbf{B}-0.5\mathbf{A}}^2}{\|\mathbf{y}^k - \mathbf{x}^k\|_{2\mathbf{B}^T \mathbf{B}}^2}$)
- 5: $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{B}(\mathbf{y}^k - \mathbf{x}^k)$
- 6: end for
- 7: Output \mathbf{x}^{k+1}

3.1 Pointwise Contraction via a Correction Strategy

This section considers a new correction strategy to achieve pointwise contraction for the proposed method to solve (1). One remarkable feature of this strategy is that the resulting iterated solutions of \mathbf{x}^k always satisfy the monotone/contractive property that $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 < \|\mathbf{x}^k - \mathbf{x}^*\|_2^2$ for all k if \mathbf{x}^k is not the optimal solution. We summarize our new algorithm in Algorithm 3.

We provide detailed theoretical analysis for Algorithm 3. The following lemmas are useful in our proof.

Lemma 3. Assume that $h(\cdot)$ is convex. For all $\mathbf{x} \in \mathbb{R}^n$, it holds that:

$$\langle \mathbf{A}\mathbf{x}^* - \mathbf{A}\mathcal{T}(\mathbf{x}) + \mathbf{C}(\mathcal{T}(\mathbf{x}) - \mathbf{x}), \mathbf{x}^* - \mathcal{T}(\mathbf{x}) \rangle \leq 0.$$

Proof. By the optimality of \mathbf{x}^* , we have: $\mathbf{0} \in \nabla q(\mathbf{x}^*) + \partial h(\mathbf{x}^*)$. Combining with (8) in Lemma (1), we obtain:

$$\begin{aligned} \mathbf{0} &\in \nabla q(\mathcal{T}(\mathbf{x})) + \partial h(\mathcal{T}(\mathbf{x})) + \mathbf{C}(\mathbf{x} - \mathcal{T}(\mathbf{x})) \\ &\quad - \nabla q(\mathbf{x}^*) - \partial h(\mathbf{x}^*) \end{aligned} \quad (26)$$

Using the monotonicity of $\partial h(\cdot)$, we obtain: $\langle \mathbf{0} - \mathbf{w}, \mathbf{x}^* - \mathcal{T}(\mathbf{x}) \rangle \geq 0$, $\forall \mathbf{0} \in \partial h(\mathbf{x}^*)$, $\mathbf{w} \in \partial h(\mathcal{T}(\mathbf{x}))$. Combining with (26), we conclude this lemma. \square

Lemma 4. Assume $\mathbf{A} \succeq \mathbf{0}$. For all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^n$, it holds that:

$$\langle \mathbf{x} - \mathbf{z}, \mathbf{A}\mathbf{z} - \mathbf{A}\mathbf{y} \rangle \leq \frac{1}{4} \|\mathbf{x} - \mathbf{y}\|_{\mathbf{A}}^2.$$

Proof. Using the variable substitution that $\mathbf{z} - \mathbf{x} = \mathbf{p}$, $\mathbf{z} - \mathbf{y} = \mathbf{u}$, $\mathbf{x} - \mathbf{y} = \mathbf{u} - \mathbf{p}$, we have the following equivalent inequalities: $\langle -\mathbf{p}, \mathbf{A}\mathbf{u} \rangle \leq \frac{1}{4} \|\mathbf{u} - \mathbf{p}\|_{\mathbf{A}}^2 \Leftrightarrow \|\mathbf{u} - \mathbf{p}\|_{\mathbf{A}}^2 + 4\langle \mathbf{A}\mathbf{p}, \mathbf{u} \rangle \geq 0 \Leftrightarrow \|\mathbf{u} + \mathbf{p}\|_{\mathbf{A}}^2 \geq 0$. Clearly, these inequalities hold since $\mathbf{A} \succeq \mathbf{0}$. \square

The following theorem provides important theoretical insights on choosing suitable parameters $\{\omega, \epsilon, \alpha^k\}$ to guarantee convergence of Algorithm 3.

Theorem 5. We define $\delta \triangleq 2\epsilon + \frac{2-\omega}{\omega} \min(\text{diag}(\mathbf{D}))$ and let $\{\omega, \epsilon\}$ be chosen such that $\delta \in (0, \infty)$. Assuming that $h(\cdot)$ is convex and \mathbf{x}^k generated by Algorithm 3 is not the optimal solution, we have the following results: (i)

$$\begin{aligned} \|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^k - \mathbf{x}^*\|_2^2 &\leq \|\mathbf{y}^k - \mathbf{x}^k\|_{\mathbf{G}^k}^2 \quad (27) \\ \mathbf{G}^k &\triangleq \frac{1}{2}(\alpha^k)^2 \mathbf{P} - \alpha^k \mathbf{Q}, \quad \mathbf{P} \triangleq 2\mathbf{B}^T \mathbf{B}, \quad \mathbf{Q} \triangleq 2\mathbf{B} - \frac{1}{2}\mathbf{A} \end{aligned}$$

(ii) If we choose a global constant $0 < \alpha^k < \frac{\delta}{\|\mathbf{B}^T \mathbf{B}\|}$, we have $\mathbf{G}^k \prec \mathbf{0}$ and $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^k - \mathbf{x}^*\|_2^2 < 0$.

(iii) If we choose a local constant $\alpha^k = \frac{\|\mathbf{y}^k - \mathbf{x}^k\|_{\mathbf{Q}}^2}{\|\mathbf{y}^k - \mathbf{x}^k\|_{\mathbf{P}}^2}$, we have $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^k - \mathbf{x}^*\|_2^2 \leq -\frac{\delta^2 \|\mathbf{y}^k - \mathbf{x}^k\|_2^2}{4\|\mathbf{B}^T \mathbf{B}\|} < 0$.

Proof. (i) First of all, we derive the following inequalities:

$$\begin{aligned} &\langle \mathbf{y}^k - \mathbf{x}^*, \mathbf{B}(\mathbf{y}^k - \mathbf{x}^k) \rangle \\ &\stackrel{(a)}{=} \langle \mathbf{y}^k - \mathbf{x}^*, \mathbf{A}(\mathbf{y}^k - \mathbf{x}^k) \rangle + \langle \mathbf{y}^k - \mathbf{x}^*, \mathbf{C}(\mathbf{x}^k - \mathbf{y}^k) \rangle \\ &\stackrel{(b)}{\leq} \langle \mathbf{y}^k - \mathbf{x}^*, \mathbf{A}(\mathbf{y}^k - \mathbf{x}^k) \rangle + \langle \mathbf{y}^k - \mathbf{x}^*, \mathbf{A}(\mathbf{x}^* - \mathbf{y}^k) \rangle \\ &\stackrel{(c)}{=} \langle \mathbf{y}^k - \mathbf{x}^*, \mathbf{A}(\mathbf{x}^* - \mathbf{x}^k) \rangle \stackrel{(d)}{\leq} \frac{1}{4} \|\mathbf{y}^k - \mathbf{x}^k\|_{\mathbf{A}}^2 \end{aligned} \quad (28)$$

where step (a) uses the fact that $\mathbf{B} = -\mathbf{C} + \mathbf{A}$; step (b) Lemma 3 with $\mathbf{x} = \mathbf{x}^k$; step (c) uses the fact that $(\mathbf{x}^* - \mathbf{y}^k) + (\mathbf{y}^k - \mathbf{x}^k) = (\mathbf{x}^* - \mathbf{x}^k)$; step (d) uses Lemma 4. We then have the following results:

$$\begin{aligned} &\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^k - \mathbf{x}^*\|_2^2 \\ &\stackrel{(a)}{=} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2^2 + 2\langle \mathbf{x}^k - \mathbf{x}^*, \mathbf{x}^{k+1} - \mathbf{x}^k \rangle \\ &\stackrel{(b)}{=} \|\alpha^k \mathbf{B}(\mathbf{y}^k - \mathbf{x}^k)\|_2^2 + 2\alpha^k \langle \mathbf{x}^k - \mathbf{x}^*, \mathbf{B}(\mathbf{y}^k - \mathbf{x}^k) \rangle \\ &\stackrel{(c)}{=} \|\alpha^k \mathbf{B}(\mathbf{y}^k - \mathbf{x}^k)\|_2^2 + 2\alpha^k \langle \mathbf{x}^k - \mathbf{y}^k, \mathbf{B}(\mathbf{y}^k - \mathbf{x}^k) \rangle \\ &\quad + 2\alpha^k \langle \mathbf{y}^k - \mathbf{x}^*, \mathbf{B}(\mathbf{y}^k - \mathbf{x}^k) \rangle \\ &\stackrel{(d)}{\leq} \|\alpha^k \mathbf{B}(\mathbf{y}^k - \mathbf{x}^k)\|_2^2 + 2\alpha^k \langle \mathbf{x}^k - \mathbf{y}^k, \mathbf{B}(\mathbf{y}^k - \mathbf{x}^k) \rangle \\ &\quad + \frac{1}{2} \alpha^k \|\mathbf{y}^k - \mathbf{x}^k\|_{\mathbf{A}}^2 \stackrel{(e)}{=} \|\mathbf{y}^k - \mathbf{x}^k\|_{\mathbf{G}^k}^2 \end{aligned}$$

where step (a) uses Pythagoras relation that $\|\mathbf{x} - \mathbf{z}\|_2^2 - \|\mathbf{y} - \mathbf{z}\|_2^2 = \|\mathbf{x} - \mathbf{y}\|_2^2 + 2\langle \mathbf{y} - \mathbf{z}, \mathbf{x} - \mathbf{y} \rangle$, $\forall \mathbf{x}, \mathbf{y}, \mathbf{z}$; step (b) uses the update rule for \mathbf{x}^{k+1} ; step (c) uses the fact that $\mathbf{x}^k - \mathbf{x}^* = (\mathbf{x}^k - \mathbf{y}^k) + (\mathbf{y}^k - \mathbf{x}^*)$; step (d) uses (28); step (e) uses the definition of \mathbf{G}^k .

(ii) We have the following inequalities:

$$\mathbf{Q} \stackrel{(a)}{=} 2\mathbf{B} - \frac{1}{2}\mathbf{A} \stackrel{(b)}{\succeq} 2\mathbf{B} - \mathbf{A} \stackrel{(c)}{=} \mathbf{B} - \mathbf{C} \succeq \delta \mathbf{I} \quad (29)$$

where (a) uses the definition of \mathbf{Q} in (27); step (b) uses $\frac{1}{2}\mathbf{A} \preceq \mathbf{A}$; step (c) uses the fact that $\mathbf{A} = \mathbf{B} + \mathbf{C}$; step (d) uses the fact that $\forall \mathbf{z}$, $\mathbf{z}^T(\mathbf{B} - \mathbf{C})\mathbf{z} \geq \delta \|\mathbf{z}\|_2^2$, which is due to (13). Then we derive the following inequalities:

$$\mathbf{G}^k = \alpha^k (\frac{1}{2} \alpha^k \mathbf{P} - \mathbf{Q}) \stackrel{(a)}{\preceq} \alpha^k (\alpha^k \mathbf{B}^T \mathbf{B} - \delta \mathbf{I}) \stackrel{(b)}{\prec} \mathbf{0}$$

where step (a) uses (29); step (b) uses the choice that $0 < \alpha^k < \frac{\delta}{\|\mathbf{B}^T \mathbf{B}\|}$.

(iii) We define $\mathbf{v} \triangleq \mathbf{y}^k - \mathbf{x}^k$. Minimizing the right-hand side of (27) over the variable α , we obtain (30).

$$\alpha^* = \arg \min_{\alpha} \psi(\alpha) \triangleq \frac{1}{2} (\|\mathbf{v}\|_{\mathbf{P}}^2) \alpha^2 - (\|\mathbf{v}\|_{\mathbf{Q}}^2) \alpha. \quad (30)$$

Setting the gradient of quadratic function $\psi(\alpha)$ to zero, we obtain the optimal solution for $\alpha^* = \|\mathbf{v}\|_{\mathbf{Q}}^2 / \|\mathbf{v}\|_{\mathbf{P}}^2$ and it holds that $\psi(\alpha^*) = \frac{1}{2} \|\mathbf{v}\|_{\mathbf{P}}^2 \cdot \frac{\|\mathbf{v}\|_{\mathbf{Q}}^2}{\|\mathbf{v}\|_{\mathbf{P}}^2} - \|\mathbf{v}\|_{\mathbf{Q}}^2 \cdot \frac{\|\mathbf{v}\|_{\mathbf{Q}}^2}{\|\mathbf{v}\|_{\mathbf{P}}^2} = -\frac{1}{2} \frac{\|\mathbf{v}\|_{\mathbf{Q}}^4}{\|\mathbf{v}\|_{\mathbf{P}}^2}$. Finally, we have $\psi(\alpha^*) \leq -\frac{\delta^2 \|\mathbf{v}\|_2^2}{4\|\mathbf{B}^T \mathbf{B}\|}$. \square

Remarks. (i) The iterated solutions generated by Algorithm 3 satisfy the monotone/contractive property. Therefore, the convergence properties in Theorem 5 are stronger than the results in Theorem 2 and Theorem 3. (ii) There are two methods to decide the value of α^k in Algorithm 3. One method is to use the global

constant as indicated in part (ii) in Theorem 5, and another method is to use a local constant parameter as shown in part (iii) in Theorem 5. We remark that a local constant is more desirable in practice since it provides a better estimation of the local structure of the problem and does not require any sparse eigenvalue solver.

3.2 Acceleration via Richardson Extrapolation

This subsection discusses using Richardson extrapolation to further accelerate GMSA.

We introduce a parameter $\theta^k \in (0, \infty)$ and consider the following iterative procedure:

$$\begin{aligned} \mathbf{y}^k &= \mathcal{T}(\mathbf{x}^k) \\ \mathbf{x}^{k+1} &= \mathbf{x}^k + \theta^k(\mathbf{y}^k - \mathbf{x}^k). \end{aligned} \quad (31)$$

Note that the SOC update rule is not a special case of this formulation. Values of $0 < \theta^k < 1$ are often used to help establish convergence of the iterative procedure, while values of $\theta^k > 1$ are used to speed up convergence of a slow-converging procedure which is also known as Richardson extrapolation [43]. Such strategy is closely related to Nesterov's extrapolation acceleration strategy [3], [31].

The following proposition provides important insights on how to choose the parameter θ^k .

Proposition 1. *We define:*

$$\begin{aligned} \min_{\theta} \varphi(\theta) &\triangleq \|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^k - \mathbf{x}^*\|_2^2 \\ &= \|\mathbf{x}^k + \theta^k(\mathbf{y}^k - \mathbf{x}^k) - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^k - \mathbf{x}^*\|_2^2 \end{aligned} \quad (32)$$

The optimal solution of (32) can be computed as $\theta^* = \frac{\langle \mathbf{x}^k - \mathbf{x}^*, \mathbf{x}^k - \mathbf{y}^k \rangle}{\|\mathbf{x}^k - \mathbf{y}^k\|_2^2}$. In addition, if $\langle \mathbf{x}^k - \mathbf{x}^*, \mathbf{x}^k - \mathbf{y}^k \rangle \neq 0$, there exists a constant $0 < \nu < 1$ such that $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 \leq \nu^{k+1} \|\mathbf{x}^0 - \mathbf{x}^*\|_2^2$.

Proof. From (32), we have:

$$\begin{aligned} \varphi(\theta) &= \|\theta(\mathbf{y}^k - \mathbf{x}^k) + \mathbf{x}^k - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^k - \mathbf{x}^*\|_2^2 \\ &= \theta^2 \|\mathbf{y}^k - \mathbf{x}^k\|_2^2 + 2\theta \langle \mathbf{x}^k - \mathbf{x}^*, \mathbf{y}^k - \mathbf{x}^k \rangle \end{aligned} \quad (33)$$

Setting the gradient of the quadratic function $\varphi(\theta)$ to zero, we obtain: $\theta^* = \frac{\langle \mathbf{x}^k - \mathbf{x}^*, \mathbf{x}^k - \mathbf{y}^k \rangle}{\|\mathbf{x}^k - \mathbf{y}^k\|_2^2}$. Putting the optimal solution θ^* to (33), we obtain: $\varphi(\theta^*) = \frac{\langle \mathbf{x}^k - \mathbf{x}^*, \mathbf{x}^k - \mathbf{y}^k \rangle^2}{\|\mathbf{x}^k - \mathbf{y}^k\|_2^2} + 2 \frac{\langle \mathbf{x}^k - \mathbf{x}^*, \mathbf{x}^k - \mathbf{y}^k \rangle}{\|\mathbf{x}^k - \mathbf{y}^k\|_2^2} \langle \mathbf{x}^k - \mathbf{x}^*, \mathbf{y}^k - \mathbf{x}^k \rangle = -\frac{\langle \mathbf{x}^k - \mathbf{x}^*, \mathbf{x}^k - \mathbf{y}^k \rangle^2}{\|\mathbf{x}^k - \mathbf{y}^k\|_2^2}$.

Under the assumption that $\langle \mathbf{x}^k - \mathbf{x}^*, \mathbf{x}^k - \mathbf{y}^k \rangle \neq 0$, we have $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 - \|\mathbf{x}^k - \mathbf{x}^*\|_2^2 < 0$ for all k . There exists a constant such that $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 \leq \nu \|\mathbf{x}^k - \mathbf{x}^*\|_2^2$ for all k . Solving this recursive inequality, we obtain: $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_2^2 \leq \nu \|\mathbf{x}^k - \mathbf{x}^*\|_2^2 \leq \nu^2 \|\mathbf{x}^{k-1} - \mathbf{x}^*\|_2^2 \leq \dots \leq \nu^{k+1} \|\mathbf{x}^0 - \mathbf{x}^*\|_2^2$. \square

Remarks. (i) The assumption $\langle \mathbf{x}^k - \mathbf{x}^*, \mathbf{x}^k - \mathbf{y}^k \rangle \neq 0$ also implies that \mathbf{x}^k is not the optimal solution since it holds that $\mathbf{x}^k \neq \mathbf{x}^*$ and $\mathbf{x}^k \neq \mathbf{y}^k$ when \mathbf{x}^k is not the optimal solution. (ii) The step size selection strategy is attractive since it guarantees contractive property for $\|\mathbf{x}^k - \mathbf{x}^*\|_2^2$. However, it is not practical since the optimal solution \mathbf{x}^* is unknown.

In what follows, we consider the following solution. Since \mathbf{y}^k is the current available solution which is the closest to \mathbf{x}^* , we

replace \mathbf{x}^* with \mathbf{y}^k , and k with $k-1$. We obtain the follow update rule for θ^k :

$$\theta^k = \frac{\langle \mathbf{x}^{k-1} - \mathbf{y}^k, \mathbf{x}^{k-1} - \mathbf{y}^{k-1} \rangle}{\|\mathbf{x}^{k-1} - \mathbf{y}^{k-1}\|_2^2}.$$

We summarize our accelerated generalized matrix splitting algorithm in Algorithm 4. Note that we set $\theta^k = 1$ in the first iteration ($k = 0$) and introduce two additional parameters L and U to avoid θ^k to become arbitrarily small or large. Since the strategy in (31) is expected to achieve acceleration when $\theta^k > 1$, we set $L = 1$ and $U = 10$ as the default parameters for Algorithm 4.

Algorithm 4 GMSA-A: Generalized Matrix Splitting Algorithm with Richardson Extrapolation Acceleration for Solving (1).

- 1: Choose suitable parameters $\{\omega, \epsilon\}$. Initialize \mathbf{x}^0 .
- 2: for $k = 0, 1, 2, \dots$
- 3: $\mathbf{y}^k = \mathcal{T}(\mathbf{x}^k)$
- 4: if $k = 0$
- 5: $\theta^k = 1$
- 6: else
- 7: $\theta^k = \frac{\langle \mathbf{x}^{k-1} - \mathbf{y}^k, \mathbf{x}^{k-1} - \mathbf{y}^{k-1} \rangle}{\|\mathbf{x}^{k-1} - \mathbf{y}^{k-1}\|_2^2}$
- 8: $\theta^k = \min[U, \max(L, \theta^k)]$
- 9: end if
- 10: $\mathbf{x}^{k+1} = \mathbf{x}^k + \theta^k(\mathbf{y}^k - \mathbf{x}^k)$
- 11: end for
- 12: Output \mathbf{x}^{k+1}

3.3 When h is Nonconvex

When $h(\mathbf{x})$ is nonconvex, our theoretical analysis breaks down in (11) and the exact solution to the triangle proximal operator $\mathcal{T}(\mathbf{x}^k)$ in (6) cannot be guaranteed. However, our Gaussian elimination procedure in Algorithm 2 can still be applied. What one needs is to solve a one-dimensional nonconvex subproblem in (7). For example, when $h_j(t) = \lambda|t|_0, \forall j = 1, 2, \dots, n$ (e.g. in the case of the ℓ_0 norm), it has an analytical solution: $t^* = \begin{cases} -\mathbf{w}_j/\mathbf{B}_{j,j}, & \mathbf{w}_j^2 > 2\lambda\mathbf{B}_{j,j} \\ 0, & \mathbf{w}_j^2 \leq 2\lambda\mathbf{B}_{j,j} \end{cases}$; when $h_j(t) = \lambda|t|^p, \forall j = 1, 2, \dots, n$ and $p < 1$, it admits a closed form solution for some special values [7], [45], such as $p = \frac{1}{2}$ or $\frac{2}{3}$.

Our generalized matrix splitting algorithm is guaranteed to converge even when $h(\cdot)$ is nonconvex. Specifically, we present the following theorem.

Theorem 6. (Proof of Global Convergence when $h(\cdot)$ is Nonconvex) We define $\delta \triangleq \epsilon + \frac{1-\omega}{\omega} \min(\text{diag}(\mathbf{D}))$ and let $\{\omega, \epsilon\}$ be chosen such that $\delta \in (0, \infty)$. Assuming the nonconvex one-dimensional subproblem in (7) can be solved globally and analytically, we have: (i)

$$f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k) \leq -\frac{\delta}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2^2 \leq 0 \quad (34)$$

(ii) Algorithm 1 is globally convergent.

Proof. (i) Due to the optimality of the one-dimensional subproblem in (7), for all $j = 1, 2, \dots, n$, we have:

$$\begin{aligned} &\frac{1}{2} \mathbf{B}_{j,j} (\mathbf{x}_j^{k+1})^2 + (\mathbf{u}_j + \sum_{i=1}^{j-1} \mathbf{B}_{j,i} \mathbf{x}_i^{k+1}) \mathbf{x}_j^{k+1} + h(\mathbf{x}_j^{k+1}) \\ &\leq \frac{1}{2} \mathbf{B}_{j,j} t_j^2 + (\mathbf{u}_j + \sum_{i=1}^{j-1} \mathbf{B}_{j,i} \mathbf{x}_i^{k+1}) t_j + h(t_j), \forall t_j \end{aligned}$$

Letting $t_1 = \mathbf{x}_1^k, t_2 = \mathbf{x}_2^k, \dots, t_n = \mathbf{x}_n^k$, we obtain:

$$\begin{aligned} &\frac{1}{2} \sum_{i=1}^n \mathbf{B}_{i,i} (\mathbf{x}_i^{k+1})^2 + \langle \mathbf{u} + \mathbf{L} \mathbf{x}^{k+1}, \mathbf{x}^{k+1} \rangle + h(\mathbf{x}^{k+1}) \\ &\leq \frac{1}{2} \sum_{i=1}^n \mathbf{B}_{i,i} (\mathbf{x}_i^k)^2 + \langle \mathbf{u} + \mathbf{L} \mathbf{x}^{k+1}, \mathbf{x}^k \rangle + h(\mathbf{x}^k) \end{aligned}$$

Since $\mathbf{u} = \mathbf{b} + \mathbf{C}\mathbf{x}^k$, we obtain the following inequality:

$$\begin{aligned} f^{k+1} + \frac{1}{2}\langle \mathbf{x}^{k+1}, (\frac{1}{\omega}\mathbf{D} + \epsilon\mathbf{I} + 2\mathbf{L} - \mathbf{A})\mathbf{x}^{k+1} + 2\mathbf{C}\mathbf{x}^k \rangle \\ \leq f^k + \frac{1}{2}\langle \mathbf{x}^k, (\frac{1}{\omega}\mathbf{D} + \epsilon\mathbf{I} + 2\mathbf{C} - \mathbf{A})\mathbf{x}^k + 2\mathbf{L}\mathbf{x}^{k+1} \rangle \end{aligned}$$

By denoting $\mathbf{S} \triangleq \mathbf{L} - \mathbf{L}^T$ and $\mathbf{T} \triangleq \frac{\omega-1}{\omega}\mathbf{D} - \epsilon\mathbf{I}$, we have: $\frac{1}{\omega}\mathbf{D} + \epsilon\mathbf{I} + 2\mathbf{L} - \mathbf{A} = \mathbf{T} - \mathbf{S}$, $\frac{1}{\omega}\mathbf{D} + \epsilon\mathbf{I} + 2\mathbf{C} - \mathbf{A} = \mathbf{S} - \mathbf{T}$, and $\mathbf{L} - \mathbf{C}^T = -\mathbf{T}$. Therefore, we have the following inequalities:

$$\begin{aligned} f^{k+1} - f^k \\ \leq \frac{1}{2}\langle \mathbf{x}^{k+1}, \mathbf{T}\mathbf{x}^{k+1} \rangle + \frac{1}{2}\langle \mathbf{x}^k, \mathbf{T}\mathbf{x}^k \rangle - \langle \mathbf{x}^k, \mathbf{T}\mathbf{x}^{k+1} \rangle \\ - \frac{1}{2}\langle \mathbf{x}^{k+1}, \mathbf{S}\mathbf{x}^{k+1} \rangle - \frac{1}{2}\langle \mathbf{x}^k, \mathbf{S}\mathbf{x}^k \rangle \\ \stackrel{(a)}{=} \frac{1}{2}\langle \mathbf{x}^k - \mathbf{x}^{k+1}, \mathbf{T}(\mathbf{x}^k - \mathbf{x}^{k+1}) \rangle \stackrel{(b)}{\leq} -\frac{\delta}{2}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2^2 \end{aligned}$$

where step (a) uses $\langle \mathbf{x}, \mathbf{S}\mathbf{x} \rangle = 0 \forall \mathbf{x}$, since \mathbf{S} is a Skew-Hermitian matrix; step (b) uses $\mathbf{T} + \delta\mathbf{I} \preceq \mathbf{0}$, since $\mathbf{x} + \min(-\mathbf{x}) \leq \mathbf{0} \forall \mathbf{x}$. Thus, we obtain the sufficient decrease inequality in (34).

(ii) Based on the sufficient decrease inequality in (34), we have: $f(\mathbf{x}^k)$ is a non-increasing sequence, $\|\mathbf{x}^k - \mathbf{x}^{k+1}\| \rightarrow 0$, and $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$ if $\mathbf{x}^k \neq \mathbf{x}^{k+1}$. We note that (8) can be still applied even $h(\cdot)$ is nonconvex. Using the same methodology as in the second part of Theorem 1, we obtain that $\nabla q(\mathbf{x}^{k+1}) + \partial h(\mathbf{x}^{k+1}) \rightarrow \mathbf{0}$, which implies the convergence of the algorithm.

Note that guaranteeing $\delta \in (0, \infty)$ can be achieved by simply choosing $\omega \in (0, 1)$ and setting ϵ to a small number. \square

3.4 When q is not Quadratic

This subsection discusses how to adapt GMSA to solve (1) even when $q(\cdot)$ is not quadratic but it is convex and twice differentiable. Following previous work [42], [50], we keep the nonsmooth function $h(\cdot)$ and approximate the smooth function $q(\cdot)$ around the current solution \mathbf{x}^k by its second-order Taylor expansion:

$$\begin{aligned} \mathcal{Q}(\mathbf{y}, \mathbf{x}^k) \triangleq h(\mathbf{y}) + q(\mathbf{x}^k) + \langle \nabla q(\mathbf{x}^k), \mathbf{y} - \mathbf{x}^k \rangle + \\ \frac{1}{2}(\mathbf{y} - \mathbf{x}^k)^T \nabla^2 q(\mathbf{x}^k)(\mathbf{y} - \mathbf{x}^k) \end{aligned}$$

where $\nabla q(\mathbf{x}^k)$ and $\nabla^2 q(\mathbf{x}^k)$ denote the gradient and Hessian of $q(\mathbf{x})$ at \mathbf{x}^k , respectively. In order to generate the next solution that decreases the objective, one can minimize the quadratic model above by solving

$$\mathbf{y}^k = \arg \min_{\mathbf{y}} \mathcal{Q}(\mathbf{y}, \mathbf{x}^k). \quad (35)$$

And then one performs line search by the update: $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \alpha^k(\mathbf{x}^{k+1} - \mathbf{x}^k)$ for the greatest descent in objective (as in the damped Newton method). Here, $\alpha^k \in (0, 1]$ is the step-size selected by backtracking line search.

In practice, one does not need to solve the Newton approximation subproblem in (35) exactly and one iteration suffices for global convergence. We use $\mathbf{x}^k = \mathcal{T}(\mathbf{x}^k; \mathbf{A}^k, \mathbf{b}^k, h)$ to denote one iteration of GMSA with the parameter $\{\mathbf{A}^k, \mathbf{b}^k, h\}$. Note that both \mathbf{A}^k and \mathbf{b}^k are changing with k . We use $\{\mathbf{B}^k, \mathbf{C}^k, \mathbf{D}^k, \mathbf{L}^k\}$ to denote the associated matrices of \mathbf{A}^k using the same splitting strategy as in (4). In some situation, we drop the superscript k for simplicity since it can be known from the context. We summarized our algorithm to solve the general convex composite problem in Algorithm 5.

Algorithm 5 iteratively calls Algorithm 1 for one time to compute the next point \mathbf{y}^k . Based on the search direction $\mathbf{d}^k = \mathbf{y}^k - \mathbf{x}^k$, we employ Armijo's rule and try step size

Algorithm 5 Generalized Matrix Splitting Algorithm for Minimizing non-Quadratic Composite Functions in (1).

- 1: Choose suitable parameters $\{\omega, \epsilon\}$. Initialize \mathbf{x}^0 .
- 2: for $k = 0, 1, 2, \dots$
- 3: Define $\mathbf{A} = \nabla^2 q(\mathbf{x}^k)$, $\mathbf{b} = \nabla q(\mathbf{x}^k) - \nabla^2 q(\mathbf{x}^k)\mathbf{x}^k$
- 4: $\mathbf{y}^k = \mathcal{T}(\mathbf{x}^k; \mathbf{A}, \mathbf{b}, h)$
- 5: Define $\mathbf{d}^k \triangleq \mathbf{y}^k - \mathbf{x}^k$
- 6: Define $\Delta^k \triangleq \langle \nabla q(\mathbf{x}^k), \mathbf{d}^k \rangle + h(\mathbf{x}^k + \mathbf{d}^k) - h(\mathbf{x}^k)$
- 7: Find the largest $\alpha^k \in \{\eta^0, \eta^1, \dots\}$ such that

$$f(\mathbf{x}^k + \alpha^k \mathbf{d}^k) \leq f(\mathbf{x}^k) + \alpha^k \tau \Delta^k \quad (36)$$

- 8: $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^k$
- 9: end for
- 10: Output \mathbf{x}^{k+1}

$\alpha \in \{\eta^0, \eta^1, \dots\}$ with a constant decrease rate $0 < \eta < 1$ until we find the smallest $t \in \mathbb{N}$ with $\alpha = \eta^t$ such that $\mathbf{x}^k + \alpha^k \mathbf{d}^k$ satisfies the sufficient decrease condition. A typical choice for the parameters $\{\eta, \tau\}$ is $\{0.1, 0.25\}$.

In what follows, we present our convergence analysis for Algorithm 5. The following lemma is useful in our proof.

Lemma 5. Let Δ^k be defined in Line 6 of Algorithm 5. It holds that:

$$\Delta^k \leq -\langle \mathbf{d}^k, \mathbf{B}^k \mathbf{d}^k \rangle.$$

Proof. It is not hard to notice that GMSA reduces to the following inclusion problem:

$$\mathbf{0} \in \mathbf{A}^k \mathbf{y}^k + \nabla q(\mathbf{x}^k) - \mathbf{A}^k \mathbf{x}^k + \partial h(\mathbf{y}^k) + \mathbf{C}^k(\mathbf{x}^k - \mathbf{y}^k)$$

Using the definition that $\mathbf{d}^k \triangleq \mathbf{y}^k - \mathbf{x}^k$ and $\mathbf{A}^k = \mathbf{B}^k + \mathbf{C}^k$, we have:

$$\mathbf{0} \in \mathbf{B}^k \mathbf{d}^k + \nabla q(\mathbf{x}^k) + \partial h(\mathbf{y}^k) \quad (37)$$

We further derive the following inequalities:

$$\begin{aligned} h(\mathbf{x}^k + \mathbf{d}^k) - h(\mathbf{x}^k) &\stackrel{(a)}{=} h(\mathbf{y}^k) - h(\mathbf{x}^k) \\ &\stackrel{(b)}{\leq} \langle \mathbf{y}^k - \mathbf{x}^k, \mathbf{o} \rangle, \forall \mathbf{o} \in \partial h(\mathbf{y}^k) \\ &\stackrel{(c)}{=} \langle \mathbf{d}^k, -\mathbf{B}^k \mathbf{d}^k - \nabla q(\mathbf{x}^k) \rangle \end{aligned}$$

where step (a) uses the fact that $\mathbf{x}^k + \mathbf{d}^k = \mathbf{y}^k$; step (b) uses the convexity of $h(\cdot)$; step (c) uses (37) and $\mathbf{y}^k - \mathbf{x}^k = \mathbf{d}^k$. Rearranging terms we finish the proof of this lemma. \square

Theorem 7. We define $\delta^k \triangleq (1/\omega - 1/2) \min(\text{diag}(\mathbf{D}^k)) + \epsilon$ and let $\{\omega, \epsilon\}$ be chosen such that $\delta^k \in (0, \infty)$. Assuming that the gradient of $q(\cdot)$ is L -Lipschitz continuous, we have the following results: (i) There always exists a strictly positive constant α^k such that the descent condition in (36) is satisfied. (ii) The sequence $\{f(\mathbf{x}^k)\}_{k=0}^\infty$ is nonincreasing and Algorithm 5 is globally convergent.

Proof. For simplicity, we drop the iteration counter k as it can be inferred from the context. First of all, for all $\mathbf{v} \in \mathbb{R}^n$, we have:

$$\begin{aligned} \mathbf{v}^T \mathbf{B} \mathbf{v} &\stackrel{(a)}{=} \mathbf{v}^T (\frac{1}{2}\mathbf{L} + \frac{1}{2}\mathbf{L}^T + \frac{1}{2}\mathbf{D} + (\frac{1}{\omega} - \frac{1}{2})\mathbf{D} + \epsilon\mathbf{I}) \mathbf{v} \\ &\stackrel{(b)}{=} \frac{1}{2}\mathbf{v}^T \mathbf{A} \mathbf{v} + \mathbf{v}^T [(\frac{1}{\omega} - \frac{1}{2})\mathbf{D} + \epsilon\mathbf{I}] \mathbf{v} \\ &\stackrel{(c)}{\geq} 0 + \delta^k \|\mathbf{v}\|_2^2 \end{aligned} \quad (38)$$

where step (a) uses the definition of \mathbf{B} that $\mathbf{B} = \mathbf{L} + \frac{1}{\omega}\mathbf{D} + \epsilon\mathbf{I}$ and the fact that $\mathbf{v}^T\mathbf{L}\mathbf{v} = \mathbf{v}^T\mathbf{L}^T\mathbf{v}$; step (b) uses $\mathbf{A} = \mathbf{L} + \mathbf{L}^T + \mathbf{D}$; step (c) uses the fact that \mathbf{A} is positive semidefinite for convex problems.

For any $\alpha \in (0, 1]$, we have the following results:

$$\begin{aligned}
& f(\mathbf{x} + \alpha\mathbf{d}) - f(\mathbf{x}) \\
&= q(\mathbf{x} + \alpha\mathbf{d}) - q(\mathbf{x}) + h(\mathbf{x} + \alpha\mathbf{d}) - h(\mathbf{x}) \\
&\stackrel{(a)}{\leq} q(\mathbf{x} + \alpha\mathbf{d}) - q(\mathbf{x}) + \alpha[h(\mathbf{x} + \mathbf{d}) - h(\mathbf{x})] \\
&\stackrel{(b)}{\leq} \frac{\alpha^2 L}{2} \|\mathbf{d}\|_2^2 + \alpha[\langle \mathbf{d}, \nabla q(\mathbf{x}) \rangle + h(\mathbf{x} + \mathbf{d}) - h(\mathbf{x})] \\
&\stackrel{(c)}{=} \frac{\alpha^2 L}{2} \|\mathbf{d}\|_2^2 + \alpha\Delta^k \stackrel{(d)}{\leq} -\frac{\alpha^2 L}{2} \frac{\Delta^k}{\delta^k} + \alpha\Delta^k \\
&\stackrel{(e)}{=} (1 - \frac{\alpha L}{2\delta^k})\alpha\Delta^k \leq \tau^k \alpha\Delta^k
\end{aligned}$$

where step (a) uses the fact that $h(\mathbf{x} + \alpha\mathbf{d}) = h(\alpha(\mathbf{x} + \mathbf{d}) + (1 - \alpha)\mathbf{x}) \leq \alpha h(\mathbf{x} + \mathbf{d}) + (1 - \alpha)h(\mathbf{x})$ which is due to the convexity of $h(\cdot)$; step (b) uses the inequality $q(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla q(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L}{2} \|\mathbf{y} - \mathbf{x}\|_2^2$ with $\mathbf{y} = \mathbf{x} + \alpha\mathbf{d}$; step (c) uses the definition of Δ^k in Algorithm 5; step (d) uses Lemma (5) that $\Delta^k \leq -\langle \mathbf{d}^k, \mathbf{B}^k \mathbf{d}^k \rangle \leq -\delta^k \|\mathbf{d}^k\|_2^2$ which is due to (38); step (e) uses the choice that $0 < \alpha < \min[1, 2\delta^k(1 - \tau^k)/L]$.

(ii) We obtain the following inequality:

$$\forall k, f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k) \leq -\tau^k \alpha^k \|\mathbf{d}^k\|_2^2 \text{ with } \tau^k \alpha^k > 0,$$

and the sequence $f(\mathbf{x}^k)$ is non-increasing. Using the same methodology as in the second part of Theorem 1, we have $\mathbf{d}^k \rightarrow 0$ as $k \rightarrow \infty$. Therefore, any cluster point of the sequence \mathbf{x}^k is a stationary point. Finally, we have $\mathbf{d}^k = \mathbf{0}$. From (8) in Lemma (1), we obtain: $\mathbf{0} = -\mathbf{C}(\mathbf{x}^k - \mathcal{T}(\mathbf{x}^k)) \in \nabla q(\mathcal{T}(\mathbf{x}^k)) + \partial h(\mathcal{T}(\mathbf{x}^k))$. Therefore, we conclude that $\mathcal{T}(\mathbf{x}^k) = \mathbf{y}^k = \mathbf{x}^k$ is the global optimal solution. \square

3.5 Adaptation into ADMM Optimization Framework

This subsection shows how to adapt GMSA into the general optimization framework of ADMM [12], [13] to solve the following structured convex problem:

$$\min_{\mathbf{x}, \mathbf{y}} \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{x}^T \mathbf{p} + h(\mathbf{x}) + r(\mathbf{y}), \quad \mathbf{E} \mathbf{x} + \mathbf{z} = \mathbf{y} \quad (39)$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ are decision variables, and $\mathbf{Q} \in \mathbb{R}^{n \times n}$, $\mathbf{p} \in \mathbb{R}^n$, $\mathbf{E} \in \mathbb{R}^{m \times n}$, $\mathbf{z} \in \mathbb{R}^m$ are given. We assume that \mathbf{Q} is positive semidefinite and $r(\cdot)$ is simple and convex (may not necessarily be separable) such that its proximal operator $\text{prox}_r(\mathbf{a}) = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{a}\|_2^2 + r(\mathbf{x})$ can be evaluated analytically. We let $\mathcal{L} : \mathbb{R}^n, \mathbb{R}^m, \mathbb{R}^m \mapsto \mathbb{R}$ be the augmented Lagrangian function of (39):

$$\begin{aligned}
\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\pi}) &= \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{x}^T \mathbf{p} + h(\mathbf{x}) + r(\mathbf{y}) \\
&+ \langle \mathbf{E} \mathbf{x} + \mathbf{z} - \mathbf{y}, \boldsymbol{\pi} \rangle + \frac{\beta}{2} \|\mathbf{E} \mathbf{x} - \mathbf{z} + \mathbf{y}\|_2^2
\end{aligned}$$

where $\boldsymbol{\pi} \in \mathbb{R}^m$ is the multiplier associated with the equality constraint $\mathbf{E} \mathbf{x} + \mathbf{z} = \mathbf{y}$, and $\beta > 0$ is the penalty parameter.

We summarize our algorithm for solving (39) in Algorithm 6. The algorithm optimizes for a set of primal variables at a time and keeps the other primal and dual variables fixed, with the dual variables updating via gradient ascent. For the \mathbf{y} -subproblem, it admits a closed-form solution. For the \mathbf{x} -subproblem, since the smooth part is quadratic and the nonsmooth part is separable, it

Algorithm 6 GMSA-ADMM: Generalized Matrix Splitting Algorithm-based Alternating Direction Method of Multipliers for Solving (39).

- 1: Choose $\omega \in (0, 2)$, $\epsilon \in [0, \infty)$. Initialize \mathbf{x}^0 .
- 2: for $k = 0, 1, 2, \dots$
- 3: Use Algorithm 1 to solve the following problem:
 $\mathbf{x}^{k+1} = \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}^k, \boldsymbol{\pi}^k)$
- 4: $\mathbf{y}^{k+1} = \min_{\mathbf{y}} \mathcal{L}(\mathbf{x}^{k+1}, \mathbf{y}, \boldsymbol{\pi}^k)$
- 5: $\boldsymbol{\pi}^{k+1} = \boldsymbol{\pi}^k + \beta(\mathbf{E} \mathbf{x}^{k+1} + \mathbf{z} - \mathbf{y}^{k+1})$
- 6: end for
- 7: Output \mathbf{x}^{k+1}

can be solved by Algorithm 1. Algorithm 6 is convergent if the \mathbf{x} -subproblem is solved exactly since it reduces to classical ADMM [12]. We remark that similar to linearized ADMM [12], Algorithm 6 also suffices for convergence empirically even if we only solve the \mathbf{x} -subproblem approximately.

3.6 When \mathbf{x} is a Matrix

In many applications (e.g. nonnegative matrix factorization and sparse coding), the solutions exist in the matrix form as follows:

$$\min_{\mathbf{X} \in \mathbb{R}^{n \times r}} \frac{1}{2} \text{tr}(\mathbf{X}^T \mathbf{A} \mathbf{X}) + \text{tr}(\mathbf{X}^T \mathbf{R}) + h(\mathbf{X}),$$

where $\mathbf{R} \in \mathbb{R}^{n \times r}$. Our matrix splitting algorithm can still be applied in this case. Using the same technique to decompose \mathbf{A} as in (4): $\mathbf{A} = \mathbf{B} + \mathbf{C}$, one needs to replace (6) to solve the following nonlinear equation: $\mathbf{B} \mathbf{Z}^* + \mathbf{U} + \partial h(\mathbf{Z}^*) \in 0$, where $\mathbf{U} = \mathbf{R} + \mathbf{C} \mathbf{X}^k$. It can be decomposed into r independent components. By updating every column of \mathbf{X} , the proposed algorithm can be used to solve the matrix problem above. Thus, our algorithm can also make good use of existing parallel architectures to solve the matrix optimization problem.

4 EXPERIMENTS

This section demonstrates the efficiency and efficacy of the proposed Generalized Matrix Splitting Algorithm (GMSA) by considering three important applications: nonnegative matrix factorization (NMF) [20], [22], ℓ_0 norm regularized sparse coding [21], [32], [35], and ℓ_1 norm regularized Danzig selectors. We implement our method in MATLAB on an Intel 2.6 GHz CPU with 8 GB RAM. Only our generalized Gaussian elimination procedure is developed in C and wrapped into the MATLAB code, since it requires an elementwise loop that is quite inefficient in native MATLAB. We consider $(\epsilon, \omega) = (0.01, 1)$ as our default parameters for GMSA in all our experiments. Some Matlab code can be found in the authors' research webpages.

4.1 Convergence Behavior of Different Methods

This subsection demonstrates the convergence behavior of different methods for solving random least squares problems with different nonsmooth composite functions, as follows:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{C} \mathbf{x} - \mathbf{d}\|_2^2, \text{ s.t. } \mathbf{x} \geq \mathbf{0} \quad (40)$$

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{C} \mathbf{x} - \mathbf{d}\|_2^2 + \|\mathbf{x}\|_1 \quad (41)$$

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{C} \mathbf{x} - \mathbf{d}\|_2^2 + 0.1 \|\mathbf{x}\|_0 \quad (42)$$

These optimization problems are referred as convex nonnegative least squares problems, convex ℓ_1 norm regularized least squares

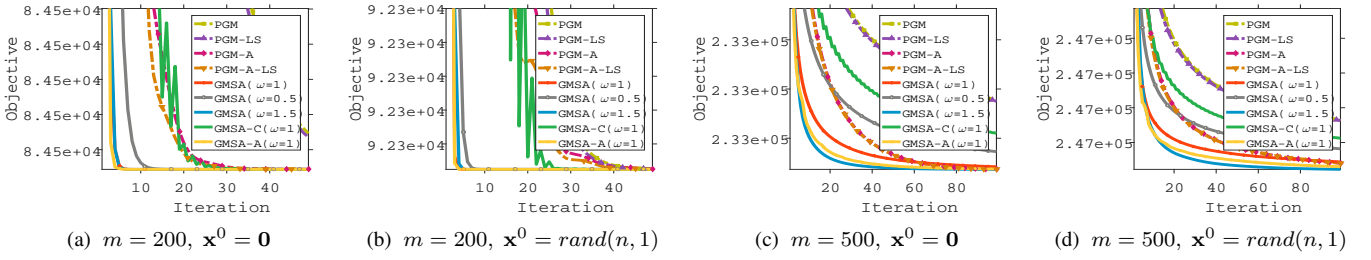


Figure 1: Convergence behavior for solving convex nonnegative least squares problem as in (40).

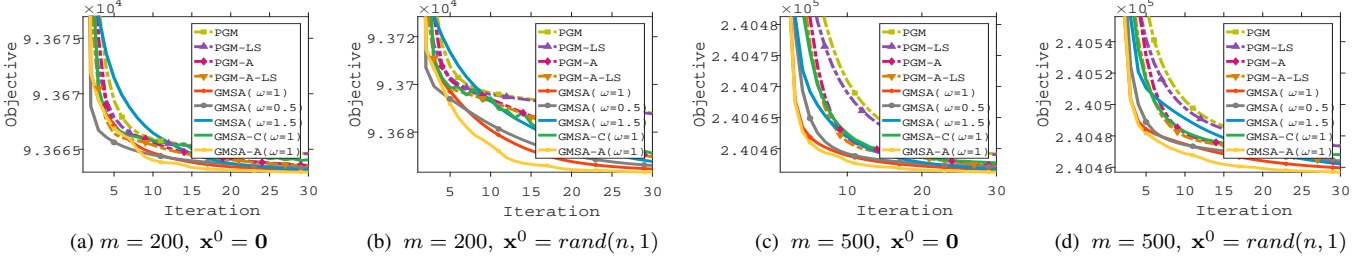


Figure 2: Convergence behavior for solving convex ℓ_1 norm regularized least squares problem as in (41).

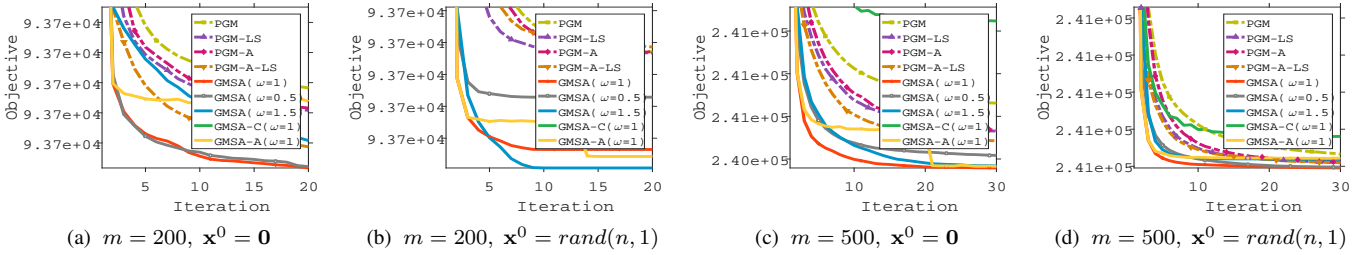


Figure 3: Convergence behavior for solving nonconvex ℓ_0 norm regularized least squares problem as in (42).

problems, and nonconvex ℓ_0 norm regularized least squares problems, respectively. Here, $\mathbf{C} \in \mathbb{R}^{m \times n}$ and $\mathbf{d} \in \mathbb{R}^m$ are generated from a standard Gaussian distribution. We fix $n = 1000$ and vary $m = \{200, 500\}$.

We compare the following methods. (i) PGM: classical proximal gradient method with constant step size [31]; (ii) PGM-LS: classical PGM with line search [3]; (iii) PGM-A: accelerated PGM with constant step size [31]; (iv) PGM-A-LS: accelerated PGM with line search [3], [31]; (v) GMSA ($\omega = 1/0.5/1.5$): generalized matrix splitting algorithm with varying the parameter ω described in Algorithm 1; (vi) GMSA-C: generalized matrix splitting algorithm with correction step described in (3), where a local constant for computing the step size α^k is used; (vii) GMSA-A: generalized matrix splitting algorithm with Richardson extrapolation acceleration described in (4).

Figure 1-3 shows the objective values of the comparing methods for each iteration when they are applied to solve Problem (40-42) using different initializations. Note that we use $\text{rand}(n, 1)$ to denote a function that returns a random vector sampled from a (0-1) uniform distribution and all the methods have the same computational complexity for one iteration. We have the following observations. (i) GMSA with the default parameters $(\omega, \theta) = (1, 0.01)$ significantly outperforms proximal gradient method and its variants. (ii) GMSA($\omega = 1.5$) gives better

performance than GMSA ($\omega = 0.5$) for solving nonnegative least squares problem but it gives worse performance than GMSA ($\omega = 0.5$) for solving ℓ_1 norm regularized least squares problem. The choice of the parameter ω seems to be sensitive to the specific data. (iii) GMSA-C converges slower than GMSA but faster than {PGM, PGM-LS}. (iv) GMSA-A generally outperforms the other methods in the convex problems. (v) GMSA generally presents the best performance in the nonconvex problems.

We notice that (i) GMSA with the choice $(\epsilon, \omega) = (0.01, 1)$ gives comparable performance to its variants and (ii) GMSA-A is not necessarily a monotonic algorithm (although it achieves acceleration upon GMSA). Therefore, in our following experiments we only report the results for GMSA with $(\epsilon, \omega) = (0.01, 1)$.

4.2 Nonnegative Matrix Factorization

Nonnegative matrix factorization [20] is a very useful tool for feature extraction and identification in the fields of text mining and image understanding. It is formulated as the following optimization problem:

$$\min_{\mathbf{W} \in \mathbb{R}^{m \times n}, \mathbf{H} \in \mathbb{R}^{n \times d}} \frac{1}{2} \|\mathbf{Y} - \mathbf{WH}\|_F^2, \text{ s.t. } \mathbf{W} \geq 0, \mathbf{H} \geq 0.$$

Following previous work [11], [16], [19], [22], we alternatively minimize the objective while keeping one of the two variables

Table 1: Comparisons of objective values for nonnegative matrix factorization for all the compared methods. The 1st, 2nd, and 3rd best results are colored with red, blue and green, respectively.

time limit $T = 20$								time limit $T = 30$							
data	n	[22] PG	[19] AS	[19] BPP	[11] APG	[16] CGD	[ours] GMSA	data	n	[22] PG	[19] AS	[19] BPP	[11] APG	[16] CGD	[ours] GMSA
20news	20	5.001e+06	2.762e+07	8.415e+06	4.528e+06	4.515e+06	4.506e+06	20news	20	4.716e+06	2.762e+07	7.471e+06	4.510e+06	4.503e+06	4.500e+06
20news	50	5.059e+06	2.762e+07	4.230e+07	3.775e+06	3.544e+06	3.467e+06	20news	50	4.569e+06	2.762e+07	5.034e+07	3.628e+06	3.495e+06	3.446e+06
20news	100	6.955e+06	5.779e+06	4.453e+07	3.658e+06	3.971e+06	2.902e+06	20news	100	6.639e+06	2.762e+07	4.316e+07	3.293e+06	3.223e+06	2.817e+06
20news	200	7.675e+06	3.036e+06	1.023e+08	4.431e+06	3.573e+07	2.819e+06	20news	200	6.991e+06	2.762e+07	1.015e+08	3.609e+06	7.676e+06	2.507e+06
20news	300	1.997e+07	2.762e+07	1.956e+08	4.519e+06	4.621e+07	3.202e+06	20news	300	1.354e+07	2.762e+07	1.942e+08	4.519e+06	4.621e+07	3.097e+06
COIL	20	2.004e+09	5.480e+09	2.031e+09	1.974e+09	1.976e+09	1.975e+09	COIL	20	1.992e+09	4.405e+09	2.014e+09	1.974e+09	1.975e+09	1.975e+09
COIL	50	1.412e+09	1.516e+10	6.962e+09	1.291e+09	1.256e+09	1.252e+09	COIL	50	1.335e+09	2.420e+10	5.772e+09	1.272e+09	1.252e+09	1.250e+09
COIL	100	2.960e+09	2.834e+10	3.222e+10	9.919e+08	8.745e+08	8.510e+08	COIL	100	2.936e+09	2.834e+10	1.814e+10	9.422e+08	8.623e+08	8.458e+08
COIL	200	3.371e+09	2.834e+10	5.229e+10	8.495e+08	5.959e+08	5.600e+08	COIL	200	3.362e+09	2.834e+10	4.627e+10	7.614e+08	5.720e+08	5.392e+08
COIL	300	3.996e+09	2.834e+10	1.017e+11	8.493e+08	5.002e+08	4.956e+08	COIL	300	3.946e+09	2.834e+10	7.417e+10	6.734e+08	4.609e+08	4.544e+08
TDT2	20	1.597e+06	2.211e+06	1.688e+06	1.591e+06	1.595e+06	1.592e+06	TDT2	20	1.595e+06	2.211e+06	1.667e+06	1.591e+06	1.594e+06	1.592e+06
TDT2	50	1.408e+06	2.211e+06	2.895e+06	1.393e+06	1.390e+06	1.385e+06	TDT2	50	1.397e+06	2.211e+06	2.285e+06	1.393e+06	1.389e+06	1.385e+06
TDT2	100	1.300e+06	2.211e+06	6.187e+06	1.222e+06	1.224e+06	1.214e+06	TDT2	100	1.241e+06	2.211e+06	5.702e+06	1.216e+06	1.219e+06	1.212e+06
TDT2	200	1.628e+06	2.211e+06	1.791e+07	1.119e+06	1.227e+06	1.079e+06	TDT2	200	1.484e+06	1.878e+06	1.753e+07	1.063e+06	1.104e+06	1.049e+06
TDT2	300	1.915e+06	1.854e+06	3.412e+07	1.172e+06	7.902e+06	1.066e+06	TDT2	300	1.879e+06	2.211e+06	3.398e+07	1.060e+06	1.669e+06	1.007e+06
time limit $T = 40$								time limit $T = 50$							
data	n	[22] PG	[19] AS	[19] BPP	[11] APG	[16] CGD	[ours] GMSA	data	n	[22] PG	[19] AS	[19] BPP	[11] APG	[16] CGD	[ours] GMSA
20news	20	4.622e+06	2.762e+07	7.547e+06	4.495e+06	4.500e+06	4.496e+06	20news	20	4.565e+06	2.762e+07	6.939e+06	4.488e+06	4.498e+06	4.494e+06
20news	50	4.386e+06	2.762e+07	1.562e+07	3.564e+06	3.478e+06	3.438e+06	20news	50	4.343e+06	2.762e+07	1.813e+07	3.525e+06	3.469e+06	3.432e+06
20news	100	6.486e+06	2.762e+07	4.223e+07	3.128e+06	2.988e+06	2.783e+06	20news	100	6.404e+06	2.762e+07	3.955e+07	3.046e+06	2.878e+06	2.765e+06
20news	200	6.731e+06	1.934e+07	1.003e+08	3.304e+06	5.744e+06	2.407e+06	20news	200	5.939e+06	2.762e+07	9.925e+07	3.121e+06	4.538e+06	2.359e+06
20news	300	1.041e+07	2.762e+07	1.932e+08	3.621e+06	4.621e+07	2.543e+06	20news	300	9.258e+06	2.762e+07	1.912e+08	3.621e+06	2.323e+07	2.331e+06
COIL	20	1.987e+09	5.141e+09	2.010e+09	1.974e+09	1.975e+09	1.975e+09	COIL	20	1.982e+09	7.136e+09	7.136e+09	1.974e+09	1.975e+09	1.975e+09
COIL	50	1.308e+09	2.403e+10	5.032e+09	1.262e+09	1.250e+09	1.248e+09	COIL	50	1.298e+09	2.834e+10	4.365e+09	1.258e+09	1.248e+09	1.248e+09
COIL	100	2.922e+09	2.834e+10	2.086e+10	9.161e+08	8.555e+08	8.430e+08	COIL	100	1.945e+09	2.834e+10	1.428e+10	9.014e+08	8.516e+08	8.414e+08
COIL	200	3.361e+09	2.834e+10	4.116e+10	7.075e+08	5.584e+08	5.289e+08	COIL	200	3.362e+09	2.834e+10	3.760e+10	6.771e+08	5.491e+08	5.231e+08
COIL	300	3.920e+09	2.834e+10	7.040e+10	6.221e+08	4.384e+08	4.294e+08	COIL	300	3.905e+09	2.834e+10	6.741e+10	5.805e+08	4.226e+08	4.127e+08
TDT2	20	1.595e+06	2.211e+06	1.643e+06	1.591e+06	1.594e+06	1.592e+06	TDT2	20	1.595e+06	2.211e+06	1.622e+06	1.591e+06	1.594e+06	1.592e+06
TDT2	50	1.394e+06	2.211e+06	1.933e+06	1.392e+06	1.388e+06	1.384e+06	TDT2	50	1.393e+06	2.211e+06	1.875e+06	1.392e+06	1.386e+06	1.384e+06
TDT2	100	1.229e+06	2.211e+06	5.259e+06	1.213e+06	1.216e+06	1.211e+06	TDT2	100	1.223e+06	2.211e+06	4.831e+06	1.212e+06	1.214e+06	1.210e+06
TDT2	200	1.389e+06	1.547e+06	1.716e+07	1.046e+06	1.070e+06	1.041e+06	TDT2	200	1.267e+06	2.211e+06	1.671e+07	1.040e+06	1.054e+06	1.036e+06
TDT2	300	1.949e+06	1.836e+06	3.369e+07	1.008e+06	1.155e+06	9.776e+05	TDT2	300	1.903e+06	2.211e+06	3.328e+07	9.775e+05	1.045e+06	9.606e+05

fixed. In each alternating subproblem, we solve a convex nonnegative least squares problem, where our GMSA is used. We conduct experiments on three datasets ⁴ 20news, COIL, and TDT2. The size of the datasets are 18774×61188 , 7200×1024 , 9394×36771 , respectively. We compare GMSA against the following state-of-the-art methods: (1) Projective Gradient (PG) [5], [22] that updates the current solution via steep gradient descent and then maps a point back to the bounded feasible region ⁵; (2) Active Set (AS) method [19] and (3) Block Principal Pivoting (BPP) method [19] ⁶ that iteratively identify an active and passive set by a principal pivoting procedure and solve a reduced linear system; (4) Accelerated Proximal Gradient (APG) [11] ⁷ that applies Nesterov's momentum strategy with a constant step size to solve the convex sub-problems; (5) Coordinate Gradient Descent (CGD) [16] ⁸ that greedily selects one coordinate by measuring the objective reduction and optimizes for a single variable via closed-form update. Similar to our method, the core procedure of CGD is developed in C and wrapped into the MATLAB code, while all other methods are implemented using builtin MATLAB functions.

We use the same settings as in [22]. We compare objective values after running T seconds with T varying from 20 to 50. Table 1 presents average results of using 10 random initial points, which are generated from a standard normal distribution. While the other methods may quickly lower objective values when the resulting dimension n is small ($n = 20$), GMSA catches up very quickly and achieves a faster convergence speed when n is large. It generally achieves the best performance in terms of objective

value among all the methods.

4.3 ℓ_0 Norm Regularized Sparse Coding

Sparse coding is a popular unsupervised feature learning technique for data representation that is widely used in computer vision and medical imaging. Motivated by recent success in ℓ_0 norm modeling [2], [46], [48], we consider the following ℓ_0 norm regularized (*i.e.* cardinality) sparse coding problem:

$$\min_{\mathbf{W}, \mathbf{H}} \frac{1}{2} \|\mathbf{Y} - \mathbf{WH}\|_F^2 + \lambda \|\mathbf{H}\|_0, \text{ s.t. } \|\mathbf{W}(:, i)\| = 1, \forall i, \quad (43)$$

with $\mathbf{W} \in \mathbb{R}^{m \times n}$ and $\mathbf{H} \in \mathbb{R}^{n \times d}$. Existing solutions for this problem are mostly based on the family of proximal point methods [2], [31]. We compare GMSA with the following methods: (1) Proximal Gradient Method (PGM) with constant step size, (2) PGM with line search, (3) accelerated PGM with constant step size, and (4) accelerated PGM with line search.

We evaluate all the methods for the application of image denoising. Following [1], [2], we set the dimension of the dictionary to $n = 256$. The dictionary is learned from $m = 1000$ image patches randomly chosen from the noisy input image. The patch size is 8×8 , leading to $d = 64$. The experiments are conducted on 16 conventional test images with different noise standard deviations σ . We tune the regularization parameter λ and compare the resulting objective values and the Signal-to-Noise Ratio (SNR) values for all methods. We also compare the SNR values for different methods. Due to space limitation, we include the experimental results in the Appendix.

We compare the objective values for all methods by *fixing* the variable \mathbf{W} to an over-complete DCT dictionary [1] and *only* optimizing over \mathbf{H} . We compare all methods with varying regularization parameter λ and different initial points that are either generated by random Gaussian sampling or the Orthogonal

4. <http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html>

5. <https://www.csie.ntu.edu.tw/~cjlin/libmf/>

6. <http://www.cc.gatech.edu/~hpar/nmfsolver.php>

7. <https://sites.google.com/site/nmfsolvers/>

8. <http://www.cs.utexas.edu/~cjhshieh/nmf/>

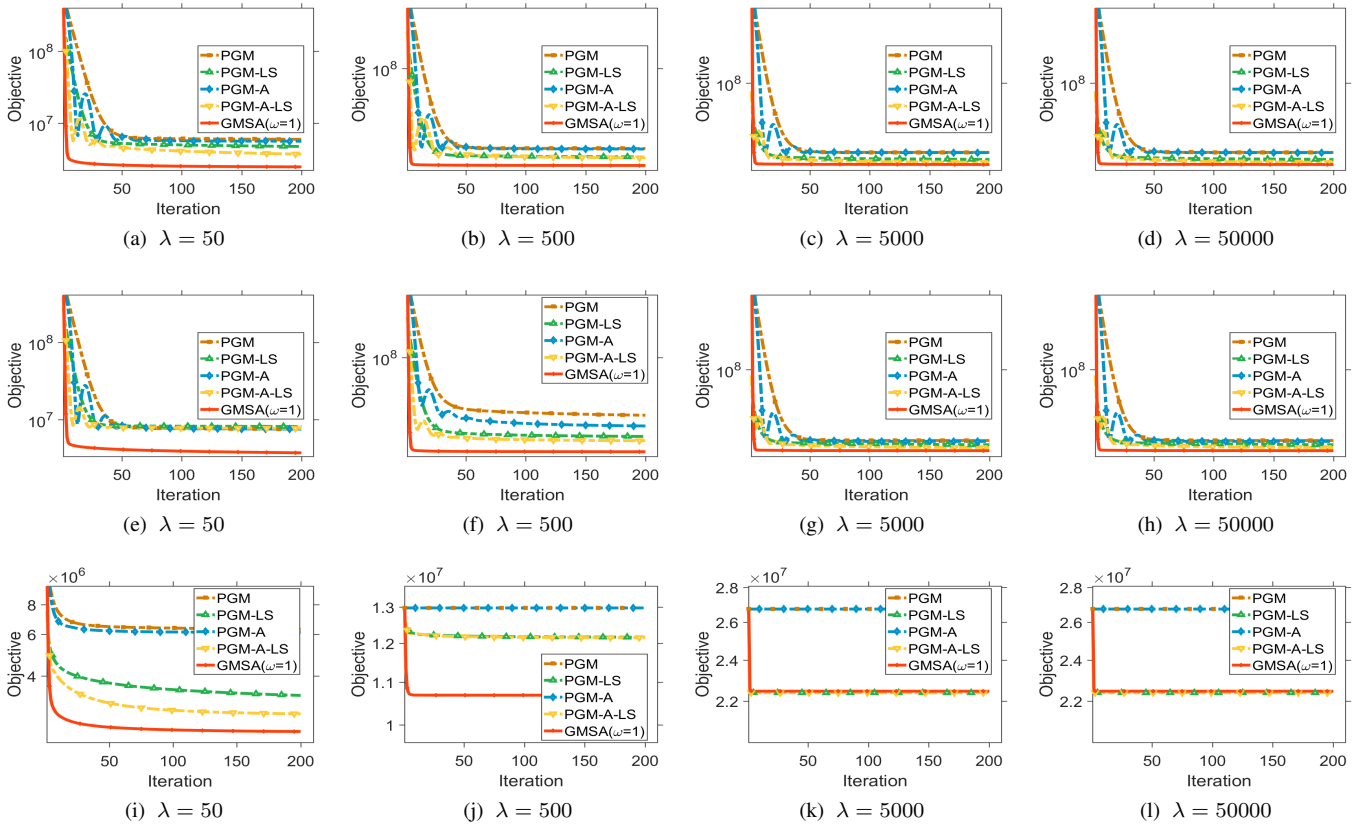


Figure 4: Convergence behavior for solving (43) with fixing \mathbf{W} for different λ and initializations. Denoting $\tilde{\mathbf{O}}$ as an arbitrary standard Gaussian random matrix of suitable size, we consider the following three initializations for \mathbf{H} . First row: $\mathbf{H} = 0.1 \times \tilde{\mathbf{O}}$. Second row: $\mathbf{H} = 10 \times \tilde{\mathbf{O}}$. Third row: \mathbf{H} is set to the output of the orthogonal matching pursuit.

Matching Pursuit (OMP) method [40]. In Figure 4, we observe that GMSA converges rapidly in 10 iterations. Moreover, it often generates much better local optimal solutions than the compared methods.

4.4 ℓ_1 Norm Regularized Danzig Selectors

Danzig selectors [6] can be formulated as the following optimization problem: $\min_{\mathbf{x}} \|\mathbf{x}\|_1$, s.t. $\|\mathbf{D}^{-1}\mathbf{W}^T(\mathbf{W}\mathbf{x} - \mathbf{y})\|_\infty \leq \delta$, where $\mathbf{W} \in \mathbb{R}^{m \times n}$, $\mathbf{y} \in \mathbb{R}^m$, $\delta \in \mathbb{R} > 0$, and $\mathbf{D} \in \mathbb{R}^{n \times n}$ is the diagonal matrix whose diagonal entries are the norm of the columns of \mathbf{W} . For the ease of discussion, we consider the following equivalent unconstrained optimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{x}\|_1 + \lambda \|\mathbf{Q}\mathbf{x} - \mathbf{s}\|_\infty \quad (44)$$

with $\mathbf{Q} = \mathbf{D}^{-1}\mathbf{W}^T\mathbf{W} \in \mathbb{R}^{n \times n}$, $\mathbf{s} = \mathbf{D}^{-1}\mathbf{W}^T\mathbf{y} \in \mathbb{R}^n$, and $\lambda \propto \frac{1}{\delta} \in \mathbb{R}$.

We generate the design matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$ via sampling from a standard Gaussian distribution. The sparse original signal $\tilde{\mathbf{x}} \in \mathbb{R}^{n \times 1}$ is generated via selecting a support set of size 20 uniformly at random and set them to arbitrary number sampled from standard Gaussian distribution. We set $\mathbf{y} = \mathbf{W}\tilde{\mathbf{x}}$. We fix $n = 1000$ and consider different choices for λ and m .

We compare the proposed method GMSA-ADMM against linearized ADMM algorithm and classical ADMM algorithm. The penalty parameter β is fixed to a constant with $\beta = 1$. For linearized ADMM, we use the same splitting strategy as in [44]. For classical ADMM, we introduce addition two variables and rewrite

(44) as: $\min_{\mathbf{x}, \mathbf{y}, \mathbf{z}} \|\mathbf{z}\|_\infty + \lambda \|\mathbf{x}\|_1$, s.t. $\mathbf{x} = \mathbf{y}$, $\mathbf{A}\mathbf{y} - \mathbf{b} = \mathbf{z}$ to make sure that the smooth subproblem of the resulting augmented Lagrangian function is quadratic and can be solved by linear equations. For GMSA-ADMM, we do not solve the \mathbf{x} -subproblem exactly using GMSA but solve it using one GMSA iteration. Note that all methods have the some memory complexity to store the matrix $\mathbf{Q}^{n \times n}$ and computational complexity of $\mathcal{O}(n^2)$ to compute the proximal / triangle proximal operator. We uses the same random initial points and demonstrate the objective values for the comparing methods. It can be been in Figure 5 that our GMSA-ADMM significantly outperforms linearized ADMM and classical ADMM.

5 CONCLUSIONS

This paper presents a new generalized matrix splitting algorithm for minimizing composite functions. We rigorously analyze its convergence behavior for convex problems and discuss its several importance extensions. Experimental results on nonnegative matrix factorization, ℓ_0 norm regularized sparse coding, and ℓ_1 norm regularized Danzig selector demonstrate that our methods achieve state-of-the-art performance.

ACKNOWLEDGMENTS

This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research. This work was also supported by the NSF-China (61772570, 61472456, 61522115, 61628212).

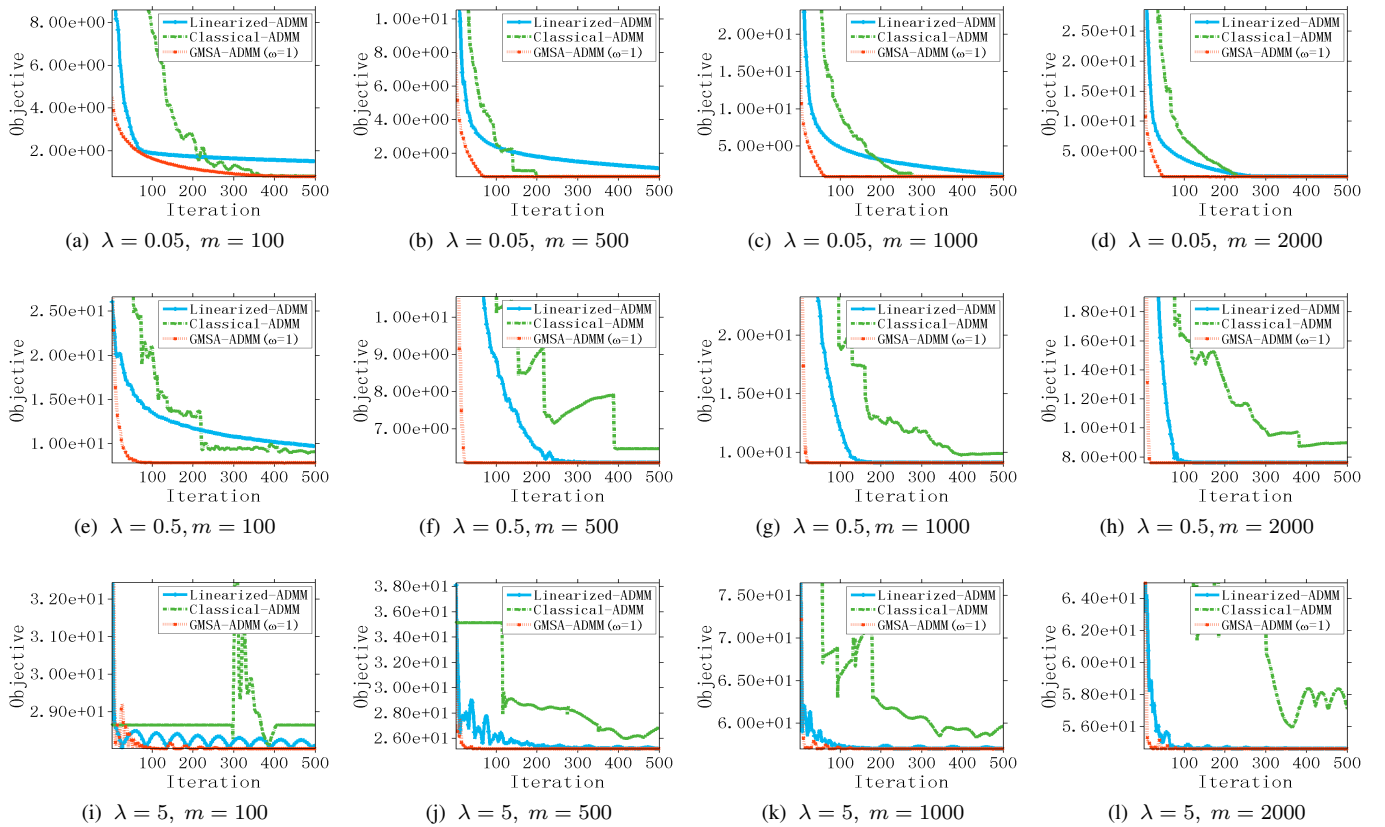


Figure 5: A comparison of linearized ADMM, classical ADMM, and GMSA-ADMM for solving the ℓ_1 regularized Danzig selectors problem.

REFERENCES

- [1] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.
- [2] Chenglong Bao, Hui Ji, Yuhui Quan, and Zuowei Shen. Dictionary learning for sparse coding: Algorithms and convergence analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(7):1356–1369, 2016.
- [3] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences (SIIMS)*, 2(1):183–202, 2009.
- [4] Amir Beck and Luba Tretushvili. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization (SIOPT)*, 23(4):2037–2060, 2013.
- [5] Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.
- [6] Emmanuel Candes, Terence Tao, et al. The dantzig selector: Statistical estimation when p is much larger than n . *The Annals of Statistics*, 35(6):2313–2351, 2007.
- [7] Wenfei Cao, Jian Sun, and Zongben Xu. Fast image deconvolution using closed-form thresholding formulas of regularization. *Journal of Visual Communication and Image Representation*, 24(1):31–41, 2013.
- [8] James W Demmel. *Applied numerical linear algebra*. SIAM, 1997.
- [9] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [10] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(11):2765–2781, 2013.
- [11] Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan. Nnmf: an optimal gradient method for nonnegative matrix factorization. *IEEE Transactions on Signal Processing*, 60(6):2882–2898, 2012.
- [12] Bingsheng He and Xiaoming Yuan. On the $\mathcal{O}(1/n)$ convergence rate of the douglas-rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.
- [13] Bingsheng He and Xiaoming Yuan. On non-ergodic convergence rate of douglas-rachford alternating direction method of multipliers. *Numerische Mathematik*, 130(3):567–577, 2015.
- [14] Mingyi Hong, Xiangfeng Wang, Meisam Razaviyayn, and Zhi-Quan Luo. Iteration complexity analysis of block coordinate descent methods. *Mathematical Programming*, pages 1–30, 2013.
- [15] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathya Keerthi, and Sellamanickam Sundararajan. A dual coordinate descent method for large-scale linear svm. In *International Conference on Machine Learning (ICML)*, pages 408–415, 2008.
- [16] Cho-Jui Hsieh and Inderjit S Dhillon. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1064–1072, 2011.
- [17] Alfredo N. Iusem. On the convergence of iterative methods for symmetric linear complementarity problems. *Math. Program.*, 59(1):33–48, March 1993.
- [18] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 315–323, 2013.
- [19] Jingu Kim and Haesun Park. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing (SISC)*, 33(6):3261–3281, 2011.
- [20] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.
- [21] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, pages 801–808, 2006.
- [22] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10):2756–2779, 2007.
- [23] Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization (SIOPT)*, 25(4):2244–2273, 2015.
- [24] Ji Liu and Stephen J Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization (SIOPT)*, 25(1):351–376, 2015.
- [25] Zhaosong Lu and Lin Xiao. Randomized block coordinate non-monotone gradient method for a class of nonlinear programming. *arXiv preprint*, 2013.

- [26] Zhaosong Lu and Lin Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, 152(1-2):615–642, 2015.
- [27] Zhi-Quan Luo and Paul Tseng. On the convergence of a matrix splitting algorithm for the symmetric monotone linear complementarity problem. *SIAM Journal on Control and Optimization*, 29(5):1037–1060, 1991.
- [28] Zhi-Quan Luo and Paul Tseng. Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem. *SIAM Journal on Optimization*, 2(1):43–54, 1992.
- [29] Zhi-Quan Luo and Paul Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178, 1993.
- [30] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization (SIOPT)*, 22(2):341–362, 2012.
- [31] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.
- [32] Bruno A Olshausen et al. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.
- [33] Neal Parikh, Stephen P Boyd, et al. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.
- [34] Andrei Patrascu and Ion Necoara. Iteration complexity analysis of random coordinate descent methods for ℓ_0 regularized convex problems. *arXiv preprint*, 2014.
- [35] Yuhui Quan, Yong Xu, Yuping Sun, Yan Huang, and Hui Ji. Sparse coding for classification via discrimination ensemble. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [36] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.
- [37] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.
- [38] Li Shen, Wei Liu, Ganzhao Yuan, and Shiqian Ma. GSOS: gauss-seidel operator splitting algorithm for multi-term nonsmooth convex composite optimization. In *International Conference on Machine Learning (ICML)*, pages 3125–3134, 2017.
- [39] Ruoyu Sun and Mingyi Hong. Improved iteration complexity bounds of cyclic block coordinate descent for convex problems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1306–1314, 2015.
- [40] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.
- [41] Paul Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Mathematical Programming*, 125(2):263–295, 2010.
- [42] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, 2009.
- [43] Richard S. Varga and J. Gillis. *Matrix Iterative Analysis*. Prentice-Hall, 1962.
- [44] Xiangfeng Wang and Xiaoming Yuan. The linearized alternating direction method of multipliers for dantzig selector. *SIAM Journal on Scientific Computing*, 34(5), 2012.
- [45] Zongben Xu, Xiangyu Chang, Fengmin Xu, and Hai Zhang. L1/2 regularization: A thresholding representation theory and a fast solver. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 23(7):1013–1027, 2012.
- [46] Yingzhen Yang, Jiashi Feng, Nebojsa Jojic, Jianchao Yang, and Thomas S. Huang. ℓ^0 -sparse subspace clustering. *European Conference on Computer Vision (ECCV)*, 2016.
- [47] Hsiang-Fu Yu, Fang-Lan Huang, and Chih-Jen Lin. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85(1-2):41–75, 2011.
- [48] Ganzhao Yuan and Bernard Ghanem. ℓ_{0tv} : A new method for image restoration in the presence of impulse noise. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5369–5377, 2015.
- [49] Ganzhao Yuan, Wei-Shi Zheng, and Bernard Ghanem. A matrix splitting method for composite function minimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5310–5319, 2017.
- [50] Xiao-Tong Yuan and Qingshan Liu. Newton greedy pursuit: A quadratic approximation method for sparsity-constrained optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4122–4129, 2014.
- [51] Sangwoon Yun, Paul Tseng, and Kim-Chuan Toh. A block coordinate gradient descent method for regularized convex separable optimization and covariance selection. *Mathematical Programming*, 129(2):331–355, 2011.
- [52] Jinshan Zeng, Zhimin Peng, and Shaobo Lin. GAITA: A gauss-seidel iterative thresholding algorithm for ℓ_q regularized least squares regression. *Journal of Computational and Applied Mathematics*, 319:220–235, 2017.
- [53] Lei-Hong Zhang and Wei Hong Yang. An efficient matrix splitting method for the second-order cone complementarity problem. *SIAM Journal on Optimization*, 24(3):1178–1205, 2014.



ACM TODS.



recipient of Royal Society-Newton Advanced Fellowship.



nition/detection in video), especially at large-scale.

Ganzhao Yuan was born in Guangdong, China. He received his Ph.D. in School of Computer Science and Engineering, South China University of Technology in 2013. He is currently a research associate professor at School of Data and Computer Science in Sun Yat-sen University. His research interests primarily center around large-scale mathematical optimization and its applications in computer vision and machine learning. He has published technical papers in ICML, SIGKDD, AAAI, CVPR, VLDB, IEEE TPAMI, and

Wei-Shi Zheng was born in Guangdong, China. He is now a Professor at Sun Yat-sen University. He has now published more than 90 papers, including more than 60 publications in main journals (TPAMI, TIP, TNNLS, PR) and top conferences (ICCV, CVPR, IJCAI). His research interests include person/object association and activity understanding in visual surveillance. He has joined Microsoft Research Asia Young Faculty Visiting Programme. He is a recipient of Excellent Young Scientists Fund of the NSFC, and a

Li Shen was born in Hubei, China. He received his Ph.D. in School of Mathematics, South China University of Technology in 2017. He is currently a research scientist at Tencent AI Lab, Shenzhen. His research interests include algorithms for nonsmooth optimization, and their applications in statistical machine learning, game theory and reinforcement learning. He has published several papers in ICML and AAAI.

Bernard Ghanem was born in Betroumine, Lebanon. He received his Ph.D. in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign (UIUC) in 2010. He is currently an associate professor at King Abdullah University of Science and Technology (KAUST), where he leads the Image and Video Understanding Lab (IVUL). His research interests focus on designing, implementing, and analyzing approaches to address computer vision problems (e.g. object tracking and action recognition/detection in video), especially at large-scale.