Journal of Computational Mathematics
Vol.xx, No.x, 2023, 1–22.
http://www.global-sci.org/jcm
doi:10.4208/jcm.2305-m2021-0107

ACCELERATED SYMMETRIC ADMM AND ITS APPLICATIONS IN LARGE-SCALE SIGNAL PROCESSING*

Jianchao Bai
Research & Development Institute of Northwestern Polytechnical University in Shenzhen, Shenzhen 518057, China
School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710129, China
Email: jianchaobai@nwpu.edu.cn
Ke Guo[1)]
Key Laboratory of Optimization Theory and Applications at China West Normal University of Sichuan Province, School of Mathematics and Information, China West Normal University, Nanchong 637009, China
Email: keguo2014@126.com
Junli Liang
School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China
Email: liangjunli@nwpu.edu.cn
Yang Jing
School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China
Email: jingyang@mail.nwpu.edu.cn
H.C. So
Department of Electrical Engineering, City University of Hong Kong, Hong Kong, SAR, China
Email: hcso@ee.cityu.edu.hk

Abstract

The alternating direction method of multipliers (ADMM) has been extensively investigated in the past decades for solving separable convex optimization problems, and surprisingly, it also performs efficiently for nonconvex programs. In this paper, we propose a symmetric ADMM based on acceleration techniques for a family of potentially nonsmooth and nonconvex programming problems with equality constraints, where the dual variables are updated twice with different stepsizes. Under proper assumptions instead of the so-called Kurdyka-Lojasiewicz inequality, convergence of the proposed algorithm as well as its pointwise iteration-complexity are analyzed in terms of the corresponding augmented Lagrangian function and the primal-dual residuals, respectively. Performance of our algorithm is verified by numerical examples corresponding to signal processing applications in sparse nonconvex/convex regularized minimization.

Mathematics subject classification: 47A30, 65Y20, 90C26, 90C90.
Key words: Nonconvex optimization, Symmetric ADMM, Acceleration technique, Complexity, Signal processing.

1. Introduction

We consider a potentially nonsmooth and nonconvex separable optimization problem subject to linear equality constraints

* Received April 16, 2021 / Revised version received August 11, 2022 / Accepted May 26, 2023 / Published online November 23, 2023 /
1) Corresponding author

$$\begin{aligned} &\min\{f(\mathbf{x})+g(\mathbf{y})\} \\ &\text{s.t.} \quad A\mathbf{x}+B\mathbf{y}=b, \quad \mathbf{x}\in\mathcal{R}^m, \quad \mathbf{y}\in\mathcal{R}^n, \end{aligned} \tag{1.1}$$

where $f:\mathcal{R}^m\to(-\infty,+\infty]$ is a proper lower semicontinuous function, $g:\mathcal{R}^n\to(-\infty,+\infty)$ is a continuously differentiable function with its gradient ∇g being L_g-Lipschitz continuous, $A\in\mathcal{R}^{l\times m}, B\in\mathcal{R}^{l\times n}, b\in\mathcal{R}^l$ are given matrices and vector, respectively. Minimization problems of the form (1.1) appear in various applications in science and engineering. For example, the following l_1-regularized problem arises in signal processing and statistical learning [4, 5, 28]:

$$\min_{x\in\mathcal{R}^m} \frac{1}{2}\|A\mathbf{x}-c\|^2+\mu\|\mathbf{x}\|_1, \tag{1.2}$$

where $c\in\mathcal{R}^l$ is the observation vector, $A\in\mathcal{R}^{l\times m}$ is the data matrix and $\mu>0$ denotes the regularization parameter and is often set as $\mu=0.1\mu_{\max}$, where $\mu_{\max}=\|A^{\mathsf{T}}c\|_\infty$ (see e.g. [12, 28]). Due to the convexity of the problem (1.2), it can be handled by a number of standard methods, to list a few, including the alternating direction method of multipliers (ADMM, [10, 13, 14]), proximal point algorithm [4, 10], interior point method [28] and primal-dual hybrid gradient method [2, 45]. However, in many cases the l_1-regularization has been shown to be sub-optimal. For instance, it cannot recover a signal with the fewest measurements when applied in compressed sensing [7]. Therefore, an acceptable improvement is to adopt the $l_{1/2}$-regularization term, which results in the following problem:

$$\min_{x\in\mathcal{R}^m} \frac{1}{2}\|A\mathbf{x}-c\|^2+\mu\|\mathbf{x}\|_{\frac{1}{2}}^{\frac{1}{2}}, \tag{1.3}$$

where,

$$\|\mathbf{x}\|_{\frac{1}{2}}=\left(\sum_{i=1}^n|\mathbf{x}_i|^{\frac{1}{2}}\right)^2$$

is a nonconvex function characterizing the sparsity, and it has been verified [42] practically to be better than l_1-norm. Clearly, by introducing an auxiliary variable, the above problem (1.3) can be converted to a special case of (1.1), i.e.

$$\begin{aligned} &\min\left\{\mu\|\mathbf{x}\|_{\frac{1}{2}}^{\frac{1}{2}}+\frac{1}{2}\|\mathbf{y}-c\|^2\right\} \\ &\text{s.t.} \quad A\mathbf{x}-\mathbf{y}=\mathbf{0}. \end{aligned} \tag{1.4}$$

The bold $\mathbf{0}$ denotes zero vector or matrix with proper dimensions. Another interesting example is the regularized empirical risk minimization arising from big data applications, such as many kinds of classification and regression models in machine learning [37, 41]. The $l_{1/2}$-regularized reformulation is of the form

$$\begin{aligned} &\min\left\{\mu\|\mathbf{x}\|_{\frac{1}{2}}^{\frac{1}{2}}+\frac{1}{N}\sum_{j=1}^N g_j(\mathbf{y})\right\} \\ &\text{s.t.} \quad \mathbf{x}-\mathbf{y}=\mathbf{0}, \end{aligned} \tag{1.5}$$

where N is a large number, $g_j(\mathbf{y})=\log(1+\exp(-b_i a_i^{\mathsf{T}}\mathbf{y}))$ denotes the logistic loss function on the feature-label pair (a_j,b_j) with $a_j\in\mathcal{R}^l$ and $b_j\in\{-1,1\}$.

In the literature, the most standard approach for solving the equality constrained problem (1.1) is the augmented Lagrangian method (ALM) which firstly solves a joint minimization problem

$$\min_{\mathbf{x},\mathbf{y}} \mathcal{L}_\beta(\mathbf{x},\mathbf{y},\lambda) := f(\mathbf{x}) + g(\mathbf{y}) - \langle \lambda, A\mathbf{x} + B\mathbf{y} - b\rangle + \frac{\beta}{2}\|A\mathbf{x} + B\mathbf{y} - b\|^2, \tag{1.6}$$

and then updates the Lagrange multiplier λ based on the newest iteration of other variables. The penalty factor $\beta > 0$, in each iterative loop, can be set as a tuned reasonable value or updated adaptively according to the ratio of the primal residual to the dual residual of the problem. However, ALM does not make full use of the separable structure of the objective function of (1.1) and hence, could not take advantage of the special properties of each component objective function. This would make it very expensive even infeasible for problems involving big data and nonconvex objectives. By contrast, a powerful first-order method, that is ADMM, aims to split the joint core problem (1.6) into relatively simple and lower-dimensional subproblems so that variables can be updated separately to make full use of special properties of each component. Another feature of ADMM is that the resultant subproblems could admit closed-form solution in special applications, or in a linearized update for the differentiable objective/quadratic penalty term. We refer the interested readers to, e.g. [4, 5, 11, 15, 21, 23, 24, 40] for reviews on ADMM.

Interestingly, under the existence assumption of a solution to the Karush-Kuhn-Tucker condition of the two-block separable convex optimization problem, it was explained [13] that the original ADMM amounts to the Douglas-Rachford splitting method (DRSM, [9,30]) when it is applied to a stationary system to the dual of the problem. Moreover, as elaborated in [13], if applying the classic Peaceman-Rachford splitting method (PRSM, [30,35]) to the dual problem, we obtain the following iterative scheme:

$$\begin{cases} \mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \mathcal{L}_\beta(\mathbf{x},\mathbf{y}_k,\lambda_k), \\ \lambda_{k+\frac{1}{2}} = \lambda_k - \beta(A\mathbf{x}_{k+1} + B\mathbf{y}_k - b), \\ \mathbf{y}_{k+1} = \arg\min_{\mathbf{y}} \mathcal{L}_\beta\big(\mathbf{x}_{k+1},\mathbf{y},\lambda_{k+\frac{1}{2}}\big), \\ \lambda_{k+1} = \lambda_{k+\frac{1}{2}} - \beta(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b). \end{cases} \tag{1.7}$$

Unfortunately, the scheme (1.7) is not convergent under the standard convexity assumptions as ADMM [8]. However, it was verified [16] that (1.7) could perform faster than ADMM when its global convergence is ensured. In view of this, He *et al.* [22] proposed and studied the convergence of a strictly contractive Peaceman-Rachford splitting method (also called the symmetric version of ADMM)

$$\begin{cases} \mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \mathcal{L}_\beta(\mathbf{x},\mathbf{y}_k,\lambda_k), \\ \lambda_{k+\frac{1}{2}} = \lambda_k - \alpha\beta(A\mathbf{x}_{k+1} + B\mathbf{y}_k - b), \\ \mathbf{y}_{k+1} = \arg\min_{\mathbf{y}} \mathcal{L}_\beta\big(\mathbf{x}_{k+1},\mathbf{y},\lambda_{k+\frac{1}{2}}\big), \\ \lambda_{k+1} = \lambda_{k+\frac{1}{2}} - \alpha\beta(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b), \end{cases} \tag{1.8}$$

where $\alpha \in (0, 1)$ is the relaxation parameter. Later, He *et al.* [23] improved the scheme (1.8) to the case with larger range of relaxation parameters, which was generalized by Bai *et al.* [3] to the multi-block separable convex programming. Besides, Chang *et al.* [6] also suggested a generalization of linearized ADMM for two-block separable convex minimization model by adding a proper proximal term to each core subproblem.

Convergence analysis of ADMM (or its variant) for the nonconvex case is much more challenging. However, for some special nonconvex problems, one can establish convergence of

ADMM by making full use of special structures of the problems, e.g. see [25] for the consensus and sharing problems. Another widely used technique to prove convergence of ADMM for nonconvex optimization problems relies on the assumption that the objective function of (1.1) satisfies the so-called Kurdyka-Lojasiewicz (KL) inequality [1], which aligns with many important classes of functions [18–20, 29, 40, 41, 43]. The recent progress on ADMM for solving two-block and multi-block nonconvex optimization problems can be found, in e.g. [27, 38, 44], and the convergence of ADMM-type methods [38, 44] also depends on the KL property. Without assuming the KL property and convexity of the objective function, recently, Goncalves *et al.* [17] established convergence rate bounds of the classical ADMM with proximal terms for solving nonconvex linearly constrained optimization problem (1.1). In addition, by linearizing the smooth part in the objective and quadratic penalty term, Liu *et al.* [31] proposed a two-block linearized ADMM for the problem (1.1) with $b = \mathbf{0}$ and extended the method to a multi-block version, but convergence of their extended method holds with an extra hypothesis on the full column rank of the matrix B compared to (A1) (see Section 3).

Motivated by the above mentioned works [17, 31] and the empirical validity of the symmetric ADMM, we present a two-stage accelerated symmetric ADMM (abbreviated as TAS-ADM) for solving the problem (1.1), whose framework is provided in Algorithm 1.1. Our algorithm combines both the so-called Nesterov's acceleration technique explained in (3.24) and the relaxation scheme, in e.g. [10, 11]. By adding a proper proximal term for the first $\mathbf{x}$-subproblem, this possibly nonsmooth and nonconvex subproblem will turn to a proximity operator shown in (3.7), which admits closed-form solution if f is well defined. Step 7 actually uses the idea of convex combination for fast convergence. Although we consider problem (1.1) with vector variables, the subsequent convergence results are applicable for the general case with matrix variables.

Algorithm 1.1: TAS-ADM for Solving Problem (1.1).

1 Initialize $(\mathbf{x}_0, \mathbf{y}_0, \lambda_0) \in \mathcal{R}^m \times \mathcal{R}^n \times \mathcal{R}^l$ and set $(\mathbf{x}_{-1}, \mathbf{y}_{-1}) = (\mathbf{x}_0, \mathbf{y}_0)$.

2 Choose parameters $\beta > 0, \gamma_k \in [0, 1/2)$, $G \succeq \mathbf{0}$ and

$$(\tau, \alpha) \in \mathcal{D} := \{(\tau, \alpha) \mid 0 < \tau + \alpha < 1\}. \tag{1.9}$$

3 **for** $k = 0, 1, \ldots,$ **do**

4 $\quad \mathbf{x}_k^{md} = \mathbf{x}_k + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1})$.

5 $\quad \mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \left\{ \mathcal{L}_\beta(\mathbf{x}, \mathbf{y}_k, \lambda_k) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_k^{md}\|_G^2 \right\}$.

6 $\quad \lambda_{k+\frac{1}{2}} = \lambda_k - \tau\beta\left(A\mathbf{x}_{k+1} + B\mathbf{y}_k - b\right)$.

7 $\quad \mathbf{x}_{k+1}^{ad} = \alpha A\mathbf{x}_{k+1} + (1-\alpha)(b - B\mathbf{y}_k)$.

8 $\quad \mathbf{y}_{k+1} = \arg\min_{\mathbf{y}} \left\{ g(\mathbf{y}) - \left\langle \lambda_{k+\frac{1}{2}}, B\mathbf{y} \right\rangle + \frac{\beta}{2}\left\|\mathbf{x}_{k+1}^{ad} + B\mathbf{y} - b\right\|^2 \right\}$.

9 $\quad \lambda_{k+1} = \lambda_{k+\frac{1}{2}} - \beta\left(\mathbf{x}_{k+1}^{ad} + B\mathbf{y}_{k+1} - b\right)$.

10 **end**

11 Output $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})$.

Remark 1.1. Notice that by forcing $\gamma_k = 0$ and $\tau = 0$, Algorithm 1.1 would reduce to the linearized version of generalized ADMM [11], but the method in [11] focuses only on the convex case of (1.1). Compared to the proximal symmetric ADMM [22] (that is the scheme (1.8)), two extra acceleration steps are employed in Algorithm 1.1 for solving the general nonconvex problem (1.1). Recently, a symmetric ADMM was considered in [41] for solving the nonconvex problem (1.1) and it can be treated as Algorithm 1.1 without the acceleration techniques and proximal term. Moreover, convergence theories of these two algorithms are established under different conditions: the so-called KL property was used in [41], but not used for Algorithm 1.1.

The remaining parts of this paper are organized as follows. In Section 2, preliminaries for convergence analysis of Algorithm 1.1 are presented. In Section 3, we show its convergence properties and its pointwise iteration complexity based on the analysis of the augmented Lagrangian sequence $\{L_\beta(w_k)\}$, where $w_k := (\mathbf{x}_k, \mathbf{y}_k, \lambda_k)$. Section 4 presents numerical examples in solving the popular sparse signal recovery problem with different regularization terms, and comparisons with the popular CVX toolbox and well-established methods are included. Finally, we conclude the paper in Section 5.

2. Preliminaries

Throughout this paper, $\mathcal{R}, \mathcal{R}^n$ and $\mathcal{R}^{m\times n}$ represent the sets of real numbers, n-dimensional real column vectors and $m \times n$ real matrices, respectively. The symbol I denotes the identity matrix with proper dimensions and σ_B denotes the smallest positive eigenvalue of the matrix BB^T. For any symmetric matrices A and B whose dimensions are the same, $A \succ B$ ($A \succeq B$) means $A - B$ is a positive definite (semidefinite) matrix. We also denote $\|x\|_G^2 = x^\mathsf{T} G x$ for any symmetric matrix G, and $\|x\|_G = \sqrt{x^\mathsf{T} G x}$, if $G \succeq \mathbf{0}$. We simply use $\|\cdot\|$ to represent the standard Euclidean norm equipped with inner product $\langle \cdot\,, \cdot\rangle$. The image space of $A \in \mathcal{R}^{m\times n}$ is defined as $\mathrm{Im}(A) := \{Az \,|\, z \in \mathcal{R}^n\}$ and a function $f : \mathcal{S} \to \mathcal{R}$ is lower semicontinuous at $\bar{x} \in \mathcal{S}$ if and only if $\lim_{x\to\bar{x}} \inf f(x) = f(\bar{x})$. The distance from any point z to the set $\mathcal{S} \subseteq \mathcal{R}^n$ is defined as $\mathrm{dist}(z, \mathcal{S}) := \inf\{\|z - y\| \,|\, y \in \mathcal{S}\}$.

Definition 2.1 ([33, 36]). *Let $f : \mathcal{R}^m \to \mathcal{R}$ be a proper lower semicontinuous function.*

(a) For a given $x \in \mathrm{dom}(f)$, the Frechet subdifferential of f at x, written as $\widehat{\partial} f(x)$, is the set of all vectors $s \in \mathcal{R}^m$, which satisfies

$$\lim_{y\to x} \inf_{y\neq x} \frac{f(y) - f(x) - \langle s, y - x\rangle}{\|y - x\|} \geq 0,$$

and we let $\widehat{\partial} f(x) = \emptyset$, when $x \notin \mathrm{dom}(f)$.

(b) The limiting subdifferential or the subdifferential of f at $x \in \mathcal{R}^m$, written as $\partial f(x)$, is defined as

$$\partial f(x) = \big\{ s \in \mathcal{R}^m \,|\, \exists x_k \to x,\ f(x_k) \to f(x),\ \widehat{\partial} f(x_k) \ni s_k \to s \text{ as } k \to \infty \big\}.$$

(c) A point x_* is called critical point or stationary point of $f(x)$ if $\mathbf{0} \in \partial f(x_*)$.

Definition 2.2. *A triple $w_* := (\mathbf{x}_*, \mathbf{y}_*, \lambda_*) \in \mathcal{R}^m \times \mathcal{R}^n \times \mathcal{R}^l$ is a stationary point of (1.1) if*

$$A^\mathsf{T}\lambda_* \in \partial f(\mathbf{x}_*), \quad B^\mathsf{T}\lambda_* = \nabla g(\mathbf{y}_*), \quad A\mathbf{x}_* + B\mathbf{y}_* - b = \mathbf{0}.$$

The following lemmas are provided to simplify convergence analysis in the sequel sections.

Lemma 2.1 ([17, Lemma A.2]). *Let $A \in \mathcal{R}^{m\times n}$ be a nonzero matrix and $\mathcal{P}_A$ be the Euclidean projection onto* $\mathrm{Im}(A)$. *Then, for any $u \in \mathcal{R}^n$ we have*

$$\|\mathcal{P}_A(u)\| \le \frac{1}{\sqrt{\sigma_A}}\|A^{\mathsf{T}}u\|.$$

Lemma 2.2. *For any vectors $a, b, c \in \mathcal{R}^n$ and symmetric matrix $\mathbf{0} \preceq M \in \mathcal{R}^{n\times n}$, it holds*

$$\big\langle a-b, M(a-c)\big\rangle = \frac{1}{2}\big(\|c-a\|_M^2 + \|a-b\|_M^2 - \|c-b\|_M^2\big). \tag{2.1}$$

3. Theoretical Results

In this section, by making use of the following primal-dual iterative residuals:

$$\triangle \mathbf{x}_k = \mathbf{x}_k - \mathbf{x}_{k-1}, \quad \triangle \mathbf{y}_k = \mathbf{y}_k - \mathbf{y}_{k-1}, \quad \triangle\lambda_k = \lambda_k - \lambda_{k-1}, \tag{3.1}$$

the proposed algorithm will be demonstrated to be convergent according to a quasi-monotonically nonincreasing property of the sequence $\{\mathcal{L}_\beta(w_k)\}$, and its pointwise iteration-complexity will be established in detail. To proceed, we first state our assumptions.

(A1) B is full column rank and $\mathrm{Im}(B) \supset (b \cup \mathrm{Im}(A))$.

(A2) The penalty parameter β satisfies

$$\beta > \frac{L_g}{\sqrt{1-\tau-\alpha\sigma_B}}, \quad (\tau,\alpha) \in \mathcal{D}$$

with $\mathcal{D}$ given in (1.9).

(A3) $\bar{g} = \inf\limits_{(\mathbf{x},\mathbf{y})}\left\{g(\mathbf{y}) - \frac{1}{2L_g}\|\nabla g(\mathbf{y})\|^2\right\} > -\infty$.

(A4) $\lim\limits_{\|x\|\to\infty} \inf f(x) = +\infty$.

Note that (A1) and (A2) are commonly used in the convergence analysis of ADMM-type methods for nonconvex programs, see e.g. [17, 19, 29, 31, 40, 41, 44], although the restrictions on β are different. Similar assumptions to (A3) and (A4) can be found in [29]. Based on (A1), we have $\triangle\lambda_{k+1} \in \mathrm{Im}(B)$, which together with Lemma 2.1 implies

$$\|\triangle\lambda_{k+1}\|^2 \le \sigma_B^{-1}\big\|B^{\mathsf{T}}\triangle\lambda_{k+1}\big\|^2. \tag{3.2}$$

Indeed, we can check that assumptions (A1)-(A3) hold for the two examples mentioned in the introduction. Hereafter, we denote $w = (\mathbf{x}, \mathbf{y}, \lambda)$.

Lemma 3.1. *Let $\{w_k\}$ be generated by Algorithm* 1.1. *Then, under (A2) we have*

$$\big\|B^{\mathsf{T}}\triangle\lambda_{k+1}\big\| \le L_g\|\triangle\mathbf{y}_{k+1}\|. \tag{3.3}$$

Proof. According to the optimality condition of $\mathbf{y}$-subproblem, it holds

$$\nabla g(\mathbf{y}_{k+1}) - B^{\mathsf{T}}\lambda_{k+\frac{1}{2}} + \beta B^{\mathsf{T}}\big(\mathbf{x}_{k+1}^{ad} + B\mathbf{y}_{k+1} - b\big) = \mathbf{0}. \tag{3.4}$$

So, we have by the update of λ_{k+1} that

$$B^{\mathsf{T}}\lambda_{k+1} = B^{\mathsf{T}}\Big[\lambda_{k+\frac{1}{2}} - \beta\big(\mathbf{x}_{k+1}^{ad} + B\mathbf{y}_{k+1} - b\big)\Big] = \nabla g(\mathbf{y}_{k+1}), \tag{3.5}$$

which further gives

$$B^{\mathsf{T}}\lambda_k = \nabla g(\mathbf{y}_k). \tag{3.6}$$

Subtracting (3.6) from (3.5) and taking norm on both sides, it follows from the L_g-Lipschitz continuity of ∇g that $\|B^{\mathsf{T}}\triangle\lambda_{k+1}\| \leq L_g\|\triangle\mathbf{y}_{k+1}\|$. The proof is complete. □

Note that optimality condition of the following problem is the same as (3.4):

$$\min_{\mathbf{y}}\left\{g(\mathbf{y}) + \frac{\beta}{2}\|B\mathbf{y} - c_y\|^2\right\},$$

where

$$c_y = b + \frac{\lambda_{k+\frac{1}{2}}}{\beta} - \mathbf{x}_{k+1}^{ad}.$$

Hence, this problem is equivalent to the $\mathbf{y}$-subproblem in Algorithm 1.1. When g is linearized (if it is smooth) and B has full column rank, the above problem will have closed-form solution. In addition, by choosing $G = \sigma I - \beta A^{\mathsf{T}}A$ with $\sigma \geq \beta\|A^{\mathsf{T}}A\|$, the quadratic term $\|A\mathbf{x}\|^2$ will be canceled in the iteration. As a result, the $\mathbf{x}$-subproblem in Algorithm 1.1 is converted to a proximity operator as follows:

$$\mathrm{Prox}_{f,\sigma}(c_x) := \arg\min\left\{f(\mathbf{x}) + \frac{\sigma}{2}\|\mathbf{x} - c_x\|^2\right\}, \tag{3.7}$$

where

$$c_x = \mathbf{x}_k^{md} - \frac{1}{\sigma}\Big(\beta A^{\mathsf{T}}\big(A\mathbf{x}_k^{md} + B\mathbf{y}_k - b\big) - A^{\mathsf{T}}\lambda_k\Big).$$

Since f is a proper lower semicontinuous function and bounded from below (in view of assumption (A3)), by the proximal behavior in [36] the set $\mathrm{Prox}_{f,\sigma}(c_x)$ is nonempty and compact.

Now, adding the update of $\lambda_{k+1/2}$ to the update of λ_{k+1}, we have

$$\begin{aligned}\frac{1}{\beta}\triangle\lambda_{k+1} &= -\tau(A\mathbf{x}_{k+1} + B\mathbf{y}_k - b) - \big[\alpha A\mathbf{x}_{k+1} + (1-\alpha)(b - B\mathbf{y}_k) + B\mathbf{y}_{k+1} - b\big] \\ &= -(\tau+\alpha)(A\mathbf{x}_{k+1} + B\mathbf{y}_k - b) - B\triangle\mathbf{y}_{k+1},\end{aligned}$$

which by $\tau + \alpha > 0$ gives the following lemma immediately.

Lemma 3.2. *Assume $\tau + \alpha > 0$, then the sequence $\{w_k\}$ generated by Algorithm 1.1 satisfies*

$$A\mathbf{x}_{k+1} + B\mathbf{y}_k - b = -\frac{1}{\tau+\alpha}\left(\frac{1}{\beta}\triangle\lambda_{k+1} + B\triangle\mathbf{y}_{k+1}\right). \tag{3.8}$$

Next, we present a fundamental lemma that plays a key role in analyzing convergence and convergence rate bound of Algorithm 1.1.

Lemma 3.3. *Under assumptions (A1) and (A2), there exist three constants $\zeta_0 \geq 0$ and $\zeta_1, \zeta_2 > 0$ such that*

$$\widetilde{L}_\beta(w_k) - \widetilde{L}_\beta(w_{k+1}) \geq \zeta_1\|\triangle\mathbf{x}_{k+1}\|_G^2 + \zeta_2\|\triangle\mathbf{y}_{k+1}\|^2, \tag{3.9}$$

where

$$\widetilde{L}_\beta(w_k) := \mathcal{L}_\beta(w_k) + \zeta_0\|\triangle\mathbf{x}_k\|_G^2.$$

Proof. The inequality (3.9) can be proved by the following four steps.

Step 1. By the update of $\mathbf{x}$-subproblem together with the way of generating $\mathbf{x}_k^{md}$, we have

$$
\begin{aligned}
&\mathcal{L}_\beta(\mathbf{x}_k, \mathbf{y}_k, \lambda_k) - \mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_k, \lambda_k) \\
&\geq \frac{1}{2}\left[\left\|\mathbf{x}_{k+1} - \mathbf{x}_k^{md}\right\|_G^2 - \left\|\mathbf{x}_k - \mathbf{x}_k^{md}\right\|_G^2\right] \\
&= \frac{1}{2}\Big[\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_G^2 + 2\big\langle \mathbf{x}_{k+1} - \mathbf{x}_k, G\big(\mathbf{x}_k - \mathbf{x}_k^{md}\big)\big\rangle\Big] \\
&= \frac{1}{2}\left[\|\triangle \mathbf{x}_{k+1}\|_G^2 - 2\gamma_k \langle \triangle \mathbf{x}_{k+1}, G\triangle \mathbf{x}_k\rangle\right] \\
&\geq \frac{1}{2}\left[\|\triangle \mathbf{x}_{k+1}\|_G^2 - \gamma_k\big(\|\triangle \mathbf{x}_{k+1}\|_G^2 + \|\triangle \mathbf{x}_k\|_G^2\big)\right] \\
&= \zeta_0\left[\|\triangle \mathbf{x}_{k+1}\|_G^2 - \|\triangle \mathbf{x}_k\|_G^2\right] + \zeta_1 \|\triangle \mathbf{x}_{k+1}\|_G^2,
\end{aligned} \tag{3.10}
$$

where

$$
\zeta_0 = \frac{\gamma_k}{2} \geq 0, \quad \zeta_1 = \frac{1 - 2\gamma_k}{2} > 0. \tag{3.11}
$$

Step 2. By the update of $\mathbf{y}$-subproblem we obtain

$$
\begin{aligned}
&g(\mathbf{y}_k) - \big\langle \lambda_{k+\frac{1}{2}}, B\mathbf{y}_k\big\rangle + \frac{\beta}{2}\big\|\mathbf{x}_{k+1}^{ad} + B\mathbf{y}_k - b\big\|^2 \\
&\geq g(\mathbf{y}_{k+1}) - \big\langle \lambda_{k+\frac{1}{2}}, B\mathbf{y}_{k+1}\big\rangle + \frac{\beta}{2}\big\|\mathbf{x}_{k+1}^{ad} + B\mathbf{y}_{k+1} - b\big\|^2,
\end{aligned}
$$

which, by Lemma 2.2, is equivalently expressed as

$$
\begin{aligned}
&g(\mathbf{y}_k) - g(\mathbf{y}_{k+1}) + \big\langle \lambda_{k+\frac{1}{2}}, B\triangle \mathbf{y}_{k+1}\big\rangle + \frac{\beta}{2}\|B\triangle \mathbf{y}_{k+1}\|^2 \\
&\geq \beta\big\langle B\triangle \mathbf{y}_{k+1}, \mathbf{x}_{k+1}^{ad} + B\mathbf{y}_{k+1} - b\big\rangle.
\end{aligned} \tag{3.12}
$$

Therefore, it can be deduced that

$$
\begin{aligned}
&\mathcal{L}_\beta\big(\mathbf{x}_{k+1}, \mathbf{y}_k, \lambda_{k+\frac{1}{2}}\big) - \mathcal{L}_\beta\big(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \lambda_{k+\frac{1}{2}}\big) \\
&= g(\mathbf{y}_k) - g(\mathbf{y}_{k+1}) + \big\langle \lambda_{k+\frac{1}{2}}, B\triangle \mathbf{y}_{k+1}\big\rangle \\
&\qquad + \frac{\beta}{2}\big(\|A\mathbf{x}_{k+1} + B\mathbf{y}_k - b\|^2 - \|A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b\|^2\big) \\
&= g(\mathbf{y}_k) - g(\mathbf{y}_{k+1}) + \big\langle \lambda_{k+\frac{1}{2}}, B\triangle \mathbf{y}_{k+1}\big\rangle \\
&\qquad - \beta\big\langle B\triangle \mathbf{y}_{k+1}, A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b\big\rangle + \frac{\beta}{2}\|B\triangle \mathbf{y}_{k+1}\|^2 \\
&\geq \beta\big\langle B\triangle \mathbf{y}_{k+1}, \mathbf{x}_{k+1}^{ad} + B\mathbf{y}_{k+1} - b\big\rangle - \beta\big\langle B\triangle \mathbf{y}_{k+1}, A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b\big\rangle \\
&= \beta(\alpha - 1)\big\langle B\triangle \mathbf{y}_{k+1}, A\mathbf{x}_{k+1} + B\mathbf{y}_k - b\big\rangle \\
&= \frac{1 - \alpha}{\tau + \alpha}\big[\beta\|B\triangle \mathbf{y}_{k+1}\|^2 + \big\langle \triangle \mathbf{y}_{k+1}, B^{\mathsf{T}}\triangle \lambda_{k+1}\big\rangle\big],
\end{aligned} \tag{3.13}
$$

where the second equality follows from Lemma 2.2, the first inequality is based on (3.12), the third equality uses the update of $\mathbf{x}_{k+1}^{ad}$ and the final equality applies (3.8).

Step 3. Note that

$$
\mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_k, \lambda_k) - \mathcal{L}_\beta\big(\mathbf{x}_{k+1}, \mathbf{y}_k, \lambda_{k+\frac{1}{2}}\big) + \mathcal{L}_\beta\big(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \lambda_{k+\frac{1}{2}}\big) - \mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \lambda_{k+1})
$$

$$
\begin{aligned}
&= \left\langle \lambda_{k+\frac{1}{2}} - \lambda_k, A\mathbf{x}_{k+1} + B\mathbf{y}_k - b \right\rangle - \left\langle \lambda_{k+\frac{1}{2}} - \lambda_{k+1}, A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b \right\rangle \\
&= \left\langle \lambda_{k+\frac{1}{2}} - \lambda_k, A\mathbf{x}_{k+1} + B\mathbf{y}_k - b \right\rangle - \left\langle \lambda_{k+\frac{1}{2}} \underbrace{-\lambda_k + \lambda_k} - \lambda_{k+1}, A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b \right\rangle \\
&= \left\langle \lambda_{k+\frac{1}{2}} - \lambda_k, -B\triangle\mathbf{y}_{k+1} \right\rangle + \left\langle \triangle\lambda_{k+1}, A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b \right\rangle \\
&= \tau\beta\left\langle A\mathbf{x}_{k+1} + B\mathbf{y}_k - b, B\triangle\mathbf{y}_{k+1} \right\rangle + \left\langle \triangle\lambda_{k+1}, A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} \underbrace{+B\mathbf{y}_k - B\mathbf{y}_k} - b \right\rangle \\
&= \left\langle A\mathbf{x}_{k+1} + B\mathbf{y}_k - b, \triangle\lambda_{k+1} + \tau\beta B\triangle\mathbf{y}_{k+1} \right\rangle + \left\langle \triangle\lambda_{k+1}, B\triangle\mathbf{y}_{k+1} \right\rangle \\
&= -\frac{1}{\tau+\alpha}\left\langle \frac{1}{\beta}\triangle\lambda_{k+1} + B\triangle\mathbf{y}_{k+1}, \triangle\lambda_{k+1} + \tau\beta B\triangle\mathbf{y}_{k+1} \right\rangle + \left\langle \triangle\lambda_{k+1}, B\triangle\mathbf{y}_{k+1} \right\rangle \\
&= -\frac{\tau\beta}{\tau+\alpha}\|B\triangle\mathbf{y}_{k+1}\|^2 - \frac{1}{(\tau+\alpha)\beta}\|\triangle\lambda_{k+1}\|^2 - \frac{1-\alpha}{\tau+\alpha}\left\langle \triangle\lambda_{k+1}, B\triangle\mathbf{y}_{k+1} \right\rangle, \qquad (3.14)
\end{aligned}
$$

where the sixth equality is based on (3.8).

Step 4. Summing the inequalities (3.10), (3.13) and the equality (3.14), we get

$$
\begin{aligned}
&\mathcal{L}_\beta(\mathbf{x}_k, \mathbf{y}_k, \lambda_k) - \mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \lambda_{k+1}) \\
&\geq \zeta_0\left[\|\triangle\mathbf{x}_{k+1}\|_G^2 - \|\triangle\mathbf{x}_k\|_G^2\right] + \zeta_1\|\triangle\mathbf{x}_{k+1}\|_G^2 + R_\triangle,
\end{aligned}
$$

where

$$
\begin{aligned}
R_\triangle &= \Big(\frac{1}{\tau+\alpha} - 1\Big)\beta\|B\triangle\mathbf{y}_{k+1}\|^2 - \frac{1}{(\tau+\alpha)\beta}\|\triangle\lambda_{k+1}\|^2 \\
&\geq \Big(\frac{1}{\tau+\alpha} - 1\Big)\beta\|B\triangle\mathbf{y}_{k+1}\|^2 - \frac{1}{(\tau+\alpha)\beta\sigma_B}\left\|B^\top\triangle\lambda_{k+1}\right\|^2 \\
&\geq \Big(\frac{1}{\tau+\alpha} - 1\Big)\beta\|B\triangle\mathbf{y}_{k+1}\|^2 - \frac{L_g^2}{(\tau+\alpha)\beta\sigma_B}\|\triangle\mathbf{y}_{k+1}\|^2 \\
&\geq \Big(\frac{1}{\tau+\alpha} - 1\Big)\beta\sigma_B\|\triangle\mathbf{y}_{k+1}\|^2 - \frac{L_g^2}{(\tau+\alpha)\beta\sigma_B}\|\triangle\mathbf{y}_{k+1}\|^2 \\
&= \zeta_2\|\triangle\mathbf{y}_{k+1}\|^2
\end{aligned}
$$

with

$$
\zeta_2 = \frac{(1-\tau-\alpha)\beta^2\sigma_B^2 - L_g^2}{(\tau+\alpha)\beta\sigma_B} > 0, \quad \text{[due to (A2)]}.
$$

Here, the first inequality of $R_\triangle$ follows from (3.2) and the third inequality is due to (A1). As a result, the whole proof is complete with the previous notation $\widetilde{L}_\beta(w_k)$. □

Theorem 3.1. *Let $\{w_k\}$ be generated by Algorithm 1.1. Then, under (A1)-(A4) we have:*

- *The sequence $\{\widetilde{L}_\beta(w_k)\}$ is convergent.*
- *The residuals $\|\triangle\mathbf{x}_{k+1}\|_G$, $\|\triangle\mathbf{y}_{k+1}\|$, and $\|\triangle\lambda_{k+1}\|$ converge to zero as $k \to \infty$.*

Proof. To show convergence of $\{\widetilde{L}_\beta(w_k)\}$, we need to ensure that the sequence $\{w_k\}$ is bounded first. By (A2), it holds

$$
L_g < \sqrt{1-\tau-\alpha}\,\beta\sigma_B < \beta\sigma_B. \qquad (3.15)
$$

We know from Lemma 3.3 that

$$\begin{aligned}
\widetilde{L}_\beta(\mathbf{x}_0,\mathbf{y}_0,\lambda_0) &\geq L_\beta(\mathbf{x}_k,\mathbf{y}_k,\lambda_k)+\zeta_0\|\triangle\mathbf{x}_k\|_G^2 \geq L_\beta(\mathbf{x}_k,\mathbf{y}_k,\lambda_k)\\
&= f(\mathbf{x}_k)+g(\mathbf{y}_k)-\frac{1}{2\beta}\|\lambda_k\|^2+\frac{\beta}{2}\left\|A\mathbf{x}_k+B\mathbf{y}_k-b-\frac{\lambda_k}{\beta}\right\|^2\\
&\geq f(\mathbf{x}_k)+g(\mathbf{y}_k)-\frac{1}{2\beta\sigma_B}\left\|B^\mathsf{T}\lambda_k\right\|^2+\frac{\beta}{2}\left\|A\mathbf{x}_k+B\mathbf{y}_k-b-\frac{\lambda_k}{\beta}\right\|^2\\
&= f(\mathbf{x}_k)+\left(g(\mathbf{y}_k)-\frac{1}{2L_g}\left\|\nabla g(\mathbf{y}_k)\right\|^2\right)+\left(\frac{1}{2L_g}-\frac{1}{2\beta\sigma_B}\right)\left\|B^\mathsf{T}\lambda_k\right\|^2\\
&\quad+\frac{\beta}{2}\left\|A\mathbf{x}_k+B\mathbf{y}_{k+1}-b-\frac{\lambda_k}{\beta}\right\|^2\\
&\geq f(x_k)+\bar{g}+\left(\frac{1}{2L_g}-\frac{1}{2\beta\sigma_B}\right)\left\|B^\mathsf{T}\lambda_k\right\|^2+\frac{\beta}{2}\left\|A\mathbf{x}_k+B\mathbf{y}_k-b-\frac{\lambda_k}{\beta}\right\|^2. \quad (3.16)
\end{aligned}$$

The assumption (A4) implies that $\inf_x f(x) > -\infty$. Then we know from (3.16) that $\{x_k\}$, $\{\|B^T\lambda_k\|\}$ and $\{\|A\mathbf{x}_k+B\mathbf{y}_k-b-\lambda_k/\beta\|\}$ are all bounded. Since $\lambda_k\in\mathrm{Im}(B)$, it follows from Lemma 2.1 that $\{\lambda_k\}$ is bounded. The boundedness of $\{x_k\}$ and $\{\lambda_k\}$ leads to the boundedness of $\{y_k\}$ because B has full column rank. Hence, the sequence $\{w_k\}$ is bounded.

Since $\{w_k\}$ is bounded, $\{\widetilde{L}_\beta(w_k)\}$ is also bounded from below and there exists at least one limit point. Without loss of generality, let w_* be the limit point of $\{w_k\}$ whose subsequence is $\{w_{k_j}\}$. Then, the lower semicontinuity of $\{\widetilde{L}_\beta(w)\}$ indicates

$$\widetilde{L}_\beta(w_*)\leq \lim_{j\to+\infty}\inf\widetilde{L}_\beta(w_{k_j}).$$

That is, $\{\widetilde{L}_\beta(w_{k_j})\}$ is bounded from below, which further implies convergence of $\{\widetilde{L}_\beta(w_k)\}$ based on Lemma 3.3.

Now, summing the inequality (3.9) over $k=0,1,\ldots,\infty$, we have by the convergence of $\{\widetilde{L}_\beta(w_k)\}$ that

$$\zeta_1\sum_{k=0}^{\infty}\|\triangle\mathbf{x}_{k+1}\|_G^2+\zeta_2\sum_{k=0}^{\infty}\|\triangle\mathbf{y}_{k+1}\|^2\leq\mathcal{L}_\beta(w_0)-\widetilde{\mathcal{L}}_\beta(w_{k+1})<\infty,$$

which suggests $\|\triangle\mathbf{x}_{k+1}\|_G\to 0$ and $\|\triangle\mathbf{y}_{k+1}\|\to 0$. So, in view of Lemma 3.1 and (3.2), the following holds:

$$\|\triangle\lambda_{k+1}\|\leq\frac{1}{\sqrt{\sigma_B}}\left\|B^\mathsf{T}\triangle\lambda_{k+1}\right\|\leq\frac{L_g}{\sqrt{\sigma_B}}\|\triangle\mathbf{y}_{k+1}\| \;\to\; 0. \quad (3.17)$$

The proof is complete. □

Theorem 3.1 indicates that the augmented Lagrangian function of the problem (1.1) is convergent, and the primal and dual residuals converge to zero. In what follows, we present a key theorem about pointwise iteration-complexity of the proposed algorithm with respect to the primal-dual residuals. Actually, the following first assertion implies that any accumulation point of $\{w_k\}$ is a stationary point of $\{L_\beta(w_k)\}$ compared to Definition 2.2.

Theorem 3.2. *Let $\{w_k\}$ be generated by Algorithm 1.1. Then, under assumptions (A1)-(A4)*

- *It holds*
$$\lim_{k\to\infty} \operatorname{dist}\big(\mathbf{0}, \partial L_\beta(w^{k+1})\big) = 0. \tag{3.18}$$
- *The sequence $\{f(\mathbf{x}_{k+1}) + g(\mathbf{y}_{k+1})\}$ is convergent.*
- *Let $C_0 := \widetilde{\mathcal{L}}_\beta(w_0) - \bar{g} - \inf_x f(x)$. Then, for any integer $k \geq 1$, there exists $j \leq k$ and $\zeta_i > 0, i = 1,2,3$, such that*
$$\|\triangle \mathbf{x}_j\|_G^2 \leq \frac{C_0}{\zeta_1(k+1)}, \quad \|\triangle \mathbf{y}_j\|^2 \leq \frac{C_0}{\zeta_2(k+1)}, \quad \|\triangle \lambda_j\|^2 \leq \frac{C_0}{\zeta_3(k+1)}. \tag{3.19}$$

Proof. Using (3.8) again, we have
$$A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b = -\frac{1}{\tau+\alpha}\left(\frac{1}{\beta}\triangle\lambda_{k+1} + B\triangle\mathbf{y}_{k+1}\right) + B\triangle\mathbf{y}_{k+1},$$
which by the second result of Theorem 3.1 suggests
$$\lim_{k\to\infty} A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b = \mathbf{0}. \tag{3.20}$$
Therefore,
$$\lim_{k\to\infty} \nabla_\lambda L_\beta(w_{k+1}) = \lim_{k\to\infty} -(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b) = \mathbf{0}. \tag{3.21}$$
By the first-order optimality condition of $\mathbf{y}$-subproblem, it holds
$$\begin{aligned}
\mathbf{0} &= \nabla g(\mathbf{y}_{k+1}) - B^\top \lambda_{k+\frac{1}{2}} + \beta B^\top\big(\mathbf{x}_{k+1}^{ad} + B\mathbf{y}_{k+1} - b\big)\\
&= \nabla g(\mathbf{y}_{k+1}) - B^\top \lambda_{k+1} + \beta B^\top(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b)\\
&\quad + B^\top\big(\lambda_{k+1} - \lambda_{k+\frac{1}{2}}\big) + \beta B^\top\big(\mathbf{x}_{k+1}^{ad} - A\mathbf{x}_{k+1}\big)\\
&= \nabla g(\mathbf{y}_{k+1}) - B^\top \lambda_{k+1} + \beta B^\top(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b)\\
&\quad - \beta B^\top(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b),
\end{aligned}$$
which gives
$$\lim_{k\to\infty} \nabla_{\mathbf{y}} L_\beta(w_{k+1}) = \lim_{k\to\infty} \beta B^\top(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b) = \mathbf{0}. \tag{3.22}$$
Analogously, by the update of $\mathbf{x}$-subproblem, there exists $d_{k+1} \in \partial f(\mathbf{x}_{k+1})$ such that
$$\begin{aligned}
\mathbf{0} &= d_{k+1} - A^\top\lambda_{k+1} + \beta A^\top(A\mathbf{x}_{k+1} + B\mathbf{y}_k - b) + G\big(\mathbf{x}_{k+1} - \mathbf{x}_k^{md}\big)\\
&= d_{k+1} - A^\top\lambda_{k+1} + \beta A^\top(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b)\\
&\quad + \beta A^\top B(\mathbf{y}_k - \mathbf{y}_{k+1}) + G(\mathbf{x}_{k+1} - \mathbf{x}_k - \gamma_k\triangle\mathbf{x}_k)\\
&= d_{k+1} - A^\top\lambda_{k+1} + \beta A^\top(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b)\\
&\quad - \beta A^\top B\triangle\mathbf{y}_{k+1} - G(\gamma_k\triangle\mathbf{x}_k - \triangle\mathbf{x}_{k+1}).
\end{aligned}$$
By defining
$$\overline{d}_{k+1} := d_{k+1} - A^\top\lambda_{k+1} + \beta A^\top(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b),$$
we have $\overline{d}_{k+1} \in \partial_{\mathbf{x}} L_\beta(w_{k+1})$ and furthermore
$$\lim_{k\to\infty} \overline{d}_{k+1} = \lim_{k\to\infty} \beta A^\top B\triangle\mathbf{y}_{k+1} + G(\gamma_k\triangle\mathbf{x}_k - \triangle\mathbf{x}_{k+1}) = \mathbf{0}. \tag{3.23}$$
Thus, it follows from (3.21)-(3.23) that (3.18) holds.

For the second assertion, since

$$f(\mathbf{x}_{k+1}) + g(\mathbf{y}_{k+1}) = L_\beta(w_{k+1}) + \langle \lambda_{k+1}, A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b \rangle - \frac{\beta}{2}\|A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b\|^2,$$

the sequence $\{f(\mathbf{x}_{k+1}) + g(\mathbf{y}_{k+1})\}$ is convergent by the first conclusion of Theorem 3.1, the boundedness of $\{\lambda_k\}$ and (3.20).

We finally prove the pointwise iteration complexity in (3.19). Recall that $\inf_x f(x) > -\infty$, employing (3.16) again yields

$$-\widetilde{L}_\beta(w_{k+1}) \le -\bar{g} - \inf_x f(x).$$

Then, for any $k \ge 0$, it follows from Lemma 3.3 that

$$\sum_{j=0}^{k} \left(\zeta_1 \|\triangle \mathbf{x}_j\|_G^2 + \zeta_2 \|\triangle \mathbf{y}_j\|^2\right) \le \widetilde{\mathcal{L}}_\beta(w_0) - \bar{g} - \inf_x f(x) = C_0,$$

which shows there exists $j \le k$ such that

$$\|\triangle \mathbf{x}_j\|_G^2 \le \frac{C_0}{\zeta_1(k+1)}, \quad \|\triangle \mathbf{y}_j\|^2 \le \frac{C_0}{\zeta_2(k+1)}.$$

The final convergence rate bound in (3.19) can be also verified by (3.17) with $\zeta_3 = L_g^2/(\sigma_B \zeta_2)$. The proof is complete. $\square$

In order to reduce error bounds of the primal-dual residuals, the following remark provides an adaptive way to update the parameter γ_k related to ζ_1 by making use of the so-called Nesterov's acceleration (proposed originally in [34]), and it also suggests how to choose reasonable values of the parameters τ and α.

Remark 3.1. By the above convergence analysis, if $G \succ \mathbf{0}$, then convergence of Algorithm 1.1 can be guaranteed by $\gamma_k \in [0, 1/2)$. In such case we can update γ_k adaptively

$$\gamma_k = \frac{\theta_{k-1} - 1}{2\theta_k}, \quad \theta_k = \frac{1}{2}\left(1 + \sqrt{1 + 4\theta_{k-1}^2}\right), \quad \theta_{-1} := 1. \tag{3.24}$$

Note that

$$\zeta_2 = -\beta\sigma_B + \frac{1}{\tau + \alpha}\left[\beta\sigma_B - \frac{L_g^2}{\beta\sigma_B}\right]$$

is inversely proportional to $(\tau + \alpha)$ since $L_g < \beta\sigma_B$. Together with the connection $\zeta_3 = L_g^2/\zeta_2$, we can choose $(\tau + \alpha) \to 1$ to get smaller error bound of $\|\triangle \lambda_j\|^2$ in (3.19). In the next section, numerical experiments will show how to determine reasonable values of τ and α in detail.

4. Numerical Experiments

In this section, we apply the proposed algorithm to solve a class of signal processing problems to investigate its numerical performance. All experiments are conducted on a PC with Windows 10 operating system and MATLAB R2018a (64-bit), with an Intel i7-8700K CPU and 16 GB memory.

Applying Algorithm 1.1 to the nonconvex problem (1.4), we have by (3.7) that

$$\mathbf{x}_{k+1} = \mathrm{Prox}_{\|\mathbf{x}\|_{1/2}^{1/2},\sigma/\mu}\left(\mathbf{x}_k^{md} - \frac{1}{\sigma}\Big(\beta A^\mathsf{T}\big(A\mathbf{x}_k^{md} - \mathbf{y}_k - b\big) - A^\mathsf{T}\lambda_k\Big)\right),$$

which is the half shrinkage operator [42] defined as

$$\mathrm{Prox}_{\|\mathbf{x}\|_{1/2}^{1/2},\nu}(\mathbf{x}) = \big(l_\nu(\mathbf{x}_1), l_\nu(\mathbf{x}_2), \cdots, l_\nu(\mathbf{x}_m)\big)^\mathsf{T},$$

where

$$l_\nu(\mathbf{x}_i) = \begin{cases} \dfrac{2\mathbf{x}_i}{3}\left[1 + \cos\dfrac{2}{3}\big(\pi - \phi(\mathbf{x}_i)\big)\right], & \text{if} \quad |\mathbf{x}_i| > \dfrac{3\sqrt[3]{2}}{4}\nu^{\frac{2}{3}}, \\ 0, & \text{otherwise}, \end{cases}$$

and

$$\phi(\mathbf{x}_i) = \arccos\left(\frac{\nu}{8}\left(\frac{|\mathbf{x}_i|}{3}\right)^{-\frac{3}{2}}\right).$$

Note that assumption (A4) holds since here $f(\mathbf{x}) = \mu\|\mathbf{x}\|_{1/2}^{1/2}$. Besides, it is easy to obtain

$$\mathbf{y}_{k+1} = \frac{c + \beta\mathbf{x}_{k+1}^{ad} - \lambda_{k+1/2}}{1+\beta}.$$

Similar way can be used for the compared algorithms to solve (1.3) and its l_1-regularization model.

Aiming to achieve fast convergence and make Algorithm 1.1 less independent on initial guess of the penalty parameter β, as suggested by He *et al.* [24] we adopt the following technique to update it adaptively:

$$\beta_{k+1} = \begin{cases} \eta^{\mathrm{incr}}\beta_k, & \text{if} \quad \|r_k\|_2 > \nu\|s_k\|_2, \\ \beta_k/\eta^{\mathrm{decr}}, & \text{if} \quad \|s_k\|_2 > \nu\|r_k\|_2, \\ \beta_k, & \text{otherwise}, \end{cases} \tag{4.1}$$

where $\nu, \eta^{\mathrm{incr}}$ and η^{decr} are three positive parameters with values larger than 1, for instance, $\nu = 10, \eta^{\mathrm{incr}} = \eta^{\mathrm{decr}} = 2$. To solve (1.1) using Algorithm 1.1, we have

$$\|r_k\| = \|A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b\|, \tag{4.2}$$

and

$$\|s_k\| = \big\|A^\mathsf{T}\triangle\lambda_{k+1} + \beta A^\mathsf{T}(A\mathbf{x}_{k+1} + B\mathbf{y}_k - b) + G(\triangle\mathbf{x}_{k+1} - \gamma_k\triangle\mathbf{x}_k)\big\|,$$

which represent the equality constrained error and the optimality error, respectively. Here, it is easy to check that $\mathbf{0} \in \partial f(\mathbf{x}_{k+1}) - A^\mathsf{T}\lambda_{k+1} + s_k$. In order to satisfy assumption (A2), we need to update

$$\beta = \min\left(\beta_{k+1}, \frac{1.01L_g}{\sqrt{1-\tau-\alpha}\sigma_B}\right)$$

at each iteration. For the problem (1.4), we have $L_g = 1$ and $\sigma_B = 1$, and thus assumption (A3) holds. If not specified, the initial penalty parameter β_0 is chosen as 0.04, the starting points $(\mathbf{x}_0, \mathbf{y}_0)$ and λ_0 are respectively set as vector of ones with proper length and zero, and the matrix $G = \sigma I - \beta A^\mathsf{T}A$ with $\sigma = 1.01\beta\|A^\mathsf{T}A\|$. The parameter γ_k is updated adaptively according to (3.24). Throughout the experiments we use the following stopping criterion (see also [29, 41]):

$$\mathrm{IRE(k)} := \frac{\max\left\{\|\mathbf{x}_k - \mathbf{x}_{k-1}\|, \|\mathbf{y}_k - \mathbf{y}_{k-1}\|, \|\lambda_k - \lambda_{k-1}\|\right\}}{\max\left\{\|\mathbf{x}_{k-1}\|, \|\mathbf{y}_{k-1}\|, \|\lambda_{k-1}\|, 1\right\}} < \epsilon, \tag{4.3}$$

where ϵ is a given tolerance error. Note that this stopping criterion corresponds to the pointwise iteration complexity (3.19), so it is well defined.

4.1. Basic experiments

We first consider the reformulated sparse signal recovery problem (1.4) with an original signal $x \in \mathcal{R}^{3072}$ containing 160 spikes with amplitude ± 1. The measurement matrix $A \in \mathcal{R}^{1024\times 3072}$ is drawn firstly from the standard norm distribution $\mathcal{N}(0,1)$ and then each of its columns is normalized. Specifically, we use the following MATLAB codes to generate the original signal x_{orig}, the data A, c and μ:

```
randn('state', 0); rand('state',0);
l = 1024; m = 3072;
T = 160;  % number of spikes
x_orig = zeros(m,1); q = randperm(m);
x_orig(q(1:T)) = sign(randn(T,1));  % original signal
A = randn(l,m);
A = A*spdiags(1./sqrt(sum(A.^2))',0,m,m);  % normalize columns
sig = 0.01;  % noise standard deviation
c = A*x_orig + sig*randn(l,1);  % noisy observations
mu_max = norm( A'*c,'inf');
mu = 0.1*mu_max;  % regularization parameter
```

Setting the tolerance $\epsilon = 10^{-15}$, we evaluate the effect of parameters (τ, α) restricted in (1.9) on the numerical performance of Algorithm 1.1 (in fact, we choose parameter values around $(\tau, \alpha) = (0.3, 0.32)$, because we find that this setting performs slightly better than some pairs after running a lot of values restricted in (1.9)). We also randomly choose four pairs of (τ, α) to carry out the experiments.

Table 4.1 reports computational results[1)] of several quality measurements, including "IT", "CPU", "IRE", "EQU", which denote respectively the iteration number, the CPU time in seconds, the final relative iterative error IRE(k) defined in (4.3) and the final feasibility error $\|r_k\|$ defined in (4.2). We use l_2-error (defined as $\|x_k - x_{\mathrm{orig}}\|/\|x_{\mathrm{orig}}\|$) to represent the relative error to measure recovery quality of a signal. As shown in Table 4.1, setting $(\tau, \alpha) = (0.65, 0.32)$ would be a reasonable choice for Algorithm 1.1 to solve (1.4), because the resulting iteration number and the CPU time are relatively smaller while the results in each of the last three columns are nearly the same when the stopping criterion is satisfied. Hence, in the following experiments, Algorithm 1.1 uses the default parameters $(\tau, \alpha) = (0.65, 0.32)$.

Next, we use the aforementioned code to investigate the effect of regularization parameter μ on Algorithm 1.1 for solving the problem (1.4) with a large data $A \in \mathcal{R}^{2048\times 5000}$ and the same spikes, and the tolerance is now set as $\epsilon = 10^{-12}$. Fig. 4.1 depicts convergence

[1)] "–" means that the stopping criterion is not satisfied after 800 iterations, and the bold number in that row indicates the best result obtained by changing (τ, α) belong to $(0, 1)$.

Table 4.1: Results of Algorithm 1.1 with different (τ, α) for solving problem (1.4).

(τ, α)	IT	CPU	IRE	EQU	l_2-error
(0.3, 0.10)	472	22.67	9.47e-16	6.10e-14	6.79e-2
(0.3, 0.15)	443	21.12	9.23e-16	5.18e-14	6.79e-2
(0.3, 0.20)	500	24.11	9.77e-16	4.82e-14	6.79e-2
(0.3, 0.25)	379	18.23	9.79e-16	6.33e-14	6.79e-2
(0.3, 0.30)	350	16.79	9.78e-16	5.88e-14	6.79e-2
(0.3, 0.32)	344	16.66	9.45e-16	6.33e-14	6.79e-2
(0.3, 0.35)	544	26.17	9.79e-16	3.28e-14	6.79e-2
(0.3, 0.40)	510	24.30	8.87e-16	5.11e-14	6.79e-2
(0.3, 0.45)	485	23.28	9.93e-16	3.34e-14	6.79e-2
(0.3, 0.50)	458	21.95	9.34e-16	3.24e-14	6.79e-2
(0.3, 0.55)	433	20.86	9.58e-16	3.29e-14	6.79e-2
(0.3, 0.60)	413	20.30	9.45e-16	3.24e-14	6.79e-2
(0.3, 0.65)	396	19.24	9.33e-16	3.26e-14	6.79e-2
(0.3, 0.68)	387	18.66	8.91e-16	3.37e-14	6.79e-2
(-0.3, 0.32)	–	–	–	–	–
(-0.2, 0.32)	–	–	–	–	–
(-0.1, 0.32)	695	33.51	9.95e-16	5.41e-14	6.79e-2
(0, 0.32)	498	23.77	9.99e-16	8.38e-14	6.79e-2
(0.1, 0.32)	697	33.49	9.33e-16	5.68e-14	6.79e-2
(0.2, 0.32)	391	18.93	9.54e-16	7.85e-14	6.79e-2
(0.3, 0.32)	344	16.62	9.45e-16	6.33e-14	6.79e-2
(0.4, 0.32)	502	24.16	8.85e-16	4.01e-14	6.79e-2
(0.5, 0.32)	451	21.61	9.49e-16	2.97e-14	6.79e-2
(0.6, 0.32)	404	19.52	9.64e-16	3.74e-14	6.79e-2
(0.62, 0.32)	397	19.20	9.71e-16	4.34e-14	6.79e-2
(0.65, 0.32)	**279**	**13.52**	9.98e-16	3.19e-14	6.79e-2
(0.67, 0.32)	318	15.59	8.43e-16	3.11e-14	6.79e-2
(0.90, 0.05)	396	19.06	9.26e-16	3.27e-14	6.79e-2
(0.80, 0.15)	396	19.02	9.33e-16	3.28e-14	6.79e-2
(0.01, 0.90)	411	19.78	9.63e-16	3.10e-14	6.79e-2
(0.05, 0.70)	485	24.62	9.62e-16	3.10e-14	6.79e-2

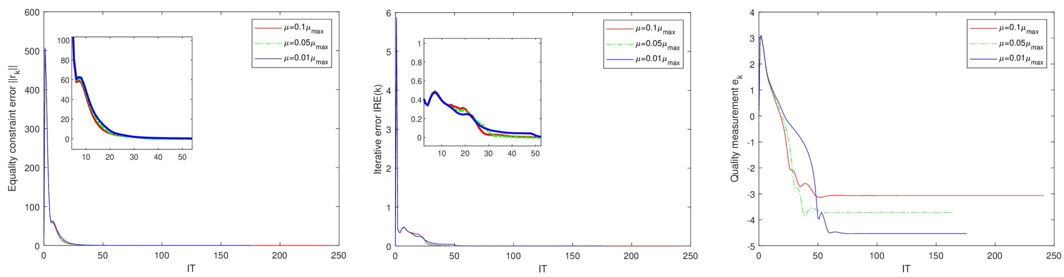

Fig. 4.1. Convergence behaviors of the equality constraint error $\|r_k\|$ (left), the iterative error IRE(k) (middle) and the recovery signal quality e_k (right) by Algorithm 1.1 for solving (1.4) with $(l, m) = (2048, 5000)$ but with different regularization parameters.

behaviors of the equality constraint error $\|r_k\|$, the iterative error IRE(k) and the recovery signal quality $e_k := \log_{10}(\|\mathbf{x}_k - \mathbf{x}_{\text{orig}}\|/\|\mathbf{x}_{\text{orig}}\|)$ along the iteration process using Algorithm 1.1 with μ=$0.1\mu_{\max}$, $0.05\mu_{\max}$, $0.01\mu_{\max}$, respectively. Fig. 4.2 plots the results to visualize the recovery quality of the signal versus the original signal, where the upper-left plot shows the minimum energy reconstruction signal $A^{\dagger}c$ (which is the point satisfying $A^{\mathsf{T}}A\mathbf{x} = A^{\mathsf{T}}c$) versus the original signal. An outstanding observation from Fig. 4.1 is that the smaller the value of μ is, the smaller the iteration number is (and the better the recovery quality of the signal is). After identifying the nonzero positions in the reconstructed signal, it always has the correct number of spikes for the case with $\mu = 0.01\mu_{\max}$ and is closer to the original signal.

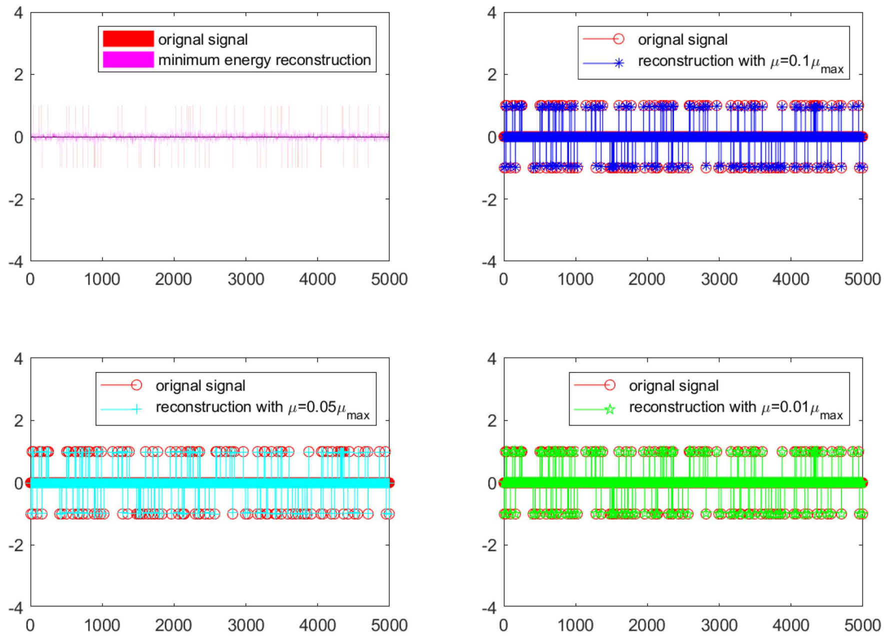

Fig. 4.2. Comparison between the original signal and reconstructed signal by Algorithm 1.1 for solving (1.4) with $(l, m) = (2048, 5000)$ but with different regularization parameter μ.

4.2. Comparative experiments

In the following, we compare Algorithm 1.1 with the regularized ADMM (RADMM, [26]) for solving two different cases of the sparse signal recovery problem:

Case 1. The nonconvex problem (1.4) with $l_{1/2}$-regularization term.

Case 2. The convex problem (1.2)[1] with l_1-regularization term.

We also apply the Bregman modification of ADMM (BADMM, [39]) and the symmetric ADMM (SADMM, [41]) to solve the problems (1.2) and (1.3) by introducing auxiliary variable $\mathbf{y} = \mathbf{x}$ for the sparse objective function. The proximal matrix in [26] is $G = \alpha I - \beta A^{\mathsf{T}}A$, where $(\alpha, \beta) = (2.5, 0.12)$. The Bregman distance and related parameters of [39] use the default settings as mentioned therein, while the penalty parameter is set as 0.15 since it performs better

[1] Note that this is also a special case of (1.1) with $f(\mathbf{x}) = \mu\|\mathbf{x}\|_1$, $g(\mathbf{y}) = \|\mathbf{y} - c\|^2/2$, $B = -I$ and $b = \mathbf{0}$.

than using the value 10. As explained in [41], the involved stepsize parameter uses the tuned value $\alpha = 0.7$, the penalty parameter is initialized as $\beta = 0.1\hat{\beta}$ where $\hat{\beta} = 2\lambda_{\max}(A^{\mathsf{T}}A)/(1-\alpha)$ and then it is updated as $\min(2\beta, 1.01\hat{\beta})$ whenever some certain conditions hold. For BADMM and SADMM, we define $\|r_k\| = \|\mathbf{x}_{k+1} - \mathbf{y}_{k+1}\|$. Table 4.2 reports the corresponding numerical

Table 4.2: Comparative results of the state-of-the-art algorithms under $\epsilon = 10^{-15}$.

$l_{1/2}$-regularizer					l_1-regularizer			
(l, m)	IT	CPU	EQU	l_2 error	IT	CPU	EQU	l_2 error
Algorithm 1.1								
(1024, 3000)	**358**	16.37	4.60e-14	**1.20e-2**	501	22.60	1.93e-14	3.70e-2
(1024, 4000)	**367**	25.93	4.18e-14	**1.28e-2**	507	35.61	4.78e-14	4.26e-2
(2048, 5000)	**215**	**30.65**	2.84e-14	**1.08e-2**	250	36.01	3.27e-14	2.66e-2
(2048, 6000)	**222**	**37.43**	3.98e-14	**1.20e-2**	266	44.96	3.47e-14	3.07e-2
(3000, 7000)	**201**	**58.08**	2.71e-14	**1.17e-2**	231	66.91	2.93e-14	2.60e-2
(3000, 8000)	**205**	**71.36**	3.77e-14	**1.10e-2**	230	79.55	3.62e-14	2.58e-2
(4000, 9000)	**199**	**90.51**	3.29e-14	**1.11e-2**	231	**105.35**	2.87e-14	2.69e-2
(4000, 10000)	**202**	**118.62**	2.37e-14	**1.03e-2**	231	135.72	2.91e-14	2.51e-2
RADMM								
(1024, 3000)	562	**12.51**	4.82e-14	1.20e-2	677	14.91	1.04e-13	3.70e-2
(1024, 4000)	570	**17.29**	5.57e-14	1.28e-2	729	**21.81**	6.54e-14	4.26e-2
(2048, 5000)	486	37.19	7.62e-14	1.08e-2	553	41.95	5.87e-14	2.66e-2
(2048, 6000)	492	45.15	5.57e-14	1.20e-2	567	51.95	6.58e-14	3.07e-2
(3000, 7000)	462	70.99	4.01e-14	1.17e-2	516	78.47	5.06e-14	2.60e-2
(3000, 8000)	461	81.13	5.98e-14	1.10e-2	508	89.06	6.27e-14	2.58e-2
(4000, 9000)	456	123.31	5.49e-14	1.11e-2	513	138.54	6.85e-14	2.69e-2
(4000, 10000)	455	137.83	5.95e-14	1.03e-2	513	154.81	5.26e-14	2.51e-2
BADMM								
(1024, 3000)	4809	14.84	8.36e-14	1.20e-2	5264	**13.08**	6.30e-14	3.70e-2
(1024, 4000)	5144	31.09	8.35e-14	1.28e-2	5635	30.02	8.12e-14	4.26e-2
(2048, 5000)	4384	42.45	8.40e-14	1.08e-2	4349	38.08	8.21e-14	2.66e-2
(2048, 6000)	4583	65.08	8.34e-14	1.20e-2	4489	56.54	8.20e-14	3.07e-2
(3000, 7000)	4365	84.15	8.39e-14	1.17e-2	4209	73.20	8.24e-14	2.60e-2
(3000, 8000)	–	247.04	1.04e-14	1.10e-2	5932	136.45	8.25e-14	2.58e-2
(4000, 9000)	4385	130.59	8.38e-14	1.11e-2	4151	109.92	8.27e-14	2.69e-2
(4000, 10000)	–	359.94	1.03e-13	1.03e-2	–	340.95	9.74e-14	2.51e-2
SADMM								
(1024, 3000)	–	26.95	5.72e-15	1.20e-2	–	26.64	5.76e-15	3.70e-2
(1024, 4000)	–	34.54	5.34e-15	1.28e-2	–	32.40	5.33e-15	4.26e-2
(2048, 5000)	–	114.23	7.54e-15	1.08e-2	–	109.13	7.58e-15	2.66e-2
(2048, 6000)	–	130.03	7.64e-15	1.20e-2	–	121.74	7.61e-15	3.07e-2
(3000, 7000)	–	243.01	9.34e-15	1.17e-2	–	232.76	9.42e-15	2.60e-2
(3000, 8000)	–	266.43	9.21e-15	1.10e-2	–	253.38	9.12e-15	2.58e-2
(4000, 9000)	–	420.45	1.58e-14	1.11e-2	–	407.61	1.62e-14	2.69e-2
(4000, 10000)	–	450.26	1.55e-14	1.03e-2	–	436.87	1.58e-14	2.51e-2

results, where the problem size varies from 3000 to 10000 with respect to the dimension of the signal, the regularization parameter is fixed as $\mu = 0.01\mu_{\max}$ and all algorithms are terminated when the tolerance $\epsilon = 10^{-15}$ or maximum iteration number 10000 is reached. Fig. 4.3 shows the comparison between the original signal and the reconstructed signal when the signal dimension is $m = 9000$.

First of all, it can be seen from Table 4.2 that all algorithms are feasible for solving both the nonconvex and convex sparse signal recovery problems, but the proposed algorithm performs better than others with a higher tolerance. However, under a smaller tolerance such as $\epsilon = 10^{-5}$, the SADMM would perform better than Algorithm 1.1 in terms of CPU. Besides, we observe that $l_{1/2}$-regularizer performs significantly better than l_1-regularizer in signal recovery, which could be checked from the results of the iteration number, the CPU time and the recovery quality (i.e. l_2 error). In terms of BADMM, the symbol "$-$" means that the maximum iteration number of 10000 is reached, which implies BADMM performs the worst for solving large-scale problem.

Finally, we would apply the proposed algorithm to solve the direction-of-arrival (DOA) estimation problem [32] with a single snapshot. Here we consider a uniform linear array of $M = 100$ sensors with half-wavelength inter-element spacing. Let $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_L]^\mathsf{T}$ denote the L angles of interest in $[-\pi/2, \pi/2]$. Denote $\mathbf{x} = [x_1, \cdots, x_L]^\mathsf{T}$ as the amplitudes of the potential signals from the L incoming angles. Thus, the received signal at the sensor array is given by: $\mathbf{y} = A\mathbf{x} + \boldsymbol{n}$, where $\mathbf{y} = [y_1, \cdots, y_M]^\mathsf{T}$, $\boldsymbol{n} = [n_1, \cdots, n_M]^\mathsf{T}$, the steering matrix $A = [\mathbf{a}(\theta_1), \cdots, \mathbf{a}(\theta_L)]$ and $\mathbf{a}(\theta_l) = [1, \exp(-j\pi \sin(\theta_l)), \cdots, \exp(-j\pi(M-1)\sin(\theta_l))]^\mathsf{T}$.

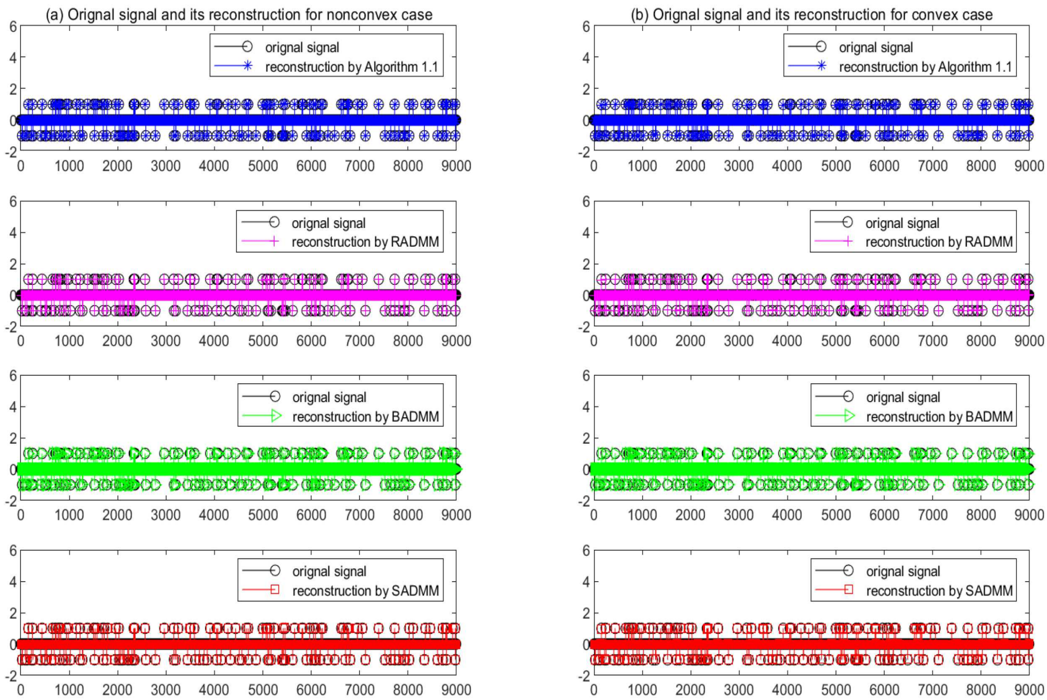

Fig. 4.3. Original signal and reconstructed signal by different algorithms for solving the sparse signal recovery problem with $(l, m) = (4000, 9000)$: the left is the case with $l_{1/2}$-norm and the right is the case with l_1-norm.

We consider the narrowband scenario with $K = 2$ uncorrelated far-field source signals with normalized DOA parameters $-\pi/6$ and $\pi/4$. To run the proposed method, we divide the potential angle region $[-\pi/2, \pi/2]$ into $L = 180$ uniform grid points, i.e.

$$\boldsymbol{\theta} = \frac{\pi}{180}[-90, -89, \ldots, 89, 90]^{\mathsf{T}}.$$

When the signal-to-noise-ratio (SNR) varies from -5 dB to 20 dB, i.e. $\{-5, 0, 5, 10, 15, 20\}$ dB, we implement the proposed Algorithm 1.1 and the well-known CVX toolbox (available at `http://cvxr.com/cvx/`) for 100 Monte Carlo runs, and compute their root mean square errors and running time, as plotted in Fig. 4.4. The CVX is directly used to solve the problem in [32], but Algorithm 1.1 deals with the problem with $l_{1/2}$-regularization. We plot the result of a single trial in the case of 5dB, in Fig. 4.5. From Figs. 4.4 and 4.5, we can see that:

- The accuracy of the two methods increases with SNR.
- The implementation of the proposed method is faster than that of CVX.
- In terms of DOA resolution and the estimation accuracy of the incoming signal power, the proposed method is superior to CVX.

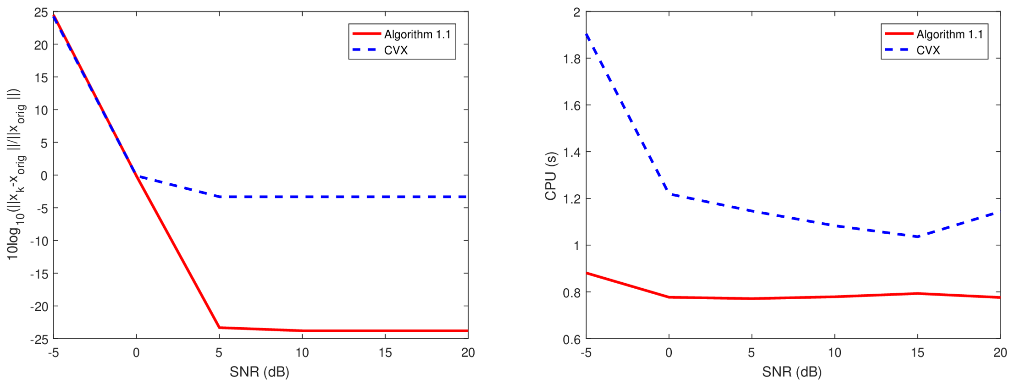

Fig. 4.4. The left and right are the errors and average run time versus SNR.

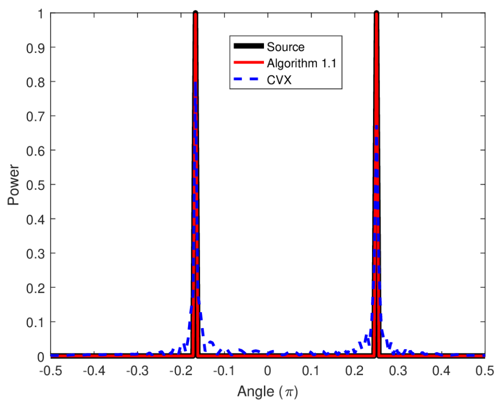

Fig. 4.5. DOAs at SNR= 5dB.

5. Conclusion Remarks

In this paper, we construct a symmetric alternating direction method of multipliers for solving a family of possibly nonconvex and nonxmooth optimization problems. Two different acceleration techniques are designed for fast convergence. Under proper assumptions, convergence of the proposed algorithm as well as its pointwise iteration complexity are analyzed. By testing the so-called sparse signal recovery problem in signal processing with nonconvex/convex regularization terms and by using adaptively updating strategy for the penalty parameter, numerical results demonstrate the feasibility and efficiency of the new algorithm and further show that the $l_{1/2}$-regularization term is better than the l_1-regularization term in terms of CPU time, iteration number and recovery error. Our future work will focus on solving stochastic nonconvex optimization problems by using a similar first-order algorithm to ADMM.

Acknowledgments. This research was partially supported by the National Natural Science Foundation of China (Grant Nos. 12001430, 11801455, 11971238), by the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023A1515012405), by the Shanxi Fundamental Science Research Project for Mathematics and Physics (Grant No. 22JSQ001), by the Sichuan Science and Technology Program (Grant No. 2023NSFSC1922), by the Innovation Team Funds of China West Normal University (Grant No. KCXTD2023-3), by the Fundamental Research Funds of China West Normal University (Grant No. 23kc010), and by the Open Project of Key Laboratory (Grant No. CSSXKFKTM202004), School of Mathematical Sciences, Chongqing Normal University.

References

[1] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality, *Math. Oper. Res.*, **35** (2010), 438–457.

[2] J. Bai, J. Li, and Z. Wu, Several variants of the primal-dual hybrid gradient algorithm with applications, *Numer. Math. Theor. Meth. Appl.*, **13** (2020), 176–199.

[3] J. Bai, J. Li, F. Xu, and H. Zhang, Generalized symmetric ADMM for separable convex optimization, *Comput. Optim. Appl.*, **70** (2018), 129–170.

[4] J. Bai, H. Zhang, and J. Li, A parameterized proximal point algorithm for separable convex optimization, *Optim. Lett.*, **12** (2018), 1589–1608.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.*, **3** (2010), 1–122.

[6] X. Chang, S. Liu, P. Zhao, and D. Song, A generalization of linearized alternating direction method of multipliers for solving two-block separable convex programming, *J. Comput. Appl. Math.*, **357** (2019), 251–272.

[7] R. Chartrand and V. Staneva, Restricted isometry properties and nonconvex compressive sensing, *Inverse Problems*, **24** (2008), 035020.

[8] E. Corman and X. Yuan, A generalized proximal point algorithm and its convergence rate, *SIAM J. Optim.*, **24** (2014), 1614–1638.

[9] J. Douglas and H. Rachford, On the numerical solution of heat conduction problems in two and three space variables, *Trans. Amer. Math. Soc.*, **82** (1956), 421–439.

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

[11] F. Fang, B. He, H. Liu, and X. Yuan, Generalized alternating direction method of multipliers: New theoretical insights and applications, *Math. Program. Comput.*, **7** (2015), 149–187.

[12] M. Figueiredo, R. Nowak, and S. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, *IEEE J. Sel. Top. Signal Process.*, **1** (2007), 586–597.

[13] D. Gabay, Applications of the method of multipliers to variational inequalities, in: *Studies in Mathematics and Its Applications*, **15** (1983), 299–331.

[14] R. Glowinski, *Lectures on Numerical Methods for Non-Linear Variational Problems*, Tata Institute of Fundamental Research, Springer-Verlag, 1980.

[15] R. Glowinski, On alternating direction methods of multipliers: A historical perspective, in: *Modeling, Simulation and Optimization for Science and Technology. Computational Methods in Applied Sciences*, **34** (2014), 59–82.

[16] R. Glowinski, T. Karkkainen, and K. Majava, On the convergence of operator-splitting methods, in: *Numerical Methods for Scientific Computing. Variational Problems and Applications*, CIMNE, (2003), 67–79.

[17] M. Goncalves, J. Melo, and R. Monteiro, Convergence rate bounds for a proximal ADMM with over-relaxation stepsize parameter for solving nonconvex linearly constrained problems, *Pacific J. Optim.*, **15** (2019), 379–398.

[18] K. Guo, D. Han, D. Wang, and T. Wu, Convergence of ADMM for multi-block nonconvex separable optimization models, *Front. Math. China*, **12** (2017), 1139–1162.

[19] K. Guo, D. Han, and T. Wu, Convergence of alternating direction method for minimizing sum of two nonconvex functions with linear constraints, *Int. J. Comput. Math.*, **94** (2017), 1653–1669.

[20] K. Guo, D. Han, and T. Wu, Convergence of ADMM for optimization problems with nonseparable nonconvex objective and linear constraints, *Pacific J. Optim.*, **14** (2018), 489–506.

[21] D. Han, A survey on some recent developments of alternating direction method of multipliers, *J. Oper. Res. Soc. China*, **10** (2022), 1–52.

[22] B. He, H. Liu, Z. Wang, and X. Yuan, A strictly contractive Peaceman-Rachford splitting method for convex programming, *SIAM J. Optim.*, **24** (2014), 1011–1040.

[23] B. He, F. Ma, and X. Yuan, Convergence study on the symmetric version of ADMM with larger step sizes, *SIAM J. Imaging Sci.*, **9** (2016), 1467–1501.

[24] B. He, H. Yang, and S. Wang, Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities, *J. Optim. Theory Appl.*, **106** (2000), 337–356.

[25] M. Hong, Z. Luo, and M. Razaviyay, Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems, *SIAM J. Optim.*, **26** (2016), 337–364.

[26] J. Jian, Y. Zhang, and M. Chao, A regularized alternating direction method of multipliers for a class of nonconvex problems, *J. Inequal. Appl.*, **193** (2019), 1–16.

[27] B. Jiang, T. Lin, S. Ma, and S. Zhang, Structured nonconvex and nonsmooth optimization: Algorithms and iteration complexity analysis, *Comput. Optim. Appl.*, **72** (2019), 115–157.

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

[29] G. Li and T. Pong, Global convergence of splitting methods for nonconvex composite optimization, *SIAM J. Optim.*, **25** (2015), 2434–2460.

[30] P. Lions and B. Mercier, Splitting algorithms for the sum of two nonlinear operators, *SIAM J. Numer. Anal.*, **16** (1979), 964–979.

[31] Q. Liu, X. Shen, and Y. Gu, Linearized ADMM for nonconvex non-smooth optimization with convergence analysis, *IEEE Access*, **7** (2019), 2169–3536.

[32] D. Malioutov, M. Cetin, and A. Willsky, A sparse signal reconstruction perspective for source localization with sensor arrays, *IEEE Trans. Signal Process.*, **53** (2005), 3010–3022.

[33] B. Mordukhovich, Variational Analysis and Generalized Differentiation I: Basic Theory, in: *Grundlehren der Mathematischen Wissenschaften*, Springer-Verlag, 2006.

[34] Y. Nesterov, A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$, *Soviet Mathematics Doklady*, **27** (1983), 372–376.

[35] D. Peaceman and H. Rachford, The numerical solution of parabolic elliptic differential equations, *SIAM J. Appl. Math.*, **3** (1955), 28–41.

[36] R. Rockafellar and R. Wets, *Variational Analysis*, Springer-Verlag, 1998.

[37] T. Sun and C. Zhang, Sparse matrix inversion with scaled Lasso, *J. Mach. Learn. Res.*, **14** (2013), 3385–3418.

[38] F. Wang, W. Cao, and Z. Xu, Convergence of multi-block Bregman ADMM for nonconvex composite problems, *Sci. China Inf. Sci.*, **61** (2018), 122101.

[39] F. Wang, Z. Xu, and H. Xu, Convergence of Bregman alternating direction method with multipliers for nonconvex composite problems, *arXiv:1410.8625v3*, 2014.

[40] W. Wang, Y. Yin, and J. Zeng, Global convergence of ADMM in nonconvex nonsmooth optimization, *J. Sci. Comput.*, **78** (2019), 29–63.

[41] Z. Wu, M. Li, D. Wang, and D. Han, A symmetric alternating direction method of multipliers for separable nonconvex minimization problems, *Asia-Pac. J. Oper. Res.*, **34** (2017), 1750030.

[42] Z. Xu, X. Chang, F. Xu, and H. Zhang, $L_{1/2}$ regularization: A thresholding representation theory and a fast solver, *IEEE Trans. Neural Netw. Learn. Syst.*, **23** (2012), 1013–1027.

[43] L. Yang, T.K. Pong, and X. Chen, Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction, *SIAM J. Imaging Sci.*, **10** (2017), 74–110.

[44] M. Yashtini, Convergence and rate analysis of a proximal linearized ADMM for nonconvex nonsmooth optimization, *J. Global Optim.*, **84** (2022), 913–939.

[45] M. Zhu and T. Chan, *An Efficient Primal-Dual Hybrid Gradient Algorithm for Total Variation Image Restoration*, CAM Report 08-34, UCLA, 2008.