A proximal-perturbed Bregman ADMM for solving nonsmooth and nonconvex composite optimization *

Jianchao Bai † Xu Cui ‡ Zhie Wu §

Abstract. In this paper, we focus on a linearly constrained composite minimization problem involving a possibly nonsmooth and nonconvex objective function. Unlike the traditional construction of the augmented Lagrangian function, we design a proximal-perturbed augmented Lagrangian to develop a new Bregman-type alternating direction method of multipliers. Under mild assumptions, we prove that the augmented Lagrangian sequence converges to the limit of the objective function sequence, and the iterative sequence generated by our method converges to a stationary point of the problem. The sublinear convergence rate of the primal residuals is also analyzed. Comparative experiments on testing the linear equation problem, graph-guided fused lasso problem and robust principal component analysis problem demonstrate the efficiency and flexibility of the proposed method.

Keywords: nonconvex optimization, ADMM, Bregman distance, convergence complexity

Mathematics Subject Classification(2010): 65Y20; 90C26

1 Introduction

Let $\mathbb{R}, \mathbb{R}^n$ and $\mathbb{R}^{p\times n}$ be the sets of real numbers, n dimensional real column vectors, and $p\times n$ real matrices, respectively. Let $\mathbb{R}_+$ and $\mathbb{R}_{++}$ be the sets of non-negative and positive real numbers, respectively. The n-simplex set is defined as $\{\mathbf{x} \mid \sum_{i=1}^n x_i = 1, \mathbf{x} = (x_1, \cdots, x_n) \in \mathbb{R}^n\}$ and the notation $\mathbf{I}$ denotes the identity matrix with proper dimension. The symbol $\nabla f(\mathbf{x})$ represents the gradient of a differentiable function f at $\mathbf{x}$, while the symbols $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ stand for the standard Euclidean norm and inner product, respectively. In this article, we aim to develop an efficient first-order method for solving the following potentially nonsmooth and nonconvex composite minimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n,\mathbf{y}\in\mathbb{R}^m} F(\mathbf{x},\mathbf{y}) := f_1(\mathbf{x}) + f_2(\mathbf{x}) + g_1(\mathbf{y}) + g_2(\mathbf{y}) \qquad \text{s.t.} \quad A\mathbf{x} + B\mathbf{y} = \mathbf{b}, \tag{1.1}$$

*Authors are ranked in alphabetical order with equal contributions. This work was partially supported by the National Natural Science Foundation of China (12471298), the Shaanxi Fundamental Science Research Project for Mathematics and Physics (23JSQ031), and the Open Project Program of Center for Applied Mathematics of Fujian Province (S20250605).

†Corresponding author. School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710129, China; Center for Applied Mathematics of Fujian Province, Fuzhou University, Fuzhou 350116, China (jianchaobai@nwpu.edu.cn)

‡Corresponding author. School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710129, China (cxu@mail.nwpu.edu.cn)

§Zhen'an Branch, State Grid Shaanxi Electric Power Compant Limited, Zhen'an 711500, China (wze92815@163.com)

where $f_1 : \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ and $g_1 : \mathbb{R}^m \to \mathbb{R}\cup\{+\infty\}$ are continuously differentiable functions (possibly nonconvex) with L_f-Lipschitz gradient and L_g-Lipschitz gradient, respectively, f_2 and g_2 are proper lower semicontinuous functions (possibly nonsmooth), $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times m}$ and $b \in \mathbb{R}^p$ are given. Problems adhering to the form of (1.1) arise in various scientific and engineering fields. Here, we take two examples that can be reformulated into the form of (1.1):

▷ *Example 1.* Consider the nonnegative minimal norm solution to the linear equation $A\mathbf{x} + B\mathbf{y} = \mathbf{b}$, where $\mathbf{y} \in [\mathbf{0}, \mathbf{d}]$ and $\mathbf{d}$ is a given nonzero vector. This problem can be modeled as the following linearly constrained minimization problem

$$\min \frac{1}{2}\|\mathbf{x}\|^2 + \frac{\rho}{2}\|\mathbf{y}\|^2 \qquad \text{s.t.} \quad A\mathbf{x} + B\mathbf{y} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{y} \in [\mathbf{0}, \mathbf{d}]. \tag{1.2}$$

By introducing the indicator functions of the set $\{\mathbf{x}|\ \mathbf{x} \geq \mathbf{0}\}$ and $\{\mathbf{y}|\ \mathbf{y} \in [\mathbf{0}, \mathbf{d}]\}$, the above problem is converted into the case of (1.1) with

$$\begin{cases} f_1(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2, \\ g_1(\mathbf{y}) = \frac{\rho}{2}\|\mathbf{y}\|^2, \end{cases} \quad f_2(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \geq \mathbf{0}, \\ +\infty, & \text{otherwise}, \end{cases} \quad g_2(\mathbf{y}) = \begin{cases} 0, & \text{if } \mathbf{y} \in [\mathbf{0}, \mathbf{d}], \\ +\infty, & \text{otherwise}. \end{cases}$$

▷ *Example 2.* Consider the graph-guided fused lasso problem

$$\min \frac{1}{N}\sum_{j=1}^{N} f_j(\mathbf{x}) + \rho_1\|A\mathbf{x}\|_1 + \frac{\rho_2}{2}\|\mathbf{x}\|^2, \tag{1.3}$$

where $f_j(\mathbf{x}) = \frac{1}{1+\exp(b_j a_j^{\mathsf{T}}\mathbf{x})}$ is the empirical loss on the feature-label pair $(a_j, b_j) \in \mathbb{R}^l \times \{-1, 1\}$, N is the data size, and ρ_1, ρ_2 are given regularization parameters. By introducing an auxiliary variable $\mathbf{y} = A\mathbf{x}$, this problem can be regarded as the case of (1.1) with $(B, \mathbf{b}) = (-\mathbf{I}, \mathbf{0})$, $f_1(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N} f_j(\mathbf{x})$, $f_2(\mathbf{x}) = \frac{\rho_2}{2}\|\mathbf{x}\|^2$, and $g_2(\mathbf{y}) = \rho_1\|\mathbf{y}\|_1$.

Beyond these examples, problem (1.1) also arises in compressed sensing [42], subspace clustering [46] and so forth. Throughout this paper, the solution set of (1.1) is assumed to be nonempty, and the proximity operators of f_2 and g_2 can be efficiently evaluated.

A benchmark method for solving linearly constrained minimization problems is the Augmented Lagrangian Method (ALM) proposed by Hestenes [22] and Powell [35]. When applying ALM to the problem (1.1), it proceeds via the following recursive iterations:

$$\begin{cases} (\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) = \arg\min\limits_{\mathbf{x}\in\mathbb{R}^n, \mathbf{y}\in\mathbb{R}^m} \mathcal{L}_\beta(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}_k), \\ \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k - \beta\big(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - \mathbf{b}\big), \end{cases}$$

where

$$\mathcal{L}_\beta(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) = \underbrace{F(\mathbf{x}, \mathbf{y}) + \big\langle \boldsymbol{\lambda}, A\mathbf{x} + B\mathbf{y} - \mathbf{b}\big\rangle}_{L(\mathbf{x},\mathbf{y},\boldsymbol{\lambda})} + \frac{\beta}{2}\big\|A\mathbf{x} + B\mathbf{y} - \mathbf{b}\big\|^2 \tag{1.4}$$

denotes the standard augmented Lagrangian function of (1.1), $\boldsymbol{\lambda}$ denotes the Lagrange multiplier, and $\beta > 0$ is the penalty parameter for the equality constraints.

As a first-order method, ALM has attracted increasing attention due to its diverse applications in signal/image processing, statistical learning, machine learning, and so on. Many

existing ALM-type methods were developed based on the classical augmented Lagrangian function, such as exact/inexact accelerated ALM [3, 24, 25, 30, 43] and stochastic ALM [4, 29] for solving equality constrained convex optimization problems, proximal ALM [28] for solving nonconvex optimization problems, and splitting versions of ALM [19, 20, 38] for solving multi-block separable structured minimization problems. Recently, a double-proximal ALM [6] with convergence guarantees was developed and shown to be efficient for several machine learning problems. Work closely related to [6] includes the balanced ALM [21] and penalty dual-primal ALM [36]. More recently, by introducing an auxiliary variable for (1.1), a new ALM was developed by Kim [27] based on a proximal-perturbed augmented Lagrangian function, and this method was subsequently extended to tackle a broader class of nonconvex optimization problems with nonlinear equality constraints [26]. One effective approach to establish the global convergence and sublinear convergence rate of ALM for convex minimization problems is to use variational analysis to characterize both the saddle-point and the iterative sequence, c.f. [4, 6, 36]. However, a practical technique to establish the convergence of ALM for nonconvex optimization problems is to construct a potential function related to the associated Lagrange function and then demonstrate the convergence by showing the monotonic decreasing property of this potential function, c.f. [5, 25, 26, 31] to list a few.

When the objective function of optimization problems has composite structures such as (1.1), the standard ALM cannot fully utilize these structures and hence cannot take full advantage of the special properties of each component objective function. Consequently, solving the involved subproblems becomes very difficult. An effective and practical approach to overcoming such difficulty is the Alternating Direction Method of Multipliers (ADMM). For example, Barber, et al. [8] developed a proximal ADMM with general penalty matrix and established its convergence under the restricted strong convexity; Wang-Cai-Chen [40] proposed a globally convergent preconditioned ADMM for solving (1.1) with $(f_1, g_1) = (0, 0)$ and f_2, g_2 being convex functions; Wang-Banerjee [39] extended the standard ADMM to Bregman ADMM:

$$\begin{cases} \mathbf{x}_{k+1} = \arg\min\limits_{\mathbf{x}\in\mathbb{R}^n} \big\{\mathcal{L}_\beta(\mathbf{x}, \mathbf{y}_k, \boldsymbol{\lambda}_k) + \mathcal{B}_{\phi_1}\big(\mathbf{x}, \mathbf{x}_k\big)\big\}, \\ \mathbf{y}_{k+1} = \arg\min\limits_{\mathbf{y}\in\mathbb{R}^m} \big\{\mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}, \boldsymbol{\lambda}_k) + \mathcal{B}_{\phi_2}\big(\mathbf{y}, \mathbf{y}_k\big)\big\}, \\ \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k - \beta\big(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - \mathbf{b}\big). \end{cases} \tag{1.5}$$

Here $\mathcal{B}_{\phi_i}$ represents the Bregman distance [9] defined as

$$\mathcal{B}_{\phi_i}\big(\mathbf{u}, \mathbf{v}\big) := \phi_i(\mathbf{u}) - \phi_i(\mathbf{v}) - \big\langle\nabla\phi_i(\mathbf{v}), \mathbf{u} - \mathbf{v}\big\rangle, \quad i = 1, 2.$$

Since $\phi_i(\cdot)$ is differentiable and convex, it is easy to check that $\mathcal{B}_{\phi_i}\big(\mathbf{u}, \mathbf{v}\big)$ is nonnegative and $\mathcal{B}_{\phi_i}\big(\mathbf{u}, \mathbf{v}\big) \geq \frac{\theta_i}{2}\big\|\mathbf{u} - \mathbf{v}\big\|^2$ if $\phi_i(\cdot)$ is strongly convex with modulus $\theta_i > 0$.

In fact, the Bregman distance includes a large number of useful loss functions such as logistic loss, Euclidean distance and KL-divergence, see Table 1. This makes ADMM more general and more flexible, allowing the resulting subproblems to be solved efficiently or even have a closed-form solution. As demonstrated in [34, Proposition 3.5], the linearized proximal ADMM is an instance of the Bregman ADMM when the distance generating functions ϕ_1 and ϕ_2 are properly chosen. Hence, the Bregman ADMM contains various variants of ADMM, including the classical ADMM, proximal ADMM, and linearized proximal ADMM. Besides, by a proper choice of the Bregman distance, the resulting subproblems can be simplified as proximity operators. For instance, if $\phi_1 = \frac{\nu}{2}\|\mathbf{x}\|^2 - \frac{\beta}{2}\|A\mathbf{x}\|^2$ with $\nu > \beta\|A^{\mathsf{T}}A\|$, then the first subproblem

Table 1: Several special instances of Bregman distance [1].

Domain	$\phi_i(\mathbf{x})$	$\mathcal{B}_{\phi_i}(\mathbf{x},\mathbf{v})$	Appellation
$\mathbb{R}$	x^2	$(x-v)^2$	Squared loss
$\mathbb{R}_+$	$x\log x$	$x\log(\frac{x}{v})-(x-v)$	
$[0,1]$	$x\log x+(1-x)\log(1-x)$	$x\log(\frac{x}{v})+(1-x)\log(\frac{1-x}{1-v})$	Logistic loss
$\mathbb{R}_{++}$	$-\log x$	$\frac{x}{v}-\log(\frac{x}{v})-1$	Itakura-Saito distance
$\mathbb{R}^n$	$\|\mathbf{x}\|^2$	$\|\mathbf{x}-\mathbf{v}\|^2$	Squared Euclidean distance
$\mathbb{R}^n$	$\mathbf{x}^\mathsf{T}A\mathbf{x}$	$(\mathbf{x}-\mathbf{v})^\mathsf{T}A(\mathbf{x}-\mathbf{v})$	Mahalanobis distance
n-Simplex	$\sum_{j=1}^n x_j\log(x_j)$	$\sum_{j=1}^n x_j\log(\frac{x_j}{v_j})$	KL-divergence

in (1.5) can be simplified as the proximity operator of f_1 with known $\mathbf{r}_k$:

$$\mathbf{prox}_{\nu f_1}(\mathbf{r_k}) = \arg\min\{f_1(\mathbf{x}) + \frac{\nu}{2}\|\mathbf{x}-\mathbf{r}_k\|^2\},$$

while the subproblem in the standard ADMM can not be converted to the above form. For more examples on the Bregman proximal step that admits a closed-form solution, we refer to [2, Example 3]. Recently, Chen, et al. [13] demonstrated the convergence of the directly extended ADMM for solving the three-block separable optimization problem whose objective function is the sum of one weakly convex and two strongly convex functions. An efficient Bregman-style ADMM [32] was also proposed for solving the problem (1.1) with $(f_2, g_1) = (0, 0)$. In order to take advantage of Bregman distance, an interesting question is: can we construct a distinct augmented Lagrangian function so as to develop a new Bregman-based ADMM for the general nonconvex and nonsmooth minimization problem (1.1)?

In this paper, motivated by the above question, we will propose a new ADMM-type method based on the novel augmented Lagrangian constructed in [26]. We further establish the convergence of the proposed method, with respect to both the corresponding augmented Lagrangian sequence and the iterative residuals for primal variables and constraint violations. Key features of our method are summarized in the subsequent section.

2 Development of 2P-ADMM

Inspired by the new Lagrangian-based first-order method [26, 27, 32], by introducing a similar perturbation variable $\mathbf{z} \in \mathbb{R}^p$ such that $\mathbf{z} = \mathbf{0}$ where bold $\mathbf{0}$ denotes the zero vector, we reformulate the problem (1.1) as the following double-constrained problem

$$\min_{\mathbf{x}\in\mathbb{R}^n,\mathbf{y}\in\mathbb{R}^m,\mathbf{z}\in\mathbb{R}^p} F(\mathbf{x},\mathbf{y}) \qquad \text{s.t.} \;\; A\mathbf{x}+B\mathbf{y}-\mathbf{b}=\mathbf{z},\;\; \mathbf{z}=\mathbf{0}. \tag{2.1}$$

Define the proximal-perturbed augmented Lagrangian of (2.1) as

$$\mathcal{L}_\beta(\mathbf{x},\mathbf{y},\boldsymbol{\lambda},\mathbf{z},\boldsymbol{\mu}) = F(\mathbf{x},\mathbf{y}) + \langle\boldsymbol{\lambda}, A\mathbf{x}+B\mathbf{y}-\mathbf{b}-\mathbf{z}\rangle + \langle\boldsymbol{\mu},\mathbf{z}\rangle + \frac{\alpha}{2}\|\mathbf{z}\|^2 - \frac{\sigma}{2}\|\boldsymbol{\lambda}-\boldsymbol{\mu}\|^2, \tag{2.2}$$

where $\boldsymbol{\lambda},\boldsymbol{\mu}\in\mathbb{R}^p$ are the Lagrange multipliers associated with the equality constraints, $\alpha>0$ is a penalty parameter, and $\sigma>0$ denotes a proximal parameter.

To predigest discussions, we simply denote $\mathcal{L}_\beta(\mathbf{x},\mathbf{y},\boldsymbol{\lambda},\mathbf{z},\boldsymbol{\mu})$ by $\mathcal{L}_\beta(\mathbf{w})$ with $\mathbf{w}=(\mathbf{x},\mathbf{y},\boldsymbol{\lambda},\mathbf{z},\boldsymbol{\mu})$. Comments on this new proximal-perturbed augmented Lagrangian function are given below:

(i) Unlike the standard augmented Lagrangian (1.4), we exploit a proximal term $\frac{\sigma}{2}\|\boldsymbol{\lambda}-\boldsymbol{\mu}\|^2$ in (2.2), instead of the widely-used quadratic penalty for the constraint $A\mathbf{x}+B\mathbf{y}-\mathbf{b}=\mathbf{z}$, to ensure the strong concavity of $\mathcal{L}_\beta(\mathbf{w})$ w.r.t. the Lagrange multipliers $\boldsymbol{\lambda}$ (for fixed $\boldsymbol{\mu}$) and $\boldsymbol{\mu}$ (for fixed $\boldsymbol{\lambda}$). This is helpful for simplifying the update of Lagrange multipliers. Besides, minimizing $\mathcal{L}_\beta(\mathbf{w})$ w.r.t. each primal variable can exploit the proximity operator of $f_2(\mathbf{x})$ or $g_2(\mathbf{y})$, when adding a customized Bregman distance as the proximal term;

(ii) Because $\mathcal{L}_\beta(\mathbf{w})$ is smooth and strongly convex about $\mathbf{z}$, there exists a unique solution for given $(\boldsymbol{\lambda},\boldsymbol{\mu})$. More specifically, by minimizing $\mathcal{L}_\beta(\mathbf{w})$ w.r.t. $\mathbf{z}$, we can derive

$$\mathbf{z}(\boldsymbol{\lambda},\boldsymbol{\mu})=\frac{\boldsymbol{\lambda}-\boldsymbol{\mu}}{\alpha}, \tag{2.3}$$

which implies $\boldsymbol{\lambda}=\boldsymbol{\mu}$ at the unique solution $\mathbf{z}^*=\mathbf{0}$. By the relationship in (2.3), we thus add the smoothing proximal term $-\frac{\beta}{2}\|\boldsymbol{\lambda}-\boldsymbol{\mu}\|^2$ to the Lagrangian in (2.2).

Now, plugging the certain relationship (2.3) into (2.2) results in

$$\mathcal{L}_\beta(\mathbf{w})=L(\mathbf{x},\mathbf{y},\boldsymbol{\lambda})-\frac{1}{2\beta}\big\|\boldsymbol{\lambda}-\boldsymbol{\mu}\big\|^2 \tag{2.4}$$

with $\beta=\frac{\alpha}{1+\alpha\sigma}$. Clearly, the function $\mathcal{L}_\beta(\mathbf{w})$ is strongly concave about $\boldsymbol{\lambda}$ for given $(\mathbf{x},\mathbf{y},\boldsymbol{\mu})$. So there exists a unique maximizer, denoted by $\boldsymbol{\lambda}(\mathbf{x},\mathbf{y},\boldsymbol{\mu})$, namely,

$$\boldsymbol{\lambda}(\mathbf{x},\mathbf{y},\boldsymbol{\mu})=\arg\max_{\boldsymbol{\lambda}\in\mathbb{R}^p}\mathcal{L}_\beta(\mathbf{w})=\boldsymbol{\mu}+\beta(A\mathbf{x}+B\mathbf{y}-\mathbf{b}).$$

Notice that directly minimizing $\mathcal{L}_\beta(\mathbf{w})$ about the primal variables $\mathbf{x}$ and $\mathbf{y}$ is still challenging, since it does not make full use of each nonsmooth objective function as well as the separable structure of the problem. To tackle these obstacles, we first employ an approximation to $\mathcal{L}_\beta(\mathbf{w})$ as follows:

$$\begin{aligned}\tilde{\mathcal{L}}_\beta(\mathbf{w},\mathbf{v}_1,\mathbf{v}_2):=\ & f_2(\mathbf{x})+g_2(\mathbf{y})+\mathcal{B}_{\phi_1}\big(\mathbf{x},\mathbf{v}_1\big)+\mathcal{B}_{\phi_2}\big(\mathbf{y},\mathbf{v}_2\big)\\ &+\bar{\mathcal{L}}_\beta(\mathbf{w})+\big\langle\nabla_{\mathbf{x}}\bar{\mathcal{L}}_\beta(\mathbf{w}),\mathbf{x}-\mathbf{v}_1\big\rangle+\big\langle\nabla_{\mathbf{y}}\bar{\mathcal{L}}_\beta(\mathbf{w}),\mathbf{y}-\mathbf{v}_2\big\rangle,\end{aligned}$$

where $\bar{\mathcal{L}}_\beta(\mathbf{w})$ is the smooth part of $\mathcal{L}_\beta(\mathbf{w})$, i.e.

$$\bar{\mathcal{L}}_\beta(\mathbf{w})=f_1(\mathbf{x})+g_1(\mathbf{y})+\big\langle\boldsymbol{\lambda},A\mathbf{x}+B\mathbf{y}-\mathbf{b}-\mathbf{z}\big\rangle+\big\langle\boldsymbol{\mu},\mathbf{z}\big\rangle+\frac{\alpha}{2}\big\|\mathbf{z}\big\|^2-\frac{\sigma}{2}\big\|\boldsymbol{\lambda}-\boldsymbol{\mu}\big\|^2.$$

Based on the above preparations and the splitting idea with respect to primal variables $\mathbf{x}$ and $\mathbf{y}$, we propose a customized Proximal-Perturbed ADMM (2P-ADMM) whose framework is described in Algorithm 2.1. In fact, both $\mathbf{x}$-subproblem and $\mathbf{y}$-subproblem update in parallel since they can be simplified as

$$\begin{cases}\mathbf{x}_{k+1}=\arg\min\limits_{\mathbf{x}\in\mathbb{R}^n}\Big\{f_2(\mathbf{x})+\big\langle\mathbf{x}-\mathbf{x}_k,\nabla f_1(\mathbf{x}_k)+A^{\mathsf{T}}\boldsymbol{\lambda}_k\big\rangle+\mathcal{B}_{\phi_1}\big(\mathbf{x},\mathbf{x}_k\big)\Big\},\\ \mathbf{y}_{k+1}=\arg\min\limits_{\mathbf{y}\in\mathbb{R}^m}\Big\{g_2(\mathbf{y})+\big\langle\mathbf{y}-\mathbf{y}_k,\nabla g_1(\mathbf{y}_k)+B^{\mathsf{T}}\boldsymbol{\lambda}_k\big\rangle+\mathcal{B}_{\phi_2}\big(\mathbf{y},\mathbf{y}_k\big)\Big\}.\end{cases} \tag{2.5}$$

Hence, the fifth step regarding $\mathbf{z}_{k+1}$ does not work and can be removed when carrying out experiments. Compared to the proximal term (i.e., squared Euclidean distance) in the proximal-perturbed ALM [27], our proximal term in (2.5) is more general and 2P-ADMM can be extremely effective by properly choosing ϕ_i (see the sequel Figure 2 in experiments). The updating

Input: $\alpha \gg 1, \sigma \in (0,1), \beta = \frac{\alpha}{1+\alpha\sigma}, r \in (0.9,1), \theta_1 > L_f$ and $\theta_2 > L_g$.
Initialization: $\mathbf{w}_0 = (\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0, \boldsymbol{\lambda}_0, \boldsymbol{\mu}_0)$ and $\delta_0 \in (0,1]$.
For $k = 0, 1, 2, \cdots$

1. $\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Big\{ f_2(x) + \big\langle \mathbf{x} - \mathbf{x}_k, \nabla_{\mathbf{x}} \bar{\mathcal{L}}_\beta(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_k, \mathbf{z}_k, \boldsymbol{\mu}_k) \big\rangle + \mathcal{B}_{\phi_1}\big(\mathbf{x}, \mathbf{x}_k\big) \Big\}$;
2. $\mathbf{y}_{k+1} = \arg\min_{\mathbf{y}\in\mathbb{R}^m} \Big\{ g_2(\mathbf{y}) + \big\langle \mathbf{y} - \mathbf{y}_k, \nabla_{\mathbf{y}} \bar{\mathcal{L}}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_k, \boldsymbol{\lambda}_k, \mathbf{z}_k, \boldsymbol{\mu}_k) \big\rangle + \mathcal{B}_{\phi_2}\big(\mathbf{y}, \mathbf{y}_k\big) \Big\}$;
3. $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \tau_k\big(\boldsymbol{\lambda}_k - \boldsymbol{\mu}_k\big)$ with $\tau_k = \frac{\delta_k}{1+\|\boldsymbol{\lambda}_k - \boldsymbol{\mu}_k\|^2}$;
4. $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\mu}_{k+1} + \beta\big(A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - \mathbf{b}\big)$;
5. $\mathbf{z}_{k+1} = \frac{\boldsymbol{\lambda}_{k+1} - \boldsymbol{\mu}_{k+1}}{\alpha}$;
6. $\delta_{k+1} = r\delta_k$;

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

Algorithm 2.1: Proximal-Perturbed ADMM (2P-ADMM) for solving (1.1).

formula of δ_{k+1} implies $\delta_k = r^k\delta_0$. So, by the region $r \in (0,1)$ and $\delta_0 \in (0,1]$, we know the sequence $\{\delta_k\}$ is summable. Due to this fact, the choice of τ_k can guarantee the boundedness of $\{\boldsymbol{\mu}_k\}$, which in turn guarantees the boundedness of $\{\boldsymbol{\lambda}_k\}$.

Remark 2.1 *Consider the nonconvex optimization problem in [45], that is, problem* (1.1) *with* $g_2 = 0$, $B = \mathbf{I}$ *and* f_2 *being the indicator function of a nonempty closed convex set* $\mathcal{X}$. *For this type of problem, by selecting* $\phi_2 = \frac{1}{2\gamma}\|\mathbf{y}\|^2$, *our proposed method reduces to Alg. 2.2. Note that the update of* $\mathbf{y}_{k+1}$ *obeys a gradient descent step, and the term* $\nabla g_1(\mathbf{y}_k) + B^{\mathsf{T}}\boldsymbol{\lambda}_k$ *is, in fact, the gradient of the corresponding* $L(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda})$ *at* $\mathbf{y}_k$.

Input: $\alpha \gg 1, \sigma \in (0,1), \beta = \frac{\alpha}{1+\alpha\sigma}, r \in (0.9,1), \theta_1 > L_f$ and $1/\gamma > L_g$.
Initialization: $\mathbf{w}_0 = (\mathbf{x}_0, \mathbf{y}_0, \boldsymbol{\lambda}_0, \boldsymbol{\mu}_0)$ and $\delta_0 \in (0,1]$.
For $k = 0, 1, 2, \cdots$

1. $\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \Big\{ \big\langle \mathbf{x} - \mathbf{x}_k, \nabla f_1(\mathbf{x}_k) + A^{\mathsf{T}}\boldsymbol{\lambda}_k \big\rangle + \mathcal{B}_{\phi_1}\big(\mathbf{x}, \mathbf{x}_k\big) \Big\}$;
2. $\mathbf{y}_{k+1} = \mathbf{y}_k - \gamma\big[\nabla g_1(\mathbf{y}_k) + B^{\mathsf{T}}\boldsymbol{\lambda}_k\big]$;
3. $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \tau_k\big(\boldsymbol{\lambda}_k - \boldsymbol{\mu}_k\big)$ with $\tau_k = \frac{\delta_k}{1+\|\boldsymbol{\lambda}_k - \boldsymbol{\mu}_k\|^2}$;
4. $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\mu}_{k+1} + \beta\big(A\mathbf{x}_{k+1} + \mathbf{y}_{k+1} - \mathbf{b}\big)$;
5. $\delta_{k+1} = r\delta_k$;

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

Algorithm 2.2: A special case of the proximal-perturbed ADMM.

3 Convergence analysis

3.1 Technical preliminarily

In this subsection, we present several lemmas that will be used to analyze the convergence of both the augmented Lagrangian sequence given in (2.4) and the iterative sequence. Throughout this paper, similar to [18], we make the following assumptions:

(A1) $\bar{f}_1 = \inf_{\mathbf{x}} \left\{ f_1(\mathbf{x}) - \frac{1}{2L_f} \|\nabla f_1(\mathbf{x})\|^2 \right\} > -\infty$ and $\bar{g}_1 = \inf_{\mathbf{y}} \left\{ g_1(\mathbf{y}) - \frac{1}{2L_g} \|\nabla g_1(\mathbf{y})\|^2 \right\} > -\infty$;

(A2) $\lim_{\|\mathbf{x}\|\to\infty} \inf f_2(\mathbf{x}) = +\infty$ and $\lim_{\|\mathbf{y}\|\to\infty} \inf g_2(\mathbf{y}) = +\infty$.

Lemma 3.1 *The sequences $\{\boldsymbol{\mu}_k\}$ and $\{\boldsymbol{\lambda}_k\}$ generated by Algorithm 2.1 are bounded.*

Proof. By the update of $\boldsymbol{\mu}_{k+1}$, we have

$$\|\boldsymbol{\mu}_{k+1}\| = \Big\|\boldsymbol{\mu}_0 + \sum_{i=0}^{k} \tau_i(\boldsymbol{\lambda}_i - \boldsymbol{\mu}_i)\Big\| \le \|\boldsymbol{\mu}_0\| + \sum_{i=0}^{+\infty} \frac{\delta_i}{\|\boldsymbol{\lambda}_i - \boldsymbol{\mu}_i\|^2 + 1} \|\boldsymbol{\lambda}_i - \boldsymbol{\mu}_i\| \le \|\boldsymbol{\mu}_0\| + \frac{1}{2}\sum_{i=0}^{\infty} \delta_i < +\infty,$$

which shows that the sequence $\{\boldsymbol{\mu}_k\}$ is bounded since $\sum_{i=0}^{\infty} \delta_i$ is convergent where $\delta_{k+1} = r\delta_k$ and $r \in (0.9, 1)$.

Besides, the update of $\boldsymbol{\mu}_{k+1}$ gives

$$\boldsymbol{\lambda}_k = \frac{1}{\tau_k}\boldsymbol{\mu}_{k+1} + \Big(1 - \frac{1}{\tau_k}\Big)\boldsymbol{\mu}_k,$$

which means $\boldsymbol{\lambda}_k$ is a combination of $\boldsymbol{\mu}_{k+1}$ and $\boldsymbol{\mu}_k$. Combine this relationship with the boundedness of $\{\boldsymbol{\mu}_k\}$ to ensure that $\{\boldsymbol{\lambda}_k\}$ is a bounded sequence. ■

The above lemma as well as the following lemma will be used to investigate some properties of the iterative sequence $\{\mathbf{w}_k\}$ generated by Algorithm 2.1.

Lemma 3.2 *Let $\{\boldsymbol{\mu}_k\}$ and $\{\boldsymbol{\lambda}_k\}$ be the sequences generated by Algorithm 2.1. Then, we have*

$$\|\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k\|^2 \le \frac{\delta_k^2}{4}, \qquad \|\boldsymbol{\lambda}_k - \boldsymbol{\mu}_k\|^2 \le \frac{\delta_k}{\tau_k}, \tag{3.1}$$

and

$$\|\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_k\|^2 \le 2\|\boldsymbol{\lambda}_{k+1} - \boldsymbol{\mu}_{k+1}\|^2 + 2\|\boldsymbol{\mu}_{k+1} - \boldsymbol{\lambda}_k\|^2. \tag{3.2}$$

Proof. By the way of updating $\boldsymbol{\mu}_{k+1}$ and $\boldsymbol{\lambda}_{k+1}$, we have

$$\|\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k\|^2 = \tau_k^2\|\boldsymbol{\lambda}_k - \boldsymbol{\mu}_k\|^2 \le \frac{\delta_k^2}{\|\boldsymbol{\lambda}_k - \boldsymbol{\mu}_k\|^2 + 2 + \frac{1}{\|\boldsymbol{\lambda}_k - \boldsymbol{\mu}_k\|^2}} \le \frac{\delta_k^2}{4},$$

where the first inequality uses the definition of τ_k and the last inequality follows from the fact $a + b \ge 2\sqrt{ab}$ for any $a, b \ge 0$. Using the the definition of τ_k again, it holds that

$$\tau_k\|\boldsymbol{\lambda}_k - \boldsymbol{\mu}_k\|^2 = \frac{\delta_k}{1 + \frac{1}{\|\boldsymbol{\lambda}_k - \boldsymbol{\mu}_k\|^2}} \le \delta_k.$$

The result in (3.2) follows directly from the fact $(a + b)^2 \le 2a^2 + 2b^2$ for any a and b. ■

By the update of τ_k, we know $\tau_k \in (0, 1)$ and it is a bounded sequence. Based on its lower bound, next we provide some core properties related to $\{\mathcal{L}_\beta(\mathbf{w}_k)\}$, which further establishes that both the iterative residual and the constraint residual converge to zero.

Theorem 3.1 *Let $\bar{\tau} > 0$ be the lower bound of $\{\tau_k\}$ and $\{\mathbf{w}_k := (\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_k, \mathbf{z}_k, \boldsymbol{\mu}_k)\}$ be the sequence generated by Algorithm 2.1. Then, the following hold:*

(i) The sequence $\{\mathcal{L}_\beta(\mathbf{w}_{k+1})\}$ defined in (2.4) satisfies

$$\mathcal{L}_\beta(\mathbf{w}_{k+1}) \leq \mathcal{L}_\beta(\mathbf{w}_k) - \frac{\theta_1 - L_f}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 - \frac{\theta_2 - L_g}{2}\|\mathbf{y}_{k+1} - \mathbf{y}_k\|^2 + \frac{\delta_{k+1} + \delta_k}{\beta\bar{\tau}}; \quad (3.3)$$

(ii) Under the assumptions (A1)-(A2), the sequence $\{\mathcal{L}_\beta(\mathbf{w}_k)\}$ is convergent. Moreover,

$$\lim_{k\to\infty} \|\mathbf{w}_{k+1} - \mathbf{w}_k\| = 0 \quad \textit{and} \quad \lim_{k\to\infty} \|A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - \mathbf{b}\| = 0. \quad (3.4)$$

Proof. To prove the assertion (i), we split $\mathcal{L}_\beta(\mathbf{w}_{k+1}) - \mathcal{L}_\beta(\mathbf{w}_k)$ into three residuals:

$$\begin{aligned}\mathcal{L}_\beta(\mathbf{w}_{k+1}) - \mathcal{L}_\beta(\mathbf{w}_k) =& \mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_k, \mathbf{z}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k) - \mathcal{L}_\beta(\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k)\\ &+ \mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \mathbf{z}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k) - \mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_k, \mathbf{z}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k)\\ &+ \mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \mathbf{z}_{k+1}, \boldsymbol{\lambda}_{k+1}, \boldsymbol{\mu}_{k+1}) - \mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \mathbf{z}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k).\end{aligned}$$

According to the equivalent expression of the $\mathbf{x}_{k+1}$-subproblem as in (2.5), we have

$$f_2(\mathbf{x}_{k+1}) + \langle \nabla f_1(\mathbf{x}_k) + A^\mathsf{T}\boldsymbol{\lambda}_k, \mathbf{x}_{k+1} - \mathbf{x}_k\rangle + \mathcal{B}_{\phi_1}(\mathbf{x}_{k+1}, \mathbf{x}_k) \leq f_2(\mathbf{x}_k),$$

implying that

$$f_2(\mathbf{x}_{k+1}) - f_2(\mathbf{x}_k) + \langle A^\mathsf{T}\boldsymbol{\lambda}_k, \mathbf{x}_{k+1} - \mathbf{x}_k\rangle \leq \langle \nabla f_1(\mathbf{x}_k), \mathbf{x}_k - \mathbf{x}_{k+1}\rangle - \frac{\theta_1}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2.$$

The last inequality together with the Lipschitz continuity of f_1:

$$f_1(\mathbf{x}_{k+1}) - f_1(\mathbf{x}_k) \leq \langle \nabla f_1(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k\rangle + \frac{L_f}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2$$

gives

$$\begin{aligned}&\mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_k, \mathbf{z}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k) - \mathcal{L}_\beta(\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k) = f_1(\mathbf{x}_{k+1}) - f_1(\mathbf{x}_k)\\ &+ f_2(\mathbf{x}_{k+1}) - f_2(\mathbf{x}_k) + \langle A^\mathsf{T}\boldsymbol{\lambda}_k, \mathbf{x}_{k+1} - \mathbf{x}_k\rangle \leq -\frac{\theta_1 - L_f}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2. \quad (3.5)\end{aligned}$$

Similarly, the Lipschitz continuity of g_1 yields

$$g_1(\mathbf{y}_{k+1}) - g_1(\mathbf{y}_k) \leq \langle \nabla g_1(\mathbf{y}_k), \mathbf{y}_{k+1} - \mathbf{y}_k\rangle + \frac{L_g}{2}\|\mathbf{y}_{k+1} - \mathbf{y}_k\|^2,$$

which, by using the following property from the $\mathbf{y}_{k+1}$-subproblem:

$$g_2(\mathbf{y}_{k+1}) - g_2(\mathbf{x}_k) + \langle B^\mathsf{T}\boldsymbol{\lambda}_k, \mathbf{y}_{k+1} - \mathbf{y}_k\rangle \leq \langle \nabla g_1(\mathbf{y}_k), \mathbf{y}_k - \mathbf{y}_{k+1}\rangle - \frac{\theta_2}{2}\|\mathbf{y}_{k+1} - \mathbf{y}_k\|^2,$$

gives

$$\mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \mathbf{z}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k) - \mathcal{L}_\beta(\mathbf{x}_{k+1}, \mathbf{y}_k, \mathbf{z}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k) \leq -\frac{\theta_2 - L_f}{2}\|\mathbf{y}_{k+1} - \mathbf{y}_k\|^2. \quad (3.6)$$

Notice that

$$\begin{aligned}&\mathcal{L}_\beta(\mathbf{x}_{k+1},\mathbf{y}_{k+1},\mathbf{z}_{k+1},\boldsymbol{\lambda}_{k+1},\boldsymbol{\mu}_{k+1})-\mathcal{L}_\beta(\mathbf{x}_{k+1},\mathbf{y}_{k+1},\mathbf{z}_k,\boldsymbol{\lambda}_k,\boldsymbol{\mu}_k)\\ =&\big\langle\boldsymbol{\lambda}_{k+1}-\boldsymbol{\lambda}_k,A\mathbf{x}_{k+1}+B\mathbf{y}_{k+1}-\mathbf{b}\big\rangle-\frac{1}{2\beta}\big\|\boldsymbol{\lambda}_{k+1}-\boldsymbol{\mu}_{k+1}\big\|^2+\frac{1}{2\beta}\big\|\boldsymbol{\lambda}_k-\boldsymbol{\mu}_k\big\|^2\end{aligned}\tag{3.7}$$

and

$$\big\|\boldsymbol{\lambda}_k-\boldsymbol{\mu}_{k+1}\big\|^2=\big\|\boldsymbol{\lambda}_k-\boldsymbol{\mu}_k+\boldsymbol{\mu}_k-\boldsymbol{\mu}_{k+1}\big\|^2=(1-\tau_k)^2\big\|\boldsymbol{\lambda}_k-\boldsymbol{\mu}_k\big\|^2\le\big\|\boldsymbol{\lambda}_k-\boldsymbol{\mu}_k\big\|^2.\tag{3.8}$$

By using $\boldsymbol{\lambda}_{k+1}-\boldsymbol{\mu}_{k+1}=\beta(A\mathbf{x}_{k+1}+B\mathbf{y}_{k+1}-b)$, $\mathbf{z}_k=\frac{1}{\alpha}(\boldsymbol{\lambda}_k-\boldsymbol{\mu}_k)$, and applying the identity

$$\big\langle a-b,a\big\rangle=\frac{1}{2}\|a-b\|^2+\frac{1}{2}\|a\|^2-\frac{1}{2}\|b\|^2$$

with $(a,b)=(\boldsymbol{\lambda}_{k+1}-\boldsymbol{\mu}_{k+1},\boldsymbol{\lambda}_k-\boldsymbol{\mu}_{k+1})$ to (3.7), we further have

$$\begin{aligned}&\mathcal{L}_\beta(\mathbf{x}_{k+1},\mathbf{y}_{k+1},\mathbf{z}_{k+1},\boldsymbol{\lambda}_{k+1},\boldsymbol{\mu}_{k+1})-\mathcal{L}_\beta(\mathbf{x}_{k+1},\mathbf{y}_{k+1},\mathbf{z}_k,\boldsymbol{\lambda}_k,\boldsymbol{\mu}_k)\\ =&\frac{1}{2\beta}\big\|\boldsymbol{\lambda}_{k+1}-\boldsymbol{\lambda}_k\big\|^2-\frac{1}{2\beta}\big\|\boldsymbol{\mu}_{k+1}-\boldsymbol{\lambda}_k\big\|^2+\frac{1}{2\beta}\big\|\boldsymbol{\lambda}_k-\boldsymbol{\mu}_k\big\|^2\\ \overset{(3.2)}{\le}&\frac{1}{\beta}\big\|\boldsymbol{\lambda}_{k+1}-\boldsymbol{\mu}_{k+1}\big\|^2+\frac{1}{2\beta}\big\|\boldsymbol{\mu}_{k+1}-\boldsymbol{\lambda}_k\big\|^2+\frac{1}{2\beta}\big\|\boldsymbol{\lambda}_k-\boldsymbol{\mu}_k\big\|^2\overset{(3.1),(3.8)}{\le}\frac{\delta_{k+1}+\delta_k}{\beta\bar{\tau}}.\end{aligned}\tag{3.9}$$

So, combining the above inequalities (3.5),(3.6) and (3.9) yields the desired result (3.3).

To prove the result (ii), we first show that $\{\mathbf{w}^k\}$ is bounded. It follows from (3.3) and the conditions $\theta_1>L_f$ and $\theta_2>L_g$ that

$$\begin{aligned}&\mathcal{L}_\beta(\mathbf{w}_0)+\frac{\delta_0}{\beta\bar{\tau}}\sum_{j=0}^{k}\left(r^{j+1}+r^j\right)\ge\mathcal{L}_\beta(\mathbf{w}_k)+\frac{\delta_{k+1}+\delta_k}{\beta\bar{\tau}}\ge\mathcal{L}_\beta(\mathbf{w}_{k+1})\\ =&F(\mathbf{x}_{k+1},\mathbf{y}_{k+1})+\big\langle\boldsymbol{\lambda}_{k+1},A\mathbf{x}_{k+1}+B\mathbf{y}_{k+1}-\mathbf{b}\big\rangle-\frac{1}{2\beta}\big\|\boldsymbol{\lambda}_{k+1}-\boldsymbol{\mu}_{k+1}\big\|^2\\ =&F(\mathbf{x}_{k+1},\mathbf{y}_{k+1})+\frac{1}{\beta}\big\langle\boldsymbol{\lambda}_{k+1},\boldsymbol{\lambda}_{k+1}-\boldsymbol{\mu}_{k+1}\big\rangle-\frac{1}{2\beta}\big\|\boldsymbol{\lambda}_{k+1}-\boldsymbol{\mu}_{k+1}\big\|^2\\ =&F(\mathbf{x}_{k+1},\mathbf{y}_{k+1})+\frac{1}{2\beta}\big\|\boldsymbol{\lambda}_{k+1}\big\|^2-\frac{1}{2\beta}\big\|\boldsymbol{\mu}_{k+1}\big\|^2\\ \ge&\Big(f_1(\mathbf{x}_{k+1})-\frac{1}{2L_f}\big\|\nabla f_1(\mathbf{x}_{k+1})\big\|^2\Big)+\Big(g_1(\mathbf{y}_{k+1})-\frac{1}{2L_g}\big\|\nabla g_1(\mathbf{y}_{k+1})\big\|^2\Big)\\ &+f_2(\mathbf{x}_{k+1})+g_2(\mathbf{y}_{k+1})+\frac{1}{2\beta}\big\|\boldsymbol{\lambda}_{k+1}\big\|^2-\frac{1}{2\beta}\big\|\boldsymbol{\mu}_{k+1}\big\|^2\\ \ge&\bar{f}_1+\bar{g}_1+f_2(\mathbf{x}_{k+1})+g_2(\mathbf{y}_{k+1})+\frac{1}{2\beta}\big\|\boldsymbol{\lambda}_{k+1}\big\|^2-\frac{1}{2\beta}\big\|\boldsymbol{\mu}_{k+1}\big\|^2,\end{aligned}$$

where the last inequality uses (A1). Then, combining the above relationship with (A2), Lemma 3.1 as well as $r<1$, we conclude that both $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$ are bounded. Consequently, the whole sequence $\{\mathbf{w}_k\}$ is bounded. Because $\{\mathbf{w}_k\}$ is bounded, the sequence $\{\mathcal{L}_\beta(\mathbf{w}_k)\}$ is also bounded from below and there exists at least one limit point. Without loss of generality, let $\{\mathbf{w}_{k_j}\}$ be a subsequence of $\{\mathbf{w}_k\}$ and $\mathbf{w}^*$ be its limit point. Then, the lower semicontinuity of $\{\mathcal{L}_\beta(\mathbf{w}_k)\}$ implies $\mathcal{L}_\beta(\mathbf{w}^*)\le\lim_{j\to\infty}\inf\mathcal{L}_\beta(\mathbf{w}_{k_j})$ So, $\{\mathcal{L}_\beta(\mathbf{w}_{k_j})\}$ is bounded from below and hence is convergent.

Let $\underline{\mathcal{L}}_\beta$ be the lower bound of $\{\mathcal{L}_\beta(\mathbf{w}_k)\}$. Then, by the result (3.3) again, we deduce

$$\sum_{k=0}^{\infty}\Big(\frac{\theta_1-L_f}{2}\big\|\mathbf{x}_{k+1}-\mathbf{x}_k\big\|^2+\frac{\theta_2-L_g}{2}\big\|\mathbf{y}_{k+1}-\mathbf{y}_k\big\|^2\Big)\leq\mathcal{L}_\beta(\mathbf{w}_0)-\underline{\mathcal{L}}_\beta+\frac{2\delta_0}{\beta\bar{\tau}(1-r)}<+\infty, \tag{3.10}$$

where the last inequality holds by the fact

$$\sum_{k=0}^{\infty}\delta_k\leq\frac{\delta_0}{1-r}<+\infty. \tag{3.11}$$

According to (3.10) and the conditions $\theta_1>L_f,\theta_2>L_g$, we deduce

$$\lim_{k\to\infty}\big\|\mathbf{x}_{k+1}-\mathbf{x}_k\big\|^2=0,\quad\text{and}\quad\lim_{k\to\infty}\big\|\mathbf{y}_{k+1}-\mathbf{y}_k\big\|^2=0.$$

Summarizing the inequalities in (3.1) over $k=0,1,\cdots,\infty$ together with (3.11) shows

$$\lim_{k\to\infty}\big\|\boldsymbol{\mu}_{k+1}-\boldsymbol{\mu}_k\big\|^2=0,\quad\text{and}\quad\lim_{k\to\infty}\big\|\boldsymbol{\lambda}_k-\boldsymbol{\mu}_k\big\|^2=0. \tag{3.12}$$

Combine the following relationship from (3.2):

$$\big\|\boldsymbol{\lambda}_{k+1}-\boldsymbol{\lambda}_k\big\|^2\leq2\big\|\boldsymbol{\lambda}_{k+1}-\boldsymbol{\mu}_{k+1}\big\|^2+4\big(\big\|\boldsymbol{\mu}_{k+1}-\boldsymbol{\mu}_k\big\|^2+\big\|\boldsymbol{\mu}_k-\boldsymbol{\lambda}_k\big\|^2\big)$$

with (3.12) immediately ensures

$$\lim_{k\to\infty}\big\|\boldsymbol{\lambda}_{k+1}-\boldsymbol{\lambda}_k\big\|^2=0. \tag{3.13}$$

Besides, the update of $\mathbf{z}_{k+1}$ gives

$$\big\|\mathbf{z}_{k+1}-\mathbf{z}_k\big\|^2\leq\frac{2}{\alpha^2}\Big(\big\|\boldsymbol{\lambda}_{k+1}-\boldsymbol{\lambda}_k\big\|^2+\big\|\boldsymbol{\mu}_{k+1}-\boldsymbol{\mu}_k\big\|^2\Big),$$

which, by (3.13) and the first result in (3.12), further implies $\lim_{k\to\infty}\|\mathbf{z}_{k+1}-\mathbf{z}_k\|^2=0$. As a result, combine this limitation and (3.12)-(3.13) to confirm the first result in (3.4). The second result in (3.4) is clearly from $\lim_{k\to\infty}\|\boldsymbol{\lambda}_k-\boldsymbol{\mu}_k\|^2=0$ and the update of $\boldsymbol{\lambda}_{k+1}$. ■

3.2 Convergence and complexity

In the following, the distance from any point x to the set Ω is defined as $\mathrm{dist}(\mathbf{x},\Omega):=\inf_{\bar{x}\in\Omega}\|\mathbf{x}-\bar{\mathbf{x}}\|$. Based on this definition, we first give an estimation on $\mathrm{dist}\big(\mathbf{0},\partial L_\beta(\mathbf{w}_{k+1})\big)$ by the iterative residuals, and then analyze the convergence of the iterative sequence $\{\mathbf{w}_{k+1}\}$. Similar analysis can be found in [7, 17, 18, 41]. For a proper lower semi-continuous function h, its *(limiting-) subdifferential* [37, Definition 8.3 (b)] at $\mathbf{x}\in\mathrm{dom}h$, denoted as $\partial h(\mathbf{x})$, is defined as

$$\partial h(\mathbf{x}):=\Big\{\boldsymbol{\nu}\in\mathbb{R}^n:\exists\mathbf{x}^k\to\mathbf{x},h(\mathbf{x}^k)\to h(\mathbf{x}),\boldsymbol{\nu}^k\to\boldsymbol{\nu}\text{ with }\boldsymbol{\nu}^k\in\hat{\partial}h(\mathbf{x}^k)\Big\},$$

where $\hat{\partial}h(\mathbf{x})$ denotes the *regular subdifferential* [37, Definition 8.3 (a)] of h at $\mathbf{x}$ given as

$$\hat{\partial}h(\mathbf{x}):=\Big\{\boldsymbol{\nu}\in\mathbb{R}^n:\liminf_{\bar{\mathbf{x}}\to\mathbf{x},\bar{\mathbf{x}}\neq\mathbf{x}}\frac{h(\bar{\mathbf{x}})-h(\mathbf{x})-\langle\boldsymbol{\nu},\bar{\mathbf{x}}-\mathbf{x}\rangle}{\|\bar{\mathbf{x}}-\mathbf{x}\|}\geq0\Big\}.$$

Corollary 3.1 *Let $\{\mathbf{w}_k = (\mathbf{v}_k, \mathbf{z}_k, \boldsymbol{\mu}_k)\}$ be the sequence generated by Algorithm 2.1. Then, for every $k \geq 0$, the following hold:*

(i) There exists a F^* such that

$$\lim_{k\to\infty} \mathcal{L}_\beta(\mathbf{w}_{k+1}) = \lim_{k\to\infty} L(\mathbf{v}_{k+1}) = \lim_{k\to\infty} F(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) = F^*.$$

(ii) It holds that $\lim_{k\to\infty} \operatorname{dist}\big(\mathbf{0}, \partial\mathcal{L}_\beta(\mathbf{w}_{k+1})\big) = \lim_{k\to\infty} \operatorname{dist}\big(\mathbf{0}, \partial L(\mathbf{v}_{k+1})\big) = 0.$

Proof. Note that

$$\begin{aligned} F(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) &= \mathcal{L}_\beta(\mathbf{w}_{k+1}) - \big\langle \boldsymbol{\lambda}_{k+1}, A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - \mathbf{b}\big\rangle + \frac{1}{2\beta}\|\boldsymbol{\lambda}_{k+1} - \boldsymbol{\mu}_{k+1}\|^2 \\ &= L(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \boldsymbol{\lambda}_{k+1}) - \big\langle \boldsymbol{\lambda}_{k+1}, A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - \mathbf{b}\big\rangle, \end{aligned}$$

which ensures the conclusion (i) by the second item of Theorem 3.1 and (3.12).

The first-order optimality condition of $\mathbf{x}_{k+1}$-subproblem implies

$$\mathbf{0} \in \partial f_2(\mathbf{x}_{k+1}) + \nabla f_1(\mathbf{x}_k) + A^\mathsf{T}\boldsymbol{\lambda}_k + \nabla\phi_1(\mathbf{x}_{k+1}) - \nabla\phi_1(\mathbf{x}_k).$$

Combining it with the reformulation (2.4) to have

$$\mathbf{e}^{\mathbf{x}}_{k+1} \in \partial_{\mathbf{x}}\mathcal{L}_\beta(\mathbf{w}_{k+1}),$$

where $\mathbf{e}^{\mathbf{x}}_{k+1} := \nabla f_1(\mathbf{x}_{k+1}) - \nabla f_1(\mathbf{x}_k) + \nabla\phi_1(\mathbf{x}_k) - \nabla\phi_1(\mathbf{x}_{k+1}) + A^\mathsf{T}(\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_k)$. Similarly, we have from the first-order optimality condition of $\mathbf{y}_{k+1}$-subproblem that

$$\mathbf{e}^{\mathbf{y}}_{k+1} \in \partial_{\mathbf{y}}\mathcal{L}_\beta(\mathbf{w}_{k+1}).$$

where $\mathbf{e}^{\mathbf{y}}_{k+1} := \nabla g_1(\mathbf{y}_{k+1}) - \nabla g_1(\mathbf{y}_k) + \nabla\phi_2(\mathbf{y}_k) - \nabla\phi_2(\mathbf{y}_{k+1}) + B^\mathsf{T}(\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_k)$. Besides, it follows from $\boldsymbol{\lambda}$-update that

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}_\beta(\mathbf{w}_{k+1}) = (A\mathbf{x}_{k+1} + B\mathbf{y}_{k+1} - b) - \frac{1}{\beta}(\boldsymbol{\lambda}_{k+1} - \boldsymbol{\mu}_{k+1}) = \mathbf{0}$$

and $\nabla_{\boldsymbol{\mu}}\mathcal{L}_\beta(\mathbf{w}_{k+1}) = -\frac{1}{\beta}(\boldsymbol{\mu}_{k+1} - \boldsymbol{\lambda}_{k+1}) := \mathbf{e}^{\boldsymbol{\mu}}_{k+1}$. Hence, the following relationship holds:

$$\mathbf{e}_{k+1} := \big(\mathbf{e}^{\mathbf{x}}_{k+1}, \mathbf{e}^{\mathbf{y}}_{k+1}, \mathbf{0}, \mathbf{e}^{\boldsymbol{\mu}}_{k+1}\big) \in \partial\mathcal{L}_\beta(\mathbf{w}_{k+1}).$$

Next, we simplify the computation of each component of $\mathbf{e}_{k+1}$. By the Lipschitz continuity of f_1 and ϕ_1, we have

$$\begin{aligned} \big\|\mathbf{e}^x_{k+1}\big\| &\leq \big\|\nabla f_1(\mathbf{x}_{k+1}) - \nabla f_1(\mathbf{x}_k)\big\| + \big\|\nabla\phi_1(\mathbf{x}_{k+1}) - \nabla\phi_1(\mathbf{x}_k)\big\| + \|A\|\big\|\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_k\big\| \\ &\leq (L_f + L_{\phi_1})\big\|\mathbf{x}_{k+1} - \mathbf{x}_k\big\| + \|A\|\big\|\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_k\big\|. \end{aligned}$$

Analogously, we have by the Lipschitz continuity of g_1 and ϕ_2 that

$$\begin{aligned} \big\|\mathbf{e}^y_{k+1}\big\| &\leq \big\|\nabla g_1(\mathbf{y}_{k+1}) - \nabla g_1(\mathbf{y}_k)\big\| + \big\|\nabla\phi_2(\mathbf{y}_{k+1}) - \nabla\phi_2(\mathbf{y}_k)\big\| + \|B\|\big\|\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_k\big\| \\ &\leq (L_g + L_{\phi_2})\big\|\mathbf{y}_{k+1} - \mathbf{y}_k\big\| + \|B\|\big\|\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_k\big\|. \end{aligned}$$

Combining the last two results, the equality $\|\mathbf{e}_{k+1}^{\boldsymbol{\mu}}\| = \frac{1}{\beta}\|\boldsymbol{\lambda}_{k+1} - \boldsymbol{\mu}_{k+1}\|$ and the relationships

$$\begin{cases} \partial_{\mathbf{x}}\mathcal{L}_\beta(\mathbf{w}_{k+1}) = \partial_{\mathbf{x}}L(\mathbf{v}_{k+1}), \quad \partial_{\mathbf{y}}\mathcal{L}_\beta(\mathbf{w}_{k+1}) = \partial_{\mathbf{y}}L(\mathbf{v}_{k+1}), \\ \partial_{\boldsymbol{\lambda}}\mathcal{L}_\beta(\mathbf{w}_{k+1}) = \partial_{\boldsymbol{\lambda}}L(\mathbf{v}_{k+1}) - \dfrac{1}{\beta}(\boldsymbol{\lambda}_{k+1} - \boldsymbol{\mu}_{k+1}), \end{cases}$$

to obtain

$$\operatorname{dist}\big(\mathbf{0}, \partial L(\mathbf{v}_{k+1})\big) \le \operatorname{dist}\big(\mathbf{0}, \partial\mathcal{L}_\beta(\mathbf{w}_{k+1})\big) + \frac{1}{\beta}\big\|\boldsymbol{\lambda}_{k+1} - \boldsymbol{\mu}_{k+1}\big\|$$

and

$$\operatorname{dist}\big(\mathbf{0}, \partial\mathcal{L}_\beta(\mathbf{w}_{k+1})\big) \le \big\|\mathbf{e}_{k+1}\big\| \le c\big(\big\|\mathbf{x}_{k+1}-\mathbf{x}_k\big\| + \big\|\mathbf{y}_{k+1}-\mathbf{y}_k\big\| + \big\|\boldsymbol{\lambda}_{k+1}-\boldsymbol{\lambda}_k\big\| + \big\|\boldsymbol{\lambda}_{k+1}-\boldsymbol{\mu}_{k+1}\big\|\big)$$

with $c = \max\{L_f + L_{\phi_1}, L_g + L_{\phi_2}, \|A\| + \|B\|, \frac{1}{\beta}\}$. Then, we confirm the result (ii) by the first equality in (3.4). ■

Corollary 3.1 shows that the objective sequence of (1.1) is convergent, but it does not point out the convergence of the iterative sequence as well as its convergence rate. In what follows, we not only show that any limit point of $\{\mathbf{v}_k = (\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_k)\}$ converges to a stationary point of (1.1) as defined by (3.14), but also establish the sublinear convergence rate of the iterative residuals of the primal variables. We say $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$ is a stationary point of (1.1) if $\mathbf{0} \in \partial L(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda})$, that is,

$$\mathbf{0} \in \nabla f_1(\mathbf{x}^*) + \partial f_2(\mathbf{x}^*) + A^{\mathsf{T}}\boldsymbol{\lambda}^*, \ \mathbf{0} \in \nabla g_1(\mathbf{y}^*) + \partial g_2(\mathbf{y}^*) + B^{\mathsf{T}}\boldsymbol{\lambda}^*, \ A\mathbf{x}^* + B\mathbf{y}^* = \mathbf{b}. \tag{3.14}$$

Theorem 3.2 *Let $\{\mathbf{w}_k = (\mathbf{v}_k, \mathbf{z}_k, \boldsymbol{\mu}_k)\}$ be the sequence generated by Algorithm 2.1. Then,*

(i) Any limit point $\mathbf{v}^*$ of the sequence $\{\mathbf{v}_k\}$ is a stationary point of (1.1);

(ii) For any integer $k \ge 1$, there exist $j \le k$ and $\zeta_1, \zeta_2 > 0$ such that

$$\big\|\mathbf{x}_{j+1} - \mathbf{x}_j\big\|^2 \le \frac{\zeta_0}{\zeta_1(k+1)} \quad \text{and} \quad \big\|\mathbf{y}_{j+1} - \mathbf{y}_j\big\|^2 \le \frac{\zeta_0}{\zeta_2(k+1)},$$

where $\zeta_0 = \mathcal{L}_\beta(\mathbf{w}_0) - \underline{\mathcal{L}}_\beta + \frac{2\delta_0}{\beta\bar{\tau}(1-r)}$ with $\underline{\mathcal{L}}_\beta$ being the lower bound of $\{\mathcal{L}_\beta(\mathbf{w}_k)\}$.

Proof. For any limit point $\mathbf{w}^* = (\mathbf{v}^*, \mathbf{z}^*, \boldsymbol{\mu}^*)$ of the sequence $\{\mathbf{w}_k\}$, it follows from the second conclusion of Corollary 3.1, together with the definition of the limiting-subdifferential $\partial L(\mathbf{v}^*)$ and the definition of the stationary point in (3.14), that the conclusion (i) holds.

Secondly, for any $k > 0$, we have from (3.3) and (3.11) that

$$\sum_{j=0}^{k}\Big(\frac{\theta_1 - L_f}{2}\big\|\mathbf{x}_{j+1} - \mathbf{x}_j\big\|^2 + \frac{\theta_2 - L_g}{2}\big\|\mathbf{y}_{j+1} - \mathbf{y}_j\big\|^2\Big) \le \mathcal{L}_\beta(\mathbf{w}_0) - \underline{\mathcal{L}}_\beta + \frac{2\delta_0}{\beta\bar{\tau}(1-r)} = \zeta_0,$$

which indicates that there exists a $j \le k$ such that

$$\big\|\mathbf{x}_{j+1} - \mathbf{x}_j\big\|^2 \le \frac{\zeta_0}{(k+1)(\theta_1 - L_f)} \quad \text{and} \quad \big\|\mathbf{y}_{j+1} - \mathbf{y}_j\big\|^2 \le \frac{\zeta_0}{(k+1)(\theta_2 - L_g)}.$$

These inequalities with $\zeta_1 = \theta_1 - L_f > 0, \zeta_2 = \theta_2 - L_g > 0$ confirm the conclusion (ii). ■

4 Numerical experiments

In this section, we investigate the performance of our proposed algorithm for solving three problems with artificial data or real data. All experiments are implemented in MATLAB R2021b (64-bit) and performed on a PC with Windows 11 operating system, with an AMD Ryzen 9 7945HX CPU and 16GB RAM.

4.1 Linear equation problem

Consider the linear equation problem (1.2) with $\rho = 0.001$ and $\mathbf{d}$ being the vector of ones. It can be verified that the assumptions (A1)-(A2) hold. In this experiment, the coefficient matrices A and B are set as tridiagonal matrices, since tridiagonal systems frequently arise in practical applications involving discretized differential equations. More specifically, we generate A, B and $\mathbf{b}$ by the following codes:

```
v1 = randn(1,m); u1 = randn(1,m-1); l1 = randn(1,m-1);
A  = diag(v1) + diag(u1, 1) + diag(l1, -1);
v2 = randn(1,m); u2 = randn(1,m-1); l2 = randn(1,m-1);
B  = diag(v2) + diag(u2, 1) + diag(l2, -1);
w1 = rand(m, 1); w2 = rand(m, 1); b  = A*w1+B*w2;
```

By selecting $\phi_i(\cdot) = \frac{\theta_i}{2}\|\cdot\|^2$ in both $\mathbf{x}$-subproblem and $\mathbf{y}$-subproblem, the Bregman distance reduces to $\mathcal{B}_{\phi_i}(\cdot,\cdot_k) = \frac{\theta_i}{2}\|\cdot - \cdot_k\|^2$ and hence the modulus θ_i plays the role of proximal parameter. Applying Algorithm 2.1 (2P-ADMM) to this problem involves two core iterations:

$$\begin{cases} \mathbf{x}_{k+1} = \mathcal{P}_{\mathbb{R}^n_+}\left[\mathbf{x_k} - \frac{1}{\theta_1}\left(\mathbf{x}_k + A^{\mathsf{T}}\boldsymbol{\lambda}_k\right)\right], \\ \mathbf{y}_{k+1} = \mathcal{P}_{[\mathbf{0},\mathbf{d}]}\left[\mathbf{y_k} - \frac{1}{\theta_2}\left(\rho\mathbf{y}_k + B^{\mathsf{T}}\boldsymbol{\lambda}_k\right)\right], \end{cases}$$

where $\mathcal{P}_{\mathbb{R}^n_+}(\mathbf{x}_k)$ denotes the projection of $\mathbf{x}_k$ onto $\mathbb{R}^n_+$(the set of n-dimensional nonnegative vectors). The constraint violation error and the relative objective error are defined as

$$\text{Equ\_err}(k) = \|A\mathbf{x}_k + B\mathbf{y}_k - \mathbf{b}\| \quad \text{and} \quad \text{Obj\_err}(k) = \frac{|F(\mathbf{x}_k,\mathbf{y}_k) - F^*|}{\max\{F^*,1\}},$$

respectively, where F^* denotes an approximate optimal objective function value obtained by running Algorithm 2.1 for more than 10^5 iterations. With the initial point $(\mathbf{x}_0, \mathbf{y}_0)$ randomly drawn from the standard Gaussian and $(\boldsymbol{\lambda}_0, \boldsymbol{\mu}_0) = (\mathbf{0}, \mathbf{0})$, all algorithms are terminated when the stopping condition $\text{Equ\_err}(k) < tol$ or $\text{Iter} > 5000$, where "Iter" denotes the total number of iterations and the tolerance tol is set as 10^{-5} in the forthcoming experiments.

Next, we investigate how the parameters $(\alpha, \sigma, r, \delta_0)$ affect the performance of our 2P-ADMM. We first fix the free parameter α as 10^3, then change other distinctively constrained parameters (σ, r, δ_0) to investigate their effects on 2P-ADMM. Table 2 reports some numerical results of 2P-ADMM with different parameters (σ, r, δ_0) for solving problem (1.2) with dimension $p = 300$, where "Time" denotes the CPU time in seconds and the bold value denotes the smallest one. We can observe from Table 2 that:

- For the parameters (σ, r, δ_0), the reported results in each column of Equ_err and Obj_err are nearly the same when two parameters are fixed and only one parameter is allowed to change;

Table 2: Numerical results of 2P-ADMM for solving (1.2) with different (σ, r, δ_0).

Parameters	Iter	Time	Equ_err	Obj_err
$\sigma(r = 1 - 10^{-7}, \delta_0 = 0.5)$				
0.5	**2983**	**0.09**	9.986e-6	4.619e-6
0.6	3178	0.10	9.963e-6	3.692e-6
0.7	3332	0.10	9.767e-6	2.931e-6
0.8	3470	0.10	9.979e-6	2.568e-6
0.9	3674	0.11	9.550e-6	2.036e-6
$r(\sigma = 0.5, \delta_0 = 0.5)$				
$1 - 10^{-3}$	5000	0.14	2.588e-4	4.371e-6
$1 - 10^{-5}$	3006	0.09	9.942e-6	4.500e-6
$1 - 10^{-7}$	2983	0.09	9.986e-6	4.619e-6
$1 - 10^{-9}$	2983	0.09	9.956e-6	4.622e-6
$1 - 10^{-11}$	**2983**	**0.09**	9.956e-6	4.622e-6
$\delta_0(\sigma = 0.5, r = 1 - 10^{-7})$				
0.1	4949	0.14	9.997e-6	3.075e-6
0.3	3543	0.11	9.988e-6	2.345e-6
0.5	2983	0.09	9.986e-6	4.619e-6
0.7	**2796**	**0.08**	9.791e-6	5.829e-6
0.9	2908	0.09	9.664e-6	5.124e-6

- With the increase of the parameter σ, both the iteration number and the CPU time tend to increase; with the increase of the parameter r, both the iteration number and the CPU time tend to decrease; with the increase of the parameter δ_0, both the iteration number and the CPU time decrease firstly and then increase.

Reported results of Table 2 indicate that the choice of (σ, r, δ_0) could have a great effect on the performance of 2P-ADMM, and it seems that setting $(\sigma, r, \delta_0) = (0.5, 1 - 10^{-11}, 0.7)$ would be a reasonable choice for solving (1.2) and these values will be set as default values in the forthcoming experiments.

In what follows, we carry out some comparative experiments about 2P-ADMM and the existing algorithms SALM_psils and SALM_psilsf proposed in [23]. The involved parameters of SALM_psils and SALM_psilsf are set as $\rho = 0.1$, $s = 10^{-5}$, $\beta = 10$, $L_0 = \beta\|B^{\mathsf{T}}B\| + 0.001$ and $\mathcal{H}_k^y \equiv \lambda\mathbf{I} + \beta\|B^{\mathsf{T}}B\|$. Figure 1 depicts the convergence curves of Equ_err(k) and Obj_err(k) against the iteration numbers. Table 3 reports some numerical results, where the notation "/" indicates that it reaches the maximum iteration numbers. As illustrated in Figure 1, 2P-ADMM initially performs worse than SALM_psils and SALM_psilsf in terms of Obj_err, but it eventually outperforms these two methods. Numerical results in Table 3 not only show that our 2P-ADMM requires fewer iterations and CPU time, but also demonstrate its robustness even for a significantly large penalty value $\alpha = 10^8$.

To investigate the impact of Bregman distance on the performance of 2P-ADMM, we next select $\phi_i(\mathbf{x}) = \sum x_j \log(x_j)$ for either $\mathbf{x}$-subprobem or $\mathbf{y}$-subprobem as toy examples, making the Bregman distance becomes KL divergence. Here, x_j denotes the j-th entry of $\mathbf{x}$. It is easy to check that each subproblem of 2P-ADMM still admits a closed-form solution. We conduct comparative experiments in four cases

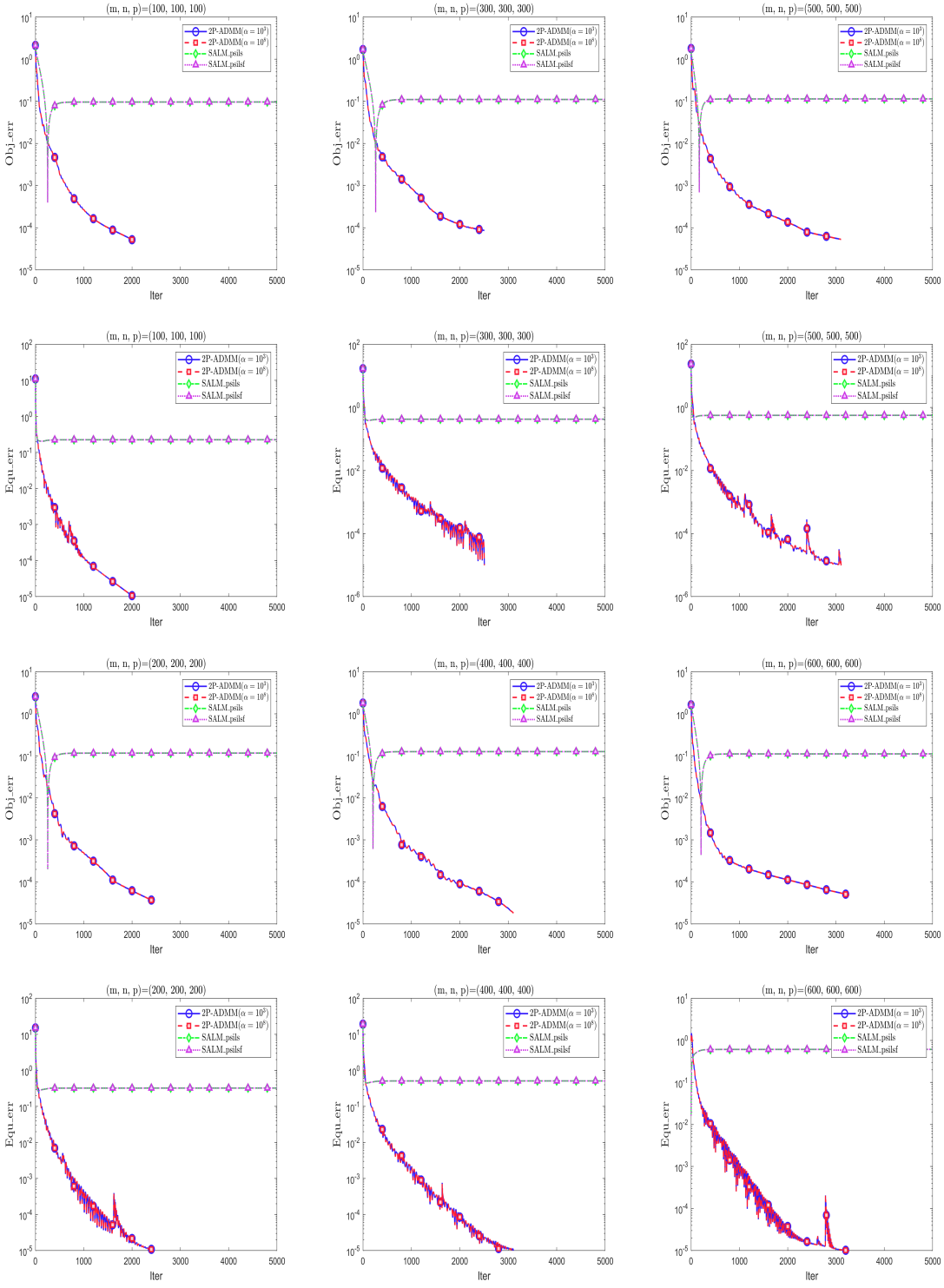

Figure 1: Comparison curves of different algorithms for (1.2) with different (m, n, p).

Table 3: Numerical results of different algorithms for (1.2).

Size	Parameters		2P-ADMM($\alpha = 10^3$)			
p	θ_1	θ_2	Iter	Time	Equ_err	Obj_err
100	3.02e+1	3.04e+1	2029	0.02	9.979e-6	5.000e-5
200	3.57e+1	3.55e+1	2464	0.03	9.981e-6	3.439e-5
300	3.77e+1	3.75e+1	2509	0.08	9.922e-6	8.672e-5
400	4.45e+1	4.18e+1	3111	0.15	9.965e-6	1.801e-5
500	3.53e+1	3.53e+1	3104	0.23	9.777e-6	5.321e-5
600	2.47e+1	2.45e+1	3282	0.35	9.999e-6	4.892e-5
Size	Parameters		2P-ADMM($\alpha = 10^8$)			
p	θ_1	θ_2	Iter	Time	Equ_err	Obj_err
100	3.02e+1	3.04e+1	**2028**	**0.02**	9.982e-6	5.008e-5
200	3.57e+1	3.55e+1	**2461**	**0.03**	9.993e-6	3.453e-5
300	3.77e+1	3.75e+1	**2507**	**0.07**	9.811e-6	8.682e-5
400	4.45e+1	4.18e+1	**3108**	**0.15**	9.973e-6	1.815e-5
500	3.53e+1	3.53e+1	**3104**	**0.22**	9.763e-6	5.321e-5
600	2.47e+1	2.45e+1	**3274**	**0.34**	9.998e-6	4.917e-5

Size	Parameter	SALM_psils			
p	L_0	Iter	Time	Equ_err	Obj_err
100	1.94e+2	/	2.69	2.263e-1	9.666e-2
200	2.13e+2	/	14.63	3.250e-1	1.169e-1
300	2.29e+2	/	29.08	4.255e-1	1.110e-1
400	1.97e+2	/	54.11	5.128e-1	1.267e-1
500	1.56e+2	/	79.49	5.644e-1	1.146e-1
600	1.93e+2	/	131.03	6.138e-1	1.119e-1
Size	Parameter	SALM_psilsf			
p	L_0	Iter	Time	Equ_err	Obj_err
100	1.94e+2	/	2.73	2.263e-1	9.666e-2
200	2.13e+2	/	14.80	3.250e-1	1.169e-1
300	2.29e+2	/	29.57	4.255e-1	1.110e-1
400	1.97e+2	/	52.94	5.128e-1	1.267e-1
500	1.56e+2	/	80.91	5.644e-1	1.146e-1
600	1.93e+2	/	132.20	6.138e-1	1.119e-1

- 2P-ADMM-I: $\mathcal{B}_{\phi_i}(\mathbf{v},\mathbf{v}_k) = \frac{\theta_i}{2}\|\mathbf{v}-\mathbf{v}_k\|^2$ for the $\mathbf{x}$-subproblem with $\mathbf{v} := \mathbf{x}$ but for the $\mathbf{y}$-subproblem with $\mathbf{v} := \mathbf{y}$;
- 2P-ADMM-II: $\mathcal{B}_{\phi_1}(\mathbf{x},\mathbf{x}_k) = \frac{\theta_1}{2}\|\mathbf{x}-\mathbf{x}_k\|^2$ for the $\mathbf{x}$-subproblem and $\mathcal{B}_{\phi_2}(\mathbf{y},\mathbf{v}) = \sum y_j \log(\frac{y_j}{v_j})$ for the $\mathbf{y}$-subproblem with $\mathbf{v} := \mathbf{y}_k$;
- 2P-ADMM-III: $\mathcal{B}_{\phi_1}(\mathbf{x},\mathbf{v}) = \sum x_j \log(\frac{x_j}{v_j})$ for the $\mathbf{x}$-subproblem with $\mathbf{v} := \mathbf{x}_k$ and $\mathcal{B}_{\phi_2}(\mathbf{y},\mathbf{y}_k) = \frac{\theta_2}{2}\|\mathbf{y}-\mathbf{y}_k\|^2$ for the $\mathbf{y}$-subproblem;
- 2P-ADMM-IV: $\mathcal{B}_{\phi_1}(\mathbf{x},\mathbf{v}) = \sum x_j \log(\frac{x_j}{v_j})$ for the $\mathbf{x}$-subproblem with $\mathbf{v} := \mathbf{x}_k$ and $\mathcal{B}_{\phi_2}(\mathbf{y},\mathbf{v}) = \sum y_j \log(\frac{y_j}{v_j})$ for the $\mathbf{y}$-subproblem with $\mathbf{v} := \mathbf{y}_k$.

Figure 2 depicts the convergence curves of both Equ_err(k) and Obj_err(k) against the iteration numbers with problem size $(m,n,p) = (300,300,300)$. As illustrated in Figure 2, the selection of Bregman distances exerts a significant influence on the performance of 2P-ADMM: 2P-ADMM-II achieves the best performance, and 2P-ADMM-I performs slightly worse than 2P-ADMM-II but better than other two types. This verifies that the selection of Bregman distance has an important effect on the numerical performance of 2P-ADMM.

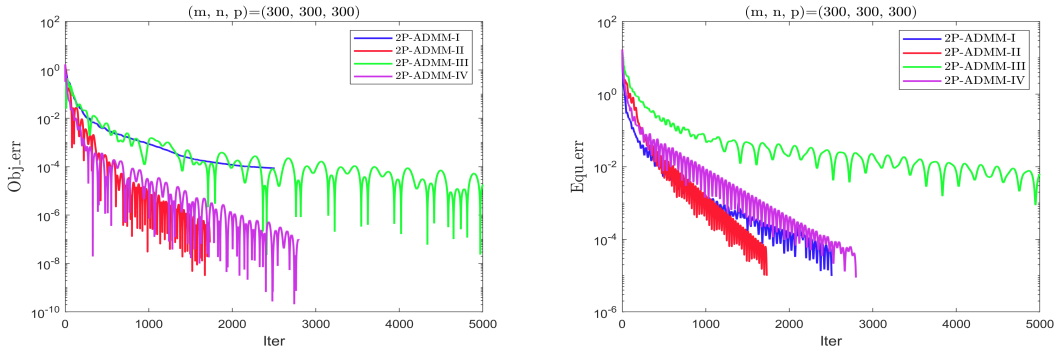

Figure 2: Comparison of different 2P-ADMM for (1.2) with $(m,n,p) = (300,300,300)$.

4.2 Graph-guided fused lasso problem

Consider an equivalent form of the graph-guided fused lasso problem (1.3):

$$\min F(\mathbf{x},\mathbf{y}) := \frac{1}{N}\sum_{j=1}^{N} f_j(\mathbf{x}) + \rho_1\|\mathbf{y}\|_1 + \frac{\rho_2}{2}\|\mathbf{x}\|^2 \quad \text{s.t.} \quad A\mathbf{x}-\mathbf{y} = \mathbf{0}. \tag{4.1}$$

Here, $A = [\mathbf{G};\mathbf{I}]$ is a matrix encoding the feature sparsity pattern, and $\mathbf{G}$ is the sparsity pattern of the graph that is obtained by sparse inverse covariance estimation [16]. Applying 2P-ADMM to (4.1) with $\mathcal{B}_{\phi_i}(\cdot,\cdot_k) = \frac{\theta_i}{2}\|\cdot-\cdot_k\|^2$, the resulting subproblems read

$$\begin{cases} \mathbf{x}_{k+1} = \frac{\theta_1}{\rho_2+\theta_1}\Big[\mathbf{x_k} - \frac{1}{\theta_1}\big(\nabla f_1(\mathbf{x}_k) + A^{\mathsf{T}}\boldsymbol{\lambda}_k\big)\Big], \\ \mathbf{y}_{k+1} = \mathrm{Shrink}\big(\mathbf{y_k} + \frac{\boldsymbol{\lambda}_k}{\theta_2}, \frac{\rho_1}{\theta_2}\big), \end{cases} \tag{4.2}$$

where Shrink(,) denotes the soft shrinkage operator and can be evaluated by the MATLAB built-in function `wthresh`.

Our 2P-ADMM using parameters $(\sigma, r, \delta_0) = (0.5, 1-10^{-7}, 0.5)$ and $\alpha \in \{10^3, 10^8\}$ is compared to the following existing algorithms for solving (4.1) with public datasets a2a (30,296 samples, 123 features), a8a (9,865 samples, 123 features) and a9a (16,281 samples, 123 features):

- Linearized Symmetric Proximal ADMM (LSPADMM, [14]) with parameter $\tau = -0.1$ and other parameters as suggested by [14, Assumption 3.1 (v)];
- Two-step Linear Inertial ADMM (TLIADMM, [15]) with tuned penalty parameter $\beta = 20$, while maintaining other parameters as suggested on [15, Page 15];
- Linearized ADMM with parameters set to the lower bound specified in [33, Theorem 1] plus an offset of 0.001.

Besides, the regularization parameters are set as $\rho_1 = \rho_2 = 10^{-6}$. All algorithms are terminated when reaching a given budget CPU time. Similar stopping criterion can be found in [4]. To measure the performance of an algorithm, we plot the loss and the maximum of the relative objective error and the constraint error, that is Opt_err(k)=max(Obj_err(k), Equ_err(k)), against the CPU time.

As different datasets correspond to different Lipschitz constants, we use tuned parameters $(\theta_1, \theta_2) = (45, 18)$ for the a2a dataset, $(\theta_1, \theta_2) = (44, 20)$ for the a8a dataset, and $(\theta_1, \theta_2) = (42.5, 5)$ for the a9a dataset. Figure 3 shows the convergence results of loss and Opt_err by the aforementioned methods. It can be seen from Figure 3 that 2P-ADMM initially performs worse than other methods during the beginning iterations, but it eventually outperforms other comparison algorithms. Besides, the results further demonstrate the robustness of 2P-ADMM with respect to the penalty parameter α, which verifies the conclusion in Section 4.1.

4.3 Robust principal component analysis problem

The robust principal component analysis problem, which arises from video surveillance and face recognition [6, 11], aims to decompose a data matrix $C \in \mathbb{R}^{m\times n}$ into a low-rank matrix X and a sparse matrix Y containing outliers and corrupt data. Generally speaking, $\|X\|_*$ (the sum of its singular values) and $\|Y\|_1$ (the sum of its absolute values) are used to characterize the low-rank matrix and sparse matrix, respectively. Similar to the technique of reformulating the compressed sensing problem [42], this paper will replace the sparse term $\|Y\|_1$ by the l_1-l_2 norm in the form of $\|\cdot\|_1 - \|\cdot\|_F^2$, resulting in the following relaxed model:

$$\min_{X,Y\in\mathbb{R}^{m\times n}} \left\{\|X\|_* + \rho(\|Y\|_1 - \|Y\|_F^2)\right\} \qquad \text{s.t.} \;\; X + Y = C, \tag{4.3}$$

where the weight parameter typically takes $\rho = 1/\sqrt{\max(m,n)}$. Clearly, problem (4.3) is a special case of (1.1) with $(A, B, \mathbf{b}) = (\mathbf{I}, \mathbf{I}, C)$, $f_2(X) = \|X\|_*$, $g_2(Y) = \rho\|Y\|_1$ and $g_1(Y) = -\rho\|Y\|_F^2$, hence our 2P-ADMM can be applied to solve it. By choosing $\mathcal{B}_{\phi_i}(\cdot, \cdot_k) = \frac{\theta_i}{2}\|\cdot - \cdot_k\|_F^2$, the resulting subproblems read

$$\begin{cases} X_{k+1} = U_k \mathrm{diag}\left(\max\left\{\sigma_i^k - \frac{1}{\theta_1}, 0\right\}\right) V_k^{\mathsf{T}}, \\ Y_{k+1} = \mathrm{Shrink}\left(Y_k + \frac{2\rho Y_k - \Lambda_k}{\theta_2}, \frac{\rho}{\theta_2}\right), \end{cases} \tag{4.4}$$

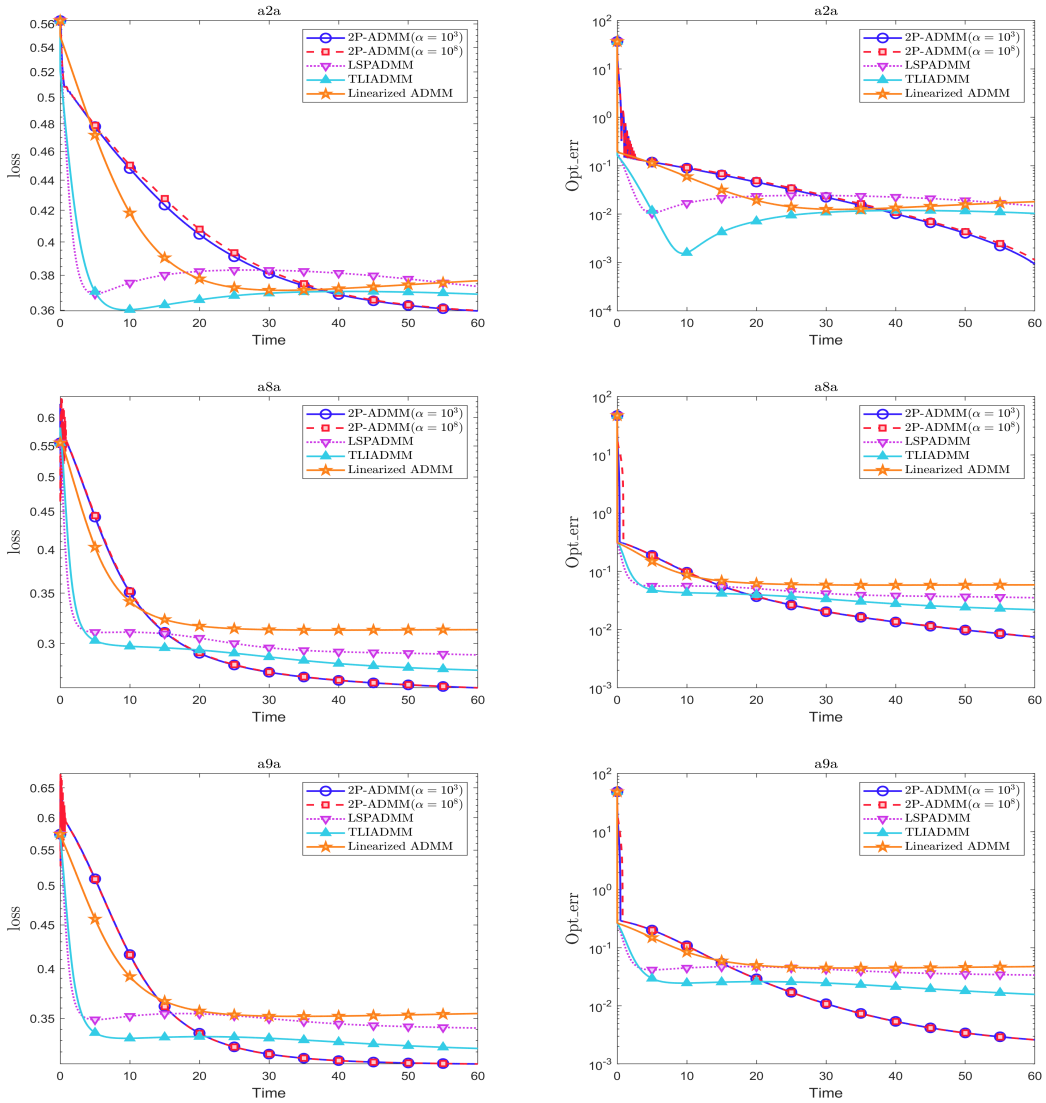

Figure 3: Comparison curves of different algorithms for solving (1.3) on public datasets.

where $U_k \in \mathbb{R}^{m\times r}$ and $V_k \in \mathbb{R}^{n\times r}$ are obtained by the singular value decomposition: $X_k - \frac{\Lambda_k}{\theta_1} = U_k\Sigma_k V_k^{\mathsf{T}}$ with $\Sigma_k = \mathrm{diag}\left(\sigma_1^k, \sigma_1^k, \cdots, \sigma_r^k\right)$.

Now, we apply our 2P-ADMM and the following three well-established algorithms to solve the problem (4.3) on the Yale B database[1] which consists of cropped and aligned images of 38 individuals under 9 poses and 64 lighting conditions:

- Standard ADMM with the penalty parameter $\frac{mn}{4\|C\|_1}$ according to [10, Page 109];
- Proximal Linearized ADMM (PL-ADMM, [44]) with the above same penalty parameter, the unit relaxation parameter and $Q_k = \theta\mathbf{I}$ with $\theta = 24$ according to [44, Page 4];
- Inertial Proximal ADMM (IPADMM, [12]) with the suggested penalty parameter on [12, Page 16], but with the tuned parameters $\rho_k = 1.4$ and $Q_k = \theta\mathbf{I}$ with $\theta = 24$ since these settings lead to better performance than the original setting;
- Two-step Inertial ADMM (TIADMM, [15]) with involved parameters tuned as $\beta = 3$, $t = 0.625$ and $\rho_1^k = \tau_1^k = \rho_2^k = \tau_2^k = 1.5$, since these settings result in better performance than the original setting.

We choose parameters $(\sigma, r, \delta_0) = (0.5, 1-10^{-7}, 0.5), \alpha \in \{10^3, 10^8\}$ and $\theta_1 = \theta_2 = 2$ for 2P-ADMM. Starting from the same initial feasible points $(\Lambda_0, M_0, Y_0) = (X_0, \mathbf{0}, C - X_0)$, all the above algorithms are terminated when the following stopping criteria are satisfied:

$$\mathrm{Opt\_err}(k) := \max\left\{\frac{\|X_{k+1} - X_k\|_F + \|Y_{k+1} - Y_k\|_F}{\|X_k\|_F + \|Y_k\|_F + 1}, \frac{\|C - X_{k+1} - Y_{k+1}\|_F}{\|C\|_F}\right\} < \epsilon$$

or the maximal iteration exceeds 10000, where ϵ is a given tolerance and X_0 is obtained by the truncated singular value decomposition:

$$X_0 = \texttt{F(:,1:l)Sigma(1:l,1:l)N(:,1:l)} \text{ where } \texttt{[F,Sigma,N]=svd(D,'econ');l=2.}$$

Similar initialization and stopping criteria can be found in [6].

Table 4 reports some comparative results under different tolerance. Figure 4 shows the comparative convergence curve of $\mathrm{Opt\_err}(k)$ against the number of iterations and the CPU time under different tolerance ϵ. The original columns of C, along with the low-rank and sparse components decomposed by different algorithms under $\epsilon = 10^{-5}$, are shown in Figure 5. We can see from Table 4 and Figure 4 that 2P-ADMM outperforms all comprative methods in terms of the iteration numbers and the CPU time. Moreover, it can effectively fill in occluded regions of the images, corresponding to shadows. In the low-rank component X as shown in Figure 5, shadows under different lighting conditions are removed and filled in with the most consistent low-rank features from the eigenfaces. Similar to the conclusions in Section 4.1 and Section 4.2, the results further demonstrate the robustness of 2P-ADMM with respect to the penalty parameter α.

5 Concluding remarks

Starting from the very beginning of algorithm design, this paper presents a novel proximal-perturbed augmented Lagrangian and then develops a corresponding Bregman-type alternating

[1] Available at http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html.

Table 4: Numerical results of different algorithms for the problem (4.3).

ϵ	Methods	Iter	Time	rank(X)	Opt_err(end)
	2P-ADMM($\alpha = 10^3$)	1023	36.97	25	9.99e-5
	2P-ADMM($\alpha = 10^8$)	**1023**	**36.67**	25	9.99e-5
10^{-4}	ADMM	1296	43.88	23	9.99e-5
	PL-ADMM	1167	41.53	23	9.99e-5
	IPADMM	1164	41.91	24	9.99e-5
	TIADMM	1261	48.70	23	9.99e-5
	2P-ADMM($\alpha = 10^3$)	4535	171.80	25	9.99e-6
	2P-ADMM($\alpha = 10^8$)	**4535**	**159.13**	25	9.99e-6
10^{-5}	ADMM	5484	174.40	25	9.99e-6
	PL-ADMM	4938	173.77	25	9.99e-6
	IPADMM	4907	171.87	25	9.99e-6
	TIADMM	5285	194.50	25	9.99e-6

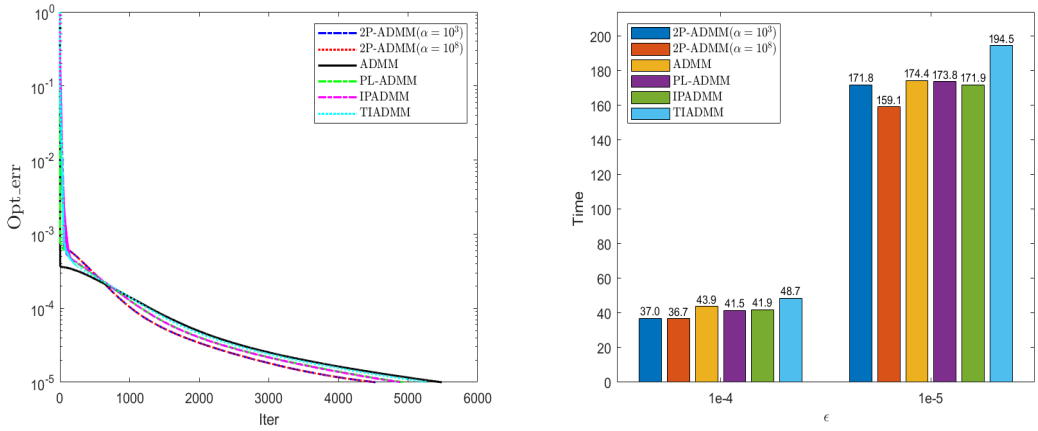

Figure 4: Comparison results of different algorithms for solving (4.3) on Yale B database.

Figure 5: Output of different algorithms for the 4th(rows 1-3), 18th(rows 4-6) and 46th(rows 7-9) images on the Yale B database. From left to right: 2P-ADMM($\alpha = 10^3$), 2P-ADMM($\alpha = 10^8$), ADMM, PL-ADMM, IPADMM, TIADMM, respectively.

direction method of multipliers (2P-ADMM) for solving a family of linearly constrained nonconvex composite minimization problems. Under mild assumptions, we theoretically establish that any limitation point of the iterative sequence generated by 2P-ADMM converges to a stationary point of the problem. The sublinear convergence rate of this 2P-ADMM is also established in terms of the iterative residual. A series of comparison experiments on testing the linear equation problem with artificial data, the graph-guided fused lasso problem and the robust principal component analysis problem with public datasets validate that the proposed algorithm outperforms several well-established algorithms. In the future work, we will focus on extending the proposed method to some stochastic versions and inexact versions for solving multi-block composite nonconvex optimization problems.

References

[1] A. Banerjee, S. Merugu, I. Dhillon, J. Ghosh, Clustering with Bregman divergences, J. Mach. Learn. Res., 6: 1705-1749, (2005)

[2] H. Bauschke, J. Bolte, M. Teboulle, A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications, Math. Oper. Res., 42: 277-575, (2017)

[3] J. Bai, J. Li, F. Xu, P. Dai, A novel method for a class of structured low rank minimization with equality constraint, J. Comput. Appl. Math., 330: 475-487, (2018)

[4] J. Bai, D. Han, H. Sun, H. Zhang, Convergence on a symmetric accelerated stochastic ADMM with larger stepsizes, CSIAM Trans. Appl. Math., 3: 448-479, (2022)

[5] J. Bai, K. Guo, J. Liang, Y. Jing, H. So, Accelerated symmetric ADMM and its applications in large-scale signal processing, J. Comput. Math., 42: 1605-1626, (2024)

[6] J. Bai, L. Jia, Z. Peng, A new insight on augmented Lagrangian method with applications in machine learning, J. Sci. Comput., 99: 53, (2024)

[7] J. Bai, M. Zhang, H. Zhang, An inexact ADMM for separable nonconvex and nonsmooth optimization, Comput. Optim. Appl., 90: 445-479, (2025)

[8] R. Barber, E. Sidky, Convergence for nonconvex ADMM, with applications to CT imaging, J. Machine Learn. Res., 25: 1-46, (2024)

[9] L. Bregman, The relaxation method of finding the common point of convex sets and itsapplication to the solution of problems in convex programming, USSR Comput. Math. Math. Phys., 7: 200-217, (1967)

[10] S. Brunton, J. Nathan Kutz, Machine Learning, Dynamical Systems, and Control, Cambridge University Press, Cambridge, (2019)

[11] E. Candes, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM. 58: 137, (2011)

[12] M. Chao, Y. Zhang, J. Jian, An inertial proximal alternating direction method of multipliers for nonconvex optimization, Int J. Comput. Math., 98(6): 1199-1217 (2021)

[13] X. Chen, C. Cui, D. Ren, Convergence of three-block ADMM for weakly convex optimization problems, SIAM J. Imaging Sci., 18: 449-493, (2025)

[14] Y. Dang, T. Cui, Convergence of linearized symmetric proximal ADMM for nonconvex nonseparable optimization, J. Sys. Sci. & Math. Scis., 43: 2949-2969, (2023)

[15] Y. Dang, X. Kun, J. Lu, Two-step inertial ADMM for the solution of nonconvex nonsmooth optimization problems with nonseparable structure, Phys. Scr., 100: 025235, (2025)

[16] J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso, Biostatistics, 9: 432-441, (2008)

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

[18] K. Guo, D. Han, T. Wu, Convergence of alternating direction method for minimizing sum of two nonconvex functions with linear constraints, INT J. Comput. Math., 94: 1653-1669, (2017)

[19] D. Han, W. Kong, W. Zhang, A partial splitting augmented Lagrangian method for low patch-rank image decomposition, J. Math. Imaging Vis., 51: 145-160, (2015)

[20] B. He, H. Xu, X. Yuan, On the proximal Jacobian decomposition of ALM for multiple-block separable convex minimization problems and its relationship to ADMM, J. Sci. Comput., 66: 1204-1217, (2016)

[21] B. He, X. Yuan, Balanced augmented Lagrangian method for convex programming, arXiv:2108.08554, (2021)

[22] M. Hestenes, Multiplier and gradient methods, J. Optim. Theory Appl. 4: 303-320, (1969)

[23] J. Jian, Q. Huang, J. Yin, W. Zhang, Splitting augmented Lagrangian-type algorithms with partial quadratic approximation to solve sparse signal recovery problems, J. Comput. Appl. Math., 449: 115972, (2024)

[24] M. Kang, M. Kang, M. Jung, Inexact accelerated augmented Lagrangian methods, Comput. Optim. Appl., 62: 373-404, (2015)

[25] Y. Ke, C. Ma, An accelerated augmented Lagrangian method for linearly constrained convex programming with the rate of convergence $\mathcal{O}(1/k^2)$, Appl. Math. (A Journal of Chinese Universities), 32: 117-126, (2017)

[26] J. Kim, A new Lagrangian-based first-order method for nonconvex constrained optimization, Oper. Res. Lett., 51: 357-363, (2024)

[27] J. Kim, A Lagrangian-based method with "false penalty" for linearly constrained nonconvex composite optimization, arXiv:2306.11299, (2023)

[28] W. Kong, J. Melo, R. Monteiro, Iteration-complexity of an inner accelerated inexact proximal augmented Lagrangian method based on the classical Lagrangian function, SIAM J. Optim. 33: 181-210, (2023)

[29] Y. Li, M. Zhao, W. Chen, Z. Wen, A stochastic composite augmented Lagrangian method for reinforcement learning, SIAM J. Optim., 33:921-949, (2023)

[30] Y. Liu, X. Liu, S. Ma, On the non-ergodic convergence rate of an inexact augmented Lagrangian framework for composite convex programming, Math. Oper. Res., 44: 632-650, (2019)

[31] P. Liu, J. Jian, B. He, X. Jiang, Convergence of Bregman Peaceman-Rachford splitting method for nonconvex nonseparable optimization, J. Oper. Res. Soc. China, 11: 707-733, (2023)

[32] P. Liu, J. Jian, H. Shao, X. Wang, J. Xu, X. Wu, A Bregman-style improved ADMM and its linearized version in the nonconvex setting: convergence and rate analyses, J. Oper. Res. Soc. China, 12: 298-340, (2024)

[33] Q. Liu, X. Shen, Y. Gu, Linearized ADMM for nonconvex nonsmooth optimization with convergence analysis, IEEE Access, 7: 76131-76144, (2019)

[34] J. Melo, R. Monteiro, Iteration-complexity of a linearized proximal multiblock ADMM class for linearly constrained nonconvex optimization problems, Optimization Online, https://optimization-online.org/wp-content/uploads/2017/04/5964.pdf, (2017)

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

[36] X. Qu, G. Yu, J. Liu, J. Chen, Z. Liu, A new penalty dual-primal augmented Lagrangian method and its extensions, Taiwanese J. Math., 28: 1223-1244, (2024)

[37] R. Rockafellar, R. Wets, Variational Analysis, Springer, New York, (1998)

[38] M. Tao, X. Yuan, An inexact parallel splitting augmented Lagrangian method for monotone variational inequalities with separable structures, Comput. Optim. Appl., 52: 439-461, (2012)

[39] H. Wang, A. Banerjee, Bregman alternating direction method of multipliers, NeurIPS, pp. 2816-2824, (2014)

[40] M. Wang, X. Cai, Y. Chen, Convergence analysis of split-Douglas-Rachford algorithm and a novel preconditioned ADMM with an improved condition, Numer. Math. Theor. Meth. Appl., 17: 658-696, (2024)

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

[42] C. Wang, M. Yan, Y. Rahimi, Y. Lou, Accelerated schemes for the L_1/L_2 minimization, IEEE Trans. Signal Process., 68: 2660-2669, (2020)

[43] Y. Xu, Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming, SIAM J. Optim., 27: 1459-1484, (2017)

[44] M. Yashtini, Convergence and rate analysis of a proximal linearized ADMM for nonconvex nonsmooth optimization, J. Global Optim., 84(4): 913-939, (2022)

[45] J. Yin, C. Tang, J. Jian, Q. Huang, A partial Bregman ADMM with a general relaxation factor for structured nonconvex and nonsmooth optimization, J. Global Optim., 89: 899-926, (2024)

[46] C. You, C. Li, D. Robinson, R. Vidal, Fast online l_0 elastic net subspace clustering via a novel dictionary update strategy, arXiv:2412.07335v2, (2024)