Computational Optimization and Applications (2019) 72:115–157
https://doi.org/10.1007/s10589-018-0034-y

Structured nonconvex and nonsmooth optimization: algorithms and iteration complexity analysis

Bo Jiang[1] · **Tianyi Lin**[2] · **Shiqian Ma**[3] · **Shuzhong Zhang**[4,5]

Received: 14 August 2017 / Published online: 25 September 2018

Abstract
Nonconvex and nonsmooth optimization problems are frequently encountered in much of statistics, business, science and engineering, but they are not yet widely recognized as a *technology* in the sense of scalability. A reason for this relatively low degree of popularity is the lack of a well developed system of theory and algorithms to support the applications, as is the case for its convex counterpart. This paper aims to take one step in the direction of *disciplined nonconvex and nonsmooth optimization*. In particular, we consider in this paper some constrained nonconvex optimization models in block decision variables, with or without coupled affine constraints. In the absence of coupled constraints, we show a sublinear rate of convergence to an ϵ-stationary solution in the form of variational inequality for a generalized conditional gradient method, where the convergence rate is dependent on the Hölderian continuity of the gradient of the smooth part of the objective. For the model with coupled affine constraints, we introduce corresponding ϵ-stationarity conditions, and apply two proximal-type variants of the ADMM to solve such a model, assuming the proximal ADMM updates can be implemented for all the block variables except for the last block, for which either a gradient step or a majorization–minimization step is implemented. We show an iteration complexity bound of $O(1/\epsilon^2)$ to reach an ϵ-stationary solution for both algorithms. Moreover, we show that the same iteration complexity of a proximal BCD method follows immediately. Numerical results are provided to illustrate the efficacy of the proposed algorithms for tensor robust PCA and tensor sparse PCA problems.

Keywords Structured nonconvex optimization · ϵ-Stationary solution · Iteration complexity · Conditional gradient method · Alternating direction method of multipliers · Block coordinate descent method

---

Bo Jiang: Research of this author was supported in part by NSFC Grants 11771269 and 11831002, and Program for Innovative Research Team of Shanghai University of Finance and Economics. Shiqian Ma: Research of this author was supported in part by a startup package in Department of Mathematics at UC Davis. Shuzhong Zhang: Research of this author was supported in part by the National Science Foundation (Grant CMMI-1462408), and in part by Shenzhen Fundamental Research Fund under Grant No. KQTD2015033114415450.

Extended author information available on the last page of the article

Mathematics Subject Classification 90C26 · 90C06 · 90C60

1 Introduction

In this paper, we consider the following nonconvex and nonsmooth optimization problem with multiple block variables:

$$\begin{aligned} &\min\ f(x_1, x_2, \ldots, x_N) + \sum_{i=1}^{N-1} r_i(x_i) \\ &\text{s.t.}\ \ \sum_{i=1}^{N} A_i x_i = b,\ \ x_i \in \mathcal{X}_i,\ i = 1, \ldots, N-1, \end{aligned} \tag{1.1}$$

where f is differentiable and possibly nonconvex, and each r_i is possibly nonsmooth and nonconvex, $i = 1, \ldots, N-1$; $A_i \in \mathbb{R}^{m \times n_i}$, $b \in \mathbb{R}^m$, $x_i \in \mathbb{R}^{n_i}$; and $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ are convex sets, $i = 1, 2, \ldots, N-1$. One restriction of model (1.1) is that the objective function is required to be smooth with respect to the last block variable x_N. However, in Sect. 4 we shall extend the result to cover the general case where $r_N(x_N)$ may be present and that x_N maybe constrained as well. A special case of (1.1) is when the affine constraints are absent, and there is no block structure of the variables (i.e., $x = x_1$ and other block variables do not show up in (1.1)), which leads to the following more compact form

$$\min\ \Phi(x) := f(x) + r(x),\ \text{s.t.}\ x \in S \subset \mathbb{R}^n, \tag{1.2}$$

where S is a convex and compact set. In this paper, we propose several first-order algorithms for computing an ϵ-stationary solution (to be defined later) for (1.1) and (1.2), and analyze their iteration complexities. Throughout, we assume the following condition.

Assumption 1.1 The sets of the stationary solutions for (1.1) and (1.2) are non-empty.

Problem (1.1) arises from a variety of interesting applications. For example, one of the nonconvex models for matrix robust PCA can be cast as follows (see, e.g., [51]), which seeks to decompose a given matrix $M \in \mathbb{R}^{m \times n}$ into a superposition of a low-rank matrix Z, a sparse matrix E and a noise matrix B:

$$\min_{X,Y,Z,E,B}\ \|Z - XY^\top\|_F^2 + \alpha \mathcal{R}(E),\ \text{s.t.}\ M = Z + E + B,\ \|B\|_F \le \eta, \tag{1.3}$$

where $X \in \mathbb{R}^{m \times r}$, $Y \in \mathbb{R}^{n \times r}$, with $r < \min(m, n)$ being the estimated rank of Z; $\eta > 0$ is the noise level, $\alpha > 0$ is a weighting parameter; $\mathcal{R}(E)$ is a regularization function that can improve the sparsity of E. One of the widely used regularization functions is the ℓ_1 norm, which is convex and nonsmooth. However, there are also many nonconvex regularization functions that are widely used in statistical learning and information theory, such as smoothly clipped absolute deviation (SCAD) [23], log-sum penalty (LSP) [15], minimax concave penalty (MCP) [58], and capped-ℓ_1

penalty [59,60], and they are nonsmooth at point 0 if composed with the absolute value function, which is usually the case in statistical learning. Clearly (1.3) is in the form of (1.1). Another example of the form (1.1) is the following nonconvex tensor robust PCA model (see, e.g., [55]), which seeks to decompose a given tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ into a superposition of a low-rank tensor $\mathcal{Z}$, a sparse tensor $\mathcal{E}$ and a noise tensor $\mathcal{B}$:

$$\begin{aligned} \min_{X_i, \mathcal{C}, \mathcal{Z}, \mathcal{E}, \mathcal{B}} \; & \|\mathcal{Z} - \mathcal{C} \times_1 X_1 \times_2 X_2 \times_3 \cdots \times_d X_d\|_F^2 + \alpha \mathcal{R}(\mathcal{E}), \\ \text{s.t. } & \mathcal{T} = \mathcal{Z} + \mathcal{E} + \mathcal{B}, \; \|\mathcal{B}\|_F \le \eta, \end{aligned}$$

where $\mathcal{C}$ is the core tensor that has a smaller size than $\mathcal{Z}$, and X_i are matrices with appropriate sizes, $i = 1, \ldots, d$. In fact, the "low-rank" tensor in the above model corresponds to the tensor with a small core; however a recent work [35] demonstrates that the CP-rank of the core regardless of its size could be as large as the original tensor. Therefore, if one wants to find the low CP-rank decomposition, then the following model is preferred:

$$\min_{X_i, \mathcal{Z}, \mathcal{E}, \mathcal{B}} \|\mathcal{Z} - [\![X_1, X_2, \ldots, X_d]\!]\|_F^2 + \alpha\, \mathcal{R}(\mathcal{E}) + \alpha_{\mathcal{N}} \|\mathcal{B}\|_F^2, \; \text{s.t. } \mathcal{T} = \mathcal{Z} + \mathcal{E} + \mathcal{B},$$

for $X_i = [a^{i,1}, a^{i,2}, \ldots, a^{i,R}] \in \mathbb{R}^{n_i \times R}$, $1 \le i \le d$ and $[\![X_1, X_2, \ldots, X_d]\!] := \sum_{r=1}^{R} a^{1,r} \otimes a^{2,r} \otimes \cdots \otimes a^{d,r}$, where "$\otimes$" denotes the outer product of vectors, and R is an estimation of the CP-rank. In addition, the so-called sparse tensor PCA problem [1], which seeks the best sparse rank-one approximation for a given d-th order tensor $\mathcal{T}$, can also be formulated in the form of (1.1):

$$\min \; -\mathcal{T}(x_1, x_2, \ldots, x_d) + \alpha \sum_{i=1}^{d} \mathcal{R}(x_i), \; \text{s.t. } x_i \in S_i = \{x \mid \|x\|_2^2 \le 1\}, \; i = 1, 2, \ldots, d, \tag{1.4}$$

where $\mathcal{T}(x_1, x_2, \ldots, x_d) = \sum_{i_1,\ldots,i_d} \mathcal{T}_{i_1,\ldots,i_d} (x_1)_{i_1} \ldots (x_d)_{i_d}$.

The convergence and iteration complexity for various nonconvex and nonsmooth optimization problems have recently attracted considerable research attention; see e.g. [3,6,7,10,11,19,20,27,28,41,46]. In this paper, we study several solution methods that use only the first-order information of the objective function, including a generalized conditional gradient method, variants of alternating direction method of multipliers, and a proximal block coordinate descent method, for solving (1.1) and (1.2). Specifically, we apply a generalized conditional gradient (GCG) method to solve (1.2). We prove that the GCG can find an ϵ-stationary solution for (1.2) in $O(\epsilon^{-q})$ iterations under certain mild conditions, where q is a parameter in the Hölder condition that characterizes the degree of smoothness for f. In other words, the convergence rate of the algorithm depends on the degree of "smoothness" of the objective function. It should be noted that a similar iteration bound that depends on the parameter q was reported for convex problems [13], and for general nonconvex problem, [14] analyzed the convergence results, but there was no iteration complexity result. Furthermore, we show that if f is concave, then GCG finds an ϵ-stationary solution for (1.2) in

$O(1/\epsilon)$ iterations. For the affinely constrained problem (1.1), we propose two algorithms (called proximal ADMM-g and proximal ADMM-m in this paper), both can be viewed as variants of the alternating direction method of multipliers (ADMM). Recently, there has been an emerging research interest on the ADMM for nonconvex problems (see, e.g., [2,32,33,38,52,53,56]). However, the results in [38,52,53,56] only show that the iterates produced by the ADMM converge to a stationary solution without providing an iteration complexity analysis. Moreover, the objective function is required to satisfy the so-called Kurdyka–Łojasiewicz (KL) property [8,9,36,42] to enable those convergence results. In [33], Hong et al. analyzed the convergence of the ADMM for solving nonconvex consensus and sharing problems. Note that they also analyzed the iteration complexity of the ADMM for the consensus problem. However, they require the nonconvex part of the objective function to be smooth, and nonsmooth part to be convex. In contrast, r_i in our model (1.1) can be nonconvex and nonsmooth at the same time. Moreover, we allow general constraints $x_i \in \mathcal{X}_i, i = 1, \ldots, N-1$, while the consensus problem in [33] only allows such constraint for one block variable. A very recent work of Hong [32] discussed the iteration complexity of an augmented Lagrangian method for finding an ϵ-stationary solution for the following problem:

$$\min \ f(x), \ \text{s.t.} \ Ax = b, x \in \mathbb{R}^n, \tag{1.5}$$

under the assumption that f is differentiable. We will compare our results with [32] in more details in Sect. 3.

Before proceeding, let us first summarize:

Our contributions

(i) We provide definitions of ϵ-stationary solution for (1.1) and (1.2) using the variational inequalities. For (1.1), our definition of the ϵ-stationary solution allows each r_i to be nonsmooth and nonconvex.

(ii) We study a generalized conditional gradient method with a suitable line search rule for solving (1.2). We assume that the gradient of f satisfies a Hölder condition, and analyze its iteration complexity for obtaining an ϵ-stationary solution for (1.2). After we released the first version of this paper, we noticed there are several recent works that study the iteration complexity of conditional gradient method for nonconvex problems. However, our results are different from these. For example, the convergence rate given in [57] is worse than ours, and [43,44] only consider smooth nonconvex problem with Lipschitz continuous gradient, but our results cover nonsmooth models.

(iii) We study two ADMM variants (proximal ADMM-g and proximal ADMM-m) for solving (1.1), and analyze their iteration complexities for obtaining an ϵ-stationary solution for nonconvex problem (1.1). In addition, the setup and the assumptions of our model are different from other recent works. For instance, [38] considers a two-block nonconvex problem with an identity coefficient matrix for one block variable in the linear constraint, and requires the coerciveness of the objective or the boundedness of the domain. [53] assumes that the objective function is coercive over the feasible set and the nonsmooth objective is *restricted prox-regular* or *piece-wise linear*. While our algorithm assumes the gradient of the smooth part of the objective function is Lipschitz continuous and the nonsmooth part does not involve the last block variable, which is weaker than the assumptions on the objective functions in [38,53].

(iv) As an extension, we also show how to use proximal ADMM-g and proximal ADMM-m to find an ϵ-stationary solution for (1.1) without assuming any condition on A_N.

(v) When the affine constraints are absent in model (1.1), as a by-product, we demonstrate that the iteration complexity of proximal block coordinate descent (BCD) method with cyclic order can be obtained directly from that of proximal ADMM-g and proximal ADMM-m. Although [11] gives an iteration complexity result of nonconvex BCD, it requires the KL property, and the complexity depends on a parameter in the KL condition, which is typically unknown.

Notation $\|x\|_2$ denotes the Euclidean norm of vector x, and $\|x\|_H^2$ denotes $x^\top Hx$ for some positive definite matrix H. For set S and scalar $p > 1$, we denote $\mathrm{diam}_p(S) := \max_{x,y\in S}\|x-y\|_p$, where $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$. Without specification, we denote $\|x\| = \|x\|_2$ and $\mathrm{diam}(S) = \mathrm{diam}_2(S)$ for short. We use $\mathrm{dist}(x, S)$ to denote the Euclidean distance of vector x to set S. Given a matrix A, its spectral norm and smallest singular value are denoted by $\|A\|_2$ and $\sigma_{\min}(A)$ respectively. We use $\lceil a\rceil$ to denote the ceiling of a.

Organization The rest of this paper is organized as follows. In Sect. 2 we introduce the notion of ϵ-stationary solution for (1.2) and apply a generalized conditional gradient method to solve (1.2) and analyze its iteration complexity for obtaining an ϵ-stationary solution for (1.2). In Sect. 3 we give two definitions of ϵ-stationarity for (1.1) under different settings and propose two ADMM variants that solve (1.1) and analyze their iteration complexities to reach an ϵ-stationary solution for (1.1). In Sect. 4 we provide some extensions of the results in Sect. 3. In particular, we first show how to remove some of the conditions that we assume in Sect. 3, and then we apply a proximal BCD method to solve (1.1) without affine constraints and provide an iteration complexity analysis. In Sect. 5, we present numerical results to illustrate the practical efficiency of the proposed algorithms.

2 A generalized conditional gradient method

In this section, we study a GCG method for solving (1.2) and analyze its iteration complexity. The conditional gradient (CG) method, also known as the Frank-Wolfe method, was originally proposed in [24], and regained a lot of popularity recently due to its capability in solving large-scale problems (see, [4,5,25,30,34,37,47]). However, these works focus on solving convex problems. Bredies et al. [14] proved the convergence of a generalized conditional gradient method for solving nonconvex problems in Hilbert space. In this section, by introducing a suitable line search rule, we provide an iteration complexity analysis for this algorithm.

We make the following assumption in this section regarding (1.2).

Assumption 2.1 In (1.2), $r(x)$ is convex and nonsmooth, and the constraint set S is convex and compact. Moreover, f is differentiable and there exist some $p > 1$ and

$\rho > 0$ such that

$$f(y) \le f(x) + \nabla f(x)^\top (y-x) + \frac{\rho}{2}\|y-x\|_p^p, \quad \forall x, y \in S. \tag{2.1}$$

The above inequality (2.1) is also known as the Hölder condition and was used in other works on first-order algorithms (e.g., [21]). It can be shown that (2.1) holds for a variety of functions. For instance, (2.1) holds for any p when f is concave, and is valid for $p = 2$ when ∇f is Lipschitz continuous.

2.1 An ϵ-stationary solution for problem (1.2)

For smooth unconstrained problem $\min_x f(x)$, it is natural to define the ϵ-stationary solution using the criterion $\|\nabla f(x)\|_2 \le \epsilon$. Nesterov [48] and Cartis et al. [17] showed that the gradient descent type methods with properly chosen step size need $O(1/\epsilon^2)$ iterations to find such a solution. Moreover, Cartis et al. [16] constructed an example showing that the $O(1/\epsilon^2)$ iteration complexity is tight for the steepest descent type algorithm. However, the case for the constrained nonsmooth nonconvex optimization is subtler. There exist some works on how to define ϵ-optimality condition for the local minimizers of various constrained nonconvex problems [18,22,28,32,49]. Cartis et al. [18] proposed an approximate measure for smooth problem with convex set constraint. [49] discussed general nonsmooth nonconvex problem in Banach space by using the tool of limiting Fréchet ϵ-subdifferential. Ngai et al. [22] showed that under certain conditions ϵ-KKT solutions can converge to a stationary solution as $\epsilon \to 0$. Here the ϵ-KKT solution is defined by relaxing the complimentary slackness and equilibrium equations of KKT conditions. Ghadimi et al. [28] considered the following notion of ϵ-stationary solution for (1.2):

$$P_S(x,\gamma) := \frac{1}{\gamma}(x - x^+), \quad \text{where } x^+ = \arg\min_{y\in S} \nabla f(x)^\top y + \frac{1}{\gamma} V(y,x) + r(y), \tag{2.2}$$

where $\gamma > 0$ and V is a prox-function. They proposed a projected gradient algorithm to solve (1.2) and proved that it takes no more than $O(1/\epsilon^2)$ iterations to find an x satisfying

$$\|P_S(x,\gamma)\|_2^2 \le \epsilon. \tag{2.3}$$

Our definition of an ϵ-stationary solution for (1.2) is as follows.

Definition 2.2 We call x an ϵ-stationary solution ($\epsilon \ge 0$) for (1.2) if the following holds:

$$\psi_S(x) := \inf_{y\in S}\{\nabla f(x)^\top (y-x) + r(y) - r(x)\} \ge -\epsilon. \tag{2.4}$$

If $\epsilon = 0$, then x is called a stationary solution for (1.2).

Observe that if $r(\cdot)$ is continuous then any cluster point of ϵ-stationary solutions defined above is a stationary solution for (1.2) as $\epsilon \to 0$. Moreover, the stationarity condition is weaker than the usual KKT optimality condition. To see this, we first rewrite (1.2) as the following equivalent unconstrained problem

$$\min_x f(x) + r(x) + \iota_S(x)$$

where $\iota_S(x)$ is the indicator function of S. Suppose that x is any local minimizer of this problem and thus also a local minimizer of (1.2). Since f is differentiable, r and ι_S are convex, Fermat's rule [50] yields

$$0 \in \partial\left(f(x) + r(x) + \iota_S(x)\right) = \nabla f(x) + \partial r(x) + \partial \iota_S(x), \tag{2.5}$$

which further implies that there exists some $z \in \partial r(x)$ such that

$$(\nabla f(x) + z)^\top (y - x) \geq 0, \quad \forall y \in S.$$

Using the convexity of $r(\cdot)$, it is equivalent to

$$\nabla f(x)^\top (y - x) + r(y) - r(x) \geq 0, \;\; \forall y \in S. \tag{2.6}$$

Therefore, (2.6) is a necessary condition for local minimum of (1.2) as well.

Furthermore, we claim that $\psi_S(x) \geq -\epsilon$ implies $\|P_S(x,\gamma)\|_2^2 \leq \epsilon/\gamma$ with the prox-function $V(y,x) = \|y - x\|_2^2/2$. In fact, (2.2) guarantees that

$$\left(\nabla f(x) + \frac{1}{\gamma}(x^+ - x) + z\right)^\top (y - x^+) \geq 0, \quad \forall\ y \in S, \tag{2.7}$$

for some $z \in \partial r(x^+)$. By choosing $y = x$ in (2.7) one obtains

$$\nabla f(x)^\top (x - x^+) + r(x) - r(x^+) \geq (\nabla f(x) + z)^\top (x - x^+) \geq \frac{1}{\gamma}\|x^+ - x\|_2^2. \tag{2.8}$$

Therefore, if $\psi_S(x) \geq -\epsilon$, then $\|P_S(x,\gamma)\|_2^2 \leq \frac{\epsilon}{\gamma}$ holds.

2.2 The algorithm

For given point z, we define an approximation of the objective function of (1.2) to be:

$$\ell(y; x) := f(x) + \nabla f(x)^\top (y - x) + r(y), \tag{2.9}$$

which is obtained by linearizing the smooth part (function f) of Φ in (1.2). Our GCG method for solving (1.2) is described in Algorithm 1, where ρ and p are from Assumption 2.1.

Algorithm 1 Generalized Conditional Gradient Algorithm (GCG) for solving (1.2)

Require: Given $x^0 \in S$
for $k = 0, 1, \ldots$ **do**
[Step 1] $y^k = \arg\min_{y \in S} \ell(y; x^k)$, and let $d^k = y^k - x^k$;
[Step 2] $\alpha_k = \arg\min_{\alpha \in [0,1]} \alpha \nabla f(x^k)^\top d^k + \alpha^p \frac{\rho}{2} \|d^k\|_p^p + (1-\alpha) r(x^k) + \alpha r(y^k)$;
[Step 3] Set $x^{k+1} = (1-\alpha_k) x^k + \alpha_k y^k$.
end for

In each iteration of Algorithm 1, we first perform an exact minimization on the approximated objective function $\ell(y; x)$ to form a direction d_k. Then the step size α_k is obtained by an exact line search (which differentiates the GCG from a normal CG method) along the direction d_k, where f is approximated by p-powered function and the nonsmooth part is replaced by its upper bound. Finally, the iterate is updated by moving along the direction d_k with step size α_k.

Note that here we assumed that solving the subproblem in Step 1 of Algorithm 1 is relatively easy. That is, we assumed the following assumption.

Assumption 2.3 All subproblems in Step 1 of Algorithm 1 can be solved relatively easily.

Remark 2.4 Assumption 2.3 is quite common in conditional gradient method. For a list of functions r and sets S such that Assumption 2.3 is satisfied, see [34].

Remark 2.5 It is easy to see that the sequence $\{\Phi(x^k)\}$ generated by GCG is monotonically nonincreasing [14], which implies that any cluster point of $\{x^k\}$ cannot be a strict local maximizer.

2.3 An iteration complexity analysis

Before we proceed to the main result on iteration complexity of GCG, we need the following lemma that gives a sufficient condition for an ϵ-stationary solution for (1.2). This lemma is inspired by [27], and it indicates that if the progress gained by minimizing (2.9) is small, then z must already be close to a stationary solution for (1.2).

Lemma 2.6 *Define $z_\ell := \operatorname{argmin}_{x \in S} \ \ell(x; z)$. The improvement of the linearization at point z is defined as*

$$\triangle \ell_z := \ell(z; z) - \ell(z_\ell; z) = -\nabla f(z)^\top (z_\ell - z) + r(z) - r(z_\ell).$$

Given $\epsilon \geq 0$, for any $z \in S$, if $\triangle \ell_z \leq \epsilon$, then z is an ϵ-stationary solution for (1.2) as defined in Definition 2.2.

Proof From the definition of z_ℓ, we have

$$\ell(y; z) - \ell(z_\ell; z) = \nabla f(z)^\top (y - z_\ell) + r(y) - r(z_\ell) \geq 0, \forall y \in S,$$

which implies that

$$
\begin{aligned}
&\nabla f(z)^\top (y-z) + r(y) - r(z) \\
&\quad = \nabla f(z)^\top (y - z_\ell) + r(y) - r(z_\ell) + \nabla f(z)^\top (z_\ell - z) + r(z_\ell) - r(z) \\
&\quad \geq \nabla f(z)^\top (z_\ell - z) + r(z_\ell) - r(z), \forall y \in S.
\end{aligned}
$$

It then follows immediately that if $\triangle \ell_z \leq \epsilon$, then $\nabla f(z)^\top (y-z) + r(y) - r(z) \geq -\triangle \ell_z \geq -\epsilon$. □

Denoting Φ^* to be the optimal value of (1.2), we are now ready to give the main result of the iteration complexity of GCG (Algorithm 1) for obtaining an ϵ-stationary solution for (1.2).

Theorem 2.7 *For any $\epsilon \in (0, \mathrm{diam}_p^p(S)\rho)$, GCG finds an ϵ-stationary solution for* (1.2) *within $\left\lceil \frac{2(\Phi(x^0)-\Phi^*)(\mathrm{diam}_p^p(S)\rho)^{q-1}}{\epsilon^q} \right\rceil$ iterations, where $\frac{1}{p} + \frac{1}{q} = 1$.*

Proof For ease of presentation, we denote $D := \mathrm{diam}_p(S)$ and $\triangle \ell^k := \triangle \ell_{x^k}$. By Assumption 2.1, using the fact that $\frac{\epsilon}{D^p \rho} < 1$, and by the definition of α_k in Algorithm 1, we have

$$
\begin{aligned}
&(\epsilon/(D^p\rho))^{\frac{1}{p-1}} \triangle \ell^k - \frac{1}{2\rho^{1/(p-1)}} (\epsilon/D)^{\frac{p}{p-1}} \\
&\quad \leq -(\epsilon/(D^p\rho))^{\frac{1}{p-1}} (\nabla f(x^k)^\top (y^k - x^k) + r(y^k) - r(x^k)) \\
&\qquad - \frac{\rho}{2} (\epsilon/(D^p\rho))^{\frac{p}{p-1}} \|y^k - x^k\|_p^p \\
&\quad \leq -\alpha_k \left(\nabla f(x^k)^\top (y^k - x^k) + r(y^k) - r(x^k) \right) - \frac{\rho \alpha_k^p}{2} \|y^k - x^k\|_p^p \\
&\quad \leq -\nabla f(x^k)^\top (x^{k+1} - x^k) + r(x^k) - r(x^{k+1}) - \frac{\rho}{2} \|x^{k+1} - x^k\|_p^p \\
&\quad \leq f(x^k) - f(x^{k+1}) + r(x^k) - r(x^{k+1}) = \Phi(x^k) - \Phi(x^{k+1}),
\end{aligned}
\tag{2.10}
$$

where the third inequality is due to the convexity of function r and the fact that $x^{k+1} - x^k = \alpha_k (y^k - x^k)$, and the last inequality is due to (2.1). Furthermore, (2.10) immediately yields

$$
\triangle \ell^k \leq (\epsilon/(D^p\rho))^{-\frac{1}{p-1}} (\Phi(x^k) - \Phi(x^{k+1})) + \frac{\epsilon}{2}. \tag{2.11}
$$

For any integer $K > 0$, summing (2.11) over $k = 0, 1, \ldots, K-1$, yields

$$
\begin{aligned}
K \min_{k \in \{0,1,\ldots,K-1\}} \triangle \ell^k &\leq \sum_{k=0}^{K-1} \triangle \ell^k \leq (\epsilon/(D^p\rho))^{-\frac{1}{p-1}} \left(\Phi(x^0) - \Phi(x^K) \right) + \frac{\epsilon}{2} K \\
&\leq (\epsilon/(D^p\rho))^{-\frac{1}{p-1}} (\Phi(x^0) - \Phi^*) + \frac{\epsilon}{2} K,
\end{aligned}
$$

where Φ^* is the optimal value of (1.2). It is easy to see that by setting $K = \left\lceil \frac{2(\Phi(x^0)-\Phi^*)(D^p\rho)^{q-1}}{\epsilon^q} \right\rceil$, the above inequality implies $\triangle\ell_{x^{k^*}} \le \epsilon$, where $k^* \in \mathrm{argmin}_{k\in\{0,\ldots,K-1\}} \triangle\ell^k$. According to Lemma 2.6, x^{k^*} is an ϵ-stationary solution for (1.2) as defined in Definition 2.2. □

Finally, if f is concave, then the iteration complexity can be improved as $O(1/\epsilon)$.

Proposition 2.8 *Suppose that f is a concave function. If we set $\alpha_k = 1$ for all k in GCG (Algorithm 1), then it returns an ϵ-stationary solution for (1.2) within $\left\lceil \frac{\Phi(x^0)-\Phi^*}{\epsilon} \right\rceil$ iterations.*

Proof By setting $\alpha_k = 1$ in Algorithm 1 we have $x^{k+1} = y^k$ for all k. Since f is concave, it holds that

$$\triangle\ell^k = -\nabla f(x^k)^\top(x^{k+1} - x^k) + r(x^k) - r(x^{k+1}) \le \Phi(x^k) - \Phi(x^{k+1}).$$

Summing this inequality over $k = 0, 1, \ldots, K-1$ yields $K \min_{k\in\{0,1,\ldots,K-1\}} \triangle\ell^k \le \Phi(x^0) - \Phi^*$, which leads to the desired result immediately. □

3 Variants of ADMM for solving nonconvex problems with affine constraints

In this section, we study two variants of the ADMM (Alternating Direction Method of Multipliers) for solving the general problem (1.1), and analyze their iteration complexities for obtaining an ϵ-stationary solution (to be defined later) under certain conditions. Throughout this section, the following two assumptions regarding problem (1.1) are assumed.

Assumption 3.1 The gradient of the function f is Lipschitz continuous with Lipschitz constant $L > 0$, i.e., for any $(x_1^1, \ldots, x_N^1)$ and $(x_1^2, \ldots, x_N^2) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_{N-1} \times \mathbb{R}^{n_N}$, it holds that

$$\begin{aligned} &\left\| \nabla f(x_1^1, x_2^1, \ldots, x_N^1) - \nabla f(x_1^2, x_2^2, \ldots, x_N^2) \right\| \\ &\quad \le L \left\| \left(x_1^1 - x_1^2, x_2^1 - x_2^2, \ldots, x_N^1 - x_N^2 \right) \right\|, \end{aligned} \tag{3.1}$$

which implies that for any $(x_1, \ldots, x_{N-1}) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_{N-1}$ and $x_N, \hat{x}_N \in \mathbb{R}^{n_N}$, we have

$$\begin{aligned} f(x_1, \ldots, x_{N-1}, x_N) \le\ & f(x_1, \ldots, x_{N-1}, \hat{x}_N) + (x_N - \hat{x}_N)^\top \nabla_N f(x_1, \ldots, x_{N-1}, \hat{x}_N) \\ & + \frac{L}{2}\|x_N - \hat{x}_N\|^2. \end{aligned} \tag{3.2}$$

Assumption 3.2 f and r_i, $i = 1, \ldots, N-1$ are all lower bounded over the appropriate domains defined via the sets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_{N-1}, \mathbb{R}^{n_N}$, and we denote

$$f^* = \inf_{x_i \in \mathcal{X}_i, i=1,\ldots,N-1; x_N \in \mathbb{R}^{n_N}} \{f(x_1, x_2, \ldots, x_N)\}$$

and $r_i^* = \inf_{x_i \in \mathcal{X}_i} \{r_i(x_i)\}$ for $i = 1, 2, \ldots, N-1$.

3.1 Preliminaries

To characterize the optimality conditions for (1.1) when r_i is nonsmooth and nonconvex, we need to recall the notion of the generalized gradient (see, e.g., [50]).

Definition 3.3 Let $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper lower semi-continuous function. Suppose $h(\bar{x})$ is finite for a given $\bar{x}$. For $v \in \mathbb{R}^n$, we say that

(i). v is a regular subgradient (also called Fréchet subdifferential) of h at $\bar{x}$, written $v \in \hat{\partial} h(\bar{x})$, if

$$\lim_{x \neq \bar{x}} \inf_{x \to \bar{x}} \frac{h(x) - h(\bar{x}) - \langle v, x - \bar{x} \rangle}{\|x - \bar{x}\|} \geq 0;$$

(ii). v is a general subgradient of h at $\bar{x}$, written $v \in \partial h(\bar{x})$, if there exist sequences $\{x^k\}$ and $\{v^k\}$ such that $x^k \to \bar{x}$ with $h(x^k) \to h(\bar{x})$, and $v^k \in \hat{\partial} h(x^k)$ with $v^k \to v$ when $k \to \infty$.

The following proposition lists some well-known facts about the lower semi-continuous functions.

Proposition 3.4 *Let $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be proper lower semi-continuous functions. Then it holds that:*

(i) *(Theorem 10.1 in* [50]*) Fermat's rule remains true: if $\bar{x}$ is a local minimum of h, then $0 \in \partial h(\bar{x})$.*
(ii) *If $h(\cdot)$ is continuously differentiable at x, then $\partial(h + g)(x) = \nabla h(x) + \partial g(x)$.*
(iii) *(Exercise 10.10 in* [50]*) If h is locally Lipschitz continuous at x, then $\partial(h + g)(x) \subset \partial h(x) + \partial g(x)$.*
(iv) *Suppose $h(x)$ is locally Lipschitz continuous, X is a closed and convex set, and $\bar{x}$ is a local minimum of h on X. Then there exists $v \in \partial h(\bar{x})$ such that $(x - \bar{x})^\top v \geq 0, \forall x \in X$.*

In our analysis, we frequently use the following identity that holds for any vectors a, b, c, d,

$$(a - b)^\top (c - d) = \frac{1}{2} \left(\|a - d\|_2^2 - \|a - c\|_2^2 + \|b - c\|_2^2 - \|b - d\|_2^2 \right). \tag{3.3}$$

Table 1 ϵ-stationary solution of (1.1) in two settings

	$r_i, i=1,\dots,N-1$	$\mathcal{X}_i, i=1,\dots,N-1$	ϵ-stationary solution
Setting 1	Lipschitz continuous	$\mathcal{X}_i \subset \mathbb{R}^{n_i}$ compact	Definition 3.5
Setting 2	Lower semi-continuous	$\mathcal{X}_i = \mathbb{R}^{n_i}$	Definition 3.6

3.2 An ϵ-stationary solution for problem (1.1)

We now introduce notions of ϵ-stationarity for (1.1) under the following two settings: (i) **Setting 1**: r_i is Lipschitz continuous, and $\mathcal{X}_i$ is a compact set, for $i=1,\dots,N-1$; (ii) **Setting 2**: r_i is lower semi-continuous, and $\mathcal{X}_i = \mathbb{R}^{n_i}$, for $i=1,\dots,N-1$.

Definition 3.5 (*ϵ-stationary solution for* (1.1) *in Setting 1*) Under the conditions in **Setting 1**, for $\epsilon \ge 0$, we call $\left(x_1^*,\dots,x_N^*\right)$ an ϵ-stationary solution for (1.1) if there exists a Lagrange multiplier λ^* such that the following holds for any $(x_1,\dots,x_N) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_{N-1} \times \mathbb{R}^{n_N}$:

$$\left(x_i - x_i^*\right)^\top \left[g_i^* + \nabla_i f(x_1^*,\dots,x_N^*) - A_i^\top \lambda^*\right] \ge -\epsilon, \quad i=1,\dots,N-1, \tag{3.4}$$

$$\left\|\nabla_N f(x_1^*,\dots,x_{N-1}^*,x_N^*) - A_N^\top \lambda^*\right\| \le \epsilon, \tag{3.5}$$

$$\left\|\sum_{i=1}^N A_i x_i^* - b\right\| \le \epsilon, \tag{3.6}$$

where g_i^* is a general subgradient of r_i at point x_i^*. If $\epsilon = 0$, we call $\left(x_1^*,\dots,x_N^*\right)$ a stationary solution for (1.1).

If $\mathcal{X}_i = \mathbb{R}^{n_i}$ for $i=1,\dots,N-1$, then the VI style conditions in Definition 3.5 reduce to the following.

Definition 3.6 [ϵ-stationary solution for (1.1) in Setting 2] Under the conditions in **Setting 2**, for $\epsilon \ge 0$, we call $\left(x_1^*,\dots,x_N^*\right)$ to be an ϵ-stationary solution for (1.1) if there exists a Lagrange multiplier λ^* such that (3.5), (3.6) and the following holds for any $(x_1,\dots,x_N) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_{N-1} \times \mathbb{R}^{n_N}$:

$$\operatorname{dist}\left(-\nabla_i f(x_1^*,\dots,x_N^*) + A_i^\top \lambda^*, \partial r_i(x_i^*)\right) \le \epsilon, \quad i=1,\dots,N-1, \tag{3.7}$$

where $\partial r_i(x_i^*)$ is the general subgradient of r_i at x_i^*, $i=1,2,\dots,N-1$. If $\epsilon=0$, we call $\left(x_1^*,\dots,x_N^*\right)$ to be a stationary solution for (1.1).

The two settings of problem (1.1) considered in this section and their corresponding definitions of ϵ-stationary solution, are summarized in Table 1.

A very recent work of Hong [32] proposes a definition of an ϵ-stationary solution for problem (1.5), and analyzes the iteration complexity of a proximal augmented

Lagrangian method for obtaining such a solution. Specifically, (x^*, λ^*) is called an ϵ-stationary solution for (1.5) in [32] if $Q(x^*, \lambda^*) \le \epsilon$, where

$$Q(x, \lambda) := \|\nabla_x \mathcal{L}_\beta(x, \lambda)\|^2 + \|Ax - b\|^2,$$

and $\mathcal{L}_\beta(x, \lambda) := f(x) - \lambda^\top(Ax - b) + \frac{\beta}{2}\|Ax - b\|^2$ is the augmented Lagrangian function of (1.5). Note that [32] assumes that f is differentiable and has bounded gradient in (1.5). It is easy to show that an ϵ-stationary solution in [32] is equivalent to an $O(\sqrt{\epsilon})$-stationary solution for (1.1) according to Definition 3.6 with $r_i = 0$ and f being differentiable. Note that there is no set constraint in (1.5), and so the notion of the ϵ-stationarity in [32] is not applicable in the case of Definition 3.5.

Proposition 3.7 *Consider the ϵ-stationary solution in Definition 3.6 applied to problem (1.5), i.e., one block variable and $r_i(x) = 0$. Then x^* is a $\gamma_1\sqrt{\epsilon}$-stationary solution in Definition 3.6, with Lagrange multiplier λ^* and $\gamma_1 = 1/(\sqrt{2\beta^2\|A\|_2^2 + 3})$, implies $Q(x^*, \lambda^*) \le \epsilon$. On the contrary, if $Q(x^*, \lambda^*) \le \epsilon$, then x^* is a $\gamma_2\sqrt{\epsilon}$-stationary solution from Definition 3.6 with Lagrange multiplier λ^*, where $\gamma_2 = \sqrt{2(1 + \beta^2\|A\|_2^2)}$.*

Proof Suppose x^* is a $\gamma_1\sqrt{\epsilon}$-stationary solution as defined in Definition 3.6. We have $\|\nabla f(x^*) - A^\top\lambda^*\| \le \gamma_1\sqrt{\epsilon}$ and $\|Ax^* - b\| \le \gamma_1\sqrt{\epsilon}$, which implies that

$$\begin{aligned}
Q(x^*, \lambda^*) &= \|\nabla f(x^*) - A^\top\lambda^* + \beta A^\top(Ax^* - b)\|^2 + \|Ax^* - b\|^2 \\
&\le 2\|\nabla f(x^*) - A^\top\lambda^*\|^2 + 2\beta^2\|A\|_2^2\|Ax^* - b\|^2 + \|Ax^* - b\|^2 \\
&\le 2\gamma_1^2\epsilon + (2\beta^2\|A\|_2^2 + 1)\gamma_1^2\epsilon = \epsilon.
\end{aligned}$$

On the other hand, if $Q(x^*, \lambda^*) \le \epsilon$, then we have $\|\nabla f(x^*) - A^\top\lambda^* + \beta A^\top(Ax^* - b)\|^2 \le \epsilon$ and $\|Ax^* - b\|^2 \le \epsilon$. Therefore,

$$\begin{aligned}
&\|\nabla f(x^*) - A^\top\lambda^*\|^2 \\
&\le 2\|\nabla f(x^*) - A^\top\lambda^* + \beta A^\top(Ax^* - b)\|^2 + 2\| - \beta A^\top(Ax^* - b)\|^2 \\
&\le 2\|\nabla f(x^*) - A^\top\lambda^* + \beta A^\top(Ax^* - b)\|^2 + 2\beta^2\|A\|_2^2\|Ax^* - b\|^2 \\
&\le 2(1 + \beta^2\|A\|_2^2)\,\epsilon.
\end{aligned}$$

The desired result then follows immediately. □

In the following, we introduce two variants of ADMM, to be called proximal ADMM-g and proximal ADMM-m, that solve (1.1) under some additional assumptions on A_N. In particular, proximal ADMM-g assumes $A_N = I$, and proximal ADMM-m assumes A_N to have full row rank.

3.3 Proximal gradient-based ADMM (proximal ADMM-g)

Our proximal ADMM-g solves (1.1) under the condition that $A_N = I$. In this case, the problem reduces to a so-called sharing problem in the literature which has the following form

$$\begin{aligned}&\min\ f(x_1,\ldots,x_N)+\sum_{i=1}^{N-1} r_i(x_i)\\&\text{s.t.}\ \sum_{i=1}^{N-1} A_i x_i + x_N = b,\ x_i\in\mathcal{X}_i,\ i=1,\ldots,N-1.\end{aligned}$$

For applications of the sharing problem, see [12,33,39,40]. Our proximal ADMM-g for solving (1.1) with $A_N = I$ is described in Algorithm 2. It can be seen from Algorithm 2 that proximal ADMM-g is based on the framework of augmented Lagrangian method, and can be viewed as a variant of the ADMM. The augmented Lagrangian function of (1.1) is defined as

$$\begin{aligned}\mathcal{L}_\beta(x_1,\ldots,x_N,\lambda) &:= f(x_1,\ldots,x_N)\\&\quad+\sum_{i=1}^{N-1} r_i(x_i) - \left\langle \lambda, \sum_{i=1}^{N} A_i x_i - b\right\rangle + \frac{\beta}{2}\left\|\sum_{i=1}^{N} A_i x_i - b\right\|_2^2,\end{aligned}$$

where λ is the Lagrange multiplier associated with the affine constraint, and $\beta > 0$ is a penalty parameter. In each iteration, proximal ADMM-g minimizes the augmented Lagrangian function plus a proximal term for block variables $x_1,\ldots,x_{N-1}$, with other variables being fixed; and then a gradient descent step is conducted for x_N, and finally the Lagrange multiplier λ is updated. The interested readers are referred to [26] for gradient-based ADMM and its various stochastic variants for convex optimization.

Algorithm 2 Proximal Gradient-based ADMM (proximal ADMM-g) for solving (1.1) with $A_N = I$

Require: Given $\left(x_1^0, x_2^0, \ldots, x_N^0\right) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_{N-1} \times \mathbb{R}^{n_N}$, $\lambda^0 \in \mathbb{R}^m$
for $k = 0, 1, \ldots$ **do**
[Step 1] $x_i^{k+1} := \operatorname{argmin}_{x_i\in\mathcal{X}_i}\ \mathcal{L}_\beta(x_1^{k+1},\ldots,x_{i-1}^{k+1}, x_i, x_{i+1}^k,\ldots,x_N^k,\lambda^k) + \frac{1}{2}\left\|x_i - x_i^k\right\|_{H_i}^2$ for some positive definite matrix H_i, $i = 1,\ldots,N-1$
[Step 2] $x_N^{k+1} := x_N^k - \gamma\nabla_N \mathcal{L}_\beta(x_1^{k+1}, x_2^{k+1},\ldots,x_N^k,\lambda^k)$
[Step 3] $\lambda^{k+1} := \lambda^k - \beta\left(\sum_{i=1}^N A_i x_i^{k+1} - b\right)$
end for

Remark 3.8 Note that here we actually assumed that all subproblems in Step 1 of Algorithm 2, though possibly nonconvex, can be solved to global optimality. Many important problems arising from statistics satisfy this assumption. In fact, when the coupled objective is absent or can be linearized, after choosing some proper matrix H_i, the solution of the corresponding subproblem is given by the proximal mappings of r_i. As we mentioned earlier, many nonconvex regularization functions such as SCAD, LSP, MCP and Capped-ℓ_1 admit closed-form proximal mappings. Moreover, in Algorithm 2, we can choose

$$\beta > \max\left(\frac{18\sqrt{3}+6}{13}L,\ \max_{i=1,2,\ldots,N-1}\frac{6L^2}{\sigma_{\min}(H_i)}\right), \tag{3.8}$$

and

$$\gamma \in \left(\frac{13\beta - \sqrt{13\beta^2 - 12\beta L - 72L^2}}{6L^2 + \beta L + 13\beta^2}, \frac{13\beta + \sqrt{13\beta^2 - 12\beta L - 72L^2}}{6L^2 + \beta L + 13\beta^2}\right) \tag{3.9}$$

which guarantee the convergence rate of the algorithm as shown in Lemma 3.9 and Theorem 3.12.

Before presenting the main result on the iteration complexity of proximal ADMM-g, we need some lemmas.

Lemma 3.9 *Suppose the sequence* $\{(x_1^k, \ldots, x_N^k, \lambda^k)\}$ *is generated by Algorithm 2. The following inequality holds*

$$\|\lambda^{k+1} - \lambda^k\|^2 \le 3(\beta - 1/\gamma)^2\|x_N^k - x_N^{k+1}\|^2 + 3((\beta - 1/\gamma)^2 + L^2)\|x_N^{k-1} - x_N^k\|^2 + 3L^2\sum_{i=1}^{N-1}\|x_i^{k+1} - x_i^k\|^2. \tag{3.10}$$

Proof Note that Steps 2 and 3 of Algorithm 2 yield that

$$\lambda^{k+1} = (\beta - 1/\gamma)(x_N^k - x_N^{k+1}) + \nabla_N f(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k). \tag{3.11}$$

Combining (3.11) and (3.1) yields that

$$\begin{aligned}
&\|\lambda^{k+1} - \lambda^k\|^2 \\
&\le \|(\nabla_N f(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k) - \nabla_N f(x_1^k, \ldots, x_{N-1}^k, x_N^{k-1})) \\
&\quad + (\beta - 1/\gamma)(x_N^k - x_N^{k+1}) - (\beta - 1/\gamma)(x_N^{k-1} - x_N^k)\|^2 \\
&\le 3\|\nabla_N f(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k) - \nabla_N f(x_1^k, \ldots, x_{N-1}^k, x_N^{k-1})\|^2 \\
&\quad + 3(\beta - 1/\gamma)^2\|x_N^k - x_N^{k+1}\|^2 + 3\left[\beta - \frac{1}{\gamma}\right]^2\left\|x_N^{k-1} - x_N^k\right\|^2 \\
&\le 3\left[\beta - \frac{1}{\gamma}\right]^2\left\|x_N^k - x_N^{k+1}\right\|^2 + 3\left[\left(\beta - \frac{1}{\gamma}\right)^2 + L^2\right]\left\|x_N^{k-1} - x_N^k\right\|^2 \\
&\quad + 3L^2\sum_{i=1}^{N-1}\left\|x_i^{k+1} - x_i^k\right\|^2.
\end{aligned}$$

□

We now define the following function, which will play a crucial role in our analysis:

$$\begin{aligned}&\Psi_G\left(x_1, x_2, \ldots, x_N, \lambda, \bar{x}\right)\\&= \mathcal{L}_\beta(x_1, x_2, \ldots, x_N, \lambda) + \frac{3}{\beta}\left[\left(\beta - \frac{1}{\gamma}\right)^2 + L^2\right]\|x_N - \bar{x}\|^2. \end{aligned} \tag{3.12}$$

Lemma 3.10 *Suppose the sequence $\{(x_1^k, \ldots, x_N^k, \lambda^k)\}$ is generated by Algorithm 2, where the parameters β and γ are taken according to (3.8) and (3.9) respectively. Then $\Psi_G(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}, x_N^k)$ monotonically decreases over $k \geq 0$.*

Proof From Step 1 of Algorithm 2 it is easy to see that

$$\mathcal{L}_\beta\left(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k, \lambda^k\right) \leq \mathcal{L}_\beta\left(x_1^k, \ldots, x_N^k, \lambda^k\right) - \sum_{i=1}^{N-1} \frac{1}{2}\left\|x_i^k - x_i^{k+1}\right\|_{H_i}^2. \tag{3.13}$$

From Step 2 of Algorithm 2 we get that

$$\begin{aligned}0 &= \left(x_N^k - x_N^{k+1}\right)^\top \Big[\nabla f(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k) - \lambda^k\\&\quad + \beta\left(\textstyle\sum_{i=1}^{N-1} A_i x_i^{k+1} + x_N^k - b\right) - \tfrac{1}{\gamma}\left(x_N^k - x_N^{k+1}\right)\Big]\\&\leq f(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k) - f(x_1^{k+1}, \ldots, x_N^{k+1})\\&\quad + \tfrac{L}{2}\left\|x_N^k - x_N^{k+1}\right\|^2 - \left(x_N^k - x_N^{k+1}\right)^\top \lambda^k\\&\quad + \tfrac{\beta}{2}\left\|x_N^k - x_N^{k+1}\right\|^2 + \tfrac{\beta}{2}\left\|\sum_{i=1}^{N-1} A_i x_i^{k+1} + x_N^k - b\right\|^2\\&\quad - \tfrac{\beta}{2}\left\|\sum_{i=1}^{N-1} A_i x_i^{k+1} + x_N^{k+1} - b\right\|^2 - \tfrac{1}{\gamma}\left\|x_N^k - x_N^{k+1}\right\|^2\\&= \mathcal{L}_\beta(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k, \lambda^k) - \mathcal{L}_\beta(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^k)\\&\quad + \left(\tfrac{L+\beta}{2} - \tfrac{1}{\gamma}\right)\left\|x_N^k - x_N^{k+1}\right\|^2,\end{aligned} \tag{3.14}$$

where the inequality follows from (3.2) and (3.3). Moreover, the following equality holds trivially

$$\mathcal{L}_\beta(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}) = \mathcal{L}_\beta(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^k) + \frac{1}{\beta}\left\|\lambda^k - \lambda^{k+1}\right\|^2. \tag{3.15}$$

Combining (3.13), (3.14), (3.15) and (3.10) yields that

$$\begin{aligned}&\mathcal{L}_\beta(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}) - \mathcal{L}_\beta(x_1^k, \ldots, x_N^k, \lambda^k)\\&\quad \leq \left(\frac{L+\beta}{2} - \frac{1}{\gamma}\right)\left\|x_N^k - x_N^{k+1}\right\|^2 - \sum_{i=1}^{N-1}\frac{1}{2}\left\|x_i^k - x_i^{k+1}\right\|_{H_i}^2 + \frac{1}{\beta}\left\|\lambda^k - \lambda^{k+1}\right\|^2\end{aligned}$$

$$
\begin{aligned}
&\leq \left(\frac{L+\beta}{2} - \frac{1}{\gamma} + \frac{3}{\beta}\left[\beta - \frac{1}{\gamma}\right]^2\right) \left\| x_N^k - x_N^{k+1} \right\|^2 \\
&\quad + \frac{3}{\beta}\left[\left(\beta - \frac{1}{\gamma}\right)^2 + L^2\right] \left\| x_N^{k-1} - x_N^k \right\|^2 \\
&\quad + \sum_{i=1}^{N-1} \left(x_i^k - x_i^{k+1}\right)^\top \left(\frac{3L^2}{\beta} I - \frac{1}{2} H_i\right) \left(x_i^k - x_i^{k+1}\right),
\end{aligned}
$$

which further implies that

$$
\begin{aligned}
&\Psi_G(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}, x_N^k) - \Psi_G(x_1^k, \ldots, x_N^k, \lambda^k, x_N^{k-1}) \\
&\quad \leq \left(\frac{L+\beta}{2} - \frac{1}{\gamma} + \frac{6}{\beta}\left[\beta - \frac{1}{\gamma}\right]^2 + \frac{3L^2}{\beta}\right) \left\| x_N^k - x_N^{k+1} \right\|^2 \\
&\quad\quad - \sum_{i=1}^{N-1} \left\| x_i^k - x_i^{k+1} \right\|^2_{\frac{1}{2} H_i - \frac{3L^2}{\beta} I}.
\end{aligned} \tag{3.16}
$$

It is easy to verify that when $\beta > \frac{18\sqrt{3}+6}{13} L$, then γ defined as in (3.9) ensures that $\gamma > 0$ and

$$
\frac{L+\beta}{2} - \frac{1}{\gamma} + \frac{6}{\beta}\left[\beta - \frac{1}{\gamma}\right]^2 + \frac{3L^2}{\beta} < 0. \tag{3.17}
$$

Therefore, choosing $\beta > \max\left(\frac{18\sqrt{3}+6}{13} L, \max_{i=1,2,\ldots,N-1} \frac{6L^2}{\sigma_{\min}(H_i)}\right)$ and γ as in (3.9) guarantees that $\Psi_G(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}, x_N^k)$ monotonically decreases over $k \geq 0$. In fact, (3.17) can be verified as follows. By denoting $z = \beta - \frac{1}{\gamma}$, (3.17) is equivalent to

$$
12z^2 + 2\beta z + \left(6L^2 + \beta L - \beta^2\right) < 0,
$$

which holds when $\beta > \frac{18\sqrt{3}+6}{13} L$ and $\frac{-\beta - \sqrt{13\beta^2 - 12\beta L - 72L^2}}{12} < z < \frac{-\beta + \sqrt{13\beta^2 - 12\beta L - 72L^2}}{12}$, i.e.,

$$
\frac{-13\beta - \sqrt{13\beta^2 - 12\beta L - 72L^2}}{12} < -\frac{1}{\gamma} < \frac{-13\beta + \sqrt{13\beta^2 - 12\beta L - 72L^2}}{12},
$$

which holds when γ is chosen as in (3.9). □

Lemma 3.11 *Suppose the sequence $\{(x_1^k, \ldots, x_N^k, \lambda^k)\}$ is generated by Algorithm 2. Under the same conditions as in Lemma 3.10, for any $k \geq 0$, we have*

$$
\Psi_G\left(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}, x_N^k\right) \geq \sum_{i=1}^{N-1} r_i^* + f^*,
$$

where r_i^* and f^* are defined in Assumption 3.2.

Proof Note that from (3.11), we have

$$
\begin{aligned}
&\mathcal{L}_\beta(x_1^{k+1},\ldots,x_N^{k+1},\lambda^{k+1})\\
&= \sum_{i=1}^{N-1} r_i(x_i^{k+1}) + f(x_1^{k+1},\ldots,x_N^{k+1})\\
&\quad -\left(\sum_{i=1}^{N-1} A_i x_i^{k+1} + x_N^{k+1} - b\right)^\top \nabla_N f(x_1^{k+1},\ldots,x_N^{k+1})\\
&\quad +\frac{\beta}{2}\left\|\sum_{i=1}^{N-1} A_i x_i^{k+1} + x_N^{k+1} - b\right\|^2\\
&\quad -\left(\sum_{i=1}^{N-1} A_i x_i^{k+1} + x_N^{k+1} - b\right)^\top \times \left[\left(\beta - \frac{1}{\gamma}\right)\left(x_N^k - x_N^{k+1}\right)\right.\\
&\quad \left.+\left(\nabla_N f(x_1^{k+1},\ldots,x_{N-1}^{k+1},x_N^k) - \nabla_N f(x_1^{k+1},\ldots,x_N^{k+1})\right)\right]\\
&\geq \sum_{i=1}^{N-1} r_i(x_i^{k+1}) + f(x_1^{k+1},\ldots,x_{N-1}^{k+1}, b - \sum_{i=1}^{N-1} A_i x_i^{k+1})\\
&\quad +\left(\frac{\beta}{2} - \frac{\beta}{6} - \frac{L}{2}\right)\left\|\sum_{i=1}^{N-1} A_i x_i^{k+1} + x_N^{k+1} - b\right\|^2\\
&\quad -\frac{3}{\beta}\left[\left(\beta - \frac{1}{\gamma}\right)^2 + L^2\right]\left\|x_N^k - x_N^{k+1}\right\|^2\\
&\geq \sum_{i=1}^{N-1} r_i^* + f^* - \frac{3}{\beta}\left[\left(\beta - \frac{1}{\gamma}\right)^2 + L^2\right]\left\|x_N^k - x_N^{k+1}\right\|^2,
\end{aligned}
$$

where the first inequality follows from (3.2), and the second inequality is due to $\beta \geq 3L/2$. The desired result follows from the definition of Ψ_G in (3.12). □

Now we are ready to give the iteration complexity of Algorithm 2 for finding an ϵ-stationary solution of (1.1).

Theorem 3.12 *Suppose the sequence $\{(x_1^k,\ldots,x_N^k,\lambda^k)\}$ is generated by Algorithm 2. Furthermore, suppose that β satisfies (3.8) and γ satisfies (3.9). Denote*

$$
\kappa_1 := \tfrac{3}{\beta^2}\left[\left(\beta - \tfrac{1}{\gamma}\right)^2 + L^2\right], \quad \kappa_2 := \left(|\beta - \tfrac{1}{\gamma}| + L\right)^2, \quad \kappa_3 := \max_{1\leq i\leq N-1} (\operatorname{diam}(\mathcal{X}_i))^2,
$$
$$
\kappa_4 := \left(L + \beta\sqrt{N}\max_{1\leq i\leq N}\left[\|A_i\|_2^2\right] + \max_{1\leq i\leq N}\|H_i\|_2\right)^2
$$

and

$$\tau := \min\left\{-\left(\frac{L+\beta}{2}-\frac{1}{\gamma}+\frac{6}{\beta}\left[\beta-\frac{1}{\gamma}\right]^2+\frac{3L^2}{\beta}\right),\right.$$
$$\left.\min_{i=1,\dots,N-1}\left\{-\left(\frac{3L^2}{\beta}-\frac{\sigma_{\min}(H_i)}{2}\right)\right\}\right\} > 0. \tag{3.18}$$

Then to get an ϵ-stationary solution, the number of iterations that the algorithm runs can be upper bounded by:

$$K := \begin{cases} \left\lceil \frac{2\max\{\kappa_1,\kappa_2,\kappa_4\cdot\kappa_3\}}{\tau\,\epsilon^2}\left(\Psi_G(x_1^1,\dots,x_N^1,\lambda^1,x_N^0)-\sum_{i=1}^{N-1} r_i^* - f^*\right)\right\rceil, & \textit{for } \textbf{Setting 1} \\ \left\lceil \frac{2\max\{\kappa_1,\kappa_2,\kappa_4\}}{\tau\,\epsilon^2}\left(\Psi_G(x_1^1,\dots,x_N^1,\lambda^1,x_N^0)-\sum_{i=1}^{N-1} r_i^* - f^*\right)\right\rceil, & \textit{for } \textbf{Setting 2} \end{cases} \tag{3.19}$$

and we can further identify one iteration $\hat{k} \in \operatorname*{argmin}_{2\le k\le K+1} \sum_{i=1}^N \left(\|x_i^k - x_i^{k+1}\|^2 + \|x_i^{k-1} - x_i^k\|^2\right)$ *such that* $(x_1^{\hat{k}},\dots,x_N^{\hat{k}})$ *is an ϵ-stationary solution for optimization problem* (1.1) *with Lagrange multiplier $\lambda^{\hat{k}}$ and $A_N = I$, for Settings 1 and 2 respectively.*

Proof For ease of presentation, denote

$$\theta_k := \sum_{i=1}^N (\|x_i^k - x_i^{k+1}\|^2 + \|x_i^{k-1} - x_i^k\|^2). \tag{3.20}$$

By summing (3.16) over $k = 1,\dots,K$, we obtain that

$$\Psi_G(x_1^{K+1},\dots,x_N^{K+1},\lambda^{K+1},x_N^K) - \Psi_G(x_1^1,\dots,x_N^1,\lambda^1,x_N^0)$$
$$\le -\tau\sum_{k=1}^K\sum_{i=1}^N \left\|x_i^k - x_i^{k+1}\right\|^2, \tag{3.21}$$

where τ is defined in (3.18). By invoking Lemmas 3.10 and 3.11, we get

$$\min_{2\le k\le K+1}\theta_k \le \frac{1}{\tau K}\left[\Psi_G(x_1^1,\dots,x_N^1,\lambda^1,x_N^0)+\Psi_G(x_1^2,\dots,x_N^2,\lambda^2,x_N^1) - 2\sum_{i=1}^N r_i^* - 2f^*\right]$$
$$\le \frac{2}{\tau K}\left[\Psi_G(x_1^1,\dots,x_N^1,\lambda^1,x_N^0) - \sum_{i=1}^N r_i^* - f^*\right].$$

We now derive upper bounds on the terms in (3.5) and (3.6) through θ_k. Note that (3.11) implies that

$$
\begin{aligned}
&\|\lambda^{k+1} - \nabla_N f(x_1^{k+1}, \ldots, x_N^{k+1})\| \\
&\quad \le |\beta - \frac{1}{\gamma}| \, \|x_N^k - x_N^{k+1}\| + \|\nabla_N f(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k) - \nabla f(x_1^{k+1}, \ldots, x_N^{k+1})\| \\
&\quad \le \left[|\beta - \frac{1}{\gamma}| + L \right] \|x_N^k - x_N^{k+1}\|,
\end{aligned}
$$

which yields

$$
\|\lambda^{k+1} - \nabla_N f(x_1^{k+1}, \ldots, x_N^{k+1})\|^2 \le \left[|\beta - \frac{1}{\gamma}| + L \right]^2 \theta_k. \tag{3.22}
$$

From Step 3 of Algorithm 2 and (3.10) it is easy to see that

$$
\begin{aligned}
&\left\| \sum_{i=1}^{N-1} A_i x_i^{k+1} + x_N^{k+1} - b \right\|^2 \\
&\quad = \tfrac{1}{\beta^2} \|\lambda^{k+1} - \lambda^k\|^2 \\
&\quad \le \tfrac{3}{\beta^2} \left[\beta - \tfrac{1}{\gamma} \right]^2 \left\| x_N^k - x_N^{k+1} \right\|^2 + \tfrac{3}{\beta^2} \left[\left(\beta - \tfrac{1}{\gamma} \right)^2 + L^2 \right] \left\| x_N^{k-1} - x_N^k \right\|^2 \\
&\qquad + \tfrac{3L^2}{\beta^2} \textstyle\sum_{i=1}^{N-1} \left\| x_i^k - x_i^{k+1} \right\|^2 \\
&\quad \le \tfrac{3}{\beta^2} \left[\left(\beta - \tfrac{1}{\gamma} \right)^2 + L^2 \right] \theta_k.
\end{aligned} \tag{3.23}
$$

We now derive upper bounds on the terms in (3.4) and (3.7) under the two settings in Table 1, respectively.

Setting 2 Because r_i is lower semi-continuous and $\mathcal{X}_i = \mathbb{R}^{n_i}$, $i = 1, \ldots, N-1$, it follows from Step 1 of Algorithm 2 that there exists a general subgradient $g_i \in \partial r_i(x_i^{k+1})$ such that

$$
\begin{aligned}
&\text{dist}\left(-\nabla_i f(x_1^{k+1}, \ldots, x_N^{k+1}) + A_i^\top \lambda^{k+1}, \partial r_i(x_i^{k+1}) \right) \\
&\quad \le \left\| g_i + \nabla_i f(x_1^{k+1}, \ldots, x_N^{k+1}) - A_i^\top \lambda^{k+1} \right\| \\
&\quad = \Bigg\| \nabla_i f(x_1^{k+1}, \ldots, x_N^{k+1}) - \nabla_i f(x_1^{k+1}, \ldots, x_i^{k+1}, x_{i+1}^k, \ldots, x_N^k) \\
&\qquad + \beta A_i^\top \left(\sum_{j=i+1}^{N} A_j (x_j^{k+1} - x_j^k) \right) - H_i (x_i^{k+1} - x_i^k) \Bigg\| \\
&\quad \le L \sqrt{\sum_{j=i+1}^{N} \|x_j^k - x_j^{k+1}\|^2} + \beta \, \|A_i\|_2 \sum_{j=i+1}^{N} \|A_j\|_2 \|x_j^{k+1} - x_j^k\| \\
&\qquad + \|H_i\|_2 \|x_i^{k+1} - x_i^k\|_2
\end{aligned}
$$

$$
\begin{aligned}
&\leq \left(L + \beta\sqrt{N} \max_{i+1\leq j\leq N} \left[\|A_j\|_2\right] \|A_i\|_2\right) \sqrt{\sum_{j=i+1}^{N} \|x_j^k - x_j^{k+1}\|^2} \\
&\quad + \|H_i\|_2 \|x_i^{k+1} - x_i^k\|_2 \\
&\leq \left(L + \beta\sqrt{N} \max_{1\leq i\leq N} \left[\|A_i\|_2^2\right] + \max_{1\leq i\leq N} \|H_i\|_2\right) \sqrt{\theta_k}.
\end{aligned} \tag{3.24}
$$

By combining (3.24), (3.22) and (3.23) we conclude that Algorithm 2 returns an ϵ-stationary solution for (1.1) according to Definition 3.6 under the conditions of Setting 2 in Table 1.

Setting 1 Under this setting, we know r_i is Lipschitz continuous and $\mathcal{X}_i \subset \mathbb{R}^{n_i}$ is convex and compact. From Assumption 3.1 and the fact that $\mathcal{X}_i$ is compact, we know $r_i(x_i) + f(x_1, \ldots, x_N)$ is locally Lipschitz continuous with respect to x_i for $i = 1, 2, \ldots, N-1$. Similar to (3.24), for any $x_i \in \mathcal{X}_i$, Step 1 of Algorithm 2 yields that

$$
\begin{aligned}
&\left(x_i - x_i^{k+1}\right)^\top \left[g_i + \nabla_i f(x_1^{k+1}, \ldots, x_N^{k+1}) - A_i^\top \lambda^{k+1}\right] \\
&\geq \left(x_i - x_i^{k+1}\right)^\top \Bigg[\nabla_i f(x_1^{k+1}, \ldots, x_N^{k+1}) - \nabla_i f(x_1^{k+1}, \ldots, x_i^{k+1}, x_{i+1}^k, \ldots, x_N^k) \\
&\quad + \beta A_i^\top \left(\sum_{j=i+1}^{N} A_j(x_j^{k+1} - x_j^k)\right) - H_i(x_i^{k+1} - x_i^k)\Bigg] \\
&\geq -L\,\mathrm{diam}(\mathcal{X}_i) \sqrt{\sum_{j=i+1}^{N} \|x_j^k - x_j^{k+1}\|^2} \\
&\quad - \beta\|A_i\|_2\,\mathrm{diam}(\mathcal{X}_i) \sum_{j=i+1}^{N} \|A_j\|_2 \|x_j^{k+1} - x_j^k\| - \mathrm{diam}(\mathcal{X}_i)\, \|H_i\|_2 \|x_i^{k+1} - x_i^k\|_2 \\
&\geq -\left(\beta\sqrt{N} \max_{1\leq i\leq N} \left[\|A_i\|_2^2\right] + L + \max_{1\leq i\leq N} \|H_i\|_2\right) \max_{1\leq i\leq N-1} \left[\mathrm{diam}(\mathcal{X}_i)\right] \sqrt{\theta_k},
\end{aligned} \tag{3.25}
$$

where $g_i \in \partial r_i(x_i^{k+1})$ is a general subgradient of r_i at x_i^{k+1}. By combining (3.25), (3.22) and (3.23) we conclude that Algorithm 2 returns an ϵ-stationary solution for (1.1) according to Definition 3.5 under the conditions of Setting 1 in Table 1. □

Remark 3.13 Note that the potential function Ψ_G defined in (3.12) is related to the augmented Lagrangian function. The augmented Lagrangian function has been used as a potential function in analyzing the convergence of nonconvex splitting and ADMM methods in [2,31–33,38]. See [32] for a more detailed discussion on this.

Remark 3.14 In Step 1 of Algorithm 2, we can also replace the function

$$
f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_N^k)
$$

by its linearization

$$
\begin{aligned}
&f(x_1^{k+1},\dots,x_{i-1}^{k+1},x_i^k,x_{i+1}^k,\dots,x_N^k)\\
&\quad+\left(x_i-x_i^k\right)^\top \nabla_i f(x_1^{k+1},\dots,x_{i-1}^{k+1},x_i^k,x_{i+1}^k,\dots,x_N^k),
\end{aligned}
$$

so that the subproblem can be solved by computing the proximal mappings of r_i, with some properly chosen matrix H_i for $i=1,\dots,N-1$, and the same iteration bound still holds.

3.4 Proximal majorization ADMM (proximal ADMM-m)

Our proximal ADMM-m solves (1.1) under the condition that A_N has full row rank. In this section, we use σ_N to denote the smallest eigenvalue of $A_N A_N^\top$. Note that $\sigma_N>0$ because A_N has full row rank. Our proximal ADMM-m can be described as follows

Algorithm 3 Proximal majorization ADMM (proximal ADMM-m) for solving (1.1) with A_N being full row rank

Require: Given $\left(x_1^0,x_2^0,\dots,x_N^0\right)\in\mathcal{X}_1\times\cdots\times\mathcal{X}_{N-1}\times\mathbb{R}^{n_N}$, $\lambda^0\in\mathbb{R}^m$
  for $k=0,1,\dots$ **do**
    [Step 1] $x_i^{k+1}:=\mathrm{argmin}_{x_i\in\mathcal{X}_i}\ \mathcal{L}_\beta(x_1^{k+1},\dots,x_{i-1}^{k+1},x_i,x_{i+1}^k,\dots,x_N^k,\lambda^k)+\frac{1}{2}\left\|x_i-x_i^k\right\|_{H_i}^2$ for some positive definite matrix H_i, $i=1,\dots,N-1$
    [Step 2] $x_N^{k+1}:=\mathrm{argmin}_{x_N}\ U(x_1^{k+1},\dots,x_{N-1}^{k+1},x_N,\lambda^k,x_N^k)$
    [Step 3] $\lambda^{k+1}:=\lambda^k-\beta\left(\sum_{i=1}^N A_i x_i^{k+1}-b\right)$
  end for

In Algorithm 3, $U(x_1,\dots,x_{N-1},x_N,\lambda,\bar{x})$ is defined as

$$
\begin{aligned}
&U(x_1,\dots,x_{N-1},x_N,\lambda,\bar{x})\\
&\quad=f(x_1,\dots,x_{N-1},\bar{x})+(x_N-\bar{x})^\top\nabla_N f(x_1,\dots,x_{N-1},\bar{x})\\
&\qquad+\frac{L}{2}\|x_N-\bar{x}\|^2-\left\langle\lambda,\sum_{i=1}^N A_i x_i-b\right\rangle+\frac{\beta}{2}\left\|\sum_{i=1}^N A_i x_i-b\right\|^2.
\end{aligned}
$$

Moreover, β can be chosen as

$$
\beta>\max\left\{\frac{18L}{\sigma_N},\ \max_{1\le i\le N-1}\left\{\frac{6L^2}{\sigma_N\sigma_{\min}(H_i)}\right\}\right\}. \tag{3.26}
$$

to guarantee the convergence rate of the algorithm shown in Lemma 3.16 and Theorem 3.18.

It is worth noting that the proximal ADMM-m and proximal ADMM-g differ only in Step 2: Step 2 of proximal ADMM-g takes a gradient step of the augmented Lagrangian function with respect to x_N, while Step 2 of proximal ADMM-m requires to minimize a quadratic function of x_N.

We provide some lemmas that are useful in analyzing the iteration complexity of proximal ADMM-m for solving (1.1).

Lemma 3.15 *Suppose the sequence $\{(x_1^k, \ldots, x_N^k, \lambda^k)\}$ is generated by Algorithm 3. The following inequality holds*

$$\left\|\lambda^{k+1}-\lambda^k\right\|^2 \le \frac{3L^2}{\sigma_N}\left\|x_N^k - x_N^{k+1}\right\|^2 + \frac{6L^2}{\sigma_N}\left\|x_N^{k-1}-x_N^k\right\|^2 + \frac{3L^2}{\sigma_N}\sum_{i=1}^{N-1}\left\|x_i^k - x_i^{k+1}\right\|^2. \tag{3.27}$$

Proof From the optimality conditions of Step 2 of Algorithm 3, we have

$$\begin{aligned}
0 &= \nabla_N f(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k) - A_N^\top \lambda^k + \beta A_N^\top \left(\sum_{i=1}^N A_i x_i^{k+1} - b\right) \\
&\quad - L\left(x_N^k - x_N^{k+1}\right) \\
&= \nabla_N f(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k) - A_N^\top \lambda^{k+1} - L\left(x_N^k - x_N^{k+1}\right),
\end{aligned}$$

where the second equality is due to Step 3 of Algorithm 3. Therefore, we have

$$\begin{aligned}
&\|\lambda^{k+1}-\lambda^k\|^2 \\
&\le \sigma_N^{-1}\|A_N^\top\lambda^{k+1} - A_N^\top\lambda^k\|^2 \\
&\le \sigma_N^{-1}\|(\nabla_N f(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k) \\
&\quad -\nabla_N f(x_1^k, \ldots, x_{N-1}^k, x_N^{k-1})) - L(x_N^k - x_N^{k+1}) + L(x_N^{k-1} - x_N^k)\|^2 \\
&\le \frac{3}{\sigma_N}\|\nabla_N f(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k) - \nabla_N f(x_1^k, \ldots, x_{N-1}^k, x_N^{k-1})\|^2 \\
&\quad + \frac{3L^2}{\sigma_N}(\|x_N^k - x_N^{k+1}\|^2 + \|x_N^{k-1} - x_N^k\|^2) \\
&\le \frac{3L^2}{\sigma_N}\|x_N^k - x_N^{k+1}\|^2 + \frac{6L^2}{\sigma_N}\|x_N^{k-1}-x_N^k\|^2 + \frac{3L^2}{\sigma_N}\sum_{i=1}^{N-1}\|x_i^k - x_i^{k+1}\|^2.
\end{aligned}$$

□

We define the following function that will be used in the analysis of proximal ADMM-m:

$$\Psi_L(x_1, \ldots, x_N, \lambda, \bar{x}) = \mathcal{L}_\beta(x_1, \ldots, x_N, \lambda) + \frac{6L^2}{\beta\sigma_N}\|x_N - \bar{x}\|^2.$$

Similar to the function used in proximal ADMM-g, we can prove the monotonicity and boundedness of function Ψ_L.

Lemma 3.16 *Suppose the sequence $\{(x_1^k, \dots, x_N^k, \lambda^k)\}$ is generated by Algorithm 3, whereβ is chosen according to (3.26). Then $\Psi_L(x^{k+1}, \dots, x_N^{k+1}, \lambda^{k+1}, x_N^k)$ monotonically decreases over $k > 0$.*

Proof By Step 1 of Algorithm 3 one observes that

$$\mathcal{L}_\beta\left(x_1^{k+1}, \dots, x_{N-1}^{k+1}, x_N^k, \lambda^k\right) \le \mathcal{L}_\beta\left(x_1^k, \dots, x_N^k, \lambda^k\right) - \sum_{i=1}^{N-1} \frac{1}{2}\left\|x_i^k - x_i^{k+1}\right\|_{H_i}^2, \tag{3.28}$$

while by Step 2 of Algorithm 3 we have

$$\begin{aligned} 0 &= \left(x_N^k - x_N^{k+1}\right)^\top \Big[\nabla_N f(x_1^{k+1}, \dots, x_{N-1}^{k+1}, x_N^k) - A_N{}^\top \lambda^k \\ &\quad + \beta A_N{}^\top \left(\textstyle\sum_{i=1}^N A_i x_i^{k+1} - b\right) - L\left(x_N^k - x_N^{k+1}\right)\Big] \\ &\le f(x_1^{k+1}, \dots, x_{N-1}^{k+1}, x_N^k) - f(x_1^{k+1}, \dots, x_N^{k+1}) - \tfrac{L}{2}\left\|x_N^k - x_N^{k+1}\right\|^2 \\ &\quad - \left(\textstyle\sum_{i=1}^{N-1} A_i x_i^{k+1} + A_N x_N^k - b\right)^\top \lambda^k + \left(\textstyle\sum_{i=1}^N A_i x_i^{k+1} - b\right)^\top \lambda^k \\ &\quad + \tfrac{\beta}{2}\left\|\textstyle\sum_{i=1}^{N-1} A_i x_i^{k+1} + A_N x_N^k - b\right\|^2 - \tfrac{\beta}{2}\left\|\textstyle\sum_{i=1}^N A_i x_i^{k+1} - b\right\|^2 \\ &\quad - \tfrac{\beta}{2}\left\|A_N x_N^k - A_N x_N^{k+1}\right\|^2 \\ &\le \mathcal{L}_\beta(x_1^{k+1}, \dots, x_{N-1}^{k+1}, x_N^k, \lambda^k) - \mathcal{L}_\beta(x_1^{k+1}, \dots, x_N^{k+1}, \lambda^k) - \tfrac{L}{2}\left\|x_N^k - x_N^{k+1}\right\|^2, \end{aligned} \tag{3.29}$$

where the first inequality is due to (3.2) and (3.3). Moreover, from (3.27) we have

$$\begin{aligned} &\mathcal{L}_\beta(x_1^{k+1}, \dots, x_N^{k+1}, \lambda^{k+1}) - \mathcal{L}_\beta(x_1^{k+1}, \dots, x_N^{k+1}, \lambda^k) \\ &\quad = \frac{1}{\beta}\|\lambda^k - \lambda^{k+1}\|^2 \\ &\quad \le \frac{3L^2}{\beta\sigma_N}\|x_N^k - x_N^{k+1}\|^2 + \frac{6L^2}{\beta\sigma_N}\|x_N^{k-1} - x_N^k\|^2 + \frac{3L^2}{\beta\sigma_N}\sum_{i=1}^{N-1}\|x_i^k - x_i^{k+1}\|^2. \end{aligned} \tag{3.30}$$

Combining (3.28), (3.29) and (3.30) yields that

$$\begin{aligned} &\mathcal{L}_\beta(x_1^{k+1}, \dots, x_N^{k+1}, \lambda^{k+1}) - \mathcal{L}_\beta(x_1^k, \dots, x_N^k, \lambda^k) \\ &\quad \le \left(\frac{3L^2}{\beta\sigma_N} - \frac{L}{2}\right)\|x_N^k - x_N^{k+1}\|^2 + \sum_{i=1}^{N-1}\|x_i^k - x_i^{k+1}\|^2_{\frac{3L^2}{\beta\sigma_N}I - \frac{1}{2}H_i} \\ &\qquad + \frac{6L^2}{\beta\sigma_N}\|x_N^{k-1} - x_N^k\|^2, \end{aligned}$$

which further implies that

$$\Psi_L(x_1^{k+1}, \dots, x_N^{k+1}, \lambda^{k+1}, x_N^k) - \Psi_L(x_1^k, \dots, x_N^k, \lambda^k, x_N^{k-1})$$

$$\leq \left(\frac{9L^2}{\beta\sigma_N} - \frac{L}{2}\right)\left\|x_N^k - x_N^{k+1}\right\|^2 + \sum_{i=1}^{N-1}\left(\frac{3L^2}{\beta\sigma_N} - \frac{\sigma_{\min}(H_i)}{2}\right)\left\|x_i^k - x_i^{k+1}\right\|^2 < 0, \tag{3.31}$$

where the second inequality is due to (3.26). This completes the proof. □

The following lemma shows that the function Ψ_L is lower bounded.

Lemma 3.17 *Suppose the sequence $\{(x_1^k, \ldots, x_N^k, \lambda^k)\}$ is generated by Algorithm 3. Under the same conditions as in Lemma 3.16, the sequence $\{\Psi_L(x^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}, x_N^k)\}$ is bounded from below.*

Proof From Step 3 of Algorithm 3 we have

$$\begin{aligned}
&\Psi_L(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}, x_N^k)\\
&\geq \mathcal{L}_\beta(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1})\\
&= \textstyle\sum_{i=1}^{N-1} r_i(x_i^{k+1}) + f(x_1^{k+1}, \ldots, x_N^{k+1})\\
&\quad - \left(\textstyle\sum_{i=1}^{N} A_i x_i^{k+1} - b\right)^\top \lambda^{k+1} + \frac{\beta}{2}\left\|\textstyle\sum_{i=1}^{N} A_i x_i^{k+1} - b\right\|^2\\
&= \textstyle\sum_{i=1}^{N-1} r_i(x_i^{k+1}) + f(x_1^{k+1}, \ldots, x_N^{k+1}) - \frac{1}{\beta}(\lambda^k - \lambda^{k+1})^\top\lambda^{k+1} + \frac{1}{2\beta}\|\lambda^k - \lambda^{k+1}\|^2\\
&= \textstyle\sum_{i=1}^{N-1} r_i(x_i^{k+1}) + f(x_1^{k+1}, \ldots, x_N^{k+1}) - \frac{1}{2\beta}\|\lambda^k\|^2 + \frac{1}{2\beta}\|\lambda^{k+1}\|^2 + \frac{1}{\beta}\|\lambda^k - \lambda^{k+1}\|^2\\
&\geq \textstyle\sum_{i=1}^{N-1} r_i^* + f^* - \frac{1}{2\beta}\|\lambda^k\|^2 + \frac{1}{2\beta}\|\lambda^{k+1}\|^2,
\end{aligned} \tag{3.32}$$

where the third equality follows from (3.3). Summing this inequality over $k = 0, 1, \ldots, K-1$ for any integer $K \geq 1$ yields that

$$\frac{1}{K}\sum_{k=0}^{K-1}\Psi_L\left(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}, x_N^k\right) \geq \sum_{i=1}^{N-1} r_i^* + f^* - \frac{1}{2\beta}\left\|\lambda^0\right\|^2.$$

Lemma 3.16 stipulates that $\{\Psi_L(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}, x_N^k)\}$ is a monotonically decreasing sequence; the above inequality thus further implies that the entire sequence is bounded from below. □

We are now ready to give the iteration complexity of proximal ADMM-m, whose proof is similar to that of Theorem 3.12.

Theorem 3.18 *Suppose the sequence $\{(x_1^k, \ldots, x_N^k, \lambda^k)\}$ is generated by proximal ADMM-m (Algorithm 3), and β satisfies (3.26). Denote*

$$\kappa_1 := \frac{6L^2}{\beta^2\sigma_N}, \quad \kappa_2 := 4L^2, \quad \kappa_3 := \max_{1\leq i\leq N-1}\left(\operatorname{diam}(\mathcal{X}_i)\right)^2,$$
$$\kappa_4 := \left(L + \beta\sqrt{N}\max_{1\leq i\leq N}\left[\|A_i\|_2^2\right] + \max_{1\leq i\leq N}\|H_i\|_2\right)^2,$$

and

$$\tau := \min\left\{-\left(\frac{9L^2}{\beta\sigma_N}-\frac{L}{2}\right), \min_{i=1,\ldots,N-1}\left\{-\left(\frac{3L^2}{\beta\sigma_N}-\frac{\sigma_{\min}(H_i)}{2}\right)\right\}\right\} > 0. \quad (3.33)$$

Then to get an ϵ-stationary solution, the number of iterations that the algorithm runs can be upper bounded by:

$$K := \begin{cases} \left\lceil \frac{2\max\{\kappa_1,\kappa_2,\kappa_4\cdot\kappa_3\}}{\tau\epsilon^2}(\Psi_L(x_1^1,\ldots,x_N^1,\lambda^1,x_N^0)-\sum_{i=1}^{N-1}r_i^*-f^*)\right\rceil, & \textit{for } \textbf{Setting 1} \\ \left\lceil \frac{2\max\{\kappa_1,\kappa_2,\kappa_4\}}{\tau\epsilon^2}(\Psi_L(x_1^1,\ldots,x_N^1,\lambda^1,x_N^0)-\sum_{i=1}^{N-1}r_i^*-f^*)\right\rceil, & \textit{for } \textbf{Setting 2} \end{cases} \quad (3.34)$$

and we can further identify one iteration $\hat{k} \in \underset{2\le k\le K+1}{\operatorname{argmin}} \sum_{i=1}^N \left(\|x_i^k - x_i^{k+1}\|^2 + \|x_i^{k-1}-x_i^k\|^2\right)$, such that $(x_1^{\hat{k}},\ldots,x_N^{\hat{k}})$ is an ϵ-stationary solution for (1.1) with Lagrange multiplier $\lambda^{\hat{k}}$ and A_N being full row rank, for Settings 1 and 2 respectively.

Proof By summing (3.31) over $k=1,\ldots,K$, we obtain that

$$\begin{aligned} &\Psi_L(x_1^{K+1},\ldots,x_N^{K+1},\lambda^{K+1},x_N^K)-\Psi_L(x_1^1,\ldots,x_N^1,\lambda^1,x_N^0) \\ &\quad\le -\tau\sum_{k=1}^K\sum_{i=1}^N\left\|x_i^k-x_i^{k+1}\right\|^2, \end{aligned} \quad (3.35)$$

where τ is defined in (3.33). From Lemma 3.17 we know that there exists a constant Ψ_L^* such that $\Psi(x_1^{k+1},\ldots,x_N^{k+1},\lambda^{k+1},x_N^k)\ge\Psi_L^*$ holds for any $k\ge1$. Therefore,

$$\min_{2\le k\le K+1}\theta_k \le \frac{2}{\tau K}\left[\Psi_L(x_1^1,\ldots,x_N^1,\lambda^1,x_N^0)-\Psi_L^*\right], \quad (3.36)$$

where θ_k is defined in (3.20), i.e., for K defined as in (3.34), $\theta_{\hat{k}} = O(\epsilon^2)$.

We now give upper bounds to the terms in (3.5) and (3.6) through θ_k. Note that Step 2 of Algorithm 3 implies that

$$\begin{aligned} &\|A_N^\top\lambda^{k+1}-\nabla_N f(x_1^{k+1},\ldots,x_N^{k+1})\| \\ &\quad\le L\,\|x_N^k-x_N^{k+1}\|+\|\nabla_N f(x_1^{k+1},\ldots,x_{N-1}^{k+1},x_N^k)-\nabla_N f(x_1^{k+1},\ldots,x_N^{k+1})\| \\ &\quad\le 2L\,\|x_N^k-x_N^{k+1}\|, \end{aligned}$$

which implies that

$$\|A_N^\top\lambda^{k+1}-\nabla_N f(x_1^{k+1},\ldots,x_N^{k+1})\|^2\le 4L^2\theta_k. \quad (3.37)$$

By Step 3 of Algorithm 3 and (3.27) we have

$$
\begin{aligned}
&\left\| \sum_{i=1}^{N} A_i x_i^{k+1} - b \right\|^2 \\
&= \frac{1}{\beta^2} \|\lambda^{k+1} - \lambda^k\|^2 \\
&\leq \frac{3L^2}{\beta^2 \sigma_N} \left\| x_N^k - x_N^{k+1} \right\|^2 + \frac{6L^2}{\beta^2 \sigma_N} \left\| x_N^{k-1} - x_N^k \right\|^2 + \frac{3L^2}{\beta^2 \sigma_N} \sum_{i=1}^{N-1} \left\| x_i^k - x_i^{k+1} \right\|^2 \\
&\leq \frac{6L^2}{\beta^2 \sigma_N} \theta_k.
\end{aligned}
\tag{3.38}
$$

The remaining proof is to give upper bounds to the terms in (3.4) and (3.7). Since the proof steps are almost the same as Theorem 3.12, we shall only provide the key inequalities below.

Setting 2 Under conditions in Setting 2 in Table 1, the inequality (3.24) becomes

$$
\begin{aligned}
&\operatorname{dist}\left(-\nabla_i f(x_1^{k+1}, \ldots, x_N^{k+1}) + A_i^\top \lambda^{k+1}, \partial r_i(x_i^{k+1})\right) \\
&\leq \left(L + \beta\sqrt{N} \max_{1 \leq i \leq N} \left[\|A_i\|_2^2\right] + \max_{1 \leq i \leq N} \|H_i\|_2\right) \sqrt{\theta_k}.
\end{aligned}
\tag{3.39}
$$

By combining (3.39), (3.37) and (3.38) we conclude that Algorithm 3 returns an ϵ-stationary solution for (1.1) according to Definition 3.6 under the conditions of Setting 2 in Table 1.

Setting 1 Under conditions in Setting 1 in Table 1, the inequality (3.25) becomes

$$
\begin{aligned}
&\left(x_i - x_i^{k+1}\right)^\top \left[g_i + \nabla_i f(x_1^{k+1}, \ldots, x_N^{k+1}) - A_i^\top \lambda^{k+1}\right] \\
&\geq -\left(\beta\sqrt{N} \max_{1 \leq i \leq N} \left[\|A_i\|_2^2\right] + L + \max_{1 \leq i \leq N} \|H_i\|_2\right) \max_{1 \leq i \leq N-1} [\operatorname{diam}(\mathcal{X}_i)] \sqrt{\theta_k}.
\end{aligned}
\tag{3.40}
$$

By combining (3.40), (3.37) and (3.38) we conclude that Algorithm 3 returns an ϵ-stationary solution for (1.1) according to Definition 3.5 under the conditions of Setting 1 in Table 1. □

Remark 3.19 In Step 1 of Algorithm 3, we can replace the function $f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_N^k)$ by its linearization

$$
\begin{aligned}
&f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i^k, x_{i+1}^k, \ldots, x_N^k) \\
&\quad + \left(x_i - x_i^k\right)^\top \nabla_i f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i^k, x_{i+1}^k, \ldots, x_N^k).
\end{aligned}
$$

Under the same conditions as in Remark 3.14, the same iteration bound follows by slightly modifying the analysis above.

4 Extensions

4.1 Relaxing the assumption on the last block variable x_N

It is noted that in (1.1), we have some restrictions on the last block variable x_N, i.e., $r_N \equiv 0$ and $A_N = I$ or is full row rank. In this subsection, we show how to remove these restrictions and consider the more general problem

$$\begin{aligned} &\min\ f(x_1, x_2, \ldots, x_N) + \sum_{i=1}^{N} r_i(x_i) \\ &\text{s.t.}\ \ \textstyle\sum_{i=1}^{N} A_i x_i = b, \end{aligned} \tag{4.1}$$

where $x_i \in \mathbb{R}^{n_i}$ and $A_i \in \mathbb{R}^{m \times n_i}$, $i = 1, \ldots, N$.

Before proceeding, we make the following assumption on (4.1).

Assumption 4.1 Denote $n = n_1 + \cdots + n_N$. For any compact set $S \subseteq \mathbb{R}^n$, and any sequence $\lambda^j \in \mathbb{R}^m$ with $\|\lambda^j\| \to \infty$, $j = 1, 2, \ldots$, the following limit

$$\lim_{j\to\infty} \operatorname{dist}(-\nabla f(x_1, \ldots, x_N) + A^\top \lambda^j, \sum_{i=1}^{N} \partial r_i(x_i)) \to \infty$$

holds uniformly for all $(x_1, \ldots, x_N) \in S$, where $A = [A_1, \ldots, A_N]$.

Remark that the above implies A to have full row-rank. Furthermore, if f is continuously differentiable and $\partial r_i(S) := \bigcup_{x\in S} \partial r_i(x)$ is a compact set for any compact set S, and A has full row rank, then Assumption 4.1 trivially holds. On the other hand, for popular non-convex regularization functions, such as SCAD, MCP and Capped ℓ_1-norm, it can be shown that the corresponding set $\partial r_i(S)$ is indeed compact set for any compact set S, and so Assumption 4.1 holds in all these cases.

We introduce the following problem that is closely related to (4.1):

$$\begin{aligned} &\min\ f(x_1, x_2, \ldots, x_N) + \sum_{i=1}^{N} r_i(x_i) + \tfrac{\mu(\epsilon)}{2}\|y\|^2 \\ &\text{s.t.}\ \ \textstyle\sum_{i=1}^{N} A_i x_i + y = b, \end{aligned} \tag{4.2}$$

where $\epsilon > 0$ is the target tolerance, and $\mu(\epsilon)$ is a function of ϵ which will be specified later. Now, proximal ADMM-m is ready to be used for solving (4.2) because $A_{N+1} = I$ and y is unconstrained. We have the following iteration complexity result for proximal ADMM-m to obtain an ϵ-stationary solution of (4.1); proximal ADMM-g can be analyzed similarly.

Theorem 4.2 *Consider problem (4.1) under Setting 2 in Table 1. Suppose that Assumption 4.1 holds, and the objective in (4.1), i.e., $f + \sum_{i=1}^N r_i$, has a bounded level set. Furthermore, suppose that f has a Lipschitz continuous gradient with Lipschitz constant L, and A is of full row rank. Now let the sequence $\{(x_1^k, \ldots, x_N^k, y^k, \lambda^k)\}$ be generated by proximal ADMM-m for solving (4.2) with initial iterates $y^0 = \lambda^0 = 0$, and $(x_1^0, \ldots, x_N^0)$ such that $\sum_{i=1}^N A_i x_i^0 = b$. Assume that the target tolerance ϵ satisfies*

$$0 < \epsilon < \min\left\{\frac{1}{L}, \frac{1}{6\bar{\tau}}\right\}, \ \textit{where } \bar{\tau} = \frac{1}{2}\min_{i=1,\ldots,N}\{\sigma_{\min}(H_i)\}. \tag{4.3}$$

Then in no more than $O(1/\epsilon^4)$ iterations we will reach an iterate $(x_1^{\hat{K}+1}, \ldots, x_N^{\hat{K}+1}, y^{\hat{K}+1})$ that is an ϵ-stationary solution for (4.2) with Lagrange multiplier $\lambda^{\hat{K}+1}$. Moreover, $(x_1^{\hat{K}+1}, \ldots, x_N^{\hat{K}+1})$ is an ϵ-stationary solution for (4.1) with Lagrange multiplier $\lambda^{\hat{K}+1}$.

Proof Denote the penalty parameter as $\beta(\epsilon)$. The augmented Lagrangian function of (4.2) is given by

$$\begin{aligned}
&\mathcal{L}_{\beta(\epsilon)}(x_1, \ldots, x_N, y, \lambda) \\
&\quad := f(x_1, \ldots, x_N) + \textstyle\sum_{i=1}^N r_i(x_i) + \frac{\mu(\epsilon)}{2}\|y\|^2 - \langle \lambda, \sum\limits_{i=1}^N A_i x_i + y - b\rangle \\
&\qquad + \tfrac{\beta(\epsilon)}{2}\|\sum\limits_{i=1}^N A_i x_i + y - b\|^2.
\end{aligned}$$

Now we set

$$\mu(\epsilon) = 1/\epsilon, \ \text{and } \beta(\epsilon) = 3/\epsilon. \tag{4.4}$$

From (4.3) we have $\mu(\epsilon) > L$. This implies that the Lipschitz constant of $f(x_1, x_2, \ldots, x_N) + \frac{\mu(\epsilon)}{2}\|y\|^2$, which is the smooth part of the objective in (4.2), is equal to $\mu(\epsilon)$. Then from the optimality conditions of Step 2 of Algorithm 3, we have $\mu(\epsilon)y^{k-1} - \lambda^k - \mu(\epsilon)(y^{k-1} - y^k) = 0$, which further implies that $\mu(\epsilon)y^k = \lambda^k, \forall k \geq 1$.

Similar to Lemma 3.16, we can prove that $\mathcal{L}_{\beta(\epsilon)}(x_1^k, \ldots, x_N^k, y^k, \lambda^k)$ monotonically decreases. Specifically, since $\mu(\epsilon)y^k = \lambda^k$, combining (3.28), (3.29) and the equality in (3.30) yields,

$$\begin{aligned}
&\mathcal{L}_{\beta(\epsilon)}(x_1^{k+1}, \ldots, x_N^{k+1}, y^{k+1}, \lambda^{k+1}) - \mathcal{L}_{\beta(\epsilon)}(x_1^k, \ldots, x_N^k, y^k, \lambda^k) \\
&\quad \leq -\frac{1}{2}\sum_{i=1}^N \|x_i^k - x_i^{k+1}\|_{H_i}^2 - \left(\frac{\mu(\epsilon)}{2} - \frac{\mu(\epsilon)^2}{\beta(\epsilon)}\right)\|y^k - y^{k+1}\|^2 < 0,
\end{aligned} \tag{4.5}$$

where the last inequality is due to (4.4).

Similar to Lemma 3.17, we can prove that $\mathcal{L}_{\beta(\epsilon)}(x_1^k, \ldots, x_N^k, y^k, \lambda^k)$ is bounded from below, i.e., the exists a constant $\mathcal{L}^* = f^* + \sum_{i=1}^N r_i^*$ such that

$$\mathcal{L}_{\beta(\epsilon)}(x_1^k, \ldots, x_N^k, y^k, \lambda^k) \geq \mathcal{L}^*, \quad \text{for all } k.$$

Actually the following inequalities lead to the above fact:

$$
\begin{aligned}
&\mathcal{L}_{\beta(\epsilon)}(x_1^k, \ldots, x_N^k, y^k, \lambda^k) \\
&= f(x_1^k, \ldots, x_N^k) + \sum_{i=1}^N r_i(x_i^k) + \frac{\mu(\epsilon)}{2}\|y^k\|^2 - \left\langle \lambda^k, \sum_{i=1}^N A_i x_i^k + y^k - b \right\rangle \\
&\quad + \frac{\beta(\epsilon)}{2} \left\| \sum_{i=1}^N A_i x_i^k + y^k - b \right\|^2 \\
&= f(x_1^k, \ldots, x_N^k) + \sum_{i=1}^N r_i(x_i^k) + \frac{\mu(\epsilon)}{2}\|y^k\|^2 - \left\langle \mu(\epsilon) y^k, \sum_{i=1}^N A_i x_i^k + y^k - b \right\rangle \\
&\quad + \frac{\beta(\epsilon)}{2} \left\| \sum_{i=1}^N A_i x_i^k + y^k - b \right\|^2 \\
&\geq \mathcal{L}^* + \mu(\epsilon) \left[\frac{1}{2} \left\| \sum_{i=1}^N A_i x_i^k - b \right\|^2 + \left(\frac{\beta(\epsilon) - \mu(\epsilon)}{2\mu(\epsilon)} \right) \left\| \sum_{i=1}^N A_i x_i^k + y^k - b \right\|^2 \right] \geq \mathcal{L}^*,
\end{aligned}
\tag{4.6}
$$

where the second equality is from $\mu(\epsilon) y^k = \lambda^k$, and the last inequality is due to (4.4). Moreover, denote $\mathcal{L}^0 \equiv \mathcal{L}_{\beta(\epsilon)}(x_1^0, \ldots, x_N^0, y^0, \lambda^0)$, which is a constant independent of ϵ.

Furthermore, for any integer $K \geq 1$, summing (4.5) over $k = 0, \ldots, K$ yields

$$
\mathcal{L}_{\beta(\epsilon)}(x_1^{K+1}, \ldots, x_N^{K+1}, y^{K+1}, \lambda^{K+1}) - \mathcal{L}^0 \leq -\bar{\tau} \sum_{k=0}^K \theta_k, \tag{4.7}
$$

where $\theta_k := \sum_{i=1}^N \|x_i^k - x_i^{k+1}\|^2 + \|y^k - y^{k+1}\|^2$. Note that (4.7) and (4.6) imply that

$$
\min_{0 \leq k \leq K} \theta_k \leq \frac{1}{\bar{\tau} K} \left(\mathcal{L}^0 - \mathcal{L}^* \right). \tag{4.8}
$$

Similar to (3.24), it can be shown that for $i = 1, \ldots, N$,

$$
\begin{aligned}
&\operatorname{dist}\left(-\nabla_i f(x_1^{k+1}, \ldots, x_N^{k+1}) + A_i^\top \lambda^{k+1}, \partial r_i(x_i^{k+1}) \right) \\
&\quad \leq \left(L + \beta(\epsilon)\sqrt{N} \max_{1 \leq i \leq N} \|A_i\|_2^2 + \max_{1 \leq i \leq N} \|H_i\|_2 \right) \sqrt{\theta_k}.
\end{aligned}
\tag{4.9}
$$

Set $K = 1/\epsilon^4$ and denote $\hat{K} = \operatorname{argmin}_{0 \leq k \leq K} \theta_k$. Then we know $\theta_{\hat{K}} = O(\epsilon^4)$. As a result,

$$
\left\| \sum_{i=1}^N A_i x_i^{\hat{K}+1} + y^{\hat{K}+1} - b \right\|^2
$$

$$= \frac{1}{\beta(\epsilon)^2}\|\lambda^{\hat{K}+1} - \lambda^{\hat{K}}\|^2 = \frac{\mu(\epsilon)^2}{\beta(\epsilon)^2}\|y^{\hat{K}+1} - y^{\hat{K}}\|^2 \le \frac{1}{9}\theta_{\hat{K}} = O(\epsilon^4). \quad (4.10)$$

Note that (4.6) also implies that $f(x_1^k, \ldots, x_N^k) + \sum_{i=1}^N r_i(x_i^k)$ is upper-bounded by a constant. Thus, from the assumption that the level set of the objective is bounded, we know $(x_1^k, \ldots, x_N^k)$ is bounded. Then Assumption 4.1 implies that λ^k bounded, which results in $\|y^k\| = O(\epsilon)$. Therefore, from (4.10) we have

$$\left\|\sum_{i=1}^N A_i x_i^{\hat{K}+1} - b\right\| \le \left\|\sum_{i=1}^N A_i x_i^{\hat{K}+1} + y^{\hat{K}+1} - b\right\| + \left\|y^{\hat{K}+1}\right\| = O(\epsilon),$$

which combining with (4.9) yields that $(x_1^{\hat{K}+1}, \ldots, x_N^{\hat{K}+1})$ is an ϵ-stationary solution for (4.1) with Lagrange multiplier $\lambda^{\hat{K}+1}$, according to Definition 3.6. □

Remark 4.3 Without Assumption 4.1, we can still provide an iteration complexity of proximal ADMM-m, but the complexity bound is worse than $O(1/\epsilon^4)$. To see this, note that because $\mathcal{L}_{\beta(\epsilon)}(x_1^k, \ldots, x_N^k, y^k, \lambda^k)$ monotonically decreases, the first inequality in (4.6) implies that

$$\mu(\epsilon)\frac{1}{2}\left\|\sum_{i=1}^N A_i x_i^k - b\right\|^2 \le \mathcal{L}^0 - \mathcal{L}^*, \forall k. \quad (4.11)$$

Therefore, by setting $K = 1/\epsilon^6$, $\mu(\epsilon) = 1/\epsilon^2$ and $\beta(\epsilon) = 3/\epsilon^2$ instead of (4.4), and combining (4.9) and (4.11), we conclude that $(x_1^{\hat{K}+1}, \ldots, x_N^{\hat{K}+1})$ is an ϵ-stationary solution for (4.1) with Lagrange multiplier $\lambda^{\hat{K}+1}$, according to Definition 3.6.

4.2 Proximal BCD (block coordinate descent)

In this section, we apply a proximal block coordinate descent method to solve the following variant of (1.1) and present its iteration complexity:

$$\begin{aligned} \min\ & F(x_1, x_2, \ldots, x_N) := f(x_1, x_2, \ldots, x_N) + \sum_{i=1}^N r_i(x_i) \\ \text{s.t.}\ & x_i \in \mathcal{X}_i,\ i = 1, \ldots, N, \end{aligned} \quad (4.12)$$

where f is differentiable, r_i is nonsmooth, and $\mathcal{X}_i \subset \mathbb{R}^{n_i}$ is a closed convex set for $i = 1, 2, \ldots, N$. Note that f and r_i can be nonconvex functions. Our proximal BCD method for solving (4.12) is described in Algorithm 4.

Similar to the settings in Table 1, depending on the properties of r_i and $\mathcal{X}_i$, the ϵ-stationary solution for (4.12) is as follows.

Definition 4.4 $(x_1^*, \ldots, x_N^*, \lambda^*)$ is called an ϵ-stationary solution for (4.12), if

Algorithm 4 A proximal BCD method for solving (4.12)

Require: Given $\left(x_1^0, x_2^0, \ldots, x_N^0\right) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_N$

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

Update block x_i in a cyclic order, i.e., for $i = 1, \ldots, N$ (H_i positive definite):

$$x_i^{k+1} := \operatorname*{argmin}_{x_i \in \mathcal{X}_i} \; F(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_N^k) + \frac{1}{2} \left\| x_i - x_i^k \right\|_{H_i}^2 . \tag{4.13}$$

end for

(i) r_i is Lipschitz continuous, $\mathcal{X}_i$ is convex and compact, and for any $x_i \in \mathcal{X}_i$, $i = 1, \ldots, N$, it holds that ($g_i = \partial r_i(x_i^*)$ denotes a generalized subgradient of r_i)

$$\left(x_i - x_i^*\right)^\top \left[\nabla_i f(x_1^*, \ldots, x_N^*) + g_i\right] \geq -\epsilon;$$

(ii) or, if r_i is lower semi-continuous, $\mathcal{X}_i = \mathbb{R}^{n_i}$ for $i = 1, \ldots, N$, it holds that

$$\operatorname{dist}\left(-\nabla_i f(x_1^*, \ldots, x_N^*), \partial r_i(x_i^*)\right) \leq \epsilon.$$

We now show that the iteration complexity of Algorithm 4 can be obtained from that of proximal ADMM-g. By introducing an auxiliary variable x_{N+1} and an arbitrary vector $b \in \mathbb{R}^m$, problem (4.12) can be equivalently rewritten as

$$\begin{aligned} &\min \; f(x_1, x_2, \ldots, x_N) + \sum_{i=1}^N r_i(x_i) \\ &\text{s.t.} \;\; x_{N+1} = b, \;\; x_i \in \mathcal{X}_i, \;\; i = 1, \ldots, N. \end{aligned} \tag{4.14}$$

It is easy to see that applying proximal ADMM-g to solve (4.14) (with x_{N+1} being the last block variable) reduces exactly to Algorithm 4. Hence, we have the following iteration complexity result of Algorithm 4 for obtaining an ϵ-stationary solution of (4.12).

Theorem 4.5 *Suppose the sequence $\{(x_1^k, \ldots, x_N^k)\}$ is generated by proximal BCD (Algorithm 4). Denote*

$$\kappa_5 := (L + \max_{1 \leq i \leq N} \|H_i\|_2)^2, \;\; \kappa_6 := \max_{1 \leq i \leq N} (\operatorname{diam}(\mathcal{X}_i))^2.$$

Letting

$$K := \begin{cases} \left\lceil \frac{\kappa_5 \cdot \kappa_6}{\tau \epsilon^2} (\Psi_G(x_1^1, \ldots, x_N^1, \lambda^1, x_N^0) - \sum_{i=1}^N r_i^* - f^*) \right\rceil & \textit{for } \textbf{Setting 1} \\ \left\lceil \frac{\kappa_5}{\tau \epsilon^2} (\Psi_G(x_1^1, \ldots, x_N^1, \lambda^1, x_N^0) - \sum_{i=1}^N r_i^* - f^*) \right\rceil & \textit{for } \textbf{Setting 2} \end{cases}$$

with τ being defined in (3.18), and $\hat{K} := \min_{1 \leq k \leq K} \sum_{i=1}^N \left(\|x_i^k - x_i^{k+1}\|^2\right)$, we have that $(x_1^{\hat{K}}, \ldots, x_N^{\hat{K}})$ is an ϵ-stationary solution for problem (4.12).

Proof Note that $A_1 = \cdots = A_N = 0$ and $A_{N+1} = I$ in problem (4.14). By applying proximal ADMM-g with $\beta > \max\left\{18L,\ \max_{1\le i\le N}\left\{\frac{6L^2}{\sigma_{\min}(H_i)}\right\}\right\}$, Theorem 3.12 holds. In particular, (3.24) and (3.25) are valid in different settings with $\beta\sqrt{N}\max_{i+1\le j\le N+1}\left[\|A_j\|_2\right]\|A_i\|_2 = 0$ for $i = 1,\ldots,N$, which leads to the choices of κ_5 and κ_6 in the above. Moreover, we do not need to consider the optimality with respect to x_{N+1} and the violation of the affine constraints, thus κ_1 and κ_2 in Theorem 3.12 are excluded in the expression of K, and the conclusion follows. □

5 Numerical experiments

5.1 Robust tensor PCA model

We consider the following nonconvex and nonsmooth model of robust tensor PCA with ℓ_1 norm regularization for third-order tensor of dimension $I_1 \times I_2 \times I_3$. Given an initial estimate R of the CP-rank, we aim to solve the following problem:

$$\begin{array}{ll}\min_{A,B,C,\mathcal{Z},\mathcal{E},\mathcal{B}} & \|\mathcal{Z} - [\![A, B, C]\!]\|_F^2 + \alpha\,\|\mathcal{E}\|_1 + \alpha_{\mathcal{N}}\|\mathcal{B}\|_F^2 \\ \text{s.t.} & \mathcal{Z} + \mathcal{E} + \mathcal{B} = \mathcal{T},\end{array} \tag{5.1}$$

where $A \in \mathbb{R}^{I_1\times R}$, $B \in \mathbb{R}^{I_2\times R}$, $C \in \mathbb{R}^{I_3\times R}$. The augmented Lagrangian function of (5.1) is given by

$$\begin{aligned}&\mathcal{L}_\beta(A, B, C, \mathcal{Z}, \mathcal{E}, \mathcal{B}, \Lambda)\\ &\quad= \|\mathcal{Z} - [\![A, B, C]\!]\|_F^2 + \alpha\,\|\mathcal{E}\|_1 + \alpha_{\mathcal{N}}\|\mathcal{B}\|_F^2 - \langle \Lambda, \mathcal{Z} + \mathcal{E} + \mathcal{B} - \mathcal{T}\rangle\\ &\qquad+\frac{\beta}{2}\|\mathcal{Z} + \mathcal{E} + \mathcal{B} - \mathcal{T}\|_F^2.\end{aligned}$$

The following identities are useful for our presentation later:

$$\begin{aligned}\|\mathcal{Z} - [\![A, B, C]\!]\|_F^2 &= \|Z_{(1)} - A(C \odot B)^\top\|_F^2\\ &= \|Z_{(2)} - B(C \odot A)^\top\|_F^2\\ &= \|Z_{(3)} - C(B \odot A)^\top\|_F^2,\end{aligned}$$

where $Z_{(i)}$ stands for the mode-i unfolding of tensor $\mathcal{Z}$ and $\odot$ stands for the Khatri-Rao product of matrices.

Note that there are six block variables in (5.1), and we choose $\mathcal{B}$ as the last block variable. A typical iteration of proximal ADMM-g for solving (5.1) can be described as follows (we chose $H_i = \delta_i I$, with $\delta_i > 0$, $i = 1,\ldots,5$):

$$\begin{cases} A^{k+1} &= \left((Z)^k_{(1)}(C^k \odot B^k) + \frac{\delta_1}{2}A^k\right)\left(((C^k)^\top C^k) \circ ((B^k)^\top B^k) + \frac{\delta_1}{2}I_{R\times R}\right)^{-1} \\ B^{k+1} &= \left((Z)^k_{(2)}(C^k \odot A^{k+1}) + \frac{\delta_2}{2}B^k\right)\left(((C^k)^\top C^k) \circ ((A^{k+1})^\top A^{k+1}) + \frac{\delta_2}{2}I_{R\times R}\right)^{-1} \\ C^{k+1} &= \left((Z)^k_{(3)}(B^{k+1} \odot A^{k+1}) + \frac{\delta_3}{2}C^k\right)\left(((B^{k+1})^\top B^{k+1}) \circ ((A^{k+1})^\top A^{k+1}) + \frac{\delta_3}{2}I_{R\times R}\right)^{-1} \\ E^{k+1}_{(1)} &= \mathcal{S}\left(\frac{\beta}{\beta+\delta_4}(T_{(1)} + \frac{1}{\beta}\Lambda^k_{(1)} - B^k_{(1)} - Z^k_{(1)}) + \frac{\delta_4}{\beta+\delta_4}E^k_{(1)}, \frac{\alpha}{\beta+\delta_4}\right) \\ Z^{k+1}_{(1)} &= \frac{1}{2+2\delta_5+\beta}\left(2A^{k+1}(C^{k+1} \odot B^{k+1})^\top + 2\delta_5\,(Z_{(1)})^k + \Lambda^k_{(1)} - \beta(E^{k+1}_{(1)} + B^k_{(1)} - T_{(1)})\right) \\ B^{k+1}_{(1)} &= B^k_{(1)} - \gamma\left(2\alpha_{\mathcal{N}}B^k_{(1)} - \Lambda^k_{(1)} + \beta(E^{k+1}_{(1)} + Z^{k+1}_{(1)} + B^k_{(1)} - T_{(1)})\right) \\ \Lambda^{k+1}_{(1)} &= \Lambda^k_{(1)} - \beta\left(Z^{k+1}_{(1)} + E^{k+1}_{(1)} + B^{k+1}_{(1)} - T_{(1)}\right) \end{cases}$$

where $\circ$ is the matrix Hadamard product and $\mathcal{S}$ stands for the soft shrinkage operator. The updates in proximal ADMM-m are almost the same as proximal ADMM-g except $B_{(1)}$ is updated as

$$B^{k+1}_{(1)} = \frac{1}{L+\beta}\left((L - 2\alpha_{\mathcal{N}})B^k_{(1)} + \Lambda^k_{(1)} - \beta(E^{k+1}_{(1)} + Z^{k+1}_{(1)} - T_{(1)})\right).$$

On the other hand, note that (5.1) can be equivalently written as

$$\min_{A,B,C,\mathcal{Z},\mathcal{E}} \|\mathcal{Z} - [\![A, B, C]\!]\|_F^2 + \alpha\,\|\mathcal{E}\|_1 + \alpha_{\mathcal{N}}\|\mathcal{Z} + \mathcal{E} - \mathcal{T}\|_F^2, \tag{5.2}$$

which can be solved by the classical BCD method as well as our proximal BCD (Algorithm 4). In addition, we can apply GCG (Algorithm 1) to solve a variant of (5.1). Note that GCG requires a compact constraint set and thus it does not apply to (5.1) directly. As a result, we consider the following variant of (5.1), where the new quadratic regularization terms in the objective are added to help construct the compact constraint sets.

$$\begin{array}{ll} \min & \|\mathcal{Z} - [\![A, B, C]\!]\|_F^2 + \alpha\,\|\mathcal{E}\|_1 + \alpha_{\mathcal{N}}\|\mathcal{Z} + \mathcal{E} - \mathcal{T}\|_F^2 + \frac{\alpha_A}{2}\|A\|_F^2 + \frac{\alpha_B}{2}\|B\|_F^2 \\ & \quad + \frac{\alpha_C}{2}\|C\|_F^2 + \frac{\alpha_Z}{2}\|\mathcal{Z}\|_F^2 \\ \text{s.t.} & \|A\|_F \le \rho_1, \|B\|_F \le \rho_2, \|C\|_F \le \rho_3, \|\mathcal{Z}\|_F \le \rho_4, \|\mathcal{E}\|_1 \le \rho_5. \end{array} \tag{5.3}$$

The new parameter ρ_1 can be identified by the following observation:

$$\frac{\alpha_A}{2}\|\mathcal{A}^*\|_F^2 \le f(A^*, B^*, C^*, \mathcal{Z}^*, \mathcal{E}^*) \le f(0) = \alpha_{\mathcal{N}}\|\mathcal{T}\|_F^2,$$

which implies that $\rho_1 = \sqrt{\frac{2\alpha_{\mathcal{N}}}{\alpha_A}}\|\mathcal{T}\|_F$. Other parameters $\rho_2, \ldots, \rho_5$ can be computed in the same manner.

In the following we shall compare the numerical performance of GCG, BCD, proximal BCD, proximal ADMM-g and proximal ADMM-m for solving (5.1). We let $\alpha = 2/\max\{\sqrt{I_1}, \sqrt{I_2}, \sqrt{I_3}\}$ and $\alpha_{\mathcal{N}} = 1$ in model (5.1). We apply proximal ADMM-g and proximal ADMM-m to solve (5.1), apply BCD and proximal BCD to solve (5.2), and apply GCG to solove (5.3) with $\alpha_A = \alpha_B = \alpha_C = 10$ and $\alpha_{Z=1}$.

Table 2 Choices of parameters in the two ADMM variants

	$H_i, i = 1, \ldots, 5$	β	γ
Proximal ADMM-g	$\frac{1}{2}\beta \cdot I$	4	$\frac{1}{\beta}$
Proximal ADMM-m	$\frac{2}{5}\beta \cdot I$	5	–

In all the four algorithms we set the maximum iteration number to be 2000, and the algorithms are terminated either when the maximum iteration number is reached or when θ_k as defined in (3.20) is less than 10^{-6}. The parameters used in the two ADMM variants are specified in Table 2.

In the experiment, we randomly generate 20 instances for fixed tensor dimension and CP-rank. Suppose the low-rank part $\mathcal{Z}^0$ is of rank R_{CP}. It is generated by

$$\mathcal{Z}^0 = \sum_{r=1}^{R_{CP}} a^{1,r} \otimes a^{2,r} \otimes a^{3,r},$$

where vectors $a^{i,r}$ are generated from standard Gaussian distribution for $i = 1, 2, 3$, $r = 1, \ldots, R_{CP}$. Moreover, a sparse tensor $\mathcal{E}^0$ is generated with cardinality of $0.001 \cdot I_1 I_2 I_3$ such that each nonzero component follows from standard Gaussian distribution. Finally, we generate noise $\mathcal{B}^0 = 0.001 * \hat{\mathcal{B}}$, where $\hat{\mathcal{B}}$ is a Gaussian tensor. Then we set $\mathcal{T} = \mathcal{Z}^0 + \mathcal{E}^0 + \mathcal{B}^0$ as the observed data in (5.1). A proper initial guess R of the true rank R_{CP} is essential for the success of our algorithms. We can borrow the strategy in matrix completion [54], and start from a large R ($R \geq R_{CP}$) and decrease it aggressively once a dramatic change in the recovered tensor $\mathcal{Z}$ is observed. We report the average performance of 20 instances of the four algorithms with initial guess $R = R_{CP}$, $R = R_{CP} + 1$ and $R = R_{CP} + \lceil 0.2 * R_{CP} \rceil$ in Tables 3, 4 and 5, respectively.

In Tables 3, 4 and 5, "Err." denotes the averaged relative error $\frac{\|\mathcal{Z}^* - \mathcal{Z}^0\|_F}{\|\mathcal{Z}^0\|_F}$ of the low-rank tensor over 20 instances, where $\mathcal{Z}^*$ is the solution returned by the corresponding algorithm; "Iter." denotes the averaged number of iterations over 20 instances; "#" records the number of solutions (out of 20 instances) that have relative error less than 0.01.

Tables 3, 4 and 5 suggest that BCD mostly converges to a local solution rather than the global optimal solution, GCG easily gets stuck at a local solution in a few iterations for this particular problem, while the other three methods are much better in finding the global optimum.

It is interesting to note that the results presented in Table 5 are better than that of Tables 4 and 3 when a larger basis is allowed in tensor factorization. Moreover, in this case, the proximal BCD usually consumes less number of iterations than the two ADMM variants.

Table 3 Numerical results for tensor robust PCA with initial guess $R = R_{CP}$

R_{CP}	Proximal ADMM-g			Proximal ADMM-m			BCD			Proximal BCD			GCG		
	Iter.	Err.	#	Iter.	Err.	#	Iter.	Err.	#	Iter.	Err.	#	Iter.	Err.	#
Tensor size $10 \times 20 \times 30$															
3	295.30	0.0027	20	330.65	0.0027	20	485.60	0.7728	0	185.75	0.0027	20	32.70	1.0160	0
10	372.25	0.0027	20	391.65	0.0027	20	1262.90	0.9081	0	537.40	0.0350	17	29.40	1.0166	0
15	515.65	0.0089	19	527.55	0.0095	19	1482.00	0.9285	0	495.00	0.0168	18	33.90	1.0178	0
Tensor size $15 \times 25 \times 40$															
5	462.65	0.0229	19	494.00	0.0229	19	709.65	0.8706	0	360.50	0.0229	19	27.05	1.0164	0
10	735.50	0.0546	15	735.60	0.0480	16	1146.50	0.8994	0	371.85	0.0156	19	29.80	1.0191	0
20	919.85	0.0425	13	731.00	0.0261	16	1849.15	0.9484	0	419.30	0.0114	18	27.65	1.0189	0
Tensor size $30 \times 50 \times 70$															
8	1182.45	0.1830	9	1256.75	0.1671	9	867.45	0.9272	0	1015.05	0.1681	9	20.05	1.0163	0
20	1407.25	0.1334	7	1285.75	0.1245	7	1664.85	0.9670	0	902.05	0.0961	9	20.15	1.0164	0
40	1445.55	0.0911	7	1668.15	0.0986	6	1996.95	0.9812	0	1121.65	0.0552	11	19.20	1.0178	0

Table 4 Numerical results for tensor robust PCA with initial guess $R = R_{CP} + 1$

R_{CP}	Proximal ADMM-g			Proximal ADMM-m			BCD			Proximal BCD			GCG		
	Iter.	Err.	#	Iter.	Err.	#	Iter.	Err.	#	Iter.	Err.	#	Iter.	Err.	#
Tensor size $10 \times 20 \times 30$															
3	1740.95	0.0034	20	1742.35	0.0033	20	385.00	0.7320	0	1816.30	0.0033	20	65.45	1.0198	0
10	1490.85	0.0028	20	1498.55	0.0028	20	1025.30	0.9062	0	1030.95	0.0028	20	30.10	1.0158	0
15	1160.90	0.0029	20	1188.70	0.0117	19	1348.40	0.9229	0	954.50	0.0029	20	36.85	1.0169	0
Tensor size $15 \times 25 \times 40$															
5	1416.95	0.0020	20	1514.20	0.0020	20	623.75	0.8344	0	1553.30	0.0020	20	40.00	1.0219	0
10	1035.60	0.0110	19	1079.70	0.0018	20	1223.00	0.9208	0	733.40	0.0018	20	32.45	1.0196	0
20	902.65	0.0079	19	961.70	0.0079	19	1748.90	0.9557	0	886.00	0.0020	20	28.20	1.0192	0
Tensor size $30 \times 50 \times 70$															
8	618.10	0.0009	20	735.50	0.0009	20	965.15	0.9225	0	421.15	0.0149	19	25.05	1.0190	0
20	1276.15	0.0773	12	1169.75	0.0594	13	1708.35	0.9676	0	645.75	0.0331	16	17.70	1.0172	0
40	1163.80	0.0337	15	1336.35	0.0462	13	1996.35	0.9816	0	987.90	0.0358	14	20.15	1.0160	0

Table 5 Numerical results for tensor robust PCA with initial guess $R = R_{CP} + \lceil 0.2 * R_{CP} \rceil$

R_{CP}	Proximal ADMM-g			Proximal ADMM-m			BCD			Proximal BCD			GCG		
	Iter.	Err.	#	Iter.	Err.	#	Iter.	Err.	#	Iter.	Err.	#	Iter.	Err.	#
Tensor size $10 \times 20 \times 30$															
3	1736.90	0.0030	20	1740.95	0.0030	20	431.00	0.7357	0	1908.20	0.0031	20	60.60	1.0216	0
10	1938.90	0.0033	20	1941.30	0.0033	20	1236.55	0.9169	0	1446.00	0.0032	20	29.45	1.0184	0
15	2000.00	0.0033	20	1918.10	0.0033	20	1730.80	0.9255	0	1813.60	0.0033	20	27.80	1.0175	0
Tensor size $15 \times 25 \times 40$															
5	1581.45	0.0020	20	1515.95	0.0019	20	679.90	0.8733	0	1463.75	0.0020	20	45.80	1.0202	0
10	1672.25	0.0021	20	1669.20	0.0021	20	925.30	0.9105	0	1620.00	0.0021	20	29.10	1.0153	0
20	2000.00	0.0021	20	1854.20	0.0021	20	1621.10	0.9534	0	1823.05	0.0022	20	22.10	1.0165	0
Tensor size $30 \times 50 \times 70$															
8	870.40	0.0009	20	984.90	0.0009	20	868.05	0.9101	0	778.05	0.0009	20	9.25	1.0122	0
20	1317.80	0.0009	20	1347.95	0.0009	20	1585.85	0.9649	0	1035.50	0.0009	20	10.60	1.0122	0
40	1810.70	0.0009	20	1993.25	0.0009	20	1998.45	0.9822	0	1796.60	0.0009	20	6.35	1.0093	0

Table 6 Numerical results for sparse tensor PCA problem

Inst. #	BCD			Proximal BCD			GCG		
	Val.	$\sum_{i=1}^{d} \|x_i\|_0$	Iter.	Val.	$\sum_{i=1}^{d} \|x_i\|_0$	Iter.	Val.	$\sum_{i=1}^{d} \|x_i\|_0$	Iter.
Dimension $n = 8$									
1	4.63	11	144	5.45	11	199	6.22	16	210
2	9.01	19	113	8.68	21	224	6.36	13	303
3	5.71	13	124	7.42	16	116	6.79	17	374
4	6.09	15	2000	6.30	13	231	6.11	16	381
5	4.79	16	2000	4.13	7	238	0.00	0	15
6	7.45	16	66	6.79	16	169	7.45	16	145
7	5.83	13	105	6.57	17	116	0.00	0	17
8	6.98	19	312	6.00	14	285	0.00	0	13
9	6.83	18	2000	8.27	20	163	8.27	20	89
10	7.24	18	103	7.13	15	94	6.95	12	107
Dimension $n = 12$									
1	8.22	21	2000	8.22	23	153	8.22	23	117
2	9.07	28	643	8.50	22	617	8.09	20	319
3	8.28	22	153	8.15	18	220	0.00	0	12
4	8.44	24	114	9.51	29	230	9.51	29	146
5	8.93	23	233	7.77	19	274	0.00	0	11
6	8.91	22	113	8.24	22	249	8.24	22	165
7	8.38	20	159	8.98	24	566	7.50	20	118
8	8.17	21	342	6.98	15	326	0.00	0	10
9	8.15	23	2000	5.70	13	152	5.33	24	90
10	8.06	23	2000	8.60	21	116	8.60	21	82
Dimension $n = 20$									
1	10.55	32	188	11.53	38	282	0.00	0	11
2	10.53	36	2000	12.07	42	430	10.31	34	326
3	9.26	31	2000	11.59	38	149	0.00	0	11
4	11.35	40	563	10.75	34	359	12.21	38	170
5	11.85	42	2000	11.71	41	1130	12.14	42	384
6	12.18	39	267	12.35	45	251	7.96	42	110
7	12.04	41	1282	11.77	42	142	11.77	42	170
8	10.59	31	507	11.83	41	411	11.98	42	351
9	0.87	30	2000	11.56	37	169	11.07	34	189
10	10.87	32	2000	11.75	37	422	8.93	47	100
Dimension $n = 30$									
1	12.89	49	2000	14.16	57	304	13.56	51	140
2	0.01	40	2000	15.58	65	926	15.03	60	398
3	14.46	52	2000	16.00	61	936	13.60	51	239
4	2.07	50	2000	14.28	54	319	13.81	54	241
5	12.30	42	2000	14.40	57	510	14.84	56	437

Table 6 continued

Inst. #	BCD			Proximal BCD			GCG		
	Val.	$\sum_{i=1}^{d} \|x_i\|_0$	Iter.	Val.	$\sum_{i=1}^{d} \|x_i\|_0$	Iter.	Val.	$\sum_{i=1}^{d} \|x_i\|_0$	Iter.
6	0.69	42	2000	13.97	52	491	13.69	51	272
7	0.63	35	2000	14.53	59	227	13.77	53	253
8	14.31	52	2000	15.20	54	660	14.28	54	346
9	0.02	34	2000	14.55	55	263	13.37	48	143
10	0.77	37	2000	15.11	57	283	14.03	54	145

5.2 Computing the leading sparse principal component of tensor

In this subsection, we consider the problem (1.4) of finding the leading sparse principal component of a given tensor. To apply the GCG method in the previous section, we adopt $\|\cdot\|_1$ as regularizer, and arrive at the following formulation

$$\begin{aligned} \min\ & -\mathcal{T}(x_1, x_2, \ldots, x_d) + \alpha \sum_{i=1}^{d} \|x_i\|_1 \\ \text{s.t.}\ & \|x_i\|_2 \le 1, i = 1, 2, \ldots, d. \end{aligned} \tag{5.4}$$

The subproblem in GCG is in the form of $\min_{\|y\|_2^2 \le 1}\{-y^\top b + \rho\|y\|_1\}$, which has a closed form solution

$$y^* = \begin{cases} z/\|z\|_2, & \text{if } \|z\|_2 \ne 0 \\ 0, & \text{otherwise.} \end{cases}$$

where $z(j) = \text{sign}(b(j)) \max\{|b(j)| - \rho, 0\}\ \forall\ j = 1, 2, \ldots, n$.

One undesirable property of the formulation (5.4) is that we may possibly get a zero solution, i.e. $x_i = 0$ for some i, which leads to $\mathcal{T}(x_1, x_2, \ldots, x_d) = 0$. To prevent this from happening, we also apply the BCD method and proximal BCD method to the following equality constrained problem:

$$\begin{aligned} \min\ & -\mathcal{T}(x_1, x_2, \ldots, x_d) + \alpha \sum_{i=1}^{d} \|x_i\|_1 \\ \text{s.t.}\ & \|x_i\|_2 = 1, i = 1, 2, \ldots, d, \end{aligned} \tag{5.5}$$

and compare the results with those returned by our proposed algorithms in Table 6.

In the tests, we let $\alpha = 0.85$, and set the maximum iteration number to be 2000. For each fixed dimension, we randomly generate 10 instances which are the fourth order tensors and the corresponding problems are solved by the three methods, starting from the same initial point. In Table 6, 'Val.' refers to the value $\mathcal{T}(x_1, x_2, \ldots, x_d)$. From this table, we see that GCG is capable of finding a nonzero local optimum within a few hundred steps in most cases, with reasonably sparsity. The three approaches in Table 6 are comparable to each other in terms of the value $\mathcal{T}(x_1, x_2, \ldots, x_d)$, but BCD consumes the maximum 2000 iterations in quite a few instances, while GCG finds the best local optimum in a few instances (e.g. instances 6 and 9 for $n = 8$, and instances 5 and 8 for $n = 20$).

Acknowledgements We would like to thank Professor Renato D. C. Monteiro and two anonymous referees for their insightful comments, which helped improve this paper significantly.

References

1. Allen, G.: Sparse higher-order principal components analysis. In: The 15th International Conference on Artificial Intelligence and Statistics (2012)
2. Ames, B., Hong, M.: Alternating direction method of multipliers for penalized zero-variance discriminant analysis. Comput. Optim. Appl. **64**(3), 725–754 (2016). https://doi.org/10.1007/s10589-016-9828-y
3. Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka–Łojasiewicz inequality. Math. Oper. Res. **35**(2), 438–457 (2010)
4. Bach, F.: Duality between subgradient and conditional gradient methods. SIAM J. Optim. **25**(1), 115–129 (2015)
5. Beck, A., Shtern, S.: Linearly convergent away-step conditional gradient for nonstrongly convex functions. Math. Program. **164**(1–2), 1–27 (2017)
6. Bian, W., Chen, X.: Worst-case complexity of smoothing quadratic regularization methods for non-Lipschitzian optimization. SIAM J. Optim. **23**, 1718–1741 (2013)
7. Bian, W., Chen, X., Ye, Y.: Complexity analysis of interior point algorithms for non-Lipschitz and nonconvex minimization. Math. Program. **149**, 301–327 (2015)
8. Bolte, J., Daniilidis, A., Lewis, A.: The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. SIAM J. Optim. **17**, 1205–1223 (2006)
9. Bolte, J., Daniilidis, A., Lewis, A., Shiota, M.: Clarke subgradients of stratifiable functions. SIAM J. Optim. **18**, 556–572 (2007)
10. Bolte, J., Daniilidis, A., Ley, O., Mazet, L.: Characterizations of Łojasiewicz inequalities: subgradient flows, talweg, convexity. Trans. Am. Math. Soc. **362**(6), 3319–3363 (2010)
11. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math. Program. **146**, 459–494 (2014)
12. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2011)
13. Bredies, K.: A forward-backward splitting algorithm for the minimization of non-smooth convex functionals in Banach space. Inverse Probl. **25**(1), 711–723 (2009)
14. Bredies, K., Lorenz, D.A., Maass, P.: A generalized conditional gradient method and its connection to an iterative shrinkage method. Comput. Optim. Appl. **42**(2), 173–193 (2009)
15. Candès, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted ℓ_1 minimization. J. Fourier Anal. Appl. **14**(5–6), 877–905 (2008)
16. Cartis, C., Gould, N.I.M., Toint, PhL: On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization. SIAM J. Optim. **20**(6), 2833–2852 (2010)
17. Cartis, C., Gould, N.I.M., Toint, P.L.: Adaptive cubic overestimation methods for unconstrained optimization. Part II: worst-case function-evaluation complexity. Math. Program. Ser. A **130**(2), 295–319 (2011)
18. Cartis, C., Gould, N.I.M., Toint, P.L.: An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity. IMA J. Numer. Anal. **32**, 1662–1695 (2012)
19. Chen, X., Ge, D., Wang, Z., Ye, Y.: Complexity of unconstrained l_2-l_p minimization. Math. Program. **143**, 371–383 (2014)
20. Curtis, F., Robinson, D.P., Samadi, M.: A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. Math. Program. **162**, 1–32 (2017)

21. Devolder, O., François, G., Nesterov, Yu.: First-order methods of smooth convex optimization with inexact oracle. Math. Program. Ser. A **146**, 37–75 (2014)
22. Dutta, J., Deb, K., Tulshyan, R., Arora, R.: Approximate KKT points and a proximity measure for termination. J. Glob. Optim. **56**, 1463–1499 (2013)
23. Fan, J., Li, R.: Variable selection via nonconcave penalized likelihood and its oracle properties. J. Am. Stat. Assoc. **96**(456), 1348–1360 (2001)
24. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Nav. Res. Logist. Q. **3**, 95–110 (1956)
25. Freund, R.M., Grigas, P.: New analysis and results for the Frank–Wolfe method. Math. Program. **155**, 199–230 (2016)
26. Gao, X., Jiang, B., Zhang, S.: On the information-adaptive variants of the ADMM: an iteration complexity perspective. J. Sci. Comput. **76**, 327–363 (2018)
27. Ge, D., He, R., He, S.: A three criteria algorithm for l_2-l_p minimization problem with linear constraints. Math. Program. **166**(1), 131–158 (2017)
28. Ghadimi, S., Lan, G., Zhang, H.: Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. Math. Program. **155**(1), 1–39 (2016)
29. Gong, P., Zhang, C., Lu, Z., Huang, J., Ye, J.: A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In: ICML, pp. 37–45 (2013)
30. Harchaoui, Z., Juditsky, A., Nemirovski, A.: Conditional gradient algorithms for norm-regularized smooth convex optimization. Math. Program. **152**, 75–112 (2015)
31. Hong, M.: A distributed, asynchronous and incremental algorithm for nonconvex optimization: an ADMM based approach. IEEE Trans. Control Netw. Syst. **5**(3), 935–945 (2018)
32. Hong, M.: Decomposing linearly constrained nonconvex problems by a proximal primal dual approach: algorithms, convergence, and applications. arXiv:1604.00543 (2016)
33. Hong, M., Luo, Z.-Q., Razaviyayn, M.M.: Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. SIAM J. Optim. **26**(1), 337–364 (2016)
34. Jaggi, M.: Revisiting Frank–Wolfe: projection-free sparse convex optimization. In: ICML (2013)
35. Jiang, B., Yang, F., Zhang, S.: Tensor and its Tucker core: the invariance relationships. Numer. Linear Algebra Appl. **24**(3), e2086 (2017)
36. Kurdyka, K.: On gradients of functions definable in o-minimal structures. Ann. Inst. Fourier **146**, 769–783 (1998)
37. Lan, G., Zhou, Y.: Conditional gradient sliding for convex optimization. SIAM J. Optim. **26**(2), 1379–1409 (2016)
38. Li, G., Pong, T.K.: Global convergence of splitting methods for nonconvex composite optimization. SIAM J. Optim. **25**(4), 2434–2460 (2015)
39. Lin, T., Ma, S., Zhang, S.: Global convergence of unmodified 3-block ADMM for a class of convex minimization problems. J. Sci. Comput **76**, 69–88 (2018)
40. Lin, T., Ma, S., Zhang, S.: Iteration complexity analysis of multi-block ADMM for a family of convex minimization without strong convexity. J. Sci. Comput. **69**(1), 52–81 (2016)
41. Liu, Y., Ma, S., Dai, Y., Zhang, S.: A smoothing SQP framework for a class of composite ℓ_q minimization over polyhedron. Math. Program. Ser. A **158**(1), 467–500 (2016)
42. Łojasiewicz, S.: Une propriété topologique des sous-ensembles analytiques réels, Les Équations aux Dérivées Partielles. Éditions du centre National de la Recherche Scientifique, Paris (1963)
43. Lacoste-Julien, S.: Convergence rate of Frank–Wolfe for non-convex objectives. Preprint arXiv:1607.00345 (2016)
44. Lafond, J., Wai, H.-T., Moulines, E.: On the Online Frank–Wolfe algorithms for convex and non-convex optimizations. Preprint arXiv:1510.01171
45. Luss, R., Teboulle, M.: Conditional gradient algorithms for rank one matrix approximations with a sparsity constraint. SIAM Rev. **55**, 65–98 (2013)
46. Martınez, J.M., Raydan, M.: Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. J. Glob. Optim. **68**, 367–385 (2017)
47. Mu, C., Zhang, Y., Wright, J., Goldfarb, D.: Scalable robust matrix recovery: Frank–Wolfe meets proximal methods. SIAM J. Sci. Comput. **38**(5), 3291–3317 (2016)
48. Nesterov, Y.: Introductory Lectures on Convex Optimization. Applied Optimization. Kluwer Academic Publishers, Boston, MA (2004)
49. Ngai, H.V., Luc, D.T., Théra, M.: Extensions of Fréchet ϵ-subdifferential calculus and applications. J. Math. Anal. Appl. **268**, 266–290 (2002)

50. Rockafellar, R.T., Wets, R.: Variational Analysis. Volume 317 of Grundlehren der Mathematischen Wissenschafte. Springer, Berlin (1998)
51. Shen, Y., Wen, Z., Zhang, Y.: Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization. Optim. Methods Softw. **29**(2), 239–263 (2014)
52. Wang, F., Cao, W., Xu, Z.: Convergence of multiblock Bregman ADMM for nonconvex composite problems. Preprint arXiv:1505.03063 (2015)
53. Wang, Y., Yin, W., Zeng, J.: Global convergence of ADMM in nonconvex nonsmooth optimization. J. Sci. Comput. 1–35 (2018)
54. Wen, Z., Yin, W., Zhang, Y.: Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Math. Program. Comput. **4**(4), 333–361 (2012)
55. Xu, Y.: Alternating proximal gradient method for sparse nonnegative Tucker decomposition. Math. Program. Comput. **7**(1), 39–70 (2015)
56. Yang, L., Pong, T.K., Chen, X.: Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction. SIAM J. Imaging Sci. **10**, 74–110 (2017)
57. Yu, Y., Zhang, X., Schuurmans, D.: Generalized conditional gradient for sparse estimation. Preprint arXiv:1410.4828v1 (2014)
58. Zhang, C.-H.: Nearly unbiased variable selection under minimax concave penalty. Ann. Stat. **38**(2), 894–942 (2010)
59. Zhang, T.: Analysis of multi-stage convex relaxation for sparse regularization. J. Mach. Learn. Res. **11**, 1081–1107 (2010)
60. Zhang, T.: Multi-stage convex relaxation for feature selection. Bernoulli **19**(5B), 2277–2293 (2013)

Affiliations

Bo Jiang[1] · Tianyi Lin[2] · Shiqian Ma[3] · Shuzhong Zhang[4,5]

✉ Bo Jiang
isyebojiang@gmail.com

[1] Research Institute for Interdisciplinary Sciences, School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China

[2] Department of Industrial Engineering and Operations Research, UC Berkeley, Berkeley, CA 94720, USA

[3] Department of Mathematics, UC Davis, Davis, CA 95616, USA

[4] Department of Industrial and Systems Engineering, University of Minnesota, Minneapolis, MN 55455, USA

[5] Institute of Data and Decision Analytics, The Chinese University of Hong Kong (Shenzhen), and Shenzhen Research Institute of Big Data, Shenzhen, China