J Sci Comput
DOI 10.1007/s10915-017-0376-0

A Globally Convergent Algorithm for Nonconvex Optimization Based on Block Coordinate Update

Yangyang Xu[1] · **Wotao Yin**[2]

Received: 14 March 2016 / Revised: 31 October 2016 / Accepted: 24 January 2017

Abstract Nonconvex optimization arises in many areas of computational science and engineering. However, most nonconvex optimization algorithms are only known to have local convergence or subsequence convergence properties. In this paper, we propose an algorithm for nonconvex optimization and establish its global convergence (of the whole sequence) to a critical point. In addition, we give its asymptotic convergence rate and numerically demonstrate its efficiency. In our algorithm, the variables of the underlying problem are either treated as one block or multiple disjoint blocks. It is assumed that each non-differentiable component of the objective function, or each constraint, applies only to one block of variables. The differentiable components of the objective function, however, can involve multiple blocks of variables together. Our algorithm updates one block of variables at a time by minimizing a certain prox-linear surrogate, along with an extrapolation to accelerate its convergence. The order of update can be either deterministically cyclic or randomly shuffled for each cycle. In fact, our convergence analysis only needs that each block be updated at least once in every fixed number of iterations. We show its global convergence (of the whole sequence) to a critical point under fairly loose conditions including, in particular, the Kurdyka–Łojasiewicz condition, which is satisfied by a broad class of nonconvex/nonsmooth applications. These results, of course, remain valid when the underlying problem is convex. We apply our convergence results to the coordinate descent iteration for non-convex regularized linear regression, as well as a modified rank-one residue iteration for nonnegative matrix factorization. We show that both applications have global convergence. Numerically, we tested our algorithm on non-

This work is supported in part by NSF DMS-1317602, EECS-1462397, and ONR N000141712162.

✉ Yangyang Xu
yangyang.xu@ua.edu

Wotao Yin
wotaoyin@math.ucla.edu

[1] Department of Mathematics, University of Alabama, Tuscaloosa, AL, USA

[2] Department of Mathematics, UCLA, Los Angeles, CA, USA

Published online: 04 February 2017

negative matrix and tensor factorization problems, where random shuffling clearly improves the chance to avoid low-quality local solutions.

Keywords Nonconvex optimization · Nonsmooth optimization · Block coordinate descent · Kurdyka–Łojasiewicz inequality · Prox-linear · Whole sequence convergence

1 Introduction

In this paper, we consider (nonconvex) optimization problems in the form of

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}}\ F(\mathbf{x}_1,\ldots,\mathbf{x}_s) \equiv f(\mathbf{x}_1,\ldots,\mathbf{x}_s) + \sum_{i=1}^{s} r_i(\mathbf{x}_i), \\ &\text{subject to } \mathbf{x}_i \in \mathcal{X}_i,\ i = 1,\ldots,s, \end{aligned} \tag{1}$$

where variable $\mathbf{x} = (\mathbf{x}_1,\ldots,\mathbf{x}_s) \in \mathbb{R}^n$ has s blocks, $s \geq 1$, function f is continuously differentiable, functions r_i, $i = 1,\ldots,s$, are proximable[1] but not necessarily differentiable. It is standard to assume that both f and r_i are closed and proper and the sets $\mathcal{X}_i$ are closed and nonempty. Convexity is *not* assumed for f, r_i, or $\mathcal{X}_i$. By allowing r_i to take the ∞-value, $r_i(\mathbf{x}_i)$ can incorporate the constraint $\mathbf{x}_i \in \mathcal{X}_i$ since enforcing the constraint is equivalent to minimizing the indicator function of $\mathcal{X}_i$, and r_i can remain proper and closed. Therefore, in the remainder of this paper, we do not include the constraints $\mathbf{x}_i \in \mathcal{X}_i$. The functions r_i can incorporate regularization functions, often used to enforce certain properties or structures in $\mathbf{x}_i$, for example, the nonconvex ℓ_p quasi-norm, $0 \leq p < 1$, which promotes solution sparsity.

Special cases of (1) include the following nonconvex problems: ℓ_p-quasi-norm ($0 \leq p < 1$) regularized sparse regression problems [10,32,42], sparse dictionary learning [1,40,62], matrix rank minimization [50], matrix factorization with nonnegativity/sparsity/orthogonality regularization [27,33,47], (nonnegative) tensor decomposition [29,57], and (sparse) higher-order principal component analysis [2].

Due to the lack of convexity, standard analysis tools such as convex inequalities and Fejér-monotonicity cannot be applied to establish the convergence of the iterate sequence. The case becomes more difficult when the problem is nonsmooth. In these cases, convergence analysis of existing algorithms is typically limited to objective convergence (to a possibly non-minimal value) or the convergence of a certain subsequence of iterates to a critical point. (Some exceptions will be reviewed below.) Although whole-sequence convergence is almost always observed, it is rarely proved. This deficiency abates some widely used algorithms. For example, KSVD [1] only has nonincreasing monotonicity of its objective sequence, and iterative reweighted algorithms for sparse and low-rank recovery in [17,32,41] only has subsequence convergence. Some other methods establish whole sequence convergence by assuming stronger conditions such as local convexity (on at least a part of the objective) and either unique or isolated limit points, which may be difficult to satisfy or to verify. In this paper, we aim to establish whole sequence convergence with conditions that are provably satisfied by a wide class of functions.

Block coordinate descent (BCD) (more precisely, block coordinate update) is very general and widely used for solving both convex and nonconvex problems in the form of (1) with multiple blocks of variables. Since only one block is updated at a time, it has a low per-iteration

[1] A function f is proximable if it is easy to obtain the minimizer of $f(x) + \frac{1}{2\gamma}\|x - y\|^2$ for any input y and $\gamma > 0$.

cost and small memory footprint. Recent literature [8,26,38,43,48,51,53] has found BCD as a viable approach for "big data" problems.

1.1 Proposed Algorithm

In order to solve (1), we propose a block prox-linear (BPL) method, which updates a block of variables at each iteration by minimizing a prox-linear surrogate function. Specifically, at iteration k, a block $b_k \in \{1, \ldots, s\}$ is selected and $\mathbf{x}^k = (\mathbf{x}_1^k, \ldots, \mathbf{x}_s^k)$ is updated as follows: for $i = 1, \ldots, s$,

$$\begin{cases} \mathbf{x}_i^k = \mathbf{x}_i^{k-1}, & \text{if } i \neq b_k, \\ \mathbf{x}_i^k \in \arg\min\limits_{\mathbf{x}_i} \langle \nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k), \mathbf{x}_i \rangle + \frac{1}{2\alpha_k} \|\mathbf{x}_i - \hat{\mathbf{x}}_i^k\|^2 + r_i(\mathbf{x}_i), & \text{if } i = b_k, \end{cases} \tag{2}$$

where $(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k)$ denotes the point $(\mathbf{x}_1^{k-1}, \ldots, \mathbf{x}_{i-1}^{k-1}, \hat{\mathbf{x}}_i^k, \mathbf{x}_{i+1}^{k-1}, \ldots, \mathbf{x}_s^{k-1})$, $\alpha_k > 0$ is a step-size and $\hat{\mathbf{x}}_i^k$ is the extrapolation

$$\hat{\mathbf{x}}_i^k = \mathbf{x}_i^{k-1} + \omega_k(\mathbf{x}_i^{k-1} - \mathbf{x}_i^{\text{prev}}), \tag{3}$$

where $\omega_k \geq 0$ is an extrapolation weight and $\mathbf{x}_i^{\text{prev}}$ is the value of $\mathbf{x}_i$ before it was updated to $\mathbf{x}_i^{k-1}$. The framework of our method is given in Algorithm 1. At each iteration k, only the block b_k is updated.

Algorithm 1: Randomized/deterministic block prox-linear (BPL) method for problem (1)

Initialization: $\mathbf{x}^{-1} = \mathbf{x}^0$.
for $k = 1, 2, \ldots$ **do**
  Pick $b_k \in \{1, 2, \ldots, s\}$ in a deterministic or random manner.
  Set α_k, ω_k and let $\mathbf{x}^k \leftarrow$ (2).
  if *stopping criterion is satisfied* **then**
    Return $\mathbf{x}^k$.

While we can simply set $\omega_k = 0$, appropriate $\omega_k > 0$ can speed up the convergence; we will demonstrate this in the numerical results below. We can set the stepsize $\alpha_k = \frac{1}{\gamma L_k}$ with any $\gamma > 1$, where $L_k > 0$ is the Lipschitz constant of $\nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \mathbf{x}_i)$ about $\mathbf{x}_i$. When L_k is unknown or difficult to bound, we can apply backtracking on α_k under the criterion:

$$f(\mathbf{x}^k) \leq f(\mathbf{x}^{k-1}) + \left\langle \nabla_{\mathbf{x}_i} f(\mathbf{x}^{k-1}), \mathbf{x}_i^k - \mathbf{x}_i^{k-1} \right\rangle + \frac{1}{2\gamma\alpha_k} \|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\|^2.$$

1.2 Special Cases

When there is only one block, i.e., $s = 1$, Algorithm 1 reduces to the well-known (accelerated) proximal gradient method (e.g., [7,22,44]). When the update block cycles from 1 through s, Algorithm 1 reduces to the cyclic block proximal gradient (Cyc-BPG) method in [8,61]. We can also randomly shuffle the s blocks at the beginning of each cycle. We demonstrate in Sect. 3 that random shuffling leads to better numerical performance. When the update block

is randomly selected following the probability $p_i > 0$, where $\sum_{i=1}^{s} p_i = 1$, Algorithm 1 reduces to the randomized block coordinate descent method (RBCD) (e.g., [37,38,43,51]). Unlike these existing results, we do not assume convexity.

In our analysis, we impose an essentially cyclic assumption—each block is selected for update at least once within every $T \geq s$ consecutive iterations—otherwise the order is arbitrary. Our convergence results apply to all the above special cases except RBCD, whose convergence analysis requires different strategies; see [38,43,51] for the convex case and [37] for the nonconvex case.

1.3 Kurdyka–Łojasiewicz Property

To establish whole sequence convergence of Algorithm 1, a key assumption is the Kurdyka–Łojasiewicz (KL) property of the objective function F.

A lot of functions are known to satisfy the KL property. Recent works [4, section 4] and [61, section 2.2] give many specific examples that satisfy the property, such as the ℓ_p-(quasi)norm $\|\mathbf{x}\|_p$ with $p \in [0, +\infty]$, any piecewise polynomial functions, indicator functions of polyhedral set, orthogonal matrix set, and positive semidefinite cone, matrix rank function, and so on.

Definition 1 (Kurdyka–Łojasiewicz property) A function $\psi(\mathbf{x})$ satisfies the KL property at point $\bar{\mathbf{x}} \in \mathrm{dom}(\partial\psi)$ if there exist $\eta > 0$, a neighborhood $\mathcal{B}_\rho(\bar{\mathbf{x}}) \triangleq \{\mathbf{x} : \|\mathbf{x} - \bar{\mathbf{x}}\| < \rho\}$, and a concave function $\phi(a) = c \cdot a^{1-\theta}$ for some $c > 0$ and $\theta \in [0, 1)$ such that for any $\mathbf{x} \in \mathcal{B}_\rho(\bar{\mathbf{x}}) \cap \mathrm{dom}(\partial\psi)$ and $\psi(\bar{\mathbf{x}}) < \psi(\mathbf{x}) < \psi(\bar{\mathbf{x}}) + \eta$, it holds

$$\phi'(|\psi(\mathbf{x}) - \psi(\bar{\mathbf{x}})|)\mathrm{dist}(\mathbf{0}, \partial\psi(\mathbf{x})) \geq 1, \tag{4}$$

where $\mathrm{dom}(\partial\psi) = \{\mathbf{x} : \partial\psi(\mathbf{x}) \neq \emptyset\}$ and $\mathrm{dist}(\mathbf{0}, \partial\psi(\mathbf{x})) = \min\{\|\mathbf{y}\| : \mathbf{y} \in \partial\psi(\mathbf{x})\}$.

The KL property was introduced by Łojasiewicz [36] for real analytic functions. Kurdyka [31] extended it to functions of the o-minimal structure. Recently, the KL inequality (4) was further extended to nonsmooth sub-analytic functions [11]. The work [12] characterizes the geometric meaning of the KL inequality.

1.4 Related Literature

There are many methods that solve general nonconvex problems. Methods in the papers [6,15,18,21], the books [9,45], and in the references therein, do not break variables into blocks. They usually have the properties of local convergence or subsequence convergence to a critical point, or global convergence in terms of the violation of optimality conditions. Next, we review BCD methods, which can significantly outperform their full coordinate update if the problems or the updates satisfy the *coordinate-friendly* structure [48,54].

BCD has been extensively used in many applications. Its original form, block coordinate minimization (BCM), which updates a block by minimizing the original objective with respect to that block, dates back to the 1950s [24] and is closely related to the Gauss–Seidel and SOR methods for linear equation systems. Its convergence was studied under a variety of settings (cf. [23,49,55] and the references therein). The convergence rate of BCM was established under the strong convexity assumption [39] for the multi-block case and under the general convexity assumption [8] for the two-block case. To have even cheaper updates, one can update a block approximately, for example, by minimizing an approximate objective like was done in (2), instead of sticking to the original objective. The work [56] is a block coordinate gradient descent (BCGD) method where taking a block gradient step is equivalent

to minimizing a certain prox-linear approximation of the objective. Its whole sequence convergence and local convergence rate were established under the assumptions of a so-called *local Lipschitzian error bound* and the convexity of the objective's nondifferentiable part. The randomized block coordinate descent (RBCD) method in [37,43] randomly chooses the block to update at each iteration and is not essentially cyclic. Objective convergence was established [43,51], and the violation of the first-order optimality condition was shown to converge to *zero* [37]. There is no iterate convergence result for RBCD.

Some special cases of Algorithm 1 have been analyzed in the literature. The work [61] uses cyclic updates of a fixed order and assumes block-wise convexity; [13] studies two blocks without extrapolation, namely, $s = 2$ and $\hat{\mathbf{x}}_i^k = \mathbf{x}_i^{k-1}$, $\forall k$ in (2). A more general result is [5, Lemma 2.6], where three conditions for whole sequence convergence are given and are met by methods including averaged projection, proximal point, and forward-backward splitting. Algorithm 1, however, does not satisfy the three conditions in [5].

The extrapolation technique in (3) has been applied to accelerate the (block) prox-linear method for solving convex optimization problems (e.g., [7,38,44,51]). Recently, [22,61] show that the (block) prox-linear iteration with extrapolation can still converge if the nonsmooth part of the problem is convex, while the smooth part can be nonconvex. Because of the convexity assumption, their convergence results do not apply to Algorithm 1 for solving the general nonconvex problem (1). Numerically, [35,58] demonstrate that extrapolation technique can also accelerate algorithms for nonconvex matrix factorization problems.

1.5 Contributions

We summarize the main contributions of this paper as follows.

- We propose a block prox-linear (BPL) method for nonconvex smooth and nonsmooth optimization. Extrapolation is used to accelerate it. To our best knowledge, this is the first work of prox-linear acceleration for fully nonconvex problems (where both smooth and nonsmooth terms are nonconvex) with a convergence guarantee. However, we have not proved any improved convergence rate.
- Assuming essentially cyclic updates of the blocks, we obtain the whole sequence convergence of BPL to a critical point with rate estimates, by first establishing subsequence convergence and then applying the Kurdyka–Łojasiewicz (KL) property. Furthermore, we tailor our convergence analysis to several existing algorithms, including non-convex regularized linear regression and nonnegative matrix factorization, to improve their existing convergence results.
- We numerically tested BPL on nonnegative matrix and tensor factorization problems. At each cycle of updates, the blocks were randomly shuffled. We observed that BPL was very efficient and that random shuffling avoided local solutions more effectively than the deterministic cyclic order.

1.6 Notation and Preliminaries

We restrict our discussion in $\mathbb{R}^n$ equipped with the Euclidean norm, denoted by $\|\cdot\|$. However, all our results can be extended to general of primal and dual norm pairs. The lower-case letter s is reserved for the number of blocks and $\ell, L, L_k, \ldots$ for various Lipschitz constants. $\mathbf{x}_{<i}$ is short for $(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1})$, $\mathbf{x}_{>i}$ for $(\mathbf{x}_{i+1}, \ldots, \mathbf{x}_s)$, and $\mathbf{x}_{\neq i}$ for $(\mathbf{x}_{<i}, \mathbf{x}_{>i})$. We simplify $f(\mathbf{x}_{<i}, \hat{\mathbf{x}}_i, \mathbf{x}_{>i})$ to $f(\mathbf{x}_{\neq i}, \hat{\mathbf{x}}_i)$. The distance of a point $\mathbf{x}$ to a set $\mathcal{Y}$ is denoted by $\operatorname{dist}(\mathbf{x}, \mathcal{Y}) = \inf_{\mathbf{y}\in\mathcal{Y}} \|\mathbf{x} - \mathbf{y}\|$.

Since the update may be aperiodic, extra notation is used for when and how many times a block is updated. Let $\mathcal{K}[i,k]$ denote the set of iterations in which the ith block has been selected to update till the kth iteration:

$$\mathcal{K}[i,k] \triangleq \{\kappa : b_\kappa = i,\ 1 \le \kappa \le k\} \subseteq \{1,\dots,k\}, \tag{5}$$

and let

$$d_i^k \triangleq \big|\mathcal{K}[i,k]\big|,$$

which is the number of times the ith block has been updated till iteration k. For $k = 1,\dots,$ we have $\cup_{i=1}^s \mathcal{K}[i,k] = [k] \triangleq \{1,2,\dots,k\}$ and $\sum_{i=1}^s d_i^k = k$.

Let $\mathbf{x}^k$ be the value of $\mathbf{x}$ after the kth iteration, and for each block i, $\tilde{\mathbf{x}}_i^j$ be the value of $\mathbf{x}_i$ after its jth update. By letting $j = d_i^k$, we have $\mathbf{x}_i^k = \tilde{\mathbf{x}}_i^j$.

The extrapolated point in (2) (for $i = b_k$) is computed from the last two updates of the same block:

$$\hat{\mathbf{x}}_i^k = \tilde{\mathbf{x}}_i^{j-1} + \omega_k(\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^{j-2}),\ \text{where } j = d_i^k, \tag{6}$$

for some weight $0 \le \omega_k \le 1$. We partition the set of Lipschitz constants and the extrapolation weights into s disjoint subsets as

$$\{L_\kappa : 1 \le \kappa \le k\} = \cup_{i=1}^s \{L_\kappa : \kappa \in \mathcal{K}[i,k]\} \triangleq \cup_{i=1}^s \left\{\tilde{L}_i^j : 1 \le j \le d_i^k\right\}, \tag{7a}$$

$$\{\omega_\kappa : 1 \le \kappa \le k\} = \cup_{i=1}^s \{\omega_\kappa : \kappa \in \mathcal{K}[i,k]\} \triangleq \cup_{i=1}^s \left\{\tilde{\omega}_i^j : 1 \le j \le d_i^k\right\}. \tag{7b}$$

Hence, for each block i, we have three sequences:

$$\text{value of } \mathbf{x}_i\colon \tilde{\mathbf{x}}_i^1, \tilde{\mathbf{x}}_i^2, \dots, \tilde{\mathbf{x}}_i^{d_i^k}, \dots; \tag{8a}$$

$$\text{Lipschitz constant: } \tilde{L}_i^1, \tilde{L}_i^2, \dots, \tilde{L}_i^{d_i^k}, \dots; \tag{8b}$$

$$\text{extrapolation weight: } \tilde{\omega}_i^1, \tilde{\omega}_i^2, \dots, \tilde{\omega}_i^{d_i^k}, \dots. \tag{8c}$$

For simplicity, we take stepsizes and extrapolation weights as follows

$$\alpha_k = \frac{1}{2L_k},\ \forall k, \qquad \tilde{\omega}_i^j \le \frac{\delta}{6}\sqrt{\tilde{L}_i^{j-1}/\tilde{L}_i^j},\ \forall i, j,\ \text{for some } \delta < 1. \tag{9}$$

However, if the problem (1) has more structures such as block convexity, we can use larger α_k and ω_k; see Remark 2. Table 1 summarizes the notation. In addition, we initialize $\tilde{\mathbf{x}}_i^{-1} = \tilde{\mathbf{x}}_i^0 = \mathbf{x}_i^0,\ \forall i$.

We make the following definitions, which can be found in [52].

Definition 2 *(Limiting Fréchet subdifferential* [30]*)* A vector $\mathbf{g}$ is a Fréchet subgradient of a lower semicontinuous function F at $\mathbf{x} \in \mathrm{dom}(F)$ if

$$\liminf_{\mathbf{y}\to\mathbf{x},\mathbf{y}\neq\mathbf{x}} \frac{F(\mathbf{y}) - F(\mathbf{x}) - \langle \mathbf{g}, \mathbf{y}-\mathbf{x}\rangle}{\|\mathbf{y}-\mathbf{x}\|} \ge 0.$$

The set of Fréchet subgradient of F at $\mathbf{x}$ is called Fréchet subdifferential and denoted as $\hat{\partial} F(\mathbf{x})$. If $\mathbf{x} \notin \mathrm{dom}(F)$, then $\hat{\partial} F(\mathbf{x}) = \emptyset$.

The limiting Fréchet subdifferential is denoted by $\partial F(\mathbf{x})$ and defined as

$$\partial F(\mathbf{x}) = \{\mathbf{g}\text{: there is } \mathbf{x}_m \to \mathbf{x} \text{ and } \mathbf{g}_m \in \hat{\partial} F(\mathbf{x}_m) \text{ such that } \mathbf{g}_m \to \mathbf{g}\}.$$

Table 1 Summary of notation

Notion	Definition
s	The total number of blocks
b_k	The update block selected at the kth iteration
$\mathcal{K}[i,k]$	The set of iterations up to k in which $\mathbf{x}_i$ is updated; see (5)
d_i^k	$\lvert\mathcal{K}[i,k]\rvert$: the number of updates to $\mathbf{x}_i$ within the first k iterations
$\mathbf{x}^k$	The value of $\mathbf{x}$ after the kth iteration
$\tilde{\mathbf{x}}_i^j$	The value of $\mathbf{x}_i$ after its jth update; see (8a)
L_k	Gradient Lipschitz constant of the update block at the kth iteration; see (11)
$\tilde{L}_i^j$	Gradient Lipschitz constant of block i at its jth update; see (7a) and (8b)
ω_k	The extrapolation weight used at the kth iteration
$\tilde{\omega}_i^j$	The extrapolation weight used at the jth update of $\mathbf{x}_i$; see (7b) and (8c)

If F is differentiable[2] at $\mathbf{x}$, then $\partial F(\mathbf{x}) = \hat{\partial} F(\mathbf{x}) = \{\nabla F(\mathbf{x})\}$; see [52, Exercise 8.8] for example, and if F is convex, then $\partial F(\mathbf{x}) = \{\mathbf{g}: F(\mathbf{y}) \geq F(\mathbf{x}) + \langle \mathbf{g}, \mathbf{y} - \mathbf{x}\rangle,\ \forall \mathbf{y} \in \mathrm{dom}(F)\}$. We use the limiting subdifferential for general nonconvex nonsmooth functions. For problem (1), it holds that (see [4, Lemma 2.1] or [52, Prop. 10.6, pp. 426])

$$\partial F(\mathbf{x}) = \{\nabla_{\mathbf{x}_1} f(\mathbf{x}) + \partial r_1(\mathbf{x}_1)\} \times \cdots \times \{\nabla_{\mathbf{x}_s} f(\mathbf{x}) + \partial r_s(\mathbf{x}_s)\}, \tag{10}$$

where $\mathcal{X}_1 \times \mathcal{X}_2$ denotes the Cartesian product of $\mathcal{X}_1$ and $\mathcal{X}_2$.

Definition 3 *(Critical point)* A point $\mathbf{x}^*$ is called a critical point of F if $\mathbf{0} \in \partial F(\mathbf{x}^*)$.

Definition 4 *(Proximal mapping)* For a proper, lower semicontinuous function r, its proximal mapping $\mathbf{prox}_r(\cdot)$ is defined as

$$\mathbf{prox}_r(\mathbf{x}) = \arg\min_{\mathbf{y}} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|^2 + r(\mathbf{y}).$$

As r is nonconvex, $\mathbf{prox}_r(\cdot)$ is generally set-valued. Using this notation, the update in (2) can be written as (assume $i = b_k$)

$$\mathbf{x}_i^k \in \mathbf{prox}_{\alpha_k r_i}\left(\hat{\mathbf{x}}_i^k - \alpha_k \nabla_{\mathbf{x}_i} f\left(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k\right)\right)$$

1.7 Organization

The rest of the paper is organized as follows. Section 2 establishes convergence results. Examples and applications are given in Sect. 3, and finally Sect. 4 concludes this paper.

2 Convergence Analysis

In this section, we analyze the convergence of Algorithm 1. Throughout our analysis, we make the following assumptions.

[2] A function F on $\mathbb{R}^n$ is differentiable at point $\mathbf{x}$ if there exists a vector $\mathbf{g}$ such that $\lim_{\mathbf{h}\to 0} \frac{|F(\mathbf{x}+\mathbf{h}) - F(\mathbf{x}) - \mathbf{g}^\top \mathbf{h}|}{\|\mathbf{h}\|} = 0$

Assumption 1 F is proper and lower bounded in $\mathrm{dom}(F) \triangleq \{\mathbf{x}{:}F(\mathbf{x}) < +\infty\}$, f is continuously differentiable, and r_i is proper lower semicontinuous for all i. Problem (1) has a critical point $\mathbf{x}^*$, i.e., $\mathbf{0} \in \partial F(\mathbf{x}^*)$.

Assumption 2 Let $i = b_k$. $\nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \mathbf{x}_i)$ has Lipschitz continuity constant L_k with respect to $\mathbf{x}_i$, i.e.,

$$\left\|\nabla_{\mathbf{x}_i} f\left(\mathbf{x}_{\neq i}^{k-1}, \mathbf{u}\right) - \nabla_{\mathbf{x}_i} f\left(\mathbf{x}_{\neq i}^{k-1}, \mathbf{v}\right)\right\| \leq L_k \|\mathbf{u} - \mathbf{v}\|, \ \forall \mathbf{u}, \mathbf{v}, \tag{11}$$

and there exist constants $0 < \ell \leq L < \infty$, such that $\ell \leq L_k \leq L$ for all k.

Assumption 3 *(Essentially cyclic block update)* In Algorithm 1, within any T consecutive iterations, every block is updated at least one time.

Our analysis proceeds with several steps. We first estimate the objective decrease after every iteration (see Lemma 1) and then establish a square summable result of the iterate differences (see Proposition 1). Through the square summable result, we show a subsequence convergence result that every limit point of the iterates is a critical point (see Theorem 1). Assuming the KL property (see Definition 1) on the objective function and a monotonicity condition (see Condition 1), we establish whole sequence convergence of our algorithm and also give estimate of convergence rate (see Theorems 2 and 3).

We will show that a range of nontrivial $\omega_k > 0$ always exists to satisfy Condition 1 under a mild assumption, and thus one can backtrack ω_k to ensure $F(\mathbf{x}^k) \leq F(\mathbf{x}^{k-1})$, $\forall k$. Also, from the result below in (12), one can simply set $\omega_k = 0$ and redo the kth update if $F(\mathbf{x}^k) > F(\mathbf{x}^{k-1})$ is detected. Maintaining the monotonicity of $F(\mathbf{x}^k)$ can significantly improve the numerical performance of the algorithm, as shown in our numerical results below and also in [46,60]. Note that subsequence convergence does not require this condition.

We begin our analysis with the following lemma. The proofs of all the lemmas and propositions are given in "Appendix 1".

Lemma 1 *Take α_k and ω_k as in (9). After each iteration k, it holds*

$$F(\mathbf{x}^{k-1}) - F(\mathbf{x}^k) \geq c_1 \tilde{L}_i^j \left\|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^j\right\|^2 - c_2 \tilde{L}_i^j \left(\tilde{\omega}_i^j\right)^2 \left\|\tilde{\mathbf{x}}_i^{j-2} - \tilde{\mathbf{x}}_i^{j-1}\right\|^2 \tag{12}$$

$$\geq c_1 \tilde{L}_i^j \left\|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^j\right\|^2 - \frac{c_2 \tilde{L}_i^{j-1}}{36} \delta^2 \left\|\tilde{\mathbf{x}}_i^{j-2} - \tilde{\mathbf{x}}_i^{j-1}\right\|^2, \tag{13}$$

where $c_1 = \frac{1}{4}$, $c_2 = 9$, $i = b_k$ and $j = d_i^k$.

Remark 1 We can relax the choices of α_k and ω_k in (9). For example, we can take $\alpha_k = \frac{1}{\gamma L_k}$, $\forall k$, and $\tilde{\omega}_i^j \leq \frac{\delta(\gamma-1)}{2(\gamma+1)} \sqrt{\tilde{L}_i^{j-1} / \tilde{L}_i^j}$, $\forall i, j$ for any $\gamma > 1$ and some $\delta < 1$. Then, (12) and (13) hold with $c_1 = \frac{\gamma-1}{4}$, $c_2 = \frac{(\gamma+1)^2}{\gamma-1}$. In addition, if $0 < \inf_k \alpha_k \leq \sup_k \alpha_k < \infty$ (not necessary $\alpha_k = \frac{1}{\gamma L_k}$), (12) holds with positive c_1 and c_2, and the extrapolation weights satisfy $\tilde{\omega}_i^j \leq \delta \sqrt{(c_1 \tilde{L}_i^{j-1})/(c_2 \tilde{L}_i^j)}$, $\forall i, j$ for some $\delta < 1$, then all our convergence results below remain valid.

Note that $d_i^k = d_i^{k-1} + 1$ for $i = b_k$ and $d_i^k = d_i^{k-1}$, $\forall i \neq b_k$. Adopting the convention that $\sum_{j=p}^q a_j = 0$ when $q < p$, we can write (13) into

$$F(\mathbf{x}^{k-1}) - F(\mathbf{x}^k) \geq \sum_{i=1}^{s} \sum_{j=d_i^{k-1}+1}^{d_i^k} \frac{1}{4} \left(\tilde{L}_i^j \left\|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^j\right\|^2 - \tilde{L}_i^{j-1} \delta^2 \left\|\tilde{\mathbf{x}}_i^{j-2} - \tilde{\mathbf{x}}_i^{j-1}\right\|^2 \right), \tag{14}$$

which will be used in our subsequent convergence analysis.

Remark 2 If f is block multi-convex, i.e., it is convex with respect to each block of variables while keeping the remaining variables fixed, and r_i is convex for all i, then taking $\alpha_k = \frac{1}{L_k}$, we have (12) holds with $c_1 = \frac{1}{2}$ and $c_2 = \frac{1}{2}$; see the proof in "Appendix 1". In this case, we can take $\tilde{\omega}_i^j \leq \delta\sqrt{\tilde{L}_i^{j-1}/\tilde{L}_i^j}$, $\forall i, j$ for some $\delta < 1$, and all our convergence results can be shown through the same arguments.

2.1 Subsequence Convergence

Using Lemma 1, we can have the following result, through which we show subsequence convergence of Algorithm 1.

Proposition 1 *(Square summable) Let $\{\mathbf{x}^k\}_{k\geq 1}$ be generated from Algorithm 1 with α_k and ω_k taken from (9). We have*

$$\sum_{k=1}^{\infty} \|\mathbf{x}^{k-1} - \mathbf{x}^k\|^2 < \infty. \tag{15}$$

Theorem 1 *(Subsequence convergence) Under Assumptions 1 through 3, let $\{\mathbf{x}^k\}_{k\geq 1}$ be generated from Algorithm 1 with α_k and ω_k taken from (9). Then any limit point $\bar{\mathbf{x}}$ of $\{\mathbf{x}^k\}_{k\geq 1}$ is a critical point of (1). If the subsequence $\{\mathbf{x}^k\}_{k\in\bar{\mathcal{K}}}$ converges to $\bar{\mathbf{x}}$, then*

$$\lim_{\bar{\mathcal{K}}\ni k\to\infty} F(\mathbf{x}^k) = F(\bar{\mathbf{x}}). \tag{16}$$

Remark 3 The existence of finite limit point is guaranteed if $\{\mathbf{x}^k\}_{k\geq 1}$ is bounded, and for some applications, the boundedness of $\{\mathbf{x}^k\}_{k\geq 1}$ can be satisfied by setting appropriate parameters in Algorithm 1; see examples in Sect. 3. If r_i's are continuous, (16) immediately holds. Since we only assume lower semi-continuity of r_i's, $F(\mathbf{x})$ may not converge to $F(\bar{\mathbf{x}})$ as $\mathbf{x} \to \bar{\mathbf{x}}$, so (16) is not obvious.

Proof Assume $\bar{\mathbf{x}}$ is a limit point of $\{\mathbf{x}^k\}_{k\geq 1}$. Then there exists an index set $\mathcal{K}$ so that the subsequence $\{\mathbf{x}^k\}_{k\in\mathcal{K}}$ converging to $\bar{\mathbf{x}}$. From (15), we have $\|\mathbf{x}^{k-1} - \mathbf{x}^k\| \to 0$ and thus $\{\mathbf{x}^{k+\kappa}\}_{k\in\mathcal{K}} \to \bar{\mathbf{x}}$ for any $\kappa \geq 0$. Define

$$\mathcal{K}_i = \left\{k \in \cup_{\kappa=0}^{T-1}(\mathcal{K}+\kappa) : b_k = i\right\}, \quad i = 1, \ldots, s.$$

Take an arbitrary $i \in \{1, \ldots, s\}$. Note $\mathcal{K}_i$ is an infinite set according to Assumption 3. Taking another subsequence if necessary, L_k converges to some $\bar{L}_i$ as $\mathcal{K}_i \ni k \to \infty$. Note that since $\alpha_k = \frac{1}{2L_k}$, $\forall k$, for any $k \in \mathcal{K}_i$,

$$\mathbf{x}_i^k \in \arg\min_{\mathbf{x}_i} \left\langle \nabla_{\mathbf{x}_i} f\left(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k\right), \mathbf{x}_i - \hat{\mathbf{x}}_i^k \right\rangle + L_k \left\|\mathbf{x}_i - \hat{\mathbf{x}}_i^k\right\|^2 + r_i(\mathbf{x}_i). \tag{17}$$

Note from (15) and (6) that $\hat{\mathbf{x}}_i^k \to \bar{\mathbf{x}}_i$ as $\mathcal{K}_i \ni k \to \infty$. Since f is continuously differentiable and r_i is lower semicontinuous, letting $\mathcal{K}_i \ni k \to \infty$ in (17) yields

$$\begin{aligned} r_i(\bar{\mathbf{x}}_i) &\leq \liminf_{\mathcal{K}_i\ni k\to\infty} \left(\nabla_{\mathbf{x}_i} f\left(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k\right), \mathbf{x}_i^k - \hat{\mathbf{x}}_i^k\rangle + L_k \left\|\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k\right\|^2 + r_i\left(\mathbf{x}_i^k\right)\right) \\ &\overset{(17)}{\leq} \liminf_{\mathcal{K}_i\ni k\to\infty} \left(\nabla_{\mathbf{x}_i} f\left(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k\right), \mathbf{x}_i - \hat{\mathbf{x}}_i^k\rangle + L_k \left\|\mathbf{x}_i - \hat{\mathbf{x}}_i^k\right\|^2 + r_i(\mathbf{x}_i)\right), \\ &= \left\langle \nabla_{\mathbf{x}_i} f(\bar{\mathbf{x}}), \mathbf{x}_i - \bar{\mathbf{x}}_i \right\rangle + \bar{L}_i \|\mathbf{x}_i - \bar{\mathbf{x}}_i\|^2 + r_i(\mathbf{x}_i), \quad \forall \mathbf{x}_i \in \mathrm{dom}(F). \end{aligned}$$

Hence,

$$\bar{\mathbf{x}}_i \in \arg\min_{\mathbf{x}_i} \langle \nabla_{\mathbf{x}_i} f(\bar{\mathbf{x}}), \mathbf{x}_i - \bar{\mathbf{x}}_i \rangle + \bar{L}_i \|\mathbf{x}_i - \bar{\mathbf{x}}_i\|^2 + r_i(\mathbf{x}_i),$$

and $\bar{\mathbf{x}}_i$ satisfies the first-order optimality condition:

$$\mathbf{0} \in \nabla_{\mathbf{x}_i} f(\bar{\mathbf{x}}) + \partial r_i(\bar{\mathbf{x}}_i). \tag{18}$$

Since (18) holds for arbitrary $i \in \{1, \ldots, s\}$, $\bar{\mathbf{x}}$ is a critical point of (1).

In addition, (17) implies

$$\begin{aligned} &\left\langle \nabla_{\mathbf{x}_i} f\left(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k\right), \mathbf{x}_i^k - \hat{\mathbf{x}}_i^k \right\rangle + L_k \left\| \mathbf{x}_i^k - \hat{\mathbf{x}}_i^k \right\|^2 + r_i\left(\mathbf{x}_i^k\right) \\ &\quad \le \left\langle \nabla_{\mathbf{x}_i} f\left(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k\right), \bar{\mathbf{x}}_i - \hat{\mathbf{x}}_i^k \right\rangle + L_k \left\| \bar{\mathbf{x}}_i - \hat{\mathbf{x}}_i^k \right\|^2 + r_i(\bar{\mathbf{x}}_i). \end{aligned}$$

Taking limit superior on both sides of the above inequality over $k \in \mathcal{K}_i$ gives $\limsup_{\mathcal{K}_i \ni k \to \infty} r_i(\mathbf{x}_i^k) \le r_i(\bar{\mathbf{x}}_i)$. Since r_i is lower semi-continuous, $\liminf_{\mathcal{K}_i \ni k \to \infty} r_i(\mathbf{x}_i^k) \ge r_i(\bar{\mathbf{x}}_i)$, and thus

$$\lim_{\mathcal{K}_i \ni k \to \infty} r_i\left(\mathbf{x}_i^k\right) = r_i(\bar{\mathbf{x}}_i),\ i = 1, \ldots, s.$$

Noting that f is continuous, we complete the proof. □

2.2 Whole Sequence Convergence and Rate

In this subsection, we establish the whole sequence convergence and rate of Algorithm 1 by assuming the following monotonicity condition.

Condition 1 (Nonincreasing objective) *The weight ω_k is chosen so that $F(\mathbf{x}^k) \le F(\mathbf{x}^{k-1}),\ \forall k$.*

Remark 4 From (12), if $\omega_k = 0, \forall k$, namely, no extrapolation, then Condition 1 holds. However, extrapolation technique can often accelerate the algorithm. Although without the monotonicity, Theorem 1 can still guarantee convergence of the algorithm, numerically we notice that maintaining monotonicity of the objective can further improve the performance of the algorithm. To employ extrapolation and also maintain monotonicity, one can first do the update with a positive ω_k and check the objective, and if $F(\mathbf{x}^k) > F(\mathbf{x}^{k-1})$, then redo the kth update by using $\omega_k = 0$. For the problems that satisfy the assumptions of the next proposition, one can find $\omega_k > 0$ through backtracking to maintain the monotonicity of $F(\mathbf{x}^k)$. In general, how to choose ω_k depends on specific applications. We will test two different settings of ω_k in the numerical experiments.

The following proposition shows that under mild assumptions, Condition 1 holds for certain $\omega_k > 0$.

Proposition 2 *Let $i = b_k$. Assume $\mathbf{prox}_{\alpha_k r_i}$ is single-valued near $\mathbf{x}_i^{k-1} - \alpha_k \nabla_{\mathbf{x}_i} f(\mathbf{x}^{k-1})$ and*

$$\mathbf{x}_i^{k-1} \notin \arg\min_{\mathbf{x}_i} \langle \nabla_{\mathbf{x}_i} f(\mathbf{x}^{k-1}), \mathbf{x}_i - \mathbf{x}_i^{k-1} \rangle + \frac{1}{2\alpha_k} \left\| \mathbf{x}_i - \mathbf{x}_i^{k-1} \right\|^2 + r_i(\mathbf{x}_i), \tag{19}$$

namely, progress can still be made by updating the ith block. Then, there is $\bar{\omega}_k > 0$ such that for any $\omega_k \in [0, \bar{\omega}_k]$, we have $F(\mathbf{x}^k) \le F(\mathbf{x}^{k-1})$.

The proof of Proposition 2 involves the continuity of $\mathbf{prox}_{\alpha_k r_i}$ and is deferred to "Proof of Proposition 2" in Appendix 1.

Under Condition 1 and the KL property of F (Definition 1), we show that the sequence $\{\mathbf{x}^k\}$ converges as long as it has a finite limit point. We first establish a lemma, which has its own importance and together with the KL property implies Lemma 2.6 of [5].

The result in Lemma 2 below is very general because we need to apply it to Algorithm 1 in its general form. To ease understanding, let us go over its special cases. If $s = 1, n_{1,m} = m$ and $\beta = 0$, then (21) below with $\alpha_{1,m} = \alpha_m$ and $A_{1,m} = A_m$ reduces to $\alpha_{m+1}A_{m+1}^2 \le B_m A_m$, which together with Young's inequality gives $\sqrt{\underline{\alpha}}A_{m+1} \le \frac{\sqrt{\underline{\alpha}}}{2}A_m + \frac{1}{2\sqrt{\underline{\alpha}}}B_m$. Hence, if $\{B_m\}_{m\ge1}$ is summable, so will be $\{A_m\}_{m\ge1}$. This result can be used to analyze the prox-linear method. The more general case of $s > 1, n_{i,m} = m,\ \forall i$ and $\beta = 0$ applies to the cyclic block prox-linear method. In this case, (21) reduces to $\sum_{i=1}^s \alpha_{i,m+1}A_{i,m+1}^2 \le B_m \sum_{i=1}^s A_{i,m}$, which together with the Young's inequality implies

$$\sqrt{\underline{\alpha}}\sum_{i=1}^s A_{i,m+1} \le \sqrt{s}\sqrt{\sum_{i=1}^s \alpha_{i,m+1}A_{i,m+1}^2} \le \frac{s\tau}{4}B_m + \frac{1}{\tau}\sum_{i=1}^s A_{i,m}, \tag{20}$$

where τ is sufficiently large so that $\frac{1}{\tau} < \sqrt{\underline{\alpha}}$. Less obviously but still, if $\{B_m\}_{m\ge1}$ is summable, so will be $\{A_{i,m}\}_{m\ge1},\ \forall i$. Finally, we will need $\beta > 0$ in (21) to analyze the accelerated block prox-linear method.

Lemma 2 *For nonnegative sequences $\{A_{i,j}\}_{j\ge0}$, $\{\alpha_{i,j}\}_{j\ge0}$, $i = 1, \ldots, s$, and $\{B_m\}_{m\ge0}$, if*

$$0 < \underline{\alpha} = \inf_{i,j}\alpha_{i,j} \le \sup_{i,j}\alpha_{i,j} = \overline{\alpha} < \infty,$$

and

$$\sum_{i=1}^s \sum_{j=n_{i,m}+1}^{n_{i,m+1}} \left(\alpha_{i,j}A_{i,j}^2 - \alpha_{i,j-1}\beta^2 A_{i,j-1}^2\right) \le B_m \sum_{i=1}^s \sum_{j=n_{i,m-1}+1}^{n_{i,m}} A_{i,j},\ 0 \le m \le M, \tag{21}$$

where $0 \le \beta < 1$, and $\{n_{i,m}\}_{m\ge0}, \forall i$ are nonnegative integer sequences satisfying: $n_{i,m} \le n_{i,m+1} \le n_{i,m} + N, \forall i, m$, for some integer $N > 0$. Then we have that for $0 \le M_1 < M_2 \le M$,

$$\sum_{i=1}^s \sum_{j=n_{i,M_2}+1}^{n_{i,M_2+1}} A_{i,j} \le \frac{4sN}{\underline{\alpha}(1-\beta)^2}\sum_{m=M_1}^{M_2} B_m + \left(\sqrt{s} + \frac{4\beta\sqrt{\overline{\alpha}sN}}{(1-\beta)\sqrt{\underline{\alpha}}}\right)\sum_{i=1}^s \sum_{j=n_{i,M_1-1}+1}^{n_{i,M_1}} A_{i,j}. \tag{22}$$

In addition, if $\sum_{m=1}^\infty B_m < \infty$, $\lim_{m\to\infty} n_{i,m} = \infty, \forall i$, and (21) holds for all m, then we have

$$\sum_{j=1}^\infty A_{i,j} < \infty,\ \forall i. \tag{23}$$

The proof of this lemma is given in "Proof of Lemma 2" in Appendix 1.

Remark 5 To apply (21) to the convergence analysis of Algorithm 1, we will use $A_{i,j}$ for $\|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^j\|$ and relate $\alpha_{i,j}$ to Lipschitz constant $\tilde{L}_i^j$. The second term in the bracket of the left hand side of (21) is used to handle the extrapolation used in Algorithm 1, and we require $\beta < 1$ such that the first term can dominate the second one after summation.

We also need the following result.

Proposition 3 *Let* $\{\mathbf{x}^k\}$ *be generated from Algorithm 1. For a specific iteration* $k \geq 3T$, *assume* $\mathbf{x}^\kappa \in \mathcal{B}_\rho(\bar{\mathbf{x}})$, $\kappa = k-3T, k-3T+1, \ldots, k$ *for some* $\bar{\mathbf{x}}$ *and* $\rho > 0$. *If for each* i, $\nabla_{\mathbf{x}_i} f(\mathbf{x})$ *is Lipschitz continuous with constant* L_G *within* $B_{4\rho}(\bar{\mathbf{x}})$ *with respect to* $\mathbf{x}$, *i.e.,*

$$\|\nabla_{\mathbf{x}_i} f(\mathbf{y}) - \nabla_{\mathbf{x}_i} f(\mathbf{z})\| \leq L_G \|\mathbf{y}-\mathbf{z}\|, \ \forall \mathbf{y}, \mathbf{z} \in B_{4\rho}(\bar{\mathbf{x}}),$$

then

$$\operatorname{dist}(\mathbf{0}, \partial F(\mathbf{x}^k)) \leq \big(2(L_G+2L)+sL_G\big) \sum_{i=1}^{s} \sum_{j=d_i^{k-3T}+1}^{d_i^k} \left\| \tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^{j} \right\|. \tag{24}$$

We are now ready to present and show the whole sequence convergence of Algorithm 1.

Theorem 2 (Whole sequence convergence) *Suppose that Assumptions 1 through 3 and Condition 1 hold. Let* $\{\mathbf{x}^k\}_{k\geq 1}$ *be generated from Algorithm 1. Assume*

1. $\{\mathbf{x}^k\}_{k\geq 1}$ *has a finite limit point* $\bar{\mathbf{x}}$;
2. F *satisfies the KL property (4) around* $\bar{\mathbf{x}}$ *with parameters* ρ, η *and* θ.
3. *For each* i, $\nabla_{\mathbf{x}_i} f(\mathbf{x})$ *is Lipschitz continuous within* $B_{4\rho}(\bar{\mathbf{x}})$ *with respect to* $\mathbf{x}$.

Then

$$\lim_{k\to\infty} \mathbf{x}^k = \bar{\mathbf{x}}.$$

Remark 6 Before proving the theorem, let us remark on the conditions 1–3. The condition 1 can be guaranteed if $\{\mathbf{x}^k\}_{k\geq 1}$ has a bounded subsequence. The condition 2 is satisfied for a broad class of applications as we mentioned in Sect. 1.3. The condition 3 is a weak assumption since it requires the Lipschitz continuity only in a bounded set.

Proof From (16) and Condition 1, we have $F(\mathbf{x}^k) \to F(\bar{\mathbf{x}})$ as $k \to \infty$. We consider two cases depending on whether there is an integer K_0 such that $F(\mathbf{x}^{K_0}) = F(\bar{\mathbf{x}})$.

Case 1 Assume $F(\mathbf{x}^k) > F(\bar{\mathbf{x}})$, $\forall k$.

Since $\bar{\mathbf{x}}$ is a limit point of $\{\mathbf{x}^k\}$ and according to (15), one can choose a sufficiently large k_0 such that the points $\mathbf{x}^{k_0+\kappa}$, $\kappa = 0, 1, \ldots, 3T$ are all sufficiently close to $\bar{\mathbf{x}}$ and in $\mathcal{B}_\rho(\bar{\mathbf{x}})$, and also the differences $\|\mathbf{x}^{k_0+\kappa} - \mathbf{x}^{k_0+\kappa+1}\|$, $\kappa = 0, 1, \ldots, 3T$ are sufficiently close to *zero*. In addition, note that $F(\mathbf{x}^k) \to F(\bar{\mathbf{x}})$ as $k \to \infty$, and thus both $F(\mathbf{x}^{3(k_0+1)T}) - F(\bar{\mathbf{x}})$ and $\phi(F(\mathbf{x}^{3(k_0+1)T}) - F(\bar{\mathbf{x}}))$ can be sufficiently small. Since $\{\mathbf{x}^k\}_{k\geq 0}$ converges if and only if $\{\mathbf{x}^k\}_{k\geq k_0}$ converges, without loss of generality, we assume $k_0 = 0$, which is equivalent to setting $\mathbf{x}^{k_0}$ as a new starting point, and thus we assume

$$F(\mathbf{x}^{3T}) - F(\bar{\mathbf{x}}) < \eta, \tag{25a}$$

$$C\phi\big(F(\mathbf{x}^{3T}) - F(\bar{\mathbf{x}})\big) + C \sum_{i=1}^{s} \sum_{j=1}^{d_i^{3T}} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^{j}\| + \sum_{i=1}^{s} \|\tilde{\mathbf{x}}_i^{d_i^{3T}} - \bar{\mathbf{x}}_i\| \leq \rho, \tag{25b}$$

where

$$C = \frac{48sT\big(2(L_G+2L)+sL_G\big)}{\ell(1-\delta)^2} \geq \sqrt{s} + \frac{4\delta\sqrt{3sTL}}{(1-\delta)\sqrt{\ell}}. \tag{26}$$

Assume that $\mathbf{x}^{3mT} \in \mathcal{B}_\rho(\bar{\mathbf{x}})$ and $F(\mathbf{x}^{3mT}) < F(\bar{\mathbf{x}}) + \eta$, $m = 0, \ldots, M$ for some $M \geq 1$. Note that from (25), we can take $M = 1$. Letting $k = 3mT$ in (24) and using KL inequality (4), we have

$$\phi'(F(\mathbf{x}^{3mT}) - F(\bar{\mathbf{x}}))\left((2(L_G+2L)+sL_G)\sum_{i=1}^{s}\sum_{j=d_i^{3(m-1)T}+1}^{d_i^{3mT}}\|\tilde{\mathbf{x}}_i^{j-1}-\tilde{\mathbf{x}}_i^{j}\|\right)\geq 1, \quad (27)$$

where L_G is a uniform Lipschitz constant of $\nabla_{\mathbf{x}_i} f(\mathbf{x})$, $\forall i$ within $\mathcal{B}_{4\rho}(\bar{\mathbf{x}})$. In addition, it follows from (14) that

$$F(\mathbf{x}^{3mT}) - F(\mathbf{x}^{3(m+1)T}) \geq \sum_{i=1}^{s}\sum_{j=d_i^{3mT}+1}^{d_i^{3(m+1)T}}\left(\frac{\tilde{L}_i^{j}}{4}\|\tilde{\mathbf{x}}_i^{j-1}-\tilde{\mathbf{x}}_i^{j}\|^2 - \frac{\tilde{L}_i^{j-1}\delta^2}{4}\|\tilde{\mathbf{x}}_i^{j-2}-\tilde{\mathbf{x}}_i^{j-1}\|^2\right). \quad (28)$$

Let $\phi_m = \phi(F(\mathbf{x}^{3mT}) - F(\bar{\mathbf{x}}))$. Note that

$$\phi_m - \phi_{m+1} \geq \phi'(F(\mathbf{x}^{3mT}) - F(\bar{\mathbf{x}}))[F(\mathbf{x}^{3mT}) - F(\mathbf{x}^{3(m+1)T})].$$

Combining (27) and (28) with the above inequality and letting $\tilde{C} = 2(L_G+2L)+sL_G$ give

$$\sum_{i=1}^{s}\sum_{j=d_i^{3mT}+1}^{d_i^{3(m+1)T}}\left(\frac{\tilde{L}_i^{j}}{4}\|\tilde{\mathbf{x}}_i^{j-1}-\tilde{\mathbf{x}}_i^{j}\|^2 - \frac{\tilde{L}_i^{j-1}\delta^2}{4}\|\tilde{\mathbf{x}}_i^{j-2}-\tilde{\mathbf{x}}_i^{j-1}\|^2\right)$$
$$\leq \tilde{C}(\phi_m-\phi_{m+1})\sum_{i=1}^{s}\sum_{j=d_i^{3(m-1)T}+1}^{d_i^{3mT}}\|\tilde{\mathbf{x}}_i^{j-1}-\tilde{\mathbf{x}}_i^{j}\|. \quad (29)$$

Letting $A_{i,j} = \|\tilde{\mathbf{x}}_i^{j-1}-\tilde{\mathbf{x}}_i^{j}\|$, $\alpha_{i,j} = \tilde{L}_i^{j}/4$, $n_{i,m} = d_i^{3mT}$, $B_m = \tilde{C}(\phi_m - \phi_{m+1})$, and $\beta = \delta$ in Lemma 2, we note $d_i^{3(m+1)T} - d_i^{3mT} \leq 3T$ and have from (22) that for any intergers N and M,

$$\sum_{i=1}^{s}\sum_{j=d_i^{3NT}+1}^{d_i^{3(M+1)T}}\|\tilde{\mathbf{x}}_i^{j-1}-\tilde{\mathbf{x}}_i^{j}\| \leq C\phi_N + C\sum_{i=1}^{s}\sum_{j=d_i^{3(N-1)T}+1}^{d_i^{3NT}}\|\tilde{\mathbf{x}}_i^{j-1}-\tilde{\mathbf{x}}_i^{j}\|, \quad (30)$$

where C is given in (26). Letting $N=1$ in the above inequality, we have

$$\begin{aligned}\|\mathbf{x}^{3(M+1)T}-\bar{\mathbf{x}}\| &\leq \sum_{i=1}^{s}\|\tilde{\mathbf{x}}_i^{d_i^{3(M+1)T}}-\bar{\mathbf{x}}_i\| \\ &\leq \sum_{i=1}^{s}\left(\sum_{j=d_i^{3T}+1}^{d_i^{3(M+1)T}}\|\tilde{\mathbf{x}}_i^{j-1}-\tilde{\mathbf{x}}_i^{j}\| + \|\tilde{\mathbf{x}}_i^{d_i^{3T}}-\bar{\mathbf{x}}_i\|\right) \\ &\leq C\phi_1 + C\sum_{i=1}^{s}\sum_{j=1}^{d_i^{3T}}\|\tilde{\mathbf{x}}_i^{j-1}-\tilde{\mathbf{x}}_i^{j}\| + \sum_{i=1}^{s}\|\tilde{\mathbf{x}}_i^{d_i^{3T}}-\bar{\mathbf{x}}_i\| \overset{(25b)}{\leq} \rho.\end{aligned}$$

Hence, $\mathbf{x}^{3(M+1)T} \in \mathcal{B}_\rho(\bar{\mathbf{x}})$. In addition $F(\mathbf{x}^{3(M+1)T}) \leq F(\mathbf{x}^{3MT}) < F(\bar{\mathbf{x}}) + \eta$. By induction, $\mathbf{x}^{3mT} \in \mathcal{B}_\rho(\bar{\mathbf{x}})$, $\forall m$, and (30) holds for all M. Using Lemma 2 again, we have that $\{\tilde{\mathbf{x}}_i^{j}\}$ is a Cauchy sequence for all i and thus converges, and $\{\mathbf{x}^k\}$ also converges. Since $\bar{\mathbf{x}}$ is a limit point of $\{\mathbf{x}^k\}$, we have $\mathbf{x}^k \to \bar{\mathbf{x}}$, as $k \to \infty$.

Case 2 Assume $F(\mathbf{x}^{K_0}) = F(\bar{\mathbf{x}})$ for a certain integer K_0.

Since $F(\mathbf{x}^k)$ is nonincreasingly convergent to $F(\bar{\mathbf{x}})$, we have $F(\mathbf{x}^k) = F(\bar{\mathbf{x}})$, $\forall k \geq K_0$. Take M_0 such that $3M_0T \geq K_0$. Then $F(\mathbf{x}^{3mT}) = F(\mathbf{x}^{3(m+1)T}) = F(\bar{\mathbf{x}})$, $\forall m \geq M_0$. Summing up (28) from $m = M \geq M_0$ gives

$$0 \geq \sum_{m=M}^{\infty} \sum_{i=1}^{s} \sum_{j=d_i^{3mT}+1}^{d_i^{3(m+1)T}} \left(\frac{\tilde{L}_i^j}{4} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^j\|^2 - \frac{\tilde{L}_i^{j-1}\delta^2}{4} \|\tilde{\mathbf{x}}_i^{j-2} - \tilde{\mathbf{x}}_i^{j-1}\|^2 \right)$$

$$= \sum_{m=M}^{\infty} \sum_{i=1}^{s} \sum_{j=d_i^{3mT}+1}^{d_i^{3(m+1)T}} \frac{\tilde{L}_i^j(1-\delta^2)}{4} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^j\|^2 - \sum_{i=1}^{s} \sum_{j=d_i^{3MT}} \frac{\tilde{L}_i^j \delta^2}{4} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^j\|^2. \quad (31)$$

Let

$$a_m = \sum_{i=1}^{s} \sum_{j=d_i^{3mT}+1}^{d_i^{3(m+1)T}} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^j\|^2, \qquad S_M = \sum_{m=M}^{\infty} a_m.$$

Noting $\ell \leq \tilde{L}_i^j \leq L$, we have from (31) that $\ell(1-\delta^2)S_{M+1} \leq L\delta^2(S_M - S_{M+1})$ and thus

$$S_M \leq \gamma^{M-M_0} S_{M_0}, \ \forall M \geq M_0,$$

where $\gamma = \frac{L\delta^2}{L\delta^2 + \ell(1-\delta^2)} < 1$. By the Cauchy–Schwarz inequality and noting that a_m is the summation of at most $3T$ nonzero terms, we have

$$\sum_{i=1}^{s} \sum_{j=d_i^{3mT}+1}^{d_i^{3(m+1)T}} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^j\| \leq \sqrt{3T}\sqrt{a_m} \leq \sqrt{3T}\sqrt{S_m} \leq \sqrt{3T}\gamma^{\frac{m-M_0}{2}} S_{M_0}, \ \forall m \geq M_0. \quad (32)$$

Since $\gamma < 1$, (32) implies

$$\sum_{m=M_0}^{\infty} \sum_{i=1}^{s} \sum_{j=d_i^{3mT}+1}^{d_i^{3(m+1)T}} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^j\| \leq \frac{\sqrt{3T} S_{M_0}}{1-\sqrt{\gamma}} < \infty,$$

and thus $\mathbf{x}^k$ converges to the limit point $\bar{\mathbf{x}}$. This completes the proof. □

In addition, we can show convergence rate of Algorithm 1 through the following lemma.

Lemma 3 *For nonnegative sequence $\{A_k\}_{k=1}^{\infty}$, if $A_k \leq A_{k-1} \leq 1$, $\forall k \geq K$ for some integer K, and there are positive constants α, β and γ such that*

$$A_k \leq \alpha(A_{k-1} - A_k)^{\gamma} + \beta(A_{k-1} - A_k), \ \forall k, \quad (33)$$

we have

1. *If $\gamma \geq 1$, then $A_k \leq \left(\frac{\alpha+\beta}{1+\alpha+\beta}\right)^{k-K} A_K$, $\forall k \geq K$;*
2. *If $0 < \gamma < 1$, then $A_k \leq \nu(k-K)^{-\frac{\gamma}{1-\gamma}}$, $\forall k \geq K$, for some positive constant ν.*

Theorem 3 (Convergence rate) *Under the assumptions of Theorem 2, we have:*

1. *If $\theta \in [0, \frac{1}{2}]$, $\|\mathbf{x}^k - \bar{\mathbf{x}}\| \leq C\alpha^k$, $\forall k$, for a certain $C > 0$, $\alpha \in [0, 1)$;*

2. *If* $\theta \in (\frac{1}{2}, 1)$, $\|\mathbf{x}^k - \bar{\mathbf{x}}\| \le Ck^{-(1-\theta)/(2\theta-1)}$, $\forall k$, *for a certain* $C > 0$.

Remark 7 Before proving the theorem, let us make a few remarks on the parameter θ. First, we see that smaller θ implies faster convergence speed, and also note that the condition in (4) is stronger if θ is smaller. Secondly, it is generally not easy to determine the value of θ. For several classes of functions, [61] gives its range of possible values. The very recent work [34] presents methods to estimate its value for a function formed from other functions, for which the values of θ are known.

Proof When $\theta = 0$, then $\phi'(a) = c, \forall a$, and there must be a sufficiently large integer k_0 such that $F(\mathbf{x}^{k_0}) = F(\bar{\mathbf{x}})$, and thus $F(\mathbf{x}^k) = F(\bar{\mathbf{x}}), \forall k \ge k_0$, by noting $F(\mathbf{x}^{k-1}) \ge F(\mathbf{x}^k)$ and $\lim_{k\to\infty} F(\mathbf{x}^k) = F(\bar{\mathbf{x}})$. Otherwise $F(\mathbf{x}^k) > F(\bar{\mathbf{x}}), \forall k$. Then from the KL inequality (4), it holds that $c \cdot \mathrm{dist}(\mathbf{0}, \partial F(\mathbf{x}^k)) \ge 1$, for all $\mathbf{x}^k \in \mathcal{B}_\rho(\bar{\mathbf{x}})$, which is impossible since $\mathrm{dist}(\mathbf{0}, \partial F(\mathbf{x}^{3mT})) \to 0$ as $m \to \infty$ from (24).

For $k > k_0$, since $F(\mathbf{x}^{k-1}) = F(\mathbf{x}^k)$, and noting that in (14) all terms but one are zero under the summation over i, we have

$$\sum_{i=1}^{s} \sum_{j=d_i^{k-1}+1}^{d_i^k} \sqrt{\tilde{L}_i^{j-1}}\delta \|\tilde{\mathbf{x}}_i^{j-2} - \tilde{\mathbf{x}}_i^{j-1}\| \ge \sum_{i=1}^{s} \sum_{j=d_i^{k-1}+1}^{d_i^k} \sqrt{\tilde{L}_i^{j}} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^{j}\|.$$

Summing the above inequality over k from $m > k_0$ to ∞ and using $\ell \le \tilde{L}_i^j \le L, \forall i, j$, we have

$$\sqrt{L}\delta \sum_{i=1}^{s} \|\tilde{\mathbf{x}}_i^{d_i^{m-1}-1} - \tilde{\mathbf{x}}_i^{d_i^{m-1}}\| \ge \sqrt{\ell}(1-\delta) \sum_{i=1}^{s} \sum_{j=d_i^{m-1}+1}^{\infty} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^{j}\|, \ \forall m > k_0. \quad (34)$$

Let

$$B_m = \sum_{i=1}^{s} \sum_{j=d_i^{m-1}+1}^{\infty} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^{j}\|.$$

Then from Assumption 3, we have

$$B_{m-T} - B_m = \sum_{i=1}^{s} \sum_{j=d_i^{m-T-1}+1}^{d_i^{m-1}} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^{j}\| \ge \sum_{i=1}^{s} \|\tilde{\mathbf{x}}_i^{d_i^{m-1}-1} - \tilde{\mathbf{x}}_i^{d_i^{m-1}}\|.$$

which together with (34) gives $B_m \le \frac{\sqrt{L}\delta}{\sqrt{\ell}(1-\delta)}(B_{m-T} - B_m)$. Hence,

$$B_{mT} \le \left(\frac{\sqrt{L}\delta}{\sqrt{L}\delta + \sqrt{\ell}(1-\delta)}\right) B_{(m-1)T} \le \left(\frac{\sqrt{L}\delta}{\sqrt{L}\delta + \sqrt{\ell}(1-\delta)}\right)^{m-\ell_0} B_{\ell_0 T},$$

where $\ell_0 = \min\{\ell : \ell T \ge k_0\}$. Letting $\alpha = \left(\frac{\sqrt{L}\delta}{\sqrt{L}\delta + \sqrt{\ell}(1-\delta)}\right)^{1/T}$, we have

$$B_{mT} \le \alpha^{mT}\left(\alpha^{-\ell_0 T} B_{\ell_0 T}\right). \quad (35)$$

Note $\|\mathbf{x}^{m-1} - \bar{\mathbf{x}}\| \le B_m$. Hence, choosing a sufficiently large $C > 0$ gives the result in item 1 for $\theta = 0$.

When $0 < \theta < 1$, if for some k_0, $F(\mathbf{x}^{k_0}) = F(\bar{\mathbf{x}})$, we have (35) by the same arguments as above and thus obtain linear convergence. Below we assume $F(\mathbf{x}^k) > F(\bar{\mathbf{x}})$, $\forall k$. Let

$$A_m = \sum_{i=1}^{s} \sum_{j=d_i^{3mT}+1}^{\infty} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^{j}\|,$$

and thus

$$A_{m-1} - A_m = \sum_{i=1}^{s} \sum_{j=d_i^{3(m-1)T}+1}^{d_i^{3mT}} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^{j}\|.$$

From (27), it holds that

$$c(1-\theta)\big(F(\mathbf{x}^{3mT}) - F(\bar{\mathbf{x}})\big)^{-\theta} \geq \big((2(L_G + 2L) + sL_G)(A_{m-1} - A_m)\big)^{-1},$$

which implies

$$\phi_m = c\big(F(\mathbf{x}^{3mT}) - F(\bar{\mathbf{x}})\big)^{1-\theta} \leq c\left(c(1-\theta)(2(L_G + 2L) + sL_G)(A_{m-1} - A_m)\right)^{\frac{1-\theta}{\theta}}. \tag{36}$$

In addition, letting $N = m$ in (30), we have

$$\sum_{i=1}^{s} \sum_{j=d_i^{3mT}+1}^{d_i^{3(M+1)T}} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^{j}\| \leq C\phi_m + C \sum_{i=1}^{s} \sum_{j=d_i^{3(m-1)T}+1}^{d_i^{3mT}} \|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^{j}\|,$$

where C is the same as that in (30). Letting $M \to \infty$ in the above inequality, we have

$$\begin{aligned} A_m \leq\ & C_1\phi_m + C_1(A_{m-1} - A_m) \\ \leq\ & C_1 c\left(c(1-\theta)(2(L_G + 2L) + sL_G)(A_{m-1} - A_m)\right)^{\frac{1-\theta}{\theta}} + C_1(A_{m-1} - A_m), \end{aligned}$$

where the second inequality is from (36). Since $A_{m-1} - A_m \leq 1$ as m is sufficiently large and $\|\mathbf{x}^m - \bar{\mathbf{x}}\| \leq A_{\lfloor \frac{m}{3T} \rfloor}$, the results in item 2 for $\theta \in (0, \frac{1}{2}]$ and item 3 now immediately follow from Lemma 3. □

Before closing this section, let us make some comparison to the recent work [61]. The whole sequence convergence and rate results in this paper are the same as those in [61]. However, the results here cover more applications. We do not impose any convexity assumption on (1) while [61] requires f to be block-wise convex and every r_i to be convex. In addition, the results in [61] only apply to cyclic block prox-linear method. Empirically, a different block-update order can give better performance. As demonstrated in [16], random shuffling can often improve the efficiency of the coordinate descent method for linear support vector machine, and [59] shows that for the Tucker tensor decomposition [see (47)], updating the core tensor more frequently can be better than cyclicly updating the core tensor and factor matrices.

3 Applications and Numerical Results

In this section, we give some specific examples of (1) and show the whole sequence convergence of some existing algorithms. In addition, we demonstrate that maintaining the nonincreasing monotonicity of the objective value can improve the convergence of accelerated

gradient method and that updating variables in a random order can improve the performance of Algorithm 1 over that in the cyclic order.

3.1 FISTA with Backtracking Extrapolation

FISTA [7] is an accelerated proximal gradient method for solving composite convex problems. It is a special case[3] of Algorithm 1 with $s = 1$ and specific ω_k's. For the readers' convenience, we present the method in Algorithm 2, where both f and g are convex functions, and L_f is the Lipschitz constant of $\nabla f(\mathbf{x})$. The algorithm reaches the optimal order of convergence rate among first-order methods, but in general, it does not guarantee monotonicity of the objective values. A restarting scheme is studied in [46] that restarts FISTA from $\mathbf{x}^k$ whenever $F(\mathbf{x}^{k+1}) > F(\mathbf{x}^k)$ occurs.[4] It is demonstrated that the restarting FISTA can significantly outperform the original one. In this subsection, we show that FISTA with backtracking extrapolation weight can do even better than the restarting one.

Algorithm 2: Fast iterative shrinkage–thresholding algorithm (FISTA)

1 **Goal:** to solve convex problem $\min_{\mathbf{x}} F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$
2 **Initialization:** set $\mathbf{x}^0 = \mathbf{x}^1$, $t_1 = 1$, and $\omega_1 = 0$
3 **for** $k = 1, 2, \ldots$ **do**
4 Let $\hat{\mathbf{x}}^k = \mathbf{x}^k + \omega_k(\mathbf{x}^k - \mathbf{x}^{k-1})$
5 Update $\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \langle \nabla f(\hat{\mathbf{x}}^k), \mathbf{x} - \hat{\mathbf{x}}^k \rangle + \frac{L_f}{2}\|\mathbf{x} - \hat{\mathbf{x}}^k\|^2 + g(\mathbf{x})$
6 Set $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$ and $\omega_{k+1} = \frac{t_k - 1}{t_{k+1}}$

We test the algorithms on solving the following problem

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \lambda\|\mathbf{x}\|_1, \tag{37}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$ are given. In the test, we set $m = 1000$, $n = 3000$ and $\lambda = 1$ and generate two data sets. The first one is generated in the same way as that in [46]: first generate $\mathbf{A}$ with all its entries independently following standard normal distribution $\mathcal{N}(0, 1)$, then a sparse vector $\mathbf{x}$ with only 50 nonzero entries independently following $\mathcal{N}(0, 1)$, and finally let $\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{y}$ with the entries in $\mathbf{y}$ sampled from $\mathcal{N}(0, 0.1)$. This way ensures the optimal solution is approximately sparse. In the second data set, we have an ill-conditioned sensing matrix $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times m}$ are matrices with orthonormal columns, and $\mathbf{\Sigma}$ is a diagonal matrix with the i-th diagonal element $\sigma_i = 10^{-4} + \frac{i-1}{10}$ for $i = 1, \ldots, m$. Hence, the condition number of $\mathbf{A}$ is about 10^6. The measurement vector $\mathbf{b}$ is generated in the same way as in the first data set. We set L_f to the spectral norm of $\mathbf{A}^*\mathbf{A}$ and the initial point to *zero* vector for all three methods. For the proposed method, at each iteration, we start the extrapolation weight at that given by FISTA and do backtracking by halving it whenever the objective is detected to increase. In addition, if the backtracked weight is smaller than 10^{-2}, we simply set it to *zero*. Figure 1 plots their convergence behavior in terms of

[3] Note that from Remark 2, for convex problems, we can take larger extrapolation weight but require it to be uniformly less than *one*. Hence, although our algorithm framework includes FISTA as a special case, our whole sequence convergence result does not imply that of FISTA.

[4] Another restarting option is tested based on gradient information.

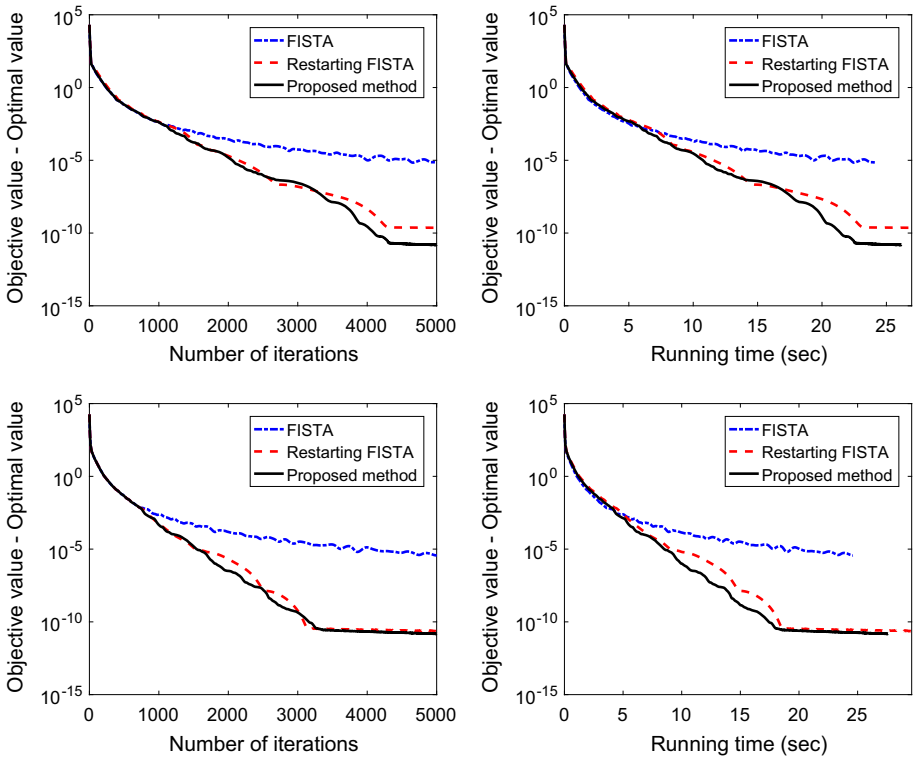

Fig. 1 Results on solving (37) by the FISTA [7], the restarting FISTA [46], and the proposed method with backtracking ω_k to ensure Condition 1. *Top row* standard Gaussian randomly generated $\mathbf{A}$; *Bottom row* ill-conditioned $\mathbf{A}$ with condition number 10^6

iteration number and also running time, where the optimal objective value is given by running the proposed method to 10,000 iterations. For both Gaussian random (well-conditioned) and ill-conditioned $\mathbf{A}$, the proposed method performs significantly better than the original FISTA. In addition, it is better than the restarting FISTA for the well-conditioned case.

3.2 Coordinate Descent Method for Nonconvex Regression

As the number of predictors is larger than sample size, variable selection becomes important to keep more important predictors and obtain a more interpretable model, and penalized regression methods are popularly used to achieve variable selection. The work [14] considers the linear regression with nonconvex penalties: the minimax concave penalty (MCP) [63] and the smoothly clipped absolute deviation (SCAD) penalty [20]. Specifically, the following model is considered

$$\min_{\boldsymbol{\beta}} \frac{1}{2n}\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|^2 + \sum_{j=1}^{p} r_{\lambda,\gamma}(\beta_j), \tag{38}$$

where $\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{X} \in \mathbb{R}^{n\times p}$ are standardized such that

$$\sum_{i=1}^{n} y_i = 0,\ \sum_{i=1}^{n} x_{ij} = 0,\ \forall j,\ \text{and}\ \frac{1}{n}\sum_{i=1}^{n} x_{ij}^2 = 1,\ \forall j, \tag{39}$$

and MCP is defined as

$$r_{\lambda,\gamma}(\theta) = \begin{cases} \lambda|\theta| - \frac{\theta^2}{2\gamma}, & \text{if } |\theta| \le \gamma\lambda, \\ \frac{1}{2}\gamma\lambda^2, & \text{if } |\theta| > \gamma\lambda, \end{cases} \tag{40}$$

and SCAD penalty is defined as

$$r_{\lambda,\gamma}(\theta) = \begin{cases} \lambda|\theta|, & \text{if } |\theta| \le \lambda, \\ \frac{2\gamma\lambda|\theta| - (\theta^2+\lambda^2)}{2(\gamma-1)}, & \text{if } \lambda < |\theta| \le \gamma\lambda, \\ \frac{\lambda^2(\gamma^2-1)}{2(\gamma-1)}, & \text{if } |\theta| > \gamma\lambda. \end{cases} \tag{41}$$

The cyclic coordinate descent method used in [14] performs the update from $j = 1$ through p

$$\beta_j^{k+1} = \arg\min_{\beta_j} \frac{1}{2n}\|\mathbf{X}(\boldsymbol{\beta}_{<j}^{k+1}, \beta_j, \boldsymbol{\beta}_{>j}^{k}) - \mathbf{y}\|^2 + r_{\lambda,\gamma}(\beta_j),$$

which can be equivalently written into the form of (2) by

$$\beta_j^{k+1} = \arg\min_{\beta_j} \frac{1}{2n}\|\mathbf{x}_j\|^2(\beta_j - \beta_j^k)^2 + \frac{1}{n}\mathbf{x}_j^\top\big(\mathbf{X}(\boldsymbol{\beta}_{<j}^{k+1}, \boldsymbol{\beta}_{\ge j}^{k}) - \mathbf{y}\big)\beta_j + r_{\lambda,\gamma}(\beta_j). \tag{42}$$

Note that the data has been standardized such that $\|\mathbf{x}_j\|^2 = n$. Hence, if $\gamma > 1$ in (40) and $\gamma > 2$ in (41), it is easy to verify that the objective in (42) is strongly convex, and there is a unique minimizer. From the convergence results of [55], it is concluded in [14] that any limit point[5] of the sequence $\{\boldsymbol{\beta}^k\}$ generated by (42) is a coordinate-wise minimizer of (38). Since $r_{\lambda,\gamma}$ in both (40) and (41) is piecewise polynomial and thus semialgebraic, it satisfies the KL property (see Definition 1). In addition, let $f(\boldsymbol{\beta})$ be the objective of (38). Then

$$f(\boldsymbol{\beta}_{<j}^{k+1}, \boldsymbol{\beta}_{\ge j}^{k}) - f(\boldsymbol{\beta}_{\le j}^{k+1}, \boldsymbol{\beta}_{>j}^{k}) \ge \frac{\mu}{2}(\beta_j^{k+1} - \beta_j^k)^2,$$

where μ is the strong convexity constant of the objective in (42). Hence, according to Theorem 2 and Remark 1, we have the following convergence result.

Theorem 4 *Assume* $\mathbf{X}$ *is standardized as in (39). Let* $\{\boldsymbol{\beta}^k\}$ *be the sequence generated from (42) or by the following update with random shuffling of coordinates*

$$\begin{aligned} \beta_{\pi_j^k}^{k+1} = \ & arg\ min_{\beta_{\pi_j^k}} \frac{1}{2n}\left\|\mathbf{x}_{\pi_j^k}\right\|^2\left(\beta_{\pi_j^k} - \beta_{\pi_j^k}^k\right)^2 \\ & + \frac{1}{n}\mathbf{x}_{\pi_j^k}^\top\left(\mathbf{X}\left(\boldsymbol{\beta}_{\pi_{<j}^k}^{k+1}, \boldsymbol{\beta}_{\pi_{\ge j}^k}^{k}\right) - \mathbf{y}\right)\beta_{\pi_j^k} + r_{\lambda,\gamma}\left(\beta_{\pi_j^k}\right), \end{aligned}$$

where $(\pi_1^k, \ldots, \pi_p^k)$ *is any permutation of* $(1, \ldots, p)$*, and* $r_{\lambda,\gamma}$ *is given by either (40) with* $\gamma > 1$ *or (41) with* $\gamma > 2$*. If* $\{\boldsymbol{\beta}^k\}$ *has a finite limit point, then* $\boldsymbol{\beta}^k$ *converges to a coordinate-wise minimizer of (38).*

3.3 Rank-One Residue Iteration for Nonnegative Matrix Factorization

The nonnegative matrix factorization can be modeled as

$$\min_{\mathbf{X},\mathbf{Y}} \|\mathbf{X}\mathbf{Y}^\top - \mathbf{M}\|_F^2, \ \text{s.t. } \mathbf{X} \in \mathbb{R}_+^{m\times p}, \ \mathbf{Y} \in \mathbb{R}_+^{n\times p}, \tag{43}$$

[5] It is stated in [14] that the sequence generated by (42) converges to a coordinate-wise minimizer of (38). However, the result is obtained directly from [55], which only guarantees subsequence convergence.

where $\mathbf{M} \in \mathbb{R}_+^{m\times n}$ is a given nonnegative matrix, $\mathbb{R}_+^{m\times p}$ denotes the set of $m \times p$ nonnegative matrices, and p is a user-specified rank. The problem in (43) can be written in the form of (1) by letting

$$f(\mathbf{X}, \mathbf{Y}) = \frac{1}{2}\|\mathbf{X}\mathbf{Y}^\top - \mathbf{M}\|_F^2, \quad r_1(\mathbf{X}) = \iota_{\mathbb{R}_+^{m\times p}}(\mathbf{X}), \quad r_2(\mathbf{Y}) = \iota_{\mathbb{R}_+^{n\times p}}(\mathbf{Y}).$$

In the literature, most existing algorithms for solving (43) update $\mathbf{X}$ and $\mathbf{Y}$ alternatingly; see the review paper [28] and the references therein. The work [25] partitions the variables in a different way: $(\mathbf{x}_1, \mathbf{y}_1, \ldots, \mathbf{x}_p, \mathbf{y}_p)$, where $\mathbf{x}_j$ denotes the jth column of $\mathbf{X}$, and proposes the rank-one residue iteration (RRI) method. It updates the variables cyclically, one column at a time. Specifically, RRI performs the updates cyclically from $i = 1$ through p,

$$\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i\geq 0} \|\mathbf{x}_i(\mathbf{y}_i^k)^\top + \mathbf{X}_{<i}^{k+1}(\mathbf{Y}_{<i}^{k+1})^\top + \mathbf{X}_{>i}^{k}(\mathbf{Y}_{>i}^{k})^\top - \mathbf{M}\|_F^2, \tag{44a}$$

$$\mathbf{y}_i^{k+1} = \arg\min_{\mathbf{y}_i\geq 0} \|\mathbf{x}_i^{k+1}(\mathbf{y}_i)^\top + \mathbf{X}_{<i}^{k+1}(\mathbf{Y}_{<i}^{k+1})^\top + \mathbf{X}_{>i}^{k}(\mathbf{Y}_{>i}^{k})^\top - \mathbf{M}\|_F^2, \tag{44b}$$

where $\mathbf{X}_{>i}^k = (\mathbf{x}_{i+1}^k, \ldots, \mathbf{x}_p^k)$. It is a cyclic block minimization method, a special case of [55]. The advantage of RRI is that each update in (44) has a closed form solution. Both updates in (44) can be written in the form of (2) by noting that they are equivalent to

$$\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i\geq 0} \frac{1}{2}\|\mathbf{y}_i^k\|^2\|\mathbf{x}_i - \mathbf{x}_i^k\|^2 + (\mathbf{y}_i^k)^\top\big(\mathbf{X}_{<i}^{k+1}(\mathbf{Y}_{<i}^{k+1})^\top + \mathbf{X}_{\geq i}^{k}(\mathbf{Y}_{\geq i}^{k})^\top - \mathbf{M}\big)^\top\mathbf{x}_i, \tag{45a}$$

$$\mathbf{y}_i^{k+1} = \arg\min_{\mathbf{y}_i\geq 0} \frac{1}{2}\|\mathbf{x}_i^{k+1}\|^2\|\mathbf{y}_i - \mathbf{y}_i^k\|^2 + \mathbf{y}_i^\top\big(\mathbf{X}_{<i}^{k+1}(\mathbf{Y}_{<i}^{k+1})^\top + \mathbf{x}_i^{k+1}(\mathbf{y}_i^k)^\top + \mathbf{X}_{>i}^{k}(\mathbf{Y}_{>i}^{k})^\top - \mathbf{M}\big)^\top\mathbf{x}_i^{k+1}. \tag{45b}$$

Since $f(\mathbf{X}, \mathbf{Y}) + r_1(\mathbf{X}) + r_2(\mathbf{Y})$ is semialgebraic and has the KL property, directly from Theorem 2, we have the following whole sequence convergence, which is stronger compared to the subsequence convergence in [25].

Theorem 5 *(Global convergence of RRI) Let* $\{(\mathbf{X}^k, \mathbf{Y}^k)\}_{k=1}^\infty$ *be the sequence generated by (44) or (45) from any starting point* $(\mathbf{X}^0, \mathbf{Y}^0)$. *If* $\{\mathbf{x}_i^k\}_{i,k}$ *and* $\{\mathbf{y}_i^k\}_{i,k}$ *are uniformly bounded and away from* zero, *then* $(\mathbf{X}^k, \mathbf{Y}^k)$ *converges to a critical point of (43).*

However, during the iterations of RRI, it may happen that some columns of $\mathbf{X}$ and $\mathbf{Y}$ become or approach to zero vector, or some of them blow up, and these cases fail the assumption of Theorem 5. To tackle with the difficulties, we modify the updates in (44) and improve the RRI method as follows.

Our first modification is to require each column of $\mathbf{X}$ to have unit Euclidean norm; the second modification is to take the Lipschitz constant of $\nabla_{\mathbf{x}_i} f(\mathbf{X}_{<i}^{k+1}, \mathbf{x}_i, \mathbf{X}_{>i}^{k}, \mathbf{Y}_{<i}^{k+1}, \mathbf{Y}_{\geq i}^{k})$ to be $L_i^k = \max(L_{\min}, \|\mathbf{y}_i^k\|^2)$ for some $L_{\min} > 0$; the third modification is that at the beginning of the kth cycle, we shuffle the blocks to a permutation $(\pi_1^k, \ldots, \pi_p^k)$. Specifically, we perform the following updates from $i = 1$ through p,

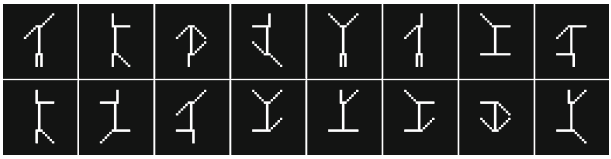

Fig. 2 Some images in the Swimmer dataset

$$\mathbf{x}_{\pi_i^k}^{k+1} = \arg\min_{\mathbf{x}_{\pi_i^k}\geq 0,\, \|\mathbf{x}_{\pi_i^k}\|=1} \frac{L_{\pi_i^k}^k}{2}\|\mathbf{x}_{\pi_i^k} - \mathbf{x}_{\pi_i^k}^k\|^2 + (\mathbf{y}_{\pi_i^k}^k)^\top \left(\mathbf{X}_{\pi_{<i}^k}^{k+1}(\mathbf{Y}_{\pi_{<i}^k}^{k+1})^\top + \mathbf{X}_{\pi_{\geq i}^k}^k(\mathbf{Y}_{\pi_{\geq i}^k}^k)^\top - \mathbf{M}\right)^\top \mathbf{x}_{\pi_i^k}, \tag{46a}$$

$$\mathbf{y}_{\pi_i^k}^{k+1} = \arg\min_{\mathbf{y}_{\pi_i^k}\geq 0} \frac{1}{2}\|\mathbf{y}_{\pi_i^k}\|^2 + \mathbf{y}_{\pi_i^k}^\top \left(\mathbf{X}_{\pi_{<i}^k}^{k+1}(\mathbf{Y}_{\pi_{<i}^k}^{k+1})^\top + \mathbf{X}_{\pi_{>i}^k}^k(\mathbf{Y}_{\pi_{>i}^k}^k)^\top - \mathbf{M}\right)^\top \mathbf{x}_{\pi_i^k}^{k+1}. \tag{46b}$$

Note that if $\pi_i^k = i$ and $L_i^k = \|\mathbf{y}_i^k\|^2$, the objective in (46a) is the same as that in (45a). Both updates in (46) have closed form solutions; see "Appendix 2". Using Theorem 2, we have the following theorem, whose proof is given in "Proof of Theorem 6" in Appendix 3. Compared to the original RRI method, the modified one automatically has bounded sequence and always has the whole sequence convergence.

Theorem 6 *(Whole iterate sequence convergence of modified RRI) Let* $\{(\mathbf{X}^k, \mathbf{Y}^k)\}_{k=1}^\infty$ *be the sequence generated by (46) from any starting point* $(\mathbf{X}^0, \mathbf{Y}^0)$. *Then* $\{\mathbf{Y}^k\}$ *is bounded, and* $(\mathbf{X}^k, \mathbf{Y}^k)$ *converges to a critical point of (43).*

3.3.1 Numerical Tests

We tested (45) and (46) on randomly generated data and also the Swimmer dataset [19]. We set $L_{\min} = 0.001$ in the tests and found that (46) with $\pi_i^k = i, \forall i, k$ produced the same final objective values as those by (45) on both random data and the Swimmer dataset. In addition, (46) with random shuffling performed almost the same as those with $\pi_i^k = i, \forall i$ (i.e., fixed cyclic order) on randomly generated data. However, random shuffling significantly improved the performance of (46) on the Swimmer dataset. There are 256 images of resolution 32×32 in the Swimmer dataset, and each image (vectorized to one column of $\mathbf{M}$) is composed of four limbs and the body. Each limb has four different positions, and all images have the body at the same position; see Fig. 2. Hence, each of these images is a nonnegative combination of 17 images: one with the body and each one of another 16 images with one limb. We set $p = 17$ in our test and ran (45) and (46) with/without random shuffling to 100 cycles. If the relative error $\|\mathbf{X}^{out}(\mathbf{Y}^{out})^\top - \mathbf{M}\|_F / \|\mathbf{M}\|_F$ is below 10^{-3}, we regard the factorization to be successful, where $(\mathbf{X}^{out}, \mathbf{Y}^{out})$ is the output. We ran the three different updates for 50 times independently, and for each run, they were fed with the same randomly generated starting point. *Both* (45) *and* (46) *without random shuffling succeed 20 times, and* (46) *with random shuffling succeeds 41 times.* Figure 3 plots all cases that occur. Every plot is in terms of running time (sec), and during that time, both methods run to 100 cycles. Since (45) and (46) without random shuffling give exactly the same results, we only show the results by

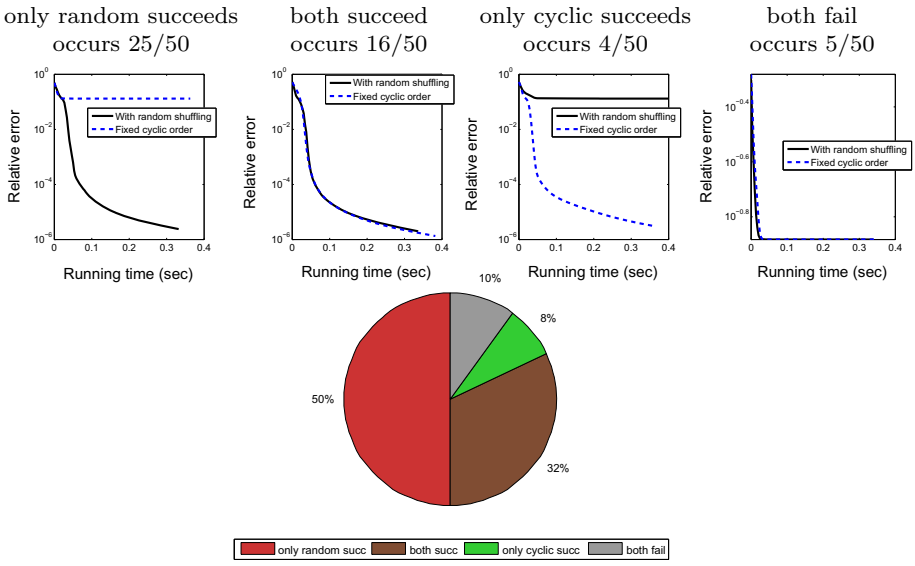

Fig. 3 All four cases of convergence behavior of the modified rank-one residue iteration (46) with fixed cyclic order and with random shuffling. Both run to 100 cycles. The first plot implies random version succeeds while the cyclic version fails and occurs 25 times among 50; the second plot implies both two versions succeed and occurs 16 times among 50; the third plot implies cyclic version succeeds while the random version fails and occurs 4 times among 50; the fourth plot implies both two versions fail and occurs 5 times among 50

(46). From the figure, we see that (46) with fixed cyclic order and with random shuffling has similar computational complexity while the latter one can more frequently avoid bad local solutions.

3.4 Block Prox-Linear Method for Nonnegative Tucker Decomposition

The nonnegative Tucker decomposition is to decompose a given nonnegative tensor (multi-dimensional array) into the product of a core nonnegative tensor and a few nonnegative factor matrices. It can be modeled as

$$\min_{\mathcal{C}\geq 0,\mathbf{A}\geq 0} \|\mathcal{C}\times_1 \mathbf{A}_1\cdots\times_N \mathbf{A}_N - \mathcal{M}\|_F^2, \tag{47}$$

where $\mathbf{A} = (\mathbf{A}_1,\ldots,\mathbf{A}_N)$ and $\mathcal{X}\times_i \mathbf{Y}$ denotes tensor-matrix multiplication along the ith mode (see [29] for example). The cyclic block proximal gradient method for solving (47) performs the following updates cyclically

$$\mathcal{C}^{k+1} = \arg\min_{\mathcal{C}\geq 0}\langle\nabla_{\mathcal{C}} f(\hat{\mathcal{C}}^k,\mathbf{A}^k),\mathcal{C}-\hat{\mathcal{C}}^k\rangle + \frac{L_c^k}{2}\|\mathcal{C}-\hat{\mathcal{C}}^k\|_F^2, \tag{48a}$$

$$\mathbf{A}_i^{k+1} = \arg\min_{\mathbf{A}_i\geq 0}\langle\nabla_{\mathbf{A}_i} f(\mathcal{C}^{k+1},\mathbf{A}_{<i}^{k+1},\hat{\mathbf{A}}_i^k,\mathbf{A}_{>i}^k),\mathbf{A}-\hat{\mathbf{A}}^k\rangle + \frac{L_i^k}{2}\|\mathbf{A}-\hat{\mathbf{A}}^k\|_F^2,\ i=1,\ldots,N. \tag{48b}$$

Here, $f(\mathcal{C},\mathbf{A}) = \frac{1}{2}\|\mathcal{C}\times_1\mathbf{A}_1\ldots\times_N\mathbf{A}_N-\mathcal{M}\|_F^2$, L_c^k and L_i^k (chosen no less than a positive $L_{\min}$) are gradient Lipschitz constants with respect to $\mathcal{C}$ and $\mathbf{A}_i$ respectively, and $\hat{\mathcal{C}}^k$ and $\hat{\mathbf{A}}_i^k$ are extrapolated points:

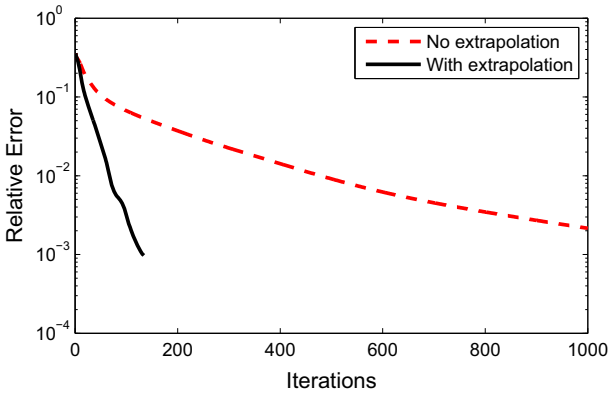

Fig. 4 Relative errors, defined as $\|\mathcal{C}^k \times_1 \mathbf{A}_1^k \cdots \times_N \mathbf{A}_N^k - \mathcal{M}\|_F / \|\mathcal{M}\|_F$, given by (48) on Gaussian randomly generated $80 \times 80 \times 80$ tensor with core size of $5 \times 5 \times 5$. No extrapolation: $\hat{\mathcal{C}}^k = \mathcal{C}^k$, $\hat{\mathbf{A}}^k = \mathbf{A}^k$, $\forall k$; With extrapolation: $\hat{\mathcal{C}}^k$, $\hat{\mathbf{A}}^k$ set as in (49) with extrapolation weights by (50)

$$\hat{\mathcal{C}}^k = \mathcal{C}^k + \omega_c^k(\mathcal{C}^k - \mathcal{C}^{k-1}),\ \hat{\mathbf{A}}_i^k = \mathbf{A}_i^k + \omega_i^k(\mathbf{A}_i^k - \mathbf{A}_i^{k-1}),\ i = 1, \ldots N. \tag{49}$$

with extrapolation weight set to

$$\omega_c^k = \min\left(\omega_k, 0.9999\sqrt{\frac{L_c^{k-1}}{L_c^k}}\right),\ \omega_i^k = \min\left(\omega_k, 0.9999\sqrt{\frac{L_i^{k-1}}{L_i^k}}\right),\ 1 \le i \le N, \tag{50}$$

where ω_k is the same as that in Algorithm 2. Our setting of extrapolated points exactly follows [59]. If after the kth iteration, the objective increases, we redo that iteration with no extrapolation. It is interesting to notice that the objective almost always decreases with the above weight and the re-update occurs less than 10 among 500 iterations. Figure 4 shows that the extrapolation technique significantly accelerates the convergence speed of the method. Note that the block-prox method with no extrapolation reduces to the block coordinate gradient method in [56].

Since the core tensor $\mathcal{C}$ interacts with all factor matrices, the work [59] proposes to update $\mathcal{C}$ more frequently to improve the performance of the block proximal gradient method. Specifically, at each cycle, it performs the following updates sequentially from $i = 1$ through N

$$\mathcal{C}^{k+1,i} = \arg\min_{\mathcal{C} \ge 0} \langle \nabla_{\mathcal{C}} f(\hat{\mathcal{C}}^{k,i}, \mathbf{A}_{<i}^{k+1}, \mathbf{A}_{\ge i}^k), \mathcal{C} - \hat{\mathcal{C}}^{k,i} \rangle + \frac{L_c^{k,i}}{2}\|\mathcal{C} - \hat{\mathcal{C}}^{k,i}\|_F^2, \tag{51a}$$

$$\mathbf{A}_i^{k+1} = \arg\min_{\mathbf{A}_i \ge 0} \langle \nabla_{\mathbf{A}_i} f(\mathcal{C}^{k+1,i}, \mathbf{A}_{<i}^{k+1}, \hat{\mathbf{A}}_i^k, \mathbf{A}_{>i}^k), \mathbf{A} - \hat{\mathbf{A}}^k \rangle + \frac{L_i^k}{2}\|\mathbf{A} - \hat{\mathbf{A}}^k\|_F^2. \tag{51b}$$

It was demonstrated that (51) numerically performs better than (48). Numerically, we observed that the performance of (51) could be further improved if the blocks of variables were randomly shuffled as in (46), namely, we performed the updates sequentially from $i = 1$ through N

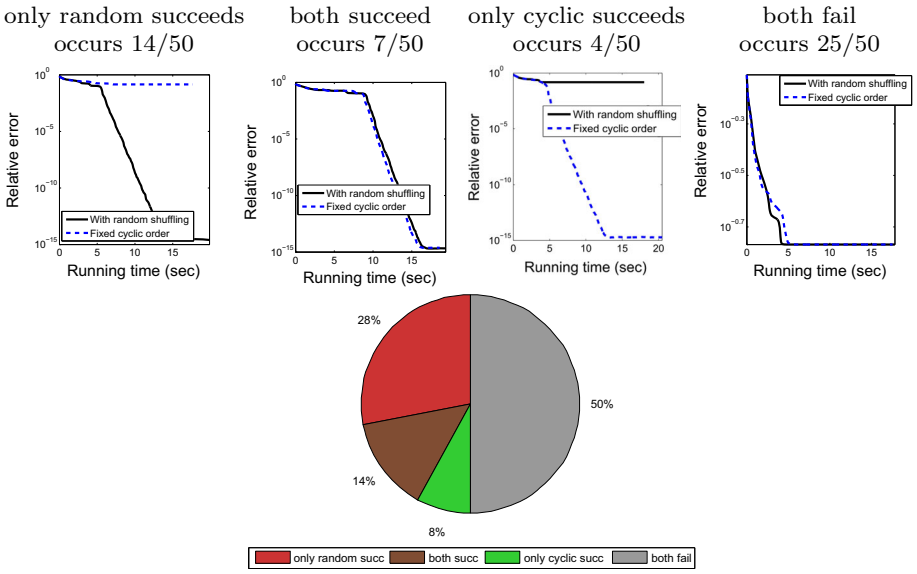

Fig. 5 All four cases of convergence behavior of the method (52) with fixed cyclic order and with random shuffling. Both run to 500 iterations. The first plot implies random version succeeds while the cyclic version fails and occurs 14 times among 50; the second plot implies both two versions succeed and occurs 7 times among 50; the third plot implies cyclic version succeeds while the random version fails and occurs 4 times among 50; the fourth plot implies both two versions fail and occurs 25 times among 50

$$\mathcal{C}^{k+1,i} = \arg\min_{\mathcal{C}\geq 0} \langle \nabla_{\mathcal{C}} f(\hat{\mathcal{C}}^{k,i}, \mathbf{A}^{k+1}_{\pi^k_{<i}}, \mathbf{A}^{k}_{\pi^k_{\geq i}}), \mathcal{C} - \hat{\mathcal{C}}^{k,i} \rangle + \frac{L^{k,i}_c}{2} \|\mathcal{C} - \hat{\mathcal{C}}^{k,i}\|_F^2, \tag{52a}$$

$$\mathbf{A}^{k+1}_{\pi^k_i} = \arg\min_{\mathbf{A}_{\pi^k_i}\geq 0} \langle \nabla_{\mathbf{A}_{\pi^k_i}} f(\mathcal{C}^{k+1,i}, \mathbf{A}^{k+1}_{\pi^k_{<i}}, \hat{\mathbf{A}}^{k}_{\pi^k_i}, \mathbf{A}^{k}_{\pi^k_{>i}}), \mathbf{A} - \hat{\mathbf{A}}^{k} \rangle + \frac{L^{k}_i}{2} \|\mathbf{A} - \hat{\mathbf{A}}^{k}\|_F^2, \tag{52b}$$

where $(\pi^k_1, \pi^k_2, \ldots, \pi^k_N)$ is a random permutation of $(1, 2, \ldots, N)$ at the k-th cycle. Note that both (48) and (52) are special cases of Algorithm 1 with $T = N + 1$ and $T = 2N + 2$ respectively. If $\{(\mathcal{C}^k, \mathbf{A}^k)\}$ is bounded, then so are L^k_c, $L^{k,i}_c$ and L^k_i's. Hence, by Theorem 2, we have the convergence result as follows.

Theorem 7 *The sequence* $\{(\mathcal{C}^k, \mathbf{A}^k)\}$ *generated from (48) or (52) is either unbounded or converges to a critical point of (47).*

We tested (51) and (52) on the $32 \times 32 \times 256$ Swimmer dataset used above and set the core size to $24 \times 17 \times 16$. We ran them to 500 cycles from the same random starting point. If the relative error $\|\mathcal{C}^{out} \times_1 \mathbf{A}^{out}_1 \ldots \times_N \mathbf{A}^{out}_N - \mathcal{M}\|_F / \|\mathcal{M}\|_F$ is below 10^{-3}, we regard the decomposition to be successful, where $(\mathcal{C}^{out}, \mathbf{A}^{out})$ is the output. *Among 50 independent runs*, (52) *with random shuffling succeeds 21 times while* (51) *succeeds only 11 times*. Figure 5 plots all cases that occur. Similar to Fig. 3, every plot is in terms of running time (s), and during that time, both methods run to 500 iterations. From the figure, we see that (52) with fixed cyclic order and with random shuffling has similar computational complexity while the latter one can more frequently avoid bad local solutions.

4 Conclusions

We have presented a block prox-linear method, in both randomized and deterministic versions, for solving nonconvex optimization problems. The method applies when the nonsmooth terms, if any, are block separable. It is easy to implement and has a small memory footprint since only one block is updated each time. Assuming that the differentiable parts have Lipschitz gradients, we showed that the method has a subsequence of iterates that converges to a critical point. Further assuming the Kurdyka–Łojasiewicz property of the objective function, we showed that the entire sequence converges to a critical point and estimated its asymptotic convergence rate. Many applications have this property. In particular, we can apply our method and its convergence results to ℓ_p-(quasi)norm ($p \in [0, +\infty]$) regularized regression problems, matrix rank minimization, orthogonality constrained optimization, semidefinite programming, and so on. Very encouraging numerical results are presented.

Acknowledgements Funding was provided in part by National Science Foundation (Grant No. DMS-1317602 and EECS-1462397) and Office of Naval Research (Grant No. N000141712162).

Appendix 1: Proofs of Key Lemmas

In this section, we give proofs of the lemmas and also propositions we used.

Proof of Lemma 1

We show the general case of $\alpha_k = \frac{1}{\gamma L_k}, \forall k$ and $\tilde{\omega}_i^j \le \frac{\delta(\gamma-1)}{2(\gamma+1)}\sqrt{\tilde{L}_i^{j-1}/\tilde{L}_i^j}$, $\forall i, j$. Assume $b_k = i$. From the Lipschitz continuity of $\nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \mathbf{x}_i)$ about $\mathbf{x}_i$, it holds that (e.g., see Lemma 2.1 in [61])

$$f(\mathbf{x}^k) \le f(\mathbf{x}^{k-1}) + \langle \nabla_{\mathbf{x}_i} f(\mathbf{x}^{k-1}), \mathbf{x}_i^k - \mathbf{x}_i^{k-1}\rangle + \frac{L_k}{2}\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\|^2. \tag{53}$$

Since $\mathbf{x}_i^k$ is the minimizer of (2), then

$$\begin{aligned}&\langle \nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k), \mathbf{x}_i^k - \hat{\mathbf{x}}_i^k\rangle + \frac{1}{2\alpha_k}\|\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k\|^2 + r_i(\mathbf{x}_i^k)\\ &\quad\le \langle \nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k), \mathbf{x}_i^{k-1} - \hat{\mathbf{x}}_i^k\rangle + \frac{1}{2\alpha_k}\|\mathbf{x}_i^{k-1} - \hat{\mathbf{x}}_i^k\|^2 + r_i(\mathbf{x}_i^{k-1}).\end{aligned} \tag{54}$$

Summing (53) and (54) and noting that $\mathbf{x}_j^{k+1} = \mathbf{x}_j^k, \forall j \neq i$, we have

$$\begin{aligned}&F(\mathbf{x}^{k-1}) - F(\mathbf{x}^k)\\ &= f(\mathbf{x}^{k-1}) + r_i(\mathbf{x}_i^{k-1}) - f(\mathbf{x}^k) - r_i(\mathbf{x}_i^k)\\ &\ge \langle \nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k) - \nabla_{\mathbf{x}_i} f(\mathbf{x}^{k-1}), \mathbf{x}_i^k - \mathbf{x}_i^{k-1}\rangle + \frac{1}{2\alpha_k}\|\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k\|^2\\ &\quad - \frac{1}{2\alpha_k}\|\mathbf{x}_i^{k-1} - \hat{\mathbf{x}}_i^k\|^2 - \frac{L_k}{2}\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\|^2\\ &= \langle \nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k) - \nabla_{\mathbf{x}_i} f(\mathbf{x}^{k-1}), \mathbf{x}_i^k - \mathbf{x}_i^{k-1}\rangle + \frac{1}{\alpha_k}\langle \mathbf{x}_i^k - \mathbf{x}_i^{k-1}, \mathbf{x}_i^{k-1} - \hat{\mathbf{x}}_i^k\rangle\\ &\quad + \left(\frac{1}{2\alpha_k} - \frac{L_k}{2}\right)\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\|^2\end{aligned}$$

$$
\begin{aligned}
&\geq -\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\| \left(\|\nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k) - \nabla_{\mathbf{x}_i} f(\mathbf{x}^{k-1})\| + \frac{1}{\alpha_k}\|\mathbf{x}_i^{k-1} - \hat{\mathbf{x}}_i^k\| \right) \\
&\quad + \left(\frac{1}{2\alpha_k} - \frac{L_k}{2}\right)\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\|^2 \\
&\geq -\left(\frac{1}{\alpha_k} + L_k\right)\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\| \cdot \|\mathbf{x}_i^{k-1} - \hat{\mathbf{x}}_i^k\| + \left(\frac{1}{2\alpha_k} - \frac{L_k}{2}\right)\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\|^2 \\
&\overset{(6)}{=} -\left(\frac{1}{\alpha_k} + L_k\right)\omega_k\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\| \cdot \|\mathbf{x}_i^{k-1} - \tilde{\mathbf{x}}_i^{d_i^{k-1}-1}\| + \left(\frac{1}{2\alpha_k} - \frac{L_k}{2}\right)\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\|^2 \\
&\geq \frac{1}{4}\left(\frac{1}{\alpha_k} - L_k\right)\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\|^2 - \frac{(1/\alpha_k + L_k)^2}{1/\alpha_k - L_k}\omega_k^2\|\mathbf{x}_i^{k-1} - \tilde{\mathbf{x}}_i^{d_i^{k-1}-1}\|^2 \\
&= \frac{(\gamma-1)L_k}{4}\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\|^2 - \frac{(\gamma+1)^2}{\gamma-1}L_k\omega_k^2\|\mathbf{x}_i^{k-1} - \tilde{\mathbf{x}}_i^{d_i^{k-1}-1}\|^2.
\end{aligned}
$$

Here, we have used Cauchy–Schwarz inequality in the second inequality, Lipschitz continuity of $\nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \mathbf{x}_i)$ in the third one, the Young's inequality in the fourth one, the fact $\mathbf{x}_i^{k-1} = \tilde{\mathbf{x}}_i^{d_i^k-1}$ to have the third equality, and $\alpha_k = \frac{1}{\gamma L_k}$ to get the last equality. Substituting $\tilde{\omega}_i^j \leq \frac{\delta(\gamma-1)}{2(\gamma+1)}\sqrt{\tilde{L}_i^{j-1}/\tilde{L}_i^j}$ and recalling (8) completes the proof.

Proof of the Claim in Remark 2

Assume $b_k = i$ and $\alpha_k = \frac{1}{L_k}$. When f is block multi-convex and r_i is convex, from Lemma 2.1 of [61], it follows that

$$
\begin{aligned}
&F(\mathbf{x}^{k-1}) - F(\mathbf{x}^k) \\
&\quad \geq \frac{L_k}{2}\|\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k\|^2 + L_k\langle \hat{\mathbf{x}}_i^k - \mathbf{x}_i^{k-1}, \mathbf{x}_i^k - \hat{\mathbf{x}}_i^k\rangle \\
&\quad \overset{(6)}{=} \frac{L_k}{2}\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1} - \omega_k(\mathbf{x}_i^{k-1} - \mathbf{x}_i^{d_i^{k-1}-1})\|^2 \\
&\quad + L_k\omega_k\left\langle \mathbf{x}_i^{k-1} - \mathbf{x}_i^{d_i^{k-1}-1}, \mathbf{x}_i^k - \mathbf{x}_i^{k-1} - \omega_k\left(\mathbf{x}_i^{k-1} - \mathbf{x}_i^{d_i^{k-1}-1}\right)\right\rangle \\
&\quad = \frac{L_k}{2}\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\|^2 - \frac{L_k\omega_k^2}{2}\|\mathbf{x}_i^{k-1} - \mathbf{x}_i^{d_i^{k-1}-1}\|^2.
\end{aligned}
$$

Hence, if $\omega_k \leq \delta\sqrt{\tilde{L}_i^{j-1}/\tilde{L}_i^j}$, we have the desired result.

Proof of Proposition 1

Summing (14) over k from 1 to K gives

$$
\begin{aligned}
F(\mathbf{x}^0) - F(\mathbf{x}^K) &\geq \sum_{i=1}^{s}\sum_{k=1}^{K}\sum_{j=d_i^{k-1}+1}^{d_i^k}\left(\frac{\tilde{L}_i^j}{4}\|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^j\|^2 - \frac{\tilde{L}_i^{j-1}\delta^2}{4}\|\tilde{\mathbf{x}}_i^{j-2} - \tilde{\mathbf{x}}_i^{j-1}\|^2\right) \\
&= \sum_{i=1}^{s}\sum_{j=1}^{d_i^K}\left(\frac{\tilde{L}_i^j}{4}\|\tilde{\mathbf{x}}_i^{j-1} - \tilde{\mathbf{x}}_i^j\|^2 - \frac{\tilde{L}_i^{j-1}\delta^2}{4}\|\tilde{\mathbf{x}}_i^{j-2} - \tilde{\mathbf{x}}_i^{j-1}\|^2\right)
\end{aligned}
$$

$$\geq \sum_{i=1}^{s}\sum_{j=1}^{d_i^K} \frac{\tilde{L}_i^j(1-\delta^2)}{4}\|\tilde{\mathbf{x}}_i^{j-1}-\tilde{\mathbf{x}}_i^j\|^2$$

$$\geq \sum_{i=1}^{s}\sum_{j=1}^{d_i^K} \frac{\ell(1-\delta^2)}{4}\|\tilde{\mathbf{x}}_i^{j-1}-\tilde{\mathbf{x}}_i^j\|^2,$$

where we have used the fact $d_i^0 = 0, \forall i$ in the first equality, $\tilde{\mathbf{x}}_i^{-1} = \tilde{\mathbf{x}}_i^0, \forall i$ to have the second inequality, and $\tilde{L}_i^j \geq \ell, \forall i, j$ in the last inequality. Letting $K \to \infty$ and noting $d_i^K \to \infty$ for all i by Assumption 3, we conclude from the above inequality and the lower boundedness of F in Assumption 1 that

$$\sum_{i=1}^{s}\sum_{j=1}^{\infty}\|\tilde{\mathbf{x}}_i^{j-1}-\tilde{\mathbf{x}}_i^j\|^2 < \infty,$$

which implies (15).

Proof of Proposition 2

From Corollary 5.20 and Example 5.23 of [52], we have that if $\mathbf{prox}_{\alpha_k r_i}$ is single valued near $\mathbf{x}_i^{k-1} - \alpha_k \nabla_{\mathbf{x}_i} f(\mathbf{x}^{k-1})$, then $\mathbf{prox}_{\alpha_k r_i}$ is continuous at $\mathbf{x}_i^{k-1} - \alpha_k \nabla_{\mathbf{x}_i} f(\mathbf{x}^{k-1})$. Let $\hat{\mathbf{x}}_i^k(\omega)$ explicitly denote the extrapolated point with weight ω, namely, we take $\hat{\mathbf{x}}_i^k(\omega_k)$ in (6). In addition, let $\mathbf{x}_i^k(\omega) = \mathbf{prox}_{\alpha_k r_i}\left(\hat{\mathbf{x}}_i^k(\omega) - \alpha_k \nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k(\omega))\right)$. Note that (14) implies

$$F(\mathbf{x}^{k-1}) - F(\mathbf{x}^k(0)) \geq \|\mathbf{x}^{k-1} - \mathbf{x}^k(0)\|^2 \overset{(19)}{>} 0. \tag{55}$$

From the optimality of $\mathbf{x}_i^k(\omega)$, it holds that

$$\langle \nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k(\omega)), \mathbf{x}_i^k(\omega) - \hat{\mathbf{x}}_i^k(\omega)\rangle + \frac{1}{2\alpha_k}\|\mathbf{x}_i^k(\omega) - \hat{\mathbf{x}}_i^k(\omega)\|^2 + r_i(\mathbf{x}_i^k(\omega))$$
$$\leq \langle \nabla_{\mathbf{x}_i} f(\mathbf{x}_{\neq i}^{k-1}, \hat{\mathbf{x}}_i^k(\omega)), \mathbf{x}_i - \hat{\mathbf{x}}_i^k(\omega)\rangle + \frac{1}{2\alpha_k}\|\mathbf{x}_i - \hat{\mathbf{x}}_i^k(\omega)\|^2 + r_i(\mathbf{x}_i),\ \forall \mathbf{x}_i.$$

Taking limit superior on both sides of the above inequality, we have

$$\langle \nabla_{\mathbf{x}_i} f(\mathbf{x}^{k-1}), \mathbf{x}_i^k(0) - \mathbf{x}_i^{k-1}\rangle + \frac{1}{2\alpha_k}\|\mathbf{x}_i^k(0) - \mathbf{x}_i^{k-1}\|^2 + \limsup_{\omega\to 0^+} r_i(\mathbf{x}_i^k(\omega))$$
$$\leq \langle \nabla_{\mathbf{x}_i} f(\mathbf{x}^{k-1}), \mathbf{x}_i - \mathbf{x}_i^{k-1}\rangle + \frac{1}{2\alpha_k}\|\mathbf{x}_i - \mathbf{x}_i^{k-1}\|^2 + r_i(\mathbf{x}_i),\ \forall \mathbf{x}_i,$$

which implies $\limsup_{\omega\to 0^+} r_i(\mathbf{x}_i^k(\omega)) \leq r_i(\mathbf{x}_i^k(0))$. Since r_i is lower semicontinuous, $\liminf_{\omega\to 0^+} r_i(\mathbf{x}_i^k(\omega)) \geq r_i(\mathbf{x}_i^k(0))$. Hence, $\lim_{\omega\to 0^+} r_i(\mathbf{x}_i^k(\omega)) = r_i(\mathbf{x}_i^k(0))$, and thus $\lim_{\omega\to 0^+} F(\mathbf{x}^k(\omega)) = F(\mathbf{x}^k(0))$. Together with (55), we conclude that there exists $\bar{\omega}_k > 0$ such that $F(\mathbf{x}^{k-1}) - F(\mathbf{x}^k(\omega)) \geq 0,\ \forall \omega \in [0, \bar{\omega}_k]$. This completes the proof.

Proof of Lemma 2

Let $\mathbf{a}_m$ and $\mathbf{u}_m$ be the vectors with their ith entries

$$(\mathbf{a}_m)_i = \sqrt{\alpha_{i,n_{i,m}}}, \quad (\mathbf{u}_m)_i = A_{i,n_{i,m}}.$$

Then (21) can be written as

$$\begin{aligned}&\|\mathbf{a}_{m+1}\odot\mathbf{u}_{m+1}\|^2+(1-\beta^2)\sum_{i=1}^{s}\sum_{j=n_{i,m}+1}^{n_{i,m+1}-1}\alpha_{i,j}A_{i,j}^2\\&\le\beta^2\|\mathbf{a}_m\odot\mathbf{u}_m\|^2+B_m\sum_{i=1}^{s}\sum_{j=n_{i,m-1}+1}^{n_{i,m}}A_{i,j}.\end{aligned}\tag{56}$$

Recall

$$\underline{\alpha}=\inf_{i,j}\alpha_{i,j},\quad\overline{\alpha}=\sup_{i,j}\alpha_{i,j}.$$

Then it follows from (56) that

$$\|\mathbf{a}_{m+1}\odot\mathbf{u}_{m+1}\|^2+\underline{\alpha}(1-\beta^2)\sum_{i=1}^{s}\sum_{j=n_{i,m}+1}^{n_{i,m+1}-1}A_{i,j}^2\le\beta^2\|\mathbf{a}_m\odot\mathbf{u}_m\|^2+B_m\sum_{i=1}^{s}\sum_{j=n_{i,m-1}+1}^{n_{i,m}}A_{i,j}.\tag{57}$$

By the Cauchy–Schwarz inequality and noting $n_{i,m+1}-n_{i,m}\le N,\forall i,m$, we have

$$\left(\sum_{i=1}^{s}\sum_{j=n_{i,m}+1}^{n_{i,m+1}-1}A_{i,j}\right)^2\le sN\sum_{i=1}^{s}\sum_{j=n_{i,m}+1}^{n_{i,m+1}-1}A_{i,j}^2\tag{58}$$

and for any positive C_1,

$$\begin{aligned}&(1+\beta)C_1\|\mathbf{a}_{m+1}\odot\mathbf{u}_{m+1}\|\left(\sum_{i=1}^{s}\sum_{j=n_{i,m}+1}^{n_{i,m+1}-1}A_{i,j}\right)\\&\le\sum_{i=1}^{s}\sum_{j=n_{i,m}+1}^{n_{i,m+1}-1}\left(\frac{4-(1+\beta)^2}{4sN}\|\mathbf{a}_{m+1}\odot\mathbf{u}_{m+1}\|^2+\frac{(1+\beta)^2C_1^2sN}{4-(1+\beta)^2}A_{i,j}^2\right)\\&\le\frac{4-(1+\beta)^2}{4}\|\mathbf{a}_{m+1}\odot\mathbf{u}_{m+1}\|^2+\frac{(1+\beta)^2C_1^2sN}{4-(1+\beta)^2}\sum_{i=1}^{s}\sum_{j=n_{i,m}+1}^{n_{i,m+1}-1}A_{i,j}^2.\end{aligned}\tag{59}$$

Taking

$$C_1\le\sqrt{\frac{\underline{\alpha}(1-\beta^2)(4-(1+\beta)^2)}{4sN}},\tag{60}$$

we have from (58) and (59) that

$$\begin{aligned}&\frac{1+\beta}{2}\|\mathbf{a}_{m+1}\odot\mathbf{u}_{m+1}\|+C_1\sum_{i=1}^{s}\sum_{j=n_{i,m}+1}^{n_{i,m+1}-1}A_{i,j}\\&\le\sqrt{\|\mathbf{a}_{m+1}\odot\mathbf{u}_{m+1}\|^2+\underline{\alpha}(1-\beta^2)\sum_{i=1}^{s}\sum_{j=n_{i,m}+1}^{n_{i,m+1}-1}A_{i,j}^2}.\end{aligned}\tag{61}$$

For any $C_2 > 0$, it holds

$$\begin{aligned}
&\sqrt{\beta^2\|\mathbf{a}_m \odot \mathbf{u}_m\|^2 + B_m \sum_{i=1}^{s} \sum_{j=n_{i,m-1}+1}^{n_{i,m}} A_{i,j}} \\
&\quad \le \beta\|\mathbf{a}_m \odot \mathbf{u}_m\| + \sqrt{B_m \sum_{i=1}^{s} \sum_{j=n_{i,m-1}+1}^{n_{i,m}} A_{i,j}} \\
&\quad \le \beta\|\mathbf{a}_m \odot \mathbf{u}_m\| + C_2 B_m + \frac{1}{4C_2} \sum_{i=1}^{s} \sum_{j=n_{i,m-1}+1}^{n_{i,m}} A_{i,j} \\
&\quad \le \beta\|\mathbf{a}_m \odot \mathbf{u}_m\| + C_2 B_m + \frac{1}{4C_2} \sum_{i=1}^{s} \sum_{j=n_{i,m-1}+1}^{n_{i,m}-1} A_{i,j} + \frac{\sqrt{s}}{4C_2}\|\mathbf{u}_m\|. \qquad (62)
\end{aligned}$$

Combining (57), (61), and (62), we have

$$\begin{aligned}
&\frac{1+\beta}{2}\|\mathbf{a}_{m+1} \odot \mathbf{u}_{m+1}\| + C_1 \sum_{i=1}^{s} \sum_{j=n_{i,m}+1}^{n_{i,m+1}-1} A_{i,j} \\
&\quad \le \beta\|\mathbf{a}_m \odot \mathbf{u}_m\| + C_2 B_m + \frac{1}{4C_2} \sum_{i=1}^{s} \sum_{j=n_{i,m-1}+1}^{n_{i,m}-1} A_{i,j} + \frac{\sqrt{s}}{4C_2}\|\mathbf{u}_m\|.
\end{aligned}$$

Summing the above inequality over m from M_1 through $M_2 \le M$ and arranging terms gives

$$\begin{aligned}
&\sum_{m=M_1}^{M_2} \left(\frac{1-\beta}{2}\|\mathbf{a}_{m+1} \odot \mathbf{u}_{m+1}\| - \frac{\sqrt{s}}{4C_2}\|\mathbf{u}_{m+1}\|\right) + \left(C_1 - \frac{1}{4C_2}\right) \sum_{m=M_1}^{M_2} \sum_{i=1}^{s} \sum_{j=n_{i,m}+1}^{n_{i,m+1}-1} A_{i,j} \\
&\quad \le \beta\|\mathbf{a}_{M_1} \odot \mathbf{u}_{M_1}\| + C_2 \sum_{m=M_1}^{M_2} B_m + \frac{1}{4C_2} \sum_{i=1}^{s} \sum_{j=n_{i,M_1-1}+1}^{n_{i,M_1}-1} A_{i,j} + \frac{\sqrt{s}}{4C_2}\|\mathbf{u}_{M_1}\| \qquad (63)
\end{aligned}$$

Take

$$C_2 = \max\left(\frac{1}{2C_1}, \frac{\sqrt{s}}{\sqrt{\underline{\alpha}}(1-\beta)}\right). \qquad (64)$$

Then (63) implies

$$\begin{aligned}
&\frac{\sqrt{\underline{\alpha}}(1-\beta)}{4} \sum_{m=M_1}^{M_2} \|\mathbf{u}_{m+1}\| + \frac{C_1}{2} \sum_{m=M_1}^{M_2} \sum_{i=1}^{s} \sum_{j=n_{i,m}+1}^{n_{i,m+1}-1} A_{i,j} \\
&\quad \le \beta\sqrt{\overline{\alpha}}\|\mathbf{u}_{M_1}\| + C_2 \sum_{m=M_1}^{M_2} B_m + \frac{1}{4C_2} \sum_{i=1}^{s} \sum_{j=n_{i,M_1-1}+1}^{n_{i,M_1}-1} A_{i,j} + \frac{\sqrt{s}}{4C_2}\|\mathbf{u}_{M_1}\|, \qquad (65)
\end{aligned}$$

which together with $\sum_{i=1}^{s} A_{i,n_{i,m+1}} \le \sqrt{s}\|\mathbf{u}_{m+1}\|$ gives

$$\begin{aligned}
&C_3 \sum_{i=1}^{s} \sum_{j=n_{i,M_1}+1}^{n_{i,M_2+1}} A_{i,j} \\
&= C_3 \sum_{m=M_1}^{M_2} \sum_{i=1}^{s} \sum_{j=n_{i,m}+1}^{n_{i,m+1}} A_{i,j} \\
&\le \beta\sqrt{\underline{\alpha}}\|\mathbf{u}_{M_1}\| + C_2 \sum_{m=M_1}^{M_2} B_m + \frac{1}{4C_2} \sum_{i=1}^{s} \sum_{j=n_{i,M_1-1}+1}^{n_{i,M_1}-1} A_{i,j} + \frac{\sqrt{s}}{4C_2}\|\mathbf{u}_{M_1}\|, \\
&\le C_2 \sum_{m=M_1}^{M_2} B_m + C_4 \sum_{i=1}^{s} \sum_{j=n_{i,M_1-1}+1}^{n_{i,M_1}} A_{i,j},
\end{aligned} \tag{66}$$

where we have used $\|\mathbf{u}_{M_1}\| \le \sum_{i=1}^{s} A_{i,n_{i,M_1}}$, and

$$C_3 = \min\left(\frac{\sqrt{\underline{\alpha}}(1-\beta)}{4\sqrt{s}}, \frac{C_1}{2}\right), \quad C_4 = \beta\sqrt{\underline{\alpha}} + \frac{\sqrt{s}}{4C_2}. \tag{67}$$

From (60), (64), and (67), we can take

$$C_1 = \frac{\sqrt{\underline{\alpha}}(1-\beta)}{2\sqrt{sN}} \le \min\left\{\sqrt{\frac{\underline{\alpha}(1-\beta^2)(4-(1+\beta)^2)}{4sN}}, \frac{\sqrt{\underline{\alpha}}(1-\beta)}{2\sqrt{s}}\right\},$$

where the inequality can be verified by noting $(1-\beta^2)(4-(1+\beta)^2)-(1-\beta)^2$ is decreasing with respect to β in $[0, 1]$. Thus from (64) and (67), we have $C_2 = \frac{1}{2C_1}$, $C_3 = \frac{C_1}{2}$, $C_4 = \beta\sqrt{\underline{\alpha}} + \frac{\sqrt{s}C_1}{2}$. Hence, from (66), we complete the proof of (22).

If $\lim_{m\to\infty} n_{i,m} = \infty, \forall i$, $\sum_{m=1}^{\infty} B_m < \infty$, and (21) holds for all m, letting $M_1 = 1$ and $M_2 \to \infty$, we have (23) from (66).

Proof of Proposition 3

For any i, assume that while updating the ith block to $\mathbf{x}_i^k$, the value of the jth block ($j \ne i$) is $\mathbf{y}_j^{(i)}$, the extrapolated point of the ith block is $\mathbf{z}_i$, and the Lipschitz constant of $\nabla_{\mathbf{x}_i} f(\mathbf{y}_{\ne i}^{(i)}, \mathbf{x}_i)$ with respect to $\mathbf{x}_i$ is $\tilde{L}_i$, namely,

$$\mathbf{x}_i^k \in \arg\min_{\mathbf{x}_i} \langle \nabla_{\mathbf{x}_i} f(\mathbf{y}_{\ne i}^{(i)}, \mathbf{z}_i), \mathbf{x}_i - \mathbf{z}_i\rangle + \tilde{L}_i\|\mathbf{x}_i - \mathbf{z}_i\|^2 + r_i(\mathbf{x}_i).$$

Hence, $\mathbf{0} \in \nabla_{\mathbf{x}_i} f(\mathbf{y}_{\ne i}^{(i)}, \mathbf{z}_i) + 2\tilde{L}_i(\mathbf{x}_i^k - \mathbf{z}_i) + \partial r_i(\mathbf{x}_i^k)$, or equivalently,

$$\nabla_{\mathbf{x}_i} f(\mathbf{x}^k) - \nabla_{\mathbf{x}_i} f(\mathbf{y}_{\ne i}^{(i)}, \mathbf{z}_i) - 2\tilde{L}_i(\mathbf{x}_i^k - \mathbf{z}_i) \in \nabla_{\mathbf{x}_i} f(\mathbf{x}^k) + \partial r_i(\mathbf{x}_i^k), \ \forall i. \tag{68}$$

Note that $\mathbf{x}_i$ may be updated to $\mathbf{x}_i^k$ not at the kth iteration but at some earlier one, which must be between $k-T$ and k by Assumption 3. In addition, for each pair (i, j), there must be some $\kappa_{i,j}$ between $k-2T$ and k such that

$$\mathbf{y}_j^{(i)} = \mathbf{x}_j^{\kappa_{i,j}}, \tag{69}$$

and for each i, there are $k-3T \le \kappa_1^i < \kappa_2^i \le k$ and extrapolation weight $\tilde{\omega}_i \le 1$ such that

$$\mathbf{z}_i = \mathbf{x}_i^{\kappa_2^i} + \tilde{\omega}_i(\mathbf{x}_i^{\kappa_2^i} - \mathbf{x}_i^{\kappa_1^i}). \tag{70}$$

By triangle inequality, $(\mathbf{y}_{\neq i}^{(i)}, \mathbf{z}_i) \in B_{4\rho}(\bar{\mathbf{x}})$ for all i. Therefore, it follows from (10) and (68) that

$$\begin{aligned}
\operatorname{dist}(\mathbf{0}, \partial F(\mathbf{x}^k)) &\overset{(68)}{\le} \sqrt{\sum_{i=1}^{s} \|\nabla_{\mathbf{x}_i} f(\mathbf{x}^k) - \nabla_{\mathbf{x}_i} f(\mathbf{y}_{\neq i}^{(i)}, \mathbf{z}_i) - 2\tilde{L}_i(\mathbf{x}_i^k - \mathbf{z}_i)\|^2} \\
&\le \sum_{i=1}^{s} \|\nabla_{\mathbf{x}_i} f(\mathbf{x}^k) - \nabla_{\mathbf{x}_i} f(\mathbf{y}_{\neq i}^{(i)}, \mathbf{z}_i) - 2\tilde{L}_i(\mathbf{x}_i^k - \mathbf{z}_i)\| \\
&\le \sum_{i=1}^{s} \left(\|\nabla_{\mathbf{x}_i} f(\mathbf{x}^k) - \nabla_{\mathbf{x}_i} f(\mathbf{y}_{\neq i}^{(i)}, \mathbf{z}_i)\| + 2\tilde{L}_i \|\mathbf{x}_i^k - \mathbf{z}_i\| \right) \\
&\le \sum_{i=1}^{s} \left(L_G \|\mathbf{x}^k - (\mathbf{y}_{\neq i}^{(i)}, \mathbf{z}_i)\| + 2\tilde{L}_i \|\mathbf{x}_i^k - \mathbf{z}_i\| \right) \\
&\le \sum_{i=1}^{s} \left((L_G + 2L)\|\mathbf{x}_i^k - \mathbf{z}_i\| + L_G \sum_{j \neq i} \|\mathbf{x}_j^k - \mathbf{y}_j^{(i)}\| \right),
\end{aligned} \tag{71}$$

where in the fourth inequality, we have used the Lipschitz continuity of $\nabla_{\mathbf{x}_i} f(\mathbf{x})$ with respect to $\mathbf{x}$, and the last inequality uses $\tilde{L}_i \le L$. Now use (71), (69), (70) and also the triangle inequality to have the desired result.

Proof of Lemma 3

The proof follows that of Theorem 2 of [3]. When $\gamma \ge 1$, since $0 \le A_{k-1} - A_k \le 1, \forall k \ge K$, we have $(A_{k-1} - A_k)^\gamma \le A_{k-1} - A_k$, and thus (33) implies that for all $k \ge K$, it holds that $A_k \le (\alpha + \beta)(A_{k-1} - A_k)$, from which item 1 immediately follows.

When $\gamma < 1$, we have $(A_{k-1} - A_k)^\gamma \ge A_{k-1} - A_k$, and thus (33) implies that for all $k \ge K$, it holds that $A_k \le (\alpha+\beta)(A_{k-1} - A_k)^\gamma$. Letting $h(x) = x^{-1/\gamma}$, we have for $k \ge K$,

$$\begin{aligned}
1 &\le (\alpha + \beta)^{1/\gamma}(A_{k-1} - A_k)A_k^{-1/\gamma} \\
&= (\alpha + \beta)^{1/\gamma} \left(\frac{A_{k-1}}{A_k} \right)^{1/\gamma} (A_{k-1} - A_k)A_{k-1}^{-1/\gamma} \\
&\le (\alpha + \beta)^{1/\gamma} \left(\frac{A_{k-1}}{A_k} \right)^{1/\gamma} \int_{A_k}^{A_{k-1}} h(x)dx \\
&= \frac{(\alpha + \beta)^{1/\gamma}}{1 - 1/\gamma} \left(\frac{A_{k-1}}{A_k} \right)^{1/\gamma} \left(A_{k-1}^{1-1/\gamma} - A_k^{1-1/\gamma} \right),
\end{aligned}$$

where we have used nonincreasing monotonicity of h in the second inequality. Hence,

$$A_k^{1-1/\gamma} - A_{k-1}^{1-1/\gamma} \ge \frac{1/\gamma - 1}{(\alpha + \beta)^{1/\gamma}} \left(\frac{A_k}{A_{k-1}} \right)^{1/\gamma}. \tag{72}$$

Let μ be the positive constant such that

$$\frac{1/\gamma - 1}{(\alpha + \beta)^{1/\gamma}} \mu = \mu^{\gamma - 1} - 1. \tag{73}$$

Note that the above equation has a unique solution $0 < \mu < 1$. We claim that

$$A_k^{1-1/\gamma} - A_{k-1}^{1-1/\gamma} \geq \mu^{\gamma-1} - 1, \ \forall k \geq K. \tag{74}$$

It obviously holds from (72) and (73) if $\left(\frac{A_k}{A_{k-1}}\right)^{1/\gamma} \geq \mu$. It also holds if $\left(\frac{A_k}{A_{k-1}}\right)^{1/\gamma} \leq \mu$ from the arguments

$$\begin{aligned}\left(\frac{A_k}{A_{k-1}}\right)^{1/\gamma} \leq \mu \Rightarrow & A_k \leq \mu^\gamma A_{k-1} \Rightarrow A_k^{1-1/\gamma} \geq \mu^{\gamma-1} A_{k-1}^{1-1/\gamma} \\ & \Rightarrow A_k^{1-1/\gamma} - A_{k-1}^{1-1/\gamma} \geq (\mu^{\gamma-1} - 1) A_{k-1}^{1-1/\gamma} \geq \mu^{\gamma-1} - 1,\end{aligned}$$

where the last inequality is from $A_{k-1}^{1-1/\gamma} \geq 1$. Hence, (74) holds, and summing it over k gives

$$A_k^{1-1/\gamma} \geq A_k^{1-1/\gamma} - A_K^{1-1/\gamma} \geq (\mu^{\gamma-1} - 1)(k - K),$$

which immediately gives item 2 by letting $\nu = (\mu^{\gamma-1} - 1)^{\frac{\gamma}{\gamma-1}}$.

Appendix 2: Solutions of (46)

In this section, we give closed form solutions to both updates in (46). First, it is not difficult to have the solution of (46b):

$$\mathbf{y}_{\pi_i}^{k+1} = \max\left(0, \left(\mathbf{X}_{\pi_{<i}}^{k+1}(\mathbf{Y}_{\pi_{<i}}^{k+1})^\top + \mathbf{X}_{\pi_{>i}}^{k}(\mathbf{Y}_{\pi_{>i}}^{k})^\top - \mathbf{M}\right)^\top \mathbf{x}_{\pi_i}^{k+1}\right).$$

Secondly, since $L_{\pi_i}^k > 0$, it is easy to write (46a) in the form of

$$\min_{\mathbf{x}\geq 0,\ \|\mathbf{x}\|=1} \frac{1}{2}\|\mathbf{x} - \mathbf{a}\|^2 + \mathbf{b}^\top \mathbf{x} + C,$$

which is apparently equivalent to

$$\max_{\mathbf{x}\geq 0,\ \|\mathbf{x}\|=1} \mathbf{c}^\top \mathbf{x}, \tag{75}$$

which $\mathbf{c} = \mathbf{a} - \mathbf{b}$. Next we give solution to (75) in three different cases.

Case 1 $\mathbf{c} < 0$. Let $i_0 = \arg\max_i c_i$ and $c_{\max} = c_{i_0} < 0$. If there are more than one components equal $c_{\max}$, one can choose an arbitrary one of them. Then the solution to (75) is given by $x_{i_0} = 1$ and $x_i = 0, \forall i \neq i_0$ because for any $\mathbf{x} \geq 0$ and $\|\mathbf{x}\| = 1$, it holds that

$$\mathbf{c}^\top \mathbf{x} \leq c_{\max}\|\mathbf{x}\|_1 \leq c_{\max}\|\mathbf{x}\| = c_{\max}.$$

Case 2 $\mathbf{c} \leq 0$ **and** $\mathbf{c} \neq 0$. Let $\mathbf{c} = (\mathbf{c}_{I_0}, \mathbf{c}_{I_-})$ where $\mathbf{c}_{I_0} = \mathbf{0}$ and $\mathbf{c}_{I_-} < 0$. Then the solution to (75) is given by $\mathbf{x}_{I_-} = \mathbf{0}$ and $\mathbf{x}_{I_0}$ being any vector that satisfies $\mathbf{x}_{I_0} \geq 0$ and $\|\mathbf{x}_{I_0}\| = 1$ because $\mathbf{c}^\top \mathbf{x} \leq 0$ for any $\mathbf{x} \geq 0$.

Case 3 $\mathbf{c} \not\leq 0$ Let $\mathbf{c} = (\mathbf{c}_{I_+}, \mathbf{c}_{I_+^c})$ where $\mathbf{c}_{I_+} > 0$ and $\mathbf{c}_{I_+^c} \leq 0$. Then (75) has a unique solution given by $\mathbf{x}_{I_+} = \frac{\mathbf{c}_{I_+}}{\|\mathbf{c}_{I_+}\|}$ and $\mathbf{x}_{I_+^c} = \mathbf{0}$ because for any $\mathbf{x} \geq 0$ and $\|\mathbf{x}\| = 1$, it holds that

$$\mathbf{c}^\top \mathbf{x} \leq \mathbf{c}_{I_+}^\top \mathbf{x}_{I_+} \leq \|\mathbf{c}_{I_+}\| \cdot \|\mathbf{x}_{I_+}\| \leq \|\mathbf{c}_{I_+}\| \cdot \|\mathbf{x}\| = \|\mathbf{c}_{I_+}\|,$$

where the second inequality holds with equality if and only if $\mathbf{x}_{I_+}$ is collinear with $\mathbf{c}_{I_+}$, and the third inequality holds with equality if and only if $\mathbf{x}_{I_+^c} = \mathbf{0}$.

Appendix 3: Proofs of Convergence of Some Examples

In this section, we give the proofs of the theorems in Sect.3.

Proof of Theorem 6

Through checking the assumptions of Theorem 2, we only need to verify the boundedness of $\{\mathbf{Y}^k\}$ to show Theorem 6. Let $\mathbf{E}^k = \mathbf{X}^k(\mathbf{Y}^k)^\top - \mathbf{M}$. Since every iteration decreases the objective, it is easy to see that $\{\mathbf{E}^k\}$ is bounded. Hence, $\{\mathbf{E}^k + \mathbf{M}\}$ is bounded, and

$$a = \sup_k \max_{i,j} (\mathbf{E}^k + \mathbf{M})_{ij} < \infty.$$

Let y_{ij}^k be the (i, j)th entry of $\mathbf{Y}^k$. Thus the columns of $\mathbf{E}^k + \mathbf{M}$ satisfy

$$a \geq \mathbf{e}_i^k + \mathbf{m}_i = \sum_{j=1}^{p} y_{ij}^k \mathbf{x}_j^k, \ \forall i, \tag{76}$$

where $\mathbf{x}_j^k$ is the jth column of $\mathbf{X}^k$. Since $\|\mathbf{x}_j^k\| = 1$, we have $\|\mathbf{x}_j^k\|_\infty \geq 1/\sqrt{m}, \ \forall j$. Note that (76) implies each component of $\sum_{j=1}^{p} y_{ij}^k \mathbf{x}_j^k$ is no greater than a. Hence from nonnegativity of $\mathbf{X}^k$ and $\mathbf{Y}^k$ and noting that at least one entry of $\mathbf{x}_j^k$ is no less than $1/\sqrt{m}$, we have $y_{ij}^k \leq a\sqrt{m}$ for all i, j and k. This completes the proof.

References

1. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**(11), 4311–4322 (2006)
2. Allen, G.: Sparse higher-order principal components analysis. In: International Conference on Artificial Intelligence and Statistics, pp. 27–36. (2012)
3. Attouch, H., Bolte, J.: On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. Math. Program. **116**(1), 5–16 (2009)
4. Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka–Lojasiewicz inequality. Math. Oper. Res. **35**(2), 438–457 (2010)
5. Attouch, H., Bolte, J., Svaiter, B.F.: Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. Math. Program. **137**(1–2), 91–129 (2013)
6. Bagirov, A.M., Jin, L., Karmitsa, N., Al Nuaimat, A., Sultanova, N.: Subgradient method for nonconvex nonsmooth optimization. J. Optim. Theory Appl. **157**(2), 416–435 (2013)
7. Beck, A., Teboulle, M.: A fast iterative shrinkage–thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
8. Beck, A., Tetruashvili, L.: On the convergence of block coordinate descent type methods. SIAM J. Optim. **23**(4), 2037–2060 (2013)
9. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmont (1999)
10. Blumensath, T., Davies, M.E.: Iterative hard thresholding for compressed sensing. Appl. Comput. Harmon. Anal. **27**(3), 265–274 (2009)
11. Bolte, J., Daniilidis, A., Lewis, A.: The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. SIAM J. Optim. **17**(4), 1205–1223 (2007)
12. 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)
13. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math. **146**(1), 459–494 (2014)
14. Breheny, P., Huang, J.: Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann. Appl. Stat. **5**(1), 232–253 (2011)

15. Burke, J.V., Lewis, A.S., Overton, M.L.: A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM J. Optim. **15**(3), 751–779 (2005)
16. Chang, K.W., Hsieh, C.J., Lin, C.J.: Coordinate descent method for large-scale l2-loss linear support vector machines. J. Mach. Learn. Res. **9**, 1369–1398 (2008)
17. Chartrand, R., Yin, W.: Iteratively reweighted algorithms for compressive sensing. In: IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008, pp. 3869–3872. IEEE (2008)
18. Chen, X.: Smoothing methods for nonsmooth, nonconvex minimization. Math. Program. **134**(1), 71–99 (2012)
19. Donoho, D., Stodden, V.: When does non-negative matrix factorization give a correct decomposition into parts. In: Advances in Neural Information Processing Systems, vol. 16. (2003)
20. Fan, J., Li, R.: Variable selection via nonconcave penalized likelihood and its oracle properties. J. Am. Stat. Assoc. **96**(456), 1348–1360 (2001)
21. Fuduli, A., Gaudioso, M., Giallombardo, G.: Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. SIAM J. Optim. **14**(3), 743–756 (2004)
22. Ghadimi, S., Lan, G.: Accelerated gradient methods for nonconvex nonlinear and stochastic programming. Math. Program. **156**(1), 59–99 (2016)
23. Grippo, L., Sciandrone, M.: Globally convergent block-coordinate techniques for unconstrained optimization. Optim. Methods Softw. **10**(4), 587–637 (1999)
24. Hildreth, C.: A quadratic programming procedure. Naval Res. Logist. Q. **4**(1), 79–85 (1957)
25. Ho, N., Van Dooren, P., Blondel, V.: Descent methods for nonnegative matrix factorization. In: Numerical Linear Algebra in Signals, Systems and Control, pp. 251–293. Springer, Netherlands (2011)
26. Hong, M., Wang, X., Razaviyayn, M., Luo, Z.Q.: Iteration complexity analysis of block coordinate descent methods. arXiv preprint arXiv:1310.6957 (2013)
27. Hoyer, P.: Non-negative matrix factorization with sparseness constraints. J. Mach. Learn. Res. **5**, 1457–1469 (2004)
28. Kim, J., He, Y., Park, H.: Algorithms for nonnegative matrix and tensor factorizations: a unified view based on block coordinate descent framework. J. Global Optim. **58**(2), 285–319 (2014)
29. Kolda, T., Bader, B.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455 (2009)
30. Kruger, A.Y.: On fréchet subdifferentials. J. Math. Sci. **116**(3), 3325–3358 (2003)
31. Kurdyka, K.: On gradients of functions definable in o-minimal structures. Ann. Inst. Fourier. **48**(3), 769–783 (1998)
32. Lai, M.J., Xu, Y., Yin, W.: Improved iteratively reweighted least squares for unconstrained smoothed ℓ_q minimization. SIAM J. Numer. Anal. **51**(2), 927–957 (2013)
33. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
34. Li, G., Pong, T.K.: Calculus of the exponent of Kurdyka–Lojasiewicz inequality and its applications to linear convergence of first-order methods. arXiv preprint arXiv:1602.02915 (2016)
35. Ling, Q., Xu, Y., Yin, W., Wen, Z.: Decentralized low-rank matrix completion. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 2925–2928. IEEE (2012)
36. Łojasiewicz, S.: Sur la géométrie semi-et sous-analytique. Ann. Inst. Fourier (Grenoble) **43**(5), 1575–1595 (1993)
37. Lu, Z., Xiao, L.: Randomized block coordinate non-monotone gradient method for a class of nonlinear programming. arXiv preprint arXiv:1306.5918 (2013)
38. Lu, Z., Xiao, L.: On the complexity analysis of randomized block-coordinate descent methods. Math. Program. **152**(1–2), 615–642 (2015)
39. Luo, Z.Q., Tseng, P.: On the convergence of the coordinate descent method for convex differentiable minimization. J. Optim. Theory Appl. **72**(1), 7–35 (1992)
40. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 689–696. ACM (2009)
41. Mohan, K., Fazel, M.: Iterative reweighted algorithms for matrix rank minimization. J. Mach. Learn. Res. **13**(1), 3441–3473 (2012)
42. Natarajan, B.K.: Sparse approximate solutions to linear systems. SIAM J. Comput. **24**(2), 227–234 (1995)
43. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J. Optim. **22**(2), 341–362 (2012)
44. Nesterov, Y.: Introductory lectures on convex optimization: a basic course, vol. 87. Springer Science & Business Media, Berlin (2013)
45. Nocedal, J., Wright, S.J.: Numerical Optimization, Springer Series in Operations Research and Financial Engineering., 2nd edn. Springer, New York (2006)

46. O'Donoghue, B., Candes, E.: Adaptive restart for accelerated gradient schemes. Found. Comput. Math. **15**(3), 715–732 (2013)
47. Paatero, P., Tapper, U.: Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics **5**(2), 111–126 (1994)
48. Peng, Z., Wu, T., Xu, Y., Yan, M., Yin, W.: Coordinate friendly structures, algorithms and applications. Ann. Math. Sci. Appl. **1**(1), 57–119 (2016)
49. Razaviyayn, M., Hong, M., Luo, Z.Q.: A unified convergence analysis of block successive minimization methods for nonsmooth optimization. SIAM J. Optim. **23**(2), 1126–1153 (2013)
50. Recht, B., Fazel, M., Parrilo, P.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev. **52**(3), 471–501 (2010)
51. Richtárik, P., Takáč, M.: Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Math. Program. **144**(1), 1–38 (2014)
52. Rockafellar, R., Wets, R.: Variational Analysis, vol. 317. Springer, Berlin (2009)
53. Saha, A., Tewari, A.: On the nonasymptotic convergence of cyclic coordinate descent methods. SIAM J. Optim. **23**(1), 576–601 (2013)
54. Shi, H.J.M., Tu, S., Xu, Y., Yin, W.: A primer on coordinate descent algorithms. arXiv preprint arXiv:1610.00040 (2016)
55. Tseng, P.: Convergence of a block coordinate descent method for nondifferentiable minimization. J. Optim. Theory Appl. **109**(3), 475–494 (2001)
56. Tseng, P., Yun, S.: A coordinate gradient descent method for nonsmooth separable minimization. Math. Program. **117**(1), 387–423 (2009)
57. Welling, M., Weber, M.: Positive tensor factorization. Pattern Recogn. Lett. **22**(12), 1255–1261 (2001)
58. 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)
59. Xu, Y.: Alternating proximal gradient method for sparse nonnegative tucker decomposition. Math. Program. Comput. **7**(1), 39–70 (2015)
60. Xu, Y., Akrotirianakis, I., Chakraborty, A.: Proximal gradient method for huberized support vector machine. Pattern Anal. Appl. **19**(4), 989–1005 (2016)
61. Xu, Y., Yin, W.: A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM J. Imaging Sci. **6**(3), 1758–1789 (2013)
62. Xu, Y., Yin, W.: A fast patch-dictionary method for whole image recovery. Inverse Probl. Imaging **10**(2), 563–583 (2016)
63. Zhang, C.H.: Nearly unbiased variable selection under minimax concave penalty. Ann. Stat. **38**(2), 894–942 (2010)