BLOCK STOCHASTIC GRADIENT ITERATION FOR CONVEX AND NONCONVEX OPTIMIZATION[*]

YANGYANG XU[†] AND WOTAO YIN[‡]

Abstract. The stochastic gradient (SG) method can quickly solve a problem with a large number of components in the objective, or a stochastic optimization problem, to a moderate accuracy. The block coordinate descent/update (BCD) method, on the other hand, can quickly solve problems with multiple (blocks of) variables. This paper introduces a method that combines the great features of SG and BCD for problems with many components in the objective and with multiple (blocks of) variables. This paper proposes a block SG (BSG) method for both convex and nonconvex programs. BSG generalizes SG by updating all the blocks of variables in the Gauss–Seidel type (updating the current block depends on the previously updated block), in either a fixed or randomly shuffled order. Although BSG has slightly more work at each iteration, it typically outperforms SG because of BSG's Gauss–Seidel updates and larger step sizes, the latter of which are determined by the smaller per-block Lipschitz constants. The convergence of BSG is established for both convex and nonconvex cases. In the convex case, BSG has the same order of convergence rate as SG. In the nonconvex case, its convergence is established in terms of the expected violation of a first-order optimality condition. In both cases our analysis is nontrivial since the typical unbiasedness assumption no longer holds. BSG is numerically evaluated on the following problems: *stochastic least squares* and *logistic regression*, which are convex, and *low-rank tensor recovery* and *bilinear logistic regression*, which are nonconvex. On the convex problems, BSG performed significantly better than SG. On the nonconvex problems, BSG significantly outperformed the deterministic BCD method because the latter tends to stagnate early near local minimizers. Overall, BSG inherits the benefits of both SG approximation and block coordinate updates and is especially useful for solving large-scale nonconvex problems.

Key words. stochastic gradient, block coordinate update, convex optimization, nonconvex optimization, nonsmooth optimization

AMS subject classifications. 49M05, 65B99, 65K10, 90C26, 90C30

DOI. 10.1137/140983938

1. Introduction. In many engineering and machine learning problems, we are facing optimization problems that involve a huge amount of data. It is often very expensive to use such a huge amount of data for every update of the problem variables, and a more efficient way is to sample a small amount from the collected data for each renewal of the variables.

Keeping this in mind, in this paper we consider the stochastic program

$$(1.1) \qquad \min_{\mathbf{x}} \Phi(\mathbf{x}) = \mathbb{E}_\xi f(\mathbf{x}; \xi) + \sum_{i=1}^{s} r_i(\mathbf{x}_i) \text{ s.t. } \mathbf{x}_i \in \mathcal{X}_i, \ i = 1, \ldots, s,$$

where $\mathcal{X}_i \subset \mathbb{R}^{n_i}$ for all i are convex constraint sets, the variable $\mathbf{x} \in \mathbb{R}^n$ is partitioned into disjoint blocks $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_s)$ of the dimension $n = \sum_{i=1}^{s} n_i$, ξ is a random variable, $\mathbb{E}_\xi f(\mathbf{x}; \xi)$ is continuously differentiable, and the r_i are regularization functions (possibly nondifferentiable) such as the ℓ_1-norm $\|\mathbf{x}_i\|_1$ or ℓ_0 seminorm $\|\mathbf{x}_i\|_0$ for

[†]Computational and Applied Mathematics, Rice University, Houston, TX 77005 (xuyang.gucas@gmail.com).

[‡]Department of Mathematics, UCLA, Los Angeles, CA 90095 (wotaoyin@math.ucla.edu).

sparse or other structured solutions. Throughout the paper, we let

$$F(\mathbf{x}) = \mathbb{E}_\xi f(\mathbf{x}; \xi), \qquad R(\mathbf{x}) = \sum_{i=1}^{s} r_i(\mathbf{x}_i)$$

and, for simplicity, we omit the subscript ξ in the expectation operator without causing confusion.

Note that by assuming $\mathrm{Prob}\{\xi = \ell\} = \frac{1}{N}, \ell = 1, \ldots, N$, (1.1) includes, as a special case, the following deterministic program

$$(1.2) \qquad \min_{\mathbf{x}} \frac{1}{N} \sum_{\ell=1}^{N} f_\ell(\mathbf{x}) + \sum_{i=1}^{s} r_i(\mathbf{x}_i) \text{ s.t. } \mathbf{x}_i \in \mathcal{X}_i \forall i,$$

where N is often very large. Many problems in applications can be written in the form of (1.1) or (1.2) such as LASSO [54], sparse logistic regression [52], bilinear logistic regression [10, 53], sparse dictionary learning [31], low-rank matrix completion [6], and so on.

We allow F and r_i to be nonconvex. When they are convex, we have sublinear convergence of the proposed method (see Algorithm 1) in terms of objective value. Without convexity, we establish global convergence in terms of the expected violation of a first-order optimality condition. In addition, numerical experiments demonstrate that our algorithm can perform very well on both convex and nonconvex problems.

1.1. Motivation. One difficulty in solving (1.1) is that it may be impossible or very expensive to accurately calculate the expectation to evaluate the objective and its gradient or subgradient. One approach is the *stochastic average approximation* (SAA) method [24], which generates a set of samples and then solves the empirical risk minimization problem by a certain optimization method.

Another approach is the *stochastic gradient* (SG) method (see [33, 40, 44] and the references therein), which assumes that an SG $\mathbf{g}(\mathbf{x}; \xi)$ of F can be obtained by a certain oracle and then iteratively performs the update

$$(1.3) \qquad \mathbf{x}^{k+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{g}(\mathbf{x}^k; \xi_k) + \tilde{\nabla} R(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{1}{2\alpha_k} \|\mathbf{x} - \mathbf{x}^k\|^2,$$

where $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_s$, and $\tilde{\nabla} R(\mathbf{x}^k)$ is a subgradient of R at $\mathbf{x}^k$. In (1.3), ξ_k is a realization of ξ at the kth iteration, and α_k is a step size that is typically required to asymptotically reduce to *zero* for convergence. The work [33] compares SAA and SG and demonstrates that the latter is competitive and sometimes significantly outperforms the former for solving a certain class of problems including the stochastic utility problem and stochastic max-flow problem. The SG method has also been popularly used (e.g., [14, 42, 50, 51, 60]) to solve deterministic programming in the form of (1.2) and exhibits advantages over the deterministic gradient method when N is large and high solution accuracy is not required.

To solve (deterministic) problems with separable nonsmooth terms as in (1.2), the block coordinate descent (BCD) method (see [20, 29, 55, 56, 57, 58] and the references therein) has been widely used. At each iteration, BCD updates only one block of variables and thus can have a much lower per-iteration complexity than methods updating all the variables together. BCD has been found efficient for solving many large-scale problems (see [7, 37, 39, 43, 57], for example).

1.2. Our algorithm. In order to take advantage of the structure of (1.1) and maintain the benefits of BCD, we generalize SG to a *block* SG (BSG) method, which is given in Algorithm 1.

ALGORITHM 1. BSG FOR SOLVING (1.1).

Input: starting point $\mathbf{x}^1$, step sizes $\{\alpha_i^k : i = 1, \ldots, s\}_{k=1}^\infty$, and positive integers $\{m_k\}_{k=1}^\infty$.

for $k = 1, 2, \ldots$ **do**

    Generate a mini batch of samples $\Xi_k = \{\xi_{k,1}, \xi_{k,2}, \ldots, \xi_{k,m_k}\}$.

    Define update order $\pi_i^k = i$, $i = 1, \ldots, s$, or randomly shuffle $\{1, 2, \ldots, s\}$ to $\{\pi_1^k, \pi_2^k, \ldots, \pi_s^k\}$.

    for $i = 1, \ldots, s$ **do**

        Compute sample gradient for the π_i^kth block

$$\tilde{\mathbf{g}}_i^k = \frac{1}{m_k} \sum_{\ell=1}^{m_k} \nabla_{\mathbf{x}_{\pi_i^k}} f(\mathbf{x}_{\pi_{<i}^k}^{k+1}, \mathbf{x}_{\pi_{\geq i}^k}^k; \xi_{k,\ell}).$$

        if $\mathcal{X}_{\pi_i^k} = \mathbb{R}^{n_{\pi_i^k}}$ *(the ith block is unconstrained)* **then**

            Update the π_i^kth block

$$(1.4) \quad \mathbf{x}_{\pi_i^k}^{k+1} = \underset{\mathbf{x}_{\pi_i^k}}{\arg\min} \langle \tilde{\mathbf{g}}_i^k, \mathbf{x}_{\pi_i^k} - \mathbf{x}_{\pi_i^k}^k \rangle + \frac{1}{2\alpha_{\pi_i^k}^k} \|\mathbf{x}_{\pi_i^k} - \mathbf{x}_{\pi_i^k}^k\|^2 + r_{\pi_i^k}(\mathbf{x}_{\pi_i^k}).$$

        else

            Update the π_i^kth block

$$(1.5) \quad \mathbf{x}_{\pi_i^k}^{k+1} = \underset{\mathbf{x}_{\pi_i^k} \in \mathcal{X}_{\pi_i^k}}{\arg\min} \langle \tilde{\mathbf{g}}_i^k + \tilde{\nabla} r_{\pi_i^k}(\mathbf{x}_{\pi_i^k}^k), \mathbf{x}_{\pi_i^k} - \mathbf{x}_{\pi_i^k}^k \rangle + \frac{1}{2\alpha_{\pi_i^k}^k} \|\mathbf{x}_{\pi_i^k} - \mathbf{x}_{\pi_i^k}^k\|^2.$$

In the algorithm, we assume that samples of ξ are randomly generated. We let $\tilde{\mathbf{g}}_i^k$ be a stochastic approximation of $\nabla_{\mathbf{x}_{\pi_i^k}} F(\mathbf{x}_{\pi_{<i}^k}^{k+1}, \mathbf{x}_{\pi_{\geq i}^k}^k)$, where $\mathbf{x}_{\pi_{<i}}$ is short for $(\mathbf{x}_{\pi_1}, \ldots, \mathbf{x}_{\pi_{i-1}})$. In (1.5), $\tilde{\nabla} r_j(\mathbf{x}_j^k)$ is a subgradient of r_j at $\mathbf{x}_j^k$, and we assume it exists for all j and k. We assume that both (1.4) and (1.5) are easy to solve. We perform two different updates. When $\mathcal{X}_i = \mathbb{R}^{n_i}$, we prefer (1.4) over (1.5) since proximal gradient iteration is typically faster than proximal subgradient iteration (see [5, 19, 36, 38] and the references therein); when $\mathcal{X}_i \neq \mathbb{R}^{n_i}$, we use (1.5), which takes the subgradient of r_i, since minimizing the nonsmooth function r_i subject to constraints is generally difficult.

One can certainly take $m_k = 1$ for all k or use larger m_k's. In general, a larger m_k leads to a lower sample variance and incurs more computation of $\tilde{\mathbf{g}}_i^k$. Note that at the beginning of each cycle, we allow a reshuffle of the blocks, which can often lead to better overall numerical performance especially for nonconvex problems, as demonstrated in [59]. For the convenience of our discussion and easy notation, we assume $\pi_i^k \equiv i$ for all i, k throughout our analysis, i.e., all iterations of the algorithm update the blocks in the same ascending order. However, our analysis still goes through if the order is shuffled at the beginning of each cycle, and some of our numerical experiments use reshuffling.

Abbreviation	Method
BSG (this paper)	block coordinate update using stochastic gradient
SG [25, 33]	stochastic gradient
BCD [29, 55]	deterministic, block coordinate (minimization) descent
BCGD [56]	deterministic, block coordinate gradient descent
SBCD [37]	stochastic block coordinate descent
SBMD [9]	stochastic block mirror descent

1.3. Related work. The BCD and SG methods are special cases of the proposed BSG method. In Algorithm 1, if $s = 1$, i.e., there is only one block of variables, the update in (1.5) becomes the SG update (1.3), and if $\pi_i^k = i$ and $\tilde{\mathbf{g}}_i^k = \nabla_{\mathbf{x}_i} F(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_{\geq i}^k)$ for all i, k, it becomes the BCD update in [56, 58]. For solving problem (1.2), a special case of (1.1), the deterministic BCD method in [56, 58] requires the partial gradients of all of the component functions for every block update while BSG uses only one or several of them, and thus the BSG update is much cheaper. On the other hand, BSG updates the variables in a Gauss–Seidel-like manner while SG does it in a Jacobi-like manner. Hence, BSG often takes fewer iterations than SG, and our numerical experiments demonstrate such an advantage of BSG over SG. For the reader's convenience, we list the related methods in Table 1.

The BCD method has a long history dating back to the 1950s [22], which considers strongly concave quadratic programming. Its original format is block coordinate minimization (BCM), which cyclically updates all the blocks by minimizing the objective with respect to one block at a time while fixing all the others at their most recent values. The convergence of BCM has been extensively analyzed in the literature for both convex and nonconvex cases; see [21, 29, 41, 55, 58], for example. The work [1] combines the proximal point method [11, 45] with BCM and proposes a proximal block coordinate update scheme. It was shown in [32] that such a scheme can perform better than the original BCM scheme for solving the tensor decomposition problem. Although BCD is straightforward to understand, there was no convergence rate known for general convex programming[1] until [37], which proposed a stochastic BCD (SBCD) method. At each iteration, the SBCD method randomly chooses one block of variables and updates it by performing one proximal gradient update. SBCD is analyzed in [37] for the smooth convex case, and the analysis is generalized to the nonsmooth convex case in [27, 43]. It is shown that SBCD has the same order of convergence rate for the nonsmooth case as it does in the smooth case. For the general convex case, SBCD has sublinear convergence in terms of expected objective value, and for the strongly convex case, it converges linearly. SBCD has also been analyzed in [28] for the nonsmooth nonconvex case. Recently, [3, 48] showed that cyclic BCD can have the same order of convergence rate as SBCD for smooth convex programming, and [23] then extended the work of [3] to the nonsmooth convex case.

The SG method also has a long history and dates back to the pioneering work [44]. Since then, the SG method became very popular in stochastic programming. The classic analysis (e.g., in [8, 47]) requires second-order differentiability and also strong convexity, and under these assumptions, the method exhibits an asymptotically optimal convergence rate $\mathcal{O}(1/k)$ in terms of expected objective value. A great improvement

---

[1]The earlier work [29] establishes the linear convergence of BCM by assuming strong convexity on the objective.

was made in [34, 40], which proposed a robust SG method applicable to general convex problems and obtained a nonasymptotic convergence rate $\mathcal{O}(1/\sqrt{k})$ by averaging all the iterates. These results were revisited in [33] which, in addition, proposed a mirror descent stochastic approximation method and achieves the same order of convergence rate with a better constant. Furthermore, the mirror descent method is accelerated in [16, 25], where the convergence results are strengthened for composite convex stochastic optimization. Recently, the SG method has been extended in [15, 17, 18] to handle nonconvex problems, for which the convergence results are established in terms of the violation of first-order optimality conditions.

A very relevant work to ours is [9], which proposes a *stochastic block mirror descent* (SBMD) method by combining SBCD with the mirror descent stochastic approximation method. The main difference between SBMD and our proposed BSG is that at each iteration SBMD randomly chooses one block of variables to update while our BSG cyclically updates all the blocks of variables, the later updated blocks depending on the early updated blocks. We will demonstrate that BSG is competitive and often performs significantly better than SBMD when using the same number of samples. The practical advantage of BSG over SBMD should be intuitive and natural. For solving (1.1), both BSG and SBMD need samples or experimental observations of ξ. When the samples arrive sequentially, the computing processor can be idle between the arrivals of two samples if they only update one block of variables and thus wastes the computing resource. Technically, SBMD has unbiased stochastic block partial gradient by choosing one block randomly to update while cyclic block update of BSG results in biased partial gradient, and thus the analysis of BSG would be more challenging than that of SBMD. To get the same order of convergence rate for BSG, we will require stronger assumptions. Specifically, the analysis of SBMD in [9] does not assume boundedness of the iterates while our analysis of BSG requires such boundedness in expectation. SBMD also appeared in [26] for solving linear systems. Although the analysis in [26] assumes that one coordinate is updated at each iteration, its implementation updates all coordinates corresponding to the nonzeros of a randomly sampled row vector, and thus it somehow explains the benefit of updating all the blocks of variables instead of just one at a time.

1.4. Contributions. We summarize our contributions as follows.

- We propose a BSG method for solving both convex and nonconvex optimization problems. The update order of the blocks of variables at each iteration is arbitrary and independent of other iterations; it can be fixed or shuffled. The new method inherits the benefits of both SG and BCD methods. It is applicable to stochastic programs in the form of (1.1), and it applies to deterministic problems in the form of (1.2) with a huge amount of training data. It applies to problems with many variables. It allows both N and n to be large.
- We analyze the BSG method for both convex and nonconvex problems. For convex problems, we show that it has the same order of convergence rate as that of the SG method, and for nonconvex problems, we establish its global convergence in terms of the expectation violation of first-order optimality conditions.
- We demonstrate applying the BSG method to two convex and two nonconvex problems with both synthetic and real-world data, and we compared it to the stochastic methods SG and SBMD, and to the deterministic method BCD. The numerical results demonstrate that it is at least comparable to, and is

often significantly better than, SG and SBMD on convex problems, and it significantly outperforms the deterministic BCD on nonconvex problems.

Notation. Throughout the paper, we restrict our points in $\mathbb{R}^n$ and use $\|\cdot\|$ for the Euclidean norm, but it is not difficult to generalize the analysis to any finite Hilbert space with a pair of primal and dual norms. We use $\mathbf{x}_{<i}$ for $(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1})$, $\nabla_{\mathbf{x}_i} F(\mathbf{x})$ for the partial gradient of F at $\mathbf{x}$ with respect to $\mathbf{x}_i$, and $\tilde{\nabla} r_i(\mathbf{x}_i)$ as a subgradient in $\partial r(\mathbf{x}_i)$, which is the limiting subdifferential (see [46]). Scalars $\alpha_i^k, \alpha_k, \ldots$ are reserved for step sizes and $L, L_{r_i}, \ldots$ for Lipschitz constants. We let Ξ_k denote the random set of samples generated at the kth iteration and $\boldsymbol{\Xi}_{[k]} = (\Xi_1, \ldots, \Xi_k)$ as the history of random sets from the 1st through kth iteration. $\mathbb{E}[X|Y]$ denotes the expectation of X conditional on Y. In addition, we partition the block set $\{1, 2, \ldots, s\}$ to $\mathcal{I}_1$ and $\mathcal{I}_2$, where

$$\mathcal{I}_1 = \{i : \mathbf{x}_i \text{ is updated by } (1.4)\} \quad \text{and} \quad \mathcal{I}_2 = \{i : \mathbf{x}_i \text{ is updated by } (1.5)\}.$$

Given any set $\mathcal{X}$, we let

$$\iota_{\mathcal{X}}(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \mathcal{X}, \\ +\infty & \text{otherwise}, \end{cases}$$

be the indicator function of $\mathcal{X}$. For any function r and any convex set $\mathcal{X}$, we let

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

be the proximal mapping, and

$$\mathcal{P}_{\mathcal{X}}(\mathbf{y}) = \arg\min_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|^2$$

be the projection onto $\mathcal{X}$. When r is convex, $\mathbf{prox}_{\alpha r}$ is nonexpansive for any $\alpha > 0$, i.e.,

$$(1.6) \qquad\qquad \|\mathbf{prox}_{\alpha r}(\mathbf{x}) - \mathbf{prox}_{\alpha r}(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\| \; \forall \mathbf{x}, \mathbf{y}.$$

Note that $\mathcal{P}_{\mathcal{X}} = \mathbf{prox}_{\iota_{\mathcal{X}}}$, and thus $\mathcal{P}_{\mathcal{X}}$ is nonexpansive for any convex set $\mathcal{X}$. Other notation will be specified when they appear.

2. Convergence analysis. In this section, we analyze the convergence of Algorithm 1 under different settings. Without loss of generality, we assume a fixed update order in Algorithm 1:

$$\pi_i^k = i \quad \forall i, k,$$

since the analysis still holds otherwise. Hence, we have

$$\tilde{\mathbf{g}}_i^k = \frac{1}{m_k} \sum_{\ell=1}^{m_k} \nabla_{\mathbf{x}_i} f(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_{\geq i}^k; \xi_{k,\ell}).$$

Define

$$\mathbf{g}_i^k = \nabla_{\mathbf{x}_i} F(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_{\geq i}^k),$$
$$\boldsymbol{\delta}_i^k = \tilde{\mathbf{g}}_i^k - \mathbf{g}_i^k.$$

Note that although $\mathbf{g}_i^k$ is defined using the full function F and does not explicitly depend on the random sample set Ξ_k, it, in fact, does depend on Ξ_k since $\mathbf{x}_j^{k+1}$ for all $j < i$, depends on Ξ_k. This is a big difference between BSG and SG and makes our analysis much more challenging. In our analysis below, some of the following assumptions will be made for each result.

ASSUMPTION 1. *There exist a constant A and a sequence $\{\sigma_k\}$ such that for any i and k,*

$$(2.1a) \qquad \left\|\mathbb{E}\left[\boldsymbol{\delta}_i^k | \boldsymbol{\Xi}_{[k-1]}\right]\right\| \leq A \cdot \max_j \alpha_j^k,$$

$$(2.1b) \qquad \mathbb{E}\|\boldsymbol{\delta}_i^k\|^2 \leq \sigma_k^2.$$

ASSUMPTION 2. *The objective function is lower bounded, i.e., $\Phi(\mathbf{x}) > -\infty$. There is a uniform Lipschitz constant $L > 0$ such that*

$$(2.2) \qquad \|\nabla_{\mathbf{x}_i} F(\mathbf{x}) - \nabla_{\mathbf{x}_i} F(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\| \quad \forall \mathbf{x}, \mathbf{y}, \ \forall i.$$

ASSUMPTION 3. *There exists a constant ρ such that $\mathbb{E}\|\mathbf{x}^k\|^2 \leq \rho^2$ for all k.*

ASSUMPTION 4. *Every function r_i is Lipschitz continuous, namely, there is a constant L_{r_i} such that*

$$\|r_i(\mathbf{x}_i) - r_i(\mathbf{y}_i)\| \leq L_{r_i}\|\mathbf{x}_i - \mathbf{y}_i\| \ \forall \mathbf{x}_i, \mathbf{y}_i.$$

We let

$$L_{\max} = \max_i L_{r_i}$$

be the dominant Lipschitz constant.

Remark 1. In Assumption 1, we assume a bounded $\mathbb{E}[\boldsymbol{\delta}_i^k | \boldsymbol{\Xi}_{k-1}]$ in (2.1a) since the common assumption $\mathbb{E}[\boldsymbol{\delta}_i^k | \boldsymbol{\Xi}_{k-1}] = \mathbf{0}$ fails to hold in our algorithm. Since the block updates in Algorithm 1 are Gauss–Seidel, the gradient error $\boldsymbol{\delta}_i^k$ typically has a nonlinear dependence on $\mathbf{x}_j^{k+1}$ for all $j < i$, which depends on Ξ_k. On the other hand, the boundedness assumption (2.1a) holds under proper conditions. For example, in (1.1), let $f(\mathbf{x}; \xi) = f_\xi(\mathbf{x})$ with $\text{Prob}\{\xi = \ell\} = \frac{1}{N}, \ell = 1, \ldots, N$. Then $F(\mathbf{x}) = \frac{1}{N}\sum_{\ell=1}^N f_\ell(\mathbf{x})$. Assume that r_i's are convex and $\mathcal{X}_i = \mathbb{R}^{n_i}$ for all i, and that f_ℓ has Lipschitz continuous partial gradient with a uniform constant L, namely,

$$\|\nabla_{\mathbf{x}_i} f_\ell(\mathbf{x}) - \nabla_{\mathbf{x}_i} f_\ell(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\| \quad \forall i, \ell, \forall \mathbf{x}, \mathbf{y}.$$

In addition, assume that each Ξ_k is a singleton, i.e., $\Xi_k = \{\xi_k\}$ with ξ_k uniformly selected from $\{1, \ldots, N\}$. Then

$$\mathbb{E}[\tilde{\mathbf{g}}_i^k | \boldsymbol{\Xi}_{[k-1]}] = \sum_{\ell=1}^N \text{Prob}\{\xi_k = \ell\} \nabla_{\mathbf{x}_i} f_\ell(\mathbf{y}_{<i}^{\ell,k+1}, \mathbf{x}_{\geq i}^k) = \frac{1}{N}\sum_{\ell=1}^N \nabla_{\mathbf{x}_i} f_\ell(\mathbf{y}_{<i}^{\ell,k+1}, \mathbf{x}_{\geq i}^k),$$

where $\mathbf{y}_1^{\ell,k+1} = \mathbf{prox}_{\alpha_1^k r_1}\big(\mathbf{x}_1^k - \alpha_1^k \nabla_{\mathbf{x}_1} f_\ell(\mathbf{x}^k)\big)$ and $\mathbf{y}_p^{\ell,k+1} = \mathbf{prox}_{\alpha_p^k r_p}\big(\mathbf{x}_p^k - \alpha_p^k \nabla_{\mathbf{x}_p} f_\ell$
$(\mathbf{y}_{<p}^{\ell,k+1}, \mathbf{x}_{\geq p}^k)\big)$ for $p \geq 2$, and

$$
\begin{aligned}
\mathbb{E}[\mathbf{g}_i^k | \mathbf{\Xi}_{[k-1]}] &= \sum_{m=1}^N \mathrm{Prob}\{\xi_k = m\} \nabla_{\mathbf{x}_i} F(\mathbf{y}_{<i}^{m,k+1}, \mathbf{x}_{\geq i}^k) && \text{(note } \mathbf{g}_i^k \text{ depends on } \xi_k) \\
&= \frac{1}{N^2} \sum_{m=1}^N \sum_{\ell=1}^N \nabla_{\mathbf{x}_i} f_\ell(\mathbf{y}_{<i}^{m,k+1}, \mathbf{x}_{\geq i}^k) && \text{(substitute } F) \\
&= \frac{1}{N^2} \sum_{\ell=1}^N \sum_{m=1}^N \nabla_{\mathbf{x}_i} f_\ell(\mathbf{y}_{<i}^{m,k+1}, \mathbf{x}_{\geq i}^k).
\end{aligned}
$$

Combining the above formulas for $\mathbb{E}[\tilde{\mathbf{g}}_i^k | \mathbf{\Xi}_{[k-1]}]$ and $\mathbb{E}[\mathbf{g}_i^k | \mathbf{\Xi}_{[k-1]}]$ gives

$$
\begin{aligned}
\big\| \mathbb{E}[\boldsymbol{\delta}_i^k | \mathbf{\Xi}_{[k-1]}] \big\| &= \big\| \mathbb{E}[\tilde{\mathbf{g}}_i^k - \mathbf{g}_i^k | \mathbf{\Xi}_{[k-1]}] \big\| \\
&\leq \frac{1}{N^2} \sum_{\ell=1}^N \sum_{m=1}^N \big\| \nabla_{\mathbf{x}_i} f_\ell(\mathbf{y}_{<i}^{\ell,k+1}, \mathbf{x}_{\geq i}^k) - \nabla_{\mathbf{x}_i} f_\ell(\mathbf{y}_{<i}^{m,k+1}, \mathbf{x}_{\geq i}^k) \big\| \\
&\leq \frac{L}{N^2} \sum_{\ell=1}^N \sum_{m=1}^N \big\| \mathbf{y}_{<i}^{\ell,k+1} - \mathbf{y}_{<i}^{m,k+1} \big\| \quad \text{(from gradient Lipschitz continuity of } f_\ell) \\
&\leq \frac{L}{N^2} \sum_{\ell=1}^N \sum_{m=1}^N \sum_{j<i} \big\| \mathbf{y}_j^{\ell,k+1} - \mathbf{y}_j^{m,k+1} \big\| \\
&\leq \frac{L}{N^2} \sum_{\ell=1}^N \sum_{m=1}^N \sum_{j<i} \alpha_j^k \big\| \nabla_{\mathbf{x}_j} f_\ell(\mathbf{y}_{<j}^{\ell,k+1}, \mathbf{x}_{\geq j}^k) - \nabla_{\mathbf{x}_j} f_m(\mathbf{y}_{<j}^{m,k+1}, \mathbf{x}_{\geq j}^k) \big\|,
\end{aligned}
$$

where the last inequality is from the nonexpansiveness of the proximal mapping in (1.6). Therefore, if $\|\nabla_{\mathbf{x}_i} f_\ell(\mathbf{x})\| \leq M$, for all, ℓ, for all $\mathbf{x}$, then we have from the above inequality that

$$
\big\| \mathbb{E}[\boldsymbol{\delta}_i^k | \mathbf{\Xi}_{k-1}] \big\| \leq 2LsM \max_j \alpha_j^k.
$$

The second condition (2.1b) in Assumption 1 is standard in the literature of the SG method. The variance bound σ_k^2 is allowed to vary with the iteration k.

Remark 2. Assumption 3 is relatively weaker than the assumption made in the literature of the SG method; for example, [25, 33] assume $\mathbf{x} \in \mathcal{X}$ for a bounded set $\mathcal{X}$. Assumption 3 is needed because the stochastic partial gradient $\tilde{\mathbf{g}}_i^k$ may be biased (see Remark 5). Our analysis below for the nonconvex case will remain valid if the assumption is weakened from the boundedness of $\{\mathbb{E}\|\mathbf{x}^k\|^2\}$ to that of $\{\mathbb{E}\|\nabla F(\mathbf{x}^k)\|^2\}$.

Note that Assumption 3 together with the partial gradient Lipschitz continuity of F in Assumption 2 implies the boundedness of $\{\mathbb{E}\|\nabla F(\mathbf{x}^k)\|^2\}$ by the following argument:

$$
\|\nabla_{\mathbf{x}_i} F(\mathbf{x}^k)\|^2 \leq 2\|\nabla_{\mathbf{x}_i} F(\mathbf{x}^k) - \nabla_{\mathbf{x}_i} F(\mathbf{0})\|^2 + 2\|\nabla_{\mathbf{x}_i} F(\mathbf{0})\|^2 \leq 2L^2 \|\mathbf{x}^k\|^2 + 2\|\nabla_{\mathbf{x}_i} F(\mathbf{0})\|^2.
$$

Throughout the paper, we let

$$
(2.3) \qquad\qquad M_\rho = \sqrt{4L^2 \rho^2 + 2 \max_i \|\nabla_{\mathbf{x}_i} F(\mathbf{0})\|^2},
$$

and thus we have $\mathbb{E}\|\nabla_{\mathbf{x}_i} F(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_{\geq i}^k)\|^2 \leq M_\rho^2$ for all i, k. In addition, by Jensen's inequality, we have $\mathbb{E}\|\mathbf{x}^k\| \leq \rho$, and $\mathbb{E}\|\nabla_{\mathbf{x}_i} F(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_{\geq i}^k)\| \leq M_\rho$ for all i, k.

Remark 3. Assumption 4 is the same as that in [25]. One example is $r_i(\mathbf{x}_i) = \lambda_i \|\mathbf{x}_i\|_1$, which satisfies (4) with $L_{r_i} = \lambda_i \sqrt{n_i}$. Note that (4) implies that $\|\mathbf{d}_i\| \leq L_r$ for any $\mathbf{d}_i \in \partial r_i(\mathbf{x}_i)$.

We first establish some lemmas, whose proofs are given in Appendix A. They are not difficult to prove and are useful in our convergence analysis.

LEMMA 2.1. *Let $\mathbf{u}^k$ be a random vector depending on $\Xi_{[k-1]}$. Under Assumption 1, if $\mathbf{u}^k$ is independent of $\boldsymbol{\delta}_i^k$ conditional on $\Xi_{[k-1]}$, then*

$$(2.4) \qquad \mathbb{E}\langle \mathbf{u}^k, \boldsymbol{\delta}_i^k \rangle \leq A\big(\max_j \alpha_j^k \big) \mathbb{E}\|\mathbf{u}^k\|.$$

LEMMA 2.2. *Let*

$$(2.5) \qquad \tilde{\mathbf{h}}_i^k = \frac{1}{\alpha_i^k}(\mathbf{x}_i^k - \mathbf{x}_i^{k+1})$$

be the SG mapping for the ith block at the kth iteration. Under Assumption 4, we have

$$\mathbb{E}\|\tilde{\mathbf{h}}_i^k\|^2 \leq 2\mathbb{E}\|\tilde{\mathbf{g}}_i^k\|^2 + 2L_{r_i}^2 \leq 4M_\rho^2 + 4\sigma_k^2 + 2L_{r_i}^2.$$

LEMMA 2.3. *If a nonnegative scalar sequence $\{A_k\}_{k\geq 1}$ satisfies*

$$A_{k+1} \leq \left(1 - \frac{a}{k}\right) A_k + \frac{b}{k^2},$$

where $a, b > 0$, then it obeys

$$(2.6) \qquad A_k \leq \frac{c}{k} \quad \forall k \geq \lfloor a \rfloor + 1,$$

with

$$c = \begin{cases} 2b/(a-1) & \text{if } a > 1, \\ \max\{b/a, A_1\} & \text{if } a \leq 1, \end{cases}$$

where $\lfloor a \rfloor$ denotes the largest integer that is no greater than a.

Next we present the convergence results of Algorithm 1 for both convex and nonconvex cases. For the convex case, where function F and the r_i's are convex, we establish a sublinear convergence rate in the same order of the SG method. For the nonconvex case, we establish the convergence result in terms of the expected violation of first-order optimality conditions.

2.1. Convex case. In this section, we first analyze Algorithm 1 for general convex problems and then for strongly convex problems to obtain stronger results.

THEOREM 2.4 (ergodic convergence for nonsmooth convex case). *Let $\{\mathbf{x}^k\}$ be generated from Algorithm 1 with $\alpha_i^k = \alpha_k = \frac{\theta}{\sqrt{k}} < \frac{1}{L}$ for all i, k, for some positive constant $\theta < \frac{1}{L}$. Under Assumptions 1 through 4, if F and the r_i's are all convex, $\mathbf{x}^*$ is a solution of (1.1) and $\sigma = \sup_k \sigma_k < \infty$, then*

$$(2.7) \qquad \mathbb{E}[\Phi(\tilde{\mathbf{x}}^K) - \Phi(\mathbf{x}^*)] \leq D\theta \frac{1 + \log K}{\sqrt{1+K}} + \frac{\|\mathbf{x}^* - \mathbf{x}^1\|^2}{2\theta\sqrt{1+K}},$$

where $\tilde{\mathbf{x}}^K = \frac{\sum_{k=1}^K \alpha_k \mathbf{x}^{k+1}}{\sum_{k=1}^K \alpha_k}$ *and*

$$(2.8) \quad D = \frac{s(\sigma^2 + 4L_{\max}^2)}{1 - L\theta} + \sqrt{s}(\|\mathbf{x}^*\| + \rho)\left(A + L\sqrt{\sum_{j=1}^s (4M_\rho^2 + 4\sigma_k^2 + 2L_{r_j}^2)}\right).$$

Furthermore, if the maximum number of iterations K is predetermined, then taking $\alpha_k = \frac{\theta}{\sqrt{K}}$ for all k, we have

$$(2.9) \qquad \mathbb{E}[\Phi(\tilde{\mathbf{x}}^K) - \Phi(\mathbf{x}^*)] \leq \frac{D\theta}{\sqrt{K}} + \frac{\|\mathbf{x}^* - \mathbf{x}^1\|^2}{2\theta\sqrt{K}}.$$

Remark 4. The value of θ usually plays a vital role on the actual speed of the algorithm. In practice, we may not know the exact value of L, and thus it is difficult to choose an appropriate θ. Even if we know or can estimate L, different θ's can make the algorithm perform very differently. This phenomenon has also been observed for the SG method; see numerical tests in section 3 and also the discussion on page 6 of [33]. The work [49] and its references study adaptive learning rates, which is beyond the discussion of this paper.

Remark 5. The proof below will clarify that the last (long) term in (2.8) is a result of the biased stochastic partial gradient (see (2.18) and (2.19) below). If unbiased, that is, $\mathbb{E}[\boldsymbol{\delta}_i^k \,|\, \boldsymbol{\Xi}_{(k-1)}] = \mathbf{0}$ holds instead of (2.1a), we will have the improved $D = \frac{s(\sigma^2 + 4L_{\max}^2)}{1 - L\theta}$ instead of (2.8), and then the result in (2.9) becomes comparable to that of SG since σ^2 is a bound of the *block partial* SG and thus $s\sigma^2$ a bound of the SG of F.

Although the existence of the second term in (2.8) makes the result in (2.9) seemingly worse than that of SG, multiblock ($s > 1$) Gauss–Seidel-type updates are generally more effective. Note that BSG has a per-iteration cost similar to that of SG. To update the first block, computing the sample partial gradient requires reading the current values of all the blocks. The subsequent updates in BSG are much cheaper because the sample partial gradients can be updated from the ones already computed. In addition, BSG can take greater step sizes than SG (because L in (2.2) is the Lipschitz constant of partial gradients). Therefore, BSG can perform much better than SG as shown by numerical results in section 3.

Minimizing the right-hand side of (2.9) by setting $\theta = \frac{\|\mathbf{x}^* - \mathbf{x}^1\|}{\sqrt{2D}}$, we have

$$(2.10) \qquad \mathbb{E}[\Phi(\tilde{\mathbf{x}}^K) - \Phi(\mathbf{x}^*)] \leq \frac{\sqrt{2D}\|\mathbf{x}^* - \mathbf{x}^1\|}{\sqrt{K}}.$$

Note that if $\mathbb{E}[\boldsymbol{\delta}_i^k \,|\, \boldsymbol{\Xi}_{(k-1)}] = \mathbf{0}$, the complexity in (2.10) becomes just a fraction $\mathcal{O}(\frac{1}{\sqrt{s}})$ of [9, (3.22)] since the quantity $s\sigma^2$ in (2.8) equals σ^2 in [9, (3.22)], and $L_{\max}$ disappears from D under the settings of [9]. Since the SBMD method of [9] only performs one block coordinate update at each iteration, its overall complexity is as good as ours. But, once again, BSG updates all the blocks while SBMD updates just a random one. Computationally, the cost of computing the sample partial gradients for all the blocks in BSG is dominated by the first one, which involves the current values of all the blocks and which is also needed by SBMD. Therefore, at each iteration, BSG and SBMD spend the same cost to update a block, but BSG then updates the rest of the blocks at very little extra cost. Therefore, BSG can have better overall performance (see numerical results in section 3.3).

Proof. From the Lipschitz continuity of $\nabla_{\mathbf{x}_i} F(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_i, \mathbf{x}_{>i}^k)$ about $\mathbf{x}_i$, it holds for any $i \in \mathcal{I}_1$ that (see [35], for example)

$$
\Phi(\mathbf{x}_{\leq i}^{k+1}, \mathbf{x}_{>i}^k) - \Phi(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_{\geq i}^k)
$$

(2.11)
$$
\leq \langle \mathbf{g}_i^k, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle + \frac{L}{2} \|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|^2 + r_i(\mathbf{x}_i^{k+1}) - r_i(\mathbf{x}_i^k).
$$

In addition, from Lemma 2 of [25], it holds for $i \in \mathcal{I}_2$ that

$$
\Phi(\mathbf{x}_{\leq i}^{k+1}, \mathbf{x}_{>i}^k) - \Phi(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_{\geq i}^k)
$$

(2.12)
$$
\leq \langle \mathbf{g}_i^k + \tilde{\nabla} r_i(\mathbf{x}_i^k), \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle + \frac{L}{2} \|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|^2 + 2L_{r_i} \|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|.
$$

By Lemma 2 of [2], the updates in (1.4) and (1.5) indicate that for any $\mathbf{x}_i \in \mathcal{X}_i$, if $i \in \mathcal{I}_1$, then

$$
\langle \tilde{\mathbf{g}}_i^k, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle + r_i(\mathbf{x}_i^{k+1}) + \frac{1}{2\alpha_k} \|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|^2
$$

(2.13)
$$
\leq \langle \tilde{\mathbf{g}}_i^k, \mathbf{x}_i - \mathbf{x}_i^k \rangle + r_i(\mathbf{x}_i) + \frac{1}{2\alpha_k} \big(\|\mathbf{x}_i - \mathbf{x}_i^k\|^2 - \|\mathbf{x}_i - \mathbf{x}_i^{k+1}\|^2 \big),
$$

and if $i \in \mathcal{I}_2$, then

$$
\langle \tilde{\mathbf{g}}_i^k + \tilde{\nabla} r_i(\mathbf{x}_i^k), \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle + \frac{1}{2\alpha_k} \|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|^2
$$

(2.14)
$$
\leq \langle \tilde{\mathbf{g}}_i^k + \tilde{\nabla} r_i(\mathbf{x}_i^k), \mathbf{x}_i - \mathbf{x}_i^k \rangle + \frac{1}{2\alpha_k} \big(\|\mathbf{x}_i - \mathbf{x}_i^k\|^2 - \|\mathbf{x}_i - \mathbf{x}_i^{k+1}\|^2 \big).
$$

Summing up (2.11) through (2.14) over i and arranging terms, we have

(2.15)
$$
\Phi(\mathbf{x}^{k+1}) - \Phi(\mathbf{x}^k) \leq \sum_{i=1}^s \langle \mathbf{g}_i^k - \tilde{\mathbf{g}}_i^k, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle - \big(\frac{1}{2\alpha_k} - \frac{L}{2} \big) \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2
$$
$$
+ \frac{1}{2\alpha_k} \big(\|\mathbf{x} - \mathbf{x}^k\|^2 - \|\mathbf{x} - \mathbf{x}^{k+1}\|^2 \big)
$$
$$
+ \sum_{i \in \mathcal{I}_1} \big(\langle \tilde{\mathbf{g}}_i^k, \mathbf{x}_i - \mathbf{x}_i^k \rangle + r_i(\mathbf{x}_i) - r_i(\mathbf{x}_i^k) \big)
$$
$$
+ \sum_{i \in \mathcal{I}_2} \Big(\langle \tilde{\mathbf{g}}_i^k + \tilde{\nabla} r_i(\mathbf{x}_i^k), \mathbf{x}_i - \mathbf{x}_i^k \rangle + 2L_{r_i} \|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\| \Big).
$$

From the convexity of F and the r_i's, we have $\Phi - \sum_{i \in \mathcal{I}_1} r_i$ convex and thus

$$
\Phi(\mathbf{x}^k) - \sum_{i \in \mathcal{I}_1} r_i(\mathbf{x}_i^k) \leq \Phi(\mathbf{x}) - \sum_{i \in \mathcal{I}_1} r_i(\mathbf{x}_i) + \langle \nabla F(\mathbf{x}^k), \mathbf{x}^k - \mathbf{x} \rangle + \sum_{i \in \mathcal{I}_2} \langle \tilde{\nabla} r_i(\mathbf{x}_i^k), \mathbf{x}_i^k - \mathbf{x}_i \rangle,
$$

which together with (2.15) gives

$$(2.16) \quad \Phi(\mathbf{x}^{k+1}) - \Phi(\mathbf{x}) \leq \sum_{i=1}^{s} \langle \mathbf{g}_i^k - \tilde{\mathbf{g}}_i^k, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2$$

$$+ \sum_{i \in \mathcal{I}_2} 2L_{r_i}\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|$$

$$+ \sum_{i=1}^{s} \langle \tilde{\mathbf{g}}_i^k - \nabla_{\mathbf{x}_i} F(\mathbf{x}^k), \mathbf{x}_i - \mathbf{x}_i^k \rangle$$

$$+ \frac{1}{2\alpha_k}\left(\|\mathbf{x} - \mathbf{x}^k\|^2 - \|\mathbf{x} - \mathbf{x}^{k+1}\|^2\right),$$

$$(\text{note } \tilde{\mathbf{g}}_i^k = \mathbf{g}_i^k + \boldsymbol{\delta}_i^k) = \sum_{i=1}^{s} \langle -\boldsymbol{\delta}_i^k, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2$$

$$+ \sum_{i \in \mathcal{I}_2} 2L_{r_i}\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|$$

$$+ \sum_{i=1}^{s} \langle \boldsymbol{\delta}_i^k + \mathbf{g}_i^k - \nabla_{\mathbf{x}_i} F(\mathbf{x}^k), \mathbf{x}_i - \mathbf{x}_i^k \rangle$$

$$+ \frac{1}{2\alpha_k}\left(\|\mathbf{x} - \mathbf{x}^k\|^2 - \|\mathbf{x} - \mathbf{x}^{k+1}\|^2\right).$$

Noting $\frac{1}{\alpha_k} - L > 0$ and using Young's inequality $ab < ta^2 + \frac{1}{4t}b^2$ in the following two inequalities, we have

$$\sum_{i=1}^{s} \langle -\boldsymbol{\delta}_i^k, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + \sum_{i \in \mathcal{I}_2} 2L_{r_i}\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|$$

$$\leq \sum_{i=1}^{s} \left(\langle -\boldsymbol{\delta}_i^k, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle - \frac{1}{4}\left(\frac{1}{\alpha_k} - L\right)\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|^2 \right)$$

$$+ \sum_{i \in \mathcal{I}_2} \left(2L_{r_i}\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\| - \frac{1}{4}\left(\frac{1}{\alpha_k} - L\right)\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|^2 \right)$$

$$(2.17) \quad \leq \sum_{i=1}^{s} \frac{\alpha_k}{1 - L\alpha_k}\|\boldsymbol{\delta}_i^k\|^2 + \sum_{i \in \mathcal{I}_2} \frac{4L_{r_i}^2 \alpha_k}{1 - L\alpha_k}.$$

In addition, letting $\mathbf{u}^k = \mathbf{x}_i - \mathbf{x}_i^k$ in Lemma 2.1, we have

$$(2.18) \qquad \qquad \mathbb{E}\langle \boldsymbol{\delta}_i^k, \mathbf{x}_i - \mathbf{x}_i^k \rangle \leq A\alpha_k \mathbb{E}\|\mathbf{x}_i - \mathbf{x}_i^k\|,$$

and also by Hölder's inequality, it holds that

$$\mathbb{E}\langle \mathbf{g}_i^k - \nabla_{\mathbf{x}_i} F(\mathbf{x}^k), \mathbf{x}_i - \mathbf{x}_i^k \rangle \leq \sqrt{\mathbb{E}\|\mathbf{g}_i^k - \nabla_{\mathbf{x}_i} F(\mathbf{x}^k)\|^2} \cdot \sqrt{\mathbb{E}\|\mathbf{x}_i - \mathbf{x}_i^k\|^2}$$

$$\leq L\alpha_k \sqrt{\mathbb{E}\sum_{j=1}^{s} \|\tilde{\mathbf{h}}_j^k\|^2} \cdot \sqrt{\mathbb{E}\|\mathbf{x}_i - \mathbf{x}_i^k\|^2}$$

$$(2.19) \qquad \qquad \leq L\alpha_k \sqrt{\sum_{j=1}^{s} (4M_\rho^2 + 4\sigma_k^2 + 2L_{r_j}^2)} \cdot \sqrt{\mathbb{E}\|\mathbf{x}_i - \mathbf{x}_i^k\|^2},$$

where we have used Lemma 2.2 in the last inequality.

Taking expectation over both sides of (2.16), substituting (2.1b), (2.17), (2.18), and (2.19) into it, and noting

$$\sum_{i=1}^{s} \sqrt{\mathbb{E}\|\mathbf{x}_i - \mathbf{x}_i^k\|^2} \leq \sqrt{s}\sqrt{\mathbb{E}\|\mathbf{x} - \mathbf{x}^k\|^2} \leq \sqrt{s}(\|\mathbf{x}\| + \rho),$$

we have after some arrangement that

$$\alpha_k \mathbb{E}[\Phi(\mathbf{x}^{k+1}) - \Phi(\mathbf{x})]$$

$$(2.20) \leq \left(\frac{s(\sigma^2 + 4L_{\max}^2)}{1 - L\alpha_k} + \sqrt{s}(\|\mathbf{x}\| + \rho)\left(A + L\sqrt{\sum_{j=1}^{s}(4M_\rho^2 + 4\sigma_k^2 + 2L_{r_j}^2)} \right) \right) \alpha_k^2$$

$$+ \frac{1}{2}\mathbb{E}(\|\mathbf{x} - \mathbf{x}^k\|^2 - \|\mathbf{x} - \mathbf{x}^{k+1}\|^2).$$

Letting $\mathbf{x} = \mathbf{x}^*$ in the above inequality and taking its sum over k, we have

$$(2.21) \qquad \sum_{k=1}^{K} \alpha_k \mathbb{E}[\Phi(\mathbf{x}^{k+1}) - \Phi(\mathbf{x}^*)] \leq D\sum_{k=1}^{K} \alpha_k^2 + \frac{1}{2}\|\mathbf{x}^* - \mathbf{x}^1\|^2.$$

Now use the convexity of Φ and $\alpha_k = \frac{\theta}{\sqrt{k}}$ in (2.21) to get

$$(2.22) \qquad \mathbb{E}[\Phi(\tilde{\mathbf{x}}^K) - \Phi(\mathbf{x}^*)] \leq D\frac{\sum_{k=1}^{K} \alpha_k^2}{\sum_{k=1}^{K} \alpha_k} + \frac{\|\mathbf{x}^* - \mathbf{x}^1\|^2}{2\sum_{k=1}^{K} \alpha_k}$$

$$\leq D\theta\frac{1 + \log K}{\sqrt{1 + K}} + \frac{\|\mathbf{x}^* - \mathbf{x}^1\|^2}{2\theta\sqrt{1 + K}}.$$

If the maximum number of iterations K is predetermined, set $\alpha_k \equiv \frac{\theta}{\sqrt{K}}$ in (2.22) to obtain

$$\mathbb{E}[\Phi(\tilde{\mathbf{x}}^K) - \Phi(\mathbf{x}^*)] \leq \frac{D\theta}{\sqrt{K}} + \frac{\|\mathbf{x}^* - \mathbf{x}^1\|^2}{2\theta\sqrt{K}},$$

which completes the proof.   □

Under the general convex setting, the rate $O(1/\sqrt{k})$ is optimal for the SG method, and for the strongly convex case, the rate $O(1/k)$ is optimal; see [25, 33, 34]. In the following theorem, we assume $\Phi(\mathbf{x})$ to be strongly convex and establish the rate $O(1/k)$. Hence, our algorithm has the same orders of convergence rates as that of the SG method.

THEOREM 2.5 (nonsmooth strongly convex case). *Let $\{\mathbf{x}^k\}$ be generated from Algorithm 1 with $\alpha_i^k = \alpha_k = \frac{\theta}{k} < \frac{1}{L}$ for all i, k. Under Assumptions 1 through 4, if F and the r_i's are convex, Φ is strongly convex with modulus $\mu > 0$, namely,*

$$\Phi(\lambda\mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda\Phi(\mathbf{x}) + (1 - \lambda)\Phi(\mathbf{y}) - \frac{\mu}{2}\lambda(1 - \lambda)\|\mathbf{x} - \mathbf{y}\|^2 \ \forall\mathbf{x}, \mathbf{y}, \forall\lambda \in [0, 1],$$

and $\sigma = \sup_k \sigma_k < \infty$, then

$$(2.23) \qquad \mathbb{E}[\|\mathbf{x}^k - \mathbf{x}^*\|^2] \leq \frac{1}{k}\max\left\{ \frac{2D\theta(1 + \mu\theta)}{\mu}, \|\mathbf{x}^1 - \mathbf{x}^*\|^2 \right\},$$

where D is defined in (2.8).

Proof. When Φ is strongly convex with modulus $\mu > 0$, it holds that

$$\Phi(\mathbf{x}^{k+1}) - \Phi(\mathbf{x}^*) \geq \frac{\mu}{2}\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2,$$

which together with (2.20) implies

$$
\begin{aligned}
\mathbb{E}\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 &\leq \frac{\mathbb{E}\|\mathbf{x}^k - \mathbf{x}^*\|^2}{1 + \mu\alpha_k} + 2D\frac{\alpha_k^2}{1 + \mu\alpha_k} \\
&= \frac{\mathbb{E}\|\mathbf{x}^k - \mathbf{x}^*\|^2}{1 + \mu\theta/k} + \frac{2D\theta^2}{k^2(1 + \mu\theta/k)} \\
&\leq \left(1 - \frac{\mu\theta}{k(1 + \mu\theta)}\right)\mathbb{E}\|\mathbf{x}^k - \mathbf{x}^*\|^2 + \frac{2D\theta^2}{k^2}.
\end{aligned}
$$
(2.24)

Using Lemma 2.3, we immediately get the desired result and complete the proof. $\square$

If the error σ_k decreases rapidly, we can show an almost linear convergence result. We assume that either $\mathcal{X}_i = \mathbb{R}^{n_i}$ or $r_i = 0$ for all i since it is impossible in general to get linear convergence for the subgradient method (see [5, 19, 38] and the references therein). For the convenience of our discussion, we let

$$\hat{r}_i(\mathbf{x}_i) = \begin{cases} r_i(\mathbf{x}_i), & i \in \mathcal{I}_1, \\ \iota_{\mathcal{X}_i}(\mathbf{x}_i), & i \in \mathcal{I}_2. \end{cases}$$
(2.25)

This way, we can write the updates (1.4) and (1.5) uniformly as

$$\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i}\langle \tilde{\mathbf{g}}_i^k, \mathbf{x}_i - \mathbf{x}_i^k\rangle + \frac{1}{2\alpha_i^k}\|\mathbf{x}_i - \mathbf{x}_i^k\|^2 + \hat{r}_i(\mathbf{x}_i).$$
(2.26)

Let

$$\hat{\Phi}(\mathbf{x}) = F(\mathbf{x}) + \sum_{i=1}^{s}\hat{r}_i(\mathbf{x}_i).$$
(2.27)

Then problem (1.1) becomes $\min_{\mathbf{x}} \hat{\Phi}(\mathbf{x})$ under the assumption that either $\mathcal{X}_i = \mathbb{R}^{n_i}$ or $r_i = 0$ for all i. When Φ and thus $\hat{\Phi}$ is strongly convex with modulus $\mu > 0$, then from the discussion in section 2.2 of [58], it follows that

$$\hat{\Phi}(\mathbf{x}) - \hat{\Phi}(\mathbf{x}^*) \leq \frac{1}{\mu}\|\mathbf{g}\|^2 \ \forall \mathbf{g} \in \partial\hat{\Phi}(\mathbf{x}).$$
(2.28)

Using this result, we establish the following theorem.

THEOREM 2.6. *Let $\{\mathbf{x}^k\}$ be generated from Algorithm 1 with $\alpha_i^k = \alpha_k < \frac{2}{L}$ for all i, k. Under Assumptions 1 and 2, if either $\mathcal{X}_i = \mathbb{R}^{n_i}$ or $r_i = 0$ for all i, Φ is strongly convex with constant μ, and all $\hat{r}_i$'s in (2.25) are convex, then*

$$\mathbb{E}[\Phi(\mathbf{x}^{k+1}) - \Phi(\mathbf{x}^*)] \leq \frac{\gamma(\alpha_k)}{1 + \gamma(\alpha_k)}\mathbb{E}[\Phi(\mathbf{x}^k) - \Phi(\mathbf{x}^*)] + \frac{s\nu(\alpha_k)}{1 + \gamma(\alpha_k)}\sigma_k^2,$$
(2.29)

where

$$\gamma(\alpha_k) = \frac{3L}{\mu^2}\left(\frac{1}{\alpha_k^2} + sL^2\right)\frac{1}{\frac{1}{2\alpha_k} - \frac{L}{4}}, \quad \nu(\alpha_k) = \frac{\gamma(\alpha_k)\alpha_k}{2 - L\alpha_k} + \frac{3L}{\mu^2}.$$
(2.30)

Remark 6. This theorem does not require Assumptions 3 or 4. Although it requires the strong convexity of Φ, it does not require the convexity of F.

Proof. Note

$$\frac{1}{\alpha_k}(\mathbf{x}_i^k - \mathbf{x}_i^{k+1}) - \tilde{\mathbf{g}}_i^k \in \partial \hat{r}_i(\mathbf{x}_i^{k+1})$$

or, equivalently,

$$\frac{1}{\alpha_k}(\mathbf{x}_i^k - \mathbf{x}_i^{k+1}) + \nabla_{\mathbf{x}_i}F(\mathbf{x}^{k+1}) - \tilde{\mathbf{g}}_i^k \in \partial \hat{r}_i(\mathbf{x}_i^{k+1}) + \nabla_{\mathbf{x}_i}F(\mathbf{x}^{k+1}).$$

Hence, from (2.28) it holds that

$$\hat{\Phi}(\mathbf{x}^{k+1}) - \hat{\Phi}(\mathbf{x}^*) \leq \frac{1}{\mu}\sum_{i=1}^s \left\| \frac{1}{\alpha_k}(\mathbf{x}_i^k - \mathbf{x}_i^{k+1}) + \nabla_{\mathbf{x}_i}F(\mathbf{x}^{k+1}) - \tilde{\mathbf{g}}_i^k \right\|^2$$

$$\leq \frac{3}{\mu}\sum_{i=1}^s \left(\left\| \frac{1}{\alpha_k}(\mathbf{x}_i^k - \mathbf{x}_i^{k+1}) \right\|^2 + \left\| \nabla_{\mathbf{x}_i}F(\mathbf{x}^{k+1}) - \mathbf{g}_i^k \right\|^2 + \|\boldsymbol{\delta}_i^k\|^2 \right).$$

Noting $\left\| \nabla_{\mathbf{x}_i}F(\mathbf{x}^{k+1}) - \mathbf{g}_i^k \right\|^2 \leq L^2\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2$, we have from the above inequality that

$$(2.31) \qquad \hat{\Phi}(\mathbf{x}^{k+1}) - \hat{\Phi}(\mathbf{x}^*) \leq \frac{3}{\mu}\left(\frac{1}{\alpha_k^2} + sL^2 \right)\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + \frac{3}{\mu}\sum_{i=1}^s \|\boldsymbol{\delta}_i^k\|^2.$$

Letting $\mathbf{x} = \mathbf{x}^k$ in (2.15) gives

$$\hat{\Phi}(\mathbf{x}^{k+1}) - \hat{\Phi}(\mathbf{x}^k) \leq -\sum_{i=1}^s \langle \boldsymbol{\delta}_i^k, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle - \left(\frac{1}{\alpha_k} - \frac{L}{2} \right)\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2$$

$$(2.32) \qquad\qquad \leq \frac{\alpha_k}{2 - L\alpha_k}\sum_{i=1}^s \|\boldsymbol{\delta}_i^k\|^2 - \left(\frac{1}{2\alpha_k} - \frac{L}{4} \right)\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2,$$

which together with (2.31) implies

$$\hat{\Phi}(\mathbf{x}^{k+1}) - \hat{\Phi}(\mathbf{x}^*)$$

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

$$\leq \frac{L}{\mu}\big(\hat{\Phi}(\mathbf{x}^{k+1}) - \hat{\Phi}(\mathbf{x}^*)\big)$$

$$\leq \frac{3L}{\mu^2}\big(\frac{1}{\alpha_k^2} + sL^2\big)\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + \frac{3L}{\mu^2}\sum_{i=1}^s \|\boldsymbol{\delta}_i^k\|^2$$

$$\leq \frac{3L}{\mu^2}\big(\frac{1}{\alpha_k^2} + sL^2\big)\frac{1}{\frac{1}{2\alpha_k} - \frac{L}{4}}\left(\Phi(\mathbf{x}^k) - \Phi(\mathbf{x}^{k+1}) + \frac{\alpha_k}{2 - L\alpha_k}\sum_{i=1}^s \|\boldsymbol{\delta}_i^k\|^2 \right) + \frac{3L}{\mu^2}\sum_{i=1}^s \|\boldsymbol{\delta}_i^k\|^2,$$

(2.33)

where the first inequality follows from the gradient Lipschitz continuity of F, the second one from the strong convexity of Φ, the third one from (2.31), and the fourth one

from (2.32). Taking expectation on both sides of (2.33), using (2.1b), and arranging terms give

$$\mathbb{E}[\hat{\Phi}(\mathbf{x}^{k+1}) - \hat{\Phi}(\mathbf{x}^*)] \le \gamma(\alpha_k)\mathbb{E}[\hat{\Phi}(\mathbf{x}^k) - \hat{\Phi}(\mathbf{x}^{k+1})] + s\nu(\alpha_k)\sigma_k^2,$$

which is equivalent to the desired result (2.29). Noting $\Phi(\mathbf{x}^k) = \hat{\Phi}(\mathbf{x}^k)$ and $\Phi(\mathbf{x}^*) = \hat{\Phi}(\mathbf{x}^*)$ completes the proof. $\quad\square$

From (2.29), one can see that the convergence rate of Algorithm 1 depends on how fast σ_k decreases. If $\sigma_k \equiv 0$, i.e., the deterministic case, (2.29) implies linear convergence. Generally, σ_k does not vanish. However, one can decrease it by increasing the batch size m_k. This point will be discussed in Remark 9 below. Directly from (2.29) and using the following lemma, we can get a convergence result in Theorem 2.8 below, whose proof follows [13].

LEMMA 2.7. *Let $e_k = \max\{a_k, \eta^k\}$. If a_{k+1}/a_k has a finite limit, then e_{k+1}/e_k also has a finite limit, which equals η or the limit of a_{k+1}/a_k, and the limit is no less than η.*

THEOREM 2.8. *Assume the assumptions in Theorem 2.6. Suppose $0 < \underline{\alpha} = \inf_k \alpha_k \le \sup_k \alpha_k = \overline{\alpha} < \frac{2}{L}$. If $\sigma_{k+1}^2/\sigma_k^2$ has a finite limit, then*

$$(2.34) \qquad \mathbb{E}[\Phi(\mathbf{x}^{k+1}) - \Phi(\mathbf{x}^*)] \le \eta^k(\Phi(\mathbf{x}^1) - \Phi(\mathbf{x}^*)) + \mathcal{O}(e_k),$$

where $\eta = \frac{\gamma(\overline{\alpha})}{1+\gamma(\overline{\alpha})}$ and $e_k = \max\{\sigma_k^2, (\eta + \epsilon)^k\}$ for any $\epsilon > 0$.

Proof. Note that $\gamma(\alpha)$ and $\nu(\alpha)$ defined in (2.30) are both increasing with respect to $\alpha \in (0, \frac{2}{L})$. Hence it follows from (2.29) that

$$\mathbb{E}[\Phi(\mathbf{x}^{k+1}) - \Phi(\mathbf{x}^*)] \le \frac{\gamma(\overline{\alpha})}{1+\gamma(\overline{\alpha})}\mathbb{E}[\Phi(\mathbf{x}^k) - \Phi(\mathbf{x}^*)] + \frac{s\nu(\overline{\alpha})}{1+\gamma(\underline{\alpha})}\sigma_k^2.$$

Let $\tau = \frac{s\nu(\overline{\alpha})}{1+\gamma(\underline{\alpha})}$. Then using the above inequality recursively yields

$$\mathbb{E}[\Phi(\mathbf{x}^{k+1}) - \Phi(\mathbf{x}^*)] \le \eta^k(\Phi(\mathbf{x}^1) - \Phi(\mathbf{x}^*)) + \tau \sum_{j=1}^{k} \eta^{k-j}\sigma_j^2.$$

From Lemma 2.7, we have $\lim_{k\to\infty} e_{k+1}/e_k \ge \eta + \epsilon$, and thus there is a sufficiently large integer K such that $e_{k+1}/e_k \ge \eta + \epsilon/2$ for all $k \ge K$. Let $\mu_k = \sum_{j=1}^{k} \eta^{k-j}\sigma_j^2$ and choose a sufficiently large number B such that $\mu_k \le Be_k$ for all $k \le K$ and $(B-1)\epsilon \ge 2\eta$. Note that such B must exist since K is finite and $\epsilon > 0$. Suppose that for some $k \ge K$, it holds $\mu_k \le Be_k$. Then

$$\mu_{k+1} = \eta\mu_k + \sigma_{k+1}^2 \le \eta Be_k + e_{k+1} \le \frac{\eta B}{\eta + \epsilon/2}e_{k+1} + e_{k+1} \le Be_{k+1},$$

where the last inequality uses the conditions $(B-1)\epsilon \ge 2\eta$ and $e_{k+1} \ge 0$. Hence, by induction, we conclude $\mu_k \le Be_k$ for all k. This completes the proof. $\quad\square$

Remark 7. If $m_k = \mathcal{O}(k)$ in Algorithm 1, then according to Remark 9, we have the convergence rate $\mathcal{O}\big(\max\{((\eta+1)/2)^k, 1/k\}\big)$ by letting $\epsilon = \frac{1-\eta}{2}$ in Theorem 2.8. If σ_k decreases linearly, then we have a linear convergence rate.

2.2. Nonconvex case. In this section, we do not assume any convexity of F or the r_i's. First, we analyze Algorithm 1 for the smooth case with $r_i = 0$ and $\mathcal{X}_i = \mathbb{R}^{n_i}$ for all i. Then, we impose a more restrictive condition on σ_k and analyze the algorithm for the unconstrained nonsmooth and constrained smooth cases. We start our analysis with the following lemma, which can be found in Lemma A.5 of [30] and Proposition 1.2.4 of [4].

LEMMA 2.9. *For two nonnegative scalar sequences $\{a_k\}$ and $\{b_k\}$, if $\sum_{k=1}^{\infty} a_k = +\infty$ and $\sum_{k=1}^{\infty} a_k b_k < +\infty$, then*

$$\liminf_{k \to \infty} b_k = 0.$$

Furthermore, if $|b_{k+1} - b_k| \leq B \cdot a_k$ for some constant $B > 0$, then

$$\lim_{k \to \infty} b_k = 0.$$

THEOREM 2.10. *Let $\{\mathbf{x}^k\}$ be generated from Algorithm 1 with $r_i = 0$ and $\mathcal{X}_i = \mathbb{R}^{n_i}$ for all i and with $\alpha_i^k = c_i^k \beta_k$, where c_i^k, β_k are positive scalars such that $\beta_k \geq \beta_{k+1}$ for all k,*

$$0 < \inf_k c_i^k \leq \sup_k c_i^k < \infty \quad \forall i, \tag{2.35}$$

and

$$\sum_{k=1}^{\infty} \beta_k = +\infty, \qquad \sum_{k=1}^{\infty} \beta_k^2 < +\infty. \tag{2.36}$$

Under Assumptions 1 through 3, if $\sigma = \sup_k \sigma_k < \infty$, then

$$\lim_{k \to \infty} \mathbb{E}\|\nabla \Phi(\mathbf{x}^k)\| = \lim_{k \to \infty} \mathbb{E}\|\nabla F(\mathbf{x}^k)\| = 0. \tag{2.37}$$

Proof. From the Lipschitz continuity of $\nabla_{\mathbf{x}_i} F$, it holds that

$$F(\mathbf{x}_{\leq i}^{k+1}, \mathbf{x}_{>i}^k) - F(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_{\geq i}^k)$$

$$\leq \langle \mathbf{g}_i^k, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle + \frac{L}{2}\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|^2$$

$$= -\alpha_i^k \langle \mathbf{g}_i^k, \tilde{\mathbf{g}}_i^k \rangle + \frac{L}{2}(\alpha_i^k)^2 \|\tilde{\mathbf{g}}_i^k\|^2$$

$$= -\left(\alpha_i^k - \frac{L}{2}(\alpha_i^k)^2\right)\|\mathbf{g}_i^k\|^2 + \frac{L}{2}(\alpha_i^k)^2\|\boldsymbol{\delta}_i^k\|^2 - (\alpha_i^k - L(\alpha_i^k)^2)\langle \mathbf{g}_i^k, \boldsymbol{\delta}_i^k \rangle$$

$$= -\left(\alpha_i^k - \frac{L}{2}(\alpha_i^k)^2\right)\|\mathbf{g}_i^k\|^2 + \frac{L}{2}(\alpha_i^k)^2\|\boldsymbol{\delta}_i^k\|^2 - (\alpha_i^k - L(\alpha_i^k)^2)\left(\langle \mathbf{g}_i^k - \nabla_{\mathbf{x}_i} F(\mathbf{x}^k), \boldsymbol{\delta}_i^k \rangle\right.$$

$$\left. + \langle \nabla_{\mathbf{x}_i} F(\mathbf{x}^k), \boldsymbol{\delta}_i^k \rangle\right)$$

$$\leq -\left(\alpha_i^k - \frac{L}{2}(\alpha_i^k)^2\right)\|\mathbf{g}_i^k\|^2 + \frac{L}{2}(\alpha_i^k)^2\|\boldsymbol{\delta}_i^k\|^2 - (\alpha_i^k - L(\alpha_i^k)^2)\langle \nabla_{\mathbf{x}_i} F(\mathbf{x}^k), \boldsymbol{\delta}_i^k \rangle$$

$$+ L(\alpha_i^k + L(\alpha_i^k)^2)\max_j \alpha_j^k \left(\|\boldsymbol{\delta}_i^k\|^2 + \sum_{j=1}^s (\|\mathbf{g}_j^k\|^2 + \|\boldsymbol{\delta}_j^k\|^2)\right),$$

$$\tag{2.38}$$

where the last inequality follows from the following argument:

$$-\left(\alpha_i^k - L(\alpha_i^k)^2\right)\langle \mathbf{g}_i^k - \nabla_{\mathbf{x}_i}F(\mathbf{x}^k), \boldsymbol{\delta}_i^k\rangle$$
$$\leq |\alpha_i^k - L(\alpha_i^k)^2|\|\boldsymbol{\delta}_i^k\|\|\mathbf{g}_i^k - \nabla_{\mathbf{x}_i}F(\mathbf{x}^k)\|$$
$$\leq L|\alpha_i^k - L(\alpha_i^k)^2|\|\boldsymbol{\delta}_i^k\|\|\mathbf{x}_{<i}^{k+1} - \mathbf{x}_{<i}^k\| \qquad \text{(from gradient Lipschitz continuity of } F)$$
$$\leq L|\alpha_i^k - L(\alpha_i^k)^2|\|\boldsymbol{\delta}_i^k\|\sqrt{\sum_{j=1}^s \|\alpha_j^k\tilde{\mathbf{g}}_j^k\|^2} \qquad \text{(note } \mathbf{x}_j^{k+1} = \mathbf{x}_j^k - \alpha_j^k\tilde{\mathbf{g}}_j^k)$$
$$\leq L\left(\alpha_i^k + L(\alpha_i^k)^2\right)\max_j \alpha_j^k\left(\|\boldsymbol{\delta}_i^k\|^2 + \sum_{j=1}^s(\|\mathbf{g}_j^k\|^2 + \|\boldsymbol{\delta}_j^k\|^2)\right) \qquad \begin{array}{l}\text{(from the} \\ \text{triangle inequality).}\end{array}$$

Summing (2.38) over i and using (2.1b) give

(2.39)

$$F(\mathbf{x}^{k+1}) - F(\mathbf{x}^k) \leq -\sum_{i=1}^s\left(\alpha_i^k - \frac{L}{2}(\alpha_i^k)^2\right)\|\mathbf{g}_i^k\|^2 - \sum_{i=1}^s\left(\alpha_i^k - L(\alpha_i^k)^2\right)\langle\nabla_{\mathbf{x}_i}F(\mathbf{x}^k), \boldsymbol{\delta}_i^k\rangle$$
$$+ \sum_{i=1}^s\left(\frac{L}{2}(\alpha_i^k)^2\|\boldsymbol{\delta}_i^k\|^2 + L\left(\alpha_i^k + L(\alpha_i^k)^2\right)\max_j \alpha_j^k\left(\|\boldsymbol{\delta}_i^k\|^2 + \sum_{j=1}^s(\|\mathbf{g}_j^k\|^2 + \|\boldsymbol{\delta}_j^k\|^2)\right)\right).$$

Note that since $\mathbf{x}^k$ is independent of Ξ_k, from Lemma 2.1 we have

$$\mathbb{E}\langle\nabla_{\mathbf{x}_i}F(\mathbf{x}^k), \boldsymbol{\delta}_i^k\rangle \leq A \cdot \max_j \alpha_j^k \cdot \mathbb{E}\|\nabla_{\mathbf{x}_i}F(\mathbf{x}^k)\| \leq M_\rho A \cdot \max_j \alpha_j^k,$$

where M_ρ is defined in (2.3). Taking expectation over (2.39), we have

$$\mathbb{E}F(\mathbf{x}^{k+1}) - \mathbb{E}F(\mathbf{x}^k)$$
$$\leq -\sum_{i=1}^s\left(\alpha_i^k - \frac{L}{2}(\alpha_i^k)^2\right)\mathbb{E}\|\mathbf{g}_i^k\|^2 + \sum_{i=1}^s M_\rho A(\alpha_i^k + L(\alpha_i^k)^2)\max_j \alpha_j^k$$
$$+ \sum_{i=1}^s\left(\frac{L}{2}(\alpha_i^k)^2\sigma^2 + L\left(\alpha_i^k + L(\alpha_i^k)^2\right)\max_j \alpha_j^k\left((s+1)\sigma^2 + sM_\rho^2\right)\right)$$

(2.40)
$$\leq -\sum_{i=1}^s\left(c\beta_k - \frac{LC^2}{2}\beta_k^2\right)\mathbb{E}\|\mathbf{g}_i^k\|^2 + \sum_{i=1}^s M_\rho AC\beta_k(C\beta_k + LC^2\beta_k^2)$$
$$+ \sum_{i=1}^s\left(\frac{LC^2}{2}\beta_k^2\sigma^2 + L\beta_k(C\beta_k + LC^2\beta_k^2)\left((s+1)\sigma^2 + sM_\rho^2\right)\right),$$

where

(2.41)
$$c = \min_i \inf_k c_i^k, \qquad C = \max_i \sup_k c_i^k.$$

Note that F is lower bounded, $\mathbb{E}\|\mathbf{g}_i^k\|^2 \leq M_\rho^2$, and $0 < c \leq C < \infty$. Summing (2.40) over k and using (2.36), we have

$$\sum_{k=1}^\infty \beta_k\mathbb{E}\|\mathbf{g}_i^k\|^2 < \infty \quad \forall i.$$

Furthermore,

$$
\begin{aligned}
&\bigl|\mathbb{E}\|\mathbf{g}_i^{k+1}\|^2 - \mathbb{E}\|\mathbf{g}_i^k\|^2\bigr| \\
\leq&\mathbb{E}\bigl[\|\mathbf{g}_i^{k+1} + \mathbf{g}_i^k\| \cdot \|\mathbf{g}_i^{k+1} - \mathbf{g}_i^k\|\bigr] \\
\leq&2LM_\rho\mathbb{E}\bigl\|(\mathbf{x}_{j<i}^{k+2}, \mathbf{x}_{j\geq i}^{k+1}) - (\mathbf{x}_{j<i}^{k+1}, \mathbf{x}_{j\geq i}^k)\bigr\| && \text{(from Remark 2 and gradient} \\
&&& \text{Lipschitz continuity of } F) \\
=&2LM_\rho\mathbb{E}\sqrt{\sum_{j<i}\|\alpha_j^{k+1}\tilde{\mathbf{g}}_j^{k+1}\|^2 + \sum_{j\geq i}\|\alpha_j^k\tilde{\mathbf{g}}_j^k\|^2} && \text{(note } \mathbf{x}_j^{k+1} = \mathbf{x}_j^k - \alpha_j^k\tilde{\mathbf{g}}_j^k) \\
\leq&2LM_\rho C\beta_k\mathbb{E}\sqrt{\sum_{j<i}\|\tilde{\mathbf{g}}_j^{k+1}\|^2 + \sum_{j\geq i}\|\tilde{\mathbf{g}}_j^k\|^2} && \text{(from definition of } \alpha_i^k \text{ and the} \\
&&& \text{monotonicity of } \beta_k) \\
\leq&2LM_\rho C\beta_k\sqrt{\mathbb{E}\bigl[\sum_{j<i}\|\tilde{\mathbf{g}}_j^{k+1}\|^2 + \sum_{j\geq i}\|\tilde{\mathbf{g}}_j^k\|^2\bigr]} && \text{(from Jensen's inequality)} \\
\leq&2LM_\rho C\beta_k\sqrt{2s(M_\rho^2 + \sigma^2)} && \text{(from Jensen's inequality).}
\end{aligned}
$$

(2.42)

According to Lemma 2.9, we have $\mathbb{E}\|\mathbf{g}_i^k\|^2 \to 0$ for all i, as $k \to \infty$ and thus $\mathbb{E}\|\mathbf{g}_i^k\| \to 0$ for all i, as $k \to \infty$ by Jensen's inequality. Hence,

$$
\begin{aligned}
\mathbb{E}\|\nabla_{\mathbf{x}_i}F(\mathbf{x}^k)\| &\leq \mathbb{E}\|\nabla_{\mathbf{x}_i}F(\mathbf{x}^k) - \mathbf{g}_i^k\| + \mathbb{E}\|\mathbf{g}_i^k\| \\
&\leq L \cdot \mathbb{E}\|\mathbf{x}_{<i}^{k+1} - \mathbf{x}_{<i}^k\| + \mathbb{E}\|\mathbf{g}_i^k\| \\
&\leq LC\sqrt{2s(M_\rho^2 + \sigma^2)}\beta_k + \mathbb{E}\|\mathbf{g}_i^k\| \to 0 \; \forall i, \text{ as } k \to \infty,
\end{aligned}
$$

where the last inequality is obtained following the same argument as for (2.42). Therefore, the desired result is obtained. $\square$

Remark 8. The above proof only needs the boundedness of $\mathbb{E}\|\mathbf{g}_i^k\|^2$ for all i, k, instead of the stronger Assumption 3.

We next analyze Algorithm 1 for the nonsmooth case with the notation in (2.25) through (2.27).

THEOREM 2.11. *Let* $\{\mathbf{x}^k\}$ *be generated from Algorithm* 1 *with the* α_i^k's *being the same as those in Theorem* 2.10. *Under Assumptions* 1 *through* 4, *if either* $\mathcal{X}_i = \mathbb{R}^{n_i}$ *or* $r_i = 0$ *for all* i, *and*

$$
(2.43) \qquad\qquad\qquad \sum_{k=1}^{\infty}\beta_k\sigma_k^2 < \infty,
$$

then there exists an index subsequence $\mathcal{K}$ *such that*

$$
(2.44) \qquad\qquad\qquad \lim_{\substack{k\to\infty \\ k\in\mathcal{K}}} \mathbb{E}\bigl[\text{dist}\bigl(\mathbf{0}, \partial\hat{\Phi}(\mathbf{x}^k)\bigr)\bigr] = 0,
$$

where $\text{dist}(\mathbf{y}, \mathcal{X}) = \min_{\mathbf{x}\in\mathcal{X}}\|\mathbf{x} - \mathbf{y}\|$.

Proof. From the optimality of $\mathbf{x}_i^{k+1}$ for (2.26), it holds that

$$
\hat{r}_i(\mathbf{x}_i^{k+1}) - \hat{r}_i(\mathbf{x}_i^k) \leq -\langle\tilde{\mathbf{g}}_i^k, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k\rangle - \frac{1}{2\alpha_i^k}\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|^2.
$$

In addition, from the Lipschitz continuity condition (2.2), it follows that

$$F(\mathbf{x}_{\leq i}^{k+1}, \mathbf{x}_{>i}^k) - F(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_{\geq i}^k) \leq \langle \mathbf{g}_i^k, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle + \frac{L}{2} \|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|^2.$$

From (2.35) and (2.36), we have $\alpha_i^k \to 0$ for all i as $k \to \infty$ and can thus take sufficiently large $\tilde{k}$ such that $\alpha_i^k < \frac{1}{2L}$ for all $i, k \geq \tilde{k}$. Summing up the above two inequalities and assuming $k \geq \tilde{k}$, we have

$$\hat{\Phi}(\mathbf{x}_{\leq i}^{k+1}, \mathbf{x}_{>i}^k) - \hat{\Phi}(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_{\geq i}^k)$$

$$(2.45) \qquad \leq -\langle \boldsymbol{\delta}_i^k, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle - \left(\frac{1}{2\alpha_i^k} - \frac{L}{2} \right) \|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|^2$$

$$(2.46) \qquad \leq \frac{\alpha_i^k}{1 - L\alpha_i^k} \|\boldsymbol{\delta}_i^k\|^2 - \frac{1}{2} \left(\frac{1}{2\alpha_i^k} - \frac{L}{2} \right) \|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|^2$$

$$= \frac{\alpha_i^k}{1 - L\alpha_i^k} \|\boldsymbol{\delta}_i^k\|^2 - \left(\frac{\alpha_i^k}{4} - \frac{L}{4}(\alpha_i^k)^2 \right) \|\tilde{\mathbf{h}}_i^k\|^2,$$

where we have used Young's inequality in the second inequality. Summing the above inequality over i yields

$$(2.47) \qquad \hat{\Phi}(\mathbf{x}^{k+1}) - \hat{\Phi}(\mathbf{x}^k) \leq \sum_{i=1}^s \frac{\alpha_i^k}{1 - L\alpha_i^k} \|\boldsymbol{\delta}_i^k\|^2 - \sum_{i=1}^s \left(\frac{\alpha_i^k}{4} - \frac{L}{4}(\alpha_i^k)^2 \right) \|\tilde{\mathbf{h}}_i^k\|^2.$$

Taking expectation on both sides of the above inequality gives

$$\mathbb{E}\hat{\Phi}(\mathbf{x}^{k+1}) - \mathbb{E}\hat{\Phi}(\mathbf{x}^k) \leq \sum_{i=1}^s \frac{\alpha_i^k}{1 - L\alpha_i^k} \sigma_k^2 - \sum_{i=1}^s \left(\frac{\alpha_i^k}{4} - \frac{L}{4}(\alpha_i^k)^2 \right) \mathbb{E}\|\tilde{\mathbf{h}}_i^k\|^2$$

$$(2.48) \qquad \qquad \leq \sum_{i=1}^s 2C\beta_k \sigma_k^2 - \sum_{i=1}^s \left(\frac{c}{4}\beta_k - \frac{L}{4}C^2\beta_k^2 \right) \mathbb{E}\|\tilde{\mathbf{h}}_i^k\|^2,$$

where in the second inequality we have assumed $\alpha_i^k < \frac{1}{2L}$ by taking $k \geq \tilde{k}$, and c and C are defined in (2.41). Note that from Lemma 2.2, we have that $\mathbb{E}\|\tilde{\mathbf{h}}_i^k\|^2$ is bounded for all i and k. In addition, by (2.36) and (2.43) and recalling that Φ, and thus $\hat{\Phi}$, is lower bounded, we have from (2.48) that

$$\sum_{k=0}^\infty \beta_k \mathbb{E}\|\tilde{\mathbf{h}}_i^k\|^2 < \infty \ \forall i.$$

Hence, from Lemma 2.9, there must be a subsequence $\{\mathbb{E}\|\tilde{\mathbf{h}}_i^k\|^2\}_{k \in \mathcal{K}}$ converging to *zero* for all i.

Furthermore, note that

$$\tilde{\mathbf{h}}_i^k = \frac{1}{\alpha_i^k}(\mathbf{x}_i^k - \mathbf{x}_i^{k+1}) \in \tilde{\mathbf{g}}_i^k + \partial \hat{r}_i(\mathbf{x}_i^{k+1}) \ \forall i,$$

which is equivalent to

$$\tilde{\mathbf{h}}_i^k + \nabla_{\mathbf{x}_i} F(\mathbf{x}^{k+1}) - \tilde{\mathbf{g}}_i^k \in \nabla_{\mathbf{x}_i} F(\mathbf{x}^{k+1}) + \partial \hat{r}_i(\mathbf{x}_i^{k+1}) \ \forall i.$$

Hence,

$$\text{dist}(\mathbf{0}, \partial\hat{\Phi}(\mathbf{x}^{k+1}))^2 \leq \sum_{i=1}^{s} \|\tilde{\mathbf{h}}_i^k + \nabla_{\mathbf{x}_i} F(\mathbf{x}^{k+1}) - \tilde{\mathbf{g}}_i^k\|^2$$

$$\leq 2\sum_{i=1}^{s} \left(\|\tilde{\mathbf{h}}_i^k\|^2 + \|\nabla_{\mathbf{x}_i} F(\mathbf{x}^{k+1}) - \tilde{\mathbf{g}}_i^k\|^2\right)$$

$$\leq 4\sum_{i=1}^{s} \left(\|\tilde{\mathbf{h}}_i^k\|^2 + \|\nabla_{\mathbf{x}_i} F(\mathbf{x}^{k+1}) - \mathbf{g}_i^k\|^2 + \|\boldsymbol{\delta}_i^k\|^2\right),$$

and thus taking expectation over the last inequality gives

$$\mathbb{E}[\text{dist}(\mathbf{0}, \partial\hat{\Phi}(\mathbf{x}^{k+1}))^2] \leq 4\sum_{i=1}^{s} \left(\mathbb{E}\|\tilde{\mathbf{h}}_i^k\|^2 + \mathbb{E}\|\nabla_{\mathbf{x}_i} F(\mathbf{x}^{k+1}) - \mathbf{g}_i^k\|^2 + \mathbb{E}\|\boldsymbol{\delta}_i^k\|^2\right).$$

Following the same argument as that at the end of the proof of Theorem 2.10, one can easily show that $\mathbb{E}\|\nabla_{\mathbf{x}_i} F(\mathbf{x}^{k+1}) - \mathbf{g}_i^k\|^2 \to 0$ for all i, as $k \to \infty$. Taking another subsequence of $\mathcal{K}$ if necessary, we have $\mathbb{E}\|\boldsymbol{\delta}_i^k\|^2 \leq \sigma_k^2 \to 0$ for all i, as $\mathcal{K} \ni k \to \infty$ from (2.43). Hence, the right-hand side of the last inequality converges to *zero* as $\mathcal{K} \ni k \to \infty$. Applying Jensen's inequality completes the proof. ☐

From (2.47) in the above proof, if σ_k decreases in a faster way, we can show that (2.44) holds for $\mathcal{K}$ being the whole index sequence. This result is summarized below.

COROLLARY 2.12. *Let $\{\mathbf{x}^k\}$ be generated from Algorithm* 1 *with the α_i^k's satisfying $0 < \inf_k \alpha_i^k \leq \sup_k \alpha_i^k < \frac{1}{L}$ for all i. Under Assumptions* 1 *through* 4, *if either $\mathcal{X}_i = \mathbb{R}^{n_i}$ or $r_i = 0$ for all i and $\sum_{k=1}^{\infty} \sigma_k^2 < \infty$, then*

$$\lim_{k\to\infty} \mathbb{E}\left[\text{dist}\left(\mathbf{0}, \partial\hat{\Phi}(\mathbf{x}^k)\right)\right] = 0,$$

which says that the optimality condition for (1.1) *asymptotically holds in expectation.*

Proof. Under the conditions that $0 < \inf_k \alpha_i^k \leq \sup_k \alpha_i^k < \frac{1}{L}$ for all i and $\sum_{k=1}^{\infty} \sigma_k^2 < \infty$, it follows from (2.47) that the subsequence $\mathcal{K}$ in the proof of Theorem 2.11 can be taken as the whole sequence. Noting $\sigma_k \to 0$ as $k \to \infty$, we immediately get the desired result from Theorem 2.11. ☐

Remark 9. One way to make $\sum_{k=1}^{\infty} \sigma_k^2 < \infty$ is to asymptotically increase m_k at a sufficiently fast rate. Let

$$\boldsymbol{\delta}_{i,\ell}^k = \nabla_{\mathbf{x}_i} f(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_{\geq i}^k; \xi_{k,\ell}) - \mathbf{g}_i^k.$$

Then, $\boldsymbol{\delta}_i^k = \frac{1}{m_k}\sum_{\ell=1}^{m_k} \boldsymbol{\delta}_{i,\ell}^k$. As in Assumption 1, assume $\|\mathbb{E}[\boldsymbol{\delta}_{i,\ell}^k|\boldsymbol{\Xi}_{[k-1]}]\| \leq A \cdot \max_j \alpha_j^k$ and also $\mathbb{E}\|\boldsymbol{\delta}_{i,\ell}^k\|^2 \leq \sigma^2$ for some constants A and σ. Then following the proof of [18, page 11], one can show that $\mathbb{E}\|\boldsymbol{\delta}_i^k\|^2 \leq \mathcal{O}(1/m_k)$. Hence, taking $m_k = \lceil k^{1+\epsilon} \rceil$ for any $\epsilon > 0$ guarantees $\sum_{k=1}^{\infty} \sigma_k^2 < \infty$, where $\lceil a \rceil$ denotes the smallest integer that is no less than a.

3. Numerical experiments. In this section, we report the simulation results of Algorithm 1, dubbed as BSG, on both convex and nonconvex problems to demonstrate its advantages. The tested convex problems include stochastic least squares (3.2) and logistic regression (3.3). The tested nonconvex problems include low-rank tensor recovery (3.4) and bilinear logistic regression (3.5).

3.1. Parameter settings. For convex problems (3.2), (3.3), we let $(\pi_1^k, \ldots, \pi_s^k)$ be a *random shuffling* of $(1, \ldots, s)$ and set the stepsize of BSG to $\alpha_i^k = \min(\frac{\theta}{\sqrt{k}}, \frac{1}{L_i^k})$ for all i, k, where the value of θ was specified in each test, and L_i^k was the Lipschitz constant of

$$(3.1) \qquad \frac{1}{m_k} \sum_{\ell=1}^{m_k} \nabla_{\mathbf{x}_{\pi_i^k}} f(\mathbf{x}_{\pi_{<i}^k}^{k+1}, \mathbf{x}_i, \mathbf{x}_{\pi_{>i}^k}^k; \xi_{k,\ell})$$

with respect to $\mathbf{x}_i$. In addition, each $\xi_{k,\ell}$ was generated uniformly at random, and m_k was set the same for all k and specified below. We treat each coordinate as a block, i.e., $s = n$. Within each iteration, although BSG requires computing block partial gradient s times, we need very little extra computation (with complexity $\mathcal{O}(1)$) to get the new partial gradient from the previous one due to cyclic update and thus greatly save on computing time.

For nonconvex problems (3.4) and (3.5), we treat each factor matrix as a block, i.e., $s = 3$ for (3.4) and $s = 2$ for (3.5). We used the fixed updated order by letting $\pi_i^k = i$ for all i, k, and set $\alpha_i^k = \min(\frac{\theta}{\sqrt{k} \log k}, \frac{1}{L_i^k})$ for smooth nonconvex problems, i.e., problems (3.4) with $\lambda = 0$ and (3.5), and set $\alpha_i^k = \frac{1}{L_i^k}$ for nonsmooth nonconvex ones, i.e., problem (3.4) with $\lambda > 0$, where L_i^k was the Lipschitz constant of (3.1). In addition, for smooth nonconvex cases, m_k was set the same for all k, and for the non-smooth nonconvex case, we asymptotically increased it by $m_k = m_1 + \lceil \frac{k-1}{10} \rceil$ for all k.

We compared BSG with the SG method and the SBMD method [9] on problems (3.2) and (3.3). Step size α_k for SG and SBMD was set in the same way as α_i^k in the above. Specifically, $\alpha_k = \min(\frac{\theta}{\sqrt{k}}, \frac{1}{L_k})$, where L_k was the Lipschitz constant of $\frac{1}{m_k} \sum_{\ell=1}^{m_k} \nabla_{\mathbf{x}} f(\mathbf{x}; \xi_{k,\ell})$ for SG and $\frac{1}{m_k} \sum_{\ell=1}^{m_k} \nabla_{\mathbf{x}_{i_k}} f(\mathbf{x}_{\neq i_k}^k, \mathbf{x}_{i_k}; \xi_{k,\ell})$ with respect to the selected block $\mathbf{x}_{i_k}$ for SBMD. On solving (3.4) and (3.5), we compared BSG with the BCGD (block coordinate gradient descent) method [56], whose step size λ_i^k was taken as the reciprocal of the Lipschitz constant of $\nabla_{\mathbf{x}_i} F(\mathbf{x}_{<i}^{k+1}, \mathbf{x}_i, \mathbf{x}_{>i}^k)$ with respect to $\mathbf{x}_i$ for all i and k. Throughout our tests, all compared algorithms were supplied with the same randomly generated starting point.

3.2. Stochastic least squares. We tested BSG, SG, and SBMD on the problem

$$(3.2) \qquad \min_{\mathbf{x}} \mathbb{E}_{\mathbf{a}, b} \frac{1}{2} (\mathbf{a}^\top \mathbf{x} - b)^2,$$

where $\mathbf{a}$ and b were random variables. In this test, entries of $\mathbf{a}$ independently followed the standard Gaussian distribution, and $b = \mathbf{a}^\top \hat{\mathbf{x}} + \eta$, where η was independent of $\mathbf{a}$ and followed the Gaussian distribution $\mathcal{N}(0, 0.01)$, and $\hat{\mathbf{x}} \in \mathbb{R}^{200}$ was a deterministic vector. It is easy to show that $\hat{\mathbf{x}}$ was the solution to (3.2), and the optimal objective value was 0.005. We first generated a Gaussian random vector $\hat{\mathbf{x}}$. Then we generated N samples of $\mathbf{a}$ and b according to their distributions, one at a time, and for each sample, we performed one update of the three algorithms, i.e., $m_k = 1$ in (3.1). All three algorithms started from the same Gaussian randomly generated point and used $\theta = 0.1$. To compare their solutions, we generated another 100,000 samples $(\mathbf{a}, b)$ following the same distribution and calculated the empirical loss. The entire process was repeated 100 times independently, and average empirical losses were shown in Table 2 for different N's. "SBMD-t" denotes an SBMD algorithm that independently selected t coordinates at each iteration. Since "SBMD-200" becomes an SG method, its results were identical to those of SG and thus not reported. From the results, we

Objective values of BSG, SG, and SBMD on solving stochastic least squares (3.2) *with data following the Gaussian distribution.* Bold numbers *are the best results.*

N (Total Samples)	BSG	SG	SBMD-10	SBMD-50	SBMD-100
4000	6.45e-3	**6.03e-3**	67.49	4.79	1.03e-1
6000	**5.69e-3**	5.79e-3	53.84	1.43	1.43e-2
8000	**5.57e-3**	5.65e-3	42.98	4.92e-1	6.70e-3
10000	**5.53e-3**	5.58e-3	35.71	2.09e-1	5.74e-3

see that SBMD performed consistently better by updating more coordinates at each iteration. BSG was better than SG except for $N = 4000$. Note that SBMD only renewed partial coordinates at each update and thus took less computing time.

3.3. Logistic regression. We tested BSG, SG, and SBMD on the problem

$$(3.3) \qquad \min_{\mathbf{w},b} \frac{1}{N} \sum_{\ell=1}^{N} \log(1 + \exp(-y_\ell(\mathbf{x}_\ell^\top \mathbf{w} + b))).$$

First, we compared the three algorithms on synthetic data. We randomly generated $N = 2000$ samples of dimension 200 with half of them belonging to the "+1" class and the other half to the "−1" class. Each sample in the positive class has components independently sampled from the Gaussian distribution $\mathcal{N}(5, 1)$ and those in the negative class from $\mathcal{N}(-5, 1)$. We ran BSG, SG, and SBMD each to 50 epochs or 2 seconds, where one epoch was equivalent to going through all samples once. At each iteration of the algorithms, we uniformly randomly selected one sample, i.e., $m_k = 1$ in (3.1). Three different values of θ were tested. Figure 1 plots the gap between the optimal objective value and those given by different algorithms for solving (3.3), where the optimal objective value was accurately obtained by running FISTA [2] to 5,000 iterations. From the figure, we see that when θ was small, SBMD performed consistently better if more coordinates were updated at each iteration, and when θ was large, its performance was almost irrelevant to the numbers of updated coordinates. In addition, the proposed BSG method performed the best, and it reached a much lower objective within the same number of epochs or the same amount of running time, especially when a large θ was used. With respect to running time, the worse performance of SBMD-1 compared to BSG is possibly because SBMD-1 used all coordinates to evaluate every partial gradient (with complexity $\mathcal{O}(n)$) while BSG used all coordinates only for the first partial gradient and then just the renewed coordinates for all other partial gradients (each with complexity $\mathcal{O}(1)$) due to cyclic update.

Second, we compared the three algorithms on the gisette dataset,[2] which has 6,000 training samples of dimension 5,000. At each iteration, SBMD updated 1, 1,000, or 3,000 coordinates. Figure 2 plots the results of the three algorithms on problem (3.3) with three different values of θ. From the figures, we see that when $\theta = 0.1$, BSG performed almost the same as SG with respect to the number of epochs, and they both outperformed SBMD within the same number of epochs. When θ becomes larger, BSG reaches much lower objective values than SG and SBMD, and the latter two performed almost the same except for SBMD-1.

---

[2] Available from http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
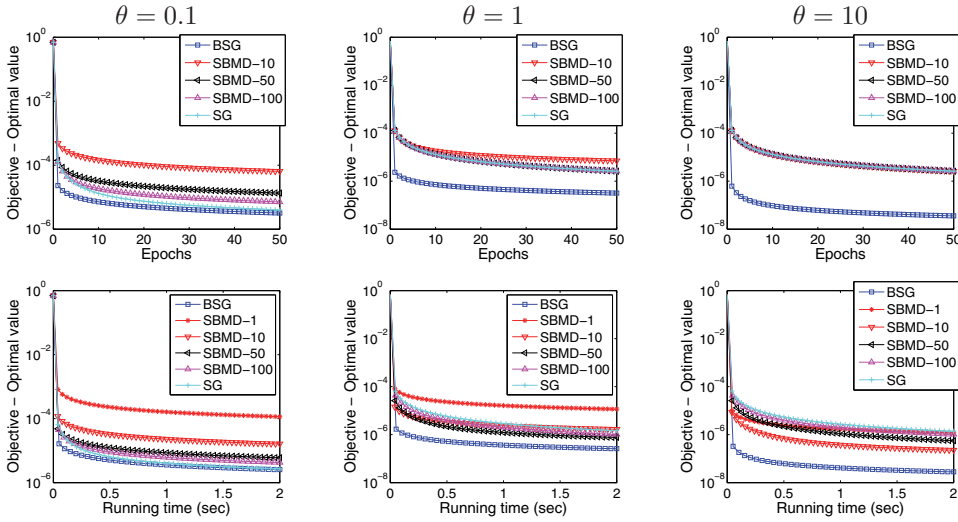
FIG. 1. *Objective values of BSG, SG, and SBMD for solving logistic regression* (3.3) *on Gaussian randomly generated samples.*
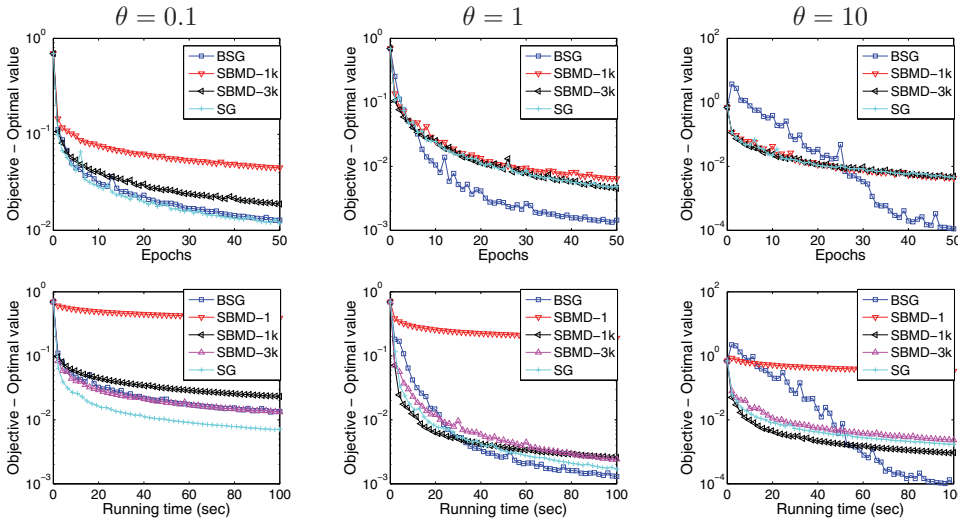

FIG. 2. *Objective values of BSG, SG, and SBMD for solving logistic regression* (3.3) *on the gisette dataset.*

3.4. Low-rank tensor recovery. We compared BSG and BCGD on the problem

$$(3.4) \qquad \min_{\mathbf{X}} \frac{1}{2N} \sum_{\ell=1}^{N} (\mathcal{A}_\ell(\mathbf{X}_1 \circ \mathbf{X}_2 \circ \mathbf{X}_3) - b_\ell)^2 + \sum_{i=1}^{3} \lambda \|\mathbf{X}_i\|_1,$$

where $b_\ell = \mathcal{A}_\ell(\mathcal{M}) = \langle \mathcal{G}_\ell, \mathcal{M} \rangle, \ell = 1, \ldots, N$, with $\mathcal{M}$ being the underlying low-rank tensor. We generated each element of $\mathcal{G}_\ell$ according to the standard Gaussian distribution. In Figure 3, we tested BSG on recovering a low-rank tensor of size $60 \times 60 \times 60$ from $N = 40,000$ Gaussian random measurements by solving (3.4) with
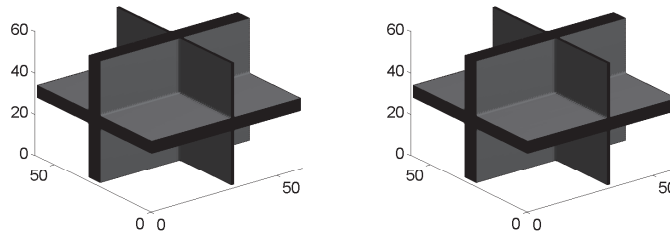
FIG. 3. *Three-dimensional shape of the original $60 \times 60 \times 60$ tensor $\mathcal{M}$ (on the left) and the corresponding recovered one (on the right) by BSG.*
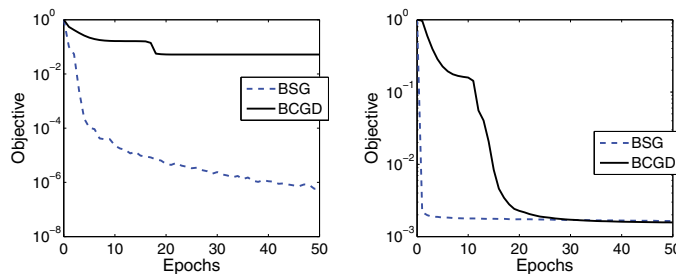

FIG. 4. *Objective values of BSG and BCGD for solving* (3.4) *on a $32 \times 32 \times 32$ low-rank tensor. Left: ℓ_1 penalty parameter $\lambda = 0$; right: $\lambda = \frac{1}{N}$.*

$\lambda = 0$. The original tensor had 10 middle slices of all 1's along each mode and all the other elements were *zero*. For this test, the data size exceeded the memory of our workstation, and thus BCGD could not be tested. Figure 3 plots the original tensor and the recovered one by BSG with 50 epochs and sample size $m_k = 256$ for all k. Its relative error is about 1.93%.

In Figure 4, we compared BSG and BCGD for solving (3.4) on a smaller tensor of size $32 \times 32 \times 32$. It has the same shape as those in Figure 3, and it has 6 middle slices of all 1's along each mode and all other elements were *zero*. We generated $N = 15,000$ Gaussian random measurements. Since BSG and BCGD are both based on block coordinate updates, they have almost the same[3] per-epoch complexity, and thus we only plot their objective values with respect to the number of epochs in Figure 4. The left plot shows the objectives by BSG with $m_k = 64$ for all k and BCGD for solving (3.4) with $\lambda = 0$, and the right plot corresponds to $\lambda = \frac{1}{N}$ and $m_k = 64 + \lceil \frac{k-1}{10} \rceil$ for all k for BSG. From the figure, we see that BSG significantly outperformed BCGD in the beginning. In the smooth case, BCGD was trapped at some local solution, and in the nonsmooth case, BCGD eventually reached a slightly lower objective than that of BSG.

3.5. Bilinear logistic regression. We compared BSG and BCGD on the problem

$$(3.5) \qquad \min_{\mathbf{U},\mathbf{V},b} \frac{1}{N} \sum_{\ell=1}^{N} \log \left(1 + \exp[-y_\ell(\mathrm{tr}(\mathbf{U}^\top \mathbf{X}_\ell \mathbf{V}) + b)] \right),$$

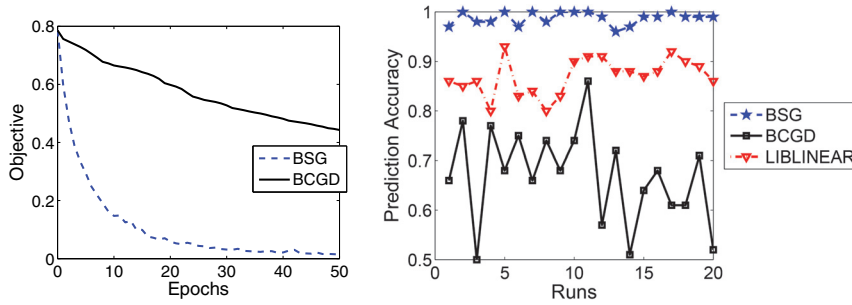[3]BSG has slightly higher complexity because of random sampling.

FIG. 5. *Left: objective values of* (3.5) *(lower is better) given by BSG and BCGD on the BCI EEG data within* 50 *epochs; right: prediction accuracies (higher is better) of* 20 *independent runs by BSG and BCGD within* 30 *epochs and also by LIBLINEAR.*

where $\{(\mathbf{X}_\ell, y_\ell)\}, \ell = 1, \ldots, N$, were given training samples with class labels $y_\ell \in \{+1, -1\}$. The bilinear logistic regression appears to be first used in [10] for EEG data classification. It applies the matrix format of the original data, in contrast to the standard linear logistic regression which collapses each feature matrix into a vector. It has been shown that the bilinear logistic regression outperforms the standard linear logistic regression in many applications such as brain-computer interface [10] and visual recognition [53].

In this test, we used the EEG dataset IVb from[4] BCI competition III and the dataset concerns motor imagery with uncued classification task. The 118 channel EEG was recorded from a healthy subject sitting in a comfortable chair with arms resting on armrests. Visual cues (letter presentation) were shown for 3.5 seconds, during which the subject performed: left hand, right foot, or tongue. The data were sampled at 100 Hz, and the cues of "left hand" and "right foot" were marked in the training data. We chose all 210 marked data points, and for each data point we randomly subsampled 100 temporal slices independently 10 times to get, in total, 2,100 samples of size 118×100.

We set $m_k = 64$ for all k in (3.1) for BSG and used the same random starting point for BSG and BCGD. The left plot of Figure 5 depicts their convergence behaviors. From the figure, we see that BSG significantly outperformed BCGD within 50 epochs. Running BCGD to more epochs, we observed that BCGD could later reach a similar objective as that of BSG. The right plot of Figure 5 shows the prediction accuracy of the solutions of BSG and BCGD, both of which ran to 30 epochs, and that of the result returned by LIBLINEAR [12], which solved linear logistic regression to its default tolerance. We ran the three methods 20 times. For each run, we randomly chose 2,000 samples for training and the remaining ones for testing. From the figure, we see that the *bilinear* logistic regression problem solved by BSG gave consistently higher prediction accuracies than the *linear* logistic regression problem. The low accuracies given by BCGD were the results of its nonconvergence in 30 epochs, which can be observed from the left plot of Figure 5. Running to more epochs, BCGD will eventually give similar predictions as BSG. However, that will take much more time.

4. Conclusions. We have proposed a BSG method and analyzed its convergence for both convex and nonconvex problems. The method has a convergence rate similar to that of the SG method for convex programming, and its convergence has been

---

[4]http://www.bbci.de/competition/iii/

established in terms of the expected violation of first-order optimality conditions for the nonconvex case. Numerical results demonstrate its clear advantages over SG and a block stochastic mirror descent method on the tested convex problems and over the BCGD method one the tested nonconvex problems.

Appendix A. Proof of some lemmas. We give the proofs of some lemmas in the paper.

A.1. Proof of Lemma 2.1. The result in (2.4) can be shown by

$$
\begin{aligned}
\mathbb{E}\langle \mathbf{u}^k, \boldsymbol{\delta}_i^k \rangle &= \mathbb{E}_{\boldsymbol{\Xi}_{[k-1]}} \left[\mathbb{E}\left[\langle \mathbf{u}^k, \boldsymbol{\delta}_i^k \rangle | \boldsymbol{\Xi}_{[k-1]} \right] \right] \\
&= \mathbb{E}_{\boldsymbol{\Xi}_{[k-1]}} \left[\langle \mathbb{E}[\mathbf{u}^k | \boldsymbol{\Xi}_{[k-1]}], \mathbb{E}[\boldsymbol{\delta}_i^k | \boldsymbol{\Xi}_{[k-1]}] \rangle \right] \\
&\leq \mathbb{E}_{\boldsymbol{\Xi}_{[k-1]}} \left[\left\| \mathbb{E}[\mathbf{u}^k | \boldsymbol{\Xi}_{[k-1]}] \right\| \cdot \left\| \mathbb{E}[\boldsymbol{\delta}_i^k | \boldsymbol{\Xi}_{[k-1]}] \right\| \right] \\
&\leq A(\max_j \alpha_j^k) \mathbb{E}_{\boldsymbol{\Xi}_{[k-1]}} \left[\left\| \mathbb{E}[\mathbf{u}^k | \boldsymbol{\Xi}_{[k-1]}] \right\| \right] \\
&\leq A(\max_j \alpha_j^k) \mathbb{E}\|\mathbf{u}^k\|,
\end{aligned}
$$

where the second equality follows from the conditional independence between $\mathbf{u}^k$ and $\boldsymbol{\delta}_i^k$, and the last inequality follows from Jensen's inequality.

A.2. Proof of Lemma 2.2. We first show the following lemma.

LEMMA A.1. *For any function ψ and positive scalar α, it holds that*

$$
\text{(A.1)} \qquad \left\| \frac{1}{\alpha} \big(\mathbf{x} - \mathbf{prox}_{\alpha\psi}(\mathbf{x} - \alpha\mathbf{y}) \big) \right\|^2 \leq 2\|\mathbf{y}\|^2 + 2\|\partial\psi(\mathbf{prox}_{\alpha\psi}(\mathbf{x} - \alpha\mathbf{y}))\|^2,
$$

where $\|\partial\psi(\mathbf{z})\| = \sup_{\mathbf{u}}\{\|\mathbf{u}\| : \mathbf{u} \in \partial\psi(\mathbf{z})\}$, and if $\partial\psi(\mathbf{z}) = \emptyset$, we let $\|\partial\psi(\mathbf{z})\| = +\infty$ by convention.

Proof. Let $\mathbf{z} = \mathbf{prox}_{\alpha\psi}(\mathbf{x} - \alpha\mathbf{y})$. Then $\mathbf{0} \in \mathbf{y} + \frac{1}{\alpha}(\mathbf{z} - \mathbf{x}) + \partial\psi(\mathbf{z})$, namely, for some $\mathbf{u} \in \partial\psi(\mathbf{z})$, it holds $\frac{1}{\alpha}(\mathbf{x} - \mathbf{z}) = \mathbf{y} + \mathbf{u}$. Hence,

$$
\left\| \frac{1}{\alpha}(\mathbf{x} - \mathbf{z}) \right\|^2 \leq 2\|\mathbf{y}\|^2 + 2\|\mathbf{u}\|^2,
$$

which completes the proof. □

Now we are ready to prove Lemma 2.2. For $i \in \mathcal{I}_1$, we have

$$
\tilde{\mathbf{h}}_i^k = \frac{1}{\alpha_i^k} \left(\mathbf{x}_i^k - \mathbf{prox}_{\alpha_i^k r_i}(\mathbf{x}_i^k - \alpha_i^k \tilde{\mathbf{g}}_i^k) \right),
$$

and thus, from Lemma A.1 and Remark 3, it follows that

$$
\mathbb{E}\|\tilde{\mathbf{h}}_i^k\|^2 \leq 2\mathbb{E}\|\tilde{\mathbf{g}}_i^k\|^2 + 2L_{r_i}^2 \leq 4\mathbb{E}\|\mathbf{g}_i^k\|^2 + 4\mathbb{E}\|\boldsymbol{\delta}_i^k\|^2 + 2L_{r_i}^2 \leq 4M_\rho^2 + 4\sigma_k^2 + 2L_{r_i}^2,
$$

where we have used the Cauchy–Schwarz inequality in the second inequality. For $i \in \mathcal{I}_2$, we have

$$
\begin{aligned}
\mathbb{E}\|\tilde{\mathbf{h}}_i^k\|^2 &= \frac{1}{(\alpha_i^k)^2} \mathbb{E} \left\| \mathbf{x}_i^k - \mathcal{P}_{\mathcal{X}_i}\big(\mathbf{x}_i^k - \alpha_i^k(\tilde{\mathbf{g}}_i^k + \tilde{\nabla}r_i(\mathbf{x}_i^k))\big) \right\|^2 \\
&\leq \mathbb{E}\|\tilde{\mathbf{g}}_i^k + \tilde{\nabla}r_i(\mathbf{x}_i^k)\|^2 \leq 4M_\rho^2 + 4\sigma_k^2 + 2L_{r_i}^2,
\end{aligned}
$$

where we have used the nonexpansiveness of the projection operator in the first inequality. This completes the proof of Lemma 2.2. □

A.3. Proof of Lemma 2.3. The result can be proved by induction. First, consider the case of $a \leq 1$. When $k = 1$, (2.6) obviously holds. Suppose it holds for some $k \geq 1$. Then

$$
\begin{aligned}
A_{k+1} - \frac{c}{k+1} &\leq \left(1 - \frac{a}{k}\right)\frac{c}{k} + \frac{b}{k^2} - \frac{c}{k+1} \\
&= \frac{1}{k^2}\left(b - \left(a + \frac{k}{k+1}\right)c\right) \\
&\leq \frac{1}{k^2}\left(b - \left(a + \frac{k}{k+1}\right)\frac{b}{a}\right) < 0,
\end{aligned}
$$

which shows $A_{k+1} \leq c/(k+1)$.

Second, consider $a > 1$. When $k = \lfloor a \rfloor + 1$, it holds from

$$
A_{\lfloor a \rfloor + 1} \leq \left(1 - \frac{a}{\lfloor a \rfloor}\right)A_{\lfloor a \rfloor} + \frac{b}{(\lfloor a \rfloor)^2} \leq \frac{b}{(\lfloor a \rfloor)^2} \leq \frac{2b}{(\lfloor a \rfloor + 1)(a - 1)}.
$$

Suppose (2.6) holds for some $k \geq \lfloor a \rfloor + 1$. Then

$$
\begin{aligned}
A_{k+1} - \frac{c}{k+1} &\leq \left(1 - \frac{a}{k}\right)\frac{c}{k} + \frac{b}{k^2} - \frac{c}{k+1} \\
&= \frac{1}{k^2}\left(b - \left(a + \frac{k}{k+1}\right)c\right) \\
&\leq \frac{1}{k^2}\left(b - \left(a + \frac{k}{k+1}\right)\frac{2b}{a-1}\right) < 0,
\end{aligned}
$$

which indicates $A_{k+1} \leq c/(k+1)$ and completes the proof. $\qquad\square$

A.4. Proof of Lemma 2.7. Suppose $\lim_{k\to\infty} a_{k+1}/a_k = \nu$. We prove the result for the cases of $\nu < \eta$ and $\nu = \eta$, and the case of $\nu > \eta$ can be shown in a similar way as that of $\nu < \eta$.

Case 1. $\nu < \eta$. Since $\lim_{k\to\infty} a_{k+1}/a_k = \nu$, for $\epsilon = \frac{\eta - \nu}{2} > 0$, there exists a sufficiently large integer $K_1 > 0$ such that $a_{k+1}/a_k \leq \nu + \epsilon = \frac{\nu + \eta}{2}$ for all $k \geq K_1$. Therefore, $a_k \leq \left(\frac{\nu + \eta}{2}\right)^{k - K_1} a_{K_1}$ for all $k \geq K_1$. Since $\nu < \eta$, we can choose another sufficiently large integer $K \geq K_1$ to have $a_k \leq \eta^k$ for all $k \geq K$. Hence, $e_k = \eta^k$ for all $k \geq K$, and the limit of e_{k+1}/e_k is η.

Case 2. $\nu = \eta$. If $a_k \geq \eta^k$ for all k, or $a_k \leq \eta^k$ for all k, then the result is obvious. Otherwise, there must exist a sequence $\{n_\ell\}_{\ell=1}^\infty$ such that

$$
\begin{cases}
a_k \geq \eta^k & \text{if } n_{2m-1} \leq k < n_{2m}, \\
a_k < \eta^k & \text{if } n_{2m} \leq k < n_{2m+1}.
\end{cases}
$$

Note that if $n_{2m-1} \leq k < k+1 < n_{2m}$ or $n_{2m} \leq k < k+1 < n_{2m+1}$, then it is easy to have $e_{k+1}/e_k \to \eta$ as $k \to \infty$. In addition,

$$
\frac{e_{n_{2m}}}{e_{n_{2m-1}}} = \frac{\eta^{n_{2m}}}{a_{n_{2m-1}}} \leq \frac{\eta^{n_{2m}}}{\eta^{n_{2m-1}}} = \eta
$$

and

$$
\frac{e_{n_{2m}}}{e_{n_{2m-1}}} = \frac{\eta^{n_{2m}}}{a_{n_{2m-1}}} \geq \frac{a_{n_{2m}}}{a_{n_{2m-1}}} \to \eta \text{ as } m \to \infty.
$$

Hence, $\lim_{m\to\infty} e_{n_{2m}}/e_{n_{2m-1}} = \eta$ and, in the same way, one can show $\lim_{m\to\infty} e_{n_{2m+1}}/e_{n_{2m+1}-1} = \eta$. Therefore, the limit of e_{k+1}/e_k is η. This completes the proof. $\qquad\square$

REFERENCES

[1] A. Auslender, *Asymptotic properties of the Fenchel dual functional and applications to decomposition problems*, J. Optim. Theory Appl., 73 (1992), pp. 427–449.

[2] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[3] A. Beck and L. Tetruashvili, *On the convergence of block coordinate descent type methods*, SIAM J. Optim., 23 (2013), pp. 2037–2060.

[4] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1999.

[5] S. Boyd, L. Xiao, and A. Mutapcic, *Subgradient Methods*, Lecture Notes of EE392o, Autumn Quarter, 2004, Stanford University, Palo Alto, CA, 2003.

[6] E. J. Candes and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717–772.

[7] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, *Coordinate descent method for large-scale L2-loss linear support vector machines*, J. Mach. Learn. Res., 9 (2008), pp. 1369–1398.

[8] K.-L. Chung, *On a stochastic approximation method*, Ann. Math. Statist., 25 (1954), pp. 463–483.

[9] C. D. Dang and G. Lan, *Stochastic block mirror descent methods for nonsmooth and stochastic optimization*, SIAM J. Optim., 25 (2015), pp. 856–881.

[10] M. Dyrholm, C. Christoforou, and L. C. Parra, *Bilinear discriminant component analysis*, J. Mach. Learn. Res., 8 (2007), pp. 1097–1111.

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

[12] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, *Liblinear: A library for large linear classification*, J. Mach. Learn. Res., 9 (2008), pp. 1871–1874.

[13] M. P. Friedlander and M. Schmidt, *Hybrid deterministic-stochastic methods for data fitting*, SIAM J. Sci. Comput., 34 (2012), pp. A1380–A1405.

[14] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, *Large-scale matrix factorization with distributed stochastic gradient descent*, in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, 2011, pp. 69–77.

[15] S. Ghadimi and G. Lan, *Accelerated gradient methods for nonconvex nonlinear and stochastic programming*, Math. Program. (2015), DOI:10.1007/s10107-015-0871-8.

[16] S. Ghadimi and G. Lan, *Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization,* II: *Shrinking procedures and optimal algorithms*, SIAM J. Optim., 23 (2013), pp. 2061–2089.

[17] S. Ghadimi and G. Lan, *Stochastic first and zeroth-order methods for nonconvex stochastic programming*, SIAM J. Optim., 23 (2013), pp. 2341–2368.

[18] S. Ghadimi, G. Lan, and H. Zhang, *Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization*, Math. Program. (2015), DOI:10.1007/s10107-014-0846-1.

[19] J. L. Goffin, *On convergence rates of subgradient optimization methods*, Math. Program., 13 (1977), pp. 329–347.

[20] L. Grippo and M. Sciandrone, *On the convergence of the block nonlinear Gauss-Seidel method under convex constraints*, Oper. Res. Lett., 26 (2000), pp. 127–136.

[21] L. Grippo and M. Sciandrone, *On the convergence of the block nonlinear Gauss-Seidel method under convex constraints*, Oper. Res. Lett., 26 (2000), pp. 127–136.

[22] C. Hildreth, *A quadratic programming procedure*, Naval Res. Logist. Q., 4 (1957), pp. 79–85.

[23] M. Hong, X. Wang, M. Razaviyayn, and Z.-Q. Luo, *Iteration Complexity Analysis of Block Coordinate Descent Methods*, preprint, arXiv:1310.6957, 2013.

[24] A. J. Kleywegt, A. Shapiro, and T. Homem-de-Mello, *The sample average approximation method for stochastic discrete optimization*, SIAM J. Optim., 12 (2002), pp. 479–502.

[25] G. Lan, *An optimal method for stochastic composite optimization*, Math. Program., 133 (2012), pp. 365–397.

[26] J. Liu, S. J. Wright, and S. Sridhar, *An Asynchronous Parallel Randomized Kaczmarz Algorithm*, preprint, arXiv:1401.4780, 2014.

[27] Z. Lu and L. Xiao, *On the complexity analysis of randomized block-coordinate descent methods*, Math. Program., 152 (2015), pp. 615–642.

[28] Z. Lu and L. Xiao, *Randomized Block Coordinate Non-Monotone Gradient Method for a Class of Nonlinear Programming*, preprint, arXiv:1306.5918, 2013.

[29] Z.-Q. Luo and P. Tseng, *On the convergence of the coordinate descent method for convex differentiable minimization*, J. Optim. Theory Appl., 72 (1992), pp. 7–35.

[30] J. Mairal, *Stochastic majorization-minimization algorithms for large-scale optimization*, in Advances in Neural Information Processing Systems 26, 2013.

[31] J. MAIRAL, F. BACH, J. PONCE, AND G. SAPIRO, *Online dictionary learning for sparse coding*, in Proceedings of the 26th Annual International Conference on Machine Learning, ACM, New York, 2009, pp. 689–696.

[32] C. NAVASCA, L. DE LATHAUWER, AND S. KINDERMANN, *Swamp reducing technique for tensor decomposition*, in Proceedings of the 16th European Signal Processing Conference (EU-SIPCO 2008), IEEE, Piscataway, NJ, 2008.

[33] A. NEMIROVSKI, A. JUDITSKY, G. LAN, AND A. SHAPIRO, *Robust stochastic approximation approach to stochastic programming*, SIAM J. Optim., 19 (2009), pp. 1574–1609.

[34] A. NEMIROVSKI AND D. B. YUDIN, *Problem Complexity and Method Efficiency in Optimization*, Wiley, Chichester, 1983.

[35] Y. NESTEROV, *Introductory Lectures on Convex Optimization*, Appl. Optim. 87, 2004.

[36] Y. NESTEROV, *Primal-dual subgradient methods for convex problems*, Math. Program., 120 (2009), pp. 221–259.

[37] Y. NESTEROV, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM J. Optim., 22 (2012), pp. 341–362.

[38] Y. NESTEROV AND V. SHIKHMAN, *Convergent Subgradient Methods for Nonsmooth Convex Minimization*, Technical report, Université Catholique de Louvain, Center for Operations Research and Econometrics (CORE), Louvain-la-neuve, Belgium, 2014.

[39] Z. PENG, M. YAN, AND W. YIN, *Parallel and distributed sparse optimization*, in Proceedings of the 2013 Asilomar Conference on Signals, Systems, and Computers, IEEE, Piscataway, NJ, 2013, pp. 659–646.

[40] B. T. POLYAK, *New stochastic approximation type procedures*, Avtomat. i Telemekh., 51 (1990), pp. 98–107.

[41] M. RAZAVIYAYN, M. HONG, AND Z.-Q. LUO, *A unified convergence analysis of block successive minimization methods for nonsmooth optimization*, SIAM J. Optim., 23 (2013), pp. 1126–1153.

[42] B. RECHT AND C. RÉ, *Parallel stochastic gradient algorithms for large-scale matrix completion*, Math. Program. Comput., 5 (2013), pp. 201–226.

[43] P. RICHTÁRIK AND M. TAKÁČ, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Math. Program., 144 (2014), pp. 1–38.

[44] H. ROBBINS AND S. MONRO, *A stochastic approximation method*, Ann. Math. Statist., 22 (1951), pp. 400–407.

[45] R. T. ROCKAFELLAR, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[46] R. T. ROCKAFELLAR AND R. J. B. WETS, *Variational Analysis*, Grundkhren Math. Wiss. 317, Springer-Verlag, Berlin, 1998.

[47] J. SACKS, *Asymptotic distribution of stochastic approximation procedures*, Ann. Math. Statist., 29 (1958), pp. 373–405.

[48] A. SAHA AND A. TEWARI, *On the nonasymptotic convergence of cyclic coordinate descent methods*, SIAM J. Optim., 23 (2013), pp. 576–601.

[49] T. SCHAUL, S. ZHANG, AND Y. LECUN, *No more pesky learning rates*, in Proceedings of the 30th International Conference on Machine Learning (ICML-13), ACM, New York, 2013, pp. 343–351.

[50] S. SHALEV-SHWARTZ, Y. SINGER, N. SREBRO, AND A. COTTER, *Pegasos: Primal estimated sub-gradient solver for SVM*, Math. Program., 127 (2011), pp. 3–30.

[51] S. SHALEV-SHWARTZ AND A. TEWARI, *Stochastic methods for l1 regularized loss minimization*, in ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, New York, 2009, pp. 929–936.

[52] S. K. SHEVADE AND S. S. KEERTHI, *A simple and efficient algorithm for gene selection using sparse logistic regression*, Bioinformatics, 19 (2003), pp. 2246–2253.

[53] J. V. SHI, Y. XU, AND R. G. BARANIUK, *Sparse Bilinear Logistic Regression*, preprint, arXiv:1404.4104, 2014.

[54] R. TIBSHIRANI, *Regression shrinkage and selection via the lasso*, J. R. Stat. Soc. Ser. B Methodol., 58 (1996), pp. 267–288.

[55] P. TSENG, *Convergence of a block coordinate descent method for nondifferentiable minimization*, J. Optim. Theory Appl., 109 (2001), pp. 475–494.

[56] P. TSENG AND S. YUN, *A coordinate gradient descent method for nonsmooth separable minimization*, Math. Program., 117 (2009), pp. 387–423.

[57] Z. WEN, D. GOLDFARB, AND K. SCHEINBERG, *Block coordinate descent methods for semidefinite programming*, in Handbook on Semidefinite, Conic and Polynomial Optimization, Springer, New York, 2012, pp. 533–564.

[58] Y. XU AND W. YIN, *A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion*, SIAM J. Imaging Sci., 6 (2013), pp. 1758–1789.

[59] Y. XU AND W. YIN, *A Globally Convergent Algorithm for Nonconvex Optimization Based on Block Coordinate Update*, preprint, arXiv:1410.1386, 2014.

[60] T. ZHANG, *Solving large scale linear prediction problems using stochastic gradient descent algorithms*, in Proceedings of the Twenty-first International Conference on Machine Learning, ACM, New York, 2004, 116.