

ACCELERATED STOCHASTIC ALGORITHMS FOR NONCONVEX
FINITE-SUM AND MULTIBLOCK OPTIMIZATION*GUANGHUI LAN[†] AND YU YANG[†]

Abstract. In this paper, we present new stochastic methods for solving two important classes of nonconvex optimization problems. We first introduce a randomized accelerated proximal gradient (RapGrad) method for solving a class of nonconvex optimization problems whose objective function consists of the summation of m components and show that it can significantly reduce the number of gradient computations especially when the condition number L/μ (i.e., the ratio between the Lipschitz constant and negative curvature) is large. More specifically, RapGrad can save up to $\mathcal{O}(\sqrt{m})$ gradient computations more than existing batch nonconvex accelerated gradient methods. Moreover, the number of gradient computations required by RapGrad can be $\mathcal{O}(m^{\frac{1}{6}}L^{\frac{1}{2}}/\mu^{\frac{1}{2}})$ (at least $\mathcal{O}(m^{\frac{2}{3}})$) times smaller than the best-known randomized nonconvex gradient methods when $L/\mu \geq m$. Inspired by RapGrad, we also develop a new randomized accelerated proximal dual (RapDual) method for solving a class of multiblock nonconvex optimization problems coupled with linear constraints and some special structural properties. We demonstrate that RapDual can also save up to a factor of $\mathcal{O}(\sqrt{m})$ block updates more than its batch counterpart, where m denotes the number of blocks. To the best of our knowledge, all these complexity results associated with RapGrad and RapDual seem to be new in the literature. We also illustrate potential advantages of these algorithms through our preliminary numerical experiments.

Key words. nonconvex optimization, stochastic algorithms, acceleration, finite-sum optimization, multiblock optimization

AMS subject classifications. 90C25, 90C06, 90C22, 49M37

DOI. 10.1137/18M1192536

1. Introduction. Nonconvex optimization plays a fundamental role in modern statistics and machine learning, e.g., for empirical risk minimization with either nonconvex loss [28] or regularization [7, 31, 32], as well as the training of deep neural networks [12]. In this paper, we consider two classes of nonconvex optimization problems that are widely used in statistical learning. The first class of problems intends to minimize the summation of many terms:

$$(1.1) \quad \min_{x \in X} \{f(x) := \frac{1}{m} \sum_{i=1}^m f_i(x)\},$$

where $X \subseteq \mathbb{R}^n$ is a closed convex set, and $f_i : X \rightarrow \mathbb{R}$, $i = 1, \dots, m$, are nonconvex smooth functions with L -Lipschitz continuous gradients over X , i.e., for some $L \geq 0$,

$$(1.2) \quad \|\nabla f_i(x_1) - \nabla f_i(x_2)\| \leq L\|x_1 - x_2\| \quad \forall x_1, x_2 \in X.$$

Moreover, we assume that there exists $0 < \mu \leq L$ such that (s.t.)

$$(1.3) \quad f_i(x_1) - f_i(x_2) - \langle \nabla f_i(x_2), x_1 - x_2 \rangle \geq -\frac{\mu}{2}\|x_1 - x_2\|^2 \quad \forall x_1, x_2 \in X.$$

Clearly, (1.2) implies (1.3) (with $\mu = L$). While in the classical nonlinear programming setting one only assumes (1.2), by using both conditions (1.2) and (1.3) we can explore

* Received by the editors June 6, 2018; accepted for publication (in revised form) August 13, 2019; published electronically October 31, 2019.

<https://doi.org/10.1137/18M1192536>

Funding: This work was funded by National Science Foundation grants CMMI-1637473, CMMI-1637474 and Army Research Office W911NF-18-1-0223.

[†]H. Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332 (george.lan@isye.gatech.edu, yangyu@gatech.edu).

more structural information for the design of solution methods of problem (1.1). In particular, we intend to develop more efficient algorithms to solve problems where the condition number L/μ associated with problem (1.1) is large. As an example, consider the nonconvex composite problem arising from variable selection in statistics [7, 9]: $f(x) = \frac{1}{m} \sum_{i=1}^m h_i(x) + \rho p(x)$, where h_i 's are smooth convex functions, p is a nonconvex function, and $\rho > 0$ is a relatively small penalty parameter. Note that some examples of the nonconvex penalties are given by minimax concave penalty (MCP) or smoothly clipped absolute deviation (SCAD) (see [7]). It can be shown that the condition number for these problems is usually larger than m (see section 4 for more details).

In addition to (1.1), we consider an important class of nonconvex multiblock optimization problems with linearly coupled constraints, i.e.,

$$(1.4) \quad \begin{aligned} & \min_{x_i \in X_i} \sum_{i=1}^m f_i(x_i) \\ & \text{s.t. } \sum_{i=1}^m A_i x_i = b. \end{aligned}$$

Here $X_i \subseteq \mathbb{R}^{d_i}$ are closed convex sets, $A_i \subseteq \mathbb{R}^{n \times d_i}$, $b \subseteq \mathbb{R}^n$, $f_i : X_i \rightarrow \mathbb{R}$ satisfy, for some $\mu \geq 0$,

$$(1.5) \quad f_i(x) - f_i(y) - \langle \nabla f_i(y), x - y \rangle \geq -\frac{\mu}{2} \|x - y\|^2 \quad \forall x, y \in X_i,$$

and $f_m : \mathbb{R}^n \rightarrow \mathbb{R}$ has L -Lipschitz continuous gradients, i.e., $\exists L \geq 0$ s.t.

$$(1.6) \quad \|\nabla f_m(x) - \nabla f_m(y)\| \leq L \|x - y\| \quad \forall x, y \in \mathbb{R}^n.$$

Moreover, we assume that $X_m = \mathbb{R}^n$ and A_m is invertible. In other words, throughout this paper we make the structural assumption that one of the blocks equals the dimension of the variable. Problems of this type arise naturally in compressed sensing and distributed optimization. For instance, consider the compressed sensing problem via nonconvex shrinkage penalties: $\min_{x_i \in X_i} \{p(x) : Ax = b\}$, where $A \in \mathbb{R}^{n \times d}$ is a big sensing matrix with $d \gg n$, and $p(x) = \sum_{i=1}^m p_i(x_i)$ is a nonconvex and separable penalty function. Since it is easy to find an invertible submatrix in A , without loss of generality, we assume that the last n columns of A forms an invertible matrix. We can then view this problem as a special case of (1.4) by grouping the last n components of x into block x_m and dividing the remaining $d - n$ components into another $m - 1$ blocks.

Much recent research effort has been directed to efficient solution algorithms for the aforementioned nonconvex finite-sum or multiblock problems. Let us start with reviewing a few complexity results associated with existing first-order methods for solving the finite-sum problem (1.1). For simplicity, let us assume that $X = \mathbb{R}^n$ for now. It is well-known (see, e.g., [23]) that the simple gradient descent (GD) method applied to problem (1.1) requires $\mathcal{O}(L/\epsilon)$ iterations to find an ϵ -stationary solution, i.e., a point \bar{x} s.t. $\|\nabla f(\bar{x})\|^2 \leq \epsilon$. Since each GD iteration requires a full gradient computation, i.e., m gradient computations for f_i 's, totally this algorithm needs $\mathcal{O}(mL/\epsilon)$ gradient computations for all the component functions f_i 's. Ghadimi and Lan [8] (see also [11]) show that by using the stochastic gradient descent (SGD) method, one only needs to compute the gradient of one randomly selected component function at each iteration, resulting in totally $\mathcal{O}(L\sigma^2/\epsilon^2)$ gradient computations to find a stochastic ϵ -stationary solution of (1.1), i.e., a point \bar{x} s.t. $\mathbb{E}[\|\nabla f(\bar{x})\|^2] \leq \epsilon$. Here the expectation is taken w.r.t. some random variables used in the algorithm and σ^2 denotes their variance. Although this complexity bound does not depend on m ,

it has a much worse dependence on ϵ than the GD method for solving problem (1.1). Inspired by the variance reduction techniques originated in convex optimization [15], Reddi and co-authors [25, 26] and Allen-Zhu and Hazan [2] recently show that one only needs $\mathcal{O}(m^{2/3}L/\epsilon)$ gradient evaluations to find an ϵ -stationary point of (1.1), which significantly improves the bound in [8] in terms of the dependence on ϵ and also dominates the one for GD by a factor of $m^{1/3}$. However, it remains unknown whether one can further improve this bound in terms of its dependence on m for nonconvex finite-sum optimization especially when L/μ is large.

A different line of research aims to incorporate Nesterov's acceleration (momentum) [22] into nonconvex optimization. Ghaidmi and Lan [9] first established the convergence of the accelerated gradient (AG) method for nonconvex optimization and show that it can improve the complexity of GD if the problem has a large condition number (i.e., L/μ is large). Their results were further improved in [10, 3, 24] and [16]. Currently the best complexity result, in terms of total gradient computations, for these methods is given by $\mathcal{O}(m\sqrt{L\mu}/\epsilon)$ for unconstrained problems [16]. However, it remains unknown if the complexity of such accelerated algorithms can be further improved in terms of the dependence on m , especially when one needs to maintain the $\mathcal{O}(1/\epsilon)$ complexity bound on gradient computations. Note that some nonconvex stochastic AG methods have been discussed in [9] but they all exhibit worse $\mathcal{O}(L\sigma^2/\epsilon^2)$ complexity bounds.

While stochastic and randomized methods are being intensively explored for solving problem (1.1), most existing studies for the nonconvex multiblock problem in (1.4) have been mainly focused on deterministic batch methods only. Many of these studies aim at the generalization of the alternating direction method of multipliers (ADMM) method for nonconvex optimization. For example, in [13], Hong established the complexity for a variant of ADMM for nonconvex multiblock problems; see also [14] and [29] for some previous work on the asymptotic analysis of ADMM for nonconvex optimization. In [21], Melo and Monteiro presented a linearized proximal multiblock ADMM with complexity $\mathcal{O}(1/\epsilon)$ to attain a nearly feasible ϵ -stationary solution, but all the blocks have to be updated in each iteration. Later, they proposed a Jacobi-type ADMM in [20] with a similar complexity bound, which shows benefits if parallel computing is available. While the idea of randomly selecting blocks in nonconvex ADMM has been explored recently, these studies focus on the asymptotical convergence of these schemes (e.g., in [14, 30]). To the best of our knowledge, there does not exist any complexity analysis regarding randomized methods for solving the nonconvex multiblock problem in (1.4) in the literature and as a consequence, it remains unclear whether stochastic or randomized methods are more advantageous over batch ones or not.

Our contribution in this paper mainly consists of the following several aspects. First, we develop a new randomized algorithm, namely, the randomized accelerated proximal gradient (RapGrad) method for solving problem (1.1) and show that it can significantly improve the complexity of existing algorithms especially for problems with a large condition number. More specifically, we show that RapGrad requires totally $\mathcal{O}(\mu(m + \sqrt{mL}/\mu)/\epsilon)$ gradient computations in order to find a stochastic ϵ -stationary point. For problems with $L/\mu \geq m$, this bound reduces to $\mathcal{O}(\sqrt{mL}/\mu/\epsilon)$, which dominates the best-known batch AG methods by a factor of \sqrt{m} [16] and outperforms those variance-reduced stochastic algorithms [25, 26, 2] by a factor of $m^{1/6}L^{1/2}/\mu^{1/2}$ (at least $m^{2/3}$). In fact, our complexity bound will be better than the latter algorithms as long as $L/\mu \log(L/\mu) > m^{1/3}$. Therefore, we provide some affirmative

answers regarding whether the complexity bounds of variance reduced algorithms and AG methods for nonconvex optimization can be further improved, especially in terms of their dependence on m . To the best of our knowledge, all these complexity results seem to be new in the literature for nonconvex finite-sum optimization. It is worth noting that some improvement over variance-reduced stochastic algorithms under the region $m \geq L/\mu$ (i.e., L/μ is small) has been presented recently in [1]. RapGrad is a proximal-point type method which iteratively transforms the original nonconvex problem into a series of convex subproblems. In RapGrad, we incorporate a modified optimal randomized incremental gradient method, namely, the randomized primal-dual gradient (see [18]) to solve these convex subproblems, and as a consequence, each iteration of RapGrad requires gradient computation for only one randomly selected component function. In comparison with existing nonconvex proximal-point type methods, the design and analysis of RapGrad appear to be more complicated. In particular, RapGrad does not require the computation of full gradients throughout its entire procedure by properly initializing a few intertwined search points and gradients using information obtained from the previous subproblems. This comes with the price of requiring additional storage (memory) for maintaining $\mathcal{O}(m)$ variables (e.g., \underline{x}^t). Moreover, the analysis of RapGrad requires us to show the convergence for some auxiliary sequences where the gradients are computed, which has not been established for the original randomized primal-dual gradient method.

Second, inspired by RapGrad, we develop a new randomized proximal-point type method, namely, the randomized accelerated proximal dual (RapDual) method, for solving the nonconvex multiblock problem in (1.4). Similarly to RapGrad, this method solves a series of strongly convex subproblems iteratively generated by adding strongly convex terms, via a novel randomized dual method developed in this paper for solving linearly constrained problems. Each iteration of RapDual requires access to only one randomly selected block and the solution of a relatively easy primal block updating operator. Note that in order to guarantee the strong concavity of the Lagrangian dual of the subproblem, we need to assume that $X_m = \mathbb{R}^n$ and the last block A_m is invertible. Moreover, we assume that it is relatively easy to compute A_m^{-1} (e.g., A_m is the identity matrix, sparse or symmetric diagonally dominant) to simplify the statement and analysis of the algorithm (see Remark 3.3 for more discussions). Let us consider for now the case when $X_i \equiv \mathbb{R}^{d_i}$ and $A_m = I$. We can show that RapDual can find a solution $(\bar{x}_1, \dots, \bar{x}_m)$ s.t. $\exists \lambda \in \mathbb{R}^n$, $\mathbb{E}[\sum_{i=1}^m \|\nabla f(\bar{x}_i) + A_i^\top \lambda\|^2] \leq \epsilon$ and $\mathbb{E}[\|\sum_{i=1}^m A_i \bar{x}_i - b\|^2] \leq \sigma$ in at most

$$N(\epsilon, \sigma) := \mathcal{O}\left(m\bar{A}\sqrt{L\mu}\log\left(\frac{L}{\mu}\right) \cdot \max\left\{\frac{1}{\epsilon}, \frac{\|A\|^2}{\sigma L^2}\right\} \mathcal{D}^0\right)$$

primal block updates, where $\bar{A} = \max_{i \in [m-1]} \|A_i\|$, $\|A\|^2 = \sum_{i=1}^{m-1} \|A_i\|^2$, and $\mathcal{D}^0 := \sum_{i=1}^m [f_i(\bar{x}_i^0) - f_i(x_i^*)]$. Moreover, we demonstrate that the total number primal block updates that RapGrad requires can be much smaller, up to a factor of $\mathcal{O}(\sqrt{m})$, than its batch counterpart. To the best of our knowledge, this is the first time that the complexity of randomized methods for solving this special class of nonconvex multiblock optimization has been established and their possible advantages over batch methods are quantified in the literature.

Third, we perform some numerical experiments on both RapGrad and RapDual for solving nonconvex finite-sum and multiblock problems in (1.1) and (1.4) and demonstrate their potential advantages over some existing algorithms.

This paper is organized as follows. In section 2, we present our algorithm RapGrad and its convergence properties for solving the nonconvex finite-sum problem in (1.1).

RapDual for nonconvex finite-sum optimization with linear constraints (1.4) and its convergence analysis are included in section 3. Section 4 is devoted to some numerical experiments of our algorithms for the above two types of problems. Finally some concluding remarks are made in section 5.

1.1. Notation and terminology. Let \mathbb{R} denote the set of real numbers. All vectors are viewed as column vectors, and for a vector $x \in \mathbb{R}^d$, we use x^\top to denote its transpose. For any $n \geq 1$, the set of integers $\{1, \dots, n\}$ is denoted by $[n]$. We use $\mathbb{E}_s[X]$ to denote the expectation of a random variable X on i_1, \dots, i_s . For a given strongly convex function ω , we define the prox-function associated with ω as

$$V_\omega(x, y) := \omega(x) - \omega(y) - \langle \omega'(y), x - y \rangle \quad \forall x, y \in X,$$

where $\omega'(y) \in \partial\omega(y)$ is an arbitrary subgradient of ω at y . For any $s \in \mathbb{R}$, $\lceil s \rceil$ denotes the nearest integer to s from above.

2. Nonconvex finite-sum optimization. In this section, we develop a RapGrad method for solving the nonconvex finite-sum optimization problem in (1.1) and demonstrate that it can significantly improve the existing rates of convergence for solving these problems, especially when their objective functions have large condition numbers. We will describe this algorithm and establish its convergence in subsections 2.1 and 2.2, respectively.

2.1. The algorithm. The basic idea of RapGrad is to solve problem (1.1) iteratively by using the proximal-point type method. More specifically, given a current search point $\bar{x}^{\ell-1}$ at the ℓ th iteration, we will employ a randomized accelerated gradient (RaGrad) obtained by properly modifying the randomized primal-dual gradient method in [18], to approximately solve

$$(2.1) \quad \min_{x \in X} \frac{1}{m} \sum_{i=1}^m f_i(x) + \frac{3\mu}{2} \|x - \bar{x}^{\ell-1}\|^2$$

to compute a new search point \bar{x}^ℓ . A similar idea can be found in [19], where a proximal-point method and randomized method are combined to minimize a convex objective.

The algorithmic schemes for RapGrad and RaGrad are described in Algorithms 1 and 2, respectively. While it seems that we can directly apply the randomized primal-dual gradient method in [18] (or another fast randomized incremental gradient method) to solve (2.1) since it is strongly convex due to (1.3), a direct application of these methods would require us to compute the full gradient from time to time whenever a new subproblem needs to be solved. In fact, if one applies a variance reduced incremental gradient method to solve (2.1), the algorithmic scheme would involve three loops (or epochs) and each epoch requires a full gradient computation. Moreover, a direct application of these existing first-order methods to solve (2.1) would result in some extra logarithmic factor ($\log(1/\epsilon)$) in the final complexity bound as shown in [3]. Therefore, we employed the RaGrad method to solve (2.1), which differs from the original randomized primal-dual gradient method in the following several aspects. First, different from the randomized primal-dual gradient method, the design and analysis of RaGrad does not involve the conjugate functions of f_i 's, but only first-order information (function values and gradients). Such an analysis enables us to build a relation between successive search points \bar{x}^ℓ , as well as the convergence of the sequences \bar{x}_i^ℓ where the gradients \bar{y}_i^ℓ are computed. With these relations at hand, we can determine the number of iterations s required by Algorithm 2 to ensure the overall RapGrad Algorithm to achieve an accelerated rate of convergence.

Algorithm 1 RapGrad for nonconvex finite-sum optimization.

Let $\bar{x}^0 \in X$, and set $\underline{x}_i^0 = \bar{x}^0$, $\bar{y}_i^0 = \nabla f_i(\bar{x}^0)$, $i = 1, \dots, m$.
for $\ell = 1, \dots, k$ **do**
 Set $x^{-1} = x^0 = \bar{x}^{\ell-1}$, $\underline{x}_i^0 = \bar{x}_i^{\ell-1}$, and $y_i^0 = \bar{y}_i^{\ell-1}$, $i = 1, \dots, m$.
 Run RaGrad (cf. Algorithm 2) with input x^{-1} , x^0 , \underline{x}_i^0 , y_i^0 , $i = 1, \dots, m$, and s to solve the subproblem

$$(2.2) \quad \min_{x \in X} \frac{1}{m} \sum_{i=1}^m \psi_i(x) + \varphi(x)$$

to obtain output x^s , \underline{x}_i^s , y_i^s , $i = 1, \dots, m$, where $\psi_i(x) \equiv \psi_i^\ell(x) := f_i(x) + \mu \|x - \bar{x}^{\ell-1}\|^2$, $i = 1, \dots, m$, and $\varphi(x) \equiv \varphi^\ell(x) := \frac{\mu}{2} \|x - \bar{x}^{\ell-1}\|^2$.
 Set $\bar{x}^\ell = x^s$, $\underline{x}_i^\ell = \underline{x}_i^s$ and $\bar{y}_i^\ell = y_i^s + 2\mu(\bar{x}^{\ell-1} - \bar{x}^\ell)$, $i = 1, \dots, m$ (note $\bar{y}_i^\ell = \nabla \psi_i^{\ell+1}(\underline{x}_i^\ell)$ always holds).
end for
return $\bar{x}^{\hat{\ell}}$ for some random $\hat{\ell} \in [k]$.

Algorithm 2 RaGrad for iteratively solving subproblem (2.2).

Input $x^{-1} = x^0 \in X$, $\underline{x}_i^0 \in X$, y_i^0 , $i = 1, \dots, m$, number of iterations s . Assume nonnegative parameters $\{\alpha_t\}$, $\{\tau_t\}$, $\{\eta_t\}$ are given.
for $t = 1, \dots, s$ **do**
 1. Generate a random variable i_t uniformly distributed over $[m]$.
 2. Update x^t and y^t according to

$$(2.3) \quad \tilde{x}^t = \alpha_t(x^{t-1} - x^{t-2}) + x^{t-1}.$$

$$(2.4) \quad \underline{x}_i^t = \begin{cases} (1 + \tau_t)^{-1}(\tilde{x}^t + \tau_t \underline{x}_i^{t-1}), & i = i_t, \\ \underline{x}_i^{t-1}, & i \neq i_t, \end{cases}$$

$$(2.5) \quad y_i^t = \begin{cases} \nabla \psi_i(\underline{x}_i^t), & i = i_t, \\ y_i^{t-1}, & i \neq i_t, \end{cases}$$

$$(2.6) \quad \tilde{y}_i^t = m(y_i^t - y_i^{t-1}) + y_i^{t-1} \quad \forall i = 1, \dots, m,$$

$$(2.7) \quad x^t = \arg \min_{x \in X} \varphi(x) + \left\langle \frac{1}{m} \sum_{i=1}^m \tilde{y}_i^t, x \right\rangle + \eta_t V_\varphi(x, x^{t-1}).$$

end for
return x^s , \underline{x}_i^s , and y_i^s , $i = 1, \dots, m$.

Second, the original randomized primal-dual gradient method requires the computation of only one randomly selected gradient at each iteration and does not require the computation of full gradients from time to time. However, it is unclear whether a full pass of all component functions is required whenever we solve a new proximal subproblem (i.e., $\bar{x}^{\ell-1}$ changes at each iteration). It turns out that by properly initializing a few intertwined primal and gradient sequences in RaGrad using information obtained from previous subproblems, we will compute full gradient only once for the very first time when this method is called and do not need to compute full gradients any more when solving all other subproblems throughout the RapGrad method. Indeed, the output y_s^i of RaGrad (Algorithm 2) represents the gradients of ψ_i at the search points \underline{x}_i^s . By using the strong convexity of the objective functions, we will be able to show that all the search points \underline{x}_s^i , $i = 1, \dots, m$, will converge, similarly to the

search point x^s , to the optimal solution of the subproblem in (2.2) (see Lemma 2.4 below). Therefore, we can use y_s^i to approximate $\nabla\psi_i(x^s)$ and thus remove the necessity of computing the full gradient of x^s when solving the next subproblem.

Before establishing the convergence of the RapGrad method, we first need to define an approximate stationary point for problem (1.1). A point $x \in X$ is called an approximate stationary point if it sits within a small neighborhood of a point $\hat{x} \in X$ which approximately satisfies the first-order optimality condition.

DEFINITION 2.1. *A point $x \in X$ is called an (ϵ, δ) -solution of (1.1) if there exists some $\hat{x} \in X$ such that*

$$[d(\nabla f(\hat{x}), -N_X(\hat{x}))]^2 \leq \epsilon \quad \text{and} \quad \|x - \hat{x}\|^2 \leq \delta.$$

A stochastic (ϵ, δ) -solution of (1.1) is one such that

$$\mathbb{E}[d(\nabla f(\hat{x}), -N_X(\hat{x}))]^2 \leq \epsilon \quad \text{and} \quad \mathbb{E}\|x - \hat{x}\|^2 \leq \delta.$$

Here, $d(x, Z) := \inf_{z \in Z} \|x - z\|$ denotes the distance from x to set Z , and $N_X(\hat{x}) := \{x \in \mathbb{R}^n \mid \langle x, y - \hat{x} \rangle \leq 0 \text{ for all } y \in X\}$ denotes the normal cone of X at \hat{x} .

To have a better understanding of the above definition, let us consider the unconstrained problem (1.1), i.e., $X = \mathbb{R}^n$. Suppose that $x \in X$ is an (ϵ, δ) -solution with $\delta = \epsilon/L^2$. Then there exists $\hat{x} \in X$ s.t. $\|\nabla f(\hat{x})\|^2 \leq \epsilon$ and $\|x - \hat{x}\|^2 \leq \epsilon/L^2$, which implies that

$$(2.8) \quad \begin{aligned} \|\nabla f(x)\|^2 &= \|\nabla f(x) - \nabla f(\hat{x}) + \nabla f(\hat{x})\|^2 \leq 2\|\nabla f(x) - \nabla f(\hat{x})\|^2 + 2\|\nabla f(\hat{x})\|^2 \\ &\leq 2L^2\|x - \hat{x}\|^2 + 2\|\nabla f(\hat{x})\|^2 \leq 4\epsilon. \end{aligned}$$

Moreover, if X is a compact set and $x \in X$ is an (ϵ, δ) -solution, we can bound the strong gap as follows:

$$(2.9) \quad \begin{aligned} \text{gap}(x) &:= \max_{z \in X} \langle \nabla f(x), x - z \rangle \\ &= \max_{z \in X} \langle \nabla f(x) - \nabla f(\hat{x}), x - z \rangle + \max_{z \in X} \langle \nabla f(\hat{x}), x - \hat{x} \rangle + \max_{z \in X} \langle \nabla f(\hat{x}), \hat{x} - z \rangle \\ &\leq L\sqrt{\delta}D_X + \sqrt{\delta}\|\nabla f(\hat{x})\| + \sqrt{\epsilon}D_X, \end{aligned}$$

where $D_X := \max_{x_1, x_2 \in X} \|x_1 - x_2\|$. In comparison with the two well-known criteria in (2.8) and (2.9), the criterion given in Definition 2.1 seems to be applicable to a wider class of problems and is particularly suitable for proximal-point type methods (see [6] for a related notion).

We are now ready to state the main convergence properties for RapGrad.

THEOREM 2.2. *Let the iterates \bar{x}^ℓ , $\ell = 1, \dots, k$, be generated by Algorithm 1 and $\hat{\ell}$ be randomly selected from $[k]$. Suppose that in Algorithm 2, the number of iterations $s = \lceil -\log \widetilde{M}/\log \alpha \rceil$ with*

$$(2.10) \quad \widetilde{M} := 6 \left(5 + \frac{2L}{\mu} \right) \max \left\{ \frac{6}{5}, \frac{L^2}{\mu^2} \right\}, \quad \alpha = 1 - \frac{2}{m(\sqrt{1+16c/m}+1)}, \quad c = 2 + \frac{L}{\mu},$$

and other parameters are set to

$$(2.11) \quad \alpha_t = \alpha, \quad \gamma_t = \alpha^{-t}, \quad \tau_t = \frac{1}{m(1-\alpha)} - 1, \quad \text{and} \quad \eta_t = \frac{\alpha}{1-\alpha} \quad \forall t = 1, \dots, s.$$

Then we have

$$\begin{aligned}\mathbb{E} \left[d \left(\nabla f(x_*^{\hat{\ell}}), -N_X(x_*^{\hat{\ell}}) \right) \right]^2 &\leq \frac{36\mu}{k} [f(\bar{x}^0) - f(x^*)], \\ \mathbb{E} \|\bar{x}^{\hat{\ell}} - x_*^{\hat{\ell}}\|^2 &\leq \frac{4\mu}{kL^2} [f(\bar{x}^0) - f(x^*)],\end{aligned}$$

where x^* and $x_*^{\hat{\ell}}$ denote the optimal solutions to problem (1.1) and the $\hat{\ell}$ th subproblem (2.1), respectively.

Theorem 2.2 guarantees, in expectation, the existence of an approximate stationary point $x_*^{\hat{\ell}}$, which is the optimal solution to the $\hat{\ell}$ th subproblem. Though $x_*^{\hat{\ell}}$ is unknown to us, we can output the computable solution $\bar{x}^{\hat{\ell}}$ since it is close enough to $x_*^{\hat{\ell}}$. Moreover, its quality can be directly measured by (2.8) and (2.9) under certain important circumstances.

In view of Theorem 2.2, we can bound the total number of gradient evaluations required by RapGrad to yield a stochastic (ϵ, δ) -solution of (1.1). Indeed, observe that the full gradient is computed only once in the first outer loop and that for each subproblem (1.1), we only need to compute s gradients with

$$s = \left\lceil -\frac{\log \tilde{M}}{\log \alpha} \right\rceil \sim \mathcal{O} \left(\left(m + \sqrt{m \frac{L}{\mu}} \right) \log \left(\frac{L}{\mu} \right) \right).$$

Hence, the total number of gradient evaluations performed by RapGrad can be bounded by

$$N(\epsilon, \delta) := \mathcal{O} \left(m + \left(m + \sqrt{\frac{mL}{\mu}} \right) \log \frac{L}{\mu} + \mu \left(m + \sqrt{m \frac{L}{\mu}} \right) \log \left(\frac{L}{\mu} \right) \cdot \max \left\{ \frac{1}{\delta L^2}, \frac{1}{\epsilon} \right\} D^0 \right),$$

where $D^0 := f(\bar{x}^0) - f(x^*)$. As a comparison, the batch version of this algorithm, obtained by viewing $\frac{1}{m} \sum_{i=1}^m f_i(x)$ as a single component would update all the \underline{x}_i^t and y_i^t for $i = 1, \dots, m$, in (2.4) and (2.5) at each iteration, and hence would require

$$\hat{N}(\epsilon, \delta) := \mathcal{O} \left(m \sqrt{\frac{L}{\mu}} \log \frac{L}{\mu} + m \sqrt{L\mu} \log \left(\frac{L}{\mu} \right) \cdot \max \left\{ \frac{1}{\delta L^2}, \frac{1}{\epsilon} \right\} D^0 \right)$$

gradient evaluations to compute an (ϵ, δ) -solution of (1.1).

It is worth noting that both complexity bounds $N(\epsilon, \delta)$ and $\hat{N}(\epsilon, \delta)$ will go to $+\infty$ as μ tends to 0. This indicates that the proximal-point method should not be applied to the situation when μ is too small, no matter whether the randomized or batch version is used. However, if μ is indeed small, say, $\mu < \epsilon$, we can see that the assumption in (1.3) also holds for $\mu = \epsilon$. Therefore, we can assume that $\mu \geq \epsilon$ when applying RapGrad. A closer examination of this issue reveals that the algorithm (i.e., RaGrad) that we used to solve the strongly convex subproblems is not optimal if the strongly convex modulus μ is too small. Hence, a different remedy would be to apply an optimal randomized incremental gradient method which can yield the best possible complexity bound even if μ is small. Using this latter approach, we possibly do not need to fix ϵ a priori as in the former one. However, this type of optimal randomized incremental gradient method has not been developed until recently (see [17]), about one year later after our paper was initially released.

For problems with $L/\mu \geq m$, RapGrad can potentially save the total number of gradient computations up to a factor of $\mathcal{O}(\sqrt{m})$ gradient evaluations more than its batch counterpart as well as other deterministic batch methods reported in [24, 16]. It is also interesting to compare RapGrad with those variance-reduced stochastic

algorithms [25, 26, 2]. For simplicity, consider the case when $\delta = \epsilon/L^2$ and $X \equiv \mathbb{R}^n$. In this case, the complexity bound of RapGrad, given by $\mathcal{O}(\sqrt{mL\mu}/\epsilon)$, is smaller than those of variance-reduced stochastic algorithms [25, 26, 2] by a factor of $\mathcal{O}(m^{\frac{1}{6}}L^{\frac{1}{2}}/\mu^{\frac{1}{2}})$, which must be greater than $\mathcal{O}(m^{\frac{2}{3}})$ due to $L/\mu \geq m$. In fact, our complexity bound minorizes those for variance-reduced stochastic algorithms as long as $L/\mu \log(L/\mu) > m^{\frac{1}{3}}$.

Theorem 2.2 only shows the convergence of RapGrad in expectation. Similarly to the nonconvex SGD methods in [8, 11], we can establish and then further improve the convergence of RapGrad with overwhelming probability by using a two-phase procedure, where one computes a short list of candidate solutions in the optimization phase by either taking a few independent runs of RapGrad or randomly selecting a few solutions from the trajectory of RapGrad, and then chooses the best solution, e.g., in terms of either (2.8) and (2.9), in the postoptimization phase.

2.2. Convergence analysis for RapGrad. In this section, we will first develop the convergence results for Algorithm 2 applied to the convex finite-sum subproblem (2.2) and then use them to establish the convergence of RapGrad. Observe that the component functions ψ_i and φ in (2.2) satisfy

$$(A) \frac{\mu}{2}\|x - y\|^2 \leq \psi_i(x) - \psi_i(y) - \langle \nabla \psi_i(y), x - y \rangle \leq \frac{\hat{L}}{2}\|x - y\|^2 \quad \forall x, y \in X, \quad i = 1, \dots, m,$$

$$(B) \varphi(x) - \varphi(y) - \langle \nabla \varphi(y), x - y \rangle \geq \frac{\mu}{2}\|x - y\|^2 \quad \forall x, y \in X,$$

where $\hat{L} = L + 2\mu$.

We first state some simple relations about the iterations generated by Algorithm 2.

LEMMA 2.3. Let $\hat{x}_i^t = (1 + \tau_t)^{-1}(\tilde{x}^t + \tau_t x_i^{t-1})$ for $i = 1, \dots, m$, $t = 1, \dots, s$.

$$(2.12) \quad \mathbb{E}_{i_t}[\psi(\hat{x}_i^t)] = m\psi(\underline{x}_i^t) - (m-1)\psi(\underline{x}_i^{t-1}),$$

$$(2.13) \quad \mathbb{E}_{i_t}[\nabla \psi(\hat{x}_i^t)] = m\nabla \psi(\underline{x}_i^t) - (m-1)\nabla \psi(\underline{x}_i^{t-1}) = \mathbb{E}_{i_t}[\tilde{y}_i^t].$$

Proof. By the definition of \hat{x}_i^t , it is easy to see that $\mathbb{E}_{i_t}[\underline{x}_i^t] = \frac{1}{m}\hat{x}_i^t + \frac{m-1}{m}\underline{x}_i^{t-1}$, thus $\mathbb{E}_{i_t}[\psi_i(\underline{x}_i^t)] = \frac{1}{m}\psi_i(\hat{x}_i^t) + \frac{m-1}{m}\psi_i(\underline{x}_i^{t-1})$, and $\mathbb{E}_{i_t}[\nabla \psi_i(\underline{x}_i^t)] = \frac{1}{m}\nabla \psi_i(\hat{x}_i^t) + \frac{m-1}{m}\nabla \psi_i(\underline{x}_i^{t-1})$, which combined with the fact $\tilde{y}_i^t = m(y_i^t - y_i^{t-1}) + y_i^{t-1}$ gives us the desired relations. \square

Lemma 2.4 below describes an important result about Algorithm 2, which improves Lemma 7 of [18] by showing the convergence of \underline{x}_i^s . The proof of this result is more involved and will be deferred until Appendix A.

LEMMA 2.4. Let the iterates x^t and y^t for $t = 1, \dots, s$ be generated by Algorithm 2 and x^* be an optimal solution of (2.2). If the parameters in Algorithm 2 satisfy for all $t = 1, \dots, s-1$,

$$(2.14) \quad \alpha_{t+1}\gamma_{t+1} = \gamma_t,$$

$$(2.15) \quad \gamma_{t+1}[m(1 + \tau_{t+1}) - 1] \leq m\gamma_t(1 + \tau_t),$$

$$(2.16) \quad \gamma_{t+1}\eta_{t+1} \leq \gamma_t(1 + \eta_t),$$

$$(2.17) \quad \frac{\eta_s\mu}{4} \geq \frac{(m-1)^2\hat{L}}{m^2\tau_s},$$

$$(2.18) \quad \frac{\eta_t\mu}{2} \geq \frac{\alpha_{t+1}\hat{L}}{\tau_{t+1}} + \frac{(m-1)^2\hat{L}}{m^2\tau_t},$$

$$(2.19) \quad \frac{\eta_s\mu}{4} \geq \frac{\hat{L}}{m(1+\tau_s)},$$

then we have

$$\begin{aligned} \mathbb{E}_s & \left[\gamma_s(1 + \eta_s)V_\varphi(x^*, x^s) + \sum_{i=1}^m \frac{\mu\gamma_s(1+\tau_s)}{4} \|\underline{x}_i^s - x^*\|^2 \right] \\ & \leq \gamma_1\eta_1\mathbb{E}_s V_\varphi(x^*, x^0) + \sum_{i=1}^m \frac{\gamma_1[(1+\tau_1)-1/m]\hat{L}}{2}\mathbb{E}_s \|\underline{x}_i^0 - x^*\|^2. \end{aligned}$$

With the help of Lemma 2.4, we now establish the main convergence properties of Algorithm 2.

THEOREM 2.5. *Let x^* be an optimal solution of (2.2), and suppose that the parameters $\{\alpha_t\}$, $\{\tau_t\}$, $\{\eta_t\}$, and $\{\gamma_t\}$ are set as in (2.10) and (2.11). If $\varphi(x) = \frac{\mu}{2}\|x - z\|^2$ for some $z \in X$, then, for any $s \geq 1$, we have*

$$\begin{aligned} \mathbb{E}_s [\|x^* - x^s\|^2] & \leq \alpha^s (1 + 2\frac{\hat{L}}{\mu}) \mathbb{E}_s [\|x^* - x^0\|^2 + \frac{1}{m} \sum_{i=1}^m \|\underline{x}_i^0 - x^0\|^2], \\ \mathbb{E}_s [\frac{1}{m} \sum_{i=1}^m \|\underline{x}_i^s - x^s\|^2] & \leq 6\alpha^s (1 + 2\frac{\hat{L}}{\mu}) \mathbb{E}_s [\|x^* - x^0\|^2 + \frac{1}{m} \sum_{i=1}^m \|\underline{x}_i^0 - x^0\|^2]. \end{aligned}$$

Proof. It is easy to check that (2.10) and (2.11) satisfy conditions (2.14), (2.15), (2.16), (2.17), (2.18), and (2.19). Then by Lemma 2.4, we have

(2.20)

$$\mathbb{E}_s [V_\varphi(x^*, x^s) + \sum_{i=1}^m \frac{\mu}{4m} \|\underline{x}_i^s - x^*\|^2] \leq \alpha^s \mathbb{E}_s [V_\varphi(x^*, x^0) + \sum_{i=1}^m \frac{\hat{L}}{2m} \|\underline{x}_i^0 - x^*\|^2].$$

Since $\varphi(x) = \frac{\mu}{2}\|x - z\|^2$, we have $V_\varphi(x^*, x^s) = \frac{\mu}{2}\|x^* - x^s\|^2$ and $V_\varphi(x^0, x^s) = \frac{\mu}{2}\|x^* - x^0\|^2$. Plugging into (2.20), we obtain the following two relations:

$$\begin{aligned} \mathbb{E}_s [\|x^* - x^s\|^2] & \leq \alpha^s \mathbb{E}_s [\|x^* - x^0\|^2 + \sum_{i=1}^m \frac{\hat{L}}{mr} \|\underline{x}_i^0 - x^*\|^2] \\ & \leq \alpha^s \mathbb{E}_s [\|x^* - x^0\|^2 + \sum_{i=1}^m \frac{\hat{L}}{mr} (2\|\underline{x}_i^0 - x^0\|^2 + 2\|x^0 - x^*\|^2)] \\ & = \alpha^s \mathbb{E}_s [(1 + 2\frac{\hat{L}}{\mu})\|x^* - x^0\|^2 + \sum_{i=1}^m \frac{2\hat{L}}{m\mu} \|\underline{x}_i^0 - x^0\|^2] \\ & \leq \alpha^s (1 + 2\frac{\hat{L}}{\mu}) \mathbb{E}_s [\|x^* - x^0\|^2 + \sum_{i=1}^m \frac{1}{m} \|\underline{x}_i^0 - x^0\|^2], \\ \mathbb{E}_s [\frac{1}{m} \sum_{i=1}^m \|\underline{x}_i^s - x^s\|^2] & \leq 2\alpha^s \mathbb{E}_s [\|x^* - x^0\|^2 + \sum_{i=1}^m \frac{\hat{L}}{m\mu} \|\underline{x}_i^0 - x^*\|^2] \\ & \leq 2\alpha^s (1 + \frac{2\hat{L}}{\mu}) \mathbb{E}_s [\|x^* - x^0\|^2 + \frac{1}{m} \sum_{i=1}^m \|\underline{x}_i^0 - x^0\|^2]. \end{aligned}$$

In view of the above two relations, we have

$$\begin{aligned} \mathbb{E}_s [\frac{1}{m} \sum_{i=1}^m \|\underline{x}_i^s - x^s\|^2] & \leq \mathbb{E}_s [\frac{1}{m} \sum_{i=1}^m 2(\|\underline{x}_i^s - x^*\|^2 + \|x^* - x^s\|^2)] \\ & = 2\mathbb{E}_s [\frac{1}{m} \sum_{i=1}^m \|\underline{x}_i^s - x^*\|^2] + 2\mathbb{E}_s \|x^* - x^s\|^2 \\ & \leq 6\alpha^s (1 + 2\frac{\hat{L}}{\mu}) \mathbb{E}_s [\|x^* - x^0\|^2 + \frac{1}{m} \sum_{i=1}^m \|\underline{x}_i^0 - x^0\|^2]. \quad \square \end{aligned}$$

In view of Theorem 2.5, Algorithm 2 applied to subproblem (2.2) exhibits a fast linear rate of convergence. Actually, as shown below we do not need to solve the subproblem too accurately, and a constant number of iteration of Algorithm 2 for each subproblem is enough to guarantee the convergence of Algorithm 1.

LEMMA 2.6. *Let the number of inner iterations $s \geq \lceil -\log(7M/6)/\log \alpha \rceil$ with $M := 6(5 + 2L/\mu)$ be given. Also let the iterates \bar{x}^ℓ , $\ell = 1, \dots, k$, be generated by*

Algorithm 1, and $\hat{\ell}$ be randomly selected from $[k]$. Then

$$\begin{aligned}\mathbb{E}\|x_*^{\hat{\ell}} - \bar{x}^{\hat{\ell}-1}\|^2 &\leq \frac{4(1-M\alpha^s)}{k\mu(6-7M\alpha^s)}[f(\bar{x}^0) - f(x^*)], \\ \mathbb{E}\|x_*^{\hat{\ell}} - \bar{x}^{\hat{\ell}}\|^2 &\leq \frac{2M\alpha^s}{3k\mu(6-7M\alpha^s)}[f(\bar{x}^0) - f(x^*)],\end{aligned}$$

where x^* and x_*^ℓ are the optimal solutions to problem (1.1) and the ℓ th subproblem (2.1), respectively.

Proof. According to Theorem 2.5 (with $\hat{L} = 2\mu + L$), we have, for $\ell \geq 1$,

$$\begin{aligned}\mathbb{E}\|x_*^\ell - \bar{x}^\ell\|^2 &\leq \alpha^s(5 + \frac{2L}{\mu})\mathbb{E}\left[\|x_*^\ell - \bar{x}^{\ell-1}\|^2 + \sum_{i=1}^m \frac{1}{m}\|\bar{x}_i^{\ell-1} - \bar{x}^{\ell-1}\|^2\right] \\ (2.21) \quad &\leq \frac{M\alpha^s}{6}\mathbb{E}\left[\|x_*^\ell - \bar{x}^{\ell-1}\|^2 + \sum_{i=1}^m \frac{1}{m}\|\bar{x}_i^{\ell-1} - \bar{x}^{\ell-1}\|^2\right],\end{aligned}$$

$$\begin{aligned}\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^m\|\bar{x}_i^\ell - \bar{x}^\ell\|^2\right] &\leq 4\alpha^s(5 + \frac{2L}{\mu})\mathbb{E}\left[\|x_*^\ell - \bar{x}^{\ell-1}\|^2 + \sum_{i=1}^m \frac{1}{m}\|\bar{x}_i^{\ell-1} - \bar{x}^{\ell-1}\|^2\right] \\ (2.22) \quad &\leq M\alpha^s\mathbb{E}\left[\|x_*^\ell - \bar{x}^{\ell-1}\|^2 + \sum_{i=1}^m \frac{1}{m}\|\bar{x}_i^{\ell-1} - \bar{x}^{\ell-1}\|^2\right].\end{aligned}$$

By induction on (2.22) and noting $\bar{x}_i^0 = \bar{x}^0$, $i = 1, \dots, m$, we have

$$\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^m\|\bar{x}_i^\ell - \bar{x}^\ell\|^2\right] \leq \sum_{j=1}^\ell (M\alpha^s)^{\ell-j+1}\mathbb{E}\|x_*^j - \bar{x}^{j-1}\|^2.$$

In view of the above relation and (2.21), for $\ell \geq 2$, we have

$$\mathbb{E}\|x_*^\ell - \bar{x}^\ell\|^2 \leq \frac{M\alpha^s}{6}\mathbb{E}\left[\|x_*^\ell - \bar{x}^{\ell-1}\|^2 + \sum_{j=1}^{\ell-1} (M\alpha^s)^{\ell-j}\|x_*^j - \bar{x}^{j-1}\|^2\right].$$

Summing up both sides of the above inequality from $\ell = 1$ to k , we then obtain

$$\begin{aligned}(2.23) \quad &\sum_{\ell=1}^k \mathbb{E}\|x_*^\ell - \bar{x}^\ell\|^2 \\ &\leq \frac{M\alpha^s}{6}\mathbb{E}\left[\|x_*^1 - \bar{x}^0\|^2 + \sum_{\ell=2}^k \left(\|x_*^\ell - \bar{x}^{\ell-1}\|^2 + \sum_{j=1}^{\ell-1} (M\alpha^s)^{\ell-j}\|x_*^j - \bar{x}^{j-1}\|^2\right)\right] \\ &= \frac{M\alpha^s}{6}\mathbb{E}\left[\|x_*^k - \bar{x}^{k-1}\|^2 + \sum_{\ell=1}^{k-1} \left(\frac{1}{1-M\alpha^s} - \frac{(M\alpha^s)^{k+1-\ell}}{1-M\alpha^s}\right)\|x_*^\ell - \bar{x}^{\ell-1}\|^2\right] \\ &\leq \frac{M\alpha^s}{6(1-M\alpha^s)}\sum_{\ell=1}^k \mathbb{E}\|x_*^\ell - \bar{x}^{\ell-1}\|^2.\end{aligned}$$

Using the fact that x_*^ℓ is optimal to the ℓ th subproblem, and letting $x_*^0 = \bar{x}^0$ (x_*^0 is a free variable), we have

$$\sum_{\ell=1}^k [\psi^\ell(x_*^\ell) + \varphi^\ell(x_*^\ell)] \leq \sum_{\ell=1}^k [\psi^\ell(x_*^{\ell-1}) + \varphi^\ell(x_*^{\ell-1})],$$

which, in view of the definition of ψ^ℓ and φ^ℓ , then implies that

$$(2.24) \quad \sum_{\ell=1}^k \mathbb{E}[f(x_*^\ell) + \frac{3\mu}{2}\|x_*^\ell - \bar{x}^{\ell-1}\|^2] \leq \sum_{\ell=1}^k \mathbb{E}[f(x_*^{\ell-1}) + \frac{3\mu}{2}\|x_*^{\ell-1} - \bar{x}^{\ell-1}\|^2].$$

Combining (2.23) and (2.24), we obtain

$$\begin{aligned}(2.25) \quad &\frac{3\mu}{2}\sum_{\ell=1}^k \mathbb{E}\|x_*^\ell - \bar{x}^{\ell-1}\|^2 \\ &\leq \sum_{\ell=1}^k \mathbb{E}\{f(x_*^{\ell-1}) - f(x_*^\ell)\} + \frac{3\mu}{2}\sum_{\ell=1}^k \mathbb{E}\|x_*^{\ell-1} - \bar{x}^{\ell-1}\|^2 \\ &\leq \sum_{\ell=1}^k \mathbb{E}\{f(x_*^{\ell-1}) - f(x_*^\ell)\} + \frac{3\mu}{2}\sum_{\ell=1}^k \mathbb{E}\|x_*^\ell - \bar{x}^\ell\|^2 \\ &\leq \sum_{\ell=1}^k \mathbb{E}\{f(x_*^{\ell-1}) - f(x_*^\ell)\} + \frac{3\mu}{2}\frac{M\alpha^s}{6(1-M\alpha^s)}\sum_{\ell=1}^k \mathbb{E}\|x_*^\ell - \bar{x}^{\ell-1}\|^2.\end{aligned}$$

Using (2.25), (2.23), and the condition on s , we have

$$\begin{aligned}\sum_{\ell=1}^k \mathbb{E} \|x_*^\ell - \bar{x}^{\ell-1}\|^2 &\leq \frac{4(1-M\alpha^s)}{\mu(6-7M\alpha^s)} [f(\bar{x}^0) - f(x^*)], \\ \sum_{\ell=1}^k \mathbb{E} \|x_*^\ell - \bar{x}^\ell\|^2 &\leq \frac{2M\alpha^s}{3\mu(6-7M\alpha^s)} [f(\bar{x}^0) - f(x^*)].\end{aligned}$$

Our results then immediately follow since $\hat{\ell}$ is chosen randomly in $[k]$. \square

Now we are ready to prove Theorem 2.2 using all the previous results we have developed.

Proof of Theorem 2.2. By the optimality condition of the $\hat{\ell}$ th subproblem (2.1),

$$(2.26) \quad \nabla \psi^{\hat{\ell}}(x_*^{\hat{\ell}}) + \nabla \varphi^{\hat{\ell}}(x_*^{\hat{\ell}}) \in -N_X(x_*^{\hat{\ell}}).$$

From the definition of $\psi^{\hat{\ell}}$ and $\varphi^{\hat{\ell}}$, we have

$$(2.27) \quad \nabla f(x_*^{\hat{\ell}}) + 3\mu(x_*^{\hat{\ell}} - \bar{x}^{\hat{\ell}-1}) \in -N_X(x_*^{\hat{\ell}}).$$

From the optimality condition of (2.7), we obtain

$$(2.28) \quad \begin{aligned}\varphi(x^t) - \varphi(x^*) + \left\langle \frac{1}{m} \sum_{i=1}^m \tilde{y}_i^t, x^t - x^* \right\rangle \\ \leq \eta_t V_\varphi(x^*, x^{t-1}) - (1 + \eta_t) V_\varphi(x^*, x^t) - \eta_t V_\varphi(x^t, x^{t-1}).\end{aligned}$$

Using the above relation and Lemma 2.6, we have

$$\begin{aligned}\mathbb{E} \|\bar{x}^{\hat{\ell}-1} - x_*^{\hat{\ell}}\|^2 &\leq \frac{4(1-M\alpha^s)}{k\mu(6-7M\alpha^s)} [f(\bar{x}^0) - f(x^*)] \leq \frac{4}{k\mu} [f(\bar{x}^0) - f(x^*)], \\ \mathbb{E} \left[d \left(\nabla f(x_*^{\hat{\ell}}), -N_X(x_*^{\hat{\ell}}) \right) \right]^2 &\leq \mathbb{E} \|3\mu(\bar{x}^{\hat{\ell}-1} - x_*^{\hat{\ell}})\|^2 \leq \frac{36\mu}{k} [f(\bar{x}^0) - f(x^*)], \\ \mathbb{E} \|\bar{x}^{\hat{\ell}} - x_*^{\hat{\ell}}\|^2 &\leq \frac{2M\alpha^s}{3k\mu(6-7M\alpha^s)} [f(\bar{x}^0) - f(x^*)] \leq \frac{4M\alpha^s}{k\mu} [f(\bar{x}^0) - f(x^*)] \\ &\leq \frac{4\mu}{kL^2} [f(\bar{x}^0) - f(x^*)].\end{aligned} \quad \square$$

3. Nonconvex multiblock optimization with linear constraints. In this section, we present a RapDual algorithm for solving the nonconvex multiblock optimization problem in (1.4) and show the potential advantages in terms of the total number of block updates.

As mentioned in section 1, we assume the inverse of the last block of the constraint matrix is easily computable. Hence, denoting $\mathbf{A}_i = A_m^{-1} A_i$, $i = 1, \dots, m-1$, and $\mathbf{b} = A_m^{-1}$, we can reformulate problem (1.4) as

$$(3.1) \quad \begin{aligned}\min_{\mathbf{x} \in X, x_m \in \mathbb{R}^n} \quad & f(\mathbf{x}) + f_m(x_m), \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + x_m = \mathbf{b},\end{aligned}$$

where $f(\mathbf{x}) := \sum_{i=1}^{m-1} f_i(x_i)$, $X = X_1 \times \dots \times X_{m-1}$, $\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_{m-1}]$, and $\mathbf{x} = (x_1, \dots, x_{m-1})$. It should be noted that except for some special cases, the computation of A_m^{-1} requires up to $\mathcal{O}(n^3)$ arithmetic operations, which will be a one-time computational cost added on top of the overall computational cost of our algorithm (see Remark 3.3 below for more discussions).

One may also reformulate problem (3.1) in the form of (1.1) and directly apply Algorithm 1 to solve it. More specifically, substituting x_m with $\mathbf{b} - \mathbf{A}\mathbf{x}$ in the objective function of (3.1), we obtain

$$(3.2) \quad \min_{\mathbf{x} \in X} \sum_{i=1}^{m-1} f_i(B_i \mathbf{x}) + f_m(\mathbf{b} - \mathbf{A}\mathbf{x}),$$

where $B_i = (\mathbf{0}, \dots, I, \dots, \mathbf{0})$ with the i th block given a $d_i \times d_i$ identity matrix and hence $x_i = B_i \mathbf{x}$. However, this method will be inefficient since we enlarge the dimension of each f_i from d_i to $\sum_{i=1}^{m-1} d_i$ and as a result, every block has to be updated in each iteration. One may also try to apply a nonconvex randomized block coordinate descent method [5] to solve the above reformulation. However, such methods do not apply to the case when f_i are both nonconvex and nonsmooth. This motivates us to design the new RapDual method which requires updating only a single block at a time, applies to the case when f_i is nonsmooth, and achieves an accelerated rate of convergence when f_i is smooth.

3.1. The algorithm. The main idea of RapDual is similar to the one used to design the RapGrad method introduced in section 2.1. Given the proximal points $\bar{\mathbf{x}}^{\ell-1}$ and $\bar{x}_m^{\ell-1}$ from the previous iteration, we define a new proximal subproblem as

$$(3.3) \quad \begin{aligned} & \min_{\mathbf{x} \in X, x_m \in \mathbb{R}^n} \psi(\mathbf{x}) + \psi_m(x_m) \\ & \text{s.t. } \mathbf{A}\mathbf{x} + x_m = \mathbf{b}, \end{aligned}$$

where $\psi(\mathbf{x}) := f(\mathbf{x}) + \mu \|\mathbf{x} - \bar{\mathbf{x}}^{\ell-1}\|^2$ and $\psi_m(x_m) := f_m(x_m) + \mu \|x_m - x_m^{\ell-1}\|^2$. Obviously, RaGrad does not apply directly to this type of subproblem. In this subsection, we present a new randomized algorithm, named the randomized accelerated dual (RaDual) method, to solve the subproblem in (3.3), which will be iteratively called the RapDual method to solve problem (3.1).

RaDual (cf. Algorithm 4) can be viewed as a randomized primal-dual type method. Indeed, by the method of multipliers and Fenchel conjugate duality, we have

$$(3.4) \quad \begin{aligned} & \min_{\mathbf{x} \in X, x_m \in \mathbb{R}^n} \{\psi(\mathbf{x}) + \psi_m(x_m) + \max_{y \in \mathbb{R}^n} \langle \sum_{i=1}^m \mathbf{A}_i x_i - \mathbf{b}, y \rangle\} \\ & = \min_{\mathbf{x} \in X} \{\psi(\mathbf{x}) + \max_{y \in \mathbb{R}^n} [\langle \mathbf{A}\mathbf{x} - \mathbf{b}, y \rangle + \min_{x_m \in \mathbb{R}^n} \{\psi_m(x_m) + \langle x_m, y \rangle\}]\} \\ & = \min_{\mathbf{x} \in X} \{\psi(\mathbf{x}) + \max_{y \in \mathbb{R}^n} [\langle \mathbf{A}\mathbf{x} - \mathbf{b}, y \rangle - h(y)]\}, \end{aligned}$$

where $h(y) := -\min_{x_m \in \mathbb{R}^n} \{\psi_m(x_m) + \langle x_m, y \rangle\} = \psi_m^*(-y)$. Observe that the above saddle-point problem is both strongly convex in \mathbf{x} and strongly concave in y . Indeed, $\psi(\mathbf{x})$ is strongly convex due to the added proximal term. Moreover, since ψ_m has \hat{L} -Lipschitz continuous gradients, $h(y) = \psi_m^*(-y)$ is $1/\hat{L}$ -strongly convex. Using the fact that h is strongly convex, we can see that (3.7)–(3.8) in Algorithm 4 is equivalent to a dual mirror-descent step with a properly chosen distance generating function $V_h(y, y^{t-1})$. Specifically,

$$\begin{aligned} y^t &= \arg \min_{y \in \mathbb{R}^n} h(y) + \langle -\mathbf{A}\tilde{\mathbf{x}}^t + \mathbf{b}, y \rangle + \tau_t V_h(y, y^{t-1}) \\ &= \arg \max_{y \in \mathbb{R}^n} \langle (\mathbf{A}\tilde{\mathbf{x}}^t - \mathbf{b} + \tau_t \nabla h(y^{t-1}))/ (1 + \tau_t), y \rangle - h(y) \\ &= \nabla h^*[(\mathbf{A}\tilde{\mathbf{x}}^t - \mathbf{b} + \tau_t \nabla h(y^{t-1}))/ (1 + \tau_t)]. \end{aligned}$$

If we set $g^0 = \nabla h(y^0) = -x_m^0$, then it is easy to see by induction that $g^t = (\tau_t g^{t-1} + \mathbf{A}\tilde{\mathbf{x}}^t - \mathbf{b})/(1 + \tau_t)$, and $y^t = \nabla h^*(g^t)$ for all $t \geq 1$. Moreover, $h^*(g) = \max_{y \in \mathbb{R}^n} \langle g, y \rangle - h(y) = \max_{y \in \mathbb{R}^n} \langle g, y \rangle - \psi_m^*(-y) = \psi_m(-g)$, and thus $y^t = -\nabla \psi_m(-g^t)$ is the negative gradient of ψ_m at point $-g^t$. Therefore, Algorithm 4 does not explicitly depend on the function h , even though the above analysis does.

Each iteration of Algorithm 4 updates only a randomly selected block i_t in (3.9), making it especially favorable when the number of blocks m is large. However, a

similar difficulty as mentioned in section 2.1 also appears when we integrate this algorithm with a proximal-point type method to yield the final RapDual method in Algorithm 3. First, Algorithm 4 also keeps a few intertwined primal and dual sequences, and thus we need to carefully decide the input and output of Algorithm 4 so that information from previous iterations of RapDual is fully used. Second, the number of iterations performed by Algorithm 4 to solve each subproblem plays a vital role in the convergence rate of RapDual, which should be carefully predetermined.

Algorithm 3 RapDual for nonconvex multiblock optimization.

Compute A_m^{-1} and reformulate problem (1.4) as (3.1).
 Let $\bar{\mathbf{x}}^0 \in X$, $\bar{x}_m^0 \in \mathbb{R}^n$, such that $\mathbf{A}\bar{\mathbf{x}}^0 + \bar{x}_m^0 = \mathbf{b}$ and $\bar{y}^0 = -\nabla f_m(\bar{x}_m^0)$.
for $\ell = 1, \dots, k$ **do**
 Set $\mathbf{x}^{-1} = \mathbf{x}^0 = \bar{\mathbf{x}}^{\ell-1}$, $x_m^0 = \bar{x}_m^{\ell-1}$.
 Run Algorithm 4 with input \mathbf{x}^{-1} , \mathbf{x}^0 , x_m^0 , and s to solve the subproblem

$$(3.5) \quad \begin{aligned} & \min_{\mathbf{x} \in X, x_m \in \mathbb{R}^n} \psi(\mathbf{x}) + \psi_m(x_m) \\ & \text{s.t. } \mathbf{A}\mathbf{x} + x_m = \mathbf{b} \end{aligned}$$

to compute output (\mathbf{x}^s, x_m^s) , where $\psi(\mathbf{x}) \equiv \psi^\ell(\mathbf{x}) := f(\mathbf{x}) + \mu\|\mathbf{x} - \bar{\mathbf{x}}^{\ell-1}\|^2$ and $\psi_m(x) \equiv \psi_m^\ell(x) := f_m(x_m) + \mu\|x_m - \bar{x}_m^{\ell-1}\|^2$.
 Set $\bar{\mathbf{x}}^\ell = \mathbf{x}^s$, $\bar{x}_m^\ell = x_m^s$.

end for
return $(\bar{\mathbf{x}}^\ell, \bar{x}_m^\ell)$ for some random $\hat{\ell} \in [k]$.

Algorithm 4 RaDual for solving subproblem (3.3).

Let $\mathbf{x}^{-1} = \mathbf{x}^0 \in X$, $x_m \in \mathbb{R}^n$, number of iterations s , and nonnegative parameters $\{\alpha_t\}$, $\{\tau_t\}$, $\{\eta_t\}$ be given. Set $g^0 = -x_m^0$.

for $t = 1, \dots, s$ **do**
 1. Generate a random variable i_t uniformly distributed over $[m - 1]$.
 2. Update x^t and y^t according to

$$(3.6) \quad \tilde{\mathbf{x}}^t = \alpha_t(\mathbf{x}^{t-1} - \mathbf{x}^{t-2}) + \mathbf{x}^{t-1},$$

$$(3.7) \quad g^t = (\tau_t g^{t-1} + \mathbf{A}\tilde{\mathbf{x}}^t - \mathbf{b})/(1 + \tau_t),$$

$$(3.8) \quad y^t = \arg \min_{y \in \mathbb{R}^n} h(y) + \langle -\mathbf{A}\tilde{\mathbf{x}}^t + \mathbf{b}, y \rangle + \tau_t V_h(y, y^{t-1}) = -\nabla \psi_m(-g^t),$$

$$(3.9) \quad x_i^t = \begin{cases} \arg \min_{x_i \in X_i} \psi_i(x_i) + \langle \mathbf{A}_i^\top y^t, x_i \rangle + \frac{\eta_t}{2} \|x_i - x_i^{t-1}\|^2, & i = i_t, \\ x_i^{t-1}, & i \neq i_t. \end{cases}$$

end for
 Compute $x_m^s = \arg \min_{x_m \in \mathbb{R}^n} \{\psi_m(x_m) + \langle x_m, y^s \rangle\}$.
return (\mathbf{x}^s, x_m^s) .

Algorithm 3 describes the basic scheme of RapDual. At the beginning, all the blocks are initialized using the output from solving the previous subproblem. Note that x_m^0 is used to initialize g , which further helps compute the dual variable y without using the conjugate function h of ψ_m . We will derive the convergence result for Algorithm 4 in terms of primal variables and construct relations between successive search points $(\mathbf{x}^\ell, x_m^\ell)$, which will be used to prove the final convergence of RapDual.

We first define an approximate stationary point for problem (1.4) before establishing the convergence of RapDual.

DEFINITION 3.1. A point $(\mathbf{x}, x_m) \in X \times \mathbb{R}^n$ is called an $(\epsilon, \delta, \sigma)$ -solution of (1.4) if there exists some $\hat{\mathbf{x}} \in X$ and $\lambda \in \mathbb{R}^n$ such that

$$\begin{aligned} [d(\nabla f(\hat{\mathbf{x}}) + \mathbf{A}^\top \lambda, -N_X(\hat{\mathbf{x}}))]^2 &\leq \epsilon, \|\nabla f_m(x_m) + \lambda\|^2 \leq \epsilon, \\ \|\mathbf{x} - \hat{\mathbf{x}}\|^2 &\leq \delta, \|\mathbf{Ax} + x_m - \mathbf{b}\|^2 \leq \sigma. \end{aligned}$$

A stochastic counterpart is one that satisfies

$$\begin{aligned} \mathbb{E} [d(\nabla f(\hat{\mathbf{x}}) + \mathbf{A}^\top \lambda, -N_X(\hat{\mathbf{x}}))]^2 &\leq \epsilon, \mathbb{E} \|\nabla f_m(x_m) + \lambda\|^2 \leq \epsilon, \\ \mathbb{E} \|\mathbf{x} - \hat{\mathbf{x}}\|^2 &\leq \delta, \mathbb{E} \|\mathbf{Ax} + x_m - \mathbf{b}\|^2 \leq \sigma. \end{aligned}$$

Consider the unconstrained problem with $X = \mathbb{R}^{\sum_{i=1}^{m-1} d_i}$. If $(\mathbf{x}, x_m) \in X \times \mathbb{R}^n$ is an $(\epsilon, \delta, \sigma)$ -solution with $\delta = \epsilon/L^2$, then exists some $\hat{\mathbf{x}} \in X$ such that $\|\nabla f(\hat{\mathbf{x}})\|^2 \leq \epsilon$ and $\|\mathbf{x} - \hat{\mathbf{x}}\|^2 \leq \delta$. By a similar argument as in (2.8), we obtain $\|\nabla f(\mathbf{x})\|^2 \leq 4\epsilon$. Besides, the definition of a $(\epsilon, \delta, \sigma)$ -solution guarantees $\|\nabla f_m(x_m) + \lambda\|^2 \leq \epsilon$ and $\|\mathbf{Ax} + x_m - \mathbf{b}\|^2 \leq \sigma$, which altogether justify that (\mathbf{x}, x_m) is a reasonably good solution.

THEOREM 3.2. Let the iterates $(\mathbf{x}^\ell, x_m^\ell)$ for $\ell = 1, \dots, k$ be generated by Algorithm 3 and $\hat{\ell}$ be randomly selected from $[k]$. Suppose in Algorithm 4 that the number of iterations $s = \lceil -\log \widehat{\mathcal{M}} / \log \alpha \rceil$ with

$$(3.10) \quad \begin{aligned} \widehat{\mathcal{M}} &= (2 + \frac{L}{\mu}) \cdot \max \left\{ 2, \frac{L^2}{\mu^2} \right\}, \quad \alpha = 1 - \frac{2}{(m-1)(\sqrt{1+8c}+1)}, \\ c &= \frac{\bar{A}^2}{\mu\bar{\mu}} = \frac{(2\mu+L)\bar{A}^2}{\mu}, \quad \bar{A} = \max_{i \in [m-1]} \|\mathbf{A}_i\|, \end{aligned}$$

and other parameters are set to

$$(3.11) \quad \alpha_t = (m-1)\alpha, \quad \gamma_t = \alpha^{-t}, \quad \tau_t = \frac{\alpha}{1-\alpha}, \quad \eta_t = \frac{(\alpha - \frac{m-2}{m-1})\mu}{1-\alpha} \quad \forall t = 1, \dots, s.$$

Then there exists some $\lambda^* \in \mathbb{R}^n$ such that

$$\begin{aligned} &\mathbb{E} [d(\nabla f(\mathbf{x}_{*}^{\hat{\ell}}) + \mathbf{A}^\top \lambda^*, -N_X(\mathbf{x}_{*}^{\hat{\ell}}))]^2 \\ &\leq \frac{8\mu}{k} \{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\}, \\ &\mathbb{E} \|\nabla f_m(x_m^{\hat{\ell}}) + \lambda^*\|^2 \\ &\leq \frac{34\mu}{k} \{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\}, \\ &\mathbb{E} \|\mathbf{x}^{\hat{\ell}} - \mathbf{x}_{*}^{\hat{\ell}}\|^2 \\ &\leq \frac{2\mu}{kL^2} \{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\}, \\ &\mathbb{E} \|\mathbf{Ax}^{\hat{\ell}} + x_m^{\hat{\ell}} - \mathbf{b}\|^2 \\ &\leq \frac{2(\|\mathbf{A}\|^2+1)\mu}{kL^2} \{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\}, \end{aligned}$$

where (\mathbf{x}^*, x_m^*) and $(\mathbf{x}_{*}^{\hat{\ell}}, x_m^{\hat{\ell}})$ denote the optimal solutions to (1.4) and the ℓ th subproblem (3.3), respectively.

Theorem 3.2 ensures that our output solution $(\mathbf{x}^{\hat{\ell}}, x_m^{\hat{\ell}})$ is close enough to an unknown approximate stationary point $(\mathbf{x}_{*}^{\hat{\ell}}, x_m^{\hat{\ell}})$. According to Theorem 3.2, we can

bound the complexity of RapDual to compute a stochastic $(\epsilon, \delta, \sigma)$ -solution of (1.4) in terms of block updates in (3.9). Note that for each subproblem (3.3), we only need to update s primal blocks with

$$s = \left\lceil -\frac{\log \widehat{\mathcal{M}}}{\log \alpha} \right\rceil \sim \mathcal{O}\left(m\bar{A}\sqrt{\frac{L}{\mu}} \log\left(\frac{L}{\mu}\right)\right).$$

Let $\mathcal{D}^0 := f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]$. It can be seen that the total number of primal block updates required to obtain a stochastic $(\epsilon, \delta, \sigma)$ -solution can be bounded by

$$(3.12) \quad N(\epsilon, \delta, \sigma) := \mathcal{O}\left(m\bar{A}\sqrt{L\mu} \log\left(\frac{L}{\mu}\right) \cdot \max\left\{\frac{1}{\epsilon}, \frac{1}{\delta L^2}, \frac{\|\mathbf{A}\|^2}{\sigma L^2}\right\} \mathcal{D}^0\right).$$

As a comparison, the batch version of this algorithm would update all the x_i^t for $i = 1, \dots, m$, in (3.9) and thus would require

$$\hat{N}(\epsilon, \delta, \sigma) := \mathcal{O}\left(m\|\mathbf{A}\|\sqrt{L\mu} \log\left(\frac{L}{\mu}\right) \cdot \max\left\{\frac{1}{\epsilon}, \frac{1}{\delta L^2}, \frac{\|\mathbf{A}\|^2}{\sigma L^2}\right\} \mathcal{D}^0\right)$$

primal block updates to obtain an $(\epsilon, \delta, \sigma)$ -solution of (1.4). Therefore, the benefit of randomization comes from the difference between $\|\mathbf{A}\|$ and \bar{A} . Obviously we always have $\|\mathbf{A}\| > \bar{A}$, and the relative gap between $\|\mathbf{A}\|$ and \bar{A} can be large when all the matrix blocks have close norms. In the case when all the blocks are identical, i.e., $\mathbf{A}_1 = \mathbf{A}_2 = \dots = \mathbf{A}_{m-1}$, we immediately have $\|\mathbf{A}\| = \sqrt{m-1}\bar{A}$, which means that RapDual can potentially save the number of primal block updates by a factor of $\mathcal{O}(\sqrt{m})$ more than its batch counterpart.

It is also interesting to compare RapDual with the nonconvex randomized block coordinate descent method in [5]. To compare these methods, let us assume that f_i is smooth with \bar{L} -Lipschitz continuous gradient for some $\bar{L} \geq \mu$ for any $i = 1, \dots, m$. Also let us assume that $\sigma > \|\mathbf{A}\|^2\epsilon/L^2$, $\delta = \epsilon/L^2$, and $X = \mathbb{R}^{\sum_{i=1}^{m-1} d_i}$. Then, after disregarding some constant factors, the bound in (3.12) reduces to $\mathcal{O}(m\bar{A}\sqrt{L\mu}\mathcal{D}^0/\epsilon)$, which is always smaller than the $\mathcal{O}(m(\bar{L} + L\bar{A}^2)\mathcal{D}^0/\epsilon)$ complexity bound implied by Corollary 4.4 of [5].

REMARK 3.3. In this paper, we assume that A_m^{-1} is easily computable. One natural question is whether we can avoid the computation of A_m^{-1} by directly solving (1.4) instead of its reformulation (3.1). To do so, we can iteratively solve the following saddle-point subproblems in place of the ones in (3.4):

$$(3.13) \quad \begin{aligned} & \min_{\mathbf{x} \in X, x_m \in \mathbb{R}^n} \{\psi(\mathbf{x}) + \psi_m(x_m) + \max_{y \in \mathbb{R}^n} \langle \sum_{i=1}^m A_i x_i - b, y \rangle\} \\ &= \min_{\mathbf{x} \in X} \{\psi(\mathbf{x}) + \max_{y \in \mathbb{R}^n} [\langle A\mathbf{x} - b, y \rangle + \min_{x_m \in \mathbb{R}^n} \{\psi_m(x_m) + \langle A_m x_m, y \rangle\}]\} \\ &= \min_{\mathbf{x} \in X} \{\psi(\mathbf{x}) + \max_{y \in \mathbb{R}^n} [\langle A\mathbf{x} - b, y \rangle - \tilde{h}(y)]\}, \end{aligned}$$

where $A := [A_1, \dots, A_{m-1}]$ and $\tilde{h}(y) := \max_{x_m \in \mathbb{R}^n} \{-\psi_m(x_m) - \langle A_m x_m, y \rangle\}$. Instead of keeping $\tilde{h}(y)$ in the projection subproblem (3.8) as we did for $h(y)$, we need to linearize it at each iteration by computing its gradients $\nabla \tilde{h}(y^{t-1}) = -A_m^T \bar{x}_m(y^{t-1})$, where $\bar{x}_m(y^{t-1}) = \arg \max_{x_m \in \mathbb{R}^n} \{-\psi_m(x_m) - \langle A_m x_m, y^{t-1} \rangle\}$. Note that the latter optimization problem can be solved by using an efficient first-order method due to the smoothness and strong concavity of its objective function. As a result, we will be able to obtain a similar rate of convergence to RapDual without computing A_m^{-1} . However, the statement and analysis of the algorithm will be much more complicated than RapDual in its current form.

3.2. Convergence analysis for RapDual. In this section, we first show the convergence of Algorithm 4 for solving the convex multiblock subproblem (3.5) with

$$(A) \quad \psi_i(x) - \psi_i(y) - \langle \nabla \psi_i(y), x - y \rangle \geq \frac{\mu}{2} \|x - y\|^2 \quad \forall x, y \in X_i, \quad i = 1, \dots, m-1,$$

$$(B) \quad \frac{\mu}{2} \|x - y\|^2 \leq \psi_m(x) - \psi_m(y) - \langle \nabla \psi_m(y), x - y \rangle \leq \frac{\hat{L}}{2} \|x - y\|^2 \quad \forall x, y \in \mathbb{R}^n.$$

Some simple relations about the iterations generated by Algorithm 4 are characterized in the following lemma, and the proof follows directly from the definition of $\hat{\mathbf{x}}$ in (3.14) and thus has been omitted.

LEMMA 3.4. *Let $\hat{\mathbf{x}}^0 = \mathbf{x}^0$ and $\hat{\mathbf{x}}^t$ for $t = 1, \dots, s$ be defined as follows:*

$$(3.14) \quad \hat{\mathbf{x}}^t = \arg \min_{\mathbf{x} \in X} \psi(\mathbf{x}) + \langle \mathbf{A}^\top y^t, \mathbf{x} \rangle + \frac{\eta_t}{2} \|\mathbf{x} - \mathbf{x}^{t-1}\|^2,$$

where \mathbf{x}^t and y^t are obtained from (3.8)–(3.9), and then we have

$$(3.15) \quad \mathbb{E}_{i_t} \{ \|\mathbf{x} - \hat{\mathbf{x}}^t\|^2 \} = \mathbb{E}_{i_t} \{ (m-1) \|\mathbf{x} - \mathbf{x}^t\|^2 - (m-2) \|\mathbf{x} - \mathbf{x}^{t-1}\|^2 \},$$

$$(3.16) \quad \mathbb{E}_{i_t} \{ \|\hat{\mathbf{x}}^t - \mathbf{x}^{t-1}\|^2 \} = \mathbb{E}_{i_t} \{ (m-1) \|x_{i_t}^t - x_{i_t}^{t-1}\|^2 \}.$$

Lemma 3.5 builds some connections between the input and output of Algorithm 4 in terms of both primal and dual variables, and the proof can be found in Appendix B.

LEMMA 3.5. *Let the iterates \mathbf{x}^t and y^t for $t = 1, \dots, s$ be generated by Algorithm 4 and (\mathbf{x}^*, y^*) be a saddle point of (3.4). Assume that the parameters in Algorithm 4 satisfy for all $t = 1, \dots, s-1$*

$$(3.17) \quad \alpha_{t+1} = (m-1)\tilde{\alpha}_{t+1},$$

$$(3.18) \quad \gamma_t = \gamma_{t+1}\tilde{\alpha}_{t+1},$$

$$(3.19) \quad \gamma_{t+1}((m-1)\eta_{t+1} + (m-2)\mu) \leq (m-1)\gamma_t(\eta_t + \mu),$$

$$(3.20) \quad \gamma_{t+1}\tau_{t+1} \leq \gamma_t(\tau_t + 1),$$

$$(3.21) \quad 2(m-1)\tilde{\alpha}_{t+1}\bar{A}^2 \leq \bar{\mu}\eta_t\tau_{t+1},$$

where $\bar{A} = \max_{i \in [m-1]} \|\mathbf{A}_i\|$. Then we have

$$(3.22) \quad \begin{aligned} & \mathbb{E}_s \left\{ \frac{\gamma_1((m-1)\eta_1 + (m-2)\mu)}{2} \|\mathbf{x}^* - \mathbf{x}^0\|^2 - \frac{(m-1)\gamma_s(\eta_s + \mu)}{2} \|\mathbf{x}^* - \mathbf{x}^s\|^2 \right\} \\ & + \mathbb{E}_s \left\{ \gamma_1\tau_1 V_h(y^*, y^0) - \frac{\gamma_s(\tau_s + 1)\bar{\mu}}{2} V_h(y^*, y^s) \right\} \geq 0. \end{aligned}$$

Now we present the main convergence result of Algorithm 4 in Theorem 3.6, which eliminates the dependence on dual variables and relates directly the successive searching points of RapDual.

THEOREM 3.6. *Let (x^*, y^*) be a saddle point of (3.4), and suppose that the parameters $\{\alpha_t\}$, $\{\tau_t\}$, $\{\eta_t\}$, and $\{\gamma_t\}$ are set as in (3.10) and (3.11), and $\tilde{\alpha}_t = \alpha$. Then, for any $s \geq 1$, we have*

$$\mathbb{E}_s \{ \|\mathbf{x}^s - \mathbf{x}^*\|^2 + \|x_m^s - x_m^*\|^2 \} \leq \alpha^s \mathcal{M} (\|\mathbf{x}^0 - \mathbf{x}^*\|^2 + \|x_m^0 - x_m^*\|^2),$$

where $x_m^* = \arg \min_{x_m \in \mathbb{R}^n} \{ \psi_m(x_m) + \langle x_m, y^* \rangle \}$ and $\mathcal{M} = 2\hat{L}/\mu$.

Proof. It is easy to check that (3.10) and (3.11) satisfy conditions (3.17), (3.18), (3.19), (3.20), and (3.21) when $\mu, \bar{\mu} > 0$. Then we have

$$\begin{aligned} \mathbb{E}_s & \left\{ \frac{(m-1)\gamma_s(\eta_s+\mu)}{2} \|\mathbf{x}^s - \mathbf{x}^*\|^2 + \frac{\gamma_s(\tau_s+1)\bar{\mu}}{2} V_h(y^s, y^*) \right\} \\ & \leq \frac{\gamma_1((m-1)\eta_1+(m-2)\mu)}{2} \|\mathbf{x}^0 - \mathbf{x}^*\|^2 + \gamma_1\tau_1 V_h(y^0, y^*). \end{aligned}$$

Therefore, by plugging in those values in (3.10) and (3.11), we have

$$(3.23) \quad \mathbb{E}_s [\mu \|\mathbf{x}^s - \mathbf{x}^*\|^2 + V_h(y^s, y^*)] \leq \alpha^s [\mu \|\mathbf{x}^0 - \mathbf{x}^*\|^2 + 2V_h(y^0, y^*)].$$

Since $h(y)$ has $1/\mu$ -Lipschitz continuous gradients and is $1/L$ -strongly convex, we obtain

$$(3.24) \quad V_h(y^s, y^*) \geq \frac{\mu}{2} \|\nabla h(y^s) - \nabla h(y^*)\|^2 = \frac{\mu}{2} \|\mathbf{x}^s - \mathbf{x}^*\|^2,$$

$$(3.25) \quad V_h(y^0, y^*) \leq \frac{\hat{L}}{2} \|\nabla h(y^0) - \nabla h(y^*)\|^2 = \frac{\hat{L}}{2} \|\mathbf{x}^0 - \mathbf{x}^*\|^2.$$

Combining (3.23), (3.24), and (3.25), we have

$$\mathbb{E}_s \{ \|\mathbf{x}^s - \mathbf{x}^*\|^2 + \|x_m^s - x_m^*\|^2 \} \leq \alpha^s \mathcal{M} (\|\mathbf{x}^0 - \mathbf{x}^*\|^2 + \|x_m^0 - x_m^*\|^2). \quad \square$$

The above theorem shows that subproblem (3.5) can be solved efficiently by Algorithm 4 with a linear rate of convergence. In fact, we need not solve it too accurately. With a fixed and relatively small number of iterations s Algorithm 4 can still converge, as shown by the following lemma.

LEMMA 3.7. *Let the inner iteration number $s \geq \lceil -\log \mathcal{M} / \log \alpha \rceil$ with $\mathcal{M} = 4 + 2L/\mu$ be given. Also let the iterates $(\mathbf{x}^\ell, x_m^\ell)$ for $\ell = 1, \dots, k$ be generated by Algorithm 3 and $\hat{\ell}$ be randomly selected from $[k]$. Then*

$$\mathbb{E}(\|\mathbf{x}_*^\ell - \bar{\mathbf{x}}^{\ell-1}\|^2 + \|x_{m*}^\ell - \bar{x}_m^{\ell-1}\|^2) \leq \frac{1}{k\mu(1-M\alpha^s)} \{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\},$$

$$\mathbb{E}(\|\mathbf{x}_*^\ell - \bar{\mathbf{x}}^\ell\|^2 + \|x_{m*}^\ell - \bar{x}_m^\ell\|^2) \leq \frac{M\alpha^s}{k\mu(1-M\alpha^s)} \{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\},$$

where (\mathbf{x}^*, x_m^*) and $(\mathbf{x}_*^\ell, x_{m*}^\ell)$ are the optimal solutions to (1.4) and the ℓ th subproblem (3.3), respectively.

Proof. According to Theorem 3.6, we have

$$(3.26) \quad \mathbb{E} (\|\bar{\mathbf{x}}^\ell - \mathbf{x}_*^\ell\|^2 + \|\bar{x}_m^\ell - x_{m*}^\ell\|^2) \leq \alpha^s \mathcal{M} (\|\bar{\mathbf{x}}^{\ell-1} - \mathbf{x}_*^{\ell-1}\|^2 + \|\bar{x}_m^{\ell-1} - x_{m*}^{\ell-1}\|^2).$$

Let us denote $(\mathbf{x}_*^0, x_{m*}^0) = (\bar{\mathbf{x}}^0, \bar{x}_m^0)$ and by selection, it is feasible to subproblem (3.3) when $\ell = 1$. Since $(\mathbf{x}_*^\ell, x_{m*}^\ell)$ is optimal and $(\mathbf{x}_*^{\ell-1}, x_{m*}^{\ell-1})$ is feasible to the ℓ th subproblem, we have

$$\psi^\ell(\mathbf{x}_*^\ell) + \psi_m^\ell(x_{m*}^\ell) \leq \psi^\ell(\mathbf{x}_*^{\ell-1}) + \psi_m^\ell(x_{m*}^{\ell-1}).$$

Plugging in the definition of ψ^ℓ and ψ_m^ℓ in the above inequality, and summing up from $\ell = 1$ to k , we have

$$\begin{aligned} \sum_{\ell=1}^k [f(\mathbf{x}_*^\ell) + f_m(x_{m*}^\ell) + \mu(\|\mathbf{x}_*^\ell - \bar{\mathbf{x}}^{\ell-1}\|^2 + \|x_{m*}^\ell - \bar{x}_m^{\ell-1}\|^2)], \\ (3.27) \quad \sum_{\ell=1}^k \leq [f(\mathbf{x}_*^{\ell-1}) + f_m(x_{m*}^{\ell-1}) + \mu(\|\mathbf{x}_*^{\ell-1} - \bar{\mathbf{x}}^{\ell-1}\|^2 + \|x_{m*}^{\ell-1} - \bar{x}_m^{\ell-1}\|^2)]. \end{aligned}$$

Combining (3.26) and (3.27) and noticing that $(\mathbf{x}_*^0, x_{m^*}^0) = (\bar{\mathbf{x}}^0, \bar{x}_m^0)$, we have

$$\begin{aligned}
 & \mu \sum_{\ell=1}^k \mathbb{E}(\|\mathbf{x}_*^\ell - \bar{\mathbf{x}}^{\ell-1}\|^2 + \|x_{m^*}^\ell - \bar{x}_m^{\ell-1}\|^2) \\
 & \leq \sum_{\ell=1}^k \{f(\mathbf{x}_*^{\ell-1}) + f_m(x_{m^*}^{\ell-1}) - [f(\mathbf{x}_*) + f_m(x_{m^*}^{\ell})]\} \\
 & \quad + \mu \sum_{\ell=1}^k \mathbb{E}(\|\mathbf{x}_*^\ell - \bar{\mathbf{x}}^\ell\|^2 + \|x_{m^*}^\ell - \bar{x}_m^\ell\|^2) \\
 & \leq f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)] \\
 (3.28) \quad & \quad + \mu \alpha^s \mathcal{M} \sum_{\ell=1}^k \mathbb{E}(\|\mathbf{x}_*^\ell - \bar{\mathbf{x}}^{\ell-1}\|^2 + \|x_{m^*}^\ell - \bar{x}_m^{\ell-1}\|^2).
 \end{aligned}$$

In view of (3.28) and (3.26), we have

$$\begin{aligned}
 & \sum_{\ell=1}^k \mathbb{E}(\|\mathbf{x}_*^\ell - \bar{\mathbf{x}}^{\ell-1}\|^2 + \|x_{m^*}^\ell - \bar{x}_m^{\ell-1}\|^2) \\
 & \leq \frac{1}{\mu(1-M\alpha^s)} \{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\} \\
 & \sum_{\ell=1}^k \mathbb{E}(\|\mathbf{x}_*^\ell - \bar{\mathbf{x}}^\ell\|^2 + \|x_{m^*}^\ell - \bar{x}_m^\ell\|^2) \\
 & \leq \frac{\mathcal{M}\alpha^s}{\mu(1-M\alpha^s)} \{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\},
 \end{aligned}$$

which, in view of the fact that $\hat{\ell}$ is chosen randomly in $[k]$, implies our results. \square

Now we are ready to prove the results in Theorem 3.2 with all the results proved above.

Proof of Theorem 3.2. By the optimality condition of the $\hat{\ell}$ th subproblem (3.3), there exists some λ^* such that

$$\begin{aligned}
 & \nabla \psi^{\hat{\ell}}(\mathbf{x}_*^{\hat{\ell}}) + \mathbf{A}^\top \lambda^* \in -N_X(\mathbf{x}_*^{\hat{\ell}}), \\
 & \nabla \psi_m^{\hat{\ell}}(x_{m^*}^{\hat{\ell}}) + \lambda^* = 0, \\
 (3.29) \quad & \mathbf{A} \mathbf{x}_*^{\hat{\ell}} + x_{m^*}^{\hat{\ell}} = \mathbf{b}.
 \end{aligned}$$

Plugging in the definition of $\psi^{\hat{\ell}}$ and $\psi_m^{\hat{\ell}}$, we have

$$(3.30) \quad \nabla f^{\hat{\ell}}(\mathbf{x}_*^{\hat{\ell}}) + 2\mu(\mathbf{x}_*^{\hat{\ell}} - \bar{\mathbf{x}}^{\hat{\ell}-1}) + \mathbf{A}^\top \lambda^* \in -N_X(\mathbf{x}_*^{\hat{\ell}}),$$

$$(3.31) \quad \nabla f_m^{\hat{\ell}}(x_{m^*}^{\hat{\ell}}) + 2\mu(x_{m^*}^{\hat{\ell}} - \bar{x}_m^{\hat{\ell}-1}) + \lambda^* = 0.$$

Now we are ready to evaluate the quality of the solution $(\bar{\mathbf{x}}^{\hat{\ell}}, \bar{x}_m^{\hat{\ell}})$. In view of (3.30) and Lemma 3.7, we have

$$\begin{aligned}
 & \mathbb{E} \left[d(\nabla f^{\hat{\ell}}(\mathbf{x}_*^{\hat{\ell}}) + \mathbf{A}^\top \lambda^*, -N_X(\mathbf{x}_*^{\hat{\ell}})) \right]^2 \\
 & \leq \mathbb{E} \|2\mu(\mathbf{x}_*^{\hat{\ell}} - \bar{\mathbf{x}}^{\hat{\ell}-1})\|^2 \\
 & \leq \frac{4\mu}{k(1-\mathcal{M}\alpha^s)} \{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\} \\
 & \leq \frac{8\mu}{k} \{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\}.
 \end{aligned}$$

Similarly, due to (3.31) and Lemma 3.7, we have

$$\begin{aligned}\mathbb{E}\|\nabla f_m^{\hat{\ell}}(x_m^{\hat{\ell}}) + \lambda^*\|^2 &= \mathbb{E}\|\nabla f_m^{\hat{\ell}}(x_m^{\hat{\ell}}) - \nabla f_m^{\hat{\ell}}(x_{m^*}^{\hat{\ell}}) - 2\mu(x_{m^*}^{\hat{\ell}} - x_m^{\hat{\ell}-1})\|^2 \\ &\leq 2\mathbb{E}\{\|\nabla f_m^{\hat{\ell}}(x_m^{\hat{\ell}}) - \nabla f_m^{\hat{\ell}}(x_{m^*}^{\hat{\ell}})\|^2 + 4\mu^2\|x_{m^*}^{\hat{\ell}} - x_m^{\hat{\ell}-1}\|^2\} \\ &\leq \mathbb{E}\left\{18\mu^2\|x_{m^*}^{\hat{\ell}} - \bar{x}_m^{\hat{\ell}}\|^2 + 8\mu^2\|x_{m^*}^{\hat{\ell}} - \bar{x}_m^{\hat{\ell}-1}\|^2\right\} \\ &\leq \mu^2(8 + 18\mathcal{M}\alpha^s)\mathbb{E}\{\|\mathbf{x}^{\hat{\ell}} - \bar{\mathbf{x}}^{\hat{\ell}-1}\|^2 + \|x_{m^*}^{\hat{\ell}} - \bar{x}_m^{\hat{\ell}-1}\|^2\} \\ &\leq \frac{2\mu(4+9\mathcal{M}\alpha^s)}{k(1-\mathcal{M}\alpha^s)}\mathbb{E}\{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\} \\ &\leq \frac{34\mu}{k}\{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\}.\end{aligned}$$

By Lemma 3.7, we have

$$\begin{aligned}\mathbb{E}\|\mathbf{x}^{\hat{\ell}} - \mathbf{x}_*^{\hat{\ell}}\|^2 &\leq \frac{\mathcal{M}\alpha^s}{k\mu(1-\mathcal{M}\alpha^s)}\{f(\bar{\mathbf{x}}^0) + f_m(x_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\} \\ &\leq \frac{2\mathcal{M}\alpha^s}{k\mu}\{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\} \\ &\leq \frac{2\mu}{kL^2}\{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\}.\end{aligned}$$

Combining (3.29) and Lemma 3.7, we have

$$\begin{aligned}\mathbb{E}\|\mathbf{A}\bar{\mathbf{x}}^{\hat{\ell}} + \bar{x}_m^{\hat{\ell}} - \mathbf{b}\|^2 &= \mathbb{E}\|\mathbf{A}(\bar{\mathbf{x}}^{\hat{\ell}} - \mathbf{x}_*^{\hat{\ell}}) + \bar{x}_m^{\hat{\ell}} - x_{m^*}^{\hat{\ell}}\|^2 \\ &\leq 2\mathbb{E}\{\|\mathbf{A}\|^2\|\bar{\mathbf{x}}^{\hat{\ell}} - \mathbf{x}_*^{\hat{\ell}}\|^2 + \|\bar{x}_m^{\hat{\ell}} - x_{m^*}^{\hat{\ell}}\|^2\} \\ &\leq 2(\|\mathbf{A}\|^2 + 1)\mathbb{E}\{\|\bar{\mathbf{x}}^{\hat{\ell}} - \mathbf{x}_*^{\hat{\ell}}\|^2 + \|\bar{x}_m^{\hat{\ell}} - x_{m^*}^{\hat{\ell}}\|^2\} \\ &\leq \frac{2(\|\mathbf{A}\|^2 + 1)\mathcal{M}\alpha^s}{k\mu(1-\mathcal{M}\alpha^s)}\{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\} \\ &\leq \frac{2(\|\mathbf{A}\|^2 + 1)\mu}{kL^2}\{f(\bar{\mathbf{x}}^0) + f_m(\bar{x}_m^0) - [f(\mathbf{x}^*) + f_m(x_m^*)]\}.\quad \square\end{aligned}$$

4. Numerical experiments. In this section, we report some preliminary numerical results for both RapGrad and RapDual and demonstrate their potential advantages in subsections 4.1 and 4.2, respectively.

4.1. Nonconvex finite-sum optimization. We consider the least square problem with the SCAD penalty as a testing problem. SCAD has been proved in [7] to be efficient in variable selection. While the original SCAD $p_{\lambda,\gamma}$ defined below does not have a smooth gradient at $x = 0$, we can bypass this potential problem by using a small positive number ϵ to obtain a smooth approximation $p_{\lambda,\gamma,\epsilon}$:

$$\begin{aligned}p_{\lambda,\gamma}(x) &= \begin{cases} \lambda|x| & \text{if } |x| \leq \lambda, \\ \frac{2\gamma\lambda|x|-x^2-\lambda^2}{2(\gamma-1)} & \text{if } \lambda < |x| < \gamma\lambda, \\ \frac{\lambda^2(\gamma+1)}{2} & \text{if } |x| \geq \gamma\lambda, \end{cases} \\ p_{\lambda,\gamma,\epsilon}(x) &= \begin{cases} \lambda(x^2 + \epsilon)^{\frac{1}{2}} & \text{if } (x^2 + \epsilon)^{\frac{1}{2}} \leq \lambda, \\ \frac{2\gamma\lambda(x^2+\epsilon)^{\frac{1}{2}}-(x^2+\epsilon)-\lambda^2}{2(\gamma-1)} & \text{if } \lambda < (x^2 + \epsilon)^{\frac{1}{2}} < \gamma\lambda, \\ \frac{\lambda^2(\gamma+1)}{2} & \text{if } (x^2 + \epsilon)^{\frac{1}{2}} \geq \gamma\lambda, \end{cases}\end{aligned}$$

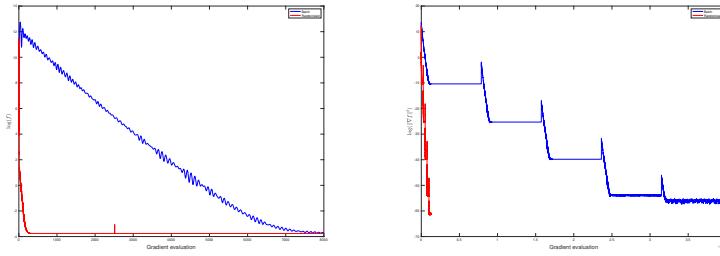


FIG. 4.1. Batch and randomized versions of RapGrad on SCAD-penalized least squares. Left figure: Comparison on objective value f . Right figure: Comparison on square of gradient norm $\|\nabla f\|^2$.

where $\gamma > 2$, $\lambda > 0$, and $\epsilon > 0$ are given. Using $p_{\lambda,\gamma,\epsilon}$, our problem of interest is given by

$$\min_{x \in \mathbb{R}^n} \frac{1}{2m} \|Ax - b\|^2 + \frac{\rho}{2} \sum_{i=1}^n p_{\lambda,\gamma,\epsilon}(x_i),$$

which can be viewed as a special case of problem (1.1) with $f_i(x) = \frac{1}{2}(a_i^\top x - b_i)^2 + \frac{\rho}{2} \sum_{i=1}^n p_{\lambda,\gamma,\epsilon}(x_i)$. Here a_i denotes the i th row of A . It is easy to see that assumptions (1.2) and (1.3) are satisfied with $\mu = \rho/[2(\gamma-1)]$ and $L = \rho\lambda\epsilon^{-1/2}/2 + \max_{1 \leq i \leq m} \|a_i\|^2$. Thus the condition number L/μ usually dominates m .

We test Algorithm 1 (RapGrad), both randomized and deterministic batch versions, on some randomly generated data sets with dimension $m = 1000$, $n = 100$. Note all entries of matrix A and 20 uniformly chosen components of \hat{x} are i.i.d. from $N(0,1)$. The remaining variables of \hat{x} are set to 0 and b is computed by $b = A\hat{x}$. The parameters used in $p_{\lambda,\gamma,\epsilon}$ are $\epsilon = 10^{-3}$, $\lambda = 2$, $\gamma = 4$ and the penalty ρ is set to 0.01. Notice that the x -axis represents the number of gradient evaluations divided by m , which is counted as the number of passes to the dataset, i.e., each iteration of randomized gradient computation is $1/m$ pass and the full-gradient computation of counts as 1 pass. As is shown by Figure 4.1, a randomized version can reduce both the objective value and norm of a gradient faster than its batch counterpart in terms of the number of gradient evaluations. We also observe that there are some zig-zag patterns for the batch algorithm in Figure 4.1 (right), which might have been related to some numerical stability issue.

We also compare our RapGrad with the full SVRG in a nonconvex setting (Algorithm 2 in [2]) and AG method in [9]. Notice that both RapGrad and SVRG are randomized algorithms, while AG is a deterministic batch method. For the sake of fairness in comparison, all the parameters in the three algorithms mentioned above are set to their theoretical values without any tuning in our experiments. Figure 4.2 shows our Algorithm 1 not only reduces the function value but also the gradient norm faster than both SVRG and AG.

In fact, our estimate on $s = \lceil -\log(6M/5)/\log \alpha \rceil$ seems to be too pessimistic, and the subproblems are solved to unnecessarily high accuracy. As a result, some spikes show in Figure 4.1, which correspond to the occasions when an inner loop for solving the subproblem completes and a new search point is obtained to update the subproblem. Early termination of the inner loops may help to remove those spikes. Reducing the number of inner iterations may not guarantee the above-mentioned convergence rate theoretically, but may improve the practical performance of RapGrad for this problems in our experiments. From Figures 4.3 and 4.4, we can conclude

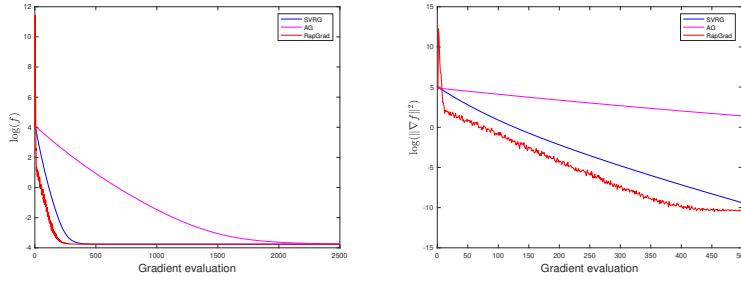


FIG. 4.2. Comparison on function value f and square of gradient norm $\|\nabla f\|^2$ for Algorithm 1, SVRG, and AG. Left figure: Comparison on function value f . Right figure: Comparison on square of gradient norm $\|\nabla f\|^2$.

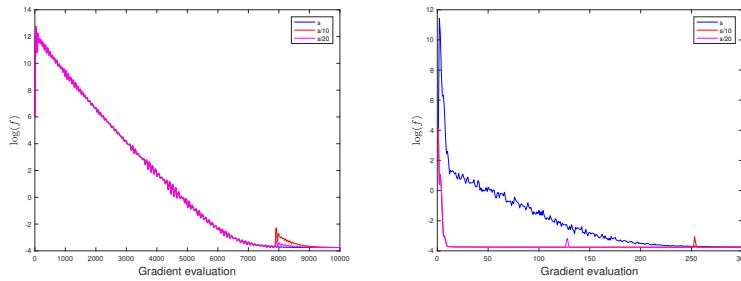


FIG. 4.3. Comparison on objective value for batch and randomized versions of RapGrad, when numbers of inner iterations are s , $s/10$, and $s/20$, respectively. Left figure: Batch version. Right figure: Randomized version.

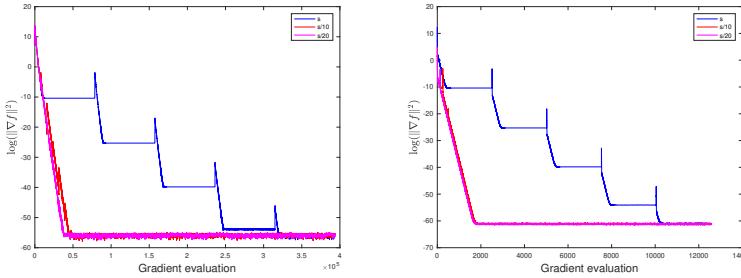


FIG. 4.4. Comparison on square of gradient norm $\|\nabla f\|^2$ for batch and randomized versions of RapGrad when numbers of inner iterations are s , $s/10$, and $s/20$, respectively. Left figure: Batch version. Right figure: Randomized version.

that, by using smaller s , our randomized algorithm is able to reduce f and $\|\nabla f\|^2$ much faster, whereas the batch version converges faster in terms of the gradient norm.

Inspired by the above experiments, we have an efficient way to tune RapGrad to yield better performance. We first run RapGrad with several different numbers of inner iterations s' , for instance, $s' = s$, $s' = s/10$, $s' = s/100$, for a fixed number, say, 100, of passes through the dataset, then we use the best s' corresponding to the smallest norm of gradient as the actual s for the tuned RapGrad. In Table 4.1,

TABLE 4.1

Comparison on numbers of passes to the dataset with stopping criteria $\|\nabla f\|^2 < 10^{-10}$ and maximal pass 3×10^4 .

	RapGrad	RapGrad_tuned	SVRG	AG
$m = 1000, n = 100$	2850	502	1143	3×10^4
$m = 1000, n = 300$	4894	874	5493	3×10^4
$m = 1000, n = 500$	11299	1165	19029	3×10^4
$m = 800, n = 100$	3113	559	1245	3×10^4
$m = 800, n = 300$	5467	970	7743	3×10^4
$m = 800, n = 500$	12673	1290	3×10^4	3×10^4
$m = 600, n = 100$	3735	667	1752	3×10^4
$m = 600, n = 300$	10978	1137	13638	3×10^4
$m = 600, n = 500$	14965	490	3×10^4	3×10^4

we compare the RapGrad without tuning, tuned RapGrad, SVRG as well as AG on testing problems of different sizes, with stoping criteria $\|\nabla f\|^2 < 10^{-10}$ and maximal pass 3×10^4 . The table shows that this simple tuning technique is able to bring a huge performance improvement. An interesting observation is that RapGrad without tuning is more likely to outperform SVRG when n is large relative to m .

4.2. Nonconvex multiblock optimization. We consider the following compressed sensing problem to test the performance of RapDual:

$$\begin{aligned} \min_{x_i \in X_i} \quad & \sum_{i=1}^m \mathcal{P}_{\lambda, \gamma, \epsilon}(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^m A_i x_i = b, \end{aligned}$$

where each $x_i = (x_i^1, \dots, x_i^{d_i})$ is a vector of dimension d_i , and $\mathcal{P}_{\lambda, \gamma, \epsilon}(x_i) = \sum_{j=1}^{d_i} p_{\lambda, \gamma, \epsilon}(x_i^j)$. Instead of using the ℓ_1 norm as the objective function, we replace it with the smoothed SCAD function, which is also capable of finding sparse solutions. Since the smoothed SCAD function is separable in each component, we can easily identify an invertible $n \times n$ submatrix from $[A_1, A_2, \dots, A_m]$. Without loss of generality, we assume the last block A_m is invertible. If we multiply both sides of the linear equation $\sum_{i=1}^m A_i x_i = b$ by A_m^{-1} , we can reformulate the above problem into (1.4), which is ready to be solved by RapDual.

The numerical experiments were performed on some randomly generated data sets of size $m = 1001, n = 100$. The first 1000 coefficient matrices $A_i, 1 \leq i \leq 1000$, are of size 100×1 and the last block A_{1001} is an 100×100 identity matrix. $A_i, 1 \leq i \leq 1000$ are sparse matrices with sparsity level 0.1, and all nonzero elements of A_i and 200 uniformly chosen components of \hat{x} are i.i.d from $N(0, 1)$. The remaining variables of \hat{x} are set to 0 and $b = \sum_{i=1}^m A_i \hat{x}_i$. The parameters used in $p_{\lambda, \gamma, \epsilon}$ are exactly the same as the first problem, i.e., $\epsilon = 10^{-3}, \lambda = 2, \gamma = 4$. From Figure 4.5, we can conclude that the randomized version converges faster than its batch counterpart, in terms of the number of primal block updates required to reduce the objective value and infeasibility. We also compare our randomized algorithm with Algorithm 4 in [14], with $\rho = L^2, L^2/10, L^2/20$. Note $\rho = L^2$ only guarantees asymptotic convergence. The results in Figure 4.6 show that our algorithm can reduce the objective value faster than ADMM. As for the feasibility, both algorithms yield solutions that have a quite tiny constraint violation. It is worth noting that the convergence of ADMM can very much depend on the update order [4, 27] and also that RapDual requires the computation of A_m^{-1} in its current form. Similar to RapGrad, if we reduce the

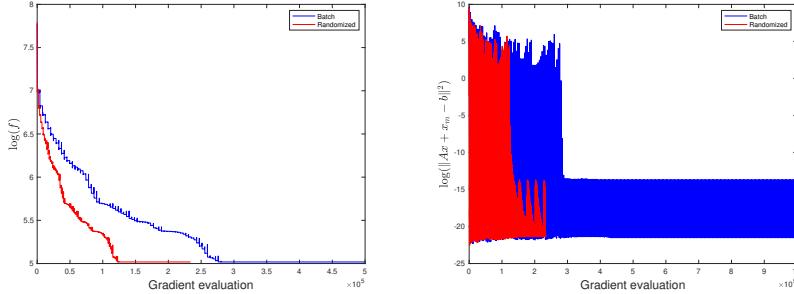


FIG. 4.5. Batch and randomized versions of Algorithm 3 on compressed sensing problem with smoothed SCAD objective. Left figure: Comparison on objective value f . Right figure: Comparison on feasibility $\|Ax + xm - b\|^2$.

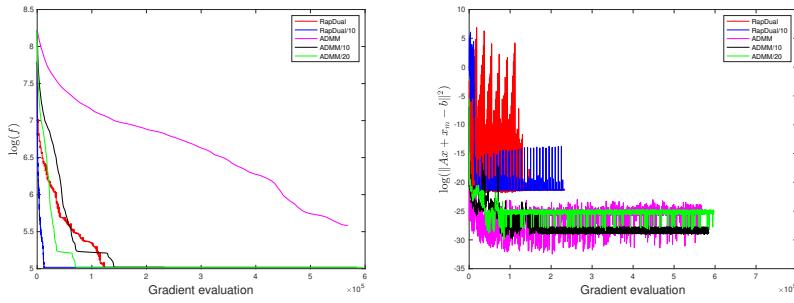


FIG. 4.6. Comparison on objective value f and feasibility $\|Ax + xm - b\|^2$ of RapDual and ADMM in [14]. Left figure: Comparison on objective value f . Right figure: Comparison on feasibility $\|Ax + xm - b\|^2$.

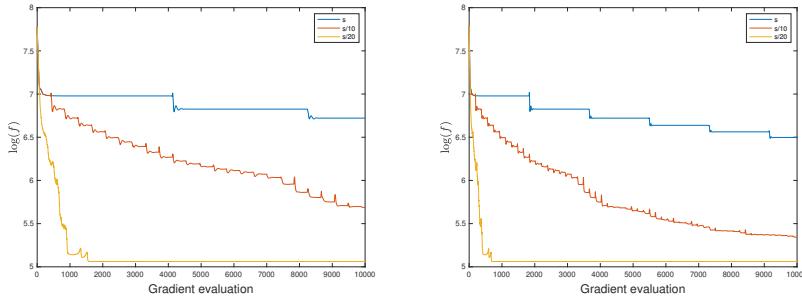


FIG. 4.7. Comparison on objective value f of batch and randomized versions of RapDual when numbers of inner iterations s per subproblem are s , $s/10$, and $s/20$, respectively. Left figure: Batch version. Right figure: Randomized version.

number of inner iterations s per subproblem by a factor of 10 or 20, we obtain results in Figures 4.7 and 4.8. As we can see, the total number of primal block updates needed to yield a good solution, in terms of both objective value and feasibility, can be much smaller when inner loops are terminated early.

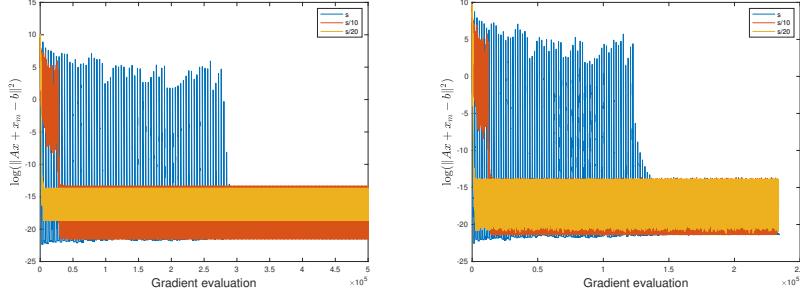


FIG. 4.8. Comparison on feasibility $\|Ax + x_m - b\|^2$ of batch and randomized versions of RapDual when numbers of inner iterations are s , $s/10$, and $s/20$, respectively. Left figure: Batch version. Right figure: Randomized version.

5. Concluding remarks. In this paper, we propose a new RapGrad method for solving nonconvex finite-sum problems (1.1) and a new RapGrad method for nonconvex multiblock problems (1.4), respectively. We demonstrate that for problem (1.1) with large condition number, our RapGrad has a much better convergence rate, in terms of dependence on the large number m , than the state-of-art nonconvex SVRG or SAGA, as well as the AG method for nonconvex optimization. Moreover, we show that our RapDual method incorporated with randomization techniques can significantly save the number of primal block updates up to a factor of \sqrt{m} more than the deterministic batch methods for solving problems (1.4). The potential advantages of RapGrad and RapDual are also demonstrated through our preliminary numerical experiments.

We observe that the main ideas of this RapDual, i.e., using the proximal point method to transform the nonconvex multiblock problem into a series of convex subproblems, and using the randomized dual method on them, can be applied for solving much more general multiblock optimization problems for which there does not exist an invertible block. In this more general case, the saddle-point subproblems will only be strongly convex in the primal space, but not in the dual space. Therefore, the complexity of solving the subproblem will only be sublinear, and as a consequence, the overall complexity will be much worse than $\mathcal{O}(1/\epsilon)$. It will be interesting to study how much benefit one can obtain by using randomized algorithms in this more general case. We leave this as an interesting topic for future research. It is also worth noting that the proposed RapGrad and RapDual implicitly assume that the parameter μ is known, or a (tight) upper bound of it can be obtained. While the values of μ for the problems considered in our numerical experiments can be tightly estimated, it will be interesting to see if we can adaptively estimate the value of μ in these algorithms applied to solve more general problems.

Appendix A. Proof of Lemma 2.4.

Proof. By convexity of ψ and optimality of x^* , we have

$$\begin{aligned}
 Q_t &:= \varphi(x^t) + \psi(x^*) + \langle \nabla \psi(x^*), x^t - x^* \rangle \\
 &\quad - [\varphi(x^*) + \frac{1}{m} \sum_{i=1}^m (\psi_i(\hat{x}_i^t) + \langle \nabla \psi_i(\hat{x}_i^t), x^* - \hat{x}_i^t \rangle)] \\
 &\geq \varphi(x^t) + \psi(x^*) + \langle \nabla \psi(x^*), x^t - x^* \rangle - [\varphi(x^*) + \psi(x^*)] \\
 (A.1) \quad &= \varphi(x^t) - \varphi(x^*) + \langle \nabla \psi(x^*), x^t - x^* \rangle \geq \langle \nabla \varphi(x^*) + \nabla \psi(x^*), x^t - x^* \rangle \geq 0.
 \end{aligned}$$

For notational convenience, let $\Psi(x, z) := \psi(x) - \psi(z) - \langle \nabla \psi(z), x - z \rangle$,

$$(A.2) \quad Q_t = \varphi(x^t) - \varphi(x^*) + \langle \frac{1}{m} \sum_{i=1}^m \tilde{y}_i^t, x^t - x^* \rangle + \delta_1^t + \delta_2^t,$$

$$\delta_1^t := \psi(x^*) - \langle \nabla \psi(x^*), x^* \rangle$$

$$- \frac{1}{m} \sum_{i=1}^m [\psi_i(\hat{x}_i^t) - \langle \nabla \psi_i(\hat{x}_i^t), \hat{x}_i^t \rangle + \langle \nabla \psi_i(\hat{x}_i^t) - \nabla \psi_i(x^*), \tilde{x}^t \rangle]$$

$$= \frac{1}{m} \sum_{i=1}^m [\tau_t \Psi(\underline{x}_i^{t-1}, x^*) - (1 + \tau_t) \Psi(\hat{x}_i^t, x^*) - \tau_t \Psi(\underline{x}_i^{t-1}, \hat{x}_i^t)],$$

$$(A.3) \quad \delta_2^t := \frac{1}{m} \sum_{i=1}^m [\langle \nabla \psi_i(\hat{x}_i^t) - \nabla \psi_i(x^*), \tilde{x}^t \rangle$$

$$- \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^t \rangle + \langle \tilde{y}_i^t - \nabla \psi_i(\hat{x}_i^t), x^* \rangle].$$

In view of (2.13), we have

$$(A.4) \quad \mathbb{E}_{i_t} \delta_2^t = \frac{1}{m} \sum_{i=1}^m \mathbb{E}_{i_t} [\langle \nabla \psi_i(\hat{x}_i^t) - \nabla \psi_i(x^*), \tilde{x}^t \rangle$$

$$- \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^t \rangle + \langle \tilde{y}_i^t - \nabla \psi_i(\hat{x}_i^t), x^* \rangle]$$

$$= \frac{1}{m} \sum_{i=1}^m \mathbb{E}_{i_t} \langle \tilde{y}_i^t - \nabla \psi_i(x^*), \tilde{x}^t - x^t \rangle.$$

Multiplying each Q_t by a nonnegative γ_t and summing them up, we obtain

$$(A.5) \quad \mathbb{E}_s [\sum_{t=1}^s \gamma_t Q_t] \leq \mathbb{E}_s \left\{ \sum_{t=1}^s \gamma_t [\eta_t V_\varphi(x^*, x^{t-1}) - (1 + \eta_t) V_\varphi(x^*, x^t) - \eta_t V_\varphi(x^t, x^{t-1})] \right\}$$

$$+ \mathbb{E}_s \left\{ \sum_{t=1}^s \sum_{i=1}^m [\gamma_t (1 + \tau_t - \frac{1}{m}) \Psi(\underline{x}_i^{t-1}, x^*) - \gamma_t (1 + \tau_t) \Psi(\underline{x}_i^t, x^*)] \right\}$$

$$+ \mathbb{E}_s \left\{ \sum_{t=1}^s \sum_{i=1}^m \gamma_t [\frac{1}{m} \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^t - x^t \rangle - \tau_t \Psi(\underline{x}_i^{t-1}, \hat{x}_i^t)] \right\}$$

$$\leq \mathbb{E}_s [\gamma_1 \eta_1 V_\varphi(x^*, x^0) - (1 + \eta_s) V_\varphi(x^*, x^s)]$$

$$+ \mathbb{E}_s \left\{ \sum_{i=1}^m [\gamma_1 (1 + \tau_1 - \frac{1}{m}) \Psi(\underline{x}_i^0, x^*) - \gamma_s (1 + \tau_s) \Psi(\underline{x}_i^s, x^*)] \right\} - \mathbb{E}_s [\sum_{t=1}^s \gamma_t \delta_t],$$

where

$$(A.6) \quad \delta_t := \eta_t V_\varphi(x^t, x^{t-1}) - \sum_{i=1}^m [\frac{1}{m} \langle \tilde{y}_i^t - \nabla \psi_i(x^*), \tilde{x}^t - x^t \rangle - \tau_t \Psi(\underline{x}_i^{t-1}, \hat{x}_i^t)]$$

$$= \eta_t V_\varphi(x^t, x^{t-1}) - \frac{1}{m} \sum_{i=1}^m \langle \tilde{y}_i^t - \nabla \psi_i(x^*), \tilde{x}^t - x^t \rangle + \tau_t \Psi(\underline{x}_i^{t-1}, \hat{x}_i^t),$$

the first inequality follows from (A.2), (A.3), (2.28), (A.4), and Lemma 2.3, and the second inequality is implied by (2.15) and (2.16).

By the definition of \tilde{x} in (2.3), we have

$$(A.7) \quad \frac{1}{m} \sum_{i=1}^m \langle \tilde{y}_i^t - \nabla \psi_i(x^*), \tilde{x}^t - x^t \rangle$$

$$= \frac{1}{m} \sum_{i=1}^m [\langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^{t-1} - x^t \rangle$$

$$- \alpha_t \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^{t-2} - x^{t-1} \rangle]$$

$$= \frac{1}{m} \sum_{i=1}^m [\langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^{t-1} - x^t \rangle$$

$$- \alpha_t \langle \tilde{y}_i^{t-1} - \nabla \psi_i(x^*), x^{t-2} - x^{t-1} \rangle - \alpha_t \langle \tilde{y}_i^t - \tilde{y}_i^{t-1}, x^{t-2} - x^{t-1} \rangle]$$

$$= \frac{1}{m} \sum_{i=1}^m [\langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^{t-1} - x^t \rangle - \alpha_t \langle \tilde{y}_i^{t-1} - \nabla \psi_i(x^*), x^{t-2} - x^{t-1} \rangle]$$

$$- \alpha_t \langle \nabla \psi_{i_t}(\underline{x}_{i_t}^t) - \nabla \psi_{i_t}(\underline{x}_{i_t}^{t-1}), x^{t-2} - x^{t-1} \rangle$$

$$- (1 - \frac{1}{m}) \alpha_t \langle \nabla \psi_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2}) - \nabla \psi_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1}), x^{t-2} - x^{t-1} \rangle.$$

From the relation (2.18) and the fact $x^{-1} = x^0$, we have

$$\begin{aligned}
 (A.8) \quad & \sum_{t=1}^s \gamma_t \frac{1}{m} \sum_{i=1}^m [\langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^{t-1} - x^t \rangle - \alpha_t \langle \tilde{y}_i^{t-1} - \nabla \psi_i(x^*), x^{t-2} - x^{t-1} \rangle] \\
 &= \gamma_s \frac{1}{m} \sum_{i=1}^m \langle \tilde{y}_i^s - \nabla \psi_i(x^*), x^{s-1} - x^s \rangle \\
 &= \gamma_s \frac{1}{m} \sum_{i=1}^m \langle \nabla \psi_i(\underline{x}_i^s) - \nabla \psi_i(x^*), x^{s-1} - x^s \rangle \\
 &\quad + \gamma_s \sum_{i=1}^m \left(1 - \frac{1}{m}\right) \langle \nabla \psi_i(\underline{x}_i^s) - \nabla \psi_i(\underline{x}_i^{s-1}), x^{s-1} - x^s \rangle \\
 &= \gamma_s \frac{1}{m} \sum_{i=1}^m \langle \nabla \psi_i(\underline{x}_i^s) - \nabla \psi_i(x^*), x^{s-1} - x^s \rangle \\
 &\quad + \gamma_s \left(1 - \frac{1}{m}\right) \langle \nabla \psi_{i_s}(\underline{x}_{i_s}^s) - \nabla \psi_{i_s}(\underline{x}_{i_s}^{s-1}), x^{s-1} - x^s \rangle.
 \end{aligned}$$

Now we are ready to bound the last term in (A.5) as follows:

$$\begin{aligned}
 (A.9) \quad & \sum_{t=1}^s \gamma_t \delta_t \\
 &\stackrel{(a)}{=} \sum_{t=1}^s \gamma_t [\eta_t V_\varphi(x^t, x^{t-1}) - \frac{1}{m} \sum_{i=1}^m \langle \tilde{y}_i^t - \nabla \psi_i(x^*), \tilde{x}^t - x^t \rangle + \tau_t \Psi(\underline{x}_{i_t}^{t-1}, \underline{x}_{i_t}^t)] \\
 &\stackrel{(b)}{=} \sum_{t=1}^s \gamma_t [\eta_t V_\varphi(x^t, x^{t-1}) + \alpha_t \langle \nabla \psi_{i_t}(\underline{x}_{i_t}^t) - \nabla \psi_{i_t}(\underline{x}_{i_t}^{t-1}), x^{t-2} - x^{t-1} \rangle \\
 &\quad + \left(1 - \frac{1}{m}\right) \alpha_t \langle \nabla \psi_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2}) - \nabla \psi_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1}), x^{t-2} - x^{t-1} \rangle + \tau_t \Psi(\underline{x}_{i_t}^{t-1}, \underline{x}_{i_t}^t)] \\
 &\quad + \gamma_s \frac{1}{m} \sum_{i=1}^m \langle \nabla \psi_i(\underline{x}_i^s) - \nabla \psi_i(x^*), x^{s-1} - x^s \rangle \\
 &\quad - \gamma_s \left(1 - \frac{1}{m}\right) \langle \nabla \psi_{i_s}(\underline{x}_{i_s}^s) - \nabla \psi_{i_s}(\underline{x}_{i_s}^{s-1}), x^{s-1} - x^s \rangle \\
 &\stackrel{(c)}{\geq} \sum_{t=1}^s \gamma_t \left[\frac{\eta_s r}{2} \|x^t - x^{t-1}\|^2 + \alpha_t \langle \nabla \psi_{i_t}(\underline{x}_{i_t}^t) - \nabla \psi_{i_t}(\underline{x}_{i_t}^{t-1}), x^{t-2} - x^{t-1} \rangle \right. \\
 &\quad + \left(1 - \frac{1}{m}\right) \alpha_t \langle \nabla \psi_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2}) - \nabla \psi_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1}), x^{t-2} - x^{t-1} \rangle \\
 &\quad + \frac{\tau_t}{2\hat{L}} \|\nabla \psi_{i_t}(\underline{x}_{i_t}^t) - \nabla \psi_{i_t}(\underline{x}_{i_t}^{t-1})\|^2 \\
 &\quad - \frac{\gamma_s}{m} \sum_{i=1}^m \langle x^{s-1} - x^s, \nabla \psi_i(\underline{x}_i^s) \nabla \psi_i(x^*) \rangle \\
 &\quad \left. - \gamma_s \left(1 - \frac{1}{m}\right) \langle x^{s-1} - x^s, \nabla \psi_{i_s}(\underline{x}_{i_s}^s) - \nabla \psi_{i_s}(\underline{x}_{i_s}^{s-1}) \rangle \right],
 \end{aligned}$$

where (a) follows from the definition δ_t in (A.6), (b) follows relations (A.7) and (A.8), and (c) follows from the fact that $V_\varphi(x^t, x^{t-1}) \geq \frac{r}{2} \|x^t - x^{t-1}\|^2$, $\Psi(\underline{x}_{i_t}^{t-1}, \underline{x}_{i_t}^t) \geq \frac{1}{2\hat{L}} \|\nabla \psi_{i_t}(\underline{x}_{i_t}^{t-1}) - \nabla \psi_{i_t}(\underline{x}_{i_t}^t)\|^2$.

By properly regrouping the terms on the right-hand side of (A.9), we have

$$\begin{aligned}
 & \sum_{t=1}^s \gamma_t \delta_t \\
 &\geq \gamma_s \left[\frac{\eta_s r}{4} \|x^s - x^{s-1}\|^2 - \frac{1}{m} \sum_{i=1}^m \langle \nabla \psi_i(\underline{x}_i^s) - \nabla \psi_i(x^*), x^{s-1} - x^s \rangle \right] \\
 &\quad + \gamma_s \left[\frac{\eta_s r}{4} \|x^s - x^{s-1}\|^2 - \left(1 - \frac{1}{m}\right) \langle \nabla \psi_{i_s}(\underline{x}_{i_s}^s) - \nabla \psi_{i_s}(\underline{x}_{i_s}^{s-1}), x^{s-1} - x^s \rangle \right. \\
 &\quad \left. + \frac{\tau_s}{4\hat{L}} \|\nabla \psi_{i_s}(\underline{x}_{i_s}^s) - \nabla \psi_{i_s}(\underline{x}_{i_s}^{s-1})\|^2 \right] \\
 &\quad + \sum_{t=2}^s \gamma_t \left[\alpha_t \langle \nabla \psi_{i_t}(\underline{x}_{i_t}^t) - \nabla \psi_{i_t}(\underline{x}_{i_t}^{t-1}), x^{t-2} - x^{t-1} \rangle + \frac{\tau_t}{4\hat{L}} \|\nabla \psi_{i_t}(\underline{x}_{i_t}^t) - \nabla \psi_{i_t}(\underline{x}_{i_t}^{t-1})\|^2 \right] \\
 &\quad + \sum_{t=2}^s \left[\gamma_t \left(1 - \frac{1}{m}\right) \alpha_t \langle \nabla \psi_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1}) - \nabla \psi_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2}), x^{t-2} - x^{t-1} \rangle \right. \\
 &\quad \left. + \frac{\tau_{t-1} \gamma_{t-1}}{4\hat{L}} \|\nabla \psi_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1}) - \nabla \psi_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})\|^2 \right] + \sum_{t=2}^s \frac{\gamma_{t-1} \eta_{t-1} r}{2} \|x^{t-1} - x^{t-2}\|^2
 \end{aligned}$$

$$\begin{aligned}
&\stackrel{(a)}{\geq} \gamma_s \left[\frac{\eta_s r}{4} \|x^s - x^{s-1}\|^2 - \frac{1}{m} \sum_{i=1}^m \langle \nabla \psi_i(\underline{x}_i^s) - \nabla \psi_i(x^*), x^{s-1} - x^s \rangle \right] \\
&\quad + \gamma_s \left(\frac{\eta_s r}{4} - \frac{(m-1)^2 \hat{L}}{m^2 \tau_s} \right) \|x^s - x^{s-1}\|^2 \\
&\quad + \sum_{t=2}^s \left(\frac{\gamma_{t-1} \eta_{t-1} r}{2} - \frac{\gamma_t \alpha_t^2 \hat{L}}{\tau_t} - \frac{(m-1)^2 \gamma_t^2 \alpha_t^2 \hat{L}}{m^2 \gamma_{t-1} \tau_{t-1}} \right) \|x^s - x^{s-1}\|^2 \\
&\stackrel{(b)}{\geq} \gamma_s \left[\frac{\eta_s r}{4} \|x^s - x^{s-1}\|^2 - \frac{1}{m} \sum_{i=1}^m \langle \nabla \psi_i(\underline{x}_i^s) - \nabla \psi_i(x^*), x^{s-1} - x^s \rangle \right],
\end{aligned}$$

where (a) follows from the simple relation that $b\langle u, v \rangle - a\|v\|^2/2 \leq b^2\|u\|^2/(2a)$ for all $a > 0$ and (b) follows from (2.14), (2.17) and (2.18). By using the above inequality, (A.1) and (A.5), we obtain

(A.10)

$$\begin{aligned}
0 &\leq \mathbb{E}_s \left[\gamma_1 \eta_1 V_\varphi(x^*, x^0) - \gamma_s (1 + \eta_s) V_\varphi(x^*, x^s) \right] \\
&\quad + \gamma_s \mathbb{E}_s \left[\frac{1}{m} \sum_{i=1}^m \langle \nabla \psi_i(\underline{x}_i^s) - \nabla \psi_i(x^*), x^{s-1} - x^s \rangle \right] - \frac{\gamma_s \eta_s r}{4} \mathbb{E}_s \|x^s - x^{s-1}\|^2 \\
&\quad + \mathbb{E}_s \left\{ \sum_{i=1}^m [\gamma_1 (1 + \tau_1 - \frac{1}{m}) \Psi(\underline{x}_i^0, x^*) - \gamma_s (1 + \tau_s) \Psi(\underline{x}_i^s, x^*)] \right\} \\
&\stackrel{(a)}{\leq} \mathbb{E}_s [\gamma_1 \eta_1 V_\varphi(x^*, x^0) - \gamma_s (1 + \eta_s) V_\varphi(x^*, x^s)] - \frac{\gamma_s \eta_s r}{4} \mathbb{E}_s \|x^s - x^{s-1}\|^2 \\
&\quad + \mathbb{E}_s \left\{ \sum_{i=1}^m [\gamma_1 (1 + \tau_1 - \frac{1}{m}) \Psi(\underline{x}_i^0, x^*) - \frac{\gamma_s (1 + \tau_s)}{2} \Psi(\underline{x}_i^s, x^*)] \right\} \\
&\quad - \gamma_s \frac{1}{m} \sum_{i=1}^m \mathbb{E}_s \left[\frac{m(1+\tau_s)}{4\hat{L}} \|\nabla \psi_i(\underline{x}_i^s) - \nabla \psi_i(x^*)\|^2 - \langle \nabla \psi_i(\underline{x}_i^s) - \nabla \psi_i(x^*), x^{s-1} - x^s \rangle \right] \\
&\stackrel{(b)}{\leq} \mathbb{E}_s [\gamma_1 \eta_1 V_\varphi(x^*, x^0) - \gamma_s (1 + \eta_s) V_\varphi(x^*, x^s)] - \gamma_s \left[\frac{\eta_s r}{4} - \frac{\hat{L}}{m(1+\tau_s)} \right] \mathbb{E}_s \|x^s - x^{s-1}\|^2 \\
&\quad + \sum_{i=1}^m \mathbb{E}_s [\gamma_1 (1 + \tau_1 - \frac{1}{m}) \Psi(\underline{x}_i^0, x^*) - \frac{\gamma_s (1 + \tau_s)}{2} \Psi(\underline{x}_i^s, x^*)] \\
&\stackrel{(c)}{\leq} \mathbb{E}_s [\gamma_1 \eta_1 V_\varphi(x^*, x^0) - \gamma_s (1 + \eta_s) V_\varphi(x^*, x^s)] \\
&\quad + \frac{\gamma_1 [(1+\tau_1) - \frac{1}{m}] \hat{L}}{2} \sum_{i=1}^m \mathbb{E}_s \|\underline{x}_i^0 - x^*\|^2 - \frac{\mu \gamma_s (1 + \tau_s)}{4} \sum_{i=1}^m \mathbb{E}_s \|\underline{x}_i^s - x^*\|^2,
\end{aligned}$$

where (a) follows from $\Psi(\underline{x}_i^0, x^*) \geq \frac{1}{2\hat{L}} \|\nabla \psi_i(\underline{x}_i^0) - \nabla \psi_i(x^*)\|^2$; (b) follows from the simple relation that $b\langle u, v \rangle - a\|v\|^2/2 \leq b^2\|u\|^2/(2a)$ for all $a > 0$, and (c) follows from (2.19), the strong convexity of ψ_i , and the Lipschitz continuity of $\nabla \psi_i$. This completes the proof. \square

Appendix B. Proof of Lemma 3.5.

Proof. For any $t \geq 1$, since (\mathbf{x}^*, y^*) is a saddle point of (3.4), we have

$$\psi(\hat{\mathbf{x}}^t) - \psi(\mathbf{x}^*) + \langle \mathbf{A}\hat{\mathbf{x}}^t - \mathbf{b}, y^* \rangle - \langle \mathbf{A}\mathbf{x}^* - \mathbf{b}, y^t \rangle + h(y^*) - h(y^t) \geq 0.$$

For nonnegative γ_t , we further obtain

$$\begin{aligned}
&(B.1) \\
&\mathbb{E}_s \left\{ \sum_{t=1}^s \gamma_t [\psi(\hat{\mathbf{x}}^t) - \psi(\mathbf{x}^*) + \langle \mathbf{A}\hat{\mathbf{x}}^t - \mathbf{b}, y^* \rangle - \langle \mathbf{A}\mathbf{x}^* - \mathbf{b}, y^t \rangle + h(y^*) - h(y^t)] \right\} \geq 0.
\end{aligned}$$

According to the optimality conditions of (3.14) and (3.8), respectively, and the

strongly convexity of ψ and h we obtain

$$\begin{aligned} & \psi(\hat{\mathbf{x}}^t) - \psi(\mathbf{x}^*) + \frac{\mu}{2} \|\mathbf{x}^* - \hat{\mathbf{x}}^t\|^2 + \langle \mathbf{A}^\top y^t, \hat{\mathbf{x}}^t - \mathbf{x}^* \rangle \\ & \leq \frac{\eta_t}{2} [\|\mathbf{x}^* - \mathbf{x}^{t-1}\|^2 - \|\mathbf{x}^* - \hat{\mathbf{x}}^t\|^2 - \|\hat{\mathbf{x}}^t - \mathbf{x}^{t-1}\|^2], \\ & h(y^t) - h(y^*) + \langle -\mathbf{A}\tilde{\mathbf{x}}^t + \mathbf{b}, y^t - y^* \rangle \\ & \leq \tau_t V_h(y^*, y^{t-1}) - (\tau_t + 1)V_h(y^*, y^t) - \tau_t V_h(y^t, y^{t-1}). \end{aligned}$$

Combining the above two inequalities with relation (B.1), we have

$$\begin{aligned} & \mathbb{E}_s \left\{ \sum_{t=1}^k \left[\frac{\gamma_t \eta_t}{2} \|\mathbf{x}^* - \mathbf{x}^{t-1}\|^2 - \frac{\gamma_t(\eta_t + \mu)}{2} \|\mathbf{x}^* - \hat{\mathbf{x}}^t\|^2 - \frac{\gamma_t \eta_t}{2} \|\hat{\mathbf{x}}^t - \mathbf{x}^{t-1}\|^2 \right] \right\} \\ & + \mathbb{E}_s \left\{ \sum_{t=1}^s \gamma_t [\tau_t V_h(y^*, y^{t-1}) - (\tau_t + 1)V_h(y^*, y^t) - \tau_t V_h(y^t, y^{t-1})] \right\} \\ & + \mathbb{E}_s [\sum_{t=1}^s \gamma_t \langle \mathbf{A}(\hat{\mathbf{x}}^t - \tilde{\mathbf{x}}^t), y^* - y^t \rangle] \geq 0. \end{aligned}$$

Observe that for $t \geq 1$,

$$\mathbb{E}_{i_t} \{ \langle \mathbf{A}(\hat{\mathbf{x}}^t - \tilde{\mathbf{x}}^t), y^* \rangle \} = \mathbb{E}_{i_t} \{ \langle \mathbf{A}((m-1)\mathbf{x}^t - (m-2)\mathbf{x}^{t-1} - \tilde{\mathbf{x}}^t), y^* \rangle \}.$$

Applying this and the results (3.15), (3.16) in Lemma 3.4, we further have

$$\begin{aligned} 0 & \leq \mathbb{E}_s \left\{ \sum_{t=1}^s \left[\frac{\gamma_t((m-1)\eta_t + (m-2)\mu)}{2} \|\mathbf{x}^* - \mathbf{x}^{t-1}\|^2 - \frac{(m-1)\gamma_t(\eta_t + \mu)}{2} \|\mathbf{x}^* - \mathbf{x}^t\|^2 \right] \right\} \\ & + \mathbb{E}_s \left\{ \sum_{t=1}^s [\gamma_t \tau_t V_h(y^*, y^{t-1}) - \gamma_t(\tau_t + 1)V_h(y^*, y^t)] \right\} + \mathbb{E}_s \left\{ \sum_{t=1}^s \gamma_t \delta_t \right\} \\ & \leq \mathbb{E}_s \left[\frac{\gamma_1((m-1)\eta_1 + (m-2)\mu)}{2} \|\mathbf{x}^* - \mathbf{x}^0\|^2 - \frac{(m-1)\gamma_s(\eta_s + \mu)}{2} \|\mathbf{x}^* - \mathbf{x}^s\|^2 \right] \\ (B.2) \quad & + \mathbb{E}_s [\gamma_1 \tau_1 V_h(y^*, y^0) - \gamma_s(\tau_s + 1)V_h(y^*, y^s)] + \mathbb{E}_s [\sum_{t=1}^s \gamma_t \delta_t], \end{aligned}$$

where

$$\begin{aligned} \delta_t &= -\frac{(m-1)\eta_t}{2} \|x_{i_t}^t - x_{i_t}^{t-1}\|^2 - \tau_t V_h(y^t, y^{t-1}) \\ &+ \langle \mathbf{A}((m-1)\mathbf{x}^t - (m-2)\mathbf{x}^{t-1} - \tilde{\mathbf{x}}^t), y^* - y^t \rangle, \end{aligned}$$

and the second inequality follows from (3.19) and (3.20).

By (3.17) and the definition of $\tilde{\mathbf{x}}^t$ in (3.6) we have

$$\begin{aligned} \sum_{t=1}^s \gamma_t \delta_t &= \sum_{t=1}^s \left[-\frac{(m-1)\gamma_t \eta_t}{2} \|x_{i_t}^t - x_{i_t}^{t-1}\|^2 - \gamma_t \tau_t V_h(y^t, y^{t-1}) \right] \\ &+ \sum_{t=1}^s \gamma_t (m-1) \langle \mathbf{A}(\mathbf{x}^t - \mathbf{x}^{t-1}), y^* - y^t \rangle \\ &- \sum_{t=1}^s \gamma_t (m-1) \tilde{\alpha}_t \langle \mathbf{A}(\mathbf{x}^{t-1} - \mathbf{x}^{t-2}), y^* - y^{t-1} \rangle \\ &- \sum_{t=1}^s \gamma_t (m-1) \tilde{\alpha}_t \langle \mathbf{A}(\mathbf{x}^{t-1} - \mathbf{x}^{t-2}), y^{t-1} - y^t \rangle \\ &= \sum_{t=1}^s \left[-\frac{(m-1)\gamma_t \eta_t}{2} \|x_{i_t}^t - x_{i_t}^{t-1}\|^2 - \gamma_t \tau_t V_h(y^t, y^{t-1}) \right] \\ &+ \gamma_s (m-1) \langle \mathbf{A}(\mathbf{x}^s - \mathbf{x}^{s-1}), y^* - y^s \rangle \\ (B.3) \quad &- \sum_{t=1}^s \gamma_t (m-1) \tilde{\alpha}_t \langle \mathbf{A}(\mathbf{x}^{t-1} - \mathbf{x}^{t-2}), y^{t-1} - y^t \rangle, \end{aligned}$$

where the secondequality follows from (3.18) and the fact that $x^0 = x^{-1}$.

Since $\langle \mathbf{A}(\mathbf{x}^{t-1} - \mathbf{x}^{t-2}), y^{t-1} - y^t \rangle = \langle \mathbf{A}_{i_{t-1}}(x_{i_{t-1}}^{t-1} - x_{i_{t-1}}^{t-2}), y^{t-1} - y^t \rangle \leq \|\mathbf{A}_{i_{t-1}}\| \|x_{i_{t-1}}^{t-1} - x_{i_{t-1}}^{t-2}\| \|y^t - y^{t-1}\|$ and $V_h(y^t, y^{t-1}) \geq \frac{\bar{\mu}}{2} \|y^t - y^{t-1}\|^2$, from (B.3) we have

$$\begin{aligned} \sum_{t=1}^s \gamma_t \delta_t &\leq \sum_{t=1}^s \left[-\frac{(m-1)\gamma_t \eta_t}{2} \|x_{i_t}^t - x_{i_t}^{t-1}\|^2 - \frac{g_t \tau_t \bar{\mu}}{2} \|y^t, y^{t-1}\|^2 \right] \\ &\quad + \gamma_s(m-1) \langle \mathbf{A}(\mathbf{x}^s - \mathbf{x}^{s-1}), y^* - y^s \rangle \\ &\quad - \sum_{t=1}^s \gamma_t(m-1) \tilde{\alpha}_t \|\mathbf{A}_{i_{t-1}}\| \|x_{i_{t-1}}^{t-1} - x_{i_{t-1}}^{t-2}\| \|y^t - y^{t-1}\| \\ &\stackrel{(a)}{=} \gamma_s(m-1) \langle \mathbf{A}(\mathbf{x}^s - \mathbf{x}^{s-1}), y^* - y^s \rangle - \frac{(m-1)\gamma_s \eta_s}{2} \|x_{i_s}^s - x_{i_s}^{s-1}\|^2 \\ &\quad + \sum_{t=2}^s \left[\gamma_t(m-1) \tilde{\alpha}_t \|\mathbf{A}_{i_{t-1}}\| \|x_{i_{t-1}}^{t-1} - x_{i_{t-1}}^{t-2}\| \|y^t - y^{t-1}\| \right. \\ &\quad \left. - \frac{(m-1)\gamma_{t-1} \eta_{t-1}}{2} \|x_{i_{t-1}}^{t-1} - x_{i_{t-1}}^{t-2}\|^2 - \frac{\bar{\mu} \gamma_t \tau_t}{2} \|y^t - y^{t-1}\|^2 \right] \\ &\stackrel{(b)}{\leq} \gamma_s(m-1) \langle \mathbf{A}(\mathbf{x}^s - \mathbf{x}^{s-1}), y^* - y^s \rangle - \frac{(m-1)\gamma_s \eta_s}{2} \|x_{i_s}^s - x_{i_s}^{s-1}\|^2 \\ &\quad + \sum_{t=2}^s \left(\frac{\gamma_t^2 (m-1)^2 \tilde{\alpha}_t^2 \bar{A}^2}{2(m-1)\gamma_{t-1} \eta_{t-1}} - \frac{\bar{\mu} \gamma_t \tau_t}{2} \right) \|y^t - y^{t-1}\|^2 \\ &\stackrel{(c)}{=} \gamma_s(m-1) \langle \mathbf{A}(\mathbf{x}^s - \mathbf{x}^{s-1}), y^* - y^s \rangle - \frac{(m-1)\gamma_s \eta_s}{2} \|x_{i_s}^s - x_{i_s}^{s-1}\|^2, \end{aligned}$$

where (a) follows from regrouping the terms; (b) follows from the definition $\bar{A} = \max_{i \in [m-1]} \|\mathbf{A}_i\|$ and the simple relation that $b\langle u, v \rangle - a\|v\|^2/2 \leq b^2\|u\|^2/(2a)$ for all $a > 0$; and (c) follows from (3.18) and (3.21).

By combining the relation above with (B.2), we obtain

$$\begin{aligned} 0 &\leq \mathbb{E}_s \left[\frac{\gamma_1((m-1)\eta_1 + (m-2)\mu)}{2} \|\mathbf{x}^* - \mathbf{x}^0\|^2 - \frac{(m-1)\gamma_s(\eta_s + \mu)}{2} \|\mathbf{x}^* - \mathbf{x}^s\|^2 \right] \\ &\quad + \mathbb{E}_s [\gamma_1 \tau_1 V_h(y^*, y^0) - \gamma_s(\tau_s + 1) V_h(y^*, y^s)] \\ (B.4) \quad &\quad + \mathbb{E}_s \left[\gamma_s(m-1) \langle \mathbf{A}(\mathbf{x}^s - \mathbf{x}^{s-1}), y^* - y^s \rangle - \frac{(m-1)\gamma_s \eta_s}{2} \|x_{i_s}^s - x_{i_s}^{s-1}\|^2 \right]. \end{aligned}$$

Notice the fact that

$$\begin{aligned} &\mathbb{E}_s \left[\frac{\gamma_s(\tau_s + 1)}{2} V_h(y^*, y^s) - \gamma_s(m-1) \langle \mathbf{A}(\mathbf{x}^s - \mathbf{x}^{s-1}), y^* - y^s \rangle \right. \\ &\quad \left. + \frac{(m-1)\gamma_s \eta_s}{2} \|x_{i_s}^s - x_{i_s}^{s-1}\|^2 \right] \\ &= \mathbb{E}_s \left[\frac{\gamma_s(\tau_s + 1)}{2} V_h(y^*, y^s) - \gamma_s(m-1) \langle \mathbf{A}(x_{i_s}^s - x_{i_s}^{s-1}), y^* - y^s \rangle \right. \\ &\quad \left. + \frac{(m-1)\gamma_s \eta_s}{2} \|x_{i_s}^s - x_{i_s}^{s-1}\|^2 \right] \\ &\geq \gamma_s \mathbb{E}_s \left[\frac{(\tau_s + 1)\bar{\mu}}{4} \|y^* - y^s\|^2 - (m-1)\bar{A} \|x_{i_s}^s - x_{i_s}^{s-1}\| \|y^* - y^s\| \right. \\ &\quad \left. + \frac{(m-1)\eta_s}{2} \|x_{i_s}^s - x_{i_s}^{s-1}\|^2 \right] \\ (B.5) \quad &\geq \gamma_s \mathbb{E}_s \left(\sqrt{\frac{(m-1)(\tau_s + 1)\bar{\mu}\eta_s}{2}} - (m-1)\bar{A} \right) \|x_{i_s}^s - x_{i_s}^{s-1}\| \|y^* - y^s\| \geq 0. \end{aligned}$$

In view of (B.4) and (B.5), we complete the proof. \square

REFERENCES

- [1] Z. ALLEN-ZHU, *Natasha: Faster Non-Convex Stochastic Optimization Via Strongly Non-Convex Parameter*, <https://arxiv.org/abs/1702.00763>, 2017.
- [2] Z. ALLEN-ZHU AND E. HAZAN, *Variance reduction for faster non-convex optimization*, in Proceedings of the International Conference on Machine Learning, 2016, pp. 699–707.
- [3] Y. CARMON, J. C. DUCHI, O. HINDER, AND A. SIDFORD, *Accelerated Methods for Non-Convex Optimization*, <https://arxiv.org/abs/1611.00756>, 2016.
- [4] C. CHEN, B. HE, Y. YE, AND X. YUAN, *The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent*, Math. Program., 155 (2016), pp. 51–79.
- [5] C. D. DANG AND G. LAN, *Stochastic block mirror descent methods for nonsmooth and stochastic optimization*, SIAM J. Optim., 25 (2015), pp. 856–881, <https://doi.org/10.1137/130936361>.
- [6] D. DAVIS AND B. GRIMMER, *Proximally Guided Stochastic Subgradient Method for Nonsmooth, Nonconvex Problems*, <https://arxiv.org/abs/1707.03505v4>, 2017.
- [7] J. FAN AND R. LI, *Variable selection via nonconcave penalized likelihood and its oracle properties*, J. Amer. Statist. Assoc., 96 (2001), pp. 1348–1360.
- [8] S. GHADIMI AND G. LAN, *Stochastic first-and zeroth-order methods for nonconvex stochastic programming*, SIAM J. Optim., 23 (2013), pp. 2341–2368.
- [9] S. GHADIMI AND G. LAN, *Accelerated gradient methods for nonconvex nonlinear and stochastic programming*, Math. Program., 156 (2016), pp. 59–99.
- [10] S. GHADIMI, G. LAN, AND H. ZHANG, *Generalized Uniformly Optimal Methods for Nonlinear Programming*, <https://arxiv.org/abs/1508.07384>, 2015.
- [11] S. GHADIMI, G. LAN, AND H. ZHANG, *Mini-batch stochastic approximation methods for non-convex stochastic composite optimization*, Math. Program., 155 (2016), pp. 267–305, <https://doi.org/10.1007/s10107-014-0846-1>.
- [12] I. GOODFELLOW, Y. BENGIO, AND A. COURVILLE, *Deep Learning*, MIT Press, Cambridge, MA, 2016.
- [13] M. HONG, *Decomposing Linearly Constrained Nonconvex Problems by a Proximal Primal Dual Approach: Algorithms, Convergence, and Applications*, <https://arxiv.org/abs/1604.00543>, 2016.
- [14] M. HONG, Z.-Q. LUO, AND M. RAZAVIYAYN, *Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems*, SIAM J. Optim., 26 (2016), pp. 337–364.
- [15] R. JOHNSON AND T. ZHANG, *Accelerating stochastic gradient descent using predictive variance reduction*, in Proceedings of the International Conference on Neural Information Processing Systems, 2013, pp. 315–323.
- [16] W. KONG, J. G. MELO, AND R. D. MONTEIRO, *Complexity of a Quadratic Penalty Accelerated Inexact Proximal Point Method for Solving Linearly Constrained Nonconvex Composite Programs*, <https://arxiv.org/abs/1802.03504>, 2018.
- [17] G. LAN, Z. LI, AND Y. ZHOU, *A Unified Variance-Reduced Accelerated Gradient Method for Convex Optimization*, <https://arxiv.org/abs/1905.12412>, 2019.
- [18] G. LAN AND Y. ZHOU, *An optimal randomized incremental gradient method*, Math. Program., 171 (2018), pp. 167–215.
- [19] H. LIN, J. MAIRAL, AND Z. HARCHAOUI, *A universal catalyst for first-order optimization*, in Proceedings of the International Conference on Neural Information Processing Systems, 2015, pp. 3384–3392.
- [20] J. G. MELO AND R. D. MONTEIRO, *Iteration-Complexity of a Jacobi-Type Non-Euclidean ADMM for Multi-Block Linearly Constrained Nonconvex Programs*, <https://arxiv.org/abs/1705.07229>, 2017.
- [21] J. G. MELO AND R. D. MONTEIRO, *Iteration-complexity of a linearized proximal multiblock ADMM class for linearly constrained nonconvex optimization problems*, <http://www.optimization-online.org>, 2017.
- [22] Y. NESTEROV, *A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$* , Dokl. USSR, 269 (1983), pp. 543–547.
- [23] Y. E. NESTEROV, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, Norwell, MA, 2004.
- [24] C. PAQUETTE, H. LIN, D. DRUSVYATSKIY, J. MAIRAL, AND Z. HARCHAOUI, *Catalyst Acceleration for Gradient-Based Non-Convex Optimization*, <https://arxiv.org/abs/1703.10993>, 2017.
- [25] S. J. REDDI, A. HEFNY, S. SRA, B. POCZOS, AND A. SMOLA, *Stochastic variance reduction for nonconvex optimization*, in Proceedings of the IEEE International Conference on Machine Learning, 2016, pp. 314–323.

- [26] S. J. REDDI, S. SRA, B. PÓCZOS, AND A. SMOLA, *Fast incremental method for smooth nonconvex optimization*, in Proceedings of the 55th Conference on Decision and Control (CDC), IEEE, 2016, pp. 1971–1977.
- [27] R. SUN, Z. LUO, AND Y. YE, *On the expected convergence of randomly permuted ADMM*.
- [28] R. TIBSHIRANI, G. JAMES, D. WITTEN, AND T. HASTIE, *An Introduction to Statistical Learning: With Applications in r*, Springer, New York, 2013.
- [29] Y. WANG, W. YIN, AND J. ZENG, *Global Convergence of ADMM in Nonconvex Nonsmooth Optimization*, <https://arxiv.org/abs/1511.06324>, 2015.
- [30] Y. YE, *Randomly sampled cyclic alternating direction method of multipliers*, private communication, 2018.
- [31] C.-H. ZHANG, *Nearly unbiased variable selection under minimax concave penalty*, Ann. Statist., 38 (2010), pp. 894–942.
- [32] C.-H. ZHANG AND T. ZHANG, *A general theory of concave regularization for high-dimensional sparse estimation problems*, Statist. Sci., (2012), pp. 576–593.