1. Introduction. Nonconvex optimization plays a fundamental role in modern statistics and machine learning, e.g., for empirical risk minimization with either nonconvex loss \( [24] \) or regularization \( [6, 27, 28] \), as well as the training of deep neural networks \( [11] \). 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:

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

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

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

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

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

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 more structural information for the design of solution methods of problem \( (1.1) \). In particular, we intend to develop more efficient algorithms to solve ill-conditioned problems where the condition number \( L/\mu \) associated with problem \( (1.1) \) is large. As an example, consider the nonconvex composite problem arising from variable selection in statistics \( [6, 8] \):

\[
f(x) = \frac{1}{m} \sum_{i=1}^{m} h_i(x) + pp(x),
\]

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 \( [6] \)). 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 multi-block optimization problems with linearly coupled constraints, i.e.,

\[
\min_{x_i \in X_i} \sum_{i=1}^{m} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{m} A_i x_i = b.
\]
Here $X_i \subseteq \mathbb{R}^d$ are closed convex sets, $A_i \subseteq \mathbb{R}^{n \times d}$, $b \subseteq \mathbb{R}^n$, $f_i : X_i \rightarrow \mathbb{R}$ satisfy, for some $\mu \geq 0$,

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

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

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

Moreover, we assume that $X_m = \mathbb{R}^n$ and $A_m$ is invertible. Problem of this type arises naturally in compressed sensing and distributed optimization. For instance, consider the compressed sensing problem via nonconvex shrinkage penalties: $\min_{x \in X} \{p(x) : Ax = b\}$, where $A \in \mathbb{R}^{n \times d}$ is a big sensing matrix with $d >> n$, 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 $A$, w.l.o.g., 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 multi-block 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., [20]) that the simple gradient descent (GD) method applied to problem (1.1) requires $O(L/\epsilon)$ iterations to find an $\epsilon$-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 $O(mL/\epsilon)$ gradient computations for all the component functions $f_i$’s. Ghadimi and Lan [7] (see also [10]) 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 total $O(Lsigma^2/\epsilon^2)$ gradient computations to find a stochastic $\epsilon$-stationary solution of (1.1), i.e., a point $\bar{x}$ s.t. $E[\|\nabla f(\bar{x})\|^2] \leq \epsilon$. Here the expectation is taken w.r.t. some random variables used in the algorithm and $sigma^2$ denotes their variance. Although this complexity bound does not depend on $m$, it has a much worse dependence on $\epsilon$ than the GD method for solving problem (1.1). Inspired by the variance reduction techniques originated in convex optimization [14], Reddi et al. [22][23], and Allen-Zhu and Hazan [2] recently show that one only needs $O(m^{2/3}L/\epsilon)$ gradient evaluations to find an $\epsilon$-stationary point of (1.1), which significantly improves the bound in [7] in terms of the dependence on $\epsilon$ 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 $L/\mu$ is large.

A different line of research aims to incorporate Nesterov’s acceleration (momentum) [19] into nonconvex optimization. Ghadimi and Lan [5] first established the convergence of the accelerated gradient method for nonconvex optimization and show that it can improve the complexity of GD if the problem is ill-conditioned (i.e., $L/\mu$ is large). Their results were further improved in [9][6][21][15]. Currently the best complexity result, in terms of total gradient computations, for these methods is given by $O\left(m\sqrt{L\mu}/\epsilon\right)$ for unconstrained problems [15]. 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 $O(1/\epsilon)$ complexity bound on gradient computations. Note that some nonconvex stochastic accelerated gradient methods have been discussed in [5] but they all exhibit a worse $O(Lsigma^2/\epsilon^2)$ complexity bounds.

While stochastic and randomized methods are being intensively explored for solving problem (1.1), most existing studies for the nonconvex multi-block problem in (1.4) have been mainly focused on deterministic 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 [12], Hong et al. established the complexity for a variant of ADMM for nonconvex multi-block problems, see also [13][25] for some previous work on the asymptotic analysis of ADMM for nonconvex optimization. In [15], Melo and Monteiro presented a linearized proximal multiblock ADMM with complexity $O(1/\epsilon)$ to attain a nearly feasible $\epsilon$-stationary solution, but all the blocks have to be updated in each iteration. Later, they proposed a Jacobi-type ADMM in [17] with 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 [13][26]). To the best of our knowledge, there does not exist any complexity analysis regarding randomized methods for solving the nonconvex multi-block...
problem in [14] in the literature and as a consequence, it remains unclear whether stochastic or randomized methods are more advantageous over deterministic ones or not.

Our contribution in this paper mainly exists in the following several aspects. Firstly, 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 ill-conditioned problems. More specifically, we show that RapGrad requires totally $O(\mu(m + \sqrt{mL/\mu})/\epsilon)$ gradient computations in order to find a stochastic $\epsilon$-stationary point. For ill-conditioned problems with $L/\mu \geq m$, this bound reduces to $O(\sqrt{mL}/\mu)$, which dominates the best-known deterministic accelerated gradient methods by a factor of $\sqrt{m}$ [15], and outperforms those variance-reduced stochastic algorithms [22, 23] by a factor of $m^{3/4}$ (at least $m^{1/2}$). In fact, our complexity bound will be better than the latter algorithms as long as $L/\mu \log(L/\mu) > m^{2/3}$. Therefore, we provide some affirmative answers regarding whether the complexity bounds of variance reduced algorithms and accelerated gradient 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., $L/\mu$ 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 [16]) 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. 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.

Secondly, inspired by RapGrad, we develop a new randomized proximal-point type method, namely the randomized accelerated proximal dual (RapDual) method, for solving the nonconvex multi-block problem in [14]. 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. For simplicity, let us consider for now the case when $X_i \equiv \mathbb{R}^{d_i}$. We show that RapDual can find a solution $(\bar{x}_1, \ldots, \bar{x}_m)$ s.t. $\exists \lambda \in \mathbb{R}^n$, $\mathbb{E}[\sum_{i=1}^m \|\nabla f(\bar{x}_i) + A_i \lambda\|^2] \leq \epsilon$ and $\mathbb{E}[\|\sum_{i=1}^m A_i \bar{x}_i - b\|^2] \leq \sigma$ in at most

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

primal block updates, where $\bar{A} = \max_{i \in [m]} \|A_i\|$, $\|A\|^2 = \sum_{i=1}^{m-1} \|A_i\|^2$, and $D^0 := \sum_{i=1}^m [f_i(x_0^0) - f_i(x_i^*)]$. Moreover, we demonstrate that the total number primal block updates that RapGrad requires can be much smaller, up to a factor of $O(\sqrt{m})$, than its deterministic counterpart. To the best of our knowledge, this is the first time that the complexity of randomized methods for nonconvex multi-block optimization has been established and their advantages over deterministic methods are quantified in the literature.

Thirdly, we perform some numerical experiments on both RapGrad and RapDual for solving nonconvex finite-sum and multi-block problems in [14] and [13] 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 [14]. RapDual for nonconvex finite-sum optimization with linear constraints [14] 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, \ldots, n\} is denoted by \([n]\). We use \(E_s[X]\) to denote the expectation of a random variable \(X\) on \(i_1, \ldots, i_s\). For a given strongly convex function \(\omega\), we define the prox-function associated with \(\omega\) 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 \(\omega\) 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 randomized accelerated proximal gradient (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 are ill-conditioned. 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 [16], to approximately solve

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

to compute a new search point \(\bar{x}_\ell\).

The algorithmic schemes for RapGrad and RaGrad are described in Algorithm 1 and Algorithm 2 respectively. While it seems that we can directly apply the randomized primal-dual gradient method in [16] (or other 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. Firstly, 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 \(\bar{x}_\ell\), as well as the convergence of the sequences \(\bar{y}_\ell\) where the gradients \(\bar{y}_\ell\) 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.

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.
Algorithm 1 RapGrad for nonconvex finite-sum optimization

Let \( \hat{x}^0 \in X \), and set \( \hat{x}_i^0 = \hat{x}^0 \), \( \gamma_i^0 = \nabla f_i(\hat{x}_i^0) \), \( i = 1, \ldots, m \).

for \( \ell = 1, \ldots, k \) do

Set \( x^{-1} = x^0 = \hat{x}^{\ell-1} \), \( \gamma_i^0 = \gamma_i^{\ell-1} \), and \( y_i^0 = \gamma_i^{\ell-1} \), \( i = 1, \ldots, m \).

Run RaGrad (c.f., Algorithm 2) with input \( x^{-1}, x^0, \gamma_i^0 \), \( i = 1, \ldots, m \), and \( s \) to solve the following subproblem

\[
\min_{x \in X} \frac{1}{m} \sum_{i=1}^{m} \psi_i(x) + \varphi(x)
\]

(2.2)

to obtain output \( x^s, \hat{x}_i^s, y_i^s \), \( i = 1, \ldots, m \), where \( \psi_i(x) = \psi_i^s(x) := f_i(x) + \mu \|x - x^{\ell-1}\|^2 \), \( i = 1, \ldots, m \), and \( \varphi(x) \equiv \varphi^s(x) := \frac{\mu}{2} \|x - x^{\ell-1}\|^2 \).

Set \( \bar{x}^\ell = x^s \), \( \bar{x}_i^\ell = x_i^s \) and \( \bar{y}_i^\ell = y_i^s + 2\mu(x^{\ell-1} - x^\ell) \), \( i = 1, \ldots, m \) (note \( \bar{y}_i^\ell = \nabla \psi_i^{\ell+1}(\bar{x}_i^\ell) \) always holds).

end for

return \( \bar{x}^\ell \) for some random \( \ell \in [k] \).

Algorithm 2 RaGrad for iteratively solving subproblem (2.2)

Input \( x^{-1} = x^0 \in X, x_i^0 \in X, y_i^0 \), \( i = 1, \ldots, m \), number of iterations \( s \). Assume nonnegative parameters \( \{\alpha_i\}, \{\tau_i\}, \{\eta_i\} \) are given.

for \( t = 1, \ldots, s \) do

1. Generate a random variable \( i_t \) uniformly distributed over \( [m] \).
2. Update \( x^t \) and \( y^t \) according to

\[ \bar{x}^t = \alpha_t(x^{t-1} - x^{t-2}) + x^{t-1} \]

(2.3)

\[ \bar{x}^t = \begin{cases} (1 + \tau_i)^{-1}(\bar{x}^t + \tau_i \tilde{x}^{t-1}), & i = i_t, \\ \tilde{x}^{t-1}, & i \neq i_t \end{cases} \]

(2.4)

\[ y_i^t = \begin{cases} \nabla \psi_i(\bar{x}^t), & i = i_t, \\ y_i^{t-1}, & i \neq i_t \end{cases} \]

(2.5)

\[ \bar{y}_i^t = m(y^t - y^{t-1}) + y^{t-1} \]

(2.6)

\[ x^t = \arg \min_{x \in X} \varphi(x) + \langle \frac{1}{m} \sum_{i=1}^{m} \tilde{y}_i^t, x \rangle + \eta \nabla \varphi(x, x^{t-1}) \]

(2.7)

end for

return \( x^t, \bar{x}_i^t \), and \( y_i^t \), \( i = 1, \ldots, m \).

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 \((\epsilon, \delta)\)-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 \((\epsilon, \delta)\)-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 \((\epsilon, \delta)\)-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

\[ \|\nabla f(x)\|^2 = \|\nabla f(x) - \nabla f(\hat{x}) + \nabla f(\hat{x})\|^2 \leq 2\|\nabla f(x) - \nabla f(\hat{x})\|^2 + 2\|\nabla f(\hat{x})\|^2 \]
\[ \leq 2L^2\|x - \hat{x}\|^2 + 2\|\nabla f(\hat{x})\|^2 \leq 4\epsilon. \] (2.8)

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

\[ \text{gap}_X(x) := \max_{z \in X} (\nabla f(x), x - z) \leq (L\sqrt{\delta} + \sqrt{\epsilon})D_X, \] (2.9)

where \( D_X := \max_{x_1, x_2 \in X} \|x_1 - x_2\| \). In comparison with the two well-known criterions 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 [5] for a related notion).

Theorem 2.2 guarantees, in expectation, the existence of an approximate stationary point \( \hat{x}^* \) and \( \bar{x}^* \) by (2.11), respectively. 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

\[ s = \left\lceil -\frac{\log \hat{M}}{\log \alpha} \right\rceil \sim \mathcal{O} \left( \left( m + \sqrt{mL} \right) \log \left( \frac{1}{\alpha} \right) \right). \]

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

\[ N(\epsilon, \delta) := \mathcal{O} \left( m^\frac{1}{2} \log \left( \frac{\epsilon}{\mu} \right) \right) \] (2.11)

where \( D_0 := f(\tilde{x}^0) - f(x^*) \). As a comparison, the deterministic 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 \( x_i^0 \) and \( y_i^0 \) for \( i = 1, \ldots, m \), in (2.4) and (2.5) at each iteration, and hence would require

\[ \tilde{N}(\epsilon, \delta) := \mathcal{O} \left( m^\frac{1}{2} \log \left( \frac{\epsilon}{\mu} \right) \right) \] (2.12)

gradient evaluations to compute an \((\epsilon, \delta)\)-solution of (1.1). For ill-conditioned 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 than its deterministic counterpart as well as other deterministic methods reported in [21] [15]. It is also interesting to compare RapGrad with those variance-reduced stochastic algorithms [22, 23, 2].
by a factor of $\mathcal{O}(m^{1/3}L^{1/2}/\mu^{1/4})$, which must be greater than $\mathcal{O}(m^{1/3}L^{1/2}/\mu^{1/4})$ due to $L/\mu \geq m$. In fact, our complexity bound minorizes those for variance-reduced stochastic algorithms as long as $L/\mu \log(L/\mu) > m^{1/2}$.

Theorem 2.2 only shows the convergence of RapGrad in expectation. Similarly to the nonconvex SGD methods in [71][10], 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 post-optimization 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 using them to establish the convergence of RapGrad. Observe that the component functions $\phi_i$ and $\varphi$ in (2.10) satisfy:

- (A) $\frac{\mu}{2}||x - y||^2 \leq \psi_i(x) - \psi_i(y) - \langle \nabla \psi_i(y), x - y \rangle \leq \frac{\mu}{2}||x - y||^2$, $\forall x, y \in X$, $i = 1, \ldots, m$,
- (B) $\varphi(x) - \varphi(y) - \langle \nabla \varphi(y), x - y \rangle \geq \frac{\mu}{2}||x - y||^2$, $\forall x, y \in X$,

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

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

**Lemma 2.3.** Let $\hat{x}_i^t = (1 + \tau_t)^{-1}(\hat{x}_t + \tau_t x_i^{t-1})$, for $i = 1, \ldots, m$, $t = 1, \ldots, s$.

$$E_i[\psi_i(\hat{x}_i^t)] = m\psi_i(\hat{x}_i) - (m - 1)\psi_i(x_i^{t-1}),$$  
$$E_i[\nabla \psi_i(\hat{x}_i^t)] = m\nabla \psi_i(\hat{x}_i) - (m - 1)\nabla \psi_i(x_i^{t-1}) = E_i[y_i^t].$$  

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

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

$$\alpha_{t+1}\gamma_{t+1} = \gamma_t,$$  
$$\gamma_{t+1} = m(1 + \tau_{t+1}) - 1 \leq m\gamma_t(1 + \tau_t),$$  
$$\gamma_{t+1}\eta_{t+1} \leq \gamma_t(1 + \eta_t),$$  
$$\frac{\eta_t \mu}{4} \geq \frac{(m - 1)^2 \hat{L}}{m^2 \tau_t},$$  
$$\frac{\eta_t \mu}{4} \geq \frac{\alpha_{t+1}\hat{L}}{m(1 + \tau_t)},$$

then we have

$$\mathbb{E}_x \left[ \gamma_t(1 + \eta_t) V_{\phi}(x^*, x^*) + \sum_{i=1}^m \frac{\gamma_t(1 + \tau_t)}{4} \| z_i^t - x^* \|^2 \right] \leq \gamma_t \eta_t \mathbb{E}_x V_{\phi}(x^*, x^0) + \sum_{i=1}^m \frac{\eta_t(1 + \tau_t)}{2} \mathbb{E}_x \| z_i^0 - x^0 \|^2.$$

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

$$\mathbb{E}_x \left[ ||x^* - x^s||^2 \right] \leq \alpha^s(1 + 2\frac{\hat{L}}{\mu}) \mathbb{E}_x \left[ ||x^* - x^0||^2 + \frac{1}{m} \sum_{i=1}^m \| z_i^0 - x^0 \|^2 \right],$$  
$$\mathbb{E}_x \left[ \frac{1}{m} \sum_{i=1}^m \| z_i^s - x^s \|^2 \right] \leq 6\alpha^s(1 + 2\frac{\hat{L}}{\mu}) \mathbb{E}_x \left[ ||x^* - x^0||^2 + \frac{1}{m} \sum_{i=1}^m \| z_i^0 - x^0 \|^2 \right].$$
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
\[ E_s [ \varphi(x^*, x^*) + \sum_{i=1}^{m} \frac{L}{\alpha m} \| z^i - x^* \|^2 ] \leq \alpha \| x^* - x^0 \|^2 + \sum_{i=1}^{m} \frac{L}{\alpha m} \| z^0 - x^* \|^2, \]  
(2.20)\]
Since \( \varphi(x) = \frac{\lambda}{2} \| x - z^0 \|^2 \), we have \( \varphi(x^*, x^*) = \frac{\lambda}{2} \| x^* - x^0 \|^2 \), and \( \varphi(x^0, x^*) = \frac{\lambda}{2} \| x^* - x^0 \|^2 \). Plugging into (2.20), we obtain the following two relations:
\[ E_s [ \| x^* - x^0 \|^2 ] \leq \alpha \| x^* - x^0 \|^2 + \sum_{i=1}^{m} \frac{L}{\alpha m} \| z^0 - x^* \|^2 \]
\[ = \alpha \| x^* - x^0 \|^2 + \sum_{i=1}^{m} \frac{L}{\alpha m} (2 \| x^0 - x^0 \|^2 + 2 \| x^0 - x^0 \|^2] \]
\[ \leq \alpha \| x^* - x^0 \|^2 + \sum_{i=1}^{m} \frac{L}{\alpha m} \| z^0 - x^* \|^2 \],
\[ E_s [ \frac{1}{m} \sum_{i=1}^{m} \| z^i - x^* \|^2 ] \leq 2 \alpha \| x^* - x^0 \|^2 + \frac{1}{m} \sum_{i=1}^{m} \| z^0 - x^* \|^2 \].

In view of the above two relations, we have
\[ E_s [ \frac{1}{m} \sum_{i=1}^{m} \| z^i - x^* \|^2 ] \leq E_s [ \frac{1}{m} \sum_{i=1}^{m} (\| z^i - x^* \|^2 + \| x^* - x^0 \|^2 ] \]
\[ = 2E_s [ \frac{1}{m} \sum_{i=1}^{m} \| z^i - x^* \|^2 ] + 2E_s \| x^* - x^0 \|^2 \]
\[ \leq 6 \alpha \| x^* - x^0 \|^2 + \frac{1}{m} \sum_{i=1}^{m} \| z^0 - x^* \|^2 \].

\[ \square \]

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 \( s \geq \lceil - \log(M \alpha) / \log(\alpha) \rceil \) with \( M = 6(5 + 2L/\mu) \) be given. Also let the iterates \( \tilde{x}^\ell, \ell = 1, \ldots, k \), be generated by Algorithm 4 and \( \ell \) be randomly selected from \( [k] \). Then
\[ E[\| x^\ell - \tilde{x}^\ell \|^2 ] \leq \frac{4(1 - M\alpha^s)}{k \mu(5 - 2M\alpha)} \| f(\tilde{x}^0) - f(x^*) \|, \]
\[ E[\| x^\ell - \tilde{x}^\ell \|^2 ] \leq \frac{2M \alpha^s}{3k \mu(5 - 2M\alpha)} \| f(\tilde{x}^0) - f(x^*) \|, \]
where \( x^* \) and \( x^\ell \) are the optimal solutions to problem (1.1) and the \( \ell \)-th subproblem (2.1), respectively.

Proof. According to Theorem 2.5 (with \( L = 2 \mu + L \)), we have, for \( \ell \geq 1 \),
\[ E[\| x^\ell - \tilde{x}^\ell \|^2 ] \leq \alpha \| x^\ell - \tilde{x}^{\ell-1} \|^2 + \sum_{i=1}^{m} \frac{1}{\alpha m} \| x^{\ell-1} - \tilde{x}^{\ell-1} \|^2 \]
\[ \leq \frac{M \alpha^s}{6} E[\| x^\ell - \tilde{x}^{\ell-1} \|^2 + \sum_{i=1}^{m} \frac{1}{\alpha m} \| x^{\ell-1} - \tilde{x}^{\ell-1} \|^2 ] \]
\[ \leq 4 \alpha \| x^\ell - \tilde{x}^{\ell-1} \|^2 + \sum_{i=1}^{m} \frac{1}{\alpha m} \| x^{\ell-1} - \tilde{x}^{\ell-1} \|^2 \]
\[ \leq 4 \alpha \| x^\ell - \tilde{x}^{\ell-1} \|^2 + \sum_{i=1}^{m} \frac{1}{\alpha m} \| x^{\ell-1} - \tilde{x}^{\ell-1} \|^2 \]
\[ \leq 4 \alpha \| x^\ell - \tilde{x}^{\ell-1} \|^2 + \sum_{i=1}^{m} \frac{1}{\alpha m} \| x^{\ell-1} - \tilde{x}^{\ell-1} \|^2 \].

By induction on (2.21) and noting \( \tilde{x}^0 = \tilde{x}^0 \), we have
\[ E[\| x^\ell - \tilde{x}^\ell \|^2 ] \leq \sum_{j=1}^{\ell} (M \alpha^s) \| x^\ell - \tilde{x}^{\ell-1} \|^2 \]
In view of the above relation and (2.21), for \( \ell \geq 2 \), we have
\[ E[\| x^\ell - \tilde{x}^\ell \|^2 ] \leq \frac{M \alpha^s}{6} E[\| x^\ell - \tilde{x}^{\ell-1} \|^2 + \sum_{j=1}^{\ell-1} (M \alpha^s) \| x^\ell - \tilde{x}^{\ell-1} \|^2 ] \].
Summing up both sides of the above inequality from \( \ell = 1 \) to \( k \), we then obtain
\[
\sum_{\ell=1}^{k} \mathbb{E}[\|x_\ell^* - \bar{x}\|^2] \leq \frac{M\alpha^*}{6} \mathbb{E} \left[ \|x_{1}^1 - \bar{x}\|^2 + \sum_{\ell=2}^{k} \left( \|x_{\ell}^* - \bar{x}\|^2 + \sum_{j=1}^{\ell-1} (M\alpha^*)^{\ell-j} \|x_{j}^j - \bar{x}\|^2 \right) \right].
\]
Using the above relation and Lemma 2.6, we have
\[
\sum_{\ell=1}^{k} \mathbb{E}[\|x_\ell^* - \bar{x}\|^2] \leq \frac{M\alpha^*}{6} \sum_{\ell=1}^{k} \mathbb{E}[\|x_\ell^* - \bar{x}\|^2].
\]
Using the fact that \( x_\ell^* \) is optimal to the \( \ell \)-th subproblem, and letting \( x_0^* = \bar{x}^0 \) (\( x_0^* \) is a free variable), we have
\[
\sum_{\ell=1}^{k} [\psi^\ell(x_\ell^*) + \varphi^\ell(x_\ell^*)] \leq \sum_{\ell=1}^{k} [\psi^\ell(x_\ell^*) + \varphi^\ell(x_\ell^*)],
\]
which, in view of the definition of \( \psi^\ell \) and \( \varphi^\ell \), then implies that
\[
\sum_{\ell=1}^{k} \mathbb{E}[f(x_\ell^*)] + \frac{3\mu}{2} \sum_{\ell=1}^{k} \mathbb{E}[\|x_\ell^* - \bar{x}\|^2] \leq \frac{1}{6} \sum_{\ell=1}^{k} \mathbb{E}[\|x_\ell^* - \bar{x}\|^2].
\]
Combining (2.23) and (2.24), we obtain
\[
\sum_{\ell=1}^{k} \mathbb{E}[f(x_\ell^*)] + \frac{3\mu}{2} \sum_{\ell=1}^{k} \mathbb{E}[\|x_\ell^* - \bar{x}\|^2] \leq \frac{1}{6} \sum_{\ell=1}^{k} \mathbb{E}[\|x_\ell^* - \bar{x}\|^2].
\]
Using (2.24), (2.23) and the condition on \( s \), we have
\[
\sum_{\ell=1}^{k} \mathbb{E}[\|x_\ell^* - \bar{x}\|^2] \leq \frac{4(1-M\alpha^*)}{\mu(6-7M\alpha^*)} [f(\bar{x}^0) - f(x^*)],
\]
and
\[
\sum_{\ell=1}^{k} \mathbb{E}[\|x_\ell^* - \bar{x}\|^2] \leq \frac{2M\alpha^*}{3\mu(6-7M\alpha^*)} [f(\bar{x}^0) - f(x^*)].
\]
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 \( \ell \)-th subproblem (2.1),
\[
\nabla \psi^\ell(x_\ell^*) + \nabla \varphi^\ell(x_\ell^*) \in -N_X(x_\ell^*).
\]
From the definition of \( \psi^\ell \) and \( \varphi^\ell \), we have
\[
\nabla f(x_\ell^*) + 3\mu(x_\ell^* - \bar{x}^\ell) \in -N_X(x_\ell^*).
\]
From the optimality condition of (2.27), we obtain
\[
\varphi(x^\ell) - \varphi(x^*) + \left( \frac{1}{m} \sum_{i=1}^{m} \ell_i, x^\ell - x^* \right) \leq \eta \nu(x^*, x^\ell) - (1 + \eta \nu)(x^*, x^\ell) - \eta \nu(x^*, x^\ell) - (1 + \eta \nu)(x^*, x^\ell).
\]
Using the above relation and Lemma 2.6, we have
\[
\mathbb{E}[\|\bar{x}^\ell - x_\ell^*\|^2] \leq \frac{4(1-M\alpha^*)}{\mu(6-7M\alpha^*)} [f(\bar{x}^0) - f(x^*)],
\]
\[
\mathbb{E} \left[ d(\nabla f(x_\ell^*), -N_X(x_\ell^*)) \right]^2 \leq \mathbb{E}[\|x_\ell^* - \bar{x}\|^2] \leq \frac{4\mu}{k} [f(\bar{x}^0) - f(x^*)],
\]
\[
\mathbb{E}[\|\bar{x}^\ell - x_\ell^*\|^2] \leq \frac{2M\alpha^*}{3\mu(6-7M\alpha^*)} [f(\bar{x}^0) - f(x^*)] \leq \frac{4\mu}{k} [f(\bar{x}^0) - f(x^*)].
\]
\( \square \)
3. Nonconvex multi-block optimization with linear constraints. In this section, we present a randomized accelerated proximal dual (RapDual) algorithm for solving the nonconvex multi-block optimization problem in (1.4) and show the potential advantages in terms of the total number of block updates.

Without loss of generality, we assume the last block of the constraint matrix is identity. Hence, problem (1.4) reduces to

$$
\min_{\mathbf{x} \in \mathcal{X}, \mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) + f_m(\mathbf{x}_m),
\text{s.t. } A\mathbf{x} + \mathbf{x}_m = b,
$$

(3.1)

where \( f(\mathbf{x}) := \sum_{i=1}^{m-1} f_i(\mathbf{x}_i), \) \( X = X_1 \times \ldots \times X_{m-1}, \) \( A = [A_1, \ldots, A_{m-1}] \), and \( \mathbf{x} = (x_1, \ldots, x_{m-1}) \). Indeed, since the last block \( A_m \) in (1.4) is invertible, we can always multiply both sides of the original linear constraint by \( A_m^{-1} \), and obtain \( A_m^{-1}A\mathbf{x} + \mathbf{x}_m = A_m^{-1}b \) with the last matrix block given by identity. We will discuss how to generalize RapDual and its convergence results for the more general \( A_m \) in the end of Subsection 3.1.

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 \( b - A\mathbf{x} \) in the objective function of (3.1), we obtain

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

(3.2)

where \( B_i = (0, \ldots, I, \ldots, 0) \) 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 [4] 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 to update only a single block at a time, applies to the case 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 \( \mathbf{x}^{t-1} \) and \( \mathbf{x}_m^{t-1} \) from the previous iteration, we define a new proximal subproblem as

$$
\min_{\mathbf{x} \in \mathcal{X}, \mathbf{x} \in \mathbb{R}^n} \psi(\mathbf{x}) + \psi_m(\mathbf{x}_m)
\text{s.t. } A\mathbf{x} + \mathbf{x}_m = b,
$$

(3.3)

where \( \psi(\mathbf{x}) := f(\mathbf{x}) + \mu\|\mathbf{x} - \mathbf{x}^{t-1}\|^2 \) and \( \psi_m(\mathbf{x}_m) := f_m(\mathbf{x}_m) + \mu\|\mathbf{x}_m - \mathbf{x}_m^{t-1}\|^2 \). 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 by the RapDual method to solve problem (3.1).

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

$$
\min_{\mathbf{x} \in \mathcal{X}, \mathbf{x} \in \mathbb{R}^n} \{\psi(\mathbf{x}) + \psi_m(\mathbf{x}_m) + \max_{y \in \mathbb{R}^n} (\sum_{i=1}^{m} \langle A_i; \mathbf{x}_i - b, y \rangle)\}
= \min_{\mathbf{x} \in \mathcal{X}} \{\psi(\mathbf{x}) + \max_{y \in \mathbb{R}^n} (\langle A\mathbf{x} - b, y \rangle + \min_{\mathbf{x} \in \mathbb{R}^n} \{\psi_m(\mathbf{x}_m) + \langle \mathbf{x}_m, y \rangle\})\}
= \min_{\mathbf{x} \in \mathcal{X}} \{\psi(\mathbf{x}) + \max_{y \in \mathbb{R}^n} (\langle A\mathbf{x} - b, y \rangle - h(y))\},
$$

(3.4)

where \( h(y) := -\min_{\mathbf{x} \in \mathbb{R}^n} \{\psi_m(\mathbf{x}_m) + \langle \mathbf{x}_m, y \rangle\} = \psi_m^{\ast}(\mathbf{x}) \). 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 \( \psi_m \) has \( L \)-Lipschitz continuous gradients, \( h(y) = \psi_m^{\ast}(\mathbf{x}) \) is \( 1/L \)-strongly convex. Using the fact that \( h \) is strongly convex, we can see that (3.3) 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,

$$
y^t = \arg \min_{y \in \mathbb{R}^n} h(y) + \langle -A\mathbf{x}^t + b, y \rangle + \tau_t V_h(y, y^{t-1}).
$$
If we set \( g^0 = \nabla h(y^0) = -x^0_m \), then it is easy to see by induction that \( g^t = \frac{(\tau_t g^{t-1} + Ax^t) - b}{1 + \tau_t} \), and \( y^t = \nabla h^*(g^t) \) for all \( t \geq 1 \). Moreover, \( h^*(g) = \max_{y \in \mathbb{R}^n} (g,y) - h(y) = \max_{y \in \mathbb{R}^n} (g,y) - \psi_m^*(-y) = \psi_m(-g) \), thus \( y^t = -\nabla \psi_m(-g^t) \) is the negative gradient of \( \psi_m \) 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 \((3,3)\), making it especially favorable when the number of blocks \( m \) is large. However, similar difficulty as mentioned in Section \(2.4\) also appears when we integrate this algorithm with proximal-point type method to yield the final RapDual method in Algorithm \([3]\). Firstly, Algorithm \([4]\) also keeps a few intertwined primal and dual sequences, thus we need to carefully decide the input and output of Algorithm \([4]\) so that information from previous iterations of RapDual is fully used. Secondly, 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 \([3]\) describes the basic scheme of RapDual. At the beginning, all the blocks are initialized using the output from solving the previous subproblem. Note that \( x^0_m \) is used to initialize \( g \), which further helps compute the dual variable \( y \) without using the conjugate function \( h \) of \( \psi_m \). We will derive the convergence result for Algorithm \([3]\) in terms of primal variables and construct relations between successive search points \((x^t, x^0_m)\), which will be used to prove the final convergence of RapDual.

**Algorithm 3** RapDual for nonconvex multi-block optimization

Let \( \bar{x}^0 \in X \), \( \bar{x}^0_m \in \mathbb{R}^n \), such that \( Ax^0 + \bar{x}^0_m = b \), and \( \bar{y}^0 = -\nabla f_m(\bar{x}^0_m) \).

for \( \ell = 1, \ldots, k \) do

- Set \( x^{-1} = x^0 = \bar{x}^{\ell-1}, x^0_m = \bar{x}^{\ell-1}_m \).
- Run Algorithm \([4]\) with input \( x^{-1}, x^0, x^0_m \) and \( s \) to solve the following subproblem

\[
\min_{x \in X, x_m \in \mathbb{R}^n} \psi(x) + \psi_m(x_m)
\quad \text{s.t.} \quad Ax + x_m = b
\]

(3.5)

to compute output \((x^\ell, x^0_m)\), where \( \psi(x) \equiv \psi^\ell(x) := f(x) + \mu||x - \bar{x}^{\ell-1}||^2 \) and \( \psi_m(x) \equiv \psi_m^\ell(x) := f_m(x) + \mu||x_m - \bar{x}^{\ell-1}_m||^2 \).

- Set \( \bar{x}^\ell = x^\ell, \bar{x}^{\ell}_m = x^0_m \).

end for

return \((\bar{x}^\ell, \bar{x}^\ell_m) \) for some random \( \ell \in [k] \).
Algorithm 4 RaDual for solving subproblem (3.3)

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

for $t = 1, \ldots, s$ do

1. Generate a random variable $i_t$ uniformly distributed over $[m-1]$.

2. Update $x^t$ and $y^t$ according to

\[
\hat{x}^t = \alpha_t (x^{t-1} - x^{t-2}) + x^{t-1},
\]

\[
g^t = (\tau_t g^{t-1} + Ax^{t} - b)/(1 + \tau_t),
\]

\[
y^t = \arg \min_{y \in \mathbb{R}^n} h(y) + \langle -Ax^{t}, y \rangle + \tau_t V_h(y, y^{t-1}) = -\nabla \psi_m(-g^t),
\]

\[
x_t = \begin{cases} \arg \min_{x_t \in X} \psi_t(x_t) + \langle A_t y^t, x_t \rangle + \frac{\delta}{2} \|x_t - x_{t-1}^\ast\|^2, & \text{if } i_t = i, \\ x_{t-1}, & \text{if } i \neq i_t. \end{cases}
\]

end for

Compute $x_s^\ast = \arg \min_{x_s \in \mathbb{R}^n} \{\psi_m(x_s) + \langle x_s, y_s^\ast \rangle\}$.

return $(x^\ast, x_s^\ast)$

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

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

\[
\left[d\left(\nabla f(\hat{x}) + A^\top \lambda - N_X(\hat{x})\right)\right]^2 \leq \epsilon \|\nabla f_m(x_m) + \lambda\|^2 \leq \epsilon,
\]

\[
\|x - \hat{x}\|^2 \leq \delta, \|Ax + x_m - b\|^2 \leq \sigma.
\]

A stochastic counterpart is one that satisfies

\[
\mathbb{E}\left[d\left(\nabla f(\hat{x}) + A^\top \lambda - N_X(\hat{x})\right)\right]^2 \leq \epsilon \mathbb{E}\|\nabla f_m(x_m) + \lambda\|^2 \leq \epsilon,
\]

\[
\mathbb{E}\|x - \hat{x}\|^2 \leq \delta, \mathbb{E}\|Ax + x_m - b\|^2 \leq \sigma.
\]

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

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

\[
\mathcal{M} = (2 + \frac{4}{p}) \cdot \max \left\{ 2, \frac{L^2}{p} \right\}, \quad \alpha = 1 - \frac{2}{(m-1)(1 + \kappa e^c)}, \quad c = \frac{\bar{\lambda}^2}{\rho \bar{M}} = \frac{(2\mu L + 2\bar{M})\bar{A}^2}{\rho}, \quad \bar{A} = \max_{i \in [m-1]} \|A_i\|,
\]

and other parameters are set to

\[
\alpha_t = (m-1)\alpha, \quad \gamma_t = \alpha^{-t}, \quad \tau_t = \frac{\alpha}{1 - \alpha}, \quad \eta_t = \frac{\alpha - m - 1}{1 - \alpha}, \quad \forall t = 1, \ldots, s.
\]

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

\[
\mathbb{E}\left[d\left(\nabla f(\hat{x}^t) + A^\top \lambda^\ast - N_X(\hat{x}^t)\right)\right]^2 \leq \frac{8m}{\bar{M}} \left\{ f(\bar{x}^0) + f_m(x_m^0) - [f(\lambda^\ast) + f_m(x_m^\ast)] \right\},
\]

\[
\mathbb{E}\|\nabla f_m(x_m^t) + \lambda^\ast\|^2 \leq \frac{34m}{k} \left\{ f(\bar{x}^0) + f_m(x_m^0) - [f(\lambda^\ast) + f_m(x_m^\ast)] \right\},
\]

\[
\mathbb{E}\|x^t - x_m^t\|^2 \leq \frac{34m}{k \bar{M}} \left\{ f(\bar{x}^0) + f_m(x_m^0) - [f(\lambda^\ast) + f_m(x_m^\ast)] \right\},
\]
\[ \mathbb{E}[\|Ax^t + x^t_m - b\|^2] \leq \frac{2(\|A\|^2 + 1)n}{L^2} \left\{ f(\bar{x}^0) + f_m(\bar{x}^0_m) - [f(\bar{x}^t) + f_m(\bar{x}^t_m)] \right\}, \]

where \((x^*, x^*_m)\) and \((\bar{x}^t, x^*_m)\) denote the optimal solutions to (1.4) and the \(t\)-th subproblem (3.3), respectively.

Theorem 3.2 ensures that our output solution \((\bar{x}^t, x^*_m)\) is close enough to an unknown approximate stationary point \((x^*, x^*_m)\). According to Theorem 3.2 we can bound the complexity of RapDual to compute the dual variable \(\hat{\mu}\). It can be easily shown that everything in Theorem 3.2 will be the same except the complexity bound implied by Corollary 4.4 of [4].

Let \(D^0 := f(\bar{x}^0) + f_m(\bar{x}^0_m) - [f(x^*) + f_m(x^*_m)]\). 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

\[ N(\epsilon, \delta, \sigma) := O\left( mA\sqrt{\mu}\log\left(\frac{\kappa}{\mu}\right) \cdot \max\left\{ 1, \frac{1}{\epsilon^2}, \frac{\|A^2\|}{\sigma^2} \right\} D^0 \right). \]

As a comparison, the deterministic version of this algorithm would update all the \(x^i_t\) for \(i = 1, \ldots, m\), in (3.9), and thus would require

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

primal block updates to obtain an \((\epsilon, \delta, \sigma)\)-solution of (1.4). Therefore, the benefit of randomization comes from the difference between \(\|A\|\) and \(\hat{A}\). Obviously we always have \(\|A\| > \hat{A}\), and the relative gap between \(\|A\|\) and \(\hat{A}\) can be large when all the matrix blocks have close norms.

In this section, we first show the convergence of Algorithm 4 for solving the convex multi-block subproblem (3.3) with

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

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

Some simple relations about the iterations generated by the Algorithm 4 are characterized in the following lemma, and the proof follows directly from the definition of \(\hat{x}\) in (3.13), thus has been omitted.

**Lemma 3.3.** Let \(\bar{x}^0 = x^0\) and \(\bar{x}^t\) for \(t = 1, \ldots, s\) be defined as follows:

\[ \bar{x}^t = \arg\min_{x \in X} \psi(x) + (A^\top y^t, x) + \frac{\mu}{2}\|x - x^{t-1}\|^2, \]

where \(x^t\) and \(y^t\) are obtained from (3.8)-(3.9), then we have

\[ \mathbb{E}_i \{ \|x - \bar{x}^t\|^2 \} = \mathbb{E}_i \{ (m - 1)\|x - x^t\|^2 - (m - 2)\|x - x^{t-1}\|^2 \}, \]

\[ \mathbb{E}_i \{ \|x^t - x_{i_t}^t\|^2 \} = \mathbb{E}_i \{ (m - 1)\|x_{i_t} - x_{i_t}^{t-1}\|^2 \}. \]
The following lemma builds some connections between the input and output of Algorithm in terms of both primal and dual variables, and the proof can be found in Appendix B.

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

$$\alpha_{t+1} = (m-1)\alpha_t + 1,$$  
$$\gamma_t = \gamma_t + \alpha_t + 1,$$  
$$\gamma_{t+1} (m-1)\eta_t + (m-2)\mu \leq (m-1)\gamma_t (\eta_t + \mu),$$  
$$\gamma_{t+1} \tau_t + 1 \leq \gamma_t (\tau_t + 1),$$  
$$2(m-1)\alpha_t + 1 A^2 \leq \bar{\mu} \eta_t \tau_t + 1,$$

where $A = \max_{i \in [n-1]} \|A_i\|$. Then we have

$$\mathbb{E}_s \left\{ \frac{\gamma_t ((m-1)\mu + (m-2)\mu)}{2} \|x^* - x^0\|^2 - \frac{\gamma_t (\eta_t + \mu)}{2} \|x^{*} - x^0\|^2 \right\}$$  
$$+ \mathbb{E}_s \left\{ \gamma_t \tau_t V_h(y^*, y^0) - \frac{\gamma_t (\tau_t + 1) \bar{\mu}}{2} V_h(y^*, y^*) \right\} \geq 0.$$  

(3.21)

Now we present the main convergence result of Algorithm 4 in Theorem 3.5 which eliminates the dependence on dual variables and relates directly the successive searching points of RapDual.

**Theorem 3.5.** 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 $\alpha_t = \alpha$. Then, for any $s \geq 1$, we have

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

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

**Proof.** It is easy to check that (3.10) and (3.11) satisfy conditions (3.16), (3.17), (3.18), (3.19), and (3.20) when $\mu, \bar{\mu} > 0$. Then we have

$$\mathbb{E}_s \left\{ \frac{(m-1)\mu + (m-2)\mu}{2} \|x^* - x^0\|^2 + \frac{\gamma_t (\tau_t + 1) \bar{\mu}}{2} V_h(y^*, y^*) \right\} \leq \frac{2(m-1)\mu + (m-2)\mu}{2} \|x^0 - x^*\|^2 + \gamma_t \tau_t V_h(y^0, y^0).$$

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

$$\mathbb{E}_s \left\{ \mu \|x^* - x^{*}\|^2 + V_h(y^0, y^*) \right\} \leq \alpha^s \left\{ \mu \|x^0 - x^*\|^2 + 2V_h(y^0, y^*) \right\},$$

(3.22)

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

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

(3.23)

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

(3.24)

Combining (3.22), (3.23) and (3.24), we have

$$\mathbb{E}_s \left\{ \|x^* - x^0\|^2 + \|x^* - x^0\|^2 \right\} \leq \alpha^s \mathcal{M}(\|x^0 - x^*\|^2 + \|x^0 - x^*\|^2).$$

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

$$\mathbb{E} \left\{ \|\tilde{x}^t - \tilde{x}^{t-1}\|^2 + \|x_m^* - x_m^*\|^2 \right\} \leq \frac{1}{\xi \mu (\tilde{1} - \mu \alpha)} \left\{ f(\tilde{x}^0) + f(\tilde{x}^0) - \tilde{f}(\tilde{x}^0) + f(x^0) + f(x^m) \right\},$$

where $\tilde{\mu}$ is the average of $\mu$.
Plugging in the definition of $\psi$, we have

$$\mathbb{E}\left(\|x^\ell - x^\ell_0\|^2 + \|x^\ell_m - x^\ell_m\|^2\right) \leq \frac{M_\alpha^*}{\lambda(1-M_\alpha)} \left\{ \mathcal{F}(x^0) + f_m(x^\ell_m) - [f(x^*) + f_m(x^\ell_m)] \right\},$$

where $(x^*, x^\ell_m)$ and $(x^\ell_0, x^\ell_m)$ are the optimal solutions to (3.24) and the $\ell$-th subproblem (3.3), respectively.

**Proof.** According to Theorem 3.2, we have

$$\mathbb{E}\left(\|x^\ell - x^\ell_0\|^2 + \|x^\ell_m - x^\ell_m\|^2\right) \leq \alpha^* \mathcal{M}(\|x^\ell - x^\ell_0\|^2 + \|x^\ell_m - x^\ell_m\|^2).$$

Let us denote $(x^0_0, x^\ell_m) = (x^0, x^\ell_m)$ and by selection, it is feasible to subproblem (3.3) when $\ell = 1$. Since $(x^\ell_0, x^\ell_m)$ is optimal and $(x^\ell_0 - x^\ell_m)$ is feasible to the $\ell$-th subproblem, we have

$$\psi^\ell(x^\ell_0) + \psi^\ell_m(x^\ell_m) \leq \psi^\ell(x^\ell_0) + \psi^\ell_m(x^\ell_m).$$

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

$$\sum_{\ell=1}^k \left[ f(x^\ell_0) + f_m(x^\ell_m) + \mu(\|x^\ell_0 - x^\ell_m\|^2 + \|x^\ell_m - x^\ell_m\|^2) \right] \leq \sum_{\ell=1}^k \left[ f(x^\ell_0) + f_m(x^\ell_m) + \mu(\|x^\ell_0 - x^\ell_m\|^2 + \|x^\ell_m - x^\ell_m\|^2) \right].$$

Combining (3.25) and (3.26) and noticing that $(x^0_0, x^\ell_m) = (x^0, x^\ell_m)$, we have

$$\mu \sum_{\ell=1}^k \mathbb{E}(\|x^\ell_0 - x^\ell_m\|^2 + \|x^\ell_m - x^\ell_m\|^2) \leq \sum_{\ell=1}^k \left[ f(x^\ell_0) + f_m(x^\ell_m) - [f(x^*) + f_m(x^\ell_m)] \right] + \mu \sum_{\ell=1}^k \mathbb{E}(\|x^\ell_0 - x^\ell_m\|^2 + \|x^\ell_m - x^\ell_m\|^2) \leq f(x^0) + f_m(x^\ell_0) - [f(x^*) + f_m(x^\ell_0)] + \mu \mathcal{M} \sum_{\ell=1}^k \mathbb{E}(\|x^\ell_0 - x^\ell_m\|^2 + \|x^\ell_m - x^\ell_m\|^2).$$

In view of (3.27) and (3.25), we have

$$\sum_{\ell=1}^k \mathbb{E}(\|x^\ell_0 - x^\ell_m\|^2 + \|x^\ell_m - x^\ell_m\|^2) \leq \frac{1}{M_\alpha^*} \sum_{\ell=1}^k \left\{ f(x^\ell_0) + f_m(x^\ell_m) - [f(x^*) + f_m(x^\ell_m)] \right\},$$

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 $\ell$-th subproblem (3.3), there exists some $\lambda^*$ such that

$$\nabla \psi^\ell(x^\ell_0) + A^T \lambda^* \in -N_X(x^\ell_0),$$

$$\nabla \psi^\ell_m(x^\ell_m) + \lambda^* = 0,$$

$$Ax^\ell + x^\ell_m = b.$$  

Plugging in the definition of $\psi^\ell$ and $\psi^\ell_m$, we have

$$\nabla f^\ell(x^\ell_0) + 2\mu(x^\ell_0 - x^\ell_m) + A^T \lambda^* \in -N_X(x^\ell_0),$$

$$\nabla f^\ell_m(x^\ell_m) + 2\mu(x^\ell_m - x^\ell_m) + \lambda^* = 0.$$  

Now we are ready to evaluate the quality of the solution $(x^\ell, x^\ell_m)$. In view of (3.29) and Lemma 3.6, we have

$$\mathbb{E}\left[d(\nabla f^\ell(x^\ell_0) + A^T \lambda^* - N_X(x^\ell_0))^2\right] \leq \frac{8\mu}{k} \left\{ f(x^0) + f_m(x^\ell_m) - [f(x^*) + f_m(x^\ell_m)] \right\}.$$  

15
Similarly, due to (3.30) and Lemma 3.6 we have

\[
\mathbb{E}\|\nabla f_m^\ell(x_m^\ell) + \lambda^\star\|^2 = \mathbb{E}\|\nabla f_m^\ell(x_m^\ell) - \nabla f_m^\ell(x_m^\star) - 2\mu(x_m^\star - x_m^\ell - 1)\|^2 \\
\leq 2\mathbb{E}\{\|\nabla f_m^\ell(x_m^\ell) - \nabla f_m^\ell(x_m^\star)\|^2 + 4\mu^2\|x_m^\star - x_m^\ell - 1\|^2\} \\
\leq \mathbb{E}\left\{18\mu^2\|x_m^\star - x_m^\ell\|^2 + 8\mu^2\|x_m^\star - x_m^\ell - 1\|^2\right\} \\
\leq \mu^2(8 + 18\mathcal{M}_\alpha^\star)\mathbb{E}\{\|\mathbf{x}^\ell - \mathbf{x}^\star - 1\|^2 + \|x_m^\star - x_m^\ell - 1\|^2\} \\
\leq \frac{2\mu}{\kappa(1 - \mathcal{M}_\alpha^\star)} \mathbb{E}\left\{f(\mathbf{x}^\ell) + f_m(x_m^0) - [f(\mathbf{x}^\star) + f_m(x_m^\star)]\right\}. \\
\]

By Lemma 3.6 we have

\[
\mathbb{E}\|\mathbf{x}^\ell - x_m^\star\|^2 \leq \frac{\mathcal{M}_\alpha^\star}{\kappa \mu(1 - \mathcal{M}_\alpha^\star)} \mathbb{E}\{f(\mathbf{x}^\ell) + f_m(x_m^0) - [f(\mathbf{x}^\star) + f_m(x_m^\star)]\} \\
\leq \frac{2\mathcal{M}_\alpha^\star}{\kappa \mu^2} \mathbb{E}\{f(\mathbf{x}^\ell) + f_m(x_m^0) - [f(\mathbf{x}^\star) + f_m(x_m^\star)]\} \\
\leq \frac{2\mu}{\kappa L^2} \mathbb{E}\{f(\mathbf{x}^\ell) + f_m(x_m^0) - [f(\mathbf{x}^\star) + f_m(x_m^\star)]\}. \\
\]

Combining (4.28) and Lemma 3.6 we have

\[
\mathbb{E}\|A\mathbf{x}^\ell + x_m^\ell - b\|^2 = \mathbb{E}\{A\mathbf{x}^\ell - x_m^\star\} + \mathbb{E}\|x_m^\star - x_m^\ell\|^2 \\
\leq 2\mathbb{E}\{\|A\|^2\|\mathbf{x}^\ell - x_m^\star\|^2 + \|x_m^\star - x_m^\ell\|^2\} \\
\leq 2(\|A\|^2 + 1)\mathbb{E}\{\|\mathbf{x}^\ell - x_m^\star\|^2 + \|x_m^\star - x_m^\ell\|^2\} \\
\leq \frac{2(\|A\|^2 + 1)\mathcal{M}_\alpha^\star}{\kappa \mu^2(1 - \mathcal{M}_\alpha^\star)} \mathbb{E}\{f(\mathbf{x}^\ell) + f_m(x_m^0) - [f(\mathbf{x}^\star) + f_m(x_m^\star)]\} \\
\leq \frac{2(\|A\|^2 + 1)\mu}{\kappa L^2} \mathbb{E}\{f(\mathbf{x}^\ell) + f_m(x_m^0) - [f(\mathbf{x}^\star) + f_m(x_m^\star)]\}. \\
\]

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

4.1. Nonconvex finite-sum optimization. We consider the least square problem with the smoothly clipped absolute deviation (SCAD) penalty as a testing problem. SCAD has been proved in [6] to be efficient in variable selection. While the original SCAD \(p_{\lambda, \gamma}\) defined below does not have smooth gradient at \(x = 0\), we can bypass this potential problem by using a small positive number \(\epsilon\) to obtain a smooth approximation \(p_{\lambda, \gamma, \epsilon}\):

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

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

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

which can be viewed as a special case of problem (1.11) with \(f_i(x) = \frac{1}{2}(a_i^\top x - b_i)^2 + \frac{\epsilon}{2} \sum_{i=1}^n p_{\lambda, \gamma, \epsilon}(x_i)\). 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 \(L/\mu\) usually dominates \(m\).

We test Algorithm 1 (RapGrad), both randomized and deterministic versions, on some randomly generated data sets with dimension \(m = 1000, n = 100\). The parameters used in \(p_{\lambda, \gamma, \epsilon}\) are \(\epsilon = 10^{-3}, \lambda = 2, \gamma = 2.5\). 
\( \gamma = 4 \) and the penalty \( \rho \) is set to 0.01. Notice that the \( x \)-axis represents the number of gradient evaluations divided by \( m \), which is counted as 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, randomized version can reduce both objective value and norm of gradient faster than its deterministic counterpart in terms of number of gradient evaluations.

**Fig. 4.1.** Deterministic 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 \( \| \nabla f \|^2 \).

We also compare our RapGrad with the full SVRG in non-convex setting (Algorithm 2 in [2]) and Accelerated Gradient method (AG) in [8]. Notice that, both RapGrad and SVRG are randomized algorithms, while AG is a deterministic one. 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 as well as gradient norm faster than both SVRG and AG.

**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 \( \| \nabla f \|^2 \).

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 above mentioned convergence rate theoretically, but may improve the practical performance of RapGrad for this problems in our experiments. From Figure 4.3 and Figure 4.4 we can conclude that, by using smaller \( s \), our randomized algorithm is able to reduce \( f \) and \( \| \nabla f \|^2 \) much faster, whereas the deterministic 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 the following table, we compare the RapGrad without tuning, tuned RapGrad, SVRG as well as AG on testing problems of different sizes, with stopping criteria $\|\nabla f\|^2 < 10^{-10}$ and maximal pass $3 \times 10^4$. The table shows that this simple tuning technique is able to bring huge performance improvement. An interesting observation is that RapGrad without tuning is more likely to outperform SVRG when $n$ is large relative to $m$. 

---

**Fig. 4.3.** Comparison on objective value for deterministic and randomized versions of RapGrad, when numbers of inner iterations are $s$, $s/10$, and $s/20$, respectively. Left Figure: Deterministic version. Right Figure: Randomized version.

**Fig. 4.4.** Comparison on square of gradient norm $\|\nabla f\|^2$ for deterministic and randomized versions of RapGrad when numbers of inner iterations are $s$, $s/10$, and $s/20$, respectively. Left Figure: Deterministic version. Right Figure: Randomized version.
RapGrad tuned SVRG AG

\begin{tabular}{|c|c|c|c|}
\hline
\textbf{m = 1000, n = 100} & 2850 & 502 & 1143 \(3 \times 10^4\) \\
\textbf{m = 1000, n = 300} & 4894 & 874 & 5493 \(3 \times 10^4\) \\
\textbf{m = 1000, n = 500} & 11299 & 1165 & 19029 \(3 \times 10^4\) \\
\textbf{m = 800, n = 100} & 3113 & 559 & 1245 \(3 \times 10^4\) \\
\textbf{m = 800, n = 300} & 5467 & 970 & 7743 \(3 \times 10^4\) \\
\textbf{m = 800, n = 500} & 12673 & 1290 & 3 \(3 \times 10^4\) \\
\textbf{m = 600, n = 100} & 3113 & 559 & 1245 \(3 \times 10^4\) \\
\textbf{m = 600, n = 300} & 5467 & 970 & 7743 \(3 \times 10^4\) \\
\textbf{m = 600, n = 500} & 12673 & 1290 & 3 \(3 \times 10^4\) \\
\hline
\end{tabular}

\textbf{Table 4.1} Comparison on numbers of passes to the dataset with stopping criteria \(\|\nabla f\| < 10^{-10}\) and maximal pass \(3 \times 10^4\).

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

\[
\min_{x_i \in X_i} \sum_{i=1}^{m} P_{\lambda, \gamma, \epsilon}(x_i) \\
\text{s.t. } \sum_{i=1}^{m} A_i x_i = b
\]

where each \(x_i = (x_{i1}, \ldots, x_{id_i})\) is a vector of dimension \(d_i\), and \(P_{\lambda, \gamma, \epsilon}(x_i) = \sum_{j=1}^{d_i} p_{\lambda, \gamma, \epsilon}(x_{ij})\). Instead of using the \(\ell_1\) 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 \([A_1, A_2, \ldots, A_m]\). W.L.O.G, 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 is performed on some randomly generated data sets of size \(m = 1001, n = 100\), and 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 deterministic 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 [13], with \(\rho = L^2\), which 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 quite tiny constraint violation. Similar to RapGrad, if we reduce the number of inner iterations \(s\) per subproblem by a factor of 10 or 20, we obtain results in Figure 4.7 and Figure 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. 4.5.** Deterministic 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 \(\|Ax + x_m - b\|^2\).
5. Concluding remarks. In this paper, we propose a new randomized accelerated proximal-gradient (RapGrad) method for solving nonconvex finite-sum problems (1.1) and a new randomized primal-dual gradient (RapGrad) method for nonconvex multi-block problems (1.4), respectively. We demonstrate that for ill-conditioned problem (1.1), our RapGrad has 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 deterministic accelerated
gradient 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 $\sqrt{n}$ than the deterministic methods for solving problems [23]. The potential advantages of RapGrad and RapDual are also demonstrated through our preliminary numerical experiments.

REFERENCES

[1] Z. Allen-Zhu, Natasha: Faster Non-Convex Stochastic Optimization Via Strongly Non-Convex Parameter, ArXiv e-prints, (2017).
[2] Zeyuan Allen-Zhu and Elad Hazan, Variance reduction for faster non-convex optimization, in International Conference on Machine Learning, 2016, pp. 699–707.
[3] Yair Carmon, John Duchi, Oliver Hinder, and Aaron Sidford, Accelerated methods for non-convex optimization, arXiv preprint arXiv:1611.00756, (2016).
[4] Cong D. Dang and Guanghui Lan, Stochastic block mirror descent methods for nonsmooth and stochastic optimization, SIAM Journal on Optimization, 25 (2015), pp. 856–881.
[5] Damek Davis and Benjamin Grimmer, Proximally guided stochastic subgradient method for nonsmooth, nonconvex problems, arXiv preprint arXiv:1707.0550v4, (2017).
[6] Jianqing Fan and Runze Li, Variable selection via nonconcave penalized likelihood and its oracle properties, Journal of the American Statistical Association, 96 (2001), pp. 134–161.
[7] Saeed Ghadimi and Guanghui Lan, Stochastic first-and-zerth-order methods for nonconvex stochastic optimization, SIAM Journal on Optimization, 23 (2013), pp. 2341–2368.
[8] -------, Accelerated gradient methods for nonconvex nonlinear and stochastic programming, Mathematical Programming, 156 (2016), pp. 59–99.
[9] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang, Generalized uniformly optimal methods for nonlinear programming, Mathematical Programming, 155 (2016), pp. 267–305.
[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016. [http://www.deeplearningbook.org]
[11] Mingyi Hong, Decomposing linearly constrained nonconvex problems by a proximal primal dual approach: Algorithms, convergence, and applications, arXiv preprint arXiv:1604.00543, (2016).
[12] Mingyi Hong, Zhe-Quan Luo, and Miasam Razaviyayn, Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems, SIAM Journal on Optimization, 26 (2016), pp. 337–364.
[13] Rie Johnson and Tong Zhang, Accelerating stochastic gradient descent using predictive variance reduction, in Advances in neural information processing systems, 2013, pp. 315–323.
[14] Weijie Kong, Jefferson G Melo, and Renato DC Monteiro, Complexity of a quadratic penalty accelerated inexact proximal point method for solving linearly constrained nonconvex composite programs, arXiv preprint arXiv:1802.03504, (2018).
[15] Guanghui Lan and Yi Zhou, An optimal randomized incremental gradient method, Mathematical programming, (2017), pp. 1–49.
[16] Jefferson G Melo and Renato DC Monteiro, Iteration-complexity of a jacobi-type non-euclidean admm for multi-block linearly constrained nonconvex programs, arXiv preprint arXiv:1705.07229, (2017).
[17] -------, Iteration-complexity of a linearized proximal multi-block admm class for linearly constrained nonconvex optimization problems, Available on: http://www.optimization-online.org, (2017).
[18] Yuri Nesterov, A method for unconstrained convex minimization problem with the rate of convergence $o (1/k^2)$, in Doklady AN USSR, vol. 269, 1983, pp. 543–547.
[19] Y. E. Nesterov, Introductory Lectures on Convex Optimization: a basic course, Kluwer Academic Publishers, Massachusetts, 2004.
[20] Courtney Paquette, Hongzhou Lin, Dmitriy Drusvyatskiy, Julien Mairal, and Zaid Harchaoui, Catalyst acceleration for gradient-based non-convex optimization, arXiv preprint arXiv:1703.10993, (2017).
[21] Sashank J Reddi, Ahmed Hefny, Suurit Sra, Barnabás Poczos, and Alex Smola, Stochastic variance reduction for nonconvex optimization, in International conference on machine learning, 2016, pp. 314–323.
[22] Sashank J Reddi, Suurit Sra, Barnabas Poczos, and Alex Smola, Fast incremental method for smooth nonconvex optimization, in Decision and Control (CDC), 2016 IEEE 55th Conference on, IEEE, 2016, pp. 1971–1977.
[23] Robert Tibshirani, G James, D Witten, and T Hastie, An introduction to statistical learning-with applications in r, 2013.
[24] Yu Wang, Wotao Yin, and Jinhua Zeng, Global convergence of admm in nonconvex nonsmooth optimization, arXiv preprint arXiv:1511.06324, (2015).
[25] Yinyu Ye, Randomly sampled cyclic alternating direction method of multipliers, private communication, (2018).
[26] Cun-Hui Zhang et al., Nearly unbiased variable selection under minimax concave penalty, The Annals of statistics, 38 (2010), pp. 894–942.
[27] Cun-Hui Zhang and Tong Zhang, A general theory of concave regularization for high-dimensional sparse estimation problems, Statistical Science, (2012), pp. 576–593.
Appendix A. Proof of Lemma 2.4

Proof. By convexity of $\psi$ and optimality of $x^*$, we have

\[
Q_t := \varphi(x^t) + \psi(x^t) + \langle \nabla \psi(x^t), x^t - x^* \rangle - [\varphi(x^*) + \frac{1}{m} \sum_{i=1}^m (\psi_i(x^t) + \langle \nabla \psi_i(x^t), x^* - x^t \rangle)] \\
\geq \varphi(x^t) + \psi(x^t) + \langle \nabla \psi(x^t), x^t - x^* \rangle - [\varphi(x^*) + \psi(x^*)] \\
= \varphi(x^t) - \varphi(x^*) + \langle \nabla \psi(x^t), x^t - x^* \rangle \geq \langle \varphi(x^*) + \nabla \psi(x^t), x^t - x^* \rangle \geq 0. \tag{A.1}
\]

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

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

\[
\begin{align*}
\delta_1^t & := \psi(x^*) - \langle \nabla \psi(x^*), x^t \rangle - \frac{1}{m} \sum_{i=1}^m \psi_i(x^t) - \langle \nabla \psi_i(x^*), x^t \rangle + \langle \nabla \psi_i(x^*), \tilde{x} \rangle \\
& = \frac{1}{m} \sum_{i=1}^m [1 \Psi(x^t, t) - (1 + \tau_t) \Psi(x^t - x^*, t)] \\
\delta_2^t & := \frac{1}{m} \sum_{i=1}^m [\langle \nabla \psi_i(x^*) - \nabla \psi_i(x^t), \tilde{x} \rangle - \langle \tilde{y}_i^t - \nabla \psi_i(x^t), x^t \rangle + \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^t \rangle]. \tag{A.3}
\end{align*}
\]

In view of \(2.13\), we have

\[
\mathbb{E}_\eta \delta_2^t = \frac{1}{m} \sum_{i=1}^m \mathbb{E}_\eta \left[ \langle \nabla \psi_i(x^t) - \nabla \psi_i(x^*), \tilde{x} \rangle - \langle \tilde{y}_i^t - \nabla \psi_i(x^t), x^t \rangle + \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^t \rangle \right] \\
= \frac{1}{m} \sum_{i=1}^m \mathbb{E}_\eta \langle \tilde{y}_i^t - \nabla \psi_i(x^*), \tilde{x} - x^t \rangle. \tag{A.4}
\]

Multiplying each $Q_t$ by a non-negative $\gamma_t$ and summing them up, we obtain

\[
\begin{align*}
\mathbb{E}_s \left[ \sum_{t=1}^s \gamma_t Q_t \right] & \leq \mathbb{E}_s \left[ \sum_{t=1}^s \gamma_t \left[ \eta_t \Psi(x^t, x^{t-1}) - (1 + \eta_t) \Psi(x^t, x^*) \right] \right] \\
& + \mathbb{E}_s \left[ \sum_{t=1}^s \gamma_t \left[ \frac{1}{m} \sum_{i=1}^m \left( \frac{1}{\tau_t} \left( 1 + \tau_t - \frac{1}{m} \sum_{i=1}^m \eta_t \Psi(x^t, x^*) - \frac{1}{m} \sum_{i=1}^m \eta_t \Psi(x^t, x^{t-1}) \right) \right) \right] \right] \\
& + \mathbb{E}_s \left[ \sum_{t=1}^s \gamma_t \left[ \frac{1}{m} \sum_{i=1}^m \left( \frac{\alpha_t}{\tau_t} \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^t - x^* \rangle - \tau_t \Psi(x^t, x^*) \right) \right] \right] \\
& \leq \mathbb{E}_s \left[ \sum_{t=1}^s \gamma_t \left[ \eta_t \Psi(x^t, x^*) - (1 + \eta_t) \Psi(x^t, x^*) \right] \right] \\
& + \mathbb{E}_s \left[ \sum_{t=1}^s \gamma_t \left[ \frac{1}{m} \sum_{i=1}^m \left( \frac{1}{\tau_t} \left( 1 + \tau_t - \frac{1}{m} \Psi(x^t, x^*) - \frac{1}{m} \Psi(x^t, x^{t-1}) \right) \right) \right] \right] \tag{A.5}
\end{align*}
\]

where

\[
\begin{align*}
\delta_t : = & \eta_t \Psi(x^t, x^{t-1}) - \frac{1}{m} \sum_{i=1}^m \left[ \frac{1}{\alpha_t} \langle \tilde{y}_i^t - \nabla \psi_i(x^*), \tilde{x} - x^t \rangle - \tau_t \Psi(x^t, x^*) \right] \\
= & \eta_t \Psi(x^t, x^*) - \frac{1}{m} \sum_{i=1}^m \langle \tilde{y}_i^t - \nabla \psi_i(x^*), \tilde{x} - x^t \rangle + \tau_t \Psi(x^t, x^*). \tag{A.6}
\end{align*}
\]

the first inequality follows from \(A.2\), \(A.3\), \(2.25\), \(A.4\) and Lemma 2.3 and the second inequality is implied by \(2.15\) and \(2.10\).

By the definition of $\tilde{x}$ in \(2.6\), we have

\[
\begin{align*}
\frac{1}{m} \sum_{i=1}^m \langle \tilde{y}_i^t - \nabla \psi_i(x^*), \tilde{x} - x^t \rangle \\
= \frac{1}{m} \sum_{i=1}^m \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^{t-1} - x^t \rangle - \alpha_t \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^{t-2} - x^{t-1} \rangle \\
= \frac{1}{m} \sum_{i=1}^m \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^{t-1} - x^t \rangle - \alpha_t \langle \tilde{y}_i^{t-1} - \nabla \psi_i(x^*), x^{t-2} - x^{t-1} \rangle - \alpha_t \langle \tilde{y}_i^t - \tilde{y}_i^{t-1}, x^{t-2} - x^{t-1} \rangle \\
= \frac{1}{m} \sum_{i=1}^m \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^{t-1} - x^t \rangle - \alpha_t \langle \tilde{y}_i^{t-1} - \nabla \psi_i(x^*), x^{t-2} - x^{t-1} \rangle - \alpha_t \langle \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}), x^{t-2} - x^{t-1} \rangle \\
- (1 - \frac{1}{m}) \alpha_t \langle \nabla \psi_{i_{t-1}}(x^t) - \nabla \psi_{i_{t-1}}(x^{t-1}), x^{t-2} - x^{t-1} \rangle. \tag{A.7}
\end{align*}
\]

From the relation \(2.6\) and the fact $x^{t-1} = x^0$, we have

\[
\begin{align*}
\sum_{t=1}^s \gamma_t \frac{1}{m} \sum_{i=1}^m \left[ \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^{t-1} - x^t \rangle - \alpha_t \langle \tilde{y}_i^{t-1} - \nabla \psi_i(x^*), x^{t-2} - x^{t-1} \rangle \right] \\
= \gamma_s \sum_{t=1}^s \left[ \langle \tilde{y}_i^t - \nabla \psi_i(x^*), x^{t-1} - x^t \rangle \frac{1}{m} \left( \langle \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}) \rangle - \langle \tilde{y}_i^{t-1} - \tilde{y}_i^t, x^{t-2} - x^{t-1} \rangle \right) \rangle \\
= \gamma_s \sum_{t=1}^s \left[ \langle \nabla \psi_i(x^t) - \nabla \psi_i(x^*), x^{t-1} - x^t \rangle + \gamma_s \left( 1 - \frac{1}{m} \right) \langle \nabla \psi_{i_{t-1}}(x^t) - \nabla \psi_{i_{t-1}}(x^{t-1}), x^{t-2} - x^{t-1} \rangle \right]. \tag{A.8}
\end{align*}
\]
Now we are ready to bound the last term in (A.10) as follows:

\[
\sum_{i=1}^{s} \gamma_i \delta_i \geq \sum_{i=1}^{s} \gamma_i \left[ \frac{\nu t}{2} \| x^t - x^{t-1} \|^2 - \frac{1}{m} \sum_{i=1}^{m} \langle \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}), x^t - x^{t-1} \rangle + \tau_i \Psi(x_{i-1}^{t-1}, x_i^t) \right]
\]

\[
= \sum_{i=1}^{s} \gamma_i \left[ \nu t \| x^t - x^{t-1} \|^2 + \alpha_i (1 - \frac{1}{m}) \langle \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}), x^t - x^{t-1} \rangle + \frac{\nu t}{2} \| \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}) \|^2 \right]
\]

where (a) follows from the definition \( \delta_i \) in (A.6), (b) follows relations (A.7) and (A.8) and (c) follows from the fact that \( \nu t \| x^t - x^{t-1} \|^2 \). By properly regrouping the term on the right hand side of (A.9), we have

\[
\sum_{i=1}^{s} \gamma_i \delta_i \geq \sum_{i=1}^{s} \gamma_i \left[ \frac{\nu t}{2} \| x^t - x^{t-1} \|^2 - \frac{1}{m} \sum_{i=1}^{m} \langle \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}), x^t - x^{t-1} \rangle + \tau_i \Psi(x_{i-1}^{t-1}, x_i^t) \right]
\]

\[
= \sum_{i=1}^{s} \gamma_i \left[ \frac{\nu t}{2} \| x^t - x^{t-1} \|^2 + \alpha_i (1 - \frac{1}{m}) \langle \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}), x^t - x^{t-1} \rangle + \frac{\nu t}{2} \| \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}) \|^2 \right]
\]

where (a) follows from the simple relation that \( \nu t \| x^t - x^{t-1} \|^2 \). By using the above inequality, (A.9) and (A.10), we obtain

\[
0 \leq \sum_{i=1}^{s} \gamma_i \left[ \frac{\nu t}{2} \| x^t - x^{t-1} \|^2 - \frac{1}{m} \sum_{i=1}^{m} \langle \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}), x^t - x^{t-1} \rangle + \tau_i \Psi(x_{i-1}^{t-1}, x_i^t) \right]
\]

\[
= \sum_{i=1}^{s} \gamma_i \left[ \frac{\nu t}{2} \| x^t - x^{t-1} \|^2 + \alpha_i (1 - \frac{1}{m}) \langle \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}), x^t - x^{t-1} \rangle + \frac{\nu t}{2} \| \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}) \|^2 \right]
\]

where (a) follows from the simple relation that \( \nu t \| x^t - x^{t-1} \|^2 \). By using the above inequality, (A.9) and (A.10), we obtain

\[
0 \leq \sum_{i=1}^{s} \gamma_i \left[ \frac{\nu t}{2} \| x^t - x^{t-1} \|^2 - \frac{1}{m} \sum_{i=1}^{m} \langle \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}), x^t - x^{t-1} \rangle + \tau_i \Psi(x_{i-1}^{t-1}, x_i^t) \right]
\]

\[
= \sum_{i=1}^{s} \gamma_i \left[ \frac{\nu t}{2} \| x^t - x^{t-1} \|^2 + \alpha_i (1 - \frac{1}{m}) \langle \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}), x^t - x^{t-1} \rangle + \frac{\nu t}{2} \| \nabla \psi_i(x^t) - \nabla \psi_i(x^{t-1}) \|^2 \right]
\]
where (a) follows from \( \Psi_t(x, x^*) \geq \frac{1}{2 \mu} \| \nabla \psi_t(y) - \nabla \psi_t(x^*) \|^2 \); (b) follows from the simple relation that \( b(u, v) - a\|v\|^2/2 \leq \nu u - (a/2) \), \( \forall \alpha > 0 \) and (c) follows from \( \ref{eq:lemma2} \), strong convexity of \( \psi_t \) and Lipschitz continuity of \( \nabla \psi_t \). This completes the proof. \( \square \)

Appendix B. Proof of Lemma 3.4

\[ \text{Proof.} \] For any \( t \geq 1 \), since \( (x^*, y^*) \) is a saddle point of \( \ref{eq:lemma4} \), we have

\[ \psi(x^t) - \psi(x^*) + \langle A x^t - b, y^* \rangle - \langle A x^* - b, y^t \rangle + h(y^*) - h(y^t) \geq 0. \]

For nonnegative \( \gamma_t \), we further obtain

\[ \mathbb{E}_s \left\{ \sum_{t=1}^s \gamma_t \left[ \psi(x^t) - \psi(x^*) + \langle A x^t - b, y^* \rangle - \langle A x^* - b, y^t \rangle + h(y^*) - h(y^t) \right] \right\} \geq 0. \quad \text{(B.1)} \]

According to optimality conditions of \( \ref{eq:lemma1} \) and \( \ref{eq:lemma2} \) respectively, and strongly convexity of \( \psi \) and \( h \) we obtain

\[ \psi(x^t) - \psi(x^*) + \frac{\mu}{2} \| x^t - \bar{x}^t \|^2 + \| A^\top y^t, \bar{x}^t - x^* \| \leq \frac{\mu}{2} \left[ \| x^t - x^{t-1} \|^2 - \| x^* - \bar{x}^t \|^2 - \| \bar{x}^t - x^{t-1} \|^2 \right], \]

\[ h(y^t) - h(y^*) + \langle - A x^t + b, y^t - y^* \rangle \leq \tau_t V_t(y^*, y^{t-1}) - (\tau_t + 1) V_t(y^*, y^t) - \tau_t V_t(y^t, y^{t-1}). \]

Combining the above two inequalities with relation \( \text{(B.1)} \), we have

\[ \mathbb{E}_s \left\{ \sum_{t=1}^k \left[ \gamma_t \left( \| x^t - x^{t-1} \|^2 - \frac{\gamma_t(\gamma_t + 1)}{2} \| x^t - x^{t-1} \|^2 \right) \right] \right\} + \mathbb{E}_s \left\{ \sum_{t=1}^s \gamma_t \left( \tau_t V_t(y^*, y^{t-1}) - (\tau_t + 1) V_t(y^*, y^t) - \tau_t V_t(y^t, y^{t-1}) \right) \right\} + \mathbb{E}_s \left\{ \sum_{t=1}^s \gamma_t(\langle A x^t - \bar{x}^t, y^* - y^t \rangle) \right\} \geq 0. \]

Observe that for \( t \geq 1 \),

\[ \mathbb{E}_t \left\{ \langle A(x^t - \bar{x}^t), y^* \rangle \right\} = \mathbb{E}_t \left\{ \langle (m-1)x^t - (m-2)x^{t-1} - \bar{x}^t, y^* \rangle \right\}. \]

Applying this and the results \( \ref{eq:lemma1} \), \( \ref{eq:lemma2} \) in Lemma 3.3 we further have

\[ 0 \leq \mathbb{E}_s \left\{ \sum_{t=1}^s \left[ \gamma_t(\mu(m-1)+\mu(m-2)) \| x^t - x^{t-1} \|^2 - \frac{\gamma_t(m+1)}{2} \| x^t - x^{t-1} \|^2 \right] \right\} + \mathbb{E}_s \left\{ \sum_{t=1}^s \left[ \tau_t V_t(y^*, y^{t-1}) - (\tau_t + 1) V_t(y^*, y^t) - \tau_t V_t(y^t, y^{t-1}) \right] \right\} \]

\[ \leq \mathbb{E}_s \left\{ \left[ \frac{\gamma_t(m+1)+\mu(m-2)}{2} \| x^t - x^0 \|^2 - \frac{\gamma_t(m+1)}{2} \| x^t - x^0 \|^2 \right] \right\} + \mathbb{E}_s \left\{ \gamma_s(\tau_s V_s(y^*, y^0) - \gamma_s(\tau_s + 1) V_s(y^*, y^s)) + \mathbb{E}_s \left[ \sum_{t=1}^s \gamma_t \delta_t \right] \right\}, \quad \text{(B.2)} \]

where

\[ \delta_t = -(m-1)\gamma_t \| x^t - x^{t-1} \|^2 - \tau_t V_t(y^*, y^{t-1}) + \langle A((m-1)x^t - (m-2)x^{t-1} - \bar{x}^t), y^* - y^t \rangle. \]

and the second inequality follows from \( \ref{eq:lemma1} \) and \( \ref{eq:lemma2} \).

By \( \ref{eq:lemma1} \) and the definition of \( \bar{x}^t \) in \( \ref{eq:lemma1} \) we have:

\[ \sum_{t=1}^s \gamma_t \delta_t = \sum_{t=1}^s \left[ -\frac{\gamma_t(m+1)}{2} \| x^t - x^{t-1} \|^2 - \tau_t V_t(y^*, y^{t-1}) \right] + \frac{\gamma_t(m-1)\alpha_t}{2} \left( A(x^{t-1} - x^{t-2}), y^* - y^t \right) - \sum_{t=1}^s \gamma_t(m-1)\alpha_t \left( A(x^{t-1} - x^{t-2}), y^* - y^t \right)
\]

\[ = \sum_{t=1}^s \left[ -\frac{\gamma_t(m-1)\alpha_t}{2} \| x^t - x^{t-1} \|^2 - \tau_t V_t(y^*, y^{t-1}) \right] + \gamma_s(m-1)\left( A(x^s - x^{s-1}), y^* - y^s \right)
\]

\[ - \sum_{t=1}^s \gamma_t(m-1)\alpha_t \left( A(x^{t-1} - x^{t-2}), y^{t-1} - y^t \right), \quad \text{(B.3)} \]

where the second equality follows from \( \ref{eq:lemma1} \) and the fact that \( x^0 = x^{-1} \).
Since \( \langle A(x^{t-1} - x^{t-2}), y^t - y^f \rangle = \langle A_{i_{t-1}}(x^{t-1}_{i_{t-1}} - x^{t-2}_{i_{t-1}}), y^t - y^f \rangle \leq \| A_{i_{t-1}} \| \| x^{t-1}_{i_{t-1}} - x^{t-2}_{i_{t-1}} \| \| y^t - y^{t-1} \| \)
and \( V_h(y^t, y^{t-1}) \geq \frac{\mu}{2} \| y^t - y^{t-1} \|^2 \), from (B.3) we have
\[
\sum_{t=1}^{s} \gamma_t \delta_t \leq \sum_{t=1}^{s} \left[ -\gamma_t (m-1) \| x_t - x^{t-1} \|^2 - \frac{\mu \gamma_t}{2} \| y^t - y^{t-1} \|^2 \right] + \gamma_s (m-1) \langle A(x^s - x^{s-1}), y^s - y^s \rangle
- \sum_{t=1}^{s} \gamma_t (m-1) \bar{A}_t \| x_t - x^{t-1} \| \| y^t - y^{t-1} \|
\geq \gamma_s (m-1) \langle A(x^s - x^{s-1}), y^s - y^s \rangle - \frac{(m-1) \gamma_t \bar{A}_t}{2} \| x_t - x^{t-1} \|^2 + \gamma_s (m-1) \langle A(x^s - x^{s-1}), y^s - y^s \rangle
+ \frac{(m-1) \gamma_t \bar{A}_t}{2} \| x_t - x^{t-1} \|^2 - \frac{\mu \gamma_t}{2} \| y^t - y^{t-1} \|^2.
\]
where (a) follows from regrouping the terms; (b) follows from the definition \( \bar{A} = \max_{i \in [m-1]} \| A_i \| \) and the simple relation that \( \langle b, u \rangle - a \| v \|^2 / 2 \leq b^2 \| u \|^2 / (2a) \), \( \forall a > 0 \); and (c) follows from (B.17) and (B.20).

By combining the relation above with (B.2), we obtain
\[
0 \leq \mathbb{E}_s \left[ \frac{(m-1) \gamma_t \bar{A}_t}{2} \| x^s_t - x^{s-1}_t \|^2 - \frac{(m-1) \gamma_t \bar{A}_t}{2} \| x^s_t - x^{s-1}_t \|^2 \right]
+ \mathbb{E}_s \left[ \gamma_t \bar{A}_t (y^t - y^s) - \gamma_s (m-1) \bar{A}_t (y^s - y^s) \right]
+ \mathbb{E}_s \left[ \gamma_t \bar{A}_t (y^s - y^s) - \gamma_s (m-1) \bar{A}_t (y^s - y^s) \right].
\]

Notice the fact that
\[
\mathbb{E}_s \left[ \frac{\gamma_s (m-1) \bar{A}_t}{2} \| y^s - y^s \|^2 \right]
= \mathbb{E}_s \left[ \frac{\gamma_s (m-1) \bar{A}_t}{2} \| y^s - y^s \|^2 \right]
\geq \gamma_s \mathbb{E}_s \left[ \frac{(m-1) \bar{A}_t}{2} \| y^s - y^s \|^2 \right] \geq 0.
\]

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