Accelerated Randomized Mirror Descent Algorithms for Composite Non-strongly Convex Optimization

Le Thi Khanh Hien1 · Cuong V. Nguyen2 · Huan Xu3 · Canyi Lu4 · Jiashi Feng1

Received: 16 April 2018 / Accepted: 31 December 2018 / Published online: 16 January 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract
We consider the problem of minimizing the sum of an average function of a large number of smooth convex components and a general, possibly non-differentiable, convex function. Although many methods have been proposed to solve this problem with the assumption that the sum is strongly convex, few methods support the non-strongly convex case. Adding a small quadratic regularization is a common devise used to tackle non-strongly convex problems; however, it may cause loss of sparsity of solutions or weaken the performance of the algorithms. Avoiding this devise, we propose an accelerated randomized mirror descent method for solving this problem without the strongly convex assumption. Our method extends the deterministic accelerated proximal gradient methods of Paul Tseng and can be applied, even when proximal points are computed inexacty. We also propose a scheme for solving the problem, when the component functions are non-smooth.

Keywords Acceleration techniques · Mirror descent method · Inexact proximal point · Composite optimization

Mathematics Subject Classification 65K05 · 90C06 · 90C30

1 Introduction

Composite convex optimization problem (CCOP) is the problem of minimizing an objective function which is the sum of an average function of a large number of smooth convex components and a general convex function. We formally define CCOP in Sect. 2. The problems of this form often appear in machine learning and statistics,
such as regularized empirical risk minimization problem, regularized logistic regression, and Łasso problem. One well-known method to solve CCOP is the proximal gradient descent (PGD) method. Methods such as gradient descent or projection gradient descent are in the class of PGD algorithms. However, the convergence rate of PGD is not optimal. Nesterov, for the first time in [1], proposed an acceleration method for solving CCOP with the general convex component being an indicator function and achieved the optimal convergence rate. Later in [2–4], he introduced two other acceleration techniques, which make one or two proximal calls together with interpolation at each iteration to accelerate the convergence. Nesterov’s ideas have been further studied and applied for solving many practical optimization problems such as rank reduction in multivariate linear regression and sparse covariance selection (see [5,6] and reference therein). Auslender and Teboulle [7] used the acceleration technique in the context of Bregman divergence, which generalizes the squared Euclidean distance. Tseng [8] unified the analysis of all these acceleration techniques, proposed new variants and gave simple analysis for the proof of the optimal convergence rate.

When the number of the components of the average function is very large, applying PGD can be unappealing since computing the full gradient in each iteration is very expensive. An effective alternative is the randomized version of PGD, which is usually called stochastic proximal gradient descend (SPGD) method. For a suitably chosen decreasing step size, SPGD was proven to have the suboptimal rate, when the objective function is strongly convex [9]. Many authors have proposed methods to obtain better convergence rate when the objective function is strongly convex; stochastic average gradient (SAG) [10], stochastic variance reduced gradient (SVRG) [11], proximal SVRG [12], and SARAH [13] are noticeable examples that have a linear convergence rate, which is optimal for this case.

There have been very few algorithms that directly support the non-strongly convex objective function. One of those is the randomized coordinate gradient descent method (RCGD), which recently has been successfully extended to accelerated versions to achieve the optimal rate [14]. However, accelerated RCGD is only applicable to block separable regularization. It is worth mentioning catalyst scheme which accelerates first-order methods to achieve a better rate [15, Alg. 1]. However, catalyst approximately solves a sequence of auxiliary problems, which are formed by adding a strongly convex regularizer to the objective function. Therefore, we cannot consider catalyst as a direct method for solving the non-strongly convex problems. Furthermore, using this scheme for first-order method to solve non-strongly convex problems only obtains near-optimal rate [15, Sect. 3.2]. To the best of our knowledge, accelerated proximal gradient descent (APG) is the only algorithm that obtains the optimal rate for directly solving CCOP under some suitable assumptions. Therefore, the main goal of this work is to extend APG to randomized variants that support the non-strongly convex case and even outperform APG on large-scale problems.

Our second goal is to solve CCOP when proximal points cannot be computed explicitly. For several choices of the general non-smooth component of the objective function, the proximal points used in the above-mentioned algorithms can be calculated efficiently [16]. However, in many cases such as nuclear norm regularization and total variation regularization, it is very expensive to compute them exactly. For that reason, many efficient methods have been proposed to calculate proximal
points inexactly [17–19]. Basic methods that allow inexact computation of proximal points were first studied by Rockafellar [20]. Since then, there has emerged a growing interest in both inexact proximal point and inexact accelerated proximal point algorithms [17,21–24]. However, although there were many works showing impressive empirical performance of inexact (accelerated) PGD methods, there has been no analysis on their randomized versions. Our work gives such an analysis.

In a concurrent work [25], an exact accelerated randomized gradient descent in the setting with Euclidean distance was independently analyzed and the same convergence rate was proven. In comparison, as we extend the general deterministic acceleration framework of Tseng [8] and consider non-uniform sampling together with a broader choice of involving parameters, our analysis for inexact accelerated randomized mirror descent algorithm is put in a more general framework but employs simpler and neater proofs in the setting with Bregman distance. By contrast, the analysis of [25] depends critically on a specific choice in Update (4) of our Algorithm 3.1. In particular, their proofs critically rely on using Variant 1 of our Example 3.1.

Our main contribution is the incorporation of the variance reduction technique and the general acceleration methods of Tseng to propose a framework of exact as well as inexact accelerated randomized mirror descent (ARMD and inexact ARMD, respectively) algorithms for the non-strongly convex CCOP. At each stage of our inexact algorithms, proximal points are allowed to be calculated inexactly. When the component functions of the average function are non-smooth, we give a scheme for minimizing the corresponding non-smooth problem. The rate obtained using our smoothing scheme significantly improves the rate obtained using subgradient methods or stochastic subgradient methods.

**Paper Structure** Section 2 provides some preliminaries. The proposed ARMD and inexact ARMD methods and their convergence analysis are presented in Sect. 3. Computational results are given in Sect. 4, and the proofs are in “Appendix.”

### 2 Preliminaries

We let $\mathcal{E}$ be a finite-dimensional real linear space endowed with a norm $\|\cdot\|$ and let $\mathcal{E}^*$ be the space of continuous linear functionals on $\mathcal{E}$. We use $\langle x^*, x \rangle$ to denote the value of $x^* \in \mathcal{E}^*$ at $x \in \mathcal{E}$, and $\|\cdot\|_*$ to denote the dual norm, i.e., $\|x^*\|_* := \sup_{\|x\| \leq 1} \langle x^*, x \rangle$.

We consider the following composite convex optimization problem:

$$
\min_{x \in \mathcal{E}} \left\{ F^p(x) := F(x) + P(x) \right\},
$$

where $F(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$. Throughout this paper, we focus on problems satisfying the following assumption.

**Assumption 2.1** Function $P(x)$ is lower semi-continuous and convex. The domain of $P(x)$, $\text{dom} \ P := \{ x \in \mathcal{E}, \ P(x) < +\infty \}$, is closed. Each function $f_i(x)$ is convex and $L_i$-Lipschitz smooth, i.e., it is differentiable on an open set containing $\text{dom} \ P$ and its gradient is Lipschitz continuous with constant $L_i$.
\[ \| \nabla f_i(x) - \nabla f_i(y) \|_2^2 \leq L_i \| x - y \|, \forall x, y \in \text{dom} P. \]

For a given continuous function \( f(x) \), a convex set \( X \), and a nonnegative number \( \varepsilon \), we write \( z \approx \varepsilon \arg \min_{x \in X} \{ f(x) + P(x) \} \) to denote \( z \in \text{dom} P \) such that \( f(z) + P(z) \leq \min_{x \in X} \{ f(x) + P(x) \} + \varepsilon \). We use \( \nabla_1 g(x, y) \) to denote the gradient of the function \( x \mapsto g(x, y) \) at \( x \).

**Lemma 2.4** If \( f_i \) is crucial for our analysis.

**Definition 2.1** Let \( h: \Omega \to \mathbb{R} \) be a strictly convex function that is differentiable on an open set containing \( \text{dom} P \). The Bregman distance is defined as:

\[ D(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle, \quad \forall y \in \text{dom} P, x \in \Omega. \]

**Example 2.1** (a) If \( h(x) = \frac{1}{2} \| x \|_2^2 \), then \( D(x, y) = \frac{1}{2} \| x - y \|_2^2 \) is the Euclidean distance.

(b) If \( \Omega = \{ x \in \mathbb{R}^P_+: \sum_{i=1}^P x_i = 1 \} \) and \( h(x) = \sum_{i=1}^P x_i \log x_i \), then we have the entropy distance \( D(x, y) = \sum_{i=1}^P x_i \log \frac{x_i}{y_i} \).

**Lemma 2.1** If \( h \) is strongly convex with constant \( \sigma \), i.e.,

\[ h(y) \geq h(x) + \langle \nabla h(x), y - x \rangle + \frac{\sigma}{2} \| x - y \|^2, \]

then the corresponding Bregman distance satisfies \( D(x, y) \geq \frac{\sigma}{2} \| x - y \|^2 \).

**Lemma 2.2** Let \( \phi(x) \) be a proper convex function whose domain is an open set containing \( C \). For any \( z \in \text{dom} P \), if \( z^* = \arg \min_{x \in C} \{ \phi(x) + D(x, z) \} \), then for all \( x \in \text{dom} P \) we have: \( \phi(x) + D(x, z) \geq \phi(z^*) + D(z^*, z) + D(x, z^*) \).

**Lemma 2.3** Let \( D(x, y) \) be the Bregman distance with respect to \( h \). We have:

\[ D(x, y) + D(y, z) = D(x, z) + \langle x - y, \nabla h(z) - \nabla h(y) \rangle. \]

If we replace the Euclidean distance in PGD by Bregman distance, we obtain the mirror descent method:

\[ x_k = \arg \min_x \left\{ \langle \nabla F(x_{k-1}), x \rangle + P(x) + \frac{1}{y_k} D(x, x_{k-1}) \right\}. \]

We refer the readers to [8,26,27] and references therein for proofs of these lemmas and applications of Bregman distance as well as mirror descent methods. Since we can scale \( h \) if necessary, we assume \( h \) is strongly convex with constant \( \sigma = 1 \) in this paper. The following lemma for \( L_i \)-Lipschitz smooth functions (with proof in [28]) is crucial for our analysis.

**Lemma 2.4** If \( f_i(\cdot) \) is a convex and \( L_i \)-Lipschitz smooth function, then:

1. \( f_i(x) \leq f_i(y) + \langle \nabla f_i(y), x - y \rangle + \frac{L_i}{2} \| x - y \|^2, \) and
2. \( \frac{1}{2L_i} \| \nabla f_i(y) - \nabla f_i(x) \|^2 + \| \nabla f_i(y) - \nabla f_i(x) \|_2^2 + \langle \nabla f_i(y), x - y \rangle \leq f_i(x) - f_i(y). \)
3 Accelerated Randomized Mirror Descent Methods

3.1 Algorithm Description

1: Choose \( \tilde{x}_0, x_{1,0}, \ldots, x_{m,0}, z_0 \in \text{dom} P, \alpha_3 > 0 \) and nonnegative sequences \( \{\alpha_{1,s}\} \), \( \{\alpha_{2,s}\} \) such that:

\[
\frac{1 - \alpha_{1,s}}{\alpha_{2,s}} \geq \frac{\alpha_3}{\alpha_{2,s+1}}
\]

2: Choose probability \( Q = \{q_1, \ldots, q_n\} \) on \( \{1, \ldots, n\} \).

3: Denote \( L_Q = \max_i \frac{L_i}{q_i/n} \), \( L_A = \frac{1}{n} \sum_{i=1}^n L_i \). Let \( \hat{T} = L_A + \frac{4L_Q}{\alpha_3} \) and \( z_{m,0} = z_0 \).

4: for \( s = 1, 2, \ldots \) do

5: Calculate \( \hat{v} = \nabla F(\tilde{x}_{s-1}) \).

6: Let \( x_{0,s} = x_{m,s-1} \) and \( z_{0,s} = z_{m,s-1} \).

7: Choose a nonempty closed convex set \( X_s \subseteq \mathcal{E} \) with \( X_s \cap \text{dom} P \neq \emptyset \).

8: for \( k = 1, \ldots, m \) do

9: - Pick \( i_k \in \{1, \ldots, n\} \) randomly according to \( Q \).

10: - Set \( y_{k,s} = \alpha_{1,s} x_{k-1,s} + \alpha_{2,s} z_{k-1,s} + \alpha_3 \tilde{x}_{s-1} \).

11: - Set \( v_k = \hat{v} + \nabla f_{i_k}(y_{k,s}) - \nabla f_{i_k}(\tilde{x}_{s-1}) \).

12: - Let \( \theta_s = \alpha_{2,s} \hat{T} \), update:

\[
z_{k,s} \approx_{\varepsilon_{k,s}} \arg \min_{x \in X_s} \{(v_k, x) + P(x) + \theta_s D(x, z_{k-1,s})\}.
\]

13: - Set \( \hat{x}_{k,s} = \alpha_{1,s} x_{k-1,s} + \alpha_{2,s} z_{k,s} + \alpha_3 \tilde{x}_{s-1} \).

14: - Choose \( x_{k,s} \) such that:

\[
\left\langle v_k, x_{k,s} \right\rangle + \frac{T}{2} \left\| x_{k,s} - y_{k,s} \right\|^2 + P(x_{k,s}) \\
\leq \left\langle v_k, \hat{x}_{k,s} \right\rangle + \frac{T}{2} \left\| \hat{x}_{k,s} - y_{k,s} \right\|^2 + P(\hat{x}_{k,s}).
\]

15: Update \( \tilde{x}_s \) such that:

\[
F^P(\tilde{x}_s) \leq \frac{1}{m} \sum_{i=1}^m F^P(x_{i,s}).
\]

Algorithm 3.1: Accelerated Randomized Mirror Descent (ARMD) Algorithm

Algorithm 3.1 details our framework. The algorithm has 2 loops—the outer loop indexed by \( s \) and the inner loop indexed by \( k \). Specifically, we use \( x_{k,s} \) to mean that the point is at step \( k \) of the inner loop, which belongs to stage \( s \) of the outer loop. Before running each inner loop, a full gradient \( \nabla F(\cdot) \) is calculated. Each inner loop is executed \( m \) steps, i.e., \( k = 1, \ldots, m \). At step \( k \) of an inner loop, under non-uniform sampling setting, we randomly pick one function \( f_{i_k} \) to calculate its derivative \( \nabla f_{i_k}(\cdot) \), then find the (inexact) proximal point in (3), and perform the update (4). We let \( \varepsilon_{k,s} \) be the error.
in calculating the proximal points in (3). When $\varepsilon_{k,s} = 0$, our algorithm reduces to exact ARMD. The non-uniform sampling method would improve the complexity of our algorithms when $L_i$ are different. The set $X_s$ should be chosen such that it contains a solution of (1) (see Proposition 3.2). Choosing $X_s = E$ is the simplest variant of $X_s$. To accelerate convergence, we can always choose a smaller $X_s$. We refer the readers to [8, Sect. 3] for examples of choosing smaller $X_s$ and omit the details here.

We stress that the update rule (4) of $x_{k,s}$ is very general, and as such we can derive many specific algorithms from the general framework 3.1. In particular, we give some examples that satisfy (4) below.

**Example 3.1**

1. $x_{k,s} = x_{k,s}^\nu$ with $x_{k,s}^\nu = \hat{x}_{k,s}$. For this choice, each step $k$ of an inner loop only computes one inexact/exact proximal point $z_{k,s}$.

2. $x_{k,s} = x_{k,s}^l$ with $x_{k,s}^l = \arg\min_{x \in X_s} \left\{ \langle v_k, x \rangle + \frac{L}{2} \| x - y_{k,s} \|^2 + P(x) \right\}$. For this choice, each step $k$ of an inner loop needs to compute two proximal points $z_{k,s}$ and $x_{k,s}$.

3. Let $x^1, x^2$ be two variants of $x_{k,s}$ that satisfy (4), then their convex combination $\lambda x^1 + (1 - \lambda) x^2$, where $0 \leq \lambda \leq 1$, is also a choice of $x_{k,s}$.

It is easy to check that setting $\alpha_{1,s} = 1 - \alpha_3 - \frac{2}{s+\nu}$, $\alpha_{2,s} = \frac{2}{s+\nu}$, $0 < \alpha_3 \leq \frac{\nu-1}{\nu+1}$, where $\nu \geq 2$, satisfies the condition (2). For the update rule (5), we can choose $\tilde{x}_s = \frac{1}{m} \sum_{i=1}^m x_{i,s}$ or $\tilde{x}_s = x_{i^*,s}$, where $i^* = \arg\min_i F^P(x_{i,s})$.

### 3.2 Convergence Analysis

We first give an upper bound for the variance of $v_k$ in Lemma 3.1. This lemma together with the inequalities in Lemmas 3.2 and 3.3 is then used to prove the upcoming Proposition 3.1, which provides a recursive inequality within $m$ steps of an inner loop of Algorithm 3.1.

**Lemma 3.1** Conditioned on $x_{k-1,s}$, we have the following expectation inequality with respect to $i_k$:

$$
\mathbb{E} \left\| \nabla F(y_{k,s}) - v_k \right\|_*^2 \leq 2\mathbb{E} \frac{1}{(ndi_k)^2} \left\| \nabla f_{i_k}(y_{k,s}) - \nabla f_{i_k}(\tilde{x}_{s-1}) \right\|_*^2 + 2 \left\| \nabla F(y_{k,s}) - \nabla F(\tilde{x}_{s-1}) \right\|_*^2.
$$

**Lemma 3.2** At each stage $s$ of the outer loop, the following inequality holds:

$$
F^P(x_{k,s}) \leq F(y_{k,s}) + \frac{\alpha_3}{8L} \left\| \nabla F(y_{k,s}) - v_k \right\|_*^2 + \alpha_{2,s} \left( \left\langle v_k, z_{k,s} - z_{k-1,s} \right\rangle + \theta_s D(z_{k,s}, z_{k-1,s}) \right) + P(\tilde{x}_{k,s}).
$$

We denote $\tilde{z}_{k,s} = \arg\min_{z \in X_s} \left\{ \langle v_k, x \rangle + P(x) + \theta_s D(x, z_{k-1,s}) \right\}$ in (3), i.e., $\tilde{z}_{k,s}$ is the exact proximal point of the randomized step $k$ at stage $s$. 

© Springer
Lemma 3.3 For all $x \in X_s \cap \text{dom } P$,

$$\left( v_k, z_{k,s} \right) + \theta_s D(z_{k,s}, z_{k-1,s}) \leq \left( v_k, x \right) + P(x) + \theta_s (D(x, z_{k-1,s}) - D(x, z_{k,s})) - P(z_{k,s}) + \varepsilon_{k,s} - \theta_s (D(z_{k,s}, \bar{z}_{k,s}) - D(x, z_{k,s})).$$

Proposition 3.1 Denote

$$r_{k,s} = \mathbb{E} \left[ a_{2,s} \theta_s (-D(z_{k,s}, \bar{z}_{k,s}) + (x - z_{k,s}, \nabla h(\bar{z}_{k,s}) - \nabla h(z_{k,s})) + a_{2,s} \varepsilon_{k,s} \right].$$

For any $x \in X_s \cap \text{dom } P$, if $F^P(x) \leq F^P(x_{k,s})$, then we have:

$$\mathbb{E} F^P(x_{k,s}) \leq \alpha_{1,s} \mathbb{E} F^P(x_{k-1,s}) + \alpha_{2,s} F^P(x) + \alpha_3 F^P(\bar{x}_{s-1}) + \alpha_{2,s}^2 \mathbb{E} D(x, z_{k-1,s}) - \mathbb{E} D(x, z_{k,s}) + r_{k,s}.$$

As $\bar{x}_{s-1}$ appears in the recursive inequality, the acceleration effect works through the outer loop. The following proposition is a consequence of Proposition 3.1. It leads to the optimal convergence rate with respect to $s$. Theorems 3.1 and 3.2 indicate the rates.

Proposition 3.2 Denote

$$d_{i,0} = F^P(x_{i,0}) - F^P(x^*)$$

and $d_s = \mathbb{E} \left( F^P(\bar{x}_s) - F^P(x^*) \right)$.

Let $x^*$ be the optimal solution of (1). Let $r_{k,s}^*$ be the value of $r_{k,s}$ at $x = x^*$. Suppose $x^* \in X_s$, then we have:

$$d_s \leq \alpha_{2,s}^2 \left( \frac{1 - \alpha_{2,1}}{\alpha_{2,1} \alpha_3 m} + \frac{m-1}{\alpha_{2,1} \alpha_3 m} \sum_{k=1}^{m-1} d_{k,0} + \frac{\bar{L}}{m \alpha_3} \left( \mathbb{E} D(x^*, z_{m,0}) - \mathbb{E} D(x^*, z_{m,s}) \right) \right) + \alpha_{2,s}^2 \sum_{i=1}^{s} \sum_{k=1}^{m} \frac{r_{k,i}^*}{\alpha_{2,i}^2 m \alpha_3}.$$

Theorem 3.1 (Convergence rate of exact ARMD) Suppose that $\varepsilon_{k,s} = 0$, $x^* \in X_s$, and $\alpha_{2,s} \leq \frac{2}{s+2}$, then we have:

$$\mathbb{E} \left( F^P(\bar{x}_s) - F^P(x^*) \right) = O \left( \frac{F^P(\bar{x}_0) - F^P(x^*) + \frac{4T}{m} D(x^*, z_0)}{(s+3)^2} \right).$$

Remark 3.1 In the case of exact ARMD, i.e., $\varepsilon_{k,s} = 0$, each stage $s$ of Algorithm 3.1 computes $(n + 2m)$ gradients (if we save all gradients when computing $\bar{v}$, then the number of gradients to be calculated at stage $s$ is only $n + m$); hence, Theorem 3.1
yields the complexity, i.e., total number of gradients computed to obtain an $\epsilon$-optimal solution of (1) is
\[
O \left( \left( \sqrt{F^P(\tilde{x}_0)} - F^P(x^*) + \frac{4L}{m} D(x^*, z_0) \right) \sqrt{\epsilon} - 3 \right) (n + 2m).
\]

The smallest possible value of $L_Q$ is $L_Q = L_A$ when we choose $q_i = L_i / \sum_j L_j$, i.e., the sampling probabilities of $f_i$ are proportional to their Lipschitz constants. In this case, $\bar{L} = (1 + 4/\alpha_3) L_A$. Now let us choose $m = O(n)$. Hiding $F^P$ and $D(x^*, z_0)$, we can rewrite the complexity as $O \left( \frac{n + \sqrt{n L_A}}{\sqrt{\epsilon}} \right)$. This improves upon the complexity of APG, which is $O \left( \frac{n \sqrt{L_A}}{\sqrt{\epsilon}} \right)$ (see [8, Corrolary 1]). This improvement results from the fact that each step of APG must compute the full gradient of $F(x)$, while our method only computes it periodically. In Sect. 4, we show experiments comparing exact ARMD with APG that verify this property.

**Theorem 3.2** (Convergence rate of inexact ARMD) Suppose that $h(\cdot)$ is $L_h$-Lipschitz smooth, and $\alpha_{2,s} \leq \frac{2}{s+2}, x^* \in X_s$. Let $\{\epsilon_s\}$ be a sequence of nonnegative numbers that satisfies $\sum_{i=1}^{\infty} \sqrt{\frac{\epsilon_i}{\alpha_{2,i}}} = \text{convergent}.$

(i) Assume that there exists a constant $C$ such that $\|z_{k,s}\| \leq C$ (for example, if $P$ is the indicator function of a bounded closed convex set, then its domain is naturally bounded; thus, as $z_{k,s} \in \text{dom} P$ we have $\{z_{k,s}\}$ is bounded) and we let the error $\epsilon_{k,s}$ in (3) be $\epsilon_{k,s} = \epsilon_s$ for $k = 1, \ldots, m$, then there exists a constant $\tilde{C} = O(\sqrt{C})$ such that
\[
\mathbb{E}(F^P(\tilde{x}_s) - F^P(x^*)) = O \left( \frac{F^P(\tilde{x}_0) - F^P(x^*) + \frac{4L}{m} D(x^*, z_0) + \tilde{C}}{(s + 3)^2} \right).
\]

(ii) Assume that $P$ is bounded below (this assumption is satisfied in many practical problems, for instance, the examples in the introduction section satisfying $P(x) \geq 0, \forall x \in \mathcal{E}$), and we use the following adaptive update rule
\[
\max \left\{ \|z_{k,s}\|^2 \epsilon_{k,s}, C \epsilon_{k,s} \right\} \leq C \epsilon_s,
\]
where $C$ is some constant, then there also exists a constant $\tilde{C} = O(\sqrt{C})$ such that (6) is satisfied.

**Remark 3.2** The type of the inexact proximal point in (3) has been first considered in [29]. It can be checked by verifying the duality gap while solving the minimization problem in (3), see e.g., [22,24]. To use the adaptive update rule (7), we give an upper bound for $\|z_{k,s}\|$ before computing the inexact proximal point in (3) as follows. Without lost of generality, we assume that $P(x) \geq 0, \forall x \in \mathcal{E}$. Since $\bar{z}_{k,s}$ is the exact proximal point of (3),
\[
\{v_k, \bar{z}_{k,s}\} + P(\bar{z}_{k,s}) + \theta_s D(\bar{z}_{k,s}, z_{k-1,s}) \leq \{v_k, z_{k-1,s}\} + P(z_{k-1,s}).
\]
This implies \( \frac{\theta_s}{2} \| \bar{z}_{k,s} - z_{k-1,s} \|_2^2 \leq \| v_k \|_* \| \bar{z}_{k,s} - z_{k-1,s} \| + P(z_{k-1,s}) \). Hence, we have

\[
\| z_{k,s} \| \leq \| z_{k,s} - z_{k-1,s} \| + \| z_{k-1,s} \| \leq \frac{\| v_k \|_*}{\theta_s} + \sqrt{\frac{\| v_k \|_*^2}{\theta_s^2} + \frac{2P(z_{k-1})}{\theta_s}} + \| z_{k-1,s} \|.
\]

**Remark 3.3** In the case of inexact ARMD, we require the series \( \sum_{i=1}^{\infty} \sqrt{\frac{\epsilon_s}{\alpha_{2,i}}} \) to be convergent to obtain the convergence rate \( O(1/s^2) \). A sufficient condition is that \( \epsilon_s \) decreases as \( O(1/s^{3+\delta}) \) for \( \delta > 0 \). It is worth mentioning that for the deterministic inexact proximal gradient (PG) method considered in [22], the sequence of errors \( \{\epsilon_s\} \) is required to decrease as \( O(1/s^{2+\delta}) \) for the basic PG method (see [22, Proposition 1]), and as \( O(1/s^{4+\delta}) \) for the accelerated PG method (see [22, Proposition 2]). Comparing with our inexactness condition in Theorem 3.2(i), the inexactness of \( \{\epsilon_s\} \) is one order more than in the basic PG method, but it is one order weaker than the accelerated PG method. This happens unsurprisingly since although our scheme applies acceleration technique for the inner loop, it still computes full gradients for the outer loop.

### 3.3 Extension to Non-smooth Problems

Problem (1) with non-smooth \( f_i \) often arises in statistics and machine learning. One well-known example is the \( l_1 \)-SVM:

\[
\min_x \left\{ \frac{1}{n} \sum_{i=1}^{n} \max \{1 - b_i \langle a_i, x \rangle, 0\} + \lambda \| x \|_1 \right\}.
\]

(8)

If \( f_i(x) \) in (1) is non-smooth, we smooth it by \( f_{\mu,i} \) and then solve the problem:

\[
\min_{x \in X} F^P_{\mu}(x) := \frac{1}{n} \sum_{i=1}^{n} f_{\mu,i}(x) + P(x),
\]

(9)

where we assume the smooth functions \( f_{\mu,i} \) satisfy the following assumptions for \( \mu > 0 \).

**Assumption 3.1**

1. Functions \( f_{\mu,i} \) are convex and \( L_{\mu,i} \)-Lipschitz smooth.
2. There exist constants \( \overline{K}_i, \underline{K}_i \geq 0 \) such that

\[
f_{\mu,i}(x) - \underline{K}_i \mu \leq f_i(x) \leq f_{\mu,i}(x) + \bar{K}_i \mu, \forall x.
\]

Note that it is not necessary to smooth \( P(x) \), and this can be useful in practice. For example, if \( P(x) \) is the \( \ell_1 \) norm, then smoothing it undermines its ability to obtain sparse solutions. We now give some examples of popular smoothing functions that satisfy Assumption 3.1.
Example 3.2  — Let $A$ be a matrix in $\mathbb{R}^{q \times p}$, $u \in \mathbb{R}^q$, $U$ be a closed convex bounded set, and $f(x) = \max_{u \in U} \{ \langle Ax, u \rangle - \hat{\phi}(u) \}$, where $\hat{\phi}(u)$ is a continuous convex function. Let $R(u)$ be a continuous strongly convex function with constant $\sigma > 0$ and $\| A \|$ be the spectral norm of $A$. Nesterov [3] proved that the convex function $f_\mu(x) = \max_{u \in U} \{ \langle Ax, u \rangle - \hat{\phi}(u) - \mu R(u) \}$ is a $L_\mu$-Lipschitz smooth function of $f(x)$ with $L_\mu = \frac{\| A \|^2}{\mu \sigma}$. We easily see that $f(x) - K \mu \leq f_\mu(x) \leq f(x)$ for $\mu > 0$, where $K = \max_{u \in U} R(u)$. Hence, $f_\mu(x)$ satisfies Assumption 3.1.

- We know that $f_\mu^{(a)}(x) = \frac{1}{2} \left( x + \sqrt{x^2 + 4 \mu^2} \right)$ is one of the smoothing functions of $[x]_+ := \max\{ x, 0 \}$; and $f_\mu^{(a)}(x)$ is convex and $L_\mu$-Lipschitz smooth with $L_\mu = \frac{1}{4 \mu}$. Since $[x]_+ \leq f_\mu^{(a)}(x) \leq [x]_+ + \mu$, Assumption 3.1 is satisfied. Another smoothing function of $[x]_+$ is the neural network smoothing function $f_\mu^{(b)}(x) = \mu \log (1 + e^{[x]_+})$, which is widely used in SVM problems [30]. It is not difficult to prove that $[x]_+ \leq f_\mu^{(b)}(x) \leq [x]_+ + (\log 2) \mu$ and $f_\mu^{(b)}(x)$ is $L_\mu$-Lipschitz smooth with $L_\mu = \frac{1}{4 \mu}$. Hence, $f_\mu^{(b)}(x)$ satisfies Assumption 3.1. Using these smoothing functions of $[x]_+$, we can smooth the $l_1$-SVM (8) to get the form (9).

We now state the main theorem for this non-smooth case.

Theorem 3.3 Denote $\overline{K} = \frac{1}{n} \sum_{i=1}^{n} K_i$ and $K = \frac{1}{n} \sum_{i=1}^{n} K_i$. Let $\{ \tilde{x}_{\mu,s} \}$ be the sequence generated by Algorithm 3.1 for solving Problem (9), $\tilde{L}_\mu$ be the corresponding value of $\tilde{L}$ in Algorithm 3.1 for Problem (9), and $x^*$ be the solution of (1). We have

$$\mathbb{E} F_P(\tilde{x}_{\mu,s}) - F_P(x^*) = O \left( \frac{1 + \frac{4T_\mu}{m} + \tilde{C}}{(s + 3)^2} \right) + (\overline{K} + K) \mu,$$

where $\tilde{C} = 0$ if the proximal points are calculated exactly, and $\tilde{C} = O \left( \sqrt{\frac{L_\mu}{\mu}} \right)$ if the proximal points are calculated inexactly under the assumptions of Theorem 3.2.

This theorem shows that if we choose $\mu < \sqrt{\frac{\epsilon}{\overline{K} + K}}$ and suppose that $\tilde{L}_\mu = O(1/\mu)$, then after running $s$ stages of Algorithm 3.1 with

$$s > \frac{\sqrt{1 + \frac{4T_\mu}{m}}}{\sqrt{\epsilon - (\overline{K} + K) \mu}} = O \left( \frac{1}{\epsilon^{1/2}} + \frac{1}{\sqrt{m \epsilon}} \right)$$

for the case of exact proximal points, or

$$s > \frac{\sqrt{1 + \frac{4T_\mu}{m} + \sqrt{L_\mu}}}{\sqrt{\epsilon - (\overline{K} + K) \mu}} = O \left( \frac{1}{\epsilon^{3/4}} + \frac{1}{\epsilon^{1/2}} + \frac{1}{\sqrt{m \epsilon}} \right)$$
for the case of inexact proximal points, we can obtain an \(\epsilon\)-optimal solution of (1). We remark that the condition \(\bar{L}_\mu = O(1/\mu)\) is satisfied by examples in Example 3.2. Problem (1) with non-smooth \(f_i\) can also be solved by subgradient methods or stochastic subgradient methods with the convergence rate \(O(1/\epsilon^2)\) [28], which is significantly slower than the rate obtained using our smoothing scheme. In terms of complexity (the total number of gradients computed to achieve an \(\epsilon\)-optimal solution), if \(m = O(n)\), our smoothing scheme computes \(O(n/\epsilon^{1/2} + \sqrt{n}/\epsilon)\) gradients for the exact proximal points case, and \(O(n/\epsilon^{3/4} + n/\epsilon^{1/2} + \sqrt{n}/\epsilon)\) for the inexact case. These complexities definitely outperform the complexity \(O(n/\epsilon^2)\) of subgradient methods.

4 Experiments

4.1 Experiment with Synthetic Datasets

We consider the Lasso problem with \(f_i(x) = \frac{1}{2}((a_i, x) - b_i)^2\) and \(P(x) = \lambda \|x\|_1\). The problem is non-strongly convex and can also be solved by FISTA [31], SAGA [32], and APG [8]. Proximal SVRG [12] can also be applied to this problem, although we need to add a regularizer \(\xi \|x\|_2^2\) to \(P(x)\) to maintain the strong convexity and hence the convergence property of this algorithm. For simplicity, we set \(c = 0.001\) in this experiment.

We generate 9 synthetic datasets with sizes \(n = 1000, 10,000, 50,000\) and with \(p = 10, 100, 500\) dimensions as follows. First, we generate \(n\) vectors \(a_1, \ldots, a_N\) uniformly on \([0, 10]^p\) and a true sparse target vector \(x^* \in \{0, 1\}^p\) with a random half of its elements being 0 and the rest having value 1. For each \(a_i\), we generate \(b_i = (a_i, x^*) + \epsilon\), where \(\epsilon \sim \mathcal{N}(0, 0.01^2)\) is a small Gaussian noise. We set \(\lambda = 0.1\) and use the same settings of ARMD on all the datasets. We test two versions of exact ARMD based on different updating rules of \(x_k\) in Algorithm 3.1. Particularly, ARMD I and ARMD II, respectively, denote ARMD when using \(x_k, \hat{s} = \hat{x}_k, s\) and \(x_k, s = \arg\min_x \left\{ (a_k, x) + \frac{L_i}{2} \|x - y_k, s\|_2^2 + P(x) \right\}\) in (4) (see Example 3.1). For both versions, we further test on two cases:

\[
(1) \quad \alpha_{2,s} = \frac{2}{s + 2}, \quad \alpha_3 = \frac{1}{3}; \quad \text{and} \quad (2) \quad \alpha_{2,s} = \frac{2}{s + 5}, \quad \alpha_3 = \frac{2}{3}.
\]

For Lasso, the Lipschitz constant of \(f_i\) is \(L_i = a_i^2\). We use \(D(x, y) = \frac{1}{2} \|x - y\|^2\) in ARMD, and thus \(z_{k,s}\) can be computed by soft thresholding. We also set \(q_i = \frac{1}{n}\) and \(m = n\). Note that ARMD reduces the complexity compared to the other algorithms. So we evaluate the performance based on the objective function value and the optimality gap vs. \# of computed gradients. From the plots in Fig. 1, the ARMD methods are significantly better than FISTA, SAGA, and APG in all datasets. Proximal SVRG performs comparable to ARMD, although in all cases it converges to a worse optimal value than ARMD II with \(\alpha_3 = 1/3\) (Fig. 1b). This justifies that ARMD requires much less gradient computations to converge than the other algorithms. From Fig. 1b,
Fig. 1. Comparison of FISTA, AGP, and exact ARMD for Lasso problem on synthetic datasets. The vertical axis is the objective value (or optimality gap), and the horizontal axis is the number of computed gradients $\frac{1}{n}$. 

(a) objective value
Fig. 1 continued

(b) optimality gaps
Fig. 2 Results on real datasets for exact ARMD (same settings as Fig. 1).
(b) optimality gaps

Fig. 2 continued
(a) objective value

Fig. 3 Comparison of inexact versions of FISTA, APG, and ARMD for overlapping group Lasso problem. The vertical axis is the objective value (or optimality gap), and the horizontal axis is the number of computed gradients/n.
Fig. 3 continued

(b) optimality gaps
ARMD II with $\alpha_3 = 1/3$ consistently performs the best among different ARMD versions.

4.2 Experiments with Real Datasets

**Exact ARMD** We consider the same Lasso problem and settings as in Sect. 4.1 on 6 real datasets from [33]: breast cancer ($n = 683, p = 10$), abalone ($n = 4177, p = 8$), mushrooms ($n = 8124, p = 112$), cpu small ($n = 8192, p = 12$), letter ($n = 20,640, p = 8$). From Fig. 2, the ARMD methods are also significantly better than FISTA, SAGA, and APG; and ARMD II with $\alpha_3 = 1/3$ also performs the best among all algorithms on the datasets except for the cadata dataset, where ARMD II with $\alpha_3 = 2/3$ is the best method.

**Inexact ARMD** We consider the overlapping group Lasso problem [34], which is also an instance of (1) with $f_i(x) = \frac{1}{2}((a_i, x) - b_i)^2$ and $P(x) = \lambda \Omega^G_{\text{overlap}}(x)$. For a collection of overlapping groups $\mathcal{G} = \{G_r\}_{r=1}^B$, where $G_r \subseteq \{1, \ldots, p\}$ and $p$ is the number of dimensions of $x$, the penalty $\Omega^G_{\text{overlap}}(x)$ is defined as:

$$\Omega^G_{\text{overlap}}(x) = \inf_{(v_1, \ldots, v_B), v_r \in \mathbb{R}^p, \supp(v_r) \subseteq G_r, \sum_{r=1}^B v_r = x} \left\{ \sum_{r=1}^B \|v_r\|_2 \right\}.$$  

For this penalty, the proximal points cannot be computed exactly in a finite number of steps [35]; thus, we need to use inexact methods to compute the proximal points up to an error $\epsilon_s$. In this experiment, we use the same real datasets and settings above, except that the proximal points $z_{k,s}$ and $x_{k,s}$ for ARMD II are estimated using the projection method proposed in [35]. We set $\mathcal{G} = \{\{1, 2, 3\}, \{3, 4, 5\}, \{5, 6, 7\}, \ldots\}$, $\lambda = 0.1$, $\epsilon_s = 0.01/s^{4.001}$, and compare inexact ARMD methods with FISTA, APG, SAGA, and SVRG for overlapping group Lasso [35]. The value of $\epsilon_s$ is chosen to ensure the convergence of APG (which requires $\epsilon_s = O(1/s^{4+b})$, see [22]) as well as ARMD (which requires $\epsilon_s = O(1/s^{3+b})$). The penalty $\Omega^G_{\text{overlap}}(x)$ is computed at any point $x$ by solving Problem (10). Figure 3 plots the objective value and optimality gap against $\frac{\text{# of computed gradients}}{n}$. From Fig. 3a, FISTA reduces the objective value slightly faster than other algorithms during the first few iterations in many cases, but it eventually converges to worse optimal values than ARMD methods. Regarding to optimality gaps, ARMD methods achieve better optimality gaps than the other algorithms in all except for the mushrooms dataset, where SVRG performs better.

5 Conclusions

We have proposed a framework of accelerated randomized mirror descent algorithms for solving the large-scale CCOP without the strongly convex assumption of the objective function. Our framework allows proximal points to be calculated inexactly and can achieve the optimal convergence rate. Using suitable parameters, our algorithms can obtain better complexity than APG. We also proposed a scheme for solving Problem
(1) with non-smooth component functions. Computational results affirm the effectiveness of our algorithms.

Acknowledgements We are grateful to the anonymous reviewers and the Editor-in-Chief for their meticulous comments and insightful suggestions. Le Thi Khanh Hien would like to give a special thank to Prof. W. B. Haskell for his support. Le Thi Khanh Hien was supported by Grant A*STAR 1421200078.

Appendix: Proofs of Lemmas, Propositions, and Theorems

Proof of Lemma 3.1 We have

\[ \mathbb{E} \left\| \nabla F(y_k, s) - v_k \right\|_*^2 \]
\[ = \mathbb{E} \left\| 1/(nq_k) \left( \nabla f_{i_k}(y_k, s) - \nabla f_{i_k}(\tilde{x}_{k-1}) \right) - (\nabla F(y_k, s) - \nabla F(\tilde{x}_{k-1})) \right\|_*^2 \]
\[ \leq \mathbb{E} \left( \left\| 1/(nq_k) \left( \nabla f_{i_k}(y_k, s) - \nabla f_{i_k}(\tilde{x}_{k-1}) \right) \right\|_n^2 + \left\| \nabla F(y_k, s) - \nabla F(\tilde{x}_{k-1}) \right\|_*^2 \right)^2 \]
\[ \leq 2 \mathbb{E} \left\| \nabla f_{i_k}(y_k, s) - \nabla f_{i_k}(\tilde{x}_{k-1}) \right\|_*^2 + 2 \left\| \nabla F(y_k, s) - \nabla F(\tilde{x}_{k-1}) \right\|_*^2. \]

\[ \square \]

Proof of Lemma 3.2 For notation succinctness, we omit the subscript \( s \) when no confusing is caused. Applying Lemma 2.4(1), we have:

\[ F^P(x_k) = \frac{1}{n} \sum_{i=1}^{n} f_i(x_k) + P(x_k) \]
\[ \leq \frac{1}{n} \sum_{i=1}^{n} \left( f_i(y_k) + \langle \nabla f_i(y_k), x_k - y_k \rangle + \frac{L_i}{2} \| x_k - y_k \|_2^2 \right) + P(x_k) \]
\[ = F(y_k) + \langle \nabla F(y_k) - v_k, x_k - y_k \rangle \]
\[ + \frac{L_A}{2} \| x_k - y_k \|_2^2 + P(x_k) + \langle v_k, x_k - y_k \rangle \]
\[ \leq F(y_k) + \frac{2LQ}{\alpha_3} \| x_k - y_k \|_2^2 + \frac{\alpha_3}{8LQ} \| \nabla F(y_k) - v_k \|_*^2 \]
\[ + \frac{L_A}{2} \| x_k - y_k \|_2^2 + P(x_k) + \langle v_k, x_k - y_k \rangle, \]

where the last inequality uses \( \langle a, b \rangle \leq \frac{1}{2} \| a \|_2^2 + \frac{1}{2} \| b \|_2^2 \). Together with the update rule (4), Lemma 2.1 with \( \sigma = 1 \), and noting that \( \hat{x}_k - y_k = \alpha_2(z_k - z_{k-1}) \), we get:

\[ F^P(x_k) \leq F(y_k) + \frac{\alpha_3}{8LQ} \| \nabla F(y_k) - v_k \|_*^2 \]
\[ + \langle v_k, \hat{x}_k - y_k \rangle + \frac{1}{2} \left( L_A + \frac{4LQ}{\alpha_3} \right) \| \hat{x}_k - y_k \|_2^2 + P(\hat{x}_k) \]
\[ = F(y_k) + \frac{\alpha_3}{8LQ} \| \nabla F(y_k) - v_k \|_*^2 + \alpha_2 \langle v_k, z_k - z_{k-1} \rangle \]
Theorem 3.1. We have $z_{k,s} = \arg\min_{x \in X_s} \{\phi_k(x) + D(x, z_{k-1,s})\}$, where we let $\phi_k(x) = \frac{1}{\theta_s}(\langle v_k, x \rangle + P(x))$. From Lemma 2.2, for all $x \in X_s \cap \text{dom} P$, we have:

$$\frac{1}{\theta_s}(\langle v_k, x \rangle + P(x) + D(x, z_{k-1,s}) \geq \min_{x \in X_s} \{\phi_k(x) + D(x, z_{k-1,s})\} + D(x, z_{k,s}).$$

Together with $z_{k,s} \approx z_{k,s} \arg\min_{x \in X_s} \theta_s(\phi(x) + D(x, z_{k-1,s}))$, we get:

$$\langle v_k, x \rangle + P(x) + \theta_s D(x, z_{k-1,s}) \geq \langle v_k, z_{k,s} \rangle + P(z_{k,s}) + \theta_s D(z_{k,s}, z_{k-1,s}) - \varepsilon_{k,s} + \theta_s D(x, z_{k,s}). \tag{11}$$

From Lemma 2.3, we get

$$D(x, z_{k,s}) = D(x, z_{k,s}) + D(z_{k,s}, z_{k,s}) - \langle x - z_{k,s}, \nabla h(z_{k,s}) - \nabla h(z_{k,s}) \rangle.$$

Thus, the result follows. \qed

\textbf{Proof of Proposition 3.1} For notation succinctness, we omit the subscript $s$ when no confusion is caused. Applying Lemma 3.2, we have:

$$F^P(x_k) \leq F(y_k) + \frac{\alpha_3}{8LQ} \|\nabla F(y_k) - v_k\|^2 + \alpha_2 ((v_k, z_k - z_{k-1}) + \theta_s D(z_k, z_{k-1})) + P(\hat{x}_k). \tag{12}$$

From Inequality (12) and Lemma 3.3, we deduce that:

$$F^P(x_k) \leq F(y_k) + \frac{\alpha_3}{8LQ} \|\nabla F(y_k) - v_k\|^2 + \alpha_2 ((v_k, x - z_{k-1}) + P(x) - P(z_k)) + P(\hat{x}_k) \tag{13}$$

$$+ \alpha_2 \theta_s (D(x, z_{k-1}) - D(x, z_k) - D(z_k, \hat{z}_k)) + \langle x - z_k, \nabla h(\hat{z}_k) - \nabla h(z_k) \rangle + \alpha_2 \varepsilon_{k,s}.$$

Note that $\mathbb{E}_{i_k}[v_k] = \nabla F(y_k)$ (we omit the subscript $i_k$ of the conditional expectation when it is clear in the context) and $P(\hat{x}_k) \leq \alpha_1 P(x_{k-1}) + \alpha_2 P(z_k) + \alpha_3 P(\hat{x}_{s-1})$. Taking expectation with respect to $i_k$ conditioned on $i_{k-1}$, it follows from (13) that:
\[ \mathbb{E} F^P(x_k) \leq F(y_k) + \alpha_3 \left( \frac{1}{8L_Q} \mathbb{E} \| \nabla F(y_k) - v_k \|^2 + \langle \nabla F(y_k), \tilde{x}_{s-1} - y_k \rangle \right) \\
- \alpha_3 \langle \nabla F(y_k), \tilde{x}_{s-1} - y_k \rangle + \alpha_2 \langle \nabla F(y_k), x - z_{k-1} \rangle + \alpha_2 P(x) \\
+ \alpha_1 P(x_{k-1}) + \alpha_3 P(\tilde{x}_{s-1}) + \alpha_2 \theta_5(D(x, z_{k-1}) - \mathbb{E}D(x, z_k)) + r_k. \]

(14)

On the other hand, applying Lemma 3.1, the second inequality of Lemma 2.4, and noting that \( \frac{1}{L_Q n q_i} \leq \frac{1}{L_i} \) and \( \frac{1}{L_Q} \leq \frac{1}{L_A} \), we have:

\[ \frac{1}{L_Q} \mathbb{E} \| \nabla F(y_k) - v_k \|^2 + \langle \nabla F(y_k), \tilde{x}_{s-1} - y_k \rangle \]
\[ \leq \frac{1}{4L_Q} \mathbb{E} \left( \frac{1}{n q_i} \right)^2 \| \nabla f_i(y_k) - \nabla f_i(\tilde{x}_{s-1}) \|^2 + \frac{1}{4L_Q} \| \nabla F(y_k) - \nabla F(\tilde{x}_{s-1}) \|^2 \\
+ \langle \nabla F(y_k), \tilde{x}_{s-1} - y_k \rangle \]
\[ = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{4L_Q} \| \nabla f_i(y_k) - \nabla f_i(\tilde{x}_{s-1}) \|^2 + \frac{1}{2n} \sum_{i=1}^{n} \langle \nabla f_i(y_k), \tilde{x}_{s-1} - y_k \rangle \]
\[ + \frac{1}{2} \langle \nabla F(y_k), \tilde{x}_{s-1} - y_k \rangle + \frac{1}{4L_Q} \| \nabla F(y_k) - \nabla F(\tilde{x}_{s-1}) \|^2. \]
\[ \leq \frac{1}{2n} \sum_{i=1}^{n} \left( \frac{1}{2L_i} \| \nabla f_i(y_k) - \nabla f_i(\tilde{x}_{s-1}) \|^2 + \langle \nabla f_i(y_k), \tilde{x}_{s-1} - y_k \rangle \right) \]
\[ + \frac{1}{2} \left( \frac{1}{2L_A} \| \nabla F(y_k) - \nabla F(\tilde{x}_{s-1}) \|^2 + \langle \nabla F(y_k), \tilde{x}_{s-1} - y_k \rangle \right) \]
\[ \leq \frac{1}{2n} \sum_{i=1}^{n} (f_i(\tilde{x}_{s-1}) - f_i(y_k)) + \frac{1}{2} (F(\tilde{x}_{s-1}) - F(y_k)) = F(\tilde{x}_{s-1}) - F(y_k). \]

(15)

Therefore, (14) and (15) imply that:

\[ \mathbb{E} F^P(x_k) \leq \frac{1}{n} \left( 1 - \alpha_3 \right) F(y_k) + \alpha_3 F^P(\tilde{x}_{s-1}) + \alpha_2 \langle \nabla F(y_k), x - y_k \rangle + \alpha_2 P(x) \\
+ \alpha_2 \langle \nabla F(y_k), y_k - z_{k-1} \rangle - \alpha_3 \langle \nabla F(y_k), \tilde{x}_{s-1} - y_k \rangle + \alpha_1 P(x_{k-1}) \\
+ \alpha_2 \theta_5(D(x, z_{k-1}) - \mathbb{E}D(x, z_k)) + r_k \]
\[ \leq \left( 1 - \alpha_3 \right) F(y_k) + \alpha_3 F^P(\tilde{x}_{s-1}) + \alpha_2 (F(x) - F(y_k)) + \alpha_2 P(x) \\
+ \alpha_1 \langle \nabla F(y_k), x_{k-1} - y_k \rangle + \alpha_1 P(x_{k-1}) + \alpha_2 \theta_5(D(x, z_{k-1}) - \mathbb{E}D(x, z_k)) + r_k \]
\[ \leq \left( 1 - \alpha_3 \right) F(y_k) + \alpha_3 F^P(\tilde{x}_{s-1}) + \alpha_2 F^P(x) \\
+ \alpha_1 (F(x_{k-1}) - F(y_k)) + \alpha_1 P(x_{k-1}) + \alpha_2 \theta_5(D(x, z_{k-1}) - \mathbb{E}D(x, z_k)) + r_k \]

\( \diamond \) Springer
\[
= \alpha_1 F^P(x_{k-1}) + \alpha_2 F^P(x) + \alpha_3 F^P(\tilde{x}_{s-1}) + \alpha_2 \delta_s(D(x, z_{k-1})
- \mathbb{E} D(x, z_k)) + r_k.
\]

Here, in (a) we use
\[
\langle \nabla F(y_k), x - y_k \rangle \leq F(x) - F(y_k) \text{ and } \alpha_2(y_k - z_{k-1}) - \alpha_3(\tilde{x}_{s-1} - y_k)
= \alpha_1(x_{k-1} - y_k).
\]

in (b) we use \(\langle \nabla F(y_k), x_{k-1} - y_k \rangle \leq F(x_{k-1}) - F(y_k)\). Finally, we take expectation with respect to \(i_{k-1}\) to get the result.

\[\square\]

**Proof of Proposition 3.2** Applying Proposition 3.1 with \(x = x^*\), we have:
\[
\mathbb{E} (F^P(x_{k,s}) - F^P(x^*)) \leq \alpha_{1,s} \mathbb{E} (F^P(x_{k-1,s}) - F^P(x^*)) + \alpha_3(F^P(\tilde{x}_{s-1}) - F^P(x^*))
+ \alpha_2^2 \mathbb{L} (\mathbb{E} D(x^*, z_{k-1,s}) - \mathbb{E} D(x^*, z_{k,s})) + r^*_k.
\]

Denote \(d_{k,s} = \mathbb{E} (F^P(x_{k,s}) - F^P(x^*))\), then
\[
d_{k,s} \leq \alpha_{1,s} d_{k-1,s} + \alpha_3 \tilde{d}_{s-1} + \alpha_2^2 \mathbb{L} (\mathbb{E} D(x^*, z_{k-1,s}) - \mathbb{E} D(x^*, z_{k,s})) + r^*_k,
\]

which implies \(\frac{1}{\alpha_2^2} d_{k,s} \leq \frac{\alpha_{1,s}}{\alpha_2^2} d_{k-1,s} + \frac{\alpha_3}{\alpha_2^2} \tilde{d}_{s-1} + \mathbb{L} (\mathbb{E} D(x^*, z_{k-1,s}) - \mathbb{E} D(x^*, z_{k,s})) + \frac{r^*_k}{\alpha_2^2}\). Summing up this inequality from \(k = 1\) to \(k = m\), we get:
\[
\frac{1}{\alpha_2^2} d_{m,s} + \frac{1 - \alpha_{1,s}}{\alpha_2^2} \sum_{k=1}^{m-1} d_{k,s} \leq \frac{\alpha_{1,s}}{\alpha_2^2} d_{0,s} + \frac{\alpha_3}{\alpha_2^2} m \tilde{d}_{s-1}
+ \mathbb{L} (\mathbb{E} D(x^*, z_{0,s}) - \mathbb{E} D(x^*, z_{m,s}))
+ \sum_{k=1}^{m} \frac{r^*_k}{\alpha_2^2}.
\]

Using the update rule (5), \(\alpha_{1,s} + \alpha_3 = 1 - \alpha_2^2\), \(z_{s,m-1} = z_{0,s}\), and \(d_{m,s-1} = d_{0,s}\), we get:
\[
\frac{1}{\alpha_2^2} d_{m,s} + \frac{1 - \alpha_{1,s}}{\alpha_2^2} \sum_{k=1}^{m-1} d_{k,s} \leq \frac{1 - \alpha_2^2}{\alpha_2^2} d_{m,s-1} + \frac{\alpha_3}{\alpha_2^2} \sum_{k=1}^{m-1} d_{k,s-1}
+ \mathbb{L} (\mathbb{E} D(x^*, z_{m-1,s}) - \mathbb{E} D(x^*, z_{m,s})) + \sum_{k=1}^{m} \frac{r^*_k}{\alpha_2^2}.
\]
Combining with the update rule (2), we obtain:

\[
\frac{1 - \alpha_{2,s+1}}{\alpha_2^2} d_{m,s} + \frac{\alpha_3}{\alpha_2^2} \sum_{k=1}^{m-1} d_{k,s} \leq \frac{1 - \alpha_{2,s}}{\alpha_2^2} d_{m,s-1} + \frac{\alpha_3}{\alpha_2^2} \sum_{k=1}^{m-1} d_{k,s-1} + \frac{1 - \alpha_{2,s+1}}{\alpha_2^2} d_{m,s} + \frac{\alpha_3}{\alpha_2^2} \sum_{k=1}^{m-1} d_{k,s} + \bar{L} \left( \mathbb{E}D(x^*, z_{m,s-1}) - \mathbb{E}D(x^*, z_{m,s}) \right) + \sum_{k=1}^{m} r_{k,s}^{*},
\]

(16)

Therefore,

\[
\frac{\alpha_3}{\alpha_2^2} m \bar{d}_s \leq \frac{\alpha_3}{\alpha_2^2} \sum_{k=1}^{m} d_{k,s} \leq \frac{1 - \alpha_{2,s+1}}{\alpha_2^2} d_{m,0} + \frac{\alpha_3}{\alpha_2^2} \sum_{k=1}^{m-1} d_{k,0} + \bar{L} \left( \mathbb{E}D(x^*, z_{m,0}) - \mathbb{E}D(x^*, z_{m,s}) \right) + \sum_{k=1}^{m} \sum_{i=1}^{s} r_{k,i}^{*},
\]

where in (a) we use the update rule (5), in (b) we use the property \( \alpha_3 \leq 1 - \alpha_{2,s+1} \), and in (c) we use the recursive inequality (16). The result then follows. \( \square \)

**Proof of Theorem 3.1** Without loss of generality, we can assume that:

\[
\frac{1}{m} \left( \frac{4(1 - \alpha_{2,1})}{\alpha_{2,1}^2 \alpha_3} d_{m,0} + \frac{4}{\alpha_{2,1}^2} \sum_{i=1}^{m} d_{i,0} \right) = O(F^P(\bar{x}_0) - F^P(x^*)).
\]

When \( \varepsilon_{k,s} = 0 \), then \( z_{k,s} = \bar{z}_{k,s} \) and we have \( r_{k,s} = 0 \). The convergence rate of exact ASMD follows from Proposition 3.2 by taking \( \alpha_{2,s} = \frac{2}{s+2} \) and noting that \( D(x^*, z_{m,s}) \geq 0 \). \( \square \)

**Proof of Theorem 3.2** We remind that Inequality (11) holds for all \( x \). Taking \( x = z_{k,s} \), (11) yields that \( D(z_{k,s}, \bar{z}_{k,s}) \leq \frac{\varepsilon_{k,s}}{\theta_s} \). On the other hand, if \( h(\cdot) \) is \( L_h \)-Lipschitz smooth, then:

\[
\| \nabla h(\bar{z}_{k,s}) - \nabla h(z_{k,s}) \| \leq L_h \| \bar{z}_{k,s} - z_{k,s} \| \leq L_h \sqrt{2D(z_{k,s}, \bar{z}_{k,s})} \leq L_h \sqrt{\frac{2\varepsilon_{k,s}}{\theta_s}}.
\]

If \( \| z_{k,s} \| \leq C \), then we let \( C_1 = \| x^* \| + C \). Noting that \( D(z_{k,s}, \bar{z}_{k,s}) \geq 0 \), we have:

\[
r_{k,s}^{*} \leq \alpha_{2,s} \theta_s \| x^* - z_{k,s} \| \| \nabla h(\bar{z}_{k,s}) - \nabla h(z_{k,s}) \| + \alpha_{2,s} \varepsilon_{k,s} \leq \alpha_{2,s} C_1 L_h \sqrt{\frac{2\varepsilon_{k,s}}{\theta_s}} + \alpha_{2,s} \varepsilon_{s}.
\]
Hence,
\[
\alpha_{2,s+1}^2 \sum_{i=1}^{s} \sum_{k=1}^{m} \frac{r^{*,i}_{k,i}}{m \alpha_3 \alpha_{2,i}^2} \leq \alpha_{2,s+1}^2 \sum_{i=1}^{s} \left( \frac{C_1 L_h \sqrt{2 \epsilon_i \bar{L}}}{\alpha_3 \sqrt{\alpha_{2,i}}} + \frac{\epsilon_i}{\alpha_3 \alpha_{2,i}} \right). \tag{17}
\]

If the adaptive inexact rule max \( \left\{ \| \tilde{z}_{k,s} \|^2 \epsilon_{k,s}, C \epsilon_{k,s} \right\} \leq C \epsilon_s \) is chosen, we have
\[
r^{*,i}_{k,i} \leq \alpha_{2,s} \theta_s \left( \| x^\ast \| + \| \tilde{z}_{k,s} \| + \| \tilde{z}_{k,s} - z_{k,s} \| \right) \| \nabla h(\tilde{z}_{k,s}) - \nabla h(z_{k,s}) \| + \alpha_{2,s} \epsilon_{k,s}
\leq \alpha_{2,s} \| x^\ast \| \sqrt{2 \epsilon_s} \theta_s + \alpha_{2,s} L_h \sqrt{2 C \epsilon_s} \theta_s + \alpha_{2,s} L_h \epsilon_s + \alpha_{2,s} \epsilon_s.
\]

In this case, we let \( C_1 = \| x^\ast \| + \sqrt{C} \). We then have
\[
\alpha_{2,s+1}^2 \sum_{i=1}^{s} \sum_{k=1}^{m} \frac{r^{*,i}_{k,i}}{m \alpha_3 \alpha_{2,i}^2} \leq \alpha_{2,s+1}^2 \sum_{i=1}^{s} \left( \frac{C_1 L_h \sqrt{2 \epsilon_i \bar{L}}}{\alpha_3 \sqrt{\alpha_{2,i}}} + \frac{(2L_h + 1)\epsilon_i}{\alpha_3 \alpha_{2,i}} \right). \tag{18}
\]

The result then follows from (17), (18), and Proposition 3.2 easily. □

**Proof of Theorem 3.3** Let \( x^\ast_\mu \) is the optimal solution of Problem (9). We have:
\[
\mathbb{E} F^P_{\mu} (\tilde{x}_{\mu,s}) - F^P_{\mu} (x^\ast_\mu) = O \left( \frac{1 + \frac{4T_\mu}{m} + \bar{C}}{(s + 3)^2} \right), \tag{19}
\]
where \( \bar{C} = O \left( \sqrt{L_\mu} \right) \), by applying Theorem 3.2. By Assumption 3.1, we have:
\[
\mathbb{E} F^P (\tilde{x}_{\mu,s}) - F^P (x^\ast) = \mathbb{E} F (\tilde{x}_{\mu,s}) + \mathbb{E} P (\tilde{x}_{\mu,s}) - F (x^\ast) - P (x^\ast)
\leq \mathbb{E} F_{\mu} (\tilde{x}_{\mu,s}) + \bar{K}_\mu + \mathbb{E} P (\tilde{x}_{\mu,s}) - F (x^\ast) - P (x^\ast)
\leq \mathbb{E} F_{\mu} (\tilde{x}_{\mu,s}) + \bar{K}_\mu + \mathbb{E} P (\tilde{x}_{\mu,s}) - F_{\mu} (x^\ast) + \bar{K}_\mu - P (x^\ast)
\leq \mathbb{E} F^P_{\mu} (\tilde{x}_{\mu,s}) - F^P_{\mu} (x^\ast) + (\bar{K} + \bar{K}) \mu.
\]
Together with (19) and noting that \( F^P_{\mu} (x^\ast) \geq F^P_{\mu} (x^\ast) \), we get:
\[
\mathbb{E} F^P (\tilde{x}_{\mu,s}) - F^P (x^\ast) \leq \mathbb{E} F^P_{\mu} (\tilde{x}_{\mu,s}) - F^P_{\mu} (x^\ast) + (\bar{K} + \bar{K})
\mu = O \left( \frac{1 + \frac{4T_\mu}{m} + \bar{C}}{(s + 3)^2} \right) + (\bar{K} + \bar{K}) \mu.
\] □
References

1. Nesterov, Y.: A method of solving a convex programming problem with convergence rate $O(1/k^2)$. Sov. Math. Dokl. 27(2), 543–547 (1983)
2. Nesterov, Y.: On an approach to the construction of optimal methods of minimization of smooth convex functions. Ekonom. i. Mat. Metody 24, 509–517 (1998)
3. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. 103(1), 127–152 (2005)
4. Nesterov, Y.: Gradient methods for minimizing composite functions. Math. Program. 140(1), 125–161 (2013)
5. Becker, S., Bobin, J., Candès, E.J.: NESTA: a fast and accurate first-order method for sparse recovery. SIAM J. Imaging Sci. 4(1), 1–39 (2011)
6. d’Aspremont, A., Banerjee, O., Ghaoui, L.E.: First-order methods for sparse covariance selection. SIAM J. Matrix Anal. Appl. 30(1), 56–66 (2008)
7. Auslender, A., Teboulle, M.: Interior gradient and proximal methods for convex and conic optimization. SIAM J. Optim. 16(3), 697–725 (2006)
8. Tseng, P.: On Accelerated Proximal Gradient Methods for Convex–Concave Optimization. Technical report (2008)
9. Nemirovski, A., Juditsky, A., Lan, G., Shapiro, A.: Robust stochastic approximation approach to stochastic programming. SIAM J. Optim. 19(4), 1574–1609 (2009)
10. Roux, N.L., Schmidt, M., Bach, F.R.: A stochastic gradient method with an exponential convergence rate for finite training sets. In: Advances in Neural Information Processing Systems, pp. 2663–2671 (2012)
11. Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. In: Advances in Neural Information Processing Systems, pp. 315–323 (2013)
12. Xiao, L., Zhang, T.: A proximal stochastic gradient method with progressive variance reduction. SIAM J. Optim. 24, 2057–2075 (2014)
13. Nguyen, L.M., Liu, J., Scheinberg, K., Takáč, M.: SARAH: a novel method for machine learning problems using stochastic recursive gradient. In: International Conference on Machine Learning, pp. 2613–2621 (2017)
14. Fercoq, O., Richtárik, P.: Accelerated, parallel, and proximal coordinate descent. SIAM J. Optim. 25(4), 1997–2023 (2015)
15. Lin, H., Mairal, J., Harchaoui, Z.: A universal catalyst for first-order optimization. In: Advances in Neural Information Processing Systems, pp. 3384–3392 (2015)
16. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends Optim. 1(3), 127–239 (2014)
17. Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. 20(4), 1956–1982 (2010)
18. Fadili, J.M., Peyre, G.: Total variation projection with first order schemes. IEEE Trans. Image Process. 20(3), 657–669 (2011)
19. Ma, S., Goldfarb, D., Chen, L.: Fixed point and Bregman iterative methods for matrix rank minimization. Math. Program. 128(1), 321–353 (2011)
20. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM J. Control Optim. 14(5), 877–898 (1976)
21. Devolder, O., Glineur, F., Nesterov, Y.: First-order methods of smooth convex optimization with inexact oracle. Math. Program. 146(1), 37–75 (2014)
22. Schmidt, M., Roux, N.L., Bach, F.R.: Convergence rates of inexact proximal-gradient methods for convex optimization. In: Advances in Neural Information Processing Systems, pp. 1458–1466 (2011)
23. Solodov, M., Svaiter, B.: Error bounds for proximal point subproblems and associated inexact proximal point algorithms. Math. Program. 88(2), 371–389 (2000)
24. Villa, S., Salzo, S., Baldassarre, L., Verri, A.: Accelerated and inexact forward--backward algorithms. SIAM J. Optim. 23(3), 1607–1633 (2013)
25. Allen-Zhu, Z.K.: The first direct acceleration of stochastic gradient methods. In: ACM SIGACT Symposium on Theory of Computing (2017)
26. Bregman, L.: The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. USSR Comput. Math. Math. Phys. 7(3), 200–217 (1967)
27. Teboulle, M.: Convergence of proximal-like algorithms. SIAM J. Optim. 7(4), 1069–1083 (1997)
28. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer, Dordrecht (2004)
29. Auslender, A.: Numerical Methods for Nondifferentiable Convex Optimization, pp. 102–126. Springer, Berlin (1987)
30. Lee, Y.J., Mangasarian, O.: SSVM: a smooth support vector machine for classification. Comput. Optim. Appl. 20(1), 5–22 (2001)
31. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. 2(1), 183–202 (2009)
32. Defazio, A., Bach, F., Lacoste-julien, S.: SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. In: Advances in Neural Information Processing Systems, pp. 1646–1654 (2014)
33. Fan, R.E., Lin, C.J.: LIBSVM Data: Classification, Regression and Multi-Label. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets (2011). Accessed 01 April 2018
34. Jacob, L., Obozinski, G., Vert, J.P.: Group Lasso with overlap and graph Lasso. In: International Conference on Machine Learning, pp. 433–440 (2009)
35. Mosci, S., Villa, S., Verri, A., Rosasco, L.: A primal–dual algorithm for group sparse regularization with overlapping groups. In: Advances in Neural Information Processing Systems, pp. 2604–2612 (2010)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Le Thi Khanh Hien1 · Cuong V. Nguyen2 · Huan Xu3 · Canyi Lu4 · Jiashi Feng1

Cuong V. Nguyen
vcn22@cam.ac.uk

Huan Xu
huan.xu@isye.gatech.edu

Canyi Lu
canyil@andrew.cmu.edu

Jiashi Feng
elefjia@nus.edu.sg

1 National University of Singapore, Singapore, Singapore
2 University of Cambridge, Cambridge, UK
3 Georgia Institute of Technology, Atlanta, USA
4 Carnegie Mellon University, Pittsburgh, USA