SIAM J. OPTIM.
Vol. 28, No. 2, pp. 1282–1300

GLOBAL CONVERGENCE RATE OF PROXIMAL INCREMENTAL AGGREGATED GRADIENT METHODS*

N. D. VANLI†, M. GÜRBÜZBALABAN†, AND A. OZDAGLAR†

Abstract. We focus on the problem of minimizing the sum of smooth component functions (where the sum is strongly convex) and a nonsmooth convex function, which arises in regularized empirical risk minimization in machine learning and distributed constrained optimization in wireless sensor networks and smart grids. We consider solving this problem using the proximal incremental aggregated gradient (PIAG) method, which at each iteration moves along an aggregated gradient (formed by incrementally updating gradients of component functions according to a deterministic order) and takes a proximal step with respect to the nonsmooth function. While the convergence properties of this method with randomized orders (in updating gradients of component functions) have been investigated, this paper, to the best of our knowledge, is the first study that establishes the convergence rate properties of the PIAG method for any deterministic order. In particular, we show that the PIAG algorithm is globally convergent with a linear rate provided that the step size is sufficiently small. We explicitly identify the rate of convergence and the corresponding step size to achieve this convergence rate. Our results improve upon the best known condition number and gradient delay bound dependence of the convergence rate of the incremental aggregated gradient methods used for minimizing a sum of smooth functions.

Key words. convex optimization, nonsmooth optimization, proximal incremental aggregated gradient method

AMS subject classifications. 90C30, 90C06, 90C25

DOI. 10.1137/16M1094415

1. Introduction.

We focus on *composite additive cost optimization problems*, where the objective function is given by the sum of m component functions $f_i(x)$ and a possibly nonsmooth regularization function $r(x)$:

$$\min_{x\in\mathbb{R}^n} F(x) \triangleq f(x) + r(x), \tag{1.1}$$

and $f(x) = \frac{1}{m}\sum_{i=1}^m f_i(x)$. We assume each component function $f_i : \mathbb{R}^n \to (-\infty, \infty)$ is continuously differentiable and the sum of the component functions f is strongly convex, while the regularization function $r : \mathbb{R}^n \to (-\infty, \infty]$ is proper, closed, and convex but not necessarily differentiable. This formulation arises in many problems in machine learning [19,37,42], distributed optimization [13,24,25,32], and signal processing [8,11,14]. Notable examples include constrained and regularized least squares problems that arise in various machine learning applications [9, 30, 41], distributed optimization problems that arise in wireless sensor network [28, 35] as well as smart grid applications [15, 16] and constrained optimization of separable problems [2, 31]. An important feature of this formulation is that the number of component functions m is large, hence solving this problem using a standard gradient method that involves evaluating the full gradient of $f(x)$, i.e., $\nabla f(x) = \frac{1}{m}\sum_{i=1}^m \nabla f_i(x)$, is costly. This motivates using *incremental methods* that exploit the additive structure of the problem and update the decision vector using one component function at a time.

*Received by the editors September 16, 2016; accepted for publication (in revised form) January 4, 2018; published electronically May 8, 2018. A short version of this work was presented at the 55th IEEE Conference on Decision and Control (CDC) December 2016.
http://www.siam.org/journals/siopt/28-2/M109441.html
†Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 (denizcan@mit.edu, mertg@mit.edu, asuman@mit.edu).

When r is continuously differentiable, one widely studied approach is the incremental gradient (IG) method [2, 27, 36]. The IG method processes the component functions one at a time by taking steps along the gradient of each individual function in a sequential manner, following a cyclic order [39, 40] or a randomized order [18, 33, 40]. A particular randomized order, which at each iteration independently picks a component function uniformly at random from all component functions, leads to the popular stochastic gradient descent (SGD) method. While SGD is the method of choice in practice for many machine learning applications due to its superior empirical performance and convergence rate estimates that do not depend on the number of component functions m, its convergence rate is sublinear, i.e., an ϵ-optimal solution can be computed within $O(1/\epsilon)$ iterations, where a vector $x \in \mathbb{R}^n$ is an ϵ-optimal solution if $F(x) - F(x^*) \leq \epsilon$ and x^* is the minimizer of F. In a seminal paper, Blatt et al. [6] proposed the *incremental aggregated gradient (IAG) method*, which maintains the savings associated with incrementally accessing the component functions, but keeps the most recent component gradients in memory to approximate the full gradient $\nabla f(x)$ and updates the iterate using this aggregated gradient. Blatt et al. showed that under some assumptions, for a sufficiently small constant step size, the IAG method is globally convergent, and when the component functions are quadratics it achieves a linear rate. Two recent papers, [34] and [17], investigated the convergence rate of this method for general component functions that are convex and smooth (i.e., with Lipschitz gradients), where the sum of the component functions is strongly convex. In [34], the authors focused on a randomized version, called the stochastic average gradient (SAG) method (which samples the component functions independently, similarly to SGD), and showed that it achieves a linear rate using a proof that relies on the stochastic nature of the algorithm. In a more recent work [17], the authors focused on deterministic IAG (i.e., component functions processed using an arbitrary deterministic order) and provided a simple analysis that uses a delayed dynamical system approach to study the evolution of the iterates generated by this algorithm.

While these recent advances suggest IAG as a promising approach with fast convergence rate guarantees for solving additive cost problems, in many applications listed above, the objective function takes a composite form and includes a nonsmooth regularization function $r(x)$ (to avoid overfitting or to induce a sparse representation). Another important case of interest is smooth constrained optimization problems which can be represented in the composite form (1.1), where the function $r(x)$ is the indicator function of a nonempty closed convex set.

In this paper, we study the *proximal incremental aggregated gradient (PIAG) method* for solving composite additive cost optimization problems. Our method computes an aggregated gradient for the function $f(x)$ (with component gradients evaluated in a *deterministic manner* at outdated iterates over a finite window K, similar to IAG) and uses a proximal operator with respect to the regularization function $r(x)$ at the intermediate iterate obtained by moving along the aggregated gradient. Under the assumptions that $f(x)$ is strongly convex and each $f_i(x)$ is smooth with Lipschitz gradients, we show the first *linear convergence rate* result for the deterministic PIAG and provide explicit convergence rate estimates that highlight the dependence on the condition number of the problem (which we denote by Q) and the gradient delay bound K over which outdated component gradients are evaluated. In particular, we show that in order to achieve an ϵ-optimal solution, the PIAG algorithm requires

$\mathcal{O}(QK\log(1/\epsilon))$ iterations.[1] This result improves upon the condition number and gradient delay bound dependence of the deterministic IAG for smooth problems; see [17], where the authors proved that to achieve an ϵ-optimal solution, the IAG algorithm requires $\mathcal{O}(Q^2K^2\log(1/\epsilon))$ iterations. We also note that three recent independent papers [12, 21, 26] have analyzed the convergence rate of the prox-gradient algorithm (which is a special case of our algorithm with $K = 0$, i.e., where we have access to a full gradient at each iteration instead of an aggregated gradient) under strong convexity-type assumptions and provided linear rate estimates. As we highlight in Remark 3.6, our analysis can be extended to nonstrongly convex settings using similar assumptions as in [12, 21, 26] (such as quadratic functional growth or an error bound condition). Our rate estimates for the PIAG algorithm with $K > 0$ matches the condition number dependence of the prox-gradient algorithm provided in [12, 21, 26]. Furthermore, for the case in which $K = 0$ (i.e., for the prox-gradient algorithm), the rate estimates obtained using our analysis technique has the same condition number dependence as the ones presented in [12, 21, 26].

Our analysis uses function values to track the evolution of the iterates generated by the PIAG algorithm. This is in contrast with the recent analysis of the IAG algorithm provided in [17], which used distances of the iterates to the optimal solution as a *Lyapunov function* and relied on the smoothness of the problem to bound the gradient errors with distances. This approach does not extend to the nonsmooth composite case, which motivates a new analysis using function values and the properties of the proximal operator. Since we work directly with function values, this approach also allows us to obtain iteration complexity results to achieve an ϵ-optimal solution.

In terms of the algorithmic structure, our paper is related to [9], where the authors introduced the SAGA method, which extends the SAG method to the composite case and provides a linear convergence rate result with an analysis that relies on the stochastic nature of the algorithm and does not extend to the deterministic case. In particular, the SAGA method samples the component functions randomly and independently at each iteration without replacement (in contrast with the PIAG method, where the component functions are processed deterministically). However, such random sampling may not be possible for applications such as decentralized information processing in wireless sensor networks (where agents are subject to communication constraints imposed by the network topology and all agents are not necessarily connected to every other agent via a low-cost link [28]), motivating the study of the deterministic PIAG method. In addition to being a natural order for such applications, deterministic orders are also important to consider since their analysis for incremental methods could be the key for understanding the convergence behavior of general component function sampling schemes [18]. In [9], the authors prove that to achieve a point in the ϵ-neighborhood of the optimal solution, SAGA requires $\mathcal{O}(\max(Q, K)\log(1/\epsilon))$ iterations, where a vector $x \in \mathbb{R}^n$ is in the ϵ-neighborhood of an optimal solution x^* if $||x - x^*|| \leq \epsilon$. However, note that this result does not translate into a guarantee in the function suboptimality of the resulting point because of lack of smoothness. Furthermore, the choice of Lyapunov function in [9] requires each $f_i(x)$ to be convex (to satisfy the nonnegativity condition), whereas we do not need this assumption in our analysis.

A recent and independent work [1] also studied the convergence rate of the PIAG algorithm. Similar to [9, 17], the authors used the distance to the optimal solution

[1] The actual dependence on K is $\mathcal{O}(QK\log(1/\epsilon))$, and hence, for the $K = 0$ case, iteration complexity reduces to $\mathcal{O}(Q\log(1/\epsilon))$.

as a Lyapunov function and proved linear convergence results. In particular, the authors proved that to achieve a point in the ϵ-neighborhood of the optimal solution, PIAG requires $\mathcal{O}(QK^2 \log(1/\epsilon))$ iterations. Similar to [9], this convergence rate on distances does not translate into a linear convergence rate in function suboptimality as the problem (1.1) is not smooth. Furthermore, the dependence of the convergence rate on the gradient delay bound is quadratic in [1], whereas we prove a linear dependence in this paper.

Our work is also related to [39], where the authors proposed a related linearly convergent incrementally updated gradient method for solving the composite additive cost problem in (1.1) under a local Lipschitzian error condition (a condition satisfied by locally strongly convex functions around an optimal solution). The PIAG algorithm is different from the algorithm proposed in [39]. Specifically, for constrained optimization problems (i.e., when the regularization function is the indicator function of a nonempty closed convex set), the iterates generated by the algorithm in [39] stay in the interior of the set since the algorithm in [39] searches for a feasible update direction. On the other hand, the PIAG algorithm uses the proximal map on the intermediate iterate obtained by moving in the opposite direction of the aggregated gradient, which operates as a projected gradient method and allows the iterates to be on the boundary of the set. Aside from algorithmic differences, [39] does not provide explicit rate estimates (even though the exact rate can be calculated after an elaborate analysis, the dependence on the condition number and the window length of the outdated gradients is significantly worse than the one presented in this paper). Furthermore, the results in [39] provide a K-step linear convergence, whereas the linear convergence results in our paper hold uniformly for each step.

Other than the papers mentioned above, our paper is also related to [5], which studies an alternative incremental aggregated proximal method and shows linear convergence when each $f_i(x)$ and $r(x)$ are continuously differentiable. This method forms a linear approximation to $f(x)$ and processes the component functions $f_i(x)$ with a proximal iteration, whereas our method processes $f_i(x)$ based on a gradient step. Furthermore, our linear convergence results do not require the differentiability of the objective function $r(x)$ in contrast to the analysis in [5].

Several recent papers in the machine learning literature (e.g., [9, 10, 20, 22, 23] and references therein) are also weakly related to our paper. In all these papers, the authors proposed randomized order algorithms similar to the SAG algorithm [34] and analyzed their convergence rates in expectation. In particular, in [10], the authors proposed an algorithm, called Finito, which is closely related to the SAG algorithm but achieves a faster convergence rate than the SAG algorithm. These ideas were then extended to composite optimization problems with nonsmooth objective functions (as in (1.1)) in [9,23]. In particular, in [23], a majorization-minimization algorithm, called MISO, was proposed to solve smooth optimization problems and its global linear convergence was shown in expectation. In [22], the ideas in [23] were then extended for nonsmooth optimization problems using proximal operators. Similarly, in [20], a variance reduction technique was applied to the SGD algorithm for smooth problems and its global linear convergence in expectation was proven.

The rest of the paper is organized as follows. In section 2, we introduce the PIAG algorithm. In section 3, we first provide the assumptions on the objective functions and then prove the global linear convergence of the proposed algorithm under these assumptions. We conclude the paper in section 5 with a summary of our results.

2. The PIAG algorithm. Similar to the IAG method, at each iteration k, we form an aggregated gradient, which we define as

$$g_k \triangleq \frac{1}{m}\sum_{i=1}^{m} \nabla f_i(x_{\tau_{i,k}}),$$

where $\nabla f_i(x_{\tau_{i,k}})$ represents the gradient of the ith component function sampled at time $\tau_{i,k}$. We assume that each component function is sampled at least once in the past $K \geq 0$ iterations, i.e., we have

$$k - K \leq \tau_{i,k} \leq k \quad \forall i \in \{1, \ldots, m\}.$$

This condition is typically satisfied in practical implementations of the deterministic incremental methods. For instance, if the functions are processed in a cyclic order, we have $K = m - 1$ [18, 39]. On the other hand, $K = 0$ corresponds to the case in which we have the full gradient of the function $f(x)$ at each iteration (i.e., $g_k = \nabla f(x_k)$) and small K may represent a setting in which the gradients of the component functions are sent to a processor with some delay upper bounded by K.

Since the regularization function r is not necessarily differentiable, we propose to solve (1.1) with the proximal incremental aggregated gradient (PIAG) method, which uses the proximal operator with respect to the regularization function at the intermediate iterate obtained using the aggregated gradient. In particular, the PIAG algorithm, at each iteration $k \geq 0$, updates x_k as

$$x_{k+1} = \text{prox}_r^{\eta}(x_k - \eta g_k), \tag{2.1}$$

where η is a constant step size and the proximal mapping is defined as follows:

$$\text{prox}_r^{\eta}(y) = \arg\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2}\left|\left|x - y\right|\right|^2 + \eta r(x) \right\}. \tag{2.2}$$

Here, we define $\phi(x) \triangleq \frac{1}{2}\left|\left|x - y\right|\right|^2 + \eta r(x)$ and let $\partial\phi(x)$ denote the set of subgradients of the function ϕ at x. Then, it follows from the optimality conditions [4] of the problem in (2.2) that $0 \in \partial\phi(x_{k+1})$. This yields $x_{k+1} - (x_k - \eta g_k) + \eta h_{k+1} = 0$ for some $h_{k+1} \in \partial r(x_{k+1})$. Hence, we can compactly represent our update rule as

$$x_{k+1} = x_k + \eta d_k, \tag{2.3}$$

where $d_k \triangleq -g_k - h_{k+1}$ is the direction of the update at time k.

3. Convergence analysis.

3.1. Assumptions. Throughout the paper, we make the following standard assumptions.

Assumption 3.1 (Lipschitz gradients). Each f_i has Lipschitz continuous gradients on $\mathbb{R}^n$ with some constant $L_i \geq 0$, i.e.,

$$\left|\left|\nabla f_i(x) - \nabla f_i(y)\right|\right| \leq L_i \left|\left|x - y\right|\right|$$

for any $x, y \in \mathbb{R}^n$.[2]

[2]If a function f has Lipschitz continuous gradients with some constant L, then f is called L-smooth. We use these terms interchangeably.

Defining $L \triangleq \frac{1}{m}\sum_{i=1}^{m} L_i$, we observe that Assumption 3.1 and the triangle inequality yield

$$||\nabla f(x) - \nabla f(y)|| \leq L\,||x - y||$$

for any $x, y \in \mathbb{R}^n$, i.e., the function f is L-smooth.

Assumption 3.2 (subdifferentiability). The regularization function $r : \mathbb{R}^n \to (-\infty, \infty]$ is proper, closed, convex, and subdifferentiable everywhere in its effective domain, i.e., $\partial r(x) \neq \emptyset$ for all $x \in \{y \in \mathbb{R}^n : r(y) < \infty\}$.

Assumption 3.3 (strong convexity). The sum function f is μ-strongly convex on $\mathbb{R}^n$ for some $\mu > 0$, i.e., the function $x \mapsto f(x) - \frac{\mu}{2}\,||x||^2$ is convex.

A consequence of Assumptions 3.2 and 3.3 is that F is strongly convex, and hence there exists a unique optimal solution of problem (1.1) [29, Lemma 6], which we denote by x^*.

We emphasize that these assumptions hold for a variety of cost functions including regularized squared error loss, hinge loss, and logistic loss [7] and similar assumptions are widely used to analyze the convergence properties of incremental gradient methods in the literature [3, 5, 9, 17, 34]. Note that in contrast with many of these analyses, we do not assume that the component functions f_i are convex.

3.2. Rate of convergence. In this section, we show that the PIAG algorithm attains a global linear convergence rate with a constant step size provided that the step size is sufficiently small. We define

$$F_k \triangleq F(x_k) - F(x^*), \tag{3.1}$$

which is the suboptimality in the objective value at iteration k. In our analysis, we will use F_k as a Lyapunov function to prove global linear convergence. Before providing the main theorems of the paper, we first introduce three lemmas that contain key relations in proving these theorems.

The first lemma investigates how the suboptimality in the objective value evolves over the iterations. In particular, it shows that the change in suboptimality $F_{k+1} - F_k$ can be bounded as a sum of two terms: The first term is negative and has a linear dependence in the step size η, whereas the second term is positive and has a quadratic dependence in η. This suggests that if the step size η is small enough, the linear term in η will be dominant guaranteeing a descent in suboptimality.

LEMMA 3.4. *Suppose that Assumptions* 3.1 *and* 3.2 *hold. Then, the PIAG algorithm in* (2.1) *yields the following guarantee:*

$$F_{k+1} \leq F_k - \frac{1}{2}\eta\,||d_k||^2 + \eta^2 \frac{L}{2} \sum_{j=(k-K)_+}^{k-1} ||d_j||^2 \tag{3.2}$$

for any step size $0 < \eta \leq \frac{1}{L(K+1)}$.

Proof. We first consider the difference of the errors in consecutive time instances and write

$$\begin{aligned} F(x_{k+1}) - F(x_k) &= f(x_{k+1}) - f(x_k) + r(x_{k+1}) - r(x_k) \\ &\leq \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2}\,||x_{k+1} - x_k||^2 + r(x_{k+1}) - r(x_k), \end{aligned}$$

where the inequality follows from the Taylor series expansion of f around x_k and since the Hessian of f at any point is upper bounded by L by Assumption 3.1. Using the update rule $x_{k+1} = x_k + \eta d_k$ in this inequality, we obtain

$$\begin{aligned}
F(x_{k+1}) - F(x_k) &\leq \eta\langle \nabla f(x_k), d_k\rangle + \eta^2 \frac{L}{2}\left|\left|d_k\right|\right|^2 + r(x_{k+1}) - r(x_k) \\
&= \eta\langle \nabla f(x_k) - g_k, d_k\rangle + \eta^2 \frac{L}{2}\left|\left|d_k\right|\right|^2 + \eta\langle g_k, d_k\rangle + r(x_{k+1}) - r(x_k) \\
&\leq \eta\left|\left|\nabla f(x_k) - g_k\right|\right|\left|\left|d_k\right|\right| + \eta^2 \frac{L}{2}\left|\left|d_k\right|\right|^2 - \eta\left|\left|d_k\right|\right|^2 - \eta\langle h_{k+1}, d_k\rangle \\
&\quad + r(x_{k+1}) - r(x_k) \\
&= \eta\left|\left|\nabla f(x_k) - g_k\right|\right|\left|\left|d_k\right|\right| + \eta\left(\eta\frac{L}{2} - 1\right)\left|\left|d_k\right|\right|^2 + \langle h_{k+1}, x_k - x_{k+1}\rangle \\
&\quad + r(x_{k+1}) - r(x_k) \\
&\leq \eta\left|\left|\nabla f(x_k) - g_k\right|\right|\left|\left|d_k\right|\right| + \eta\left(\eta\frac{L}{2} - 1\right)\left|\left|d_k\right|\right|^2, \qquad (3.3)
\end{aligned}$$

where the second inequality follows by the triangle inequality and the last inequality follows from the convexity of r.

The gradient error term in (3.3), i.e., $||\nabla f(x_k) - g_k||$, can be upper bounded as follows:

$$\begin{aligned}
\left|\left|\nabla f(x_k) - g_k\right|\right| &\leq \frac{1}{m}\sum_{i=1}^{m}\left|\left|\nabla f_i(x_k) - \nabla f_i(x_{\tau_{i,k}})\right|\right| \\
&\leq \frac{1}{m}\sum_{i=1}^{m} L_i\left|\left|x_k - x_{\tau_{i,k}}\right|\right| \\
&\leq \frac{1}{m}\sum_{i=1}^{m} L_i \sum_{j=\tau_{i,k}}^{k-1}\eta\left|\left|d_j\right|\right| \\
&\leq \eta L \sum_{j=(k-K)_+}^{k-1}\left|\left|d_j\right|\right|, \qquad (3.4)
\end{aligned}$$

where the first and third inequalities follow by the triangle inequality, the second inequality follows since each f_i is L_i-smooth and the last inequality follows since $\tau_{i,k} \geq k - K$. Using (3.4) we can upper bound (3.3) as follows:

$$\begin{aligned}
F(x_{k+1}) - F(x_k) &\leq \eta\left(\eta\frac{L}{2} - 1\right)\left|\left|d_k\right|\right|^2 + \eta^2 L \sum_{j=(k-K)_+}^{k-1}\left|\left|d_j\right|\right|\left|\left|d_k\right|\right| \\
&\leq \eta\left(\eta\frac{L(K+1)}{2} - 1\right)\left|\left|d_k\right|\right|^2 + \eta^2\frac{L}{2}\sum_{j=(k-K)_+}^{k-1}\left|\left|d_j\right|\right|^2 \\
&\leq -\frac{\eta}{2}\left|\left|d_k\right|\right|^2 + \eta^2\frac{L}{2}\sum_{j=(k-K)_+}^{k-1}\left|\left|d_j\right|\right|^2, \qquad (3.5)
\end{aligned}$$

where the second inequality follows from the arithmetic-geometric mean inequality, i.e., $||d_j||\,||d_k|| \leq \frac{1}{2}(||d_j||^2 + ||d_k||^2)$ and the last inequality follows since $0 < \eta \leq \frac{1}{L(K+1)}$. This concludes the proof of Lemma 3.4. ∎

We next introduce the following lemma, which can be viewed as an extension of [38, Theorem 4] into our framework with aggregated gradients. We provide a simplified proof compared to [38] with a tighter upper bound. This lemma can be interpreted as follows. When the regularization function is zero (i.e., $r(x) = 0$ for all $x \in \mathbb{R}^n$) and we have access to full gradients (i.e., $K = 0$), this lemma simply follows from the strong convexity of the sum function f since $||x_k - x^*|| \leq \frac{1}{\mu} ||\nabla f(x_k) - \nabla f(x^*)||$ and $\nabla f(x^*) = 0$ due to the optimality condition of the problem. The following lemma indicates that even though we do not have such control over the subgradients of the regularization function (as the regularization function is neither strongly convex nor smooth), the properties of the proximal step yield a similar relation at the expense of a constant of 2 (instead of 1 compared to the $r(x) = 0$ case) and certain history dependent terms (which arise due to the incremental nature of the PIAG algorithm) that has a linear dependence in step size η. This lemma will be a key step in the proof of Lemma 3.7, where we illustrate how the descent term in Lemma 3.4 relates to our Lyapunov function.

LEMMA 3.5. *Suppose that Assumptions* 3.1–3.3 *hold and let* $Q = L/\mu$ *denote the condition number of the problem. Then, the distance of the iterates from the optimal solution is upper bounded as*

$$||x_k - x^*|| \leq \frac{2}{\mu} ||d_k|| + 2\eta Q \sum_{j=(k-K)_+}^{k-1} ||d_j||$$

for any $k \geq 0$ *and* $0 < \eta \leq \frac{1}{L}$.

Proof. Define

$$d'_k \triangleq \arg\min_{d \in \mathbb{R}^n} \left\{ \frac{\eta}{2} ||\nabla f(x_k) + d||^2 + r(x_k + \eta d) \right\},$$

as the direction of update with the full gradient. The nonexpansiveness property of the proximal map implies

$$||\text{prox}_r^\eta(x) - \text{prox}_r^\eta(y)||^2 \leq \langle \text{prox}_r^\eta(x) - \text{prox}_r^\eta(y), x - y \rangle.$$

Putting $x = x_k - \eta \nabla f(x_k)$ and $y = x^* - \eta \nabla f(x^*)$ in the above inequality, we obtain

$$\begin{aligned} ||x_k + \eta d'_k - x^*||^2 &\leq \langle x_k + \eta d'_k - x^*,\, x_k - \eta \nabla f(x_k) - x^* + \eta \nabla f(x^*) \rangle \\ &= \langle x_k + \eta d'_k - x^*,\, x_k + \eta d'_k - x^* \rangle \\ &\quad + \langle x_k + \eta d'_k - x^*,\, -\eta d'_k + \eta \nabla f(x^*) - \eta \nabla f(x_k) \rangle, \end{aligned}$$

which implies

$$0 \leq \langle x_k + \eta d'_k - x^*,\, -d'_k + \nabla f(x^*) - \nabla f(x_k) \rangle.$$

This inequality can be rewritten as follows:

$$\begin{aligned} \langle x_k - x^*,\, \nabla f(x_k) - \nabla f(x^*) \rangle &\leq \langle x_k - x^*,\, -d'_k \rangle - \eta ||d'_k||^2 + \eta \langle d'_k,\, \nabla f(x^*) - \nabla f(x_k) \rangle \\ &\leq \langle x_k - x^*,\, -d'_k \rangle + \eta \langle d'_k,\, \nabla f(x^*) - \nabla f(x_k) \rangle \\ &\leq ||d'_k|| \left(||x_k - x^*|| + \eta ||\nabla f(x^*) - \nabla f(x_k)|| \right) \\ &\leq ||d'_k|| \left(||x_k - x^*|| + \eta L ||x_k - x^*|| \right) \\ &\leq 2 ||d'_k|| \, ||x_k - x^*||, \end{aligned} \tag{3.6}$$

where the second inequality follows since $-||d'_k||^2 \leq 0$, the third inequality follows by the Cauchy–Schwarz inequality, the fourth inequality follows from the L-smoothness of f, and the last inequality follows since $\eta \leq \frac{1}{L}$. Since μ-strong convexity of f implies

$$\mu\,||x_k - x^*||^2 \leq \langle x_k - x^*,\, \nabla f(x_k) - \nabla f(x^*)\rangle, \tag{3.7}$$

combining (3.6) and (3.7) we obtain

$$\mu\,||x_k - x^*|| \leq 2\,||d'_k||\,. \tag{3.8}$$

In order to relate d'_k to the original direction of update d_k, we use the triangle inequality and write

$$\begin{aligned}
||d'_k|| &\leq ||d_k|| + ||d'_k - d_k|| \\
&= ||d_k|| + \frac{1}{\eta}\,||x_k + \eta d'_k - x_k - \eta d_k|| \\
&= ||d_k|| + \frac{1}{\eta}\,||\mathrm{prox}_r^\eta(x_k - \eta\nabla f(x_k)) - \mathrm{prox}_r^\eta(x_k - \eta g_k)|| \\
&\leq ||d_k|| + ||g_k - \nabla f(x_k)|| \\
&\leq ||d_k|| + \eta L \sum_{j=(k-K)_+}^{k-1} ||d_j||\,, \qquad (3.9)
\end{aligned}$$

where the last line follows by (3.4). Putting (3.9) back into (3.8) concludes the proof of Lemma 3.5. ∎

Remark 3.6. Lemma 3.5 can also be extended for nonstrongly convex problems. In particular, in the proof of Lemma 3.5, we use the strong convexity assumption to arrive at inequality (3.8). This inequality is known as the error bound condition for the prox-gradient mapping (cf. [12, Definition 3.1]). In the remainder of the proof, we generalize this error bound condition for the PIAG algorithm (cf. (3.9)). Hence, we can replace the strong convexity assumption by the error bound condition assumption and Lemma 3.5 still holds (in the error bound condition case, x^* need not be unique, and therefore the distances should be defined with respect to the set of optimal solutions as in [12]). In the remainder of the paper, we do not explicitly use the strong convexity assumption but instead make use of Lemma 3.5. Hence, the rate results we present hold under the error bound condition assumption (instead of the strong convexity assumption) as well. We also emphasize that in [12, Corollary 3.6], the authors show that the quadratic functional growth and error bound condition are equivalent when f is C^1-smooth and convex while r is closed and convex (which are the same assumptions we make, excluding strong convexity). The quadratic functional growth and error bound condition are, to our knowledge, the weakest assumptions under which linear convergence of the prox-gradient algorithm is proven [12, 21, 26].

In the following lemma, we relate the direction of update to the suboptimality in the objective value at a given iteration k. In particular, we show that the descent term presented in Lemma 3.4 (i.e., $-||d_k||^2$) can be upper bounded by the negative of the suboptimality in the objective value of the next iteration (i.e., $-F_{k+1}$) and additional history-dependent terms that arise due to the incremental nature of the PIAG algorithm.

LEMMA 3.7. *Suppose that Assumptions* 3.1–3.3 *hold. Then, for any* $0 < \eta \leq \frac{1}{L(K+1)}$, *the PIAG algorithm in* (2.1) *yields the following guarantee:*

$$-\left\|d_k\right\|^2 \leq -\frac{\mu}{4} F_{k+1} + \eta L \sum_{j=(k-K)_+}^{k-1} \left\|d_j\right\|^2 .$$

Proof. In order to prove this lemma, we use Lemma 3.5, which can be rewritten as follows:

$$-\left\|d_k\right\| \leq -\frac{\mu}{2} \left\|x_k - x^*\right\| + \eta L \sum_{j=(k-K)_+}^{k-1} \left\|d_j\right\| .$$

Then, we can upper bound $-\left\|d_k\right\|^2$ as

$$\begin{aligned} -\left\|d_k\right\|^2 &\leq -\frac{\mu}{2} \left\|d_k\right\| \left\|x_k - x^*\right\| + \eta L \sum_{j=(k-K)_+}^{k-1} \left\|d_k\right\| \left\|d_j\right\| \\ &\leq -\frac{\mu}{2} \langle d_k, x^* - x_k \rangle + \eta \frac{KL}{2} \left\|d_k\right\|^2 + \eta \frac{L}{2} \sum_{j=(k-K)_+}^{k-1} \left\|d_j\right\|^2 , \end{aligned} \tag{3.10}$$

where the last line follows by the Cauchy–Schwarz inequality and the arithmetic-geometric mean inequality. We can upper bound the inner product term in (3.10) as

$$\begin{aligned} -\langle d_k, x^* - x_k \rangle &= \langle g_k + h_{k+1}, x^* - x_k \rangle \\ &= \langle \nabla f(x_k), x^* - x_k \rangle + \langle h_{k+1}, x^* - x_k \rangle + \langle g_k - \nabla f(x_k), x^* - x_k \rangle \\ &\leq f(x^*) - f(x_k) + \langle h_{k+1}, x^* - x_{k+1} \rangle + \eta \langle h_{k+1}, d_k \rangle \\ &\quad + \langle g_k - \nabla f(x_k), x^* - x_k \rangle \\ &\leq f(x^*) - f(x_k) + r(x^*) - r(x_{k+1}) + \eta \langle h_{k+1}, d_k \rangle \\ &\quad + \left\|g_k - \nabla f(x_k)\right\| \left\|x^* - x_k\right\| , \end{aligned} \tag{3.11}$$

where the first inequality follows from the convexity of f and the second inequality follows from the convexity of r and the triangle inequality. The inner product term in (3.11) can be upper bounded as

$$\begin{aligned} \eta \langle h_{k+1}, d_k \rangle &= -\eta \left\|d_k\right\|^2 - \langle g_k, \eta d_k \rangle \\ &= -\eta \left\|d_k\right\|^2 + \langle \nabla f(x_k), -\eta d_k \rangle + \langle g_k - \nabla f(x_k), -\eta d_k \rangle \\ &\leq -\eta \left\|d_k\right\|^2 + \langle \nabla f(x_k), x_k - x_{k+1} \rangle + \eta \left\|d_k\right\| \left\|g_k - \nabla f(x_k)\right\| \\ &\leq -\eta \left\|d_k\right\|^2 + f(x_k) - f(x_{k+1}) + \eta^2 \frac{L}{2} \left\|d_k\right\|^2 + \eta \left\|d_k\right\| \left\|g_k - \nabla f(x_k)\right\| , \end{aligned} \tag{3.12}$$

where the first inequality follows by the triangle inequality and the second inequality follows from the L-smoothness of f. Putting (3.12) back in (3.11), we obtain

$$-\langle d_k, x^* - x_k \rangle \leq -F_{k+1} + \eta \left(\eta \frac{L}{2} - 1 \right) \left\|d_k\right\|^2 + \left\|g_k - \nabla f(x_k)\right\| \left(\left\|x^* - x_k\right\| + \eta \left\|d_k\right\| \right) . \tag{3.13}$$

The final term in (3.13) can be upper bounded as follows:

$$\begin{aligned}
&||g_k - \nabla f(x_k)|| \left(||x^* - x_k|| + \eta \, ||d_k||\right) \\
&\quad \leq \eta L \left(\sum_{j=(k-K)_+}^{k-1} ||d_j|| \right) \left(||x^* - x_k|| + \eta \, ||d_k||\right) \\
&\quad \leq \eta L \left(\sum_{j=(k-K)_+}^{k-1} ||d_j|| \right) \left[\left(\eta + \frac{2}{\mu}\right) ||d_k|| + 2\eta Q \sum_{j=(k-K)_+}^{k-1} ||d_j|| \right],
\end{aligned}$$

where the first line follows by (3.4) and the last line follows by Lemma 3.5. Using the arithmetic-geometric mean inequality in the above inequality, we obtain

$$\begin{aligned}
||g_k - \nabla f(x_k)|| \left(||x^* - x_k|| + \eta \, ||d_k||\right) \leq \eta \frac{KL}{2} \left(\eta + \frac{2}{\mu}\right) ||d_k|| \\
+ \eta \left[\eta \frac{L}{2} + Q + 2\eta KQL\right] \sum_{j=(k-K)_+}^{k-1} ||d_j|| \,. \qquad (3.14)
\end{aligned}$$

Putting (3.14) back in (3.13) yields

$$\begin{aligned}
-\langle d_k, x^* - x_k \rangle &\leq -F_{k+1} + \eta \left(\eta \frac{L}{2} - 1 + \frac{KL}{2} \left(\eta + \frac{2}{\mu}\right)\right) ||d_k||^2 \\
&\quad + \eta \left[\eta \frac{L}{2} + Q + 2\eta KQL\right] \sum_{j=(k-K)_+}^{k-1} ||d_j|| \\
&= -F_{k+1} + \eta \left(\eta \frac{(K+1)L}{2} - 1 + KQ\right) ||d_k||^2 \\
&\quad + \eta \left[\eta \frac{L}{2} + Q + 2\eta KQL\right] \sum_{j=(k-K)_+}^{k-1} ||d_j|| \\
&\leq -F_{k+1} + \eta \left(KQ - \frac{1}{2}\right) ||d_k||^2 \\
&\quad + \eta \left(\eta \frac{L}{2} + Q + 2\eta KQL\right) \sum_{j=(k-K)_+}^{k-1} ||d_j|| \,, \qquad (3.15)
\end{aligned}$$

where the last line follows since $\eta \leq \frac{1}{L(K+1)}$. Finally, using (3.15) in our original inequality in (3.10), we obtain

$$\begin{aligned}
-||d_k||^2 &\leq -\frac{\mu}{2} F_{k+1} + \eta \left(\frac{KL}{2} - \frac{\mu}{4} + \frac{KL}{2}\right) ||d_k||^2 \\
&\quad + \eta \left(\eta \frac{\mu L}{4} + \frac{L}{2} + \eta KL^2 + \frac{L}{2}\right) \sum_{j=(k-K)_+}^{k-1} ||d_j||^2 \\
&\leq -\frac{\mu}{2} F_{k+1} + \eta KL \, ||d_k||^2 + \eta L \left(\frac{\mu}{4} + \eta KL + 1\right) \sum_{j=(k-K)_+}^{k-1} ||d_j||^2
\end{aligned}$$

$$\leq -\frac{\mu}{2}F_{k+1} + \eta KL\,||d_k||^2 + \eta L\,(\eta(K+1)L+1)\sum_{j=(k-K)_+}^{k-1}||d_j||^2$$

$$\leq -\frac{\mu}{2}F_{k+1} + ||d_k||^2 + 2\eta L\sum_{j=(k-K)_+}^{k-1}||d_j||^2, \tag{3.16}$$

where the second inequality follows since $\mu \geq 0$, the third inequality follows since $\frac{\mu}{4} \leq L$, and the last inequality follows since $\eta \leq \frac{1}{L(K+1)}$. Rearranging the terms in (3.16), we obtain

$$-||d_k||^2 \leq -\frac{\mu}{4}F_{k+1} + \eta L\sum_{j=(k-K)_+}^{k-1}||d_j||^2, \tag{3.17}$$

which completes the proof of Lemma 3.7. ∎

Before presenting the main result of the paper, we first introduce the following lemma, which was presented in [1, Lemma 1] in a slightly different form. This lemma presents a sufficient condition on the magnitudes of the shocks to the system (represented by Y_k in the lemma) such that these shocks do not disrupt the linear convergence of the original system $\alpha Z_{k+1} \leq Z_k$.

LEMMA 3.8. *Let $\{Z_k\}$ and $\{Y_k\}$ be a sequence of nonnegative real numbers satisfying*

$$\alpha Z_{k+1} \leq Z_k - \beta\,Y_k + \gamma\sum_{j=k-A}^{k}Y_j \tag{3.18}$$

for any $k \geq 0$, for some constants $\alpha > 1$, $\beta \geq 0$, $\gamma \geq 0$ and $A \in \mathbb{Z}^+$. If

$$\gamma(\alpha^{A+1} - 1) \leq \beta(\alpha - 1) \tag{3.19}$$

holds, then $Z_k \leq \alpha^{-k}Z_0$ for all $k \geq 0$.

We next present the main theorem of the paper, which characterizes the linear convergence rate of the PIAG algorithm. In particular, we show that when the step size is sufficiently small, the PIAG algorithm is linearly convergent with a factor that depends on the step size and the strong convexity constant.

THEOREM 3.9. *Suppose that Assumptions* 3.1–3.3 *hold. Then, the PIAG algorithm in* (2.1) *with step size $0 < \eta \leq \frac{16}{\mu}[(1+\frac{1}{48Q})^{\frac{1}{K+1}} - 1]$ is linearly convergent satisfying*

$$F_k \leq \left(1 + \eta\frac{\mu}{16}\right)^{-k}F_0 \tag{3.20}$$

for any $k \geq 0$.

Proof. By Lemma 3.7, we have

$$-\frac{1}{4}\eta\,||d_k||^2 \leq -\eta\frac{\mu}{16}F_{k+1} + \eta^2\frac{L}{4}\sum_{j=(k-K)_+}^{k-1}||d_j||^2.$$

Using this inequality in (3.2) of Lemma 3.4, we get

$$\left(1+\eta\frac{\mu}{16}\right)F_{k+1} \leq F_k - \frac{1}{4}\eta\left|\left|d_k\right|\right|^2 + \eta^2\frac{3L}{4}\sum_{j=(k-K)_+}^{k-1}\left|\left|d_j\right|\right|^2. \tag{3.21}$$

Applying Lemma 3.8 to (3.21) with $Z_k = F_k$ and $Y_k = \left|\left|d_k\right|\right|^2$, we prove (3.20). For this, we need $0 < \eta \leq \frac{1}{L(K+1)}$ in order for Lemmas 3.4 and 3.7 to hold, and

$$\eta^2\frac{3L}{4}\left(\left(1+\eta\frac{\mu}{16}\right)^{K+1}-1\right) \leq \frac{1}{4}\eta\left(\left(1+\eta\frac{\mu}{16}\right)-1\right) \tag{3.22}$$

with $\eta > 0$ for Lemma 3.8 to hold. Simplifying and rearranging terms in (3.22), we obtain

$$\left(1+\eta\frac{\mu}{16}\right)^{K+1}-1 \leq \frac{1}{48Q}.$$

Therefore, for any step size satisfying

$$0 < \eta \leq \frac{16}{\mu}\left[\left(1+\frac{1}{48Q}\right)^{\frac{1}{K+1}}-1\right], \tag{3.23}$$

Lemma 3.8 holds. We can also observe that the right-hand side of (3.23) can be upper bounded using the Bernoulli inequality, i.e., $(1+x)^r \leq 1+rx$ for any $x \geq -1$ and $r \in [0,1]$, as follows:

$$\begin{aligned}\eta &\leq \frac{16}{\mu}\left(1+\frac{1}{48Q(K+1)}-1\right)\\ &= \frac{1}{3L(K+1)}.\end{aligned} \tag{3.24}$$

Thus, the constraint (3.23) satisfies the constraint $0 < \eta \leq \frac{1}{L(K+1)}$ in Lemmas 3.4 and 3.7 as well. Then, applying Lemma 3.8 to (3.21) yields (3.20). ∎

We next introduce the following corollary, which highlights the main result of the paper. This corollary indicates that for an appropriately chosen step size (which does not depend on the strong convexity constant μ), the PIAG algorithm is guaranteed to return an ϵ-optimal solution after $\mathcal{O}(QK\log(1/\epsilon))$ iterations.

COROLLARY 3.10. *Suppose that Assumptions* 3.1–3.3 *hold. Then, the PIAG algorithm in* (2.1) *with step size* $0 < \eta \leq \frac{16}{49L(K+1)}$ *is linearly convergent satisfying*

$$F_k \leq \left(1+\eta\frac{\mu}{16}\right)^{-k}F_0 \tag{3.25}$$

for any $k \geq 0$. *In particular, the PIAG algorithm with step size* $\eta = \frac{16}{49L(K+1)}$ *is guaranteed to return an* ϵ*-optimal solution after at most* $50Q(K+1)\log(F_0/\epsilon)$ *iterations.*

Proof. By straightforward algebra, we observe that

$$\begin{aligned}\left(1+\frac{1}{48Q}\right)^{\frac{1}{K+1}} &= \sum_{i=0}^{\infty}\binom{\frac{1}{K+1}}{i}\left(\frac{1}{48Q}\right)^{i}\\ &\geq 1+\frac{1}{48Q(K+1)}+\frac{1}{2(K+1)}\left(\frac{1}{K+1}-1\right)\left(\frac{1}{48Q}\right)^{2}\\ &= 1+\frac{1}{48Q(K+1)}\left(1-\frac{1}{96Q}\left(1-\frac{1}{K+1}\right)\right)\\ &\geq 1+\frac{1}{48Q(K+1)}\left(1-\frac{1}{96}\right)\\ &\geq 1+\frac{1}{49Q(K+1)}.\end{aligned}$$

Therefore, we have

$$\frac{16}{\mu}\left[\left(1+\frac{1}{48Q}\right)^{\frac{1}{K+1}}-1\right]\geq\frac{16}{\mu}\left[1+\frac{1}{49Q(K+1)}-1\right]=\frac{16}{49L(K+1)}.$$

Hence, by Theorem 3.9, we conclude that (3.20) holds for any $0<\eta\leq\frac{16}{49L(K+1)}$. Putting the step size $\eta=\frac{16}{49L(K+1)}$ in (3.20), we obtain

$$F_k\leq\left(1+\frac{1}{49Q(K+1)}\right)^{-k}F_0\leq\left(1-\frac{1}{50Q(K+1)}\right)^{k}F_0,$$

where the last inequality follows since $Q\geq 1$ and $K\geq 0$. Taking logarithms of both sides yields

$$\begin{aligned}\log(F_k)&\leq\log(F_0)+k\log\left(1-\frac{1}{50Q(K+1)}\right)\\ &\leq\log(F_0)-\frac{k}{50Q(K+1)},\end{aligned}$$

where the last line follows since $\log(1+x)\leq x$ for any $x>-1$. Therefore, for any k satisfying

$$\log(F_0)-\frac{k}{50Q(K+1)}\leq\log(\epsilon), \tag{3.26}$$

x_k is an ϵ-optimal solution. Rearranging terms in (3.26), we conclude that for any $k\geq 50Q(K+1)\log(F_0/\epsilon)$, x_k is an ϵ-optimal solution. ∎

In the following corollary, we provide a guarantee on the iterates generated by the PIAG algorithm, which directly follows by Theorem 3.9 and the strong convexity of F, i.e., $F(x_k)-F(x^*)\geq\frac{\mu}{2}\left\|x_k-x^*\right\|^2$.

COROLLARY 3.11. *Suppose that Assumptions* 3.1–3.3 *hold. Then, the iterates generated by the PIAG algorithm with step size* $0<\eta\leq\frac{16}{\mu}[(1+\frac{1}{48Q})^{\frac{1}{K+1}}-1]$ *satisfy the following guarantee:*

$$\left\|x_k-x^*\right\|^2\leq\left(1+\eta\frac{\mu}{16}\right)^{-k}\frac{2F_0}{\mu} \tag{3.27}$$

for any $k\geq 0$.

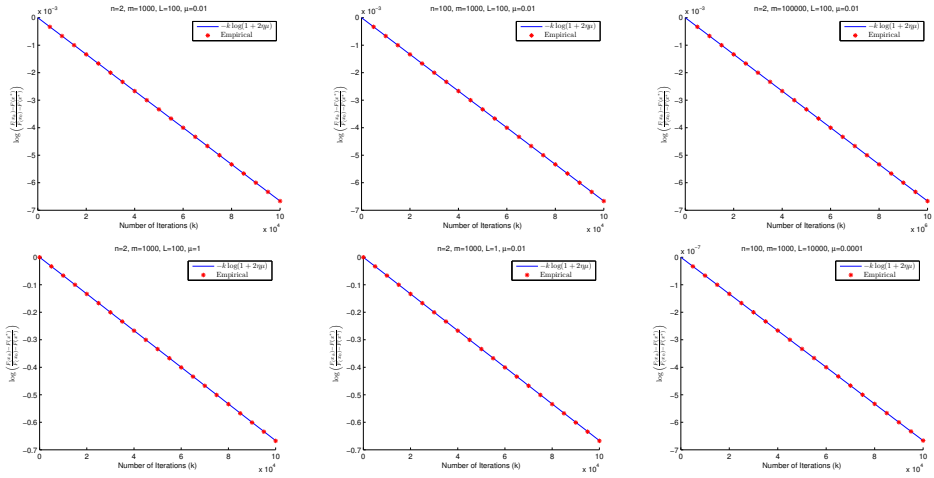

FIG. 4.1. *Red dots show the empirical performance of the PIAG algorithm for various m, L, μ, and n under the worst-case initialization. Blue lines have slope $-\log(1+2\eta\mu)$.*

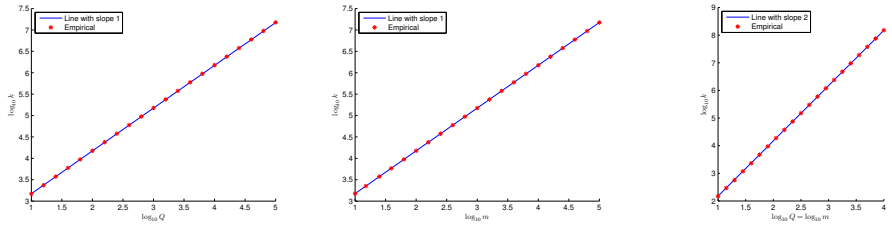

FIG. 4.2. *Number of iterations required for the PIAG algorithm to decrease the function suboptimality by e^{-1} versus Q (left figure), m (middle figure), and Qm (right figure).*

4. Numerical examples. In this section, we investigate the tightness of our bounds via numerical examples and compare the performance of the PIAG algorithm with the state-of-the-art methods in the literature. First, we construct an example and vary the parameters (n, m, L, and μ) of this example to illustrate that our upper bounds are tight up to constants. Then, we present how the iteration complexity of PIAG changes with the condition number of the problem Q and the number of component functions m. In our experiments, we let $r = 0$, $f_i = f$ for all $i \in \{1, \dots, m\}$ and define $f(x) = \frac{1}{2}x^T Ax$, where $A = \mathrm{diag}(a_1, \dots, a_n)$, where $a_1 = \mu$, $a_2 = L$, and $\{a_i\}_{i>2}$ are chosen independently and uniformly at random from the interval $[\mu, L]$. We initialize the PIAG algorithm with $x_0 = [1, 0, \dots, 0]^T$ in order to show the worst-case rates. We apply the PIAG algorithm to this problem with the maximum allowable step size in Theorem 3.9.

In Figure 4.1, we plot the empirical performance of PIAG applied to the described problem for various values of n, m, L, and μ. The red dots in figures show the empirical performance of PIAG, whereas blue lines have slope $-\log(1+2\eta\mu)$ (see color figures in the online version). From Figure 4.1, we observe that even though the condition number of the problem Q and the number of component functions m change, the convergence rate of the PIAG algorithm stays around $-\log(1+2\eta\mu)$. This verifies that the contraction factor in Theorem 3.9 is tight up to constants.

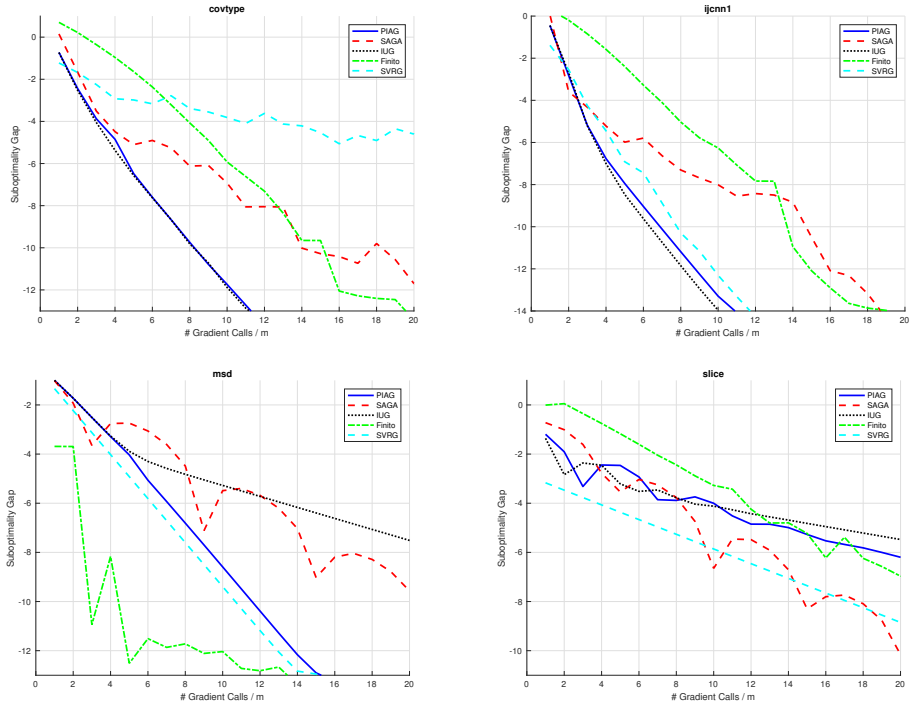

FIG. 4.3. *Convergence rate plots for the training loss on benchmark datasets.*

We next illustrate (in Figure 4.2) how the number of iterations required to decrease the function suboptimality by a factor of e^{-1} changes as Q and m increase. In the left figure, we set $n = 2$, $m = 100$, $\mu = 1$ and vary $L \in [10, 10^5]$, whereas in the middle figure, we set $n = 2$, $L = 100$, $\mu = 1$ and vary $m \in [10, 10^5]$. As can be seen from the figures, the number of iterations to achieve the same suboptimality in function values increases linearly with Q and m. Finally, in the right figure, we set $n = 2$, $\mu = 1$ and vary $m = L \in [10, 10^4]$. The x-axis in this figure is set to $\log(m)$ (or equivalently $\log(Q)$ since $L = m$ and $\mu = 1$). Corollary 3.10 implies that the iteration complexity of PIAG for this problem is $\mathcal{O}(Qm \log(1/e^{-1})) = \mathcal{O}(m^2)$ and as we observe in the right figure, the logarithm of the number of iterations (to decrease function suboptimality by e^{-1}) grows as $2\log(m)$, which verifies Corollary 3.10.

Finally, we apply the PIAG algorithm to several datasets. We compare the PIAG algorithm with the incrementally updated gradient (IUG) algorithm of [39] and the SAGA [9], Finito [10], and SVRG [20] algorithms. For all algorithms, we use step sizes that yields the fastest convergence rate. At each iteration, all algorithms make a single call to the gradient oracle. For the PIAG and IUG algorithms, the order of the data is reshuffled between each epoch, and hence we have $K = 2m - 1$. In SAGA, SVRG and Finito, the data is chosen uniformly at random as stated in their algorithm descriptions.

We performed L2-regularized logistic regression on the covtype and ijcnn1 datasets[3] (for binary classification), with regularization parameters 10^{-4} and 10^{-3}, respectively. Then, we considered the LASSO problem on the million song year and

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

slice-localization datasets[4], with L1-regularization parameters 10^{-2} and 10^{-3}, respectively. Figure 4.3 shows the results of our experiments. We can observe that when the regularization parameter λ is small, the performances of the PIAG and IUG algorithms are similar, whereas there is a distinct difference between them as λ increases. This follows due to the difference in the scaling of the step lengths with respect to λ in the proximal mapping of these two algorithms. For the regression problems with relatively large strong convexity constant and/or large number of data points (i.e., when the big data condition holds [10], which is the regime under which the convergence rate results for FInito hold), we observe that the convergence rate of Finito is faster with respect to the other algorithms. Overall, we can conclude that the performance of the PIAG algorithm is competitive with respect to the state-of-the-art algorithms and the rates of convergence of these algorithms are usually comparable to one another.

5. Concluding remarks. In this paper, we studied the PIAG method for additive composite optimization problems of the form (1.1). We showed the first linear convergence rate result for the PIAG method and provided explicit convergence rate estimates that highlight the dependence on the condition number of the problem and the size of the window K over which outdated component gradients are evaluated (under the assumptions that $f(x)$ is strongly convex and each $f_i(x)$ is smooth with Lipschitz gradient). Our results hold for any deterministic order (in processing the component functions) in contrast to the existing work on stochastic variants of our algorithm, which presents convergence results in expectation.

Appendix A. Proof of Lemma 3.8. Multiplying both sides of (3.18) by α^k and summing from $k = 0$ to p, we get

$$\begin{aligned}
\sum_{k=0}^{p} \alpha^{k+1} Z_{k+1} &\leq \sum_{k=0}^{p} \left(\alpha^k Z_k - \alpha^k \beta Y_k + \alpha^k \gamma \sum_{j=k-A}^{k} Y_j \right) \\
&= \sum_{k=0}^{p} \alpha^k Z_k - \beta \sum_{k=0}^{p} \alpha^k Y_k + \gamma \sum_{k=0}^{p} \sum_{j=k-A}^{k} \alpha^k Y_j \\
&\leq \sum_{k=0}^{p} \alpha^k Z_k - \beta \sum_{k=0}^{p} \alpha^k Y_k + \gamma \sum_{k=0}^{p} \left(\sum_{j=k}^{k+A} \alpha^j \right) Y_k \\
&= \sum_{k=0}^{p} \alpha^k Z_k - \beta \sum_{k=0}^{p} \alpha^k Y_k + \gamma \sum_{k=0}^{p} \left(\alpha^k \frac{\alpha^{A+1} - 1}{\alpha - 1} \right) Y_k \\
&= \sum_{k=0}^{K} \alpha^k Z_k - \sum_{k=0}^{K} \alpha^k \left(\beta - \gamma \frac{\alpha^{A+1} - 1}{\alpha - 1} \right) Y_k \\
&\leq \sum_{k=0}^{p} \alpha^k Z_k, \qquad \text{(A.1)}
\end{aligned}$$

where the last line follows by (3.19). Telescoping the terms in both sides, we get $\alpha^{p+1} Z_{p+1} \leq Z_0$.

[4] https://archive.ics.uci.edu/ml/datasets.html

REFERENCES

[1] A. Aytekin, H. R. Feyzmahdavian, and M. Johansson, *Analysis and Implementation of an Asynchronous Optimization Algorithm for the Parameter Server*, preprint, http://arxiv.org/abs/1610.05507, 2016.

[2] D. P. Bertsekas, *Incremental gradient, subgradient, and proximal methods for convex optimization: A survey*, in Optimization for Machine Learning, S. Sra, S. Nowozin, and S. Wright, eds., MIT Press, Cambridge, MA, 2011, pp. 1–38.

[3] D. P. Bertsekas, *Incremental proximal methods for large scale convex optimization*, Math. Program., 129 (2011), pp. 163–195.

[4] D. P. Bertsekas, *Convex Optimization Algorithms*, Athena Scientific, Nashua, NH, 2015.

[5] D. P. Bertsekas, *Incremental Aggregated Proximal and Augmented Lagrangian Algorithms*, preprint, arXiv:1509.09257 [cs.SY], 2015.

[6] D. Blatt, A. Hero, and H. Gauchman, *A convergent incremental gradient method with a constant step size*, SIAM J. Optim., 18 (2007), pp. 29–51.

[7] S. Bubeck, *Theory of Convex Optimization for Machine Learning*, preprint, https://arxiv.org/abs/1405.4980, 2014.

[8] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comput., 20 (1998), pp. 33–61.

[9] A. Defazio, F. Bach, and S. Lacoste-Julien, *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, in Adv. Neural Inf. Process. Syst. 27, MIT Press, Cambridge, MA, 2014, pp. 1646–1654.

[10] A. J. Defazio, T. S. Caetano, and J. Domke, *Finito: A faster, permutable incremental gradient method for big data problems*, in Adv. Neural Inf. Process. Syst. 26, MIT Press, Cambridge, MA, 2013, pp. 315–323.

[11] D. L. Donoho, *Compressed sensing*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1289–1306.

[12] D. Drusvyatskiy and A. S. Lewis, *Error Bounds, Quadratic Growth, and Linear Convergence of Proximal Methods*, preprint, http://arxiv.org/abs/1602.06661, 2016.

[13] J. C. Duchi, A. Agarwal, and M. J. Wainwright, *Dual averaging for distributed optimization: Convergence analysis and network scaling*, IEEE Trans. Automat. Control, 57 (2012), pp. 592–606.

[14] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE J. Sel. Topics Signal Process., 1 (2007), pp. 586–597.

[15] G. B. Giannakis, V. Kekatos, N. Gatsis, S. J. Kim, H. Zhu, and B. F. Wollenberg, *Monitoring and optimization for power grids: A signal processing perspective*, IEEE Signal Process. Mag., 30 (2013), pp. 107–128.

[16] F. Guo, C. Wen, J. Mao, and Y. D. Song, *Distributed economic dispatch for smart grids with random wind power*, IEEE Trans. Smart Grid, 7 (2016), pp. 1572–1583.

[17] M. Gurbuzbalaban, A. Ozdaglar, and P. Parrilo, *On the Convergence Rate of Incremental Aggregated Gradient Algorithms*, preprint, arXiv:1506.02081, 2015.

[18] M. Gurbuzbalaban, A. Ozdaglar, and P. Parrilo, *Why Random Reshuffling Beats Stochastic Gradient Descent*, preprint, arXiv:1510.08560, 2015.

[19] T. Hazan, A. Man, and A. Shashua, *A parallel decomposition solver for svm: Distributed dual ascend using fenchel duality*, in IEEE Conference on Computer Vision and Pattern Recognition, 2008, IEEE Press, Piscataway, NJ, 2008, pp. 1–8.

[20] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in Proceedings of the Conference Neural Information Processing Systems 2013, Adv. Neural Inf. Process. Syst. 26, MIT Press, Cambridge, MA, 2014, pp. 315–323.

[21] H. Karimi and M. Schmidt, *Linear convergence of proximal-gradient methods under the Polyak–Łojasiewicz condition*, in Machine Learning and Knowledge Discovery in Databases, P. Frasconi, N. Landwehr, G. Manco, and J. Vreeken, eds., Lecture Notes in Comput. Sci. 9851, Springer, Berlin, 2016, pp. 795–811.

[22] H. Lin, J. Mairal, and Z. Harchaoui, *A universal catalyst for first-order optimization*, in Proceedings of the Conference Neural Information Processing Systems 2015, Adv. Neural Inf. Process. Syst. 28, MIT Press, Cambridge, MA, 2016, pp. 3384–3392.

[23] J. Mairal, *Incremental majorization-minimization optimization with application to large-scale machine learning*, SIAM J. Optim., 25 (2015), pp. 829–855.

[24] G. Mateos, J. A. Bazerque, and G. B. Giannakis, *Distributed sparse linear regression*, IEEE Trans. Signal Process., 58 (2010), pp. 5262–5276.

[25] I. Necoara and D. Clipici, *Parallel random coordinate descent method for composite minimization: Convergence analysis and error bounds*, SIAM J. Optim., 26 (2016), pp. 197–226.

[26] I. Necoara, Y. Nesterov, and F. Glineur, *Linear Convergence of First Order Methods for Non-Strongly Convex Optimization*, preprint, http://arxiv.org/abs/1504.06298, 2016.
[27] A. Nedic and D. P. Bertsekas, *Incremental subgradient methods for nondifferentiable optimization*, SIAM J. Optim., 12 (2001), pp. 109–138; also available online from https://doi.org/10.1137/S1052623499362111.
[28] A. Nedic, D. P. Bertsekas, and V. S. Borkar, *Distributed asynchronous incremental subgradient methods*, in Inherently Parallel Algorithms in Feasibility and Optimization and their Applications, Stud. Comput. Math. 8, Elsevier, New York, 2001, pp. 381–407.
[29] Y. Nesterov, *Primal-dual subgradient methods for convex problems*, Math. Program., 120 (2009), pp. 221–259.
[30] F. Niu, B. Recht, C. Re, and S. Wright, *Hogwild: A lock-free approach to parallelizing stochastic gradient descent*, in Proceedings of the Conference Neural Information Processing Systems 2010, Adv. Neural Inf. Process. Syst. 24, MIT Press, Cambridge, MA, 2011, pp. 693–701.
[31] A. Padakandla and R. Sundaresan, *Separable convex optimization problems with linear ascending constraints*, SIAM J. Optim., 20 (2010), pp. 1185–1204.
[32] N. Parikh and S. Boyd, *Proximal algorithms*, Found. Trends Optim., 1 (2014), pp. 127–239.
[33] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Stat., 22 (1951), pp. 400–407; also available online from http://www.jstor.org/stable/2236626.
[34] N. L. Roux, M. Schmidt, and F. R. Bach, *A stochastic gradient method with an exponential convergence rate for finite training sets*, in Proceedings of the Conference Neural Information Processing Systems 2011, Adv. Neural Inf. Process. Syst. 25, MIT Press, Cambridge, MA, 2012, pp. 2663–2671.
[35] W. Shi, Q. Ling, G. Wu, and W. Yin, *A proximal gradient algorithm for decentralized composite optimization*, IEEE Trans. Signal Process., 63 (2015), pp. 6013–6023.
[36] M. V. Solodov, *Incremental gradient algorithms with stepsizes bounded away from zero*, Comput. Optim. Appl, 11 (1998), pp. 23–35.
[37] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. R. Stat. Soc. Ser. B Stat. Methodol., 58 (1994), pp. 267–288.
[38] P. Tseng and S. Yun, *A coordinate gradient descent method for nonsmooth separable minimization*, Math. Program., 117 (2009), pp. 387–423.
[39] P. Tseng and S. Yun, *Incrementally updated gradient methods for constrained and regularized optimization*, J. Optim. Theory Appl., 160 (2014), pp. 832–853.
[40] J. Tsitsiklis, D. Bertsekas, and M. Athans, *Distributed asynchronous deterministic and stochastic gradient optimization algorithms*, IEEE Trans. Automat. Control, 31 (1986), pp. 803–812.
[41] L. Xiao, *Dual averaging methods for regularized stochastic learning and online optimization*, J. Mach. Learn. Res., 11 (2010), pp. 2543–2596.
[42] H.-F. Yu, F.-L. Huang, and C.-J. Lin, *Dual coordinate descent methods for logistic regression and maximum entropy models*, Mach. Learn., 85 (2011), pp. 41–75.