Journal of Machine Learning Research 19 (2018) 1-32 Submitted 07/17; Revised 06/18; Published 07/18

Distributed Proximal Gradient Algorithm for Partially Asynchronous Computer Clusters *

Yi Zhou ZHOU.1172@OSU.EDU
Yingbin Liang LIANG.889@OSU.EDU
Department of Electrical and Computer Engineering
The Ohio State University

Yaoliang Yu YAOLIANG.YU@UWATERLOO.CA
Department of Computer Science
University of Waterloo

Wei Dai WDAI@CS.CMU.EDU
Eric P. Xing EPXING@CS.CMU.EDU
Machine Learning Department
Carnegie Mellon University

Editor: Tong Zhang

Abstract

With ever growing data volume and model size, an error-tolerant, communication efficient, yet versatile distributed algorithm has become vital for the success of many large-scale machine learning applications. In this work we propose m-PAPG, an implementation of the flexible proximal gradient algorithm in model parallel systems equipped with the partially asynchronous communication protocol. The worker machines communicate asynchronously with a controlled staleness bound s and operate at different frequencies. We characterize various convergence properties of m-PAPG: 1) Under a general non-smooth and non-convex setting, we prove that every limit point of the sequence generated by m-PAPG is a critical point of the objective function; 2) Under an error bound condition of convex objective functions , we prove that the optimality gap decays linearly for every s steps; 3) Under the Kurdyka-Łojasiewicz inequality and a sufficient decrease assumption , we prove that the sequences generated by m-PAPG converge to the same critical point, provided that a proximal Lipschitz condition is satisfied.

Keywords: proximal gradient, distributed system, model parallel, partially asynchronous, machine learning

1. Introduction

The composite minimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \; f(\mathbf{x}) + g(\mathbf{x}) \tag{1}$$

*. The material in this paper is presented in part at the 19th International Conference on Artificial Intelligence and Statistics (AISTATS), Cadiz, Spain, 2016.

has drawn a lot of recent attention due to its ubiquity in machine learning and statistical applications. Typically, the first term

$$f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \tag{2}$$

is a smooth loss function over n training samples that describes the fitness to data, and the second term g is a nonsmooth regularization function that encodes *a priori* information. We list below some popular examples under this framework.

- Lasso: least squares loss $f_i(\mathbf{x}) = (y_i - \mathbf{a}_i^\top \mathbf{x})^2$ and ℓ_1 norm regularizer $g(\mathbf{x}) = \|\mathbf{x}\|_1$;
- Logistic regression: logistic loss $f_i = \log(1 + \exp(-y_i \mathbf{a}_i^\top \mathbf{x}_i))$;
- Boosting: exponential loss $f_i(\mathbf{x}) = \exp(-y_i \mathbf{a}_i^\top \mathbf{x})$;
- Support vector machines: hinge loss $f_i(\mathbf{x}) = \max\{0, 1 - y_i \mathbf{a}_i^\top \mathbf{x}\}$ and (squared) ℓ_2 norm regularizer $g(\mathbf{x}) = \|\mathbf{x}\|_2^2$.

Over the years there is also a rising interest in using nonconvex losses f (mainly for robustness against outlying observations) Collobert et al. (2006); Wu and Liu (2007); Xu et al. (2006); Yu et al. (2015) and nonconvex regularizers g (mainly for smaller bias in statistical estimation) Fan and Li (2001); Zhang and Zhang (2012).

Due to the apparent importance of the composite minimization framework and the rapidly growing size in both dimension (d) and volume (n) of data, there is a strong need to develop a practical *parallel* system that can solve the problem in (1) efficiently and in a scale that is impossible for a single machine Agarwal and Duchi (2011); Bertsekas and Tsitsiklis (1989); Dean and Ghemawat (2008); Feyzmahdavian et al. (2014); Ho et al. (2013); Li et al. (2014); Low et al. (2012); Zaharia et al. (2010). Existing systems can be categorized by how communication among worker machines is managed: bulk synchronous (also called fully synchronous) Dean and Ghemawat (2008); Valiant (1990); Zaharia et al. (2010); Lorenzo and Scutari (2016), totally asynchronous Baudet (1978); Bertsekas and Tsitsiklis (1989); Low et al. (2012), and partially asynchronous (a.k.a. stale synchronous or chaotic) Agarwal and Duchi (2011); Bertsekas and Tsitsiklis (1989); Chazan and Miranker (1969); Feyzmahdavian et al. (2014); Ho et al. (2013); Li et al. (2014); Tseng (1991). Bulk synchronous parallel (BSP) systems explicitly force synchronization barriers so that the worker machines can stay on the same page to ensure correctness. However, in a real deployed parallel system, BSP usually suffers from the straggler problem, that is, the performance of the whole system is bottlenecked at the bandwidth of communication and the *slowest* worker machine. On the other hand, totally asynchronous systems do not put any constraint on synchronization, hence achieve much greater throughputs by potentially sacrificing the correctness of the algorithm. Partially asynchronous parallel (PAP) systems Bertsekas and Tsitsiklis (1989); Chazan and Miranker (1969) are a compromise between the previous two: it allows the worker machines to communicate asynchronously up to a controlled staleness and to perform updates at different paces. PAP is particularly suitable for machine learning applications, where iterative algorithms that are robust to small computational errors are usually favored for finding an appropriate solution. Due to its flexibility, the PAP mechanism has been the method of choice in many recent practical implementations Agarwal and Duchi (2011);

Feyzmahdavian et al. (2014); Ho et al. (2013); Li et al. (2014); Liu and Wright (2015); Recht et al. (2011).

Existing parallel systems can also be categorized by how computation is divided among worker machines: data parallel and model parallel. Data parallel systems usually distribute the computation involving each component function f_i in (2) into different worker machines, which is suitable when $n \gg d$, i.e., large data volume but moderate model size. In this setting the stochastic proximal gradient algorithm, along with the PAP protocol, has been shown to be quite effective in solving the composite problem (1) Agarwal and Duchi (2011); Feyzmahdavian et al. (2014); Ho et al. (2013); Li et al. (2014). Some other works developed ADMM-based algorithms for data parallelism Hong et al. (2016) and stochastic variance-reduced gradient algorithms under the PAP protocol Huo and Huang (2017); Fang and Lin (2017), and proved their effectiveness both theoretically and empirically. In this work, we focus on the "dual" model parallel regime where $d \gg n$, i.e., large model size but moderate data volume. In modern machine learning and statistics applications, it is not uncommon that the dimensionality of data largely exceeds its volume, for example, in computational biology, conducting an experimental study that involves many patients can be very expensive but for each patient, technology (e.g. next-generation genome sequencing) has advanced to a stage where taking a large number of measurements (model parameters) is relatively cheap. Deep neural networks are another example that calls for model parallelism. Not surprisingly, the design of a model parallel system is fundamentally different from that of a data parallel system, and so is the subsequent analysis.

To achieve model parallelism, the model $\mathbf{x}$ is partitioned into different (disjoint) blocks and is distributed among many worker machines. In this setting, the block proximal gradient algorithm has been proposed to solve the composite problem (1) Fercoq and Richtárik (2015); Lu and Xiao (2015); Richtárik and Takáč (2014), although under the more restrictive BSP protocol. Other works proposed ADMM-based algorithm for model parallelism to solve the sparse PCA problem Hajinezhad and Hong (2015). Under the PAP protocol, the only work that we are aware of is Bertsekas and Tsitsiklis (1989) which focused on a special case of (1) where g is an indicator function of a convex set, and Tseng (1991) which established a periodic linear rate of convergence under an error bound condition. Our main goal in this work is to provide a formal convergence analysis of the model parallel proximal gradient algorithm under the more flexible PAP communication protocol, and our results naturally extend those in Bertsekas and Tsitsiklis (1989); Tseng (1991) to allow nonsmooth and nonconvex functions.

Our main contributions in this work are: 1). We propose m-PAPG, an extension of the proximal gradient algorithm to the model parallel and partially asynchronous setting. In specific, the worker machines in the system can communicate with each other to synchronize the model parameters with staleness. 2). We provide a rigorous analysis of the convergence properties of m-PAPG, allowing both *nonsmooth* and *nonconvex* functions. In particular, we prove in Theorem 7 that any limit point of the sequences generated by m-PAPG is a critical point. 3) Under an additional error bound condition of convex objective functions, we prove in Theorem 9 that the function values generated by m-PAPG decays periodically linearly. 4) Lastly, using the Kurdyka-Łojasiewicz (KŁ) inequality Bolte et al. (2014) and under a sufficient decrease assumption, we prove in Theorem 11 that for functions that

satisfy a proximal Lipschitz condition the whole sequences of m-PAPG converge to a single critical point.

This paper proceeds as follows: We first set up the notations and definitions in Section 2. The proposed algorithm m-PAPG is presented in Section 3, and convergence analysis are detailed in Sections 4 to 6. The implementation of m-PAPG on a distributed system is detailed in Section 7, and numerical experiments are reported in Section 8. Section 9 concludes our work.

2. Preliminaries

We first recall some fundamental definitions that will be needed in our analysis. Throughout, $h : \mathbb{R}^d \to (-\infty, +\infty]$ denotes an extended real-valued function that is proper and closed, i.e., its domain $\operatorname{dom} h := \{\mathbf{x} : h(\mathbf{x}) < +\infty\}$ is nonempty and its sublevel set $\{\mathbf{x} : h(\mathbf{x}) \leq \alpha\}$ is closed for all $\alpha \in \mathbb{R}$. Since the function h may not be smooth or convex, we need the following generalized notion of "derivative."

Definition 1 (Subdifferential and critical point, e.g. Rockafellar and Wets (1997)) *The Frechét subdifferential $\hat{\partial} h$ of h at $\mathbf{x} \in \operatorname{dom} h$ is the set of $\mathbf{u}$ such that*

$$\liminf_{\mathbf{z} \neq \mathbf{x}, \mathbf{z} \to \mathbf{x}} \frac{h(\mathbf{z}) - h(\mathbf{x}) - \mathbf{u}^\top (\mathbf{z} - \mathbf{x})}{\|\mathbf{z} - \mathbf{x}\|} \geq 0, \tag{3}$$

while the (limiting) subdifferential ∂h at $\mathbf{x} \in \operatorname{dom} h$ is the "closure" of $\hat{\partial} h$:

$$\{\mathbf{u} : \exists \mathbf{x}^k \to \mathbf{x}, h(\mathbf{x}^k) \to h(\mathbf{x}), \mathbf{u}^k \in \hat{\partial} h(\mathbf{x}^k), \mathbf{u}^k \to \mathbf{u}\}. \tag{4}$$

The critical points of h are $\operatorname{crit} h := \{\mathbf{x} : \mathbf{0} \in \partial h(\mathbf{x})\}$.

When h is continuously differentiable or convex, the subdifferential ∂h and the set of critical points $\operatorname{crit} h$ coincide with the usual notions. For a closed function h, its subdifferential is either nonempty at any point in its domain or the subgradient diverges to some "direction" (Rockafellar and Wets, 1997, Corollary 8.10).

Definition 2 (Distance and projection) *The distance function w.r.t. a closed set $\Omega \subseteq \mathbb{R}^d$ is defined as:*

$$\operatorname{dist}_\Omega(\mathbf{x}) := \min_{\mathbf{y} \in \Omega} \|\mathbf{y} - \mathbf{x}\|, \tag{5}$$

while the metric projection onto Ω is defined as:

$$\operatorname{proj}_\Omega(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{y} \in \Omega} \|\mathbf{y} - \mathbf{x}\|, \tag{6}$$

where $\|\cdot\|$ is the usual Euclidean norm.

Note that $\operatorname{proj}_\Omega(\mathbf{x})$ is single-valued for all $\mathbf{x} \in \mathbb{R}^d$ if and only if Ω is convex.

Definition 3 (Proximal map, e.g. Rockafellar and Wets (1997)) *The proximal map of a closed and proper function h is (with parameter $\eta > 0$):*

$$\operatorname{prox}_h^\eta(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{z} \in \mathbb{R}^d} h(\mathbf{z}) + \tfrac{1}{2\eta} \|\mathbf{z} - \mathbf{x}\|^2. \tag{7}$$

Occasionally, we will write prox_h *instead of* prox_h^1.

Clearly, for the indicator function $h(\mathbf{x}) = \iota_\Omega(\mathbf{x})$, which takes the value 0 for $\mathbf{x} \in \Omega$ and ∞ otherwise, its proximal map (with any $\eta > 0$) reduces to the metric projection proj_Ω. If h decreases slower than a quadratic function (in particular, when h is bounded below), then its proximal map is well-defined for all (small) η Rockafellar and Wets (1997). If h is convex, then its proximal map is always a singleton while for nonconvex h, the proximal map can be set-valued. In the latter case we will also abuse the notation $\mathrm{prox}_h^\eta(\mathbf{x})$ for an arbitrary element from that set. For convex functions, the proximal map is nonexpansive:

$$\forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^d,\ \|\mathrm{prox}_h^\eta(\mathbf{x}) - \mathrm{prox}_h^\eta(\mathbf{y})\| \le \|\mathbf{x} - \mathbf{y}\|, \tag{8}$$

while for nonconvex functions this may not hold everywhere.

The proximal map is the key component of the proximal gradient algorithm Fukushima and Mine (1981) (a.k.a. forward-backward splitting):

$$\forall\, t = 0, 1, \ldots, \quad \mathbf{x}(t+1) = \mathrm{prox}_g^\eta(\mathbf{x}(t) - \eta \nabla f(\mathbf{x}(t))), \tag{9}$$

where ∇f is the (sub)gradient of f, and η is a suitable step size (that may change with t). It is known that when f is convex with L-Lipschitz continuous gradient and $0 < \eta < 2/L$, then $F_t := f(\mathbf{x}(t)) + g(\mathbf{x}(t))$ converges to the minimum at the rate $O(1/t)$ and $\mathbf{x}(t)$ converges to some minimizer $\mathbf{x}^*$. Accelerated versions Beck and Teboulle (2009); Nesterov (2013) where F_t converges at the faster rate $O(1/t^2)$ are also well-known. Recently, Bolte et al. (2014) proved that $\mathbf{x}(t)$ converges to a critical point even for nonconvex f and nonconvex and nonsmooth g as long as together they satisfy a certain KŁ inequality.

3. Formulation of m-PAPG

Recall the composite minimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^d} F(\mathbf{x}), \qquad \text{where} \qquad F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}). \tag{P}$$

We are interested in the case where d is so large that implementing the proximal gradient algorithm (9) on a single machine is no longer feasible, hence distributed computation is necessary.

We consider a **model** parallel system with p machines in total. The machines are fully connected and can communicate with each other. Decompose the d model parameters into p disjoint groups. Formally, consider the decomposition $\mathbb{R}^d = \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times \cdots \times \mathbb{R}^{d_p}$, and denote x_i and $\nabla_i f(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^{d_i}$ as the i-th component of $\mathbf{x}$ and $\nabla f(\mathbf{x})$, respectively. Clearly, $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ and $\nabla f = (\nabla_1 f, \nabla_2 f, \cdots, \nabla_p f)$. The i-th machine is responsible for updating the component $x_i \in \mathbb{R}^{d_i}$, and for the purpose of evaluating the partial gradient $\nabla_i f(\mathbf{x})$ we assume the i-th machine also has access to a local, full model parameter $\mathbf{x}^i \in \mathbb{R}^d$. The last assumption is made only to simplify our presentation; it can be removed for many machine learning problems, see for instance Richtárik and Takáč (2014); Zhou et al. (2016).

We make the following standard assumptions regarding problem (P):

Assumption 1 (Bounded Below) *The function $F = f + g$ is bounded below.*

Assumption 2 (Smooth) *The function f is L-smooth, i.e.,*

$$\forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^d,\ \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|. \tag{10}$$

Assumption 3 (Separable) *The function g is closed and separable, i.e., $g(\mathbf{x}) = \sum_{i=1}^{p} g_i(x_i)$.*

Assumption 1 simply allows us to have a finite minimum value and is usually satisfied in practice. The smoothness assumption is critical in two aspects: (1) It allows us to upper bound f by its quadratic expansion at the current iterate—a standard step in the convergence proof of gradient type algorithms:

$$\forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^d,\ f(\mathbf{x}) \leq f(\mathbf{y}) + \langle \mathbf{x} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \tfrac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2. \tag{11}$$

(2) It allows us to bound the inconsistencies in different machines due to asynchronous updates, see Theorem 4 below. The separable assumption makes model parallelism interesting and feasible, and is satisfied by many popular regularizers. Popular examples include vector norms such as ℓ_0, ℓ_1, $\ell_{1,2}$ (i.e., group norm), ℓ_2^2, elastic net, and matrix norms such as Frobenious norm, etc. We remark that both Assumption 2 and Assumption 3 can be relaxed using techniques in Beck and Teboulle (2012) and Yu et al. (2015), respectively. For brevity we do not pursue these extensions here. Note that we do *not* assume convexity on either f or g, and g need not even be continuous.

We now specify the m-PAPG algorithm for solving (P) under model parallelism and the PAP protocol. The separable assumption on g implies that

$$\mathrm{prox}_g^\eta(\mathbf{x}) = \big(\mathrm{prox}_{g_1}^\eta(x_1),\ \ldots\ ,\ \mathrm{prox}_{g_p}^\eta(x_p)\big). \tag{12}$$

Then, the update on machine i is defined as:

$$x_i \leftarrow \mathrm{prox}_{g_i}^\eta(x_i - \eta \nabla_i f(\mathbf{x}^i)). \tag{13}$$

That is, machine i computes a partial gradient mapping Nesterov (2013) w.r.t. the i-th component using the local component x_i and the local full model $\mathbf{x}^i$. To define the latter, consider a global clock shared by all machines and denote T_i as the set of active clocks when machine i performs an update. Note that the global clock is introduced solely for the purpose of our analysis, and the machines need not maintain it in a practical implementation. Denote $\tau_j^i(t)$ as the iteration of the block model x_j that is accessed by machine i at its t-th iteration. Then, the t-th iteration on machine i can be formally written as:

$$\begin{cases} \forall i,\ x_i(t+1) = \begin{cases} x_i(t), & t \notin T_i \\ \mathrm{prox}_{g_i}^\eta(x_i(t) - \eta \nabla_i f(\mathbf{x}^i(t))), & t \in T_i \end{cases}, \\ \text{(local)} \quad \mathbf{x}^i(t) = (x_1(\tau_1^i(t)),\ \ldots\ ,\ x_p(\tau_p^i(t))), \\ \text{(global)} \quad \mathbf{x}(t) = (x_1(t),\ \ldots\ ,\ x_p(t)). \end{cases} \tag{m-PAPG}$$

That is, machine i only performs its update operator at its active clocks. The local full model $\mathbf{x}^i(t)$ assembles all components from other machines, and is possibly a delayed version of the global model $\mathbf{x}(t)$, which assembles the most up-to-date component in each machine. Note that the global model is introduced for our analysis, and is not accessible in a real implementation. More specifically, $\tau_j^i(t) \leq t$ models the communication delay among machines: when machine i conducts its t-th update it only has access to $x_j(\tau_j^i(t))$, a delayed version of the component $x_j(t)$ that is received by the i-th machine from the j-th machine.

We refer to the above algorithm as m-PAPG (for **m**odel parallel, **P**artially **A**synchronous, **P**roximal **G**radient).

In a practical distributed system, communication among machines is much slower than local computations, and the performance of a *synchronous* system is often bottlenecked at the *slowest* machine, due to the need of synchronization in every step. The delays $\tau_j^i(t)$ and active clocks T_i that we introduced in m-PAPG aim to address such issues. For our convergence proofs, we need the following assumptions:

Assumption 4 (Bounded Delay) $\exists s \in \mathbb{N},\ \forall i, \forall j, \forall t,\ 0 \leq t - \tau_j^i(t) \leq s,\ \tau_i^i(t) \equiv t.$

Assumption 5 (Frequent Update) $\exists s \in \mathbb{N},\ \forall i, \forall t, T_i \cap \{t, t+1, \cdots, t+s\} \neq \emptyset.$

Intuitively, Assumption 4 guarantees the information that machine i gathered from other machines at the t-th iteration are not too obsolete (bounded by at most s clocks apart). The assumption $\tau_i^i(t) \equiv t$ is natural since the i-th worker machine is maintaining x_i hence would always have the latest copy. Assumption 5 requires each machine to update at least once in every $s+1$ iterations, for otherwise some component x_i may not be updated at all. We remark that Assumption 4 and Assumption 5 are very natural and have been widely adopted in previous works Baudet (1978); Bertsekas and Tsitsiklis (1989); Chazan and Miranker (1969); Feyzmahdavian et al. (2014); Tseng (1991). Clearly, when $s = 0$ (i.e., no delay), m-PAPG reduces to the fully synchronous, model parallel proximal gradient algorithm.

Before closing this section, we provide a technical tool to control the inconsistency between the local models $\mathbf{x}^i(t)$ and the global model $\mathbf{x}(t)$. Recall that $(t)_+ = \max\{t, 0\}$ is the positive part of t.

Lemma 4 *Let Assumption 4 hold, then the global model* $\mathbf{x}(t)$ *and the local models* $\{\mathbf{x}^i(t)\}_{i=1}^p$ *satisfy:*

$$\forall i = 1, \cdots, p, \quad \|\mathbf{x}(t) - \mathbf{x}^i(t)\| \leq \sum_{k=(t-s)_+}^{t-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|, \tag{14}$$

$$\|\mathbf{x}^i(t+1) - \mathbf{x}^i(t)\| \leq \sum_{k=(t-s)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|. \tag{15}$$

Proof Indeed, by the definitions in (m-PAPG):

$$\begin{aligned}
\|\mathbf{x}(t) - \mathbf{x}^i(t)\|^2 &= \sum_{j=1}^p \|x_j(t) - x_j(\tau_j^i(t))\|^2 \\
&\leq \sum_{j=1}^p \left(\sum_{k=\tau_j^i(t)}^{t-1} \|x_j(k+1) - x_j(k)\| \right)^2 \\
&\leq \sum_{j=1}^p \left(\sum_{k=(t-s)_+}^{t-1} \|x_j(k+1) - x_j(k)\| \right)^2
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{p} \sum_{k=(t-s)_+}^{t-1} \sum_{k'=(t-s)_+}^{t-1} \|x_j(k+1) - x_j(k)\| \|x_j(k'+1) - x_j(k')\| \\
&= \sum_{k=(t-s)_+}^{t-1} \sum_{k'=(t-s)_+}^{t-1} \sum_{j=1}^{p} \|x_j(k+1) - x_j(k)\| \|x_j(k'+1) - x_j(k')\| \\
&\leq \sum_{k=(t-s)_+}^{t-1} \sum_{k'=(t-s)_+}^{t-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \|\mathbf{x}(k'+1) - \mathbf{x}(k')\| \\
&= \left(\sum_{k=(t-s)_+}^{t-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \right)^2,
\end{aligned}
$$

where the first inequality is due to the triangle inequality; the second inequality is due to Assumption 4; and the last inequality follows from the Cauchy-Schwarz inequality.

Similarly,

$$
\begin{aligned}
\|\mathbf{x}^i(t) - \mathbf{x}^i(t+1)\|^2 &= \sum_{j=1}^{p} \|x_j(\tau_j^i(t)) - x_j(\tau_j^i(t+1))\|^2 \\
&\leq \sum_{j=1}^{p} \left(\sum_{k=\tau_j^i(t)}^{\tau_j^i(t+1)-1} \|x_j(k+1) - x_j(k)\| \right)^2 \\
&\leq \sum_{j=1}^{p} \left(\sum_{k=(t-s)_+}^{t} \|x_j(k+1) - x_j(k)\| \right)^2,
\end{aligned}
$$

and the rest of the proof is completely similar to the previous case. ■

4. Characterizing the limit points

In this section, we characterize the convergence property of the sequences generated by m-PAPG under very general conditions. Recall from Assumption 2 that ∇f is L-Lipschitz continuous. Our first result is as follows:

Theorem 5 *Let Assumptions 1 to 5 hold. If the step size* $\eta \in \left(0, \frac{1}{L(1+2\sqrt{ps})}\right)$, *then the sequence generated by m-PAPG is square summable, i.e.*

$$
\sum_{t=0}^{\infty} \|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 < \infty. \tag{16}
$$

In particular, $\lim_{t\to\infty} \|\mathbf{x}(t+1) - \mathbf{x}(t)\| = 0$ *and* $\lim_{t\to\infty} \|\mathbf{x}(t) - \mathbf{x}^i(t)\| = 0$.

Remark 6 *Our bound on the step size* η *is natural: If* $s = 0$, *i.e., there is no asynchronism then we recover the standard step size rule* $\eta < 1/L$ *(we can increase* η *by another factor of 2, had convexity on* g *been assumed). As staleness* s *increases, we need a smaller step size to*

"damp" the system to still ensure convergence. The factor $\sqrt{p}$ is another measurement of the degree of "dependency" among worker machines: Indeed, we can reduce $\sqrt{p}$ to $\sqrt{\sum_i L_i^2}/L$, where L_i is the Lipschitz constant of $\nabla_i f$ (cf. (21)).

Proof The last claim follows immediately from eq. (16) and eq. (14), so we only need to prove (16).

Consider machine i and any $t \in T_i$. Combining eq. (13) with eq. (m-PAPG) gives

$$x_i(t+1) = \operatorname{prox}_{g_i}^{\eta}(x_i(t) - \eta \nabla_i f(\mathbf{x}^i(t))). \tag{17}$$

Then, from Definition 3 of the proximal map we have for all $z \in \mathbb{R}^{d_i}$:

$$\begin{aligned} g_i(x_i(t+1)) + \frac{1}{2\eta}\|x_i(t+1) - x_i(t) + \eta \nabla_i f(\mathbf{x}^i(t))\|^2 \\ \leq g_i(z) + \frac{1}{2\eta}\left\|z - x_i(t) + \eta \nabla_i f(\mathbf{x}^i(t))\right\|^2. \end{aligned} \tag{18}$$

Set $z = x_i(t)$ and simplify, we obtain:

$$\begin{aligned} g_i(x_i(t+1)) - g_i(x_i(t)) \\ \leq -\frac{1}{2\eta}\|x_i(t+1) - x_i(t)\|^2 - \left\langle \nabla_i f(\mathbf{x}^i(t)), x_i(t+1) - x_i(t)\right\rangle. \end{aligned} \tag{19}$$

Note that if $t \notin T_i$, then $x_i(t+1) = x_i(t)$ and eq. (19) still holds. On the other hand, Assumption 2 implies that for all t (cf. (11)):

$$f(\mathbf{x}(t+1)) - f(\mathbf{x}(t)) \leq \langle \mathbf{x}(t+1) - \mathbf{x}(t), \nabla f(\mathbf{x}(t))\rangle + \frac{L}{2}\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2. \tag{20}$$

Adding up eq. (20) and eq. (19) (for all i) and recall $F = f + \sum_i g_i$, we have

$$\begin{aligned} F(\mathbf{x}(t+1)) - F(\mathbf{x}(t)) - \tfrac{1}{2}(L - 1/\eta)\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 \\ &\leq \sum_{i=1}^{p} \left\langle x_i(t+1) - x_i(t), \nabla_i f(\mathbf{x}(t)) - \nabla_i f(\mathbf{x}^i(t))\right\rangle \\ &\leq \sum_{i=1}^{p} \|x_i(t+1) - x_i(t)\| \cdot \|\nabla_i f(\mathbf{x}(t)) - \nabla_i f(\mathbf{x}^i(t))\| \\ &\overset{\text{(i)}}{\leq} \sum_{i=1}^{p} \|x_i(t+1) - x_i(t)\| \cdot L\|\mathbf{x}(t) - \mathbf{x}^i(t)\| \qquad (21) \\ &\overset{\text{(ii)}}{\leq} L \cdot \sum_{i=1}^{p} \|x_i(t+1) - x_i(t)\| \cdot \sum_{k=(t-s)_+}^{t-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \\ &\overset{\text{(iii)}}{\leq} \sqrt{p}L\|\mathbf{x}(t+1) - \mathbf{x}(t)\| \cdot \sum_{k=(t-s)_+}^{t-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \qquad (22) \\ &\overset{\text{(iv)}}{\leq} \frac{\sqrt{p}L}{2} \sum_{k=(t-s)_+}^{t-1} \left[\|\mathbf{x}(k+1) - \mathbf{x}(k)\|^2 + \|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2\right] \end{aligned}$$

$$\leq \frac{\sqrt{p}Ls}{2}\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2 + \frac{\sqrt{p}L}{2}\sum_{k=(t-s)_+}^{t-1}\|\mathbf{x}(k+1)-\mathbf{x}(k)\|^2, \quad (23)$$

where (i) is due to the L-Lipschitz continuity of ∇f, (ii) follows from eq. (14), (iii) is the Cauchy-Schwarz inequality, and (iv) follows from the elementary inequality $ab \leq \frac{a^2+b^2}{2}$. Summing the above inequality over t from 0 to $m-1$ and rearranging we obtain

$$\begin{aligned} F(\mathbf{x}(m)) - F(\mathbf{x}(0)) &\leq \frac{1}{2}(L + \sqrt{p}Ls - 1/\eta)\sum_{t=0}^{m-1}\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2 \\ &\quad + \frac{L}{2}\sum_{t=0}^{m-1}\sum_{k=(t-s)_+}^{t-1}\|\mathbf{x}(k+1)-\mathbf{x}(k)\|^2 \\ &\leq \frac{1}{2}(L + 2\sqrt{p}Ls - 1/\eta)\sum_{t=0}^{m-1}\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2. \end{aligned}$$

Therefore, if we choose $0 < \eta < \frac{1}{L(1+2\sqrt{p}s)}$, then let $m \to \infty$ we deduce

$$\sum_{t=0}^{\infty}\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2 \leq \frac{2}{1/\eta - L - 2\sqrt{p}Ls}[F(\mathbf{x}(0)) - \inf_{\mathbf{z}} F(\mathbf{z})]. \quad (24)$$

By Assumption 1, F is bounded from below, hence the right-hand side is finite. ■

The first assertion of the above theorem states that the global sequence $\mathbf{x}(t)$ has square summable successive differences, while the second assertion implies that both the successive difference of the global sequence and the inconsistency between the local sequences and the global sequence diminish as the number of iterations grows. These two conclusions provide a prelimenary stability guarantee for m-PAPG.

Next, we prove that the limit points (if exist) of the sequences $\mathbf{x}(t)$ and $\mathbf{x}^i(t), i = 1, \ldots, p$ coincide, and they are critical points of F. Recall that the set of critical points of the function F is denoted as $\operatorname{crit} F$.

Theorem 7 *Consider the same setting as in Theorem 5. Then, the sequences* $\{\mathbf{x}(t)\}$ *and* $\{\mathbf{x}^i(t)\}, i = 1, \ldots, p$, *generated by m-PAPG share the same set of limit points, which is a subset of* $\operatorname{crit} F$.

Proof It is clear from Theorem 5 that $\{\mathbf{x}(t)\}$ and $\{\mathbf{x}^i(t)\}, i = 1, \ldots, p$, share the same set of limit points, and we need to show that any limit point of $\{\mathbf{x}(t)\}$ is also a critical point of F.

Let $\mathbf{x}^*$ be a limit point of $\{\mathbf{x}(t)\}$. By Theorem 1 it suffices to exhibit a sequence $\mathbf{x}(k)$ satisfying[1]

$$\mathbf{x}(k) \to \mathbf{x}^*,\ F(\mathbf{x}(k)) \to F(\mathbf{x}^*),\ \mathbf{0} \leftarrow \mathbf{u}(k) \in \partial F(\mathbf{x}(k)). \quad (25)$$

1. Technically, from Theorem 1 we should have the Frechét subdifferential $\hat{\partial}F$ in eq. (25), however, a standard argument allows us to use the more convenient subdifferential (Rockafellar and Wets, 1997, Proposition 8.7).

Let us first construct the subgradient sequence $\mathbf{u}(k)$. Consider machine i and any $\hat{t} \in T_i$, the optimality condition of eq. (17) gives

$$u_i(\hat{t}+1) := -\tfrac{1}{\eta}\left[x_i(\hat{t}+1) - x_i(\hat{t}) + \eta \nabla_i f(\mathbf{x}^i(\hat{t}))\right] \in \partial g_i(x_i(\hat{t}+1)). \tag{26}$$

It then follows that

$$\begin{aligned}
&\|u_i(\hat{t}+1) + \nabla_i f(\mathbf{x}(\hat{t}+1))\| \\
&\quad \le \|u_i(\hat{t}+1) + \nabla_i f(\mathbf{x}(\hat{t}))\| + \|\nabla_i f(\mathbf{x}(\hat{t}+1)) - \nabla_i f(\mathbf{x}(\hat{t}))\| \\
&\quad \overset{\text{(i)}}{\le} \left\|\tfrac{1}{\eta}\left[x_i(\hat{t}+1) - x_i(\hat{t})\right] + \nabla_i f(\mathbf{x}^i(\hat{t})) - \nabla_i f(\mathbf{x}(\hat{t}))\right\| + L\|\mathbf{x}(\hat{t}+1) - \mathbf{x}(\hat{t})\| \\
&\quad \overset{\text{(ii)}}{\le} \tfrac{1}{\eta}\|x_i(\hat{t}+1) - x_i(\hat{t})\| + L\|\mathbf{x}^i(\hat{t}) - \mathbf{x}(\hat{t})\| + L\|\mathbf{x}(\hat{t}+1) - \mathbf{x}(\hat{t})\| \\
&\quad \overset{\text{(iii)}}{\le} \tfrac{1}{\eta}\|x_i(\hat{t}+1) - x_i(\hat{t})\| + L\sum_{k=(\hat{t}-s)_+}^{\hat{t}} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|,
\end{aligned} \tag{27}$$

where (i) and (ii) are due to the L-Lipschitz continuity of ∇f, and (iii) follows from eq. (14). Next, consider any other $t \notin T_i$ and $t \ge s$, we denote $\hat{t}$ as the *largest* element in the set $\{k \le t : k \in T_i\}$. By Assumption 5 $\hat{t}$ always exists and $t - \hat{t} \le s$. Since no update is performed on machine i at any clock in $[\hat{t}+1, t]$, we have $x_i(t+1) = x_i(\hat{t}+1)$. Thus, we can choose $u_i(t+1) = u_i(\hat{t}+1) \in \partial g_i(x_i(\hat{t}+1)) = \partial g_i(x_i(t+1))$, and obtain

$$\begin{aligned}
&\|u_i(t+1) + \nabla_i f(\mathbf{x}(t+1)) - u_i(\hat{t}+1) - \nabla_i f(\mathbf{x}(\hat{t}+1))\| \qquad (28)\\
&\qquad = \|\nabla_i f(\mathbf{x}(t+1)) - \nabla_i f(\mathbf{x}(\hat{t}+1))\| \\
&\qquad \le \sum_{k=\hat{t}+1}^{t} \|\nabla_i f(\mathbf{x}(k+1)) - \nabla_i f(\mathbf{x}(k))\| \\
&\qquad \le \sum_{k=(t-s+1)_+}^{t} \|\nabla_i f(\mathbf{x}(k+1)) - \nabla_i f(\mathbf{x}(k))\| \\
&\qquad \le \sum_{k=(t-s+1)_+}^{t} L\|\mathbf{x}(k+1) - \mathbf{x}(k)\|. \qquad (29)
\end{aligned}$$

Combining the two cases in eq. (27) and eq. (29), we have for all t and all i:

$$\begin{aligned}
\|u_i(t+1) + \nabla_i f(\mathbf{x}(t+1))\| &\le \tfrac{1}{\eta}\|x_i(\hat{t}+1) - x_i(\hat{t})\| + L\sum_{k=(\hat{t}-s)_+}^{\hat{t}} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \\
&\quad + L\sum_{k=(t-s+1)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \\
&\le (\tfrac{1}{\eta} + 2L)\sum_{k=(t-2s)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|,
\end{aligned}$$

where the last inequality uses the fact that $t-s \le \hat{t} \le t$. Observing that the right hand side of the above inequality does not depend on i, we can sum the square of the above inequality over i and further conclude that

$$\|\mathbf{u}(t+1) + \nabla f(\mathbf{x}(t+1))\| \le \sqrt{p}(\tfrac{1}{\eta} + 2L) \sum_{k=(t-2s)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|, \tag{30}$$

where $\mathbf{u}(t+1) = (u_1(t+1),\ \dots\ ,\ u_p(t+1)) \in \partial g(\mathbf{x}(t+1))$. Therefore, by eq. (30) and Theorem 5 we deduce

$$\lim_{t\to\infty} \mathrm{dist}_{\partial F(\mathbf{x}(t+1))}(\mathbf{0}) \le \lim_{t\to\infty} \|\mathbf{u}(t+1) + \nabla f(\mathbf{x}(t+1))\| = 0. \tag{31}$$

Recall that $\mathbf{x}^*$ is a limit point of $\{\mathbf{x}(t)\}$, thus there exists a subsequence $\mathbf{x}(t_m) \to \mathbf{x}^*$. Next we verify the function value convergence in eq. (25). The challenge here is that the component function g is only closed, hence may not be continuous. For any $t \in T_i$, applying eq. (18) with $z = x_i^*$ and rearranging gives

$$\begin{aligned} g_i(x_i(t+1)) &\le g_i(x_i^*) + \frac{1}{2\eta}\|x_i^* - x_i(t)\|^2 - \frac{1}{2\eta}\|x_i(t+1) - x_i(t)\|^2 \\ &\quad + \langle x_i^* - x_i(t+1), \nabla_i f(\mathbf{x}^i(t))\rangle \\ &= g_i(x_i^*) + \frac{1}{2\eta}\|x_i^* - x_i(t)\|^2 - \frac{1}{2\eta}\|x_i(t+1) - x_i(t)\|^2 \\ &\quad + \langle x_i^* - x_i(t+1), \nabla_i f(\mathbf{x}^*)\rangle + \langle x_i^* - x_i(t+1), \nabla_i f(\mathbf{x}^i(t)) - \nabla_i f(\mathbf{x}^*)\rangle. \end{aligned} \tag{32}$$

We note that the above inequality holds only for the iterations $t \in T_i$. Next, observe that $\lim_{m\to\infty} \|\mathbf{x}(t_m) - \mathbf{x}^*\| = 0$. Since $\lim_{t\to\infty} \|\mathbf{x}(t+1) - \mathbf{x}(t)\| = 0$, we further conclude that

$$\lim_{m\to\infty} \max_{t\in[t_m - s, t_m + s]\cap T_i} \|\mathbf{x}(t) - \mathbf{x}^*\| = 0. \tag{33}$$

Moreover, note that $\lim_{t\to\infty} \|\mathbf{x}(t) - \mathbf{x}^i(t)\| = 0$. Then, the above equation further implies that

$$\begin{aligned} &\lim_{m\to\infty} \max_{t\in[t_m - s, t_m + s]\cap T_i} \|\nabla_i f(\mathbf{x}^i(t)) - \nabla_i f(\mathbf{x}^*)\| \\ &\le L \lim_{m\to\infty} \max_{t\in[t_m - s, t_m + s]\cap T_i} \|\mathbf{x}^* - \mathbf{x}^i(t)\| \\ &\le L \lim_{m\to\infty} \max_{t\in[t_m - s, t_m + s]\cap T_i} \left[\|\mathbf{x}^* - \mathbf{x}(t)\| + \|\mathbf{x}(t) - \mathbf{x}^i(t)\|\right] = 0. \end{aligned} \tag{34}$$

By Assumption 5, $[t_m - s, t_m + s] \cap T_i \neq \emptyset$ for all i. We can now take the limsup on both sides of eq. (32) and utilize eqs. (33) and (34) to obtain that

$$\limsup_{m\to\infty} \max_{t\in[t_m - s, t_m + s]\cap T_i} g_i(x_i(t+1)) \le g_i(x_i^*). \tag{35}$$

Denote $\hat{t}_m \in T_i$ as the largest element such that $\hat{t}_m \le t_m$. Note that $t_m - s \le \hat{t}_m$ due to the constraint on the maximum delay. It then follows that

$$\max_{t\in[t_m, t_m + s]} g_i(x_i(t+1)) = \max_{t\in[\hat{t}_m, t_m + s]\cap T_i} g_i(x_i(t+1)) \le \max_{t\in[t_m - s, t_m + s]\cap T_i} g_i(x_i(t+1)),$$

where the first equality is due to the fact that no update is performed during $[\hat{t}_m, t_m]$ and machine i updates only at its active clocks T_i, and the second inequality uses the fact that $\hat{t}_m \geq t_m - s$. Hence, we further obtain from (35) that

$$\limsup_{m\to\infty} \max_{t\in[t_m,t_m+s]} g_i(x_i(t+1)) \leq g_i(x_i^*). \tag{36}$$

To complete the proof, choose any $k_m \in [t_m, t_m + s]$. Since $\mathbf{x}(t_m) \to \mathbf{x}^*$, Theorem 5 implies that

$$\mathbf{x}(k_m) \to \mathbf{x}^*. \tag{37}$$

From eq. (36) we know for all i, $\limsup_{m\to\infty} g_i(x_i(k_m)) \leq g_i(x_i^*)$. On the other hand, it follows from the closedness of the function g_i (cf. Assumption 3) that $\liminf_{m\to\infty} g_i(x_i(k_m)) \geq g_i(x_i^*)$, thus in fact $\lim_{m\to\infty} g_i(x_i(k_m)) = g_i(x_i^*)$. Since f is continuous, we know

$$\lim_{m\to\infty} F(\mathbf{x}(k_m)) = \lim_{m\to\infty} f(\mathbf{x}(k_m)) + \sum_i g_i(x_i(k_m)) = F(\mathbf{x}^*). \tag{38}$$

Combining eq. (31), eq. (37) and eq. (38) we know from Theorem 1 that $\mathbf{x}^* \in \operatorname{crit} F$. ■

Theorem 7 further justifies m-PAPG by showing that any limit point it produces is necessarily a critical point. Of course, for convex functions any critical point is a global minimizer. The closest result to Theorem 5 and Theorem 7 we are aware of is (Bertsekas and Tsitsiklis, 1989, Proposition 7.5.3), where essentially the same conclusion was reached but under the much more restrictive assumption that g is an indicator function of a product *convex* set. Thus, our result is new even when g is a convex function such as the ℓ_1 norm that is widely used to promote sparsity. Furthermore, we allow g to be any closed separable function (convex or not), covering the many recent nonconvex regularization functions in machine learning and statistics (see e.g. Fan and Li (2001); Mazumder et al. (2011); Zhang (2010); Zhang and Zhang (2012)). We also note that the proof of Theorem 7 (for nonconvex g) involves significantly new ideas beyond those of Bertsekas and Tsitsiklis (1989).

We note that the existence of limit points can be guaranteed, for instance, if $\{\mathbf{x}(t)\}$ is bounded or the sublevel set $\{\mathbf{x} \mid F(\mathbf{x}) \leq \alpha\}$ is bounded for all $\alpha \in \mathbb{R}$. However, we have yet to prove that the sequence $\{\mathbf{x}(t)\}$ generated by m-PAPG does converge to one of the critical points, and we fill this gap under two complementary sets of assumptions on the objective function in Sections 5 and 6, respectively.

5. Convergence under Error Bound

In this section we prove that the global sequence $\{\mathbf{x}(t)\}$ produced by m-PAPG converges periodically linearly to a global minimizer, by assuming an error bound condition on the objective function in (P) and a convexity assumption that serves to simplify the presentation:

Assumption 6 (Convex) *The functions f and g in* (P) *are convex.*

Note that for convex functions g the proximal mapping prox_g^η is single valued for any $\eta > 0$. The error bound condition we need is as follows:

Assumption 7 (Error Bound) *For every $\alpha > 0$, there exist $\delta, \kappa > 0$ such that for all $\mathbf{x} \in \mathbb{R}^d$ with $f(\mathbf{x}) \le \alpha$ and $\|\mathbf{x} - \text{prox}_g(\mathbf{x} - \nabla f(\mathbf{x}))\| \le \delta$,*

$$\text{dist}_{\text{crit}\, F}(\mathbf{x}) \le \kappa \|\mathbf{x} - \text{prox}_g(\mathbf{x} - \nabla f(\mathbf{x}))\|, \tag{39}$$

where recall that crit F *is the set of critical points of F.*

Equation (39) is a proximal extension of the Luo-Tseng error bound Luo and Tseng (1993) where g is the indicator function of a closed convex set. A prototypic convex function F satisfying (39) is the following:

$$F(\mathbf{x}) = f(A\mathbf{x}) + g(\mathbf{x}), \tag{40}$$

where f is strongly convex (i.e., $f - \frac{\mu}{2}\|\cdot\|^2$ is convex for some $\mu > 0$), A is a linear map, and g is either an indicator function of a convex set Luo and Tseng (1993) or the ℓ_p norm for $p \in [1, 2] \cup \{\infty\}$ Zhou et al. (2015). Many machine learning formulations such as Lasso and sparse logistic regression fit into this form. In fact, for convex functions F taking such form, the error bound condition in eq. (39) is recently shown to be equivalent to the following conditions Drusvyatskiy and Lewis (2016); Zhang (2016):

$$\begin{aligned} \text{Restricted strong convexity}: &\ \langle \mathbf{x} - \text{prox}_g(\mathbf{x}), \mathbf{x} - \text{proj}_{\text{crit}\, F}(\mathbf{x}) \rangle \ge \mu \cdot \text{dist}^2_{\text{crit}\, F}(\mathbf{x}), \\ \text{Quadratic growth}: &\ F(\mathbf{x}) - F^* \ge \mu \cdot \text{dist}^2_{\text{crit}\, F}(\mathbf{x}), \end{aligned}$$

where F^* is the minimum value of F and $\mu > 0$ is a constant. In general, the error bound condition in eq. (39) is not exclusive to convex functions. For instance, it holds for $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$ and any function g that has a unique global minimizer at 0 (such as the cardinality function $g(\mathbf{x}) = \|\mathbf{x}\|_0$). However, it is often quite challenging to establish the error bound condition for a large family of nonconvex functions.

We define the following nonnegative quantities that measure the progress of m-PAPG:

$$A(t) := F(\mathbf{x}(t)) - F^*, \ F^* := \inf_{\mathbf{x}} F(\mathbf{x}), \tag{41}$$

$$B(t) := \sum_{k=(t-s-1)_+}^{t-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|^2, \tag{42}$$

In the following key lemma we relate the gap quantities defined above inductively.

Lemma 8 *Let Assumptions 1 to 7 hold. Then, we have*

$$\begin{aligned} &A(t+s+1) \le A(t) - \tfrac{1}{2}(\tfrac{1}{\eta} - L - 2sL\sqrt{p})B(t+s+1) + \tfrac{1}{2}sL\sqrt{p}B(t) \\ &0 \le A(t+s+1) \le a_\eta B(t+s+1) + bB(t), \end{aligned}$$

where a_η and b are given in (53) *below.*

Proof The first inequality is obtained by summing the inequality eq. (23) over $t, t+1, \cdots, t+s$. So we need only prove the second inequality.

Let us introduce some notations to simplify the proof. For each machine i let t_i be the largest clock in $[t, t+s] \cap T_i$, and denote

$$\mathbf{z} = (x_1(t_1), \ \dots \ , x_p(t_p)) \tag{43}$$
$$\mathbf{z}^+ = (x_1(t_1+1), \ \dots \ , x_p(t_p+1)) = (x_1(t+s+1), \ \dots \ , x_p(t+s+1)), \tag{44}$$

where the last equality is due to the maximality of each t_i. From the optimality condition of the proximal map $z_i^+ = \mathrm{prox}_{g_i}^{\eta}(z_i - \eta \nabla_i f(\mathbf{x}^i(t_i)))$ we deduce

$$\eta^{-1}(z_i - z_i^+) - \nabla_i f(\mathbf{x}^i(t_i)) \in \partial g_i(z_i^+). \tag{45}$$

Since the gradient of f is L-Lipschitz continuous and the function g is convex, we obtain

$$f(\mathbf{z}^+) - f(\bar{\mathbf{z}}) \leq \sum_{i=1}^{p} \langle z_i^+ - \bar{z}_i, \nabla_i f(\bar{\mathbf{z}}) \rangle + \frac{L}{2} \|\mathbf{z}^+ - \bar{\mathbf{z}}\|^2,$$
$$g(\mathbf{z}^+) - g(\bar{\mathbf{z}}) \leq \sum_{i=1}^{p} \langle z_i^+ - \bar{z}_i, \eta^{-1}(z_i - z_i^+) - \nabla_i f(\mathbf{x}^i(t_i)) \rangle,$$

where we define $\bar{\mathbf{z}} := \mathrm{proj}_{\mathrm{crit}\, F}(\mathbf{z})$, i.e., the projection of $\mathbf{z}$ onto the set of critical points of F, and the last inequality follows from eq. (45). Adding up the above two inequalities we obtain

$$\begin{aligned}
F(\mathbf{z}^+) - F^* - \tfrac{L}{2}\|\mathbf{z}^+ - \bar{\mathbf{z}}\|^2 &\leq \sum_{i=1}^{p} \langle z_i^+ - \bar{z}_i, \nabla_i f(\bar{\mathbf{z}}) + \eta^{-1}(z_i - z_i^+) - \nabla_i f(\mathbf{x}^i(t_i)) \rangle \\
&\overset{\text{(i)}}{\leq} \sum_{i=1}^{p} [\|z_i^+ - z_i\| + \|z_i - \bar{z}_i\|][\|\nabla_i f(\mathbf{x}^i(t_i)) - \nabla_i f(\bar{\mathbf{z}})\| + \eta^{-1}\|z_i - z_i^+\|] \\
&\overset{\text{(ii)}}{\leq} \sum_{i=1}^{p} 4 \left[\|z_i^+ - z_i\|^2 + \|z_i - \bar{z}_i\|^2 + \eta^{-2}\|z_i^+ - z_i\|^2 + \|\nabla_i f(\mathbf{x}^i(t_i)) - \nabla_i f(\bar{\mathbf{z}})\|^2 \right] \\
&\leq 4 \left[\|\bar{\mathbf{z}} - \mathbf{z}\|^2 + (1+\eta^{-2})\|\mathbf{z}^+ - \mathbf{z}\|^2 + \sum_{i=1}^{p} L^2 \|\mathbf{x}^i(t_i) - \bar{\mathbf{z}}\|^2 \right],
\end{aligned}$$

where (i) is due to the Cauchy-Schwarz inequality and the triangle inequality, (ii) is due to the elementary inequality $(a+b)(c+d) \leq 4(a^2+b^2+c^2+d^2)$, and the last inequality is due to the L-Lipschitz continuity of ∇f. Using again the triangle inequality we obtain from the above inequality that

$$\begin{aligned}
F(\mathbf{z}^+) - F^* &\leq (L+4)\|\bar{\mathbf{z}} - \mathbf{z}\|^2 + (L+4+\tfrac{4}{\eta^2})\|\mathbf{z}^+ - \mathbf{z}\|^2 + 4L^2 \sum_{i=1}^{p} \|\mathbf{x}^i(t_i) - \bar{\mathbf{z}}\|^2 \\
&\overset{\text{(i)}}{=} (L+4)\|\bar{\mathbf{z}} - \mathbf{z}\|^2 + \sum_{i=1}^{p} [(L+4+\tfrac{4}{\eta^2})\|x_i(t_i+1) - x_i(t_i)\|^2 + 4L^2 \|\mathbf{x}^i(t_i) - \bar{\mathbf{z}}\|^2], \\
&\overset{\text{(ii)}}{\leq} (L+4)\|\bar{\mathbf{z}} - \mathbf{z}\|^2 + (L+4+\tfrac{4}{\eta^2})B(t+s+1) + 4L^2 \sum_{i=1}^{p} \|\mathbf{x}^i(t_i) - \bar{\mathbf{z}}\|^2,
\end{aligned} \tag{46}$$

$$\leq (L+4+8L^2p)\|\bar{\mathbf{z}}-\mathbf{z}\|^2 + (L+4+\tfrac{4}{\eta^2})B(t+s+1) + 8L^2\sum_{i=1}^{p}\|\mathbf{x}^i(t_i)-\mathbf{z}\|^2, \tag{47}$$

where (i) is due to our definition of $\mathbf{z}$ and $\mathbf{z}^+$ in (43) and (44), and (ii) is due to the fact that $t_i \in [t, t+s]$ for all i.

We next bound the terms $\|\bar{\mathbf{z}}-\mathbf{z}\|^2$ and $\|\mathbf{x}^i(t_i)-\mathbf{z}\|^2$. We recall that $\mathbf{x}^i(t_i)$ corresponds to the local model on machine i at the iteration t_i. Since $t_i \in T_i$, the update rule for the i-th machine implies that

$$\begin{aligned}
\|x_i(t_i+1) - x_i(t_i)\| &= \|\mathrm{prox}_{g_i}^{\eta}(x_i(t_i) - \eta\nabla_i f(\mathbf{x}^i(t_i))) - x_i(t_i)\| \\
&\geq \|\mathrm{prox}_{g_i}^{\eta}(x_i(t_i) - \eta\nabla_i f(\mathbf{z})) - x_i(t_i)\| \\
&\quad - \|\mathrm{prox}_{g_i}^{\eta}(x_i(t_i) - \eta\nabla_i f(\mathbf{x}^i(t_i))) - \mathrm{prox}_{g_i}^{\eta}(x_i(t_i) - \eta\nabla_i f(\mathbf{z}))\| \\
&\overset{\text{(i)}}{\geq} \|\mathrm{prox}_{g_i}^{\eta}(x_i(t_i) - \eta\nabla_i f(\mathbf{z})) - x_i(t_i)\| - \eta L\|\mathbf{z} - \mathbf{x}^i(t_i)\|,
\end{aligned}$$

where (i) follows from the non-expansiveness of prox_g^{η} (recall that g is convex) and the L-Lipschitz continuity of ∇f. Rearranging the above inequality and summing over all i, we obtain

$$\begin{aligned}
\|\mathrm{prox}_g^{\eta}(\mathbf{z} - \eta\nabla f(\mathbf{z})) - \mathbf{z}\|^2 &\leq \sum_{i=1}^{p}\Big[\|x_i(t_i+1) - x_i(t_i)\| + \eta L\|\mathbf{z} - \mathbf{x}^i(t_i)\|\Big]^2 \\
&\leq 2\sum_{i=1}^{p}\Big[\|x_i(t_i+1) - x_i(t_i)\|^2 + \eta^2 L^2\|\mathbf{z} - \mathbf{x}^i(t_i)\|^2\Big]. \qquad (48)
\end{aligned}$$

The last term $\|\mathbf{z} - \mathbf{x}^i(t_i)\|^2$ can be further bounded as follows:

$$\begin{aligned}
\|\mathbf{z} - \mathbf{x}^i(t_i)\|^2 &= \sum_{j=1}^{p}\|x_j(t_j) - x_j(\tau_j^i(t_i))\|^2 \\
&= \sum_{j=1}^{p}\Big\|\sum_{k=\min\{t_j,\tau_j^i(t_i)\}}^{\max\{t_j,\tau_j^i(t_i)\}-1} x_j(k+1) - x_j(k)\Big\|^2 \\
&\leq \sum_{j=1}^{p}\Big[\sum_{k=\min\{t_j,\tau_j^i(t_i)\}}^{\max\{t_j,\tau_j^i(t_i)\}-1}\|x_j(k+1) - x_j(k)\|\Big]^2 \\
&\overset{\text{(i)}}{\leq} \sum_{j=1}^{p} 2s\sum_{k=t-s}^{t+s-1}\|x_j(k+1) - x_j(k)\|^2 \\
&= 2s\sum_{k=t-s}^{t+s-1}\|\mathbf{x}(k+1) - \mathbf{x}(k)\|^2 \\
&\leq 2s[B(t) + B(t+s+1)], \qquad (49)
\end{aligned}$$

where (i) is due to the fact that $t_j \in [t, t+s]$ and $\tau_j^i(t_i) \in [t-s, t+s]$. Combining (48) and (49) we obtain

$$\|\mathrm{prox}_g^{\eta}(\mathbf{z} - \eta\nabla f(\mathbf{z})) - \mathbf{z}\|^2 \leq 2B(t+s+1) + 4ps\eta^2L^2[B(t) + B(t+s+1)]. \tag{50}$$

Thanks to Theorem 5, we know for t sufficiently large, $\|\text{prox}_g^\eta(\mathbf{z}-\eta\nabla f(\mathbf{z}))-\mathbf{z}\| \le \eta\delta$. Since the function $\eta \mapsto \frac{1}{\eta}\|\text{prox}_g^\eta(\mathbf{z}-\eta\nabla f(\mathbf{z}))-\mathbf{z}\|$ is monotonically decreasing Sra (2012), we can apply the error bound condition in Assumption 7 for $\eta < 1$ and t sufficiently large, and obtain

$$\|\bar{\mathbf{z}}-\mathbf{z}\|^2 \le \kappa\|\mathbf{z}-\text{prox}_g(\mathbf{z}-\nabla f(\mathbf{z}))\|^2 \le \kappa\eta^{-2}\|\mathbf{z}-\text{prox}_g^\eta(\mathbf{z}-\eta\nabla f(\mathbf{z}))\|^2. \tag{51}$$

Finally, combining (46), (49), (50) and (51) we arrive at:

$$\begin{aligned} F(\mathbf{x}(t+s+1)) - F^* &= F(\mathbf{z}^+) - F^* \\ &\le (L+4+8L^2p)\|\bar{\mathbf{z}}-\mathbf{z}\|^2 + (L+4+\tfrac{4}{\eta^2})B(t+s+1) + 8L^2\sum_{i=1}^p \|\mathbf{x}^i(t_i)-\mathbf{z}\|^2, \\ &\le a_\eta B(t+s+1) + bB(t), \end{aligned} \tag{52}$$

where the coefficients are

$$a_\eta = L + 4 + 16psL^2 + 4ps\kappa L^2(L+4+8L^2p) + \tfrac{2}{\eta^2}(2+4\kappa+\kappa L), \tag{53}$$

$$b = 16psL^2 + 4ps\kappa L^2(L+4+8L^2p). \tag{54}$$

■

Theorem 8 improves the analysis of Tseng (1991) in three aspects: (1) it is shorter and simpler; (2) it allows any convex function g; and (3) the leading coefficient for $B(t)$ is reduced from $O(1/\eta)$ to $O(1)$. The two recursive relations in Lemma 8, as shown in (Tseng, 1991, Lemma 4.5), easily imply the following convergence guarantee.

Theorem 9 *Let Assumptions 1 to 7 hold. Then, there exists some $\eta_0 > 0$ such that if $0 < \eta < \eta_0$, then the sequences $\{A(t), B(t)\}$ generated by m-PAPG satisfy for all $r = 0, 1, 2, \cdots$*

$$A(r(s+1)) \le C_1(1-\gamma\eta)^r,\ B(r(s+1)) \le C_2(1-\gamma\eta)^r, \tag{55}$$

where $C_1, C_2, \gamma < 1/\eta$ are positive constants.

Hence, the gaps $A(t)$ and $B(t)$ that measure the progress of m-PAPG decrease by a constant factor $(1-\gamma\eta)$ for every $s+1$ steps, which makes intuitive sense since in the worst case each worker machine only performs one update in every $s+1$ steps. In other words, $(s+1)$ is the natural time scale for measuring progress here. Note that since $\|\mathbf{x}(t+s+1)-\mathbf{x}(t)\|^2 \le (s+1)B(t+s+1)$, it follows easily that the global sequence $\mathbf{x}(t)$ and consequently also the local sequences $\{\mathbf{x}^i(t)\}$ all converge to the same limit point in $\text{crit}\, F$ at a $(s+1)$-periodically linear rate.

6. Convergence with KŁ inequality

The error bound condition considered in the previous section is not easy to verify in general. It has been discovered recently that the error bound condition is equivalent to other notions in optimization that can be verified in alternative ways Drusvyatskiy and Lewis (2016);

Zhang (2016), see e.g. (40). However, for nonconvex functions, sometimes even the simple ones, it remains a challenging task to verify if the error bound condition holds. This failure motivates us to investigate another property, the Kurdyka-Łojasiewicz (KŁ) inequality, that has been shown to be quite effective in dealing with nonconvex functions.

Definition 10 (KŁ property, (Bolte et al., 2014, Lemma 6)) *Let $\Omega \subset \mathrm{dom} h$ be a compact set on which the function h is a constant. We say that h satisfies the KŁ property if there exist $\varepsilon, \lambda > 0$ such that for all $\bar{\mathbf{x}} \in \Omega$ and all $\mathbf{x} \in \{\mathbf{z} \in \mathbb{R}^d : \mathrm{dist}_\Omega(\mathbf{z}) < \varepsilon\} \cap [\mathbf{z} : h(\bar{\mathbf{x}}) < h(\mathbf{z}) < h(\bar{\mathbf{x}}) + \lambda]$, it holds that*

$$\varphi'(h(\mathbf{x}) - h(\bar{\mathbf{x}})) \cdot \mathrm{dist}_{\partial h(\mathbf{x})}(\mathbf{0}) \geq 1, \tag{56}$$

where the function $\varphi : [0, \lambda) \to \mathbb{R}_+, 0 \mapsto 0$, is continuous, concave, and has continuous and positive derivative φ' on $(0, \lambda)$.

The KŁ inequality in eq. (56) is an important tool to bound the trajectory length of a dynamical system (see Bolte et al. (2010); Kurdyka (1998) and the references therein for some historic developments). It has recently been used to analyze discrete-time algorithms in Absil et al. (2005) and proximal algorithms in Attouch and Bolte (2009); Attouch et al. (2010); Bolte et al. (2014). As we shall see, the function φ will serve as a Lyapunov potential function. Quite conveniently, most practical functions, in particular, the quasi-norm $\|\cdot\|_p$ for positive rational p, as well as convex functions with certain growth conditions, are KŁ. For a more detailed discussion of KŁ functions, including many familiar examples, see (Bolte et al., 2014, Section 5) and (Attouch et al., 2010, Section 4).

Following the recipe in Bolte et al. (2014), we need the following assumption to guarantee the algorithm is making *sufficient* progress:

Assumption 8 (Sufficient decrease) *There exists $\alpha > 0$ such that for all large t,*

$$F(\mathbf{x}(t+1)) \leq F(\mathbf{x}(t)) - \alpha \|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2. \tag{57}$$

The sufficient decrease assumption is automatically satisfied in many descent algorithms, e.g., the proximal gradient algorithm. However, in the partially asynchronous parallel (PAP) setting, it is highly nontrivial to satisfy the sufficient decrease assumption because of the complication due to communication delays and update skips. Note also that none of the worker machines actually has access to the global sequence $\mathbf{x}(t)$, so even verifying the sufficient decrease property is not trivial. To simplify the presentation, we first analyze the performance of m-PAPG using the KŁ inequality and taking the sufficient decrease property for granted, and later we we will give some verifiable conditions to justify this simplification.

Our first result in this section strengthens the convergence properties in Theorems 5 and 7 for m-PAPG:

Theorem 11 (Finite Length) *Let Assumptions 1 to 5 and 8 hold for m-PAPG, and let F satisfy the KŁ property in Theorem 10. Then, with step size $\eta \in \left(0, \frac{1}{L(1+2\sqrt{ps})}\right)$, every bounded sequence $\{\mathbf{x}(t)\}$ generated by m-PAPG satisfies*

$$\sum_{t=0}^{\infty} \|\mathbf{x}(t+1) - \mathbf{x}(t)\| < \infty, \tag{58}$$

$$\forall i = 1, \dots, p, \ \sum_{t=0}^{\infty} \|\mathbf{x}^i(t+1) - \mathbf{x}^i(t)\| < \infty. \tag{59}$$

Furthermore, $\{\mathbf{x}(t)\}$ *and* $\{\mathbf{x}^i(t)\}_{i=1}^p$ *converge to the same critical point of* F*.*

Proof We first show that eq. (58) implies eq. (59). Indeed, recall from (15):

$$\|\mathbf{x}^i(t+1) - \mathbf{x}^i(t)\| \leq \sum_{k=(t-s)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|.$$

Therefore, summing for $t = 0, 1, \cdots, n$ gives

$$\begin{aligned}\sum_{t=0}^{n} \|\mathbf{x}^i(t+1) - \mathbf{x}^i(t)\| &\leq \sum_{t=0}^{n} \sum_{k=(t-s)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \\ &\leq (2s+1) \sum_{t=0}^{n} \|\mathbf{x}(t+1) - \mathbf{x}(t)\|.\end{aligned}$$

The claim then follows by letting n tend to infinity.

By Theorem 5, the limit points of $\{\mathbf{x}(t)\}$ and $\{\mathbf{x}^i(t)\}_{i=1}^p$ coincide and are critical points of F. Thus, the only thing left to prove is the finite length property in eq. (58). By Assumption 8 and Assumption 1, the objective value $F(\mathbf{x}(t))$ decreases to a finite limit F^*. Since $\{\mathbf{x}(t)\}$ is assumed to be bounded, the set of its limit points Ω is nonempty and compact. Summing eq. (18) over all i and set $\mathbf{z} \in \Omega$, we obtain

$$g(\mathbf{x}(t+1)) \leq g(\mathbf{z}) - \frac{1}{2\eta}\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 - \sum_{i=1}^{p} \langle \nabla_i f(\mathbf{x}^i(t)), \mathbf{x}(t+1) - \mathbf{x}(t) \rangle.$$

Note that $\mathbf{x}(t+1) - \mathbf{x}(t) \to 0$. Also, since $\{\mathbf{x}(t)\}$ is bounded and $\mathbf{x}(t) - \mathbf{x}^i(t) \to 0$ for all i, $\{\mathbf{x}^i(t)\}_{i=1}^p$ are all bounded. we then take limsup on both sides and obtain that $\limsup_{t\to\infty} g(\mathbf{x}(t+1)) \leq g(\mathbf{z})$. Together with the closedness of g we further obtain that $\lim_{t\to\infty} g(\mathbf{x}(t+1)) = g(\mathbf{z})$. Note that f is continuous, we thus conclude that $\lim_{t\to\infty} F(\mathbf{x}(t+1)) = F(\mathbf{z})$ for all $\mathbf{z} \in \Omega$. Note that $F(\mathbf{x}(t)) \downarrow F^*$. Thus for all $\mathbf{x}^* \in \Omega$, we have $F(\mathbf{x}^*) \equiv F^*$. Now fix $\varepsilon > 0$. Since Ω is compact, for t sufficiently large we have $\mathrm{dist}_\Omega(\mathbf{x}(t)) \leq \varepsilon$. We now have all ingredients to apply the KŁ inequality in Theorem 10: for all sufficiently large t,

$$\varphi'(F(\mathbf{x}(t)) - F^*) \cdot \mathrm{dist}_{\partial F(\mathbf{x}(t))}(\mathbf{0}) \geq 1. \tag{60}$$

Since φ is concave, we obtain

$$\begin{aligned}\Delta_{t,t+1} &:= \varphi(F(\mathbf{x}(t)) - F^*) - \varphi(F(\mathbf{x}(t+1)) - F^*) \\ &\geq \varphi'(F(\mathbf{x}(t)) - F^*)(F(\mathbf{x}(t)) - F(\mathbf{x}(t+1))) \\ &\overset{\text{(i)}}{\geq} \frac{\alpha\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2}{\mathrm{dist}_{\partial F(\mathbf{x}(t))}(\mathbf{0})},\end{aligned} \tag{61}$$

where (i) follows from Assumption 8 and eq. (60). It is clear that the function φ (composed with F) serves as a Lyapunov function. Using the elementary inequality $2\sqrt{ab} \leq a + b$ we obtain from eq. (61) that for t sufficiently large,

$$2\|\mathbf{x}(t+1) - \mathbf{x}(t)\| \leq \tfrac{\delta}{\alpha}\Delta_{t,t+1} + \tfrac{1}{\delta}\mathrm{dist}_{\partial F(\mathbf{x}(t))}(\mathbf{0}),$$

where $\delta > 0$ will be specified later. Recalling the bound for $\partial F(\mathbf{x}(t))$ in eq. (30), and summing over t from m (sufficiently large) to n gives:

$$\begin{aligned}
2\sum_{t=m}^{n}\|\mathbf{x}(t+1)-\mathbf{x}(t)\| &\le \sum_{t=m}^{n}\frac{\delta}{\alpha}\Delta_{t,t+1}+\sum_{t=m}^{n}\frac{1}{\delta}\mathrm{dist}_{\partial F(\mathbf{x}(t))}(\mathbf{0})\\
&\overset{\text{(i)}}{\le} \frac{\delta}{\alpha}\varphi\big(F(\mathbf{x}(m))-F^*\big)+\sum_{t=m}^{n}\frac{\sqrt{p}(1/\eta+2L)}{\delta}\sum_{k=(t-2s)_+}^{t}\|\mathbf{x}(k+1)-\mathbf{x}(k)\|\\
&\le \frac{\delta}{\alpha}\varphi\big(F(\mathbf{x}(m))-F^*\big)+\frac{(2s+1)\sqrt{p}(1/\eta+2L)}{\delta}\sum_{k=(m-2s)_+}^{m-1}\|\mathbf{x}(k+1)-\mathbf{x}(k)\|\\
&\quad+\frac{(2s+1)\sqrt{p}(1/\eta+2L)}{\delta}\sum_{t=m}^{n}\|\mathbf{x}(t+1)-\mathbf{x}(t)\|,
\end{aligned}$$

where (i) is due to eq. (30). Setting $\delta=(2s+1)\sqrt{p}(1/\eta+2L)$ and rearranging gives

$$\begin{aligned}
\sum_{t=m}^{n}\|\mathbf{x}(t+1)-\mathbf{x}(t)\| &\le \frac{(2s+1)\sqrt{p}(1/\eta+2L)}{\alpha}\varphi\big(F(\mathbf{x}(m))-F^*\big)\\
&\quad+\sum_{k=(m-2s)_+}^{m-1}\|\mathbf{x}(k+1)-\mathbf{x}(k)\|.
\end{aligned}$$

Since the right-hand side is finite, let n tend to infinity completes the proof for eq. (58). ■

Compared with (16) in Theorem 5, we now have the successive differences to be absolutely summable (instead of square summable). This is a significantly stronger result as it immediately implies that the whole sequence is Cauchy and hence convergent, whereas we cannot get the same conclusion from the square summable property in Theorem 5. We note that local maxima are excluded from being the limit in Theorem 11, due to Assumption 8. Also, the boundedness assumption on the trajectory $\{\mathbf{x}(t)\}$ is easy to satisfy, for instance, when F has bounded sublevel sets. We refer to (Attouch et al., 2010, Remark 3.3) for more conditions that imply the boundedness condition. Moreover, following similar arguments in Attouch et al. (2010) we can also determine the local convergence rates of the sequences generated by m-PAPG.

In the remaining part of this section we provide some justifications for the sufficient decrease property in Assumption 8. For simplicity we assume all worker machines perform updates in each time step t:

Assumption 9 $\forall i=1,\cdots,p,\forall t, t\in T_i$.

Note that Assumption 9 is commonly adopted in the analysis of many recent parallel systems Agarwal and Duchi (2011); Feyzmahdavian et al. (2014); Ho et al. (2013); Li et al. (2014); Liu and Wright (2015); Recht et al. (2011). In fact, Assumption 9 is somewhat necessary to justify Assumption 8. This is because Assumption 8 requires a sufficient decrease of the function value at every iteration k, which may not hold under the PAP as all machines can be idle for s iterations in the worst case. In other words, to achieve convergence of $\{\mathbf{x}_k\}_k$

to a critical point in nonconvex optimization under the KL inequality, the parallel system should make a steady progress per-iteration. As we show next, this is guaranteed under Assumptions 9 and 10.

We will replace the sufficient decrease property in Assumption 8 with the following key property that turns out to be easier to verify:

Assumption 10 (Proximal Lipschitz) *We say a pair of functions f and g satisfy the proximal Lipschitz property on a sequence $\{\mathbf{x}(t)\}$ if for all η sufficiently small, there exists $L_\eta \in o(1)$, i.e. $L_\eta \to 0$ as $\eta \to 0$, such that for all large t,*

$$\|\Delta_\eta(\mathbf{x}(t)) - \Delta_\eta(\mathbf{x}(t+1))\| \leq L_\eta \|\mathbf{x}(t) - \mathbf{x}(t+1)\|, \tag{62}$$

where[2] $\Delta_\eta(\mathbf{x}) \in \operatorname{prox}_g^\eta(\mathbf{x} - \eta\nabla f(\mathbf{x})) - \mathbf{x}$.

The proximal Lipschitz assumption is motivated by the special case where $g \equiv 0$ and hence $\Delta_\eta(\mathbf{x}) = -\eta\nabla f(\mathbf{x})$ is η-Lipschitz, thanks to Assumption 2. As we have seen in previous sections, Lipschitz continuity plays a crucial role in our proof where a major difficulty is to control the inconsistencies among different worker machines due to communication delays. Similarly here, the proximal Lipschitz property, as we show next, allows us to remove the sufficient decrease property in Assumption 8—the seemingly strong assumption that we needed in proving our main result Theorem 11.

Let us first present a quick justification for Assumption 10.

Lemma 12 *Suppose the functions f and g both have Lipschitz continuous gradient, then Assumption 10 holds for any sequence $\{\mathbf{x}(t)\}$.*

Proof Let us denote L_f and L_g as the Lipschitz constant of the gradient ∇f and ∇g, respectively. Since $\Delta_\eta(\mathbf{x}) \in \operatorname{prox}_g^\eta(\mathbf{x} - \eta\nabla f(\mathbf{x})) - \mathbf{x}$, using the optimality condition for the proximal map, see for instance (Yu et al., 2015, Proposition 7(iii)), we have

$$\mathbf{x} + \Delta_\eta(\mathbf{x}) + \eta\nabla g(\mathbf{x} + \Delta_\eta(\mathbf{x})) = \mathbf{x} - \eta\nabla f(\mathbf{x}),$$

and similarly

$$\mathbf{z} + \Delta_\eta(\mathbf{z}) + \eta\nabla g(\mathbf{z} + \Delta_\eta(\mathbf{z})) = \mathbf{z} - \eta\nabla f(\mathbf{z}).$$

Subtracting one inequality from another, we obtain

$$\begin{aligned}\|\Delta_\eta(\mathbf{x}) - \Delta_\eta(\mathbf{z})\| &= \|\eta\nabla g(\mathbf{z} + \Delta_\eta(\mathbf{z})) - \eta\nabla g(\mathbf{x} + \Delta_\eta(\mathbf{x})) + \eta\nabla f(\mathbf{z}) - \eta\nabla f(\mathbf{x})\| \\ &\leq \eta L_g \|\mathbf{z} - \mathbf{x} + \Delta_\eta(\mathbf{z}) - \Delta_\eta(\mathbf{x})\| + \eta L_f \|\mathbf{z} - \mathbf{x}\| \\ &\leq \eta L_g \|\Delta_\eta(\mathbf{z}) - \Delta_\eta(\mathbf{x})\| + \eta(L_f + L_g)\|\mathbf{z} - \mathbf{x}\|.\end{aligned}$$

Rearranging we obtain

$$\|\Delta_\eta(\mathbf{x}) - \Delta_\eta(\mathbf{z})\| \leq \frac{\eta(L_f + L_g)}{1 - \eta L_g}\|\mathbf{z} - \mathbf{x}\|,$$

when $0 < \eta < 1/L_g$. Clearly, when η is mall, the leading coefficient $\frac{\eta(L_f+L_g)}{1-\eta L_g} \in \mathcal{O}(\eta) \subseteq o(1)$, and our proof is complete. ■

2. Should the proximal map be multi-valued, we contend with any single-valued selection.

It is clear that Lemma 12 captures the motivating case $g \equiv 0$, but also many other important functions, such as the widely-used regularization function $g = \|\cdot\|_p^p$ for any $p > 1$. We can now continue with our next result in this section.

Theorem 13 *Let Assumptions 1 to 4 and 9 hold for m-PAPG, and let F satisfy the KŁ property in Theorem 10. Fix any $r > 1$ with $C = \frac{r^{s+1}-1}{r-1}$ and step size η such that $\eta < \frac{1}{L(1+2\sqrt{p}C+2\sqrt{p}s)}$. If for each local sequence $\{\mathbf{x}^i(t)\}$ generated by m-PAPG, Assumption 10 holds with $L_\eta \leq \frac{r^2-1}{2pr^2C^2}$, and the global sequence $\{\mathbf{x}(t)\}$ is bounded, then the finite length properties in (58) and (59) hold. Then, Assumption 8 holds, and consequently, $\{\mathbf{x}(t)\}$ and $\{\mathbf{x}^i(t)\}_{i=1}^p$ converge to the same critical point of F based on Theorem 11.*

Theorem 13 assumes that $L_\eta \leq \frac{r^2-1}{2pr^2C^2}$. We note that L_η implicitly depends on the stepsize η, i.e., $L_\eta \to 0$ as $\eta \to 0$ (see Assumption 10). Thus, one can tune the stepsize η to be small enough such that L_η satisfies the requirement. As an example, if $g = \|\mathbf{x}\|_2^2$, then one can calculate that $L_\eta = \mathcal{O}(\eta)$. In this case, we should choose the stepsize to be roughly $\eta \leq \frac{r^2-1}{2pr^2C^2}$.

Proof Using the elementary inequality $\|a\|^2 - \|b\|^2 \leq 2\|a\|\|a-b\|$, we have for all t:

$$\begin{aligned}
&\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2 - \|\mathbf{x}(t+2)-\mathbf{x}(t+1)\|^2 \\
&\quad\leq 2\|\mathbf{x}(t+1)-\mathbf{x}(t)\| \cdot \|(\mathbf{x}(t+1)-\mathbf{x}(t)) - (\mathbf{x}(t+2)-\mathbf{x}(t+1))\| \\
&\quad\leq 2\|\mathbf{x}(t+1)-\mathbf{x}(t)\| \cdot \sum_{i=1}^p \|(x_i(t+1)-x_i(t)) - (x_i(t+2)-x_i(t+1))\| \\
&\quad\overset{\text{(i)}}{\leq} 2\|\mathbf{x}(t+1)-\mathbf{x}(t)\| \cdot \sum_{i=1}^p \left\|\Delta_\eta(\mathbf{x}^i(t)) - \Delta_\eta(\mathbf{x}^i(t+1))\right\| \\
&\quad\overset{\text{(ii)}}{\leq} 2\|\mathbf{x}(t+1)-\mathbf{x}(t)\| \left(\sum_{i=1}^p L_\eta \|\mathbf{x}^i(t)-\mathbf{x}^i(t+1)\|\right) \\
&\quad\overset{\text{(iii)}}{\leq} 2pL_\eta\|\mathbf{x}(t+1)-\mathbf{x}(t)\| \cdot \sum_{k=(t-s)_+}^t \|\mathbf{x}(k+1)-\mathbf{x}(k)\|,
\end{aligned} \tag{63}$$

where (i) is due to Assumption 9 hence $t \in T_i$ for all t, (ii) follows from Assumption 10, and (iii) is due to (15).

If for some $r > 1$ there exists some T such that for all $t \geq T$,

$$\sum_{k=(t-s)_+}^t \|\mathbf{x}(k+1)-\mathbf{x}(k)\| \geq C\|\mathbf{x}(t+1)-\mathbf{x}(t)\|, \tag{64}$$

where $C = \frac{r^{s+1}-1}{r-1} > s+1$ (since $r > 1$ and w.l.o.g. $s > 0$). Summing the index t from T to n yields

$$C\sum_{t=T}^n \|\mathbf{x}(t+1)-\mathbf{x}(t)\| \leq \sum_{t=T}^n \sum_{k=(t-s)_+}^t \|\mathbf{x}(k+1)-\mathbf{x}(k)\|$$

$$\leq (s+1) \sum_{t=(T-s)_+}^{n} \|\mathbf{x}(t+1) - \mathbf{x}(t)\|,$$

which after rearranging terms becomes

$$(C-s-1) \sum_{t=T}^{n} \|\mathbf{x}(t+1) - \mathbf{x}(t)\| \leq (s+1) \sum_{t=(T-s)_+}^{T-1} \|\mathbf{x}(t+1) - \mathbf{x}(t)\|.$$

Since the right hand side does not depend on n, letting n tend to infinity we conclude

$$\sum_{t=0}^{\infty} \|\mathbf{x}(t+1) - \mathbf{x}(t)\| < \infty, \tag{65}$$

and the proof of the finite length property would be complete.

Therefore, in the remaining part of the proof, we can assume (64) fails for infinitely many t. Take any such $t = \hat{t}$, we have

$$\sum_{k=(t-s)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \leq C\|\mathbf{x}(t+1) - \mathbf{x}(t)\| \leq C^2\|\mathbf{x}(t+1) - \mathbf{x}(t)\|, \tag{66}$$

since $C > 1$. Combining (63) and (66) we have for $t = \hat{t}$:

$$\begin{aligned}\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 - \|\mathbf{x}(t+2) - \mathbf{x}(t+1)\|^2 &\leq 2pL_\eta C^2 \|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 \\ &\leq \left(1 - \frac{1}{r^2}\right) \|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2,\end{aligned}$$

if η is small enough (recall that $L_\eta = o(1)$). After rearranging terms we conclude that for $t = \hat{t}$:

$$\|\mathbf{x}(t+1) - \mathbf{x}(t)\| \leq r\|\mathbf{x}(t+2) - \mathbf{x}(t+1)\|. \tag{67}$$

Using induction we can continue the same process for any $t \geq \hat{t}$. Indeed, suppose (67) is true for any $t \leq m-1$, then (63) holds (for any t), and (66) also holds: If $m \leq \hat{t} + s$, then

$$\begin{aligned}\sum_{k=(m-s)_+}^{m} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| &= \sum_{k=(m-s)_+}^{\hat{t}} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| + \sum_{k=\hat{t}+1}^{m} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \\ &\overset{\text{(i)}}{\leq} \sum_{k=(\hat{t}-s)_+}^{\hat{t}} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| + \sum_{k=\hat{t}+1}^{m} r^{m-k}\|\mathbf{x}(m+1) - \mathbf{x}(m)\| \\ &\overset{\text{(ii)}}{\leq} C\left[\|\mathbf{x}(\hat{t}+1) - \mathbf{x}(\hat{t})\| + \sum_{k=\hat{t}+1}^{m} r^{m-k}\|\mathbf{x}(m+1) - \mathbf{x}(m)\|\right] \\ &\overset{\text{(iii)}}{\leq} C\sum_{k=\hat{t}}^{m} r^{m-k}\|\mathbf{x}(m+1) - \mathbf{x}(m)\| \\ &\overset{\text{(iv)}}{\leq} C^2\|\mathbf{x}(m+1) - \mathbf{x}(m)\|,\end{aligned}$$

where (i) is due to the induction hypothesis, (ii) is due to the definition of $\hat{t}$ and the fact that $C > 1$, (iii) is due to again the induction hypothesis, and finally (iv) is due to the definition of C (recall $m \leq \hat{t} + s$). If $m > \hat{t} + s$, the same inequality, with C^2 replaced by C, would still hold (essentially dropping all the first terms on the right hand side of the above inequalities). Thus, (63) and (66) would imply again (67) for $t = m$.

Lastly, we recall from eq. (22) that for large t,

$$\begin{aligned} F(\mathbf{x}(t+1)) - F(\mathbf{x}(t)) &\leq \tfrac{1}{2}(L - 1/\eta)\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 \\ &\quad + \sqrt{p}L\|\mathbf{x}(t+1) - \mathbf{x}(t)\| \sum_{k=(t-s)_+}^{t-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|. \\ &\leq \tfrac{1}{2}(L - 1/\eta)\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 + \sqrt{p}CL\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2. \\ &\leq -\alpha\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2, \end{aligned}$$

where $\alpha = \frac{1}{2}(1/\eta - L - 2\sqrt{p}CL) > 0$ if η is small. Hence, the sufficient decrease property in Assumption 8 is verified and the finite length properties follow from Theorem 11. ■

Lastly, we show that Assumption 10 also holds for the important cardinality function $\|\mathbf{x}\|_0$ (number of nonzero entries).

Lemma 14 *Consider the same setting as in Theorem 5, then Assumption 10 holds for any function f and $g = \|\cdot\|_0$ on all local sequences $\{\mathbf{x}^i(t)\}$ of m-PAPG.*

Proof The crucial observation here is that for the cardinality function $g = \|\cdot\|_0$, its proximal map on the j-th entry can be chosen as:

$$\text{prox}_{g_j}^{\eta}(z_j) = \begin{cases} z_j, \text{ if } |z_j| > \sqrt{2\eta} \\ 0, \text{ otherwise} \end{cases}. \tag{68}$$

However, Theorem 5 implies that $\lim_{t\to\infty} \|\mathbf{x}^i(t+1) - \mathbf{x}^i(t)\| = 0$. Thus, for t sufficiently large, the sequence $\{\mathbf{x}^i(t)\}$ will have the same support Ω (indices that have nonzero entries), for otherwise $\|\mathbf{x}^i(t+1) - \mathbf{x}^i(t)\| \geq \sqrt{2\eta}$ even if one index in the support changes. Therefore,

$$\begin{aligned} \|\Delta_\eta(\mathbf{x}^i(t+1)) - \Delta_\eta(\mathbf{x}^i(t))\| &\overset{\text{(i)}}{\leq} \sum_{j\in\Omega} \|\text{prox}_{g_j}^{\eta}(x_j^i(t+1) - \eta\nabla_j f(\mathbf{x}^i(t+1))) - x_j^i(t+1) \\ &\qquad - \text{prox}_{g_j}^{\eta}(x_j^i(t) - \eta\nabla_j f(\mathbf{x}^i(t))) - x_j^i(t)\| \\ &\overset{\text{(ii)}}{\leq} \sum_{j\in\Omega} \|\eta\nabla_j f(\mathbf{x}^i(t+1)) - \eta\nabla_j f(\mathbf{x}^i(t))\| \\ &\overset{\text{(iii)}}{\leq} \eta pL\|\mathbf{x}^i(t+1) - \mathbf{x}^i(t)\|, \end{aligned}$$

where (i) is the triangle inequality, (ii) uses the property of the proximal map (68), and (iii) is due to Assumption 2. ■

Note that similar results as Theorem 14 can be derived for the rank function, and more generally for functions whose proximal map is discontinuous with pieces satisfying Theorem 12 (for instance, the group cardinality norm $\|\cdot\|_{0,2}$).

7. Economical Implementation for Linear Models

In this section, we provide an economical implementation of m-PAPG on a distributed system for the widely used linear models:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(A\mathbf{x}) + g(\mathbf{x}), \tag{69}$$

where $A \in \mathbb{R}^{n\times d}$ corresponds to the data matrix. Typically $f : \mathbb{R}^n \to \mathbb{R}$ is the likelihood function and $g : \mathbb{R}^d \to \mathbb{R}$ is the regularizer. The data matrix A consists of n sample points and we have suppressed the labels in classification or the responses in regression. Support vector machines (SVM), Lasso, logistic regression, boosting, etc., all fit under this framework. Our interest here is when the model dimension d is much higher than the number of samples n (d can be up to hundreds of millions and n can be up to millions). This is also the usual setup in many computational biology and health care problems.

A direct implementation of m-PAPG can be inefficient in terms of both network communication and parameter storage. First, each machine needs to communicate with every other machine to synchronize the model blocks. This leads to a peer-to-peer network topology and result in a dense connection when the system holds hundreds of machines. Second, each machine needs to keep a local copy of the full model (i.e. $\mathbf{x}^i(t)$), which incurs a high storage cost when the dimension is high. Note that the local models $\mathbf{x}^i(t)$ are kept solely for the convenience of evaluating the partial gradient $\nabla_i f : \mathbb{R}^d \to \mathbb{R}^{d_i}$. For some problems such as the Lasso, a seemingly workaround is to pre-compute the Hessian $H = A^\top A$ and distribute the corresponding row blocks of H to each worker machine. This scheme, however, is problematic in the high dimensional setting: the pre-computation of the Hessian can be very costly, and each row block of H has a very large size ($d_i \times d$).

The above issues can be avoided by exploiting the structure of the linear model in eq. (69) and adopting the parameter server distributed system Ho et al. (2013); Li et al. (2014). The system dedicates a central server to store the key parameters, and let each worker machine to communicate only with the server. To be specific, we partition the data matrix A into p *column* blocks $A = [A_1, \ldots, A_p]$ and distribute the block $A_i \in \mathbb{R}^{n\times d_i}$ to machine i Boyd et al. (2010); Richtárik and Takáč (to appear) . Note the local update computed by machine i at the t-th iteration is

$$U_i(\mathbf{x}^i(t)) = \mathrm{prox}^{\eta}_{g_i}(x_i(t) - \eta A_i^\top f'(A\mathbf{x}^i(t))) - x_i(t). \tag{70}$$

Since machine i is in charge of updating the i-th block $x_i(t)$ of the global model, it suffices to have the matrix-vector product $A\mathbf{x}^i(t)$ to compute the local update in eq. (70) . If we initialize $\forall i,\ \mathbf{x}^i(0) \equiv \mathbf{0}$, then $A\mathbf{x}^i(t)$ can be written in a cumulative form as

$$A\mathbf{x}^i(t) = \sum_{j=1}^{p} A_j[\mathbf{x}^i(t)]_j = \sum_{j=1}^{p}\sum_{k=0}^{\tau_j^i(t)} \underbrace{A_j \mathbb{I}_{\{k\in T_j\}} U_j(\mathbf{x}^j(k))}_{\Delta_j(k)},$$

where recall that machine i only has access to a delayed copy $x_j(\tau_j^i(t))$ of the parameters in machine j. Hence, to evaluate the matrix-vector product, every machine needs to accumulate $\Delta_j(k)$ over all machines upto a delayed clock. Thus, we aggregate $\Delta_j(t) \in \mathbb{R}^n$ on the parameter server whenever it is generated and sent by the worker machines. In details,

Algorithm 1 Economic Implementation of m-PAPG

For the server:
  while recieves update Δ_i from machine i **do**
    $\blacktriangle \leftarrow \blacktriangle + \Delta_i$
  end while
  while machine i sends a pull request **do**
    send $\blacktriangle$ to machine i
  end while
For machine i at active clock $t \in T_i$:
  pull $\blacktriangle$ from the server
  $U_i \leftarrow \mathrm{prox}_{g_i}^{\eta}(x_i - \eta A_i^\top f'(\blacktriangle)) - x_i$
  send $\Delta_i = A_i U_i$ to the server
  update $x_i \leftarrow x_i + U_i$

the worker machines first pull this matrix-vector product (denoted as $\blacktriangle$) from the server to conduct the local computation in eq. (70). Then machine i performs the local update:

$$x_i(t+1) = x_i(t) + U_i(\mathbf{x}^i(t)). \tag{71}$$

Note that machine i does not maintain or update other blocks of parameters $x_j(t), j \neq i$. Lastly, machine i computes and sends the vector $\Delta_i(t) = A_i U_i(\mathbf{x}^i(t)) \in \mathbb{R}^n$ to the server, and the server immediately performs the aggregation:

$$\blacktriangle \leftarrow \blacktriangle + \Delta_i(t). \tag{72}$$

We summarize the above economical implementation in Algorithm 1, where $\blacktriangle$ denotes the aggregated matrix-vector product. The storage cost for each worker machine is $O(nd_i)$ (for storing A_i only). Each iteration requires two matrix-vector products that cost $O(nd_i)$ in the dense case, and the communication of a length n vector between the server and the worker machines. Note that the cost is significantly lower than the direct implementation.

8. Experiments

In this section, we empirically verify the convergence properties and time efficiency of m-PAPG. All data are generated via normal distribution with the columns being normalized to have unit norm. We first test the convergence properties of m-PAPG via a non-convex Lasso problem with the group regularizer $\|\cdot\|_{0,2}$, which takes the form

$$\min_{\mathbf{x}\in\mathbb{R}^d} \ \tfrac{1}{2}\|A\mathbf{x} - \mathbf{b}\|^2 + \lambda\|\mathbf{x}\|_{0,2}, \tag{73}$$

where we set sample size $n = 1000$ and dimension size $d = 2000$, and the group norm divides the whole model into 20 groups with equal dimension. We use 4 machines (cores) with each handling five groups of coordinates, and consider maximal staleness $s = 0, 10, 20, 30$, respectively. To better demonstrate the effect of staleness, we let machines only communicate when exceed the maximum staleness. This can be viewed as the worst case communication scheme and a larger s brings more staleness into the system. We set the learning rate to

have the form $\eta(\alpha s) = 1/(L_f + 2L\alpha s), \alpha > 0$, that is, a linear dependency on staleness s as suggested by Theorem 5. Then we run Algorithm 1 with different staleness and use $\eta(0), \eta(10), \eta^*(\alpha s)$, respectively, where $\eta^*(\alpha s)$ is the largest step size we tuned for each s that achieves a stable convergence. We track the global model $\mathbf{x}(t)$ and plot the results in Figure 1. Note that with the large step size $\eta(0)$ all instances (with nonzero staleness) diverge hence are not presented. With $\eta(10)$ (Figure 1, left), the staleness does not substantially affect the convergence in terms of the objective value. We note that the objective curves converge to slightly different minimal values due to the non-convexity of problem (73). With $\eta^*(\alpha s)$ (Figure 1, middle), it can be observed that adding a slight penalty αs on the learning rate suffices to achieve a stable convergence, and the penalty grows as s increases, which is intuitive since a larger staleness requires a smaller step size to cancel the inconsistency. In particular, for $s = 10$ the best convergence is comparable to the bulk synchronized case $s = 0$. (Figure 1, right) further shows the asymptotic convergence behavior of the global model $\mathbf{x}(t)$ under the step size $\eta^*(\alpha s)$. It is clear that a linear convergence is eventually attained, which confirms the finite length property in Theorem 11.

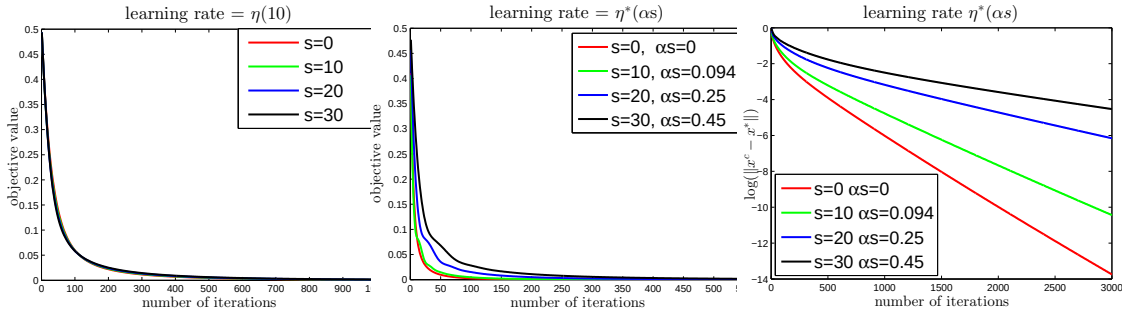

Figure 1: Convergence curves of m-PAPG under different staleness parameter s and step size η.

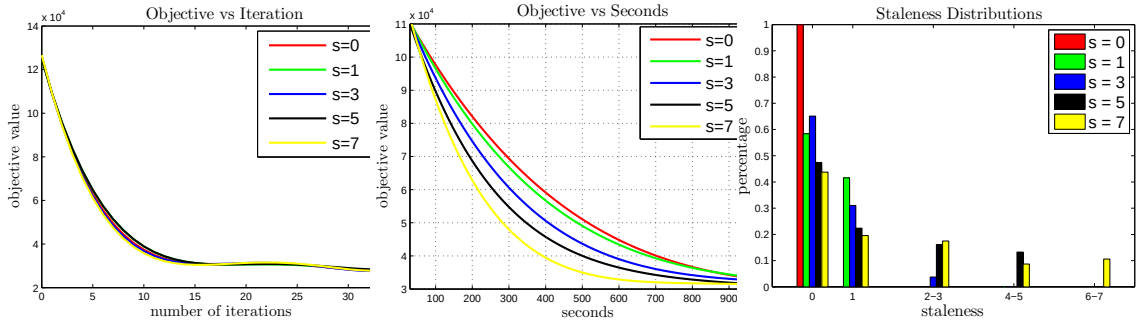

Figure 2: Efficiency of m-PAPG on a large scale Lasso problem.

Next, we verify the time and communication efficiency of m-PAPG via an l_1 regularized quadratic programming problem with very high dimensions, taking the form

$$\min_{\mathbf{x}} \ \tfrac{1}{2}\mathbf{x}^\top A^\top A\mathbf{x} + \lambda\|\mathbf{x}\|_1. \tag{74}$$

We generate samples of size $n = 1$Million and dimension $d = 100$Millions. We implement Algorithm 1 on Petuum Ho et al. (2013); Dai et al. (2014) — a stale synchronous parallel system which updates the local parameter caches via stale synchronous communications. The system contains 100 computing nodes and each is equipped with 16 AMD Opteron processors and 16GB RAM linked by 1Gbps ethernet. We fix the learning rate $\eta = 10^{-3}$ and consider maximum staleness $s = 0, 1, 3, 5, 7$, respectively. (Figure 2, left) shows that per-iteration progress is virtually indistinguishable among various staleness settings, which is consistent with our previous experiment. (Figure 2, middle) shows that system throughput is significantly higher when we introduce staleness. This is due to lower synchronization overheads, which offsets any potential loss due to staleness in progress per iteration. We also track the distributions of staleness during the experiments, where we record in $\blacktriangle$ the clocks of the freshest updates that accumulate from all the machines. Then whenever a machine pulls $\blacktriangle$ from the server, it compares its local clock with these clocks and records the clock differences. (Figure 2, right) shows the distributions of staleness under different maximal staleness settings. Observe that bulk synchronous ($s = 0$) peaks at staleness 0 by design, and the distribution concentrates in small staleness area due to the eager communication mechanism of Petuum. It can be seen that a small amount of staleness is sufficient to relax the communication bottlenecks without affecting the iterative convergence rate much.

9. Conclusion

We have proposed m-PAPG as an extension of the proximal gradient algorithm to the model parallel and partially asynchronous setting. m-PAPG allows worker machines to operate asynchronously as long as they are not too far apart, hence greatly improves the system throughput. The convergence properties of m-PAPG are thoroughly analyzed. In particular, we proved that: 1) every limit point of the sequences generated by m-PAPG is a critical point of the objective function; 2) under an additional error bound condition, the function values decay periodically linearly; 3) under the additional Kurdyka-Łojasiewicz inequality, the sequences generated by m-PAPG converge to the same critical point, provided that a proximal Lipschitz condition is satisfied. In the future we plan to further weaken the proximal Lipschitz condition so that our analysis can handle many more nonsmooth functions.

Acknowledgment

This work of Y. Zhou and Y. Liang is supported in part by the grants AFOSR FA9550-16-1-0077, NSF ECCS-1818904 and CCF-1761506.

References

P.-A. Absil, R. Mahony, and B. Andrews. Convergence of the iterates of descent methods for analytic cost functions. *SIAM Journal on Optimization*, 16(2):531–547, 2005.

Alekh Agarwal and John C. Duchi. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems 24*, pages 873–881. 2011.

Hedy Attouch and Jerome Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116(1-2):5–16, 2009. ISSN 0025-5610.

Hedy Attouch, Jerome Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

Gérard M. Baudet. Asynchronous iterative methods for multiprocessors. *Journal of the Association for Computing Machinery*, 25(2):226–244, 1978.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, 2009.

Amir Beck and Marc Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

Jéróme Bolte, Aris Danilidis, Olivier Ley, and Laurent Mazet. Characterizations of Łojasiewicz inequalities and applications: Subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010.

Jerome Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2): 459–494, 2014.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

D. Chazan and W. Miranker. Chaotic relaxation. *Linear Algebra and Its Applications*, 2: 199–222, 1969.

Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou. Trading convexity for scalability. pages 201–208, 2006.

Wei Dai, Abhimanu Kumar, Jinliang Wei, Qirong Ho, Garth Gibson, and Eric P. Xing. High-performance distributed ml at scale through parameterserver consistency models. In *AAAI*, 2014.

Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications of ACM*, 51(1):107–113, 2008.

Dmitriy Drusvyatskiy and Adrian S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods, 2016.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

Cong Fang and Zhouchen Lin. Parallel asynchronous stochastic variance reduction for nonconvex optimization, 2017.

Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

H.R. Feyzmahdavian, A. Aytekin, and M. Johansson. A delayed proximal gradient method with linear convergence rate. In *IEEE International Workshop on Machine Learning for Signal Processing*, 2014.

Masao Fukushima and Hisashi Mine. A generalized proximal point algorithm for certain non-convex minimization problems. *International Journal of Systems Science*, 12(8): 989–1000, 1981.

D. Hajinezhad and M. Hong. Nonconvex alternating direction method of multipliers for distributed sparse principal component analysis. In *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 255–259, Dec 2015.

Qirong Ho, James Cipar, Henggang Cui, Seunghak Lee, Jin Kyu Kim, Phillip B. Gibbons, Garth A Gibson, Greg Ganger, and Eric P Xing. More effective distributed ml via a stale synchronous parallel parameter server. In *Advances in Neural Information Processing Systems 26*, pages 1223–1231. 2013.

Mingyi Hong, Zhi Quan Luo, and Meisam Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 1 2016.

Zhouyuan Huo and Heng Huang. Asynchronous mini-batch gradient descent with variance reduction for non-convex optimization, 2017.

Krzysztof Kurdyka. On gradients of functions definable in o-minimal structures. *Annales de l'institut Fourier*, 48(3):769–783, 1998.

Mu Li, David G. Andersen, Jun Woo Park, Alexander J. Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J. Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 583–598, 2014.

Ji Liu and Stephen J. Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015.

P. D. Lorenzo and G. Scutari. NEXT: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, June 2016.

Yucheng Low, Danny Bickson, Joseph Gonzalez, Carlos Guestrin, Aapo Kyrola, and Joseph M. Hellerstein. Distributed graphlab: A framework for machine learning and data mining in the cloud. *Proc. VLDB Endow.*, 5(8):716–727, 2012.

Zhaosong Lu and Lin Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, 152:615–642, 2015.

Zhi-Quan Luo and Paul Tseng. Error bounds and convergence analysis of feasible descent methods: A general approach. *Annals of Operations Research*, 46(1):157–178, 1993.

Rahul Mazumder, Jerome H. Friedman, and Trevor Hastie. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495): 1125–1138, 2011.

Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming, Series B*, 140:125–161, 2013.

Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems 24*, pages 693–701. 2011.

P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

P. Richtárik and M. Takáč. Distributed Coordinate Descent Method for Learning with Big Data. *Journal of Machine Learning Research*, to appear.

R.T. Rockafellar and R.J.B. Wets. *Variational Analysis*. Springer, 1997.

Suvrit Sra. Scalable nonconvex inexact proximal splitting. In *Advances of Neural Information Processing Systems*, 2012.

Paul Tseng. On the rate of convergence of a partially asynchronous gradient projection algorithm. *SIAM Journal on Optimization*, 1(4):603–619, 1991.

Leslie G. Valiant. A bridging model for parallel computation. *Communications of ACM*, 33(8):103–111, 1990.

Yichao Wu and Yufeng Liu. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102(479):974–983, 2007.

Linli Xu, Koby Crammer, and Dale Schuurmans. Robust support vector machine training via convex outlier ablation. 2006.

Yaoliang Yu, Xun Zheng, Micol Marchetti-Bowick, and Eric P. Xing. Minimizing nonconvex non-separable functions. In *The* 17^{th} *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. pages 10–10, 2010.

Cun-Hui Zhang. Nearly unbaised variable selection under minimax concave penalty. *Annals of Statistics*, 38(2):894–942, 2010.

Cun-Hui Zhang and Tong Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, 27(4):576–593, 2012.

Hui Zhang. The restricted strong convexity revisited: analysis of equivalence to error bound and quadratic growth. *Optimization Letters*, pages 1–17, 2016.

Y. Zhou, Y.L. Yu, W. Dai, Y.B. Liang, and E.P. Xing. On convergence of model parallel proximal gradient algorithm for stale synchronous parallel system. In *The* 19^{th} *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

Zirui Zhou, Qi Zhang, and Anthony Man-Cho So. $\ell_{1,p}$-norm regularization: Error bounds and convergence rate analysis of first-order methods. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1501–1510. JMLR Workshop and Conference Proceedings, 2015.