·前沿聚焦：优化技术与应用专栏·　　DOI：10. 13471/j. cnki. acta. snus. 2023A023

Fully asynchronous distributed optimization with linear convergence over directed networks*

SHA Xingyu[1], *ZHANG Jiaqi*[2], *YOU Keyou*[1]✉

1. Department of Automation, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China

2. Beijing Sankuai Online Technology Co., Ltd., Beijing 100102, China

Abstract: We study distributed optimization problems over a directed network, where nodes aim to minimize the sum of local objective functions via directed communications with neighbors. Many algorithms are designed to solve it for synchronized or randomly activated implementation, which may create deadlocks in practice. In sharp contrast, we propose a fully asynchronous push-pull gradient (APPG) algorithm, where each node updates without waiting for any other node by using possibly delayed information from neighbors. Then, we construct two novel augmented networks to analyze asynchrony and delays, and quantify its convergence rate from the worst-case point of view. Particularly, all nodes of APPG converge to the same optimal solution at a linear rate of $\mathcal{O}(\lambda^k)$ if local functions have Lipschitz-continuous gradients and their sum satisfies the Polyak-Łojasiewicz condition (convexity is not required), where $\lambda \in (0,1)$ is explicitly given and the virtual counter k increases by one when any node updates. Finally, the advantage of APPG over the synchronous counterpart and its linear speedup efficiency are numerically validated via a logistic regression problem.

Key words: fully asynchronous; distributed optimization; linear convergence; Polyak-Łojasiewicz condition

CLC number: O221.2　　**Document code:** A　　**Article ID:** 2097－0137（2023）05－0001－23

1　Introduction

As data get larger and more spatially distributed, distributed optimization over a network of computing nodes (aka. agents or workers) has found numerous applications in multi-agent problems (Chang et al. , 2015), machine learning (Assran et al. , 2019), and resources allocation (Zhang et al. , 2020b). It aims to minimize the sum of local objective functions, i. e. ,

$$\underset{\boldsymbol{x}\in\mathbb{R}^m}{\text{minimize}}\ f(\boldsymbol{x}) \coloneqq \sum_{i=1}^{n} f_i(\boldsymbol{x}), \tag{1}$$

where n is the number of nodes and the local objective function f_i is only known by node i. Nodes are expected to solve (1) by only communicating with neighbors that are defined by the network. For example, f_i often takes the form $f_i(\boldsymbol{x}) \coloneqq \sum_{\xi\in\mathcal{D}_i} F_i(\boldsymbol{x};\xi)$ in empirical risk minimization problems(Assran et al. , 2019), where $\mathcal{D}_i$ is a local data-

* **Received**: 2023－03－31　　**Accepted**: 2023－04－11　　**Published online**: 2023－08－30

Supported by National Natural Science Foundation of China（62033006，62203254）

✉**Corresponding author**: YOU Keyou（*youky@tsinghua.edu.cn*）

SHA Xingyu（*shaxy18@mails.tsinghua.edu.cn*）；ZHANG Jiaqi（*jqzhang61@gmail.com*）

set of node i, $\boldsymbol{x}$ is the model parameter to be optimized, and $F_i(\boldsymbol{x};\xi)$ is the loss of a single sample ξ. See Fig. 1 for an illustration.

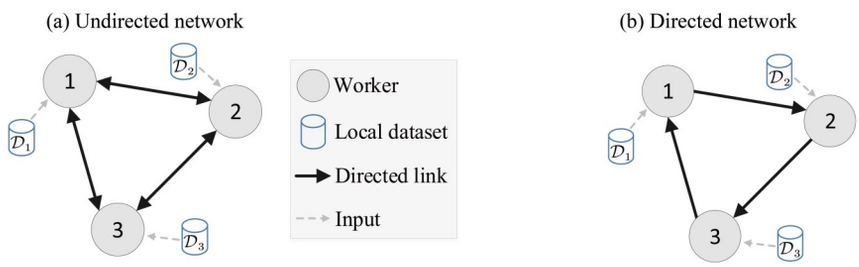

Fig. 1 Undirected and directed peer-to-peer networks

For large-scale optimization problems, it is crucial to design an easily implementable algorithm that is robust to heterogeneous nodes and communication delays. Many existing works focus on synchronous algorithms where all nodes essentially start to compute each iteration simultaneously (c.f. Fig. 2(a)) by using a global synchronization scheme that is often not amenable to the distributed setting.

Asynchronous updates are demonstrated to perform better than the synchronous counterparts (Nedić, 2010; Lian et al. , 2018; Wu et al. , 2018; Xu et al. , 2018; Assran et al. , 2020; Zhang et al. , 2020a). A popular one is gossip-based (Nedić, 2010; Lian et al. , 2018; Xu et al. , 2018) where a pair of neighbors is randomly selected to concurrently update via information exchange, see Fig. 2(b). However, this may create deadlocks in practice (Lian et al. , 2018), especially for networks with many cycles. It is also sensitive to communication delays and cannot work on directed networks.

To simultaneously address these issues, this work considers the *fully asynchronous* setting (c.f. Fig. 2(c) (Tian et al. , 2020; Zhang et al. , 2020a; Assran et al. , 2021)) over *directed* networks and proposes an asynchronous push-pull gradient (APPG) algorithm to distributedly solve (1). In APPG, a node starts to update without waiting for others with locally accessed (possibly stale) information. It does not need any network synchronization, and can tolerate uneven update frequencies and communication delays among nodes.

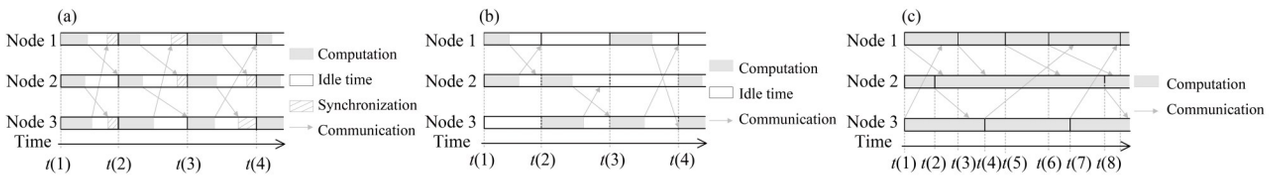

Fig. 2 （a）A synchronous algorithm over directed network；(b)A gossip-based algorithm；（c）A fully asynchronous algorithm

To theoretically evaluate its performance, we develop an augmented network approach and use the machinery of linear matrix inequalities (LMIs) to capture some key quantities via a novel λ-sequence. If all local functions f_i have Lipschitz continuous gradients and the global objective function f satisfies the Polyak-Łojasiewicz (PL) condition (no convexity requirement), we prove from the worst-case point of view that APPG converges linearly to an optimal solution at a rate $\mathcal{O}(\lambda^k)$, where $\lambda \in (0,1)$ is explicitly provided in terms of the asynchrony level and delay bounds, and the virtual counter k increases by one no matter which node updates. Note that the convergence guarantee for gossip-based algorithms is generally given in the stochastic sense.

Then, we implement APPG on a multi-core server with the Message Passing Interface (MPI) to solve a multi-class logistic regression problem over the *Covertype* dataset. The result confirms that its empirical convergence rate in running time is faster than its synchronous counterpart, and achieves linear speedup efficiency with respect to the node number. Its robustness to slow computing nodes is also validated, which is essential in heterogeneous

large-scale networks.

The rest of the paper is organized as follows. Section 2 briefly reviews related works. Section 3 formulates the problem and proposes APPG. We provide the theoretical guarantees for APPG in Section 4. To prove them, we first develop a time-varying augmented network approach in Section 5, and then establish the LMIs in Section 6. In Section 7, we conduct numerical experiments on APPG. Some concluding remarks are drawn in Section 8.

2 Related work

The past decade has witnessed an increasing attention on distributed optimization, especially for the design of synchronous algorithms over undirected networks, e. g. , DGD (Nedić et al. , 2009), DIGing (Nedić et al. , 2017), NIDS (Jakovetić, 2019;Li et al. , 2019; Sun et al. , 2019; Yuan et al. ,2019; Li et al. ,2020; Xin et al. , 2020; Yi et al. , 2020; Xu et al. , 2021). Extensions to directed networks while maintaining linear convergence are challenging due to the network unbalancedness. To resolve it, epigraph reformulation (Xie et al. , 2018) and the push-sum-based method (Nedić et al. , 2015; Assran et al. , 2019; Scutari et al. , 2019) have been proposed. For example, Push-DIGing (Nedić et al. , 2017) combines the push-sum method and gradient tracking (Nedić et al. , 2017; Xu et al. , 2018) to achieve linear convergence for strongly convex and Lipschitz smooth functions over directed networks. However, it involves nonlinear operators per update and may cause numerical issues in implementation, which is resolved in the push-pull/$\mathcal{AB}$ algorithm (Xin et al. , 2018; Saadatniaki et al. , 2020; Pu et al. , 2021) by simultaneously adopting a row-stochastic matrix and a column-stochastic matrix for information mixing.

In the meanwhile, asynchronous algorithms are proposed to alleviate the synchronization overhead. The gossip-based AsynDGM (Xu et al. , 2018) assumes that each pair of neighbors is activated to update with a fixed probability, and has been extended in Lian et al. (2018) where the activated nodes admit a doubly-stochastic mixing matrix. Nedić (2010) considers that each node is activated by a Poisson process. More general models are studied in Zhao et al. (2015); Wu et al. (2018) by assuming nodes or edges are independently randomly activated with fixed probabilities. It is unclear whether these methods can be applied to directed networks especially when there exist communication delays. Moreover, the coordination between neighbors is indispensable.

Inspired by the seminal works (Tsitsiklis et al. , 1986; Li et al. , 1987), the fully asynchronous setting has emerged as a more scalable and easier implementable alternative. An ADMM-based asynchronous algorithm is studied in Chang et al. (2016) for the master-slave networks. For peer-to-peer networks, an initial attempt is made in Assran et al. (2021) to extend the synchronous gradient-push algorithm (Nedić et al. , 2015) to the fully asynchronous setting. However, they cannot achieve exact convergence to an optimal solution if nodes have different update frequencies. To address it, Zhang et al. (2020a) proposes a novel adaptive mechanism to dynamically adjust stepsize, which is also adopted in Spiridonoff et al. (2020) to analyze stochastic algorithms. Nonetheless, the algorithms in Zhang et al. (2020a); Spiridonoff et al. (2020) only have sublinear convergence rates even if the objective function is strongly convex and Lipschitz smooth. Asy-SONATA (Tian et al. , 2020) exploits the perturbed push-sum method with gradient tracking (Nedić et al. , 2017; Xu et al. , 2018) in the fully asynchronous setting, which converges linearly for strongly convex problems and sublinearly for non-convex problems. The differences between Tian et al. (2020) and our work include: (i) APPG further converges linearly even under the PL condition, which holds for some important non-convex problems, e. g. , the policy optimization for LQR (Fazel et al. , 2018). (ii) APPG uses uncoordinated constant stepsizes while Asy-SONATA can only use uncoordinated diminishing stepsizes, which however leads to a sublinear convergence rate. (iii) We provide explicit convergence rate for APPG in terms of key parameters of the problem, while an explicit rate seems unclear for Asy-SONATA.

Notation: We use the following notations:

(i) $[\boldsymbol{A}]_{ij}$ denotes the element in row i and column j of $\boldsymbol{A}$. $|\mathcal{A}|$ denotes the cardinality of set $\mathcal{A}$. $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. $\mathbb{R}^n$ and $\mathbb{N}$ denote the set of n-dimensional real numbers and natural numbers, respectively.

(ii) $\|\cdot\|_2$ denotes the l_2-norm of a vector or matrix. $\|\cdot\|_{\mathrm{F}}$ denotes the matrix Frobenius norm. $\mathcal{O}(\cdot)$ denotes the big-O notation.

(iii) $\mathbf{1}_n$ and $\mathbf{0}_n$ denote respectively the n-dimensional vector with all ones and all zeros. The subscript may be omitted if the dimension is clear from the context.

(iv) $\nabla f(x)$ denotes the gradient of a function f at x.

(v) $\boldsymbol{a}$ is called a stochastic vector if it is nonnegative and $\boldsymbol{a}^{\mathrm{T}}\mathbf{1} = 1$. $\boldsymbol{A}$ is called a row-stochastic matrix if $\boldsymbol{A}$ is nonnegative and $\boldsymbol{A}\mathbf{1} = \mathbf{1}$. $\boldsymbol{A}$ is column-stochastic if $\boldsymbol{A}^{\mathrm{T}}$ is row-stochastic.

(vi) $[\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n]$ and $[\boldsymbol{a}_1; \boldsymbol{a}_2; \cdots; \boldsymbol{a}_n]$ denote the horizontal stack and vertical stack of $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n$, respectively.

(vii) $\Pi_{\mathcal{X}}(\boldsymbol{x})$ denotes the projection of $\boldsymbol{x}$ onto the set $\mathcal{X}$.

3 Problem formulation and the APPG

3.1 Problem formulation

We aim to solve (1) over a directed network (digraph), which is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \cdots, n\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. The directed edge $(i, j) \in \mathcal{E}$ if node i can directly send information to node j. Let $\mathcal{N}_{\mathrm{in}}^{i} = \{j \mid (j, i) \in \mathcal{E}\} \cup \{i\}$ denote the set of in-neighbors of node i and $\mathcal{N}_{\mathrm{out}}^{i} = \{j \mid (i, j) \in \mathcal{E}\} \cup \{i\}$ denote the set of out-neighbors of i. A path from node i to node j is a sequence of consecutively directed edges from node i to node j. Then, $\mathcal{G}$ is *strongly connected* if there exists a directed path between any pair of nodes.

For a distributed algorithm over $\mathcal{G}$, each node i has a local state vector $\boldsymbol{x}_i$ and iteratively update it via directed communications with neighbors, the objective of which is to ensure that all local states $\boldsymbol{x}_i, i \in \mathcal{V}$ converge to an optimal solution of (1). In this work, we make the following assumptions.

Assumption 1

(a) The digraph $\mathcal{G}$ is strongly connected.

(b) All local functions f_i are β-Lipschitz smooth, i.e. , there exists a $\beta > 0$ such that

$$\left\| \nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y}) \right\|_2 \leqslant \beta \left\| \boldsymbol{x} - \boldsymbol{y} \right\|_2, \ \forall i \in \mathcal{V}, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m.$$

(c) The global objective function f has at least one minimizer and satisfies the Polyak-Łojasiewicz condition (Karimi et al. , 2016) with parameter $\alpha > 0$, i.e. , $\mathcal{X}^\star = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^m} f(\boldsymbol{x}) \neq \varnothing$, and

$$2\alpha\left(f(\boldsymbol{x}) - f^\star\right) \leqslant \left\| \nabla f(\boldsymbol{x}) \right\|_2^2, \ \forall \boldsymbol{x} \in \mathbb{R}^m,$$

where $f^\star = \min_{\boldsymbol{x} \in \mathbb{R}^m} f(\boldsymbol{x})$.

Assumptions 1(a)-(b) are standard in the distributed smooth optimization over directed networks (Nedić et al. , 2017; Pu et al. , 2021). The PL condition in Assumption 1(c) is satisfied in some important *non-convex* problems such as the policy optimization for LQR (Fazel et al. , 2018). It is strictly weaker than the strongly convex condition that is commonly used to derive the linear convergence of gradient-based methods (Nedić et al. 2017; Tian et al. , 2020). Particularly, the strong convexity implies the uniqueness of the minimizer, which is clearly not the case for the PL condition.

3.2 The APPG

We propose APPG to solve (1), which is given in Algorithm 1 from the view of a single node. Note that we

do not introduce any iteration index to emphasize the fact of fully asynchronous implementation.

Algorithm 1　The APPG — from the view of node i

(i) **Initialization:** Each node i selects local stepsize γ_i, initializes $\boldsymbol{x}_i$ as an arbitrary vector in $\mathbb{R}^m$, computes $\boldsymbol{g}_i = \boldsymbol{y}_i = \nabla f_i(\boldsymbol{x}_i)$, and creates local buffers $\mathcal{X}_i$ and $\mathcal{Y}_i$. Then it broadcasts $\tilde{\boldsymbol{x}}_i := \boldsymbol{x}_i$ and $\tilde{\boldsymbol{y}}_i := \boldsymbol{y}_i / |\mathcal{N}_{\text{out}}^i|$ to its out-neighbors.

(ii) **Repeat**

a) Keep receiving $\tilde{\boldsymbol{x}}_j$ and $\tilde{\boldsymbol{y}}_j$ from in-neighbors of node i and copy to $\mathcal{X}_i$ and $\mathcal{Y}_i$ respectively[①], until node i is activated to update.

b) Update $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ as

$$\begin{cases} \boldsymbol{x}_i \leftarrow \text{avg}(\mathcal{X}_i), \\ \boldsymbol{g}_i^- \leftarrow \boldsymbol{g}_i, \quad \boldsymbol{g}_i \leftarrow \nabla f_i(\boldsymbol{x}_i), \\ \boldsymbol{y}_i \leftarrow \text{sum}(\mathcal{Y}_i) + \boldsymbol{g}_i - \boldsymbol{g}_i^-, \\ \tilde{\boldsymbol{x}}_i \leftarrow \boldsymbol{x}_i - \gamma_i \boldsymbol{y}_i, \end{cases} \tag{2}$$

where $\text{avg}(\mathcal{X}_i)$ returns the average[②] of vectors in $\mathcal{X}_i$, $\text{sum}(\mathcal{Y}_i)$ takes the sum of vectors in $\mathcal{Y}_i$.

c) Broadcast $\tilde{\boldsymbol{x}}_i$ and $\tilde{\boldsymbol{y}}_i := \boldsymbol{y}_i / |\mathcal{N}_{\text{out}}^i|$ to all out-neighbors of i, after which *empty* both $\mathcal{X}_i$ and $\mathcal{Y}_i$.

(iii) **Until** a stopping criterion is satisfied. e. g., node i stops if $\|\boldsymbol{y}_i\|_2 < \varepsilon$ for some predefined $\varepsilon > 0$.

(iv) **Return** $\mathbf{x}_i$.

Our novel idea lies in the use of local buffers to achieve asynchronous updates. Particularly, a node just keeps receiving messages from its in-neighbors and storing them to its local buffers until it is activated to compute a new update via (2), which involves only summation and (weighted) average of data in the buffers, and a local gradient computation. Clearly, each buffer may contain zero, one or multiple receptions from the same in-neighbor, which is unavoidable in the fully asynchronous setting. Another feature of APPG is that the node computes a new update by using all messages in the buffers instead of only using the latest reception (Tian et al., 2020). This enables APPG to be robust to bounded communication delays, out of sequence issues and there is no need to use time-stamps (Tian et al., 2020). Note that APPG can also use only the latest information by carefully assigning weights to receptions. In addition, the local buffers are only conceptually needed and can be waived in implementation since the average and summation operators in (2) can be recursively computed. After (2), the node broadcasts the updated vectors to its out-neighbors and discards all the used messages. Such a process is repeated until a local stopping criterion is satisfied.

Different from synchronous or gossip-based algorithms, APPG does not require any global clock or coordination among neighbors, and each node does not wait for others for new updates. For example, a node can simply start to compute a new update once it completes the current one or receives a message. Thus, there is no deadlock problem. Moreover, nodes do not need to use the same stepsize.

3.3　The idea of APPG

This subsection aims to intuitively explain the linear convergence of the APPG. To this end, we first show how APPG works if nodes are forced to update synchronously. In this case, we can use a global iteration index k to record the update progress.

① In-neighbors include i itself, i. e., $\tilde{\boldsymbol{x}}_i$ and $\tilde{\boldsymbol{y}}_i$ are copied to $\mathcal{X}_i$ and $\mathcal{Y}_i$;

② The average can be weighted, e. g., one may assign higher weights to more recent messages to potentially improve the convergence rate in practice.

Let $\boldsymbol{X}(k)$, $\boldsymbol{Y}(k)$ and $\nabla\mathbf{f}(\boldsymbol{X}(k))$ be the stacked local states and gradients at the k-th iteration, i. e. ,

$$\boldsymbol{X}(k)=\left[\boldsymbol{x}_1(k),\boldsymbol{x}_2(k),\cdots,\boldsymbol{x}_n(k)\right]^{\mathrm{T}}\in\mathbb{R}^{n\times m},$$
$$\boldsymbol{Y}(k)=\left[\boldsymbol{y}_1(k),\boldsymbol{y}_2(k),\cdots,\boldsymbol{y}_n(k)\right]^{\mathrm{T}}\in\mathbb{R}^{n\times m},$$
$$\nabla\mathbf{f}(\boldsymbol{X}(k))=\left[\nabla f_1(\boldsymbol{x}_1(k)),\cdots,\nabla f_n(\boldsymbol{x}_n(k))\right]^{\mathrm{T}}\in\mathbb{R}^{n\times m},$$

and $\boldsymbol{\Gamma}=\mathrm{diag}(\gamma_1,\cdots,\gamma_n)$. Then, we obtain that

$$\boldsymbol{X}(k+1)=\boldsymbol{A}\left(\boldsymbol{X}(k)-\boldsymbol{\Gamma}\boldsymbol{Y}(k)\right), \tag{3}$$
$$\boldsymbol{Y}(k+1)=\boldsymbol{B}\boldsymbol{Y}(k)+\nabla\mathbf{f}(\boldsymbol{X}(k+1))-\nabla\mathbf{f}(\boldsymbol{X}(k)) \tag{4}$$

where the row-stochastic matrix $\boldsymbol{A}$ and column-stochastic matrix $\boldsymbol{B}$ result from avg(·) and sum(·) in (2), respectively. Clearly, (3)-(4) reduces to the algorithm(Xin et al. , 2018; Saadatniaki et al. ,2020;Pu et al. ,2021), which has been proved to converge linearly for strongly convex and Lipschitz smooth functions.

The key to the linear convergence of (3)-(4) is the introduction of $\boldsymbol{y}_i$ to distributedly track the gradient of f. To see it, we left multiply (4) with $\mathbf{1}^{\mathrm{T}}$, use the column-stochasticity of B and notice $\boldsymbol{Y}(0)=\nabla\mathbf{f}(\boldsymbol{X}(0))$. This implies that

$$\mathbf{1}^{\mathrm{T}}\boldsymbol{Y}(k)=\mathbf{1}^{\mathrm{T}}\nabla\mathbf{f}(\boldsymbol{X}(k)). \tag{5}$$

Now suppose that $\boldsymbol{X}(k)$ and $\boldsymbol{Y}(k)$ have already converged to $\boldsymbol{X}^{\infty}$ and $\boldsymbol{Y}^{\infty}$, respectively. It follows from (4) that $\boldsymbol{Y}^{\infty}=\boldsymbol{B}\boldsymbol{Y}^{\infty}$. Jointly with (5), we obtain that

$$\boldsymbol{y}_i^{\infty}=\pi_i^{B}\left(\mathbf{1}^{\mathrm{T}}\nabla\mathbf{f}(\boldsymbol{X}^{\infty})\right)^{\mathrm{T}}$$

where π^B is the Perron vector of $\boldsymbol{B}$, i. e. , $\boldsymbol{B}\pi^B=\pi^B$. Moreover, the row-stochasticity of $\boldsymbol{A}$ implies that $\boldsymbol{X}^{\infty}=\mathbf{1}(\boldsymbol{x}^{\infty})^{\mathrm{T}}$ and

$$\left(\mathbf{1}^{\mathrm{T}}\nabla\mathbf{f}(\boldsymbol{X}^{\infty})\right)^{\mathrm{T}}=\left(\mathbf{1}^{\mathrm{T}}\nabla\mathbf{f}\left(\mathbf{1}(\boldsymbol{x}^{\infty})^{\mathrm{T}}\right)\right)^{\mathrm{T}}=\nabla f(\boldsymbol{x}^{\infty}).$$

That is, $\boldsymbol{y}_i^{\infty}=\pi_i^{B}\nabla f(\boldsymbol{x}^{\infty})$. Substituting $\boldsymbol{X}^{\infty}$ and $\boldsymbol{Y}^{\infty}$ into (3) implies

$$\boldsymbol{X}(k+1)=\boldsymbol{A}\left(\boldsymbol{X}^{\infty}-\boldsymbol{\Gamma}\pi^{B}\nabla f(\boldsymbol{x}^{\infty})^{\mathrm{T}}\right). \tag{6}$$

Let π^A be Perron vector of $\boldsymbol{A}^{\mathrm{T}}$. We left multiply (6) with $(\pi^A)^{\mathrm{T}}$ and notice that $\boldsymbol{X}^{\infty}=\mathbf{1}(\boldsymbol{x}^{\infty})^{\mathrm{T}}$. Then,

$$\bar{\boldsymbol{x}}(k+1)=\boldsymbol{x}^{\infty}-(\pi^A)^{\mathrm{T}}\boldsymbol{\Gamma}\pi^{B}\nabla f(\boldsymbol{x}^{\infty})=\boldsymbol{x}^{\infty}-\rho\nabla f(\boldsymbol{x}^{\infty}), \tag{7}$$

where $\rho=(\pi^A)^{\mathrm{T}}\boldsymbol{\Gamma}\pi^B$ and $\bar{\boldsymbol{x}}(k+1)=(\pi^A)^{\mathrm{T}}\boldsymbol{X}(k+1)$. Clearly, (7) is a gradient descent update, which converges linearly under Assumption 1. It also shows that the limiting point $\boldsymbol{x}^{\infty}$ must be an optimal point $\boldsymbol{x}^{\star}$ and $\boldsymbol{y}_i$ converges to $\boldsymbol{y}_i^{\infty}=\pi_i^{B}\nabla f(\boldsymbol{x}^{\infty})=\pi_i^{B}\nabla f(\boldsymbol{x}^{\star})=0$. Moreover, the smaller the $\boldsymbol{y}_i$, the closer $\boldsymbol{x}_i$ to an optimal solution. Therefore, $\boldsymbol{y}_i$ can serve as a stopping criterion in Algorithm 1.

In the fully asynchronous setting, $\boldsymbol{y}_i$ plays a similar role. However, a theoretical understanding of APPG is much more complicated. The information delays make the key relation (5) invalid and the uncoordinated updates degrade the tracking performance of $\boldsymbol{y}_i$. In essence, APPG is a multi-timescale decision-making problem. To resolve it, we develop an augmented network approach to prove its linear convergence by associating each node with some virtual nodes, under which the asynchronous updates and communications of nodes are transformed into synchronous operations over the augmented network. Moreover, the key technique(Xin et al. , 2018; Saadatniaki et al. , 2020;Pu et al. , 2021) cannot be applied since the transformed system is time-varying and lacks important properties(e. g. the irreducibility of the weighting matrix). To this end, we further introduce an absolute probability sequence and a λ-sequence for the proof.

4 Linear convergence of APPG

Two assumptions on the asynchrony and communication delays are needed for the convergence of APPG.

Assumption 2(Bounded activation time interval)　Let t_i and t_i^+ be any two consecutive activation time of node i. There exist two positive constants $\underline{\tau}$ and $\overline{\tau}$ such that $0 < \underline{\tau} \leqslant \left| t_i^+ - t_i \right| \leqslant \overline{\tau} < +\infty$ for all $i \in \mathcal{V}$.

Assumption 2 is easily satisfied and desirable in practice. In fact, both the lower and upper bounds exist naturally since computing update consumes time and can be finished in finite time. If violated, e. g. , some node is broken, then the information from this node can no longer be accessed, and hence it is impossible to find an optimal solution of (1).

Assumption 3(Bounded transmission delays)　For any $(i,j) \in \mathcal{E}$, the transmission delay from node i to node j is bounded by a constant $\tau > 0$.

Note that transmission delays can be time-varying, and the parameters $\underline{\tau}$, $\overline{\tau}$ and τ are not needed for implementing APPG.

Let $\mathcal{T} = \{ t(k) \}_{k \geqslant 1}$ be an increasing sequence of updating time of all nodes, i. e. , $t \in \mathcal{T}$ if some node starts to update at time t. Denote the state of node i just before time $t(k)$ by $\boldsymbol{x}_i(k)$ and $\boldsymbol{y}_i(k)$. Then, the following lemma introduces an important quantity b to characterize the information delay in terms of $\mathcal{T}$.

Lemma 1　The following statements hold.

(i) Under Assumption 2, let $b_1 = (n-1)\left\lfloor \overline{\tau}/\underline{\tau} \right\rfloor + 1$. Each node is activated at least once within the time interval $\left(t(k), t(k+b_1) \right]$.

(ii) Under Assumptions 2-3, let $b_2 = n\left\lfloor \tau/\underline{\tau} \right\rfloor$ and $b = b_1 + b_2$. The information sent from node i at time $t(k)$ can be received by node j before time $t(k+b_2)$ and used to compute an update before time $t(k+b)$ for any k and $(i,j) \in \mathcal{E}$.

Proof　(i) Suppose that node i is not activated during the time interval $\left(t(p), t(q) \right]$, $p, q \in \mathbb{N}$ but is activated at $t(q+1)$. It follows from Assumption 2 that $t(q) - t(p) \leqslant \overline{\tau}$. Moreover, any other node can be activated at most $\left\lfloor \left(t(q) - t(p) \right) \big/ \underline{\tau} \right\rfloor \leqslant \left\lfloor \overline{\tau}/\underline{\tau} \right\rfloor$ times during the time interval $\left(t(p), t(q) \right]$, which implies $q - p \leqslant (n-1)\left\lfloor \overline{\tau}/\underline{\tau} \right\rfloor$. Hence, the first part of the result follows.

(ii) Suppose that node i sends information at time $t(p)$, $p \in \mathbb{N}$ and node j receives it in the time interval $\left(t(q), t(q+1) \right]$, $q \in \mathbb{N}$. It follows from Assumption 3 that $t(q) - t(p) \leqslant \tau$. Moreover, Assumption 2 implies that any node can be activated at most $\left\lfloor \tau/\underline{\tau} \right\rfloor$ times during the time interval $\left[t(p), t(q) \right]$, i. e. , $q - p + 1 \leqslant n\left\lfloor \tau/\underline{\tau} \right\rfloor$, and hence $q + 1 \leqslant p + n\left\lfloor \tau/\underline{\tau} \right\rfloor$. The result follows by letting $p = k$. Jointly with Lemma 1(i), the rest of proof follows immediately.

The quantity b measures the largest possible information staleness caused by asynchrony and delays. For synchronous algorithms without communication delays, we have $b = 1$. For cyclic updating modes where node i updates at $t(nl + i), \forall l \in \mathbb{N}$, we can set $b = n$. Note that a larger b does not imply that the algorithm updates slower in wall-clock time. For example, b is small in synchronous algorithms but $t(k+1) - t(k)$ may be large due to idle time, while in fully asynchronous algorithms b is often large but $t(k+b) - t(k)$ can be small. Our main theoretical result is given below.

Theorem 1　Suppose that Assumptions 1-3 hold. Let $\overline{\gamma} = \max\limits_i \gamma_i$ and $\underline{\gamma} = \min\limits_i \gamma_i$. If $\underline{\gamma} > 0$ and $\overline{\gamma} < \sqrt{\underline{\gamma}} \left(\dfrac{\alpha \log_2 (1-\theta)^{-1}}{16 n^{b+2} (b+2)^2 \beta} \right)^4$, then there exists an optimal solution $\boldsymbol{x}^\star \in \mathcal{X}^\star$ such that

$$\left\| \boldsymbol{x}_i(k) - \boldsymbol{x}^\star \right\|_2 = \mathcal{O}(\lambda^k), \quad \left\| \boldsymbol{y}_i(k) \right\|_2 = \mathcal{O}(\lambda^k), \ \forall i \in \mathcal{V} \tag{8}$$

where

$$\max\left\{(1-\theta)^{\frac{1}{n(b+3)^2\log_2 n}},\ 1-\frac{1}{8}\underline{\gamma}\,\alpha n^{-b}\right\}\leqslant\lambda<1, \tag{9}$$

α,β are given in Assumption 1, b is defined in Lemma 1, and $\theta=(nb)^{-nb}$.

Theorem 1 shows that the local decision vector of each node in APPG converges to the same optimal solution at a linear rate, which is characterized in terms of key parameters n,b,α,β and γ_i. The quantity θ in Theorem 1 reflects the speed of achieving consensus and depends on the network topology and asynchrony. Let $v(k)\coloneqq\max\limits_i \boldsymbol{x}_i(k)-\min\limits_i \boldsymbol{x}_i(k)$ be the maximum difference among nodes at time $t(k)$, then we can show that $v(k+nb)\leqslant(1-\theta)v(k)$ when applying APPG on the consensus problem with $f_i=0,\forall i$. Clearly, the convergence rate of APPG cannot exceed the consensus speed, which is reflected in the first term of the left-hand-side of (9).

To obtain a deterministic convergence rate, Theorem 1 adopts the *worst-case* point of view under the setting that: (a) The underlying network is heavily unbalanced, i. e. , the differences between the numbers of in-neighbors and out-neighbors of a node is large. (b) Some nodes compute much faster than the others (match the lower and upper bounds of Assumption 2, respectively). (c) Communication delays are based on the upper bound in Assumption 3. The values of b and θ in Lemma 1 and Theorem 1 are given in this case. Thus, the theoretical rate is expected to be very conservative, but the practical performance is empirically much better. For the synchronous delay-free case, we have $b=1$ and $\theta=n^{-n}$, and the convergence rate in Theorem 1 reduces to $\mathcal{O}\left(\max\left\{(1-n^{-n}),1-\frac{\alpha\underline{\gamma}}{8n}\right\}^k\right)$, where the first term in $\max(\cdot)$ matches the consensus result in Olshevsky et al. (2011, Theorem 8. 1), and the second term is close to the push-pull method (Pu et al. , 2021, Remark 6).

Note that the virtual counter k in (8) increases by one no matter which node updates, and hence more nodes generally lead to faster increase of k. To some extent, this suggests a linear speedup efficiency of APPG, which is confirmed via experiments in Section 7.

5 The time-varying augmented system

This section develops our time-varying augmented network approach for the convergence analysis.

5. 1 Construction of the augmented digraph

Let $\mathcal{T}_i\subseteq\mathcal{T}$ be the sequence of activation time of node i, i. e. , $t\in\mathcal{T}_i$ if node i computes an update at time t. Then, it is clear that

$$\left[\boldsymbol{x}_i(k+1),\boldsymbol{y}_i(k+1),\boldsymbol{g}_i(k+1),\boldsymbol{g}_i^-(k+1)\right]=\left[\boldsymbol{x}_i(k),\boldsymbol{y}_i(k),\boldsymbol{g}_i(k),\boldsymbol{g}_i^-(k)\right],\ \forall t(k)\notin\mathcal{T}_i.$$

To handle bounded time-varying transmission delays and asynchrony, we associate each node i with two types of virtual nodes, and each type has b virtual nodes, where b is given in Lemma 1(ii). We denote the above two types of virtual nodes by $\{v_{x,i}^{(1)},\cdots,v_{x,i}^{(b)}\}$ and $\{v_{y,i}^{(1)},\cdots,v_{y,i}^{(b)}\}$, respectively. We call the first type virtual nodes x-type nodes, which is to deal with the staleness of the state $\tilde{\boldsymbol{x}}_i,i\in\mathcal{V}$. The second type with subscript y is called y-type nodes, which is to handle the staleness of $\tilde{\boldsymbol{y}}_i,i\in\mathcal{V}$. Then, we construct an augmented time-varying digraph $\tilde{\mathcal{G}}(k)=\left(\tilde{\mathcal{V}},\tilde{\mathcal{E}}(k)\right)$ to represent the communication topology of all these nodes at time $t(k)$, where $\tilde{\mathcal{V}}$ contains $n(2b+1)$ nodes, including n nodes of $\mathcal{G}$ and $2nb$ virtual nodes.

The edge set $\tilde{\mathcal{E}}(k)$ is described as follows. We first note that there is no edge between any x-type node and any y-type node. For the x-type virtual nodes, the edges $\left(i,v_{x,i}^{(1)}\right),\left(v_{x,i}^{(1)},v_{x,i}^{(2)}\right),\cdots,\left(v_{x,i}^{(b-2)},v_{x,i}^{(b-1)}\right)$ and $\left(v_{x,i}^{(b-1)},v_{x,i}^{(b)}\right)$ always exist for all $k\in\mathbb{N}$ and $i\in\mathcal{V}$, see Fig. 3. If $(i,j)\in\mathcal{E}$ in $\mathcal{G}$ and node j receives $\tilde{\boldsymbol{x}}_i$ at time $t(k)$, then some of the edges $\left(v_{x,i}^{(1)},j\right),\left(v_{x,i}^{(2)},j\right),\cdots,\left(v_{x,i}^{(b)},j\right)$ and (i,j) are included in $\mathcal{E}(k)$ (c.f. Fig. 4(a)), depending on the transmission delay of the received message. For example, if node j receives $\tilde{\boldsymbol{x}}_i(t-u)$ and $\tilde{\boldsymbol{x}}_i(t-v)$ at time $t(k)$ for some $u,v>1$, then $\left(v_{x,i}^{(u-1)},j\right),\left(v_{x,i}^{(v-1)},j\right)\in\tilde{\mathcal{E}}(k)$. If $u=1$, which means that there is no communication delay, then $(i,j)\in\tilde{\mathcal{E}}(k)$.

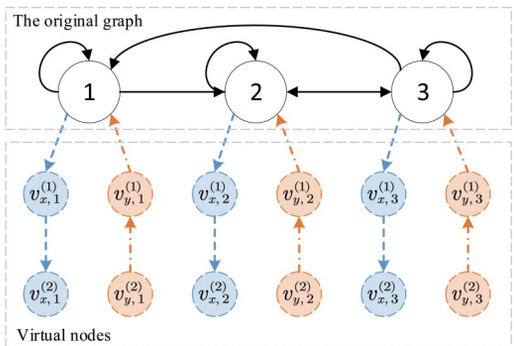

Fig. 3　An augmented graph with virtual nodes to address delays of the original graph

Fig. 4(a) illustrates such an augmented graph[①] at some time $t(k)$, where node 1 uses the 2-steps delayed information $\tilde{\boldsymbol{x}}_3(k-3)$ and the latest information $\tilde{\boldsymbol{x}}_1(k-1)$ to compute $\boldsymbol{x}_1(k)$, and hence $\left(v_{x,3}^{(2)},1\right)\in\tilde{\mathcal{E}}(k)$ Node 2 uses $\tilde{\boldsymbol{x}}_2(k-1)$ and the 1-step delayed information $\tilde{\boldsymbol{x}}_1(k-2)$ and $\tilde{\boldsymbol{x}}_3(k-2)$ to compute $\boldsymbol{x}_2(k)$. Node 3 use the latest information $\tilde{\boldsymbol{x}}_2(k-1)$ and $\tilde{\boldsymbol{x}}_3(k-1)$ to compute $\boldsymbol{x}_3(k)$. The corresponding row-stochastic matrix $\tilde{\boldsymbol{A}}(k)$ is as demonstrated in Fig.4(b).

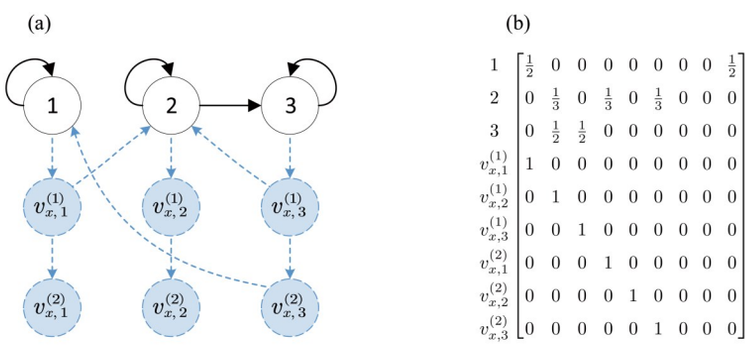

Fig. 4　(a) The topology of the x-type virtual nodes in the augmented graph at some time $t(k)$; (b) The corresponding row-stochastic matrix

The topology of the y-type virtual nodes is similarly developed with reversed edge directions (c.f. Fig. 5(a)), which is the main motivation of using two types of virtual nodes. Firstly, edges $\left(v_{y,j}^{(1)},j\right)$, $\left(v_{y,j}^{(2)},v_{y,j}^{(1)}\right)$,⋯, and $\left(v_{y,j}^{(b)},v_{y,j}^{(b-1)}\right)$ are always included in $\tilde{\mathcal{E}}(k)$ with the reversed directions of x-type nodes, see Fig. 3. Secondly, if $(i,j)\in\mathcal{E}$ in $\mathcal{G}$ and $k\in\mathcal{T}_i$, then only one edge in edges $\left(i,v_{y,j}^{(1)}\right)$, $\left(i,v_{y,j}^{(2)}\right)$,⋯,$\left(i,v_{y,j}^{(nb)}\right)$ and (i,j) is included in $\mathcal{E}(k)$, which also depends on the transmission delay of $\tilde{\boldsymbol{y}}_i$ sent from node i to node j. At time $t(k)$, suppose that node i sends $\boldsymbol{y}_i(k)$ to node j, which is received at $t(k+u)$ for $u>1$, then $\left(i,v_{y,j}^{(u-1)}\right)\in\tilde{\mathcal{E}}(k)$ and the delay is $u-1$. Similarly, if there is no communication delay, i. e. , $u=1$, then $(i,j)\in\tilde{\mathcal{E}}(k)$. Fig. 5(a) illustrates such an augmented graph at some time $t(k)$, which represents that node 1 sends $\tilde{\boldsymbol{y}}_1(k)$ to node 2 and node 2 use it at $t(k+3)$ to compute $\boldsymbol{y}_2(k+3)$. Other edges can be interpreted similarly. The corresponding column-stochastic matrix $\tilde{\boldsymbol{B}}(k)$ is as

[①] The idea of adding virtual nodes to address asynchrony or delays was firstly adopted in Nedić et al.(2010) to study consensus problems and in Zhang et al.(2020a); Tian et al.(2020); Assran et al.(2021) for distributed optimization. Nevertheless, Nedić et al.(2010); Zhang et al.(2020a); Assran et al.(2021) use only x-type virtual nodes and involve division operators. We further introduce y-type nodes to accommodate linear update rules. Tian et al.(2020) associates virtual nodes to original edges rather than original nodes. Since there are generally more edges than nodes in a strongly connected graph and the theoretical convergence rate becomes slower as the number of virtual nodes increases, our analysis may potentially be sharper than Tian et al.(2020).

depicted in Fig. 5(b).

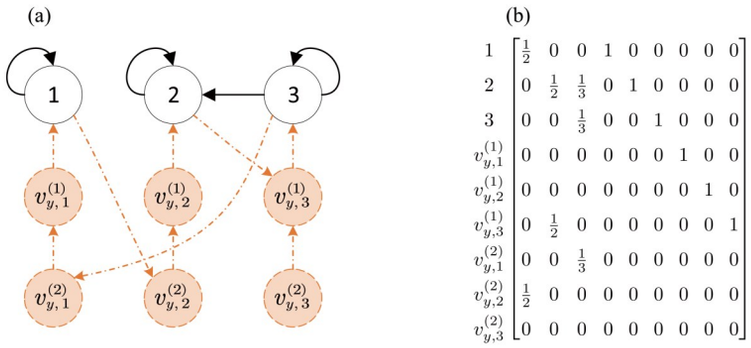

Fig. 5 (a) The topology of the y-type virtual nodes in theaugmented graph at some time $t(k)$; (b) The corresponding column-stochastic matrix

We provide a simple example to visualize the augmented graph approach. Consider that node i sends $\tilde{\boldsymbol{x}}_i(k)$ and $\tilde{\boldsymbol{y}}_i(k)$ to node j at time $t(k)$, and node j receives it at time $t(k+2)$, i.e., the delay is 1. In the augmented graph, this can be viewed as node i directly sends $\tilde{\boldsymbol{x}}_i(k)$ to the virtual node $v_{x,i}^{(1)}$, and sends $\tilde{\boldsymbol{y}}_i(k)$ to the virtual node $v_{y,j}^{(1)}$ at time $t(k)$. Nodes $v_{x,i}^{(1)}$ and $v_{y,j}^{(1)}$ respectively receive $\tilde{\boldsymbol{x}}_i(k)$ and $\tilde{\boldsymbol{y}}_i(k)$ at time $t(k+1)$, and immediately send them to node j at time $t(k+1)$. Finally, node j receives $\tilde{\boldsymbol{x}}_i(k)$ and $\tilde{\boldsymbol{y}}_i(k)$ at time $t(k+2)$. Clearly, all non-virtual nodes in $\tilde{\mathcal{G}}$ receive the same information as that in $\mathcal{G}$ and hence their updates appear to be synchronous and delay-free.

Under the time-varying augmented digraph, we are able to rewrite the APPG in a compact form.

5.2 A compact form of the APPG over the augmented digraph

Let $\boldsymbol{x}_i^{(u)}(k)$ and $\boldsymbol{y}_i^{(u)}(k)$ denote the states of virtual node $v_{x,i}^{(u)}$ and $v_{y,i}^{(u)}$ just after time $t(k)$, and $\tilde{n} = bn$. Then, APPG can be rewritten in a compact form over $\tilde{\mathcal{G}}(k)$,

$$\begin{cases} \tilde{\boldsymbol{X}}(k+1) = \tilde{\boldsymbol{A}}(k)\left(\tilde{\boldsymbol{X}}(k) - \tilde{\boldsymbol{\Gamma}}\boldsymbol{I}_k^a\tilde{\boldsymbol{Y}}(k)\right), \\ \tilde{\boldsymbol{Y}}(k+1) = \tilde{\boldsymbol{B}}(k)\tilde{\boldsymbol{Y}}(k) + \boldsymbol{I}_k^a\left(\nabla(k+1) - \nabla(k)\right) = \tilde{\boldsymbol{B}}(k)\tilde{\boldsymbol{Y}}(k) + \nabla(k+1) - \nabla(k), \end{cases} \tag{10}$$

where $\tilde{\boldsymbol{\Gamma}} = \boldsymbol{\Gamma} \otimes \boldsymbol{I} \in \mathbb{R}^{\tilde{n}\times\tilde{n}}$,

$$\begin{cases} \tilde{\boldsymbol{X}}(k) = \left[\boldsymbol{X}(k);\ \boldsymbol{X}^{(1)}(k);\ \cdots;\ \boldsymbol{X}^{(b-1)}(k)\right] \in \mathbb{R}^{\tilde{n}\times m}, \\ \boldsymbol{X}^{(u)}(k) = \left[\boldsymbol{x}_1^{(u)}(k), \cdots, \boldsymbol{x}_n^{(u)}(k)\right]^{\mathrm{T}}, \\ \tilde{\boldsymbol{Y}}(k) = \left[\boldsymbol{Y}(k);\ \boldsymbol{Y}^{(1)}(k);\ \cdots;\ \boldsymbol{Y}^{(b-1)}(k)\right] \in \mathbb{R}^{\tilde{n}\times m}, \\ \boldsymbol{Y}^{(u)}(k) = \left[\boldsymbol{y}_1^{(u)}(k), \cdots, \boldsymbol{y}_n^{(u)}(k)\right]^{\mathrm{T}}, \\ \nabla(k) = \left[\nabla\mathbf{f}\left(\boldsymbol{X}(k)\right);\ \boldsymbol{0}_{(\tilde{n}-n)\times m}\right], \end{cases} \tag{11}$$

the initial condition is $\tilde{\boldsymbol{X}}(0) = \left[\boldsymbol{X}(0); \boldsymbol{0}_{(\tilde{n}-n)\times m}\right]$, $\tilde{\boldsymbol{Y}}(0) = \nabla(0)$, and $\tilde{\boldsymbol{A}}(k), \tilde{\boldsymbol{B}}(k), \boldsymbol{I}_k^a \in \mathbb{R}^{\tilde{n}\times\tilde{n}}$ are

$$\left[\tilde{\boldsymbol{A}}(k)\right]_{ij} = \begin{cases} \dfrac{1}{\left|\mathcal{X}_i(k)\right|}, & \text{if } i, v \in \mathcal{V},\ j = nu + v,\ t(k+1) \in \mathcal{T}_i, \\ & \text{and node } i \text{ receives } \boldsymbol{x}_v(k-u) \text{ at } t(k+1), \\ 1, & \text{if } i \in \mathcal{V},\ t(k+1) \notin \mathcal{T}_i \text{ and } j = i, \\ 1, & \text{if } i \notin \mathcal{V} \text{ and } j = i - n\ , \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

$$\left[\tilde{\boldsymbol{B}}(k)\right]_{ji}=\begin{cases}\dfrac{1}{\left|\mathcal{N}_{\text{out}}^{i}\right|}, & \text{if } i,v\in\mathcal{V},\ j=nu+v,\ t(k+1)\in\mathcal{T}_i,\\ & \text{and node } v \text{ receives } \boldsymbol{x}_i(k) \text{ at } t(k+u),\\ 1, & \text{if } i\in\mathcal{V},\ t(k+1)\notin\mathcal{T}_i \text{ and } j=i,\\ 1, & \text{if } i\notin\mathcal{V} \text{ and } j=i-n,\\ 0, & \text{otherwise,}\end{cases}\tag{13}$$

$$\left[\boldsymbol{I}_k^a\right]_{ij}=\begin{cases}1, & \text{if } i=j,\ i\in\mathcal{V},\ \text{and } t(k+1)\in\mathcal{T}_i,\\ 0, & \text{otherwise,}\end{cases}$$

where $\left|\mathcal{X}_i(k)\right|$ is the number of elements in the buffer $\mathcal{X}_i$ at time $t(k+1)$.

An example of $\tilde{\boldsymbol{A}}(k)$ and $\tilde{\boldsymbol{B}}(k)$ is illustrated in Fig. 4(b) and Fig. 5(b), respectively. $\boldsymbol{I}_k^a$ is a diagonal matrix with the i-th diagonal element be 1 if node i is activated at time $t(k+1)$. The second equality in (10) follows from $\nabla(k+1)=\nabla(k)$ for any $i\in\left\{i\middle|\left[\boldsymbol{I}_k^a\right]_{ii}=0\right\}$. An important fact is that $\tilde{\boldsymbol{A}}(k)$ is a row-stochastic matrix and $\tilde{\boldsymbol{B}}(k)$ is a column-stochastic matrix by the use of two types virtual nodes, but they may not be irreducible. Moreover,

$$\mathbf{1}_{\tilde{n}}^{\mathrm{T}}\tilde{\boldsymbol{Y}}(k)=\mathbf{1}_{\tilde{n}}^{\mathrm{T}}\nabla(k)=\mathbf{1}_n^{\mathrm{T}}\nabla\mathbf{f}\left(\boldsymbol{X}(k)\right),\tag{14}$$

which is obtained by left multiplying the second equality of (10) with $\mathbf{1}_{\tilde{n}}^{\mathrm{T}}$.

Note that (10) generates the same sequence of the states $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ as that of APPG. Hence, it is sufficient to study the convergence of $\tilde{\boldsymbol{X}}(k)$ and $\tilde{\boldsymbol{Y}}(k)$ in (10). To this end, we define

$$\begin{cases}\boldsymbol{\Phi}_t^A(k)=\tilde{\boldsymbol{A}}(k+t-1)\tilde{\boldsymbol{A}}(k+t-2)\cdots\tilde{\boldsymbol{A}}(k+1)\tilde{\boldsymbol{A}}(k),\\ \boldsymbol{\Phi}_t^B(k)=\tilde{\boldsymbol{B}}(k+t-1)\tilde{\boldsymbol{B}}(k+t-2)\cdots\tilde{\boldsymbol{B}}(k+1)\tilde{\boldsymbol{B}}(k),\end{cases}\tag{15}$$

where $k,t\in\mathbb{N}$, and we adopt the convention that $\boldsymbol{\Phi}_0^A(k)=\boldsymbol{\Phi}_0^B(k)=\boldsymbol{I}$ and $\boldsymbol{\Phi}_t^A(k)=\boldsymbol{\Phi}_t^B(k)=\mathbf{0}$ for any $k\in\mathbb{N}$ and $t<0$.

The following lemma states that $\boldsymbol{\Phi}_t^A(k)$ and $\boldsymbol{\Phi}_t^B(k)$ linearly converge to rank-one matrices.

Lemma 2 Under Assumptions 1-3, the following statements are in force.

(i) Let θ be the minimum nonzero element of $\boldsymbol{\Phi}_{nb}^A(k)$ and $\boldsymbol{\Phi}_{nb}^B(k)$. It holds that $\tilde{n}^{-nb}\leqslant\theta<1$, where b is defined in Lemma 1(ii).

(ii) There exist stochastic vectors $\boldsymbol{\phi}_t^A(k)$ and $\boldsymbol{\phi}_t^{\mathrm{B}}(k)$ such that

$$\left\|\boldsymbol{\Phi}_t^A(k)-\mathbf{1}\boldsymbol{\phi}_t^A(k)^{\mathrm{T}}\right\|_2\leqslant 2\rho^t,\quad \left\|\boldsymbol{\Phi}_t^B(k)-\boldsymbol{\phi}_t^B(k)\mathbf{1}^{\mathrm{T}}\right\|_2\leqslant 2\rho^t,$$

for all $k,t\in\mathbb{N}$, where $\rho=(1-\theta)^{1/\tilde{n}}<1$.

(iii) $\sum_{j=1}^{n}\left[\boldsymbol{\Phi}_t^B(0)\right]_{ij}\geqslant n^{-b},\ \forall i\in\mathcal{V},t\in\mathbb{N}$.

Proof Part (i): In view of (12) and (13), the minimum nonzero element of $\tilde{\boldsymbol{A}}(k)$ and $\tilde{\boldsymbol{B}}(k)$ is greater than $1/\tilde{n}$ for all k. Thus, the minimum nonzero element of $\boldsymbol{\Phi}_t^A(k)$ and $\boldsymbol{\Phi}_t^B(k)$ is greater than $\tilde{n}^{-t}$ for any $t>0$.

Part (ii): In view of Lemma 1, both $\boldsymbol{\Phi}_b^A(k)$ and $\boldsymbol{\Phi}_b^B(k)$ are indecomposable for all k, and thus $\boldsymbol{\Phi}_{nb}^A(k)$ and $\boldsymbol{\Phi}_{nb}^B(k)$ have positive columns. Then, the proof is similar with that Lemma 5 (Nedić et al., 2010) or Lemma 3 (Nedić et al. ,2015), and we omit it for saving space.

Part (iii): We study two cases separately. If $t<b$, then

$$\left[\boldsymbol{\Phi}_t^B(0)\right]_{ii}\geqslant\left[\tilde{\boldsymbol{B}}(t-1)\right]_{ii}\left[\tilde{\boldsymbol{B}}(t-2)\right]_{ii}\cdots\left[\tilde{\boldsymbol{B}}(0)\right]_{ii}\geqslant(1/n)^b.$$

If $t\geqslant b$, it follows from a similar argument with the Lemma 2(ii) in Nedić et al. (2010) that $\left[\boldsymbol{\Phi}_b(k)\right]_{ij}\geqslant n^{-b}$ for all $i\in\mathcal{V}$ and $j\in\tilde{\mathcal{V}}$. Then,

$$\left[\boldsymbol{\Phi}_{b+1}(k-1)\right]_{ij}=\sum_{u=1}^{\tilde{n}}\left[\boldsymbol{\Phi}_{b}(k)\right]_{iu}\left[\tilde{\boldsymbol{B}}(k)\right]_{uj}\geqslant n^{-b}\sum_{u=1}^{\tilde{n}}\left[\tilde{\boldsymbol{B}}(k)\right]_{uj}\geqslant n^{-b}.$$

where the last inequality follows from the column-stochasticity of $\tilde{\boldsymbol{B}}(k)$. The desired result is obtained by induction.

The following lemma is a direct result of Lemma 2, which specifies μ and $\tilde{t}$ used later.

Lemma 3 Under assumptions of Lemma 2, let $\mu=\frac{1}{4}n^{-(b+1)}$ and $\tilde{t}=\left\lfloor nb\log_{1-\theta}\left(\frac{\mu}{2}\right)\right\rfloor+1$. If $\lambda\geqslant(1-\theta)^{\frac{1}{n(b+3)^2\log_2 n}}$, then we have $\lambda^{\tilde{t}}\geqslant\frac{1}{2}$ and

$$\left\|\boldsymbol{\Phi}_{\tilde{t}}^{A}(k)-\mathbf{1}\boldsymbol{\phi}_{\tilde{t}}^{A}(k)^{\mathrm{T}}\right\|_{2}\leqslant\mu,\qquad\left\|\boldsymbol{\Phi}_{\tilde{t}}^{B}(k)-\boldsymbol{\phi}_{\tilde{t}}^{B}(k)\mathbf{1}^{\mathrm{T}}\right\|_{2}\leqslant\mu.\tag{16}$$

Proof In view of Lemma 2(ii), it can be readily checked that (16) satisfies. We have

$$\lambda^{\tilde{t}}\geqslant(1-\theta)^{\frac{\tilde{t}}{n(b+3)^2\log_2 n}}\geqslant\left(\frac{\mu}{2}\right)^{\frac{b+1}{(b+3)^2\log_2 n}}=n^{-\frac{(b+1)^2}{(b+3)^2\log_2 n}}8^{-\frac{b+1}{(b+3)^2\log_2 n}}=2^{-\frac{(b+1)^2}{(b+3)^2}}2^{-\frac{3(b+1)}{(b+3)^2\log_2 n}}=2^{-\frac{(b+1)^2\log_2 n+3(b+1)}{(b+3)^2\log_2 n}}\geqslant\frac{1}{2}.$$

Finally, we introduce the *absolute probability sequence* (Touri, 2012).

Lemma 4 (Theorem 4. 2 in Touri (2012)) For a sequence of row-stochastic matrices $\{\boldsymbol{A}(k)\}$, there exists a sequence of stochastic vectors $\{\boldsymbol{\pi}(k)\}$ satisfying

$$\boldsymbol{\pi}(k+1)^{\mathrm{T}}\boldsymbol{A}(k)=\boldsymbol{\pi}(k)^{\mathrm{T}},\quad\forall k\in\mathbb{N}.\tag{17}$$

$\{\boldsymbol{\pi}(k)\}$ is called an absolute probability sequence of $\{\boldsymbol{A}(k)\}$.

In the sequel, we use $\boldsymbol{\pi}(k)\in\mathbb{R}^{\tilde{n}}$ to denote an absolute probability sequence of $\tilde{\boldsymbol{A}}(k)$, which implies that $\boldsymbol{\pi}(k+t)^{\mathrm{T}}\boldsymbol{\Phi}_{t}^{A}(k)=\boldsymbol{\pi}(k)^{\mathrm{T}},\forall k,t\in\mathbb{N}$.

6 Proof of Theorem 1 via LMIs

Under the augmented time-varying digraph, we are ready to prove Theorem 1.

6. 1 Outline of the proof

As Nedić et al. (2017), for a nonnegative sequence $\{p(k)\}$, we define that

$$p^{\lambda,k}=\sup_{t\in\mathbb{N},t\leqslant k}\frac{p(t)}{\lambda^{t}},\tag{18}$$

for $\lambda\in(0,1)$. We call $\{p^{\lambda,k}\}$ the λ-sequence of $p(k)$. Clearly, if $p^{\lambda,k}$ is uniformly bounded by some constant c, then $p(k)\leqslant c\lambda^{k}$ for all k. Our method to prove Theorem 1 is to show the boundedness of $p^{\lambda,k},k\in\mathbb{N}$ in (18) for some nonnegative sequences $p(k)$. Under Assumptions 1-3 and (10), the proof of Theorem 1 relies on the following four lemmas, whose proofs are given in next subsections.

Lemma 5 Let

$$\boldsymbol{Q}(k)=\boldsymbol{I}_{\tilde{n}}-\mathbf{1}_{\tilde{n}}\boldsymbol{\pi}(k-1)^{\mathrm{T}},\qquad\left\|\tilde{\boldsymbol{X}}(k)\right\|_{Q}=\left\|\boldsymbol{Q}(k)\tilde{\boldsymbol{X}}(k)\right\|_{\mathrm{F}},$$

and $\left\|\tilde{\boldsymbol{X}}\right\|_{Q}^{\lambda,k}$ be the λ-sequence of $\left\|\tilde{\boldsymbol{X}}(k)\right\|_{Q}$. If $\lambda^{\tilde{t}}\geqslant 1/2$, then

$$\left\|\tilde{\boldsymbol{X}}\right\|_{Q}^{\lambda,k}\leqslant 4\tilde{\gamma}\tilde{n}\tilde{t}\left\|\tilde{\boldsymbol{Y}}\right\|_{Q}^{\lambda,k}+c_{1},\tag{19}$$

where $\left\|\tilde{\boldsymbol{Y}}\right\|_{\mathrm{F}}^{\lambda,k}$ is the λ-sequence of $\left\|\tilde{\boldsymbol{Y}}(k)\right\|_{\mathrm{F}}$, $\tilde{t}$ is defined in Lemma 3, and c_{1} is a constant given in (31).

$\left\|\tilde{\boldsymbol{X}}(k)\right\|_{Q}$ in Lemma 5 is a weighted difference among nodes' states $\boldsymbol{x}_{i}(k)$, and (19) bounds it by the gradient estimate $\left\|\tilde{\boldsymbol{Y}}(k)\right\|_{\mathrm{F}}$.

Lemma 6 Let

$$\boldsymbol{v}(k+1)=\tilde{\boldsymbol{B}}(k)\boldsymbol{v}(k),\quad\boldsymbol{v}(0)=\left[\mathbf{1}_{n};\mathbf{0}_{\tilde{n}-n}\right],\quad\boldsymbol{V}(k)=\operatorname{diag}\left(\boldsymbol{v}(k)\right),\quad\boldsymbol{Y}_{V}(k)=\boldsymbol{V}(k)^{\dagger}\tilde{\boldsymbol{Y}}(k),$$

where $\boldsymbol{V}(k)^{\dagger}$ is the pseudo inverse of $\boldsymbol{V}(k)$, i. e. ,

$$\left[\boldsymbol{V}(k)^{\dagger}\right]_{ij}=\begin{cases}1/\left[\boldsymbol{V}(k)\right]_{ii}, & \text{if } i=j \text{ and} \left[\boldsymbol{V}(k)\right]_{ii}>0,\\ 0, & \text{otherwise.}\end{cases}$$

Let $\tilde{\boldsymbol{I}}(k)=\boldsymbol{V}(k)\boldsymbol{V}(k)^{\dagger}$, $\tilde{\mathbf{1}}(k)=\tilde{\boldsymbol{I}}(k)\mathbf{1}_{\tilde{n}}$ and

$$\boldsymbol{S}(k)=\tilde{\boldsymbol{I}}(k)-\frac{1}{n}\tilde{\mathbf{1}}(k)\boldsymbol{v}(k)^{\mathrm{T}},\qquad \left\|\boldsymbol{Y}_{\boldsymbol{V}}(k)\right\|_{S}=\left\|\boldsymbol{S}(k)\boldsymbol{Y}_{\boldsymbol{V}}(k)\right\|_{\mathrm{F}}. \tag{20}$$

Define the corresponding λ-sequence $\left\|\boldsymbol{Y}_{\boldsymbol{V}}\right\|_{S}^{\lambda,k}$ of $\left\{\left\|\boldsymbol{Y}_{\boldsymbol{V}}(k)\right\|_{S}\right\}$. If $\lambda^{\tilde{t}}\geqslant 1/2$, then

$$\left\|\boldsymbol{Y}_{\boldsymbol{V}}\right\|_{S}^{\lambda,k}\leqslant 4\beta n^{b+1}b\tilde{t}\left(2\left\|\tilde{\boldsymbol{X}}\right\|_{Q}^{\lambda,k}+\overline{\gamma}\left\|\tilde{\boldsymbol{Y}}\right\|_{\mathrm{F}}^{\lambda,k}\right)+c_{2}, \tag{21}$$

where β is given in Assumption 1, b is defined in Lemma 1(ii), $\tilde{t}$ is defined in Lemma 3, and c_2 is given by (38).

Similarly, $\left\|\boldsymbol{Y}_{\boldsymbol{V}}(k)\right\|_{S}$ measures the differences between the weighted gradient estimates of different nodes, which is bounded by $\left\|\tilde{\boldsymbol{X}}(k)\right\|_{Q}$ and $\left\|\tilde{\boldsymbol{Y}}(k)\right\|_{\mathrm{F}}$.

Lemma 7　Let

$$\boldsymbol{x}_{\boldsymbol{\pi}}(k)=\boldsymbol{\pi}(k)^{\mathrm{T}}\tilde{\boldsymbol{X}}(k). \tag{22}$$

Define $\tilde{f}^{\lambda,k}$ be the λ-sequence of $\{\sqrt{f\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k)\right)-f^{\star}}\}$. If $\overline{\gamma}\leqslant\dfrac{1}{4nb\beta}$ and $\lambda^{b}\geqslant 1-\dfrac{1}{8}\alpha\underline{\gamma}n^{-b}$, then

$$\tilde{f}^{\lambda,k}\leqslant\frac{16n^{\frac{b}{2}}b}{\alpha\sqrt{\underline{\gamma}}}\left(n\beta\left\|\tilde{\boldsymbol{X}}\right\|_{Q}^{\lambda,k}+\left\|\boldsymbol{Y}_{\boldsymbol{V}}\right\|_{S}^{\lambda,k}+n\beta\overline{\gamma}\left\|\tilde{\boldsymbol{Y}}\right\|_{\mathrm{F}}^{\lambda,k}\right)+c_{3}, \tag{23}$$

where α,β are given in Assumption 1, b is defined in Lemma 1(ii), and c_3 is a constant given in (44).

Intuitively, $\boldsymbol{x}_{\boldsymbol{\pi}}(k)$ is a weighted average of $\boldsymbol{x}_{i}(k), i\in\mathcal{V}$, and $f\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k)\right)-f^{\star}$ is the optimality gap. Eq. (22) shows that the square root of the optimality gap can be bounded by $\left\|\tilde{\boldsymbol{X}}\right\|_{Q}^{\lambda,k}$, $\left\|\boldsymbol{Y}_{\boldsymbol{V}}\right\|_{S}^{\lambda,k}$, and $\left\|\tilde{\boldsymbol{Y}}\right\|_{\mathrm{F}}^{\lambda,k}$.

Lemma 8　With the above-defined $\left\|\tilde{\boldsymbol{X}}\right\|_{Q}^{\lambda,k}$, $\left\|\boldsymbol{Y}_{\boldsymbol{V}}\right\|_{S}^{\lambda,k}$ and $\tilde{f}^{\lambda,k}$, it holds that

$$\left\|\tilde{\boldsymbol{Y}}\right\|_{\mathrm{F}}^{\lambda,k}\leqslant n\sqrt{b}\,\beta\left\|\tilde{\boldsymbol{X}}\right\|_{Q}^{\lambda,k}+n\left\|\boldsymbol{Y}_{\boldsymbol{V}}\right\|_{S}^{\lambda,k}+\frac{\beta n\sqrt{2b}}{\sqrt{\alpha}}\tilde{f}^{\lambda,k}, \tag{24}$$

where β and b are given in Assumption 1 and Lemma 1, respectively.

Proof of Theorem 1

Note that λ and $\gamma_i, i\in\mathcal{V}$ defined in Theorem 1 satisfy all the conditions on λ in the above four lemmas and the condition on γ_i in Lemma 7. Thus, (19), (21), (23) and (24) hold. Let

$$\boldsymbol{e}(k)=\left[\left\|\tilde{\boldsymbol{X}}\right\|_{Q}^{\lambda,k},\left\|\boldsymbol{Y}_{\boldsymbol{V}}\right\|_{S}^{\lambda,k},\tilde{f}^{\lambda,k},\left\|\tilde{\boldsymbol{Y}}\right\|_{\mathrm{F}}^{\lambda,k}\right]^{\mathrm{T}},\qquad \boldsymbol{c}=\left[c_{1},c_{2},c_{3},0\right]^{\mathrm{T}}.$$

Combining (19), (21), (23), and (24), we obtain that for all $k\in\mathbb{N}$,

$$\boldsymbol{e}(k)\leqslant\boldsymbol{M}\boldsymbol{e}(k)+\boldsymbol{c}, \tag{25}$$

where $\leqslant$ is the element-wise inequality and $\boldsymbol{M}$ is a nonnegative matrix

$$\boldsymbol{M}=\begin{bmatrix}0 & 0 & 0 & 4nb\tilde{t}\,\overline{\gamma}\\ 8\beta n^{b+1}b\tilde{t} & 0 & 0 & 4\beta n^{b+1}b\tilde{t}\,\overline{\gamma}\\ \dfrac{16n^{b/2+1}b\beta}{\alpha\sqrt{\underline{\gamma}}} & \dfrac{16n^{b/2}b}{\alpha\sqrt{\underline{\gamma}}} & 0 & \dfrac{16n^{b/2+1}b\beta\,\overline{\gamma}}{\alpha\sqrt{\underline{\gamma}}}\\ n\beta\sqrt{b} & n & \dfrac{n\beta\sqrt{2b}}{\sqrt{\alpha}} & 0\end{bmatrix}.$$

It follows from (25) that if the spectral radius $\varrho(\boldsymbol{M})$ of $\boldsymbol{M}$ is strictly less than 1, then $\left\|\tilde{\boldsymbol{X}}\right\|_{Q}^{\lambda,k}$, $\left\|\boldsymbol{Y}_{\boldsymbol{V}}\right\|_{S}^{\lambda,k}$ and $\tilde{f}^{\lambda,k}$ are all bounded for all $k\in\mathbb{N}$.

Define a transformation matrix $\boldsymbol{T} = \mathrm{diag}\left(\left[1,1,\sqrt{\gamma},\sqrt{\gamma}\right]\right)$. Then, $\boldsymbol{TMT}^{-1}$ can be arbitrarily close to a strictly lower triangular matrix by making $\bar{\gamma}/\sqrt{\gamma}$ sufficiently small, which implies that $\varrho(\boldsymbol{M}) = \varrho\left(\boldsymbol{TMT}^{-1}\right) < 1$. To obtain an upper bound of $\bar{\gamma}$, we notice that $\varrho\left(\boldsymbol{M}^4\right) = \mathcal{O}\left(\bar{\gamma}/\sqrt{\gamma}\right)$. By letting $\varrho\left(\boldsymbol{M}^4\right) < 1$, we have

$$\bar{\gamma} \ < \frac{\alpha^4\sqrt{\gamma}}{16^4 n^{4(b+1)} b^4 \beta^4 \tilde{t}^2}.$$

The bound in Theorem 1 is obtained by substituting $\tilde{t}$ in Lemma 3.

Define $\bar{c} \coloneqq \dfrac{\max\{c_1,c_2,c_3\}}{1-\varrho(\boldsymbol{M})} < \infty$, where c_1,c_2 and c_3 are given in (31), (38) and (44), respectively. It follows from (25) that $\left\|\tilde{\boldsymbol{X}}\right\|_Q^{\lambda,k} \leqslant \bar{c}$ and $\tilde{f}^{\lambda,k} \leqslant \bar{c}$, $\forall k$. We have

$$\begin{aligned}\left\|\boldsymbol{x}_i(k) - \Pi_{\mathcal{X}^*}\left(\boldsymbol{x}_{\pi}(k-1)\right)\right\|_2 &\leqslant \left\|\boldsymbol{x}_i(k) - \boldsymbol{x}_{\pi}(k-1)\right\|_2 + \left\|\boldsymbol{x}_{\pi}(k-1) - \Pi_{\mathcal{X}^*}\left(\boldsymbol{x}_{\pi}(k-1)\right)\right\|_2 \\ &\leqslant \left\|\tilde{\boldsymbol{X}}(k)\right\|_Q + \sqrt{2/\alpha}\sqrt{f\left(\boldsymbol{x}_{\pi}(k-1)\right) - f^*},\end{aligned}$$

where the last inequality used the equivalence between the Polyak-Łojasiewicz condition and the quadratic growth condition (Karimi et al., 2016, Theorem 2), i. e., $f(\boldsymbol{x}) - f^* \geqslant \dfrac{\alpha}{2}\left\|\boldsymbol{x} - \Pi_{\mathcal{X}^*}(\boldsymbol{x})\right\|_2^2, \forall \boldsymbol{x} \in \mathbb{R}^m$. Let $\boldsymbol{x}^*(k) = \Pi_{\mathcal{X}^*}\left(\boldsymbol{x}_{\pi}(k-1)\right)$ and $\left\|\boldsymbol{x}_i - \boldsymbol{x}^*\right\|_2^{\lambda,k}$ be the λ-sequence of $\left\{\left\|\boldsymbol{x}_i(k) - \boldsymbol{x}^*(k)\right\|_2\right\}$. We have

$$\left\|\boldsymbol{x}_i - \boldsymbol{x}^*\right\|^{\lambda,k} \leqslant \left\|\tilde{\boldsymbol{X}}\right\|_Q^{\lambda,k} + \sqrt{2/\alpha}\ \tilde{f}^{\lambda,k} \leqslant \left(1 + \sqrt{2/\alpha}\right)\bar{c},\ \forall i.$$

Combined with the boundedness of $\left\|\tilde{\boldsymbol{Y}}\right\|_{\mathrm{F}}^{\lambda,k}$, the result in Theorem 1 follows by the definition of λ-sequence.

6.2 Two useful propositions

We establish two important results in this subsection. The first one shows a property of λ-sequence and the second one recalls the contraction relation of gradient methods.

Proposition 1 Let $\{p(k)\}, \{q(k)\}$ be nonnegative sequences satisfying

$$p(t+j) \leqslant rp(t) + \sum_{i=0}^{j-1} q(t+i), \tag{26}$$

where $r \in [0,1)$. If we choose λ such that $\lambda^j \in (r,1)$, then the λ-sequences $p^{\lambda,k}$ and $q^{\lambda,k}$ in (18) satisfy

$$p^{\lambda,k} \leqslant \frac{j}{\lambda^j - r} q^{\lambda,k} + c_{\lambda},\ \forall k \in \mathbb{N},$$

where $c_{\lambda} = \dfrac{\lambda^j}{\lambda^j - r}\sum_{t=1}^{m}\lambda^{-t}p(t)$ is a constant.

It follows from (26) that

$$\lambda^{-(t+j)}p(t+j) \leqslant \frac{r}{\lambda^j}\lambda^{-t}p(t) + \sum_{i=0}^{j-1}\frac{1}{\lambda^{j-i}}\lambda^{-(t+i)}q(t+i),\ \forall t = 1,\cdots,k.$$

This gives k inequalities by selecting $t = 1,\cdots,k$. On the other hand, we have $\lambda^{-t}p(t) \leqslant \lambda^{-t}p(t)$, which gives another j inequalities by selecting $t = 1,\cdots,j$. Take the maximum on both sides of these $k+j$ inequalities and use the definition of λ-sequence, we obtain

$$p^{\lambda,k+j} \leqslant \frac{r}{\lambda^j}p^{\lambda,k} + \max\left\{q^{\lambda,k+j}\sum_{i=0}^{j-1}\frac{1}{\lambda^{j-i}},\ \max_{t=1,\cdots,j}\lambda^{-t}p(t)\right\} \leqslant \frac{r}{\lambda^j}p^{\lambda,k+j} + q^{\lambda,k+j}\sum_{i=0}^{j-1}\frac{1}{\lambda^{j-i}} + \sum_{t=1}^{j}\lambda^{-t}p(t). \tag{27}$$

If $\lambda^j \in (r,1)$, then (27) implies

$$\begin{aligned}p^{\lambda,k+j} &\leqslant \frac{\lambda^j\sum_{i=0}^{j-1}\frac{1}{\lambda^{j-i}}}{\lambda^j - r}q^{\lambda,k+j} + \frac{\lambda^j}{\lambda^j - r}\sum_{t=1}^{j}\lambda^{-t}p(t) = \frac{1-\lambda^j}{(\lambda^j - r)(1-\lambda)}q^{\lambda,k+j} + \frac{\lambda^j}{\lambda^j - r}\sum_{t=1}^{j}\lambda^{-t}p(t) \\ &\leqslant \frac{j}{\lambda^j - r}q^{\lambda,k+j} + \frac{\lambda^j}{\lambda^j - r}\sum_{t=1}^{j}\lambda^{-t}p(t),\end{aligned}$$

for all $k+j\in\mathbb{N}$, where we have used that $1-\lambda^j\leqslant j(1-\lambda)$. The result is obtained immediately.

For any nonnegative sequences $\{p(k)\}$, $\{q(k)\}$, letting $r(k)=p(k)+q(k)$, it holds that $r^{\lambda,k}\leqslant p^{\lambda,k}+q^{\lambda,k}$, which can be easily checked by definition.

The following proposition shows the convergence rate of a perturbed gradient descent method for minimizing functions satisfying the PL condition. As a special case, it recovers the linear convergence rate of the standard gradient descent method.

Proposition 2 Suppose that f is β-Lipschitz smooth and satisfies the Polyak-Łojasiewicz condition in Assumption 1(iii). Let $\eta\in\left(0,\dfrac{1}{2\beta}\right)$, $\sigma=1-\alpha\eta\left(1-2\eta\beta\right)<1$, and $\boldsymbol{x}^+=\boldsymbol{x}-\eta\nabla f(\boldsymbol{x})+\boldsymbol{\varepsilon}$. Then

$$f(\boldsymbol{x}^+)-f^\star\leqslant\sigma\left(f(\boldsymbol{x})-f^\star\right)+\left(2\eta^{-1}+\beta\right)\|\boldsymbol{\varepsilon}\|_2^2,\ \forall\boldsymbol{x}\in\mathbb{R}^n.$$

Proof It follows from the β-Lipschitz smoothness that

$$\begin{aligned}f(\boldsymbol{x}^+)&\leqslant f(\boldsymbol{x})+\nabla f(\boldsymbol{x})^{\mathrm{T}}\left(-\eta\nabla f(\boldsymbol{x})+\boldsymbol{\varepsilon}\right)+\frac{\beta}{2}\left\|-\eta\nabla f(\boldsymbol{x})+\boldsymbol{\varepsilon}\right\|_2^2\\&\leqslant f(\boldsymbol{x})-\eta\left\|\nabla f(\boldsymbol{x})\right\|_2^2+\frac{\eta}{2}\left\|\nabla f(\boldsymbol{x})\right\|_2^2+\frac{2}{\eta}\|\boldsymbol{\varepsilon}\|_2^2+\beta\eta^2\left\|\nabla f(\boldsymbol{x})\right\|_2^2+\beta\|\boldsymbol{\varepsilon}\|_2^2\\&\leqslant f(\boldsymbol{x})-\eta\left(\frac{1}{2}-\eta\beta\right)\left\|\nabla f(\boldsymbol{x})\right\|_2^2+\left(\frac{2}{\eta}+\beta\right)\|\boldsymbol{\varepsilon}\|_2^2.\end{aligned}$$

Applying the Polyak-Łojasiewicz inequality on $\left\|\nabla f(\boldsymbol{x})\right\|_2^2$ implies the desired result:

$$\begin{aligned}f(\boldsymbol{x}^+)-f^\star&\leqslant f(\boldsymbol{x})-f^\star-2\alpha\eta\left(\frac{1}{2}-\eta\beta\right)\left(f(\boldsymbol{x})-f^\star\right)+\left(\frac{2}{\eta}+\beta\right)\|\boldsymbol{\varepsilon}\|_2^2\\&\leqslant\left(1-\alpha\eta\left(1-2\eta\beta\right)\right)\left(f(\boldsymbol{x})-f^\star\right)+\left(2\eta^{-1}+\beta\right)\|\boldsymbol{\varepsilon}\|_2^2.\end{aligned}$$

We end this subsection with two inequalities which will be frequently used later. For any $\boldsymbol{A},\boldsymbol{B}\in\mathbb{R}^{n\times n}$,

$$\|\boldsymbol{AB}\|_{\mathrm{F}}\leqslant\|\boldsymbol{A}\|_2\|\boldsymbol{B}\|_{\mathrm{F}},\qquad\|\boldsymbol{A}\|_2\leqslant\sqrt{\|\boldsymbol{A}\|_\infty\|\boldsymbol{A}\|_1},$$

and $\|\boldsymbol{A}\|_2\leqslant\|\boldsymbol{A}\|_{\mathrm{F}}\leqslant\sqrt{n}$ for any row-stochastic matrix $\boldsymbol{A}$.

6.3 Proof of Lemma 5

Let $\tilde{t}$ be defined in Lemma 3. It follows from (10) and (15) that

$$\begin{aligned}\left\|\tilde{\boldsymbol{X}}(k+\tilde{t})\right\|_Q&=\left\|\boldsymbol{Q}(k+\tilde{t})\tilde{\boldsymbol{X}}(k+\tilde{t})\right\|_{\mathrm{F}}\\&\leqslant\left\|\boldsymbol{Q}(k+\tilde{t})\boldsymbol{\Phi}_{\tilde{t}}^{\boldsymbol{A}}(k)\tilde{\boldsymbol{X}}(k)\right\|_{\mathrm{F}}+\sum_{t=0}^{\tilde{t}-1}\left\|\boldsymbol{Q}(k+\tilde{t})\boldsymbol{\Phi}_{\tilde{t}-t}^{\boldsymbol{A}}(k+t)\tilde{\boldsymbol{\Gamma}}\boldsymbol{I}_{k+t}^{a}\tilde{\boldsymbol{Y}}(k+t)\right\|_{\mathrm{F}}\end{aligned}\tag{28}$$

By the stochastic vector $\boldsymbol{\phi}_{\tilde{t}}^{\boldsymbol{A}}(k)$ in Lemma 3, (28) implies that

$$\begin{aligned}\left\|\tilde{\boldsymbol{X}}(k+\tilde{t})\right\|_Q&\leqslant\left\|\boldsymbol{Q}(k+\tilde{t})\left(\boldsymbol{\Phi}_{\tilde{t}}^{\boldsymbol{A}}(k)-\mathbf{1}\boldsymbol{\phi}_{\tilde{t}-1}^{\boldsymbol{A}}(k)^{\mathrm{T}}\right)\boldsymbol{Q}(k)\tilde{\boldsymbol{X}}(k)\right\|_{\mathrm{F}}\\&\quad+\bar{\gamma}\sum_{t=0}^{\tilde{t}-1}\left\|\boldsymbol{Q}(k+\tilde{t})\right\|_2\left\|\boldsymbol{\Phi}_{\tilde{t}-t}^{\boldsymbol{A}}(k+t)\right\|_2\left\|\boldsymbol{I}_{k+t}^{a}\tilde{\boldsymbol{Y}}(k+t)\right\|_{\mathrm{F}}\\&\leqslant\left\|\boldsymbol{Q}(k+\tilde{t})\right\|_2\left\|\boldsymbol{\Phi}_{\tilde{t}}^{\boldsymbol{A}}(k)-\mathbf{1}\boldsymbol{\phi}_{\tilde{t}}^{\boldsymbol{A}}(k)^{\mathrm{T}}\right\|_2\left\|\boldsymbol{Q}(k)\tilde{\boldsymbol{X}}(k)\right\|_{\mathrm{F}}+\tilde{n}\bar{\gamma}\sum_{t=0}^{\tilde{t}-1}\left\|\tilde{\boldsymbol{Y}}(k+t)\right\|_{\mathrm{F}}\\&\leqslant\sqrt{\tilde{n}}\,\mu\left\|\tilde{\boldsymbol{X}}(k)\right\|_Q+\bar{\gamma}\tilde{n}\sum_{t=0}^{\tilde{t}-1}\left\|\tilde{\boldsymbol{Y}}(k+t)\right\|_{\mathrm{F}},\end{aligned}\tag{29}$$

where $\mu=\dfrac{1}{4}n^{-(b+1)}<1$ is given in Lemma 3, the first inequality used the relation

$$\begin{aligned}\boldsymbol{Q}(k+t)\left(\boldsymbol{A}-\mathbf{1}\boldsymbol{r}^{\mathrm{T}}\right)\boldsymbol{Q}(k)&=\boldsymbol{Q}(k+t)\boldsymbol{A}\boldsymbol{Q}(k)-\boldsymbol{Q}(k+t)\mathbf{1}\boldsymbol{r}^{\mathrm{T}}\boldsymbol{Q}(k)\\&=\boldsymbol{Q}(k+t)\left(\boldsymbol{A}-\mathbf{1}_{\tilde{n}}\boldsymbol{\pi}(k-1)^{\mathrm{T}}\right)-\boldsymbol{Q}(k+t)\left(\mathbf{1}_{\tilde{n}}\boldsymbol{r}^{\mathrm{T}}-\mathbf{1}_{\tilde{n}}\boldsymbol{\pi}(k-1)^{\mathrm{T}}\right)\\&=\boldsymbol{Q}(k+t)\boldsymbol{A}-\boldsymbol{Q}(k+t)\mathbf{1}_{\tilde{n}}\boldsymbol{r}^{\mathrm{T}}=\boldsymbol{Q}(k+t)\boldsymbol{A},\end{aligned}\tag{30}$$

for any row-stochastic matrix $\boldsymbol{A}$ and stochastic vector $\boldsymbol{r}$, and the last inequality follows from Lemma 3, $\left\|\boldsymbol{Q}(k)\right\|_2 \leqslant \sqrt{\tilde{n}}$ and $\left\|\boldsymbol{\Phi}_{\tilde{t}}^{A}(k)\right\|_2 \leqslant \sqrt{\tilde{n}}$, $\forall k$.

Note that $\sqrt{\tilde{n}}\,\mu \leqslant 1/4$ and $\lambda^{\tilde{t}} \geqslant 1/2$. By Proposition 1 and (29), we obtain that $\left\|\tilde{\boldsymbol{X}}\right\|_Q^{\lambda,k} \leqslant 4\overline{\gamma}\,\tilde{n}\tilde{t}\left\|\tilde{\boldsymbol{Y}}\right\|_Q^{\lambda,k} + c_1$ where

$$c_1 = 4\overline{\gamma}\,\tilde{n}\sum_{t=1}^{\tilde{t}}\lambda^{-t}\left\|\tilde{\boldsymbol{X}}(t)\right\|_Q. \tag{31}$$

6.4 Proof of Lemma 6

Let $\boldsymbol{v}(k) = \left[v_1(k),\cdots,v_{\tilde{n}}(k)\right]^{\mathrm{T}}$ and $\mathcal{I}_V(k) = \left\{i \,|\, v_i(k) = \left[\boldsymbol{V}(k)\right]_{ii} = 0\right\}$. Note that $v_i(k) \geqslant n^{-b}$ for all $i \in \mathcal{V}, k \in \mathbb{N}$ from Lemma 2(iii). It can be shown that the i-th row of $\tilde{Y}(k)$ is $\mathbf{0}_m^{\mathrm{T}}$ for all $i \in \mathcal{I}_V(k)$, and thus $\tilde{\boldsymbol{Y}}(k) = \boldsymbol{V}(k)\boldsymbol{Y}_V(k)$.

Let $\boldsymbol{R}(k) = \nabla(k+1) - \nabla(k)$. It then follows from (10) that

$$\boldsymbol{Y}_V(k+1) = \tilde{\boldsymbol{B}}_V(k)\boldsymbol{Y}_V(k) + \boldsymbol{V}(k+1)^{\dagger}\boldsymbol{R}(k), \tag{32}$$

where $\tilde{\boldsymbol{B}}_V(k) = \boldsymbol{V}(k+1)^{\dagger}\tilde{\boldsymbol{B}}(k)\boldsymbol{V}(k)$ and one can prove that each row except the i-th ($i \in \mathcal{I}_V(k)$) row of $\tilde{\boldsymbol{B}}_V(k)$ has row sum 1 using similar arguments as in Lemma 4 of Nedić et al. (2016;2017). By $\nabla(k)$ in (11) and Assumption 1, we have

$$\left\|\boldsymbol{R}(k)\right\|_{\mathrm{F}} = \left\|\nabla(k+1) - \nabla(k)\right\|_{\mathrm{F}} \leqslant \beta\left\|\boldsymbol{X}(k+1) - \boldsymbol{X}(k)\right\|_{\mathrm{F}} \leqslant \beta\left\|\tilde{\boldsymbol{X}}(k+1) - \tilde{\boldsymbol{X}}(k)\right\|_{\mathrm{F}}. \tag{33}$$

Notice that

$$\begin{aligned}\left\|\tilde{\boldsymbol{X}}(k+1) - \tilde{\boldsymbol{X}}(k)\right\|_{\mathrm{F}} &= \left\|\tilde{\boldsymbol{A}}(k)\left(\tilde{\boldsymbol{X}}(k) - \tilde{\boldsymbol{\Gamma}}\boldsymbol{I}_k^a\tilde{\boldsymbol{Y}}(k)\right) - \tilde{\boldsymbol{X}}(k)\right\|_{\mathrm{F}} \leqslant \left\|\left(\tilde{\boldsymbol{A}}(k) - \boldsymbol{I}\right)\boldsymbol{Q}(k)\tilde{\boldsymbol{X}}(k)\right\|_{\mathrm{F}} + \overline{\gamma}\sqrt{n}\left\|\tilde{\boldsymbol{Y}}(k)\right\|_{\mathrm{F}} \\ &\leqslant 2\sqrt{n}\left\|\tilde{\boldsymbol{X}}(k)\right\|_Q + \overline{\gamma}\sqrt{n}\left\|\tilde{\boldsymbol{Y}}(k)\right\|_{\mathrm{F}},\end{aligned} \tag{34}$$

where the first inequality follows from the row-stochasticity of $\tilde{\boldsymbol{A}}(k)$, and the last inequality is from that $\left\|\tilde{\boldsymbol{A}}(k) - \boldsymbol{I}\right\|_2 \leqslant \sqrt{n} + 1 \leqslant 2\sqrt{n}$. By Combining (33) and (34), we obtain

$$\left\|\boldsymbol{R}(k)\right\|_{\mathrm{F}} \leqslant 2\beta\sqrt{n}\left\|\tilde{\boldsymbol{X}}(k)\right\|_Q + \beta\overline{\gamma}\sqrt{n}\left\|\tilde{\boldsymbol{Y}}(k)\right\|_{\mathrm{F}}. \tag{35}$$

To analyze the sequence $\{\boldsymbol{Y}_V(k)\}$, we define

$$\begin{aligned}\tilde{\boldsymbol{\Phi}}_t(k) &\coloneqq \tilde{\boldsymbol{B}}_V(k+t-1)\tilde{\boldsymbol{B}}_V(k+t-2)\cdots\tilde{\boldsymbol{B}}_V(k+1)\tilde{\boldsymbol{B}}_V(k) = \boldsymbol{V}(k+t)^{\dagger}\left(\prod_{l=t-1}^{1}\tilde{\boldsymbol{B}}(k+l)\tilde{\boldsymbol{I}}(k+l)\right)\tilde{\boldsymbol{B}}(k)\boldsymbol{V}(k) \\ &= \boldsymbol{V}(k+t)^{\dagger}\boldsymbol{\Phi}_t^{B}(k)\boldsymbol{V}(k).\end{aligned}$$

where

$$\left[\tilde{\boldsymbol{I}}(k)\right]_{ij} = \left[\boldsymbol{V}(k)\boldsymbol{V}(k)^{\dagger}\right]_{ij} = \begin{cases}1, & \text{if } i = j,\ i \notin \mathcal{I}_V(k), \\ 0, & \text{otherwise.}\end{cases}$$

and the last equality follows from that $\tilde{\boldsymbol{I}}(k+1)\tilde{\boldsymbol{B}}(k)\boldsymbol{V}(k) = \tilde{\boldsymbol{B}}(k)\boldsymbol{V}(k)$, $\forall k \in \mathbb{N}$, where we used the fact that $\left[\tilde{\boldsymbol{B}}(k)\boldsymbol{V}(k)\mathbf{1}_{\tilde{n}}\right] = v_i(k+1) = 0$ for any $i \in \mathcal{I}_V(k+1)$, and thus the i-th row of $\tilde{\boldsymbol{B}}(k)\boldsymbol{V}(k)$ is $\mathbf{0}_m^{\mathrm{T}}$.

By Lemma 2 and Lemma 3, it holds that $\boldsymbol{\Phi}_t^{B}(k)$ can be written as

$$\boldsymbol{\Phi}_t^{B}(k) = \boldsymbol{\phi}_t^{B}(k)\mathbf{1}^{\mathrm{T}} + \Delta\boldsymbol{\Phi}_t(k),$$

where $\left\|\Delta\boldsymbol{\Phi}_t(k)\right\|_2 \leqslant 2\rho^t$ and $\left\|\Delta\boldsymbol{\Phi}_{\tilde{t}}(k)\right\|_2 \leqslant \mu$. Hence,

$$\boldsymbol{v}(k+t) = \boldsymbol{\Phi}_t^{B}(k)\boldsymbol{v}(k) = \boldsymbol{\phi}_t^{B}(k)\mathbf{1}^{\mathrm{T}}\boldsymbol{v}(k) + \Delta\boldsymbol{\Phi}_t(k)\boldsymbol{v}(k) = \boldsymbol{\phi}_t^{B}(k)\mathbf{1}^{\mathrm{T}}\boldsymbol{v}(0) + \Delta\boldsymbol{\Phi}_t(k)\boldsymbol{v}(k) = n\boldsymbol{\phi}_t^{B}(k) + \Delta\boldsymbol{\Phi}_t(k)\boldsymbol{v}(k),$$

which implies that

$$\boldsymbol{\phi}_t^B(k)=\frac{1}{n}\big(\boldsymbol{v}(k+t)-\Delta\boldsymbol{\Phi}_t(k)\boldsymbol{v}(k)\big). \tag{36}$$

Let $\tilde{\mathbf{1}}(k)=\tilde{\boldsymbol{I}}(k)\mathbf{1}$ and $\boldsymbol{C}_t(k)=\boldsymbol{V}(k+t)^{\dagger}\Delta\boldsymbol{\Phi}_t(k)\Big(\boldsymbol{I}-\frac{1}{\tilde{n}}\boldsymbol{v}(k)\mathbf{1}_{\tilde{n}}^{\mathrm{T}}\Big)\boldsymbol{V}(k)$. It follows from (35)-(36) that $\tilde{\boldsymbol{\Phi}}_t(k)=\frac{1}{n}\tilde{\mathbf{1}}(k+t)\boldsymbol{v}(k)^{\mathrm{T}}+\boldsymbol{C}_t(k)$.

Now we turn to the sequence $\{\boldsymbol{Y}_V(k)\}$. It follows from (20) and (32) that

$$\begin{aligned}\big\|\boldsymbol{Y}_V(k+\tilde{t})\big\|_S&=\big\|\boldsymbol{S}(k+\tilde{t})\boldsymbol{Y}_V(k+\tilde{t})\big\|_{\mathrm{F}}\\&\leqslant\big\|\boldsymbol{S}(k+\tilde{t})\tilde{\boldsymbol{\Phi}}_{\tilde{t}}(k)\boldsymbol{Y}_V(k)\big\|_{\mathrm{F}}+\sum_{t=1}^{\tilde{t}}\big\|\boldsymbol{S}(k+\tilde{t})\tilde{\boldsymbol{\Phi}}_{\tilde{t}-t}(k+t)\boldsymbol{V}(k+t)^{\dagger}\boldsymbol{R}(k+t-1)\big\|_{\mathrm{F}}\\&\leqslant\big\|\boldsymbol{S}(k+\tilde{t})\tilde{\boldsymbol{\Phi}}_{\tilde{t}}(k)\boldsymbol{Y}_V(k)\big\|_{\mathrm{F}}+\sum_{t=1}^{\tilde{t}}\big\|\boldsymbol{S}(k+\tilde{t})\boldsymbol{V}(k+\tilde{t})^{\dagger}\boldsymbol{\Phi}_{\tilde{t}-t}^B(k+t)\boldsymbol{R}(k+t-1)\big\|_{\mathrm{F}}\\&\leqslant\big\|\boldsymbol{S}(k+\tilde{t})\tilde{\boldsymbol{\Phi}}_{\tilde{t}}(k)\boldsymbol{Y}_V(k)\big\|_{\mathrm{F}}+\sqrt{\tilde{n}}\,n^b\sum_{t=1}^{\tilde{t}}\big\|\boldsymbol{R}(k+t-1)\big\|_{\mathrm{F}},\end{aligned} \tag{37}$$

where we used in the last inequality that $\big\|\boldsymbol{S}(k)\boldsymbol{V}(k)^{\dagger}\big\|_2=\Big\|\boldsymbol{V}(k)^{\dagger}-\frac{1}{n}\tilde{\mathbf{1}}(k)\tilde{\mathbf{1}}(k)^{\mathrm{T}}\Big\|_2\leqslant n^b,\forall k$ from Lemma 2(iii).

Similar to (30), we have $\boldsymbol{S}(k+\tilde{t})\tilde{\boldsymbol{\Phi}}_{\tilde{t}}(k)=\boldsymbol{S}(k+\tilde{t})\Big(\tilde{\boldsymbol{\Phi}}_{\tilde{t}}(k)-\frac{1}{n}\tilde{\mathbf{1}}(k+t)\boldsymbol{v}(k)^{\mathrm{T}}\Big)\boldsymbol{S}(k)$, and

$$\begin{aligned}\Big\|\boldsymbol{S}(k+\tilde{t})\Big(\tilde{\boldsymbol{\Phi}}_{\tilde{t}}(k)-\frac{1}{n}\tilde{\mathbf{1}}(k+t)\Big)\Big\|_2&=\big\|\boldsymbol{S}(k+\tilde{t})\boldsymbol{C}_t(k)\big\|_2\\&\leqslant\Big\|\boldsymbol{V}(k+t)^{\dagger}-\frac{1}{n}\tilde{\mathbf{1}}\tilde{\mathbf{1}}^{\mathrm{T}}\Big\|_2\big\|\Delta\boldsymbol{\Phi}_{\tilde{t}}(k)\big\|_2\Big\|\Big(\boldsymbol{I}-\frac{1}{n}\boldsymbol{v}(k)\mathbf{1}_{\tilde{n}}^{\mathrm{T}}\Big)\boldsymbol{V}(k)\Big\|_2\\&\leqslant n^{b+1}\mu=\frac{1}{4}.\end{aligned}$$

Following a similar argument as (29), it follows from (35) and (37) that

$$\big\|\boldsymbol{Y}_V(k+\tilde{t})\big\|_S\leqslant\frac{1}{4}\big\|\boldsymbol{Y}_V(k)\big\|_S+\beta n^{b+1}b\sum_{t=0}^{\tilde{t}-1}\Big(2\big\|\tilde{\boldsymbol{X}}(k+t)\big\|_Q+\overline{\gamma}\big\|\tilde{\boldsymbol{Y}}(k+t)\big\|_{\mathrm{F}}\Big).$$

By Proposition 1, we have for any $\lambda^{\tilde{t}}\geqslant 1/2$ that

$$\big\|\boldsymbol{Y}_V\big\|_S^{\lambda,k}\leqslant 4\beta n^{b+1}b\tilde{t}\Big(2\big\|\tilde{\boldsymbol{X}}\big\|_Q^{\lambda,k}+\overline{\gamma}\big\|\tilde{\boldsymbol{Y}}\big\|_{\mathrm{F}}^{\lambda,k}\Big)+c_2,$$

where

$$c_2=4\beta n^{b+1}b\sum_{t=1}^{\tilde{t}}\lambda^{-t}\Big(2\big\|\tilde{\boldsymbol{X}}(t)\big\|_Q+\overline{\gamma}\big\|\tilde{\boldsymbol{X}}(t)\big\|_Q\Big). \tag{38}$$

6.5　Proof of Lemma 7

It follows from (10), (17) and (22) that

$$\begin{aligned}\boldsymbol{x}_{\boldsymbol{\pi}}(k+b)&=\boldsymbol{\pi}(k+b)^{\mathrm{T}}\tilde{\boldsymbol{X}}(k+b)\\&=\boldsymbol{\pi}(k+b)^{\mathrm{T}}\boldsymbol{\Phi}_b^A(k)\tilde{\boldsymbol{X}}(k)-\boldsymbol{\pi}(k+b)^{\mathrm{T}}\sum_{t=0}^{b-1}\boldsymbol{\Phi}_{b-t}^A(k+t)\boldsymbol{I}_{k+t}^a\tilde{\boldsymbol{\Gamma}}\boldsymbol{V}(k+t)\boldsymbol{Y}_V(k+t)\\&=\boldsymbol{\pi}(k)^{\mathrm{T}}\tilde{\boldsymbol{X}}(k)-\frac{1}{n}\sum_{t=0}^{b-1}\boldsymbol{\eta}_k(t)\mathbf{1}_{\tilde{n}}^{\mathrm{T}}\nabla(k+t)-\sum_{t=0}^{b-1}\boldsymbol{r}_k(t)\Big(\boldsymbol{Y}_V(k+t)-\frac{1}{n}\tilde{\mathbf{1}}(k+t)\mathbf{1}_{\tilde{n}}^{\mathrm{T}}\nabla(k+t)\Big),\end{aligned} \tag{39}$$

where $\boldsymbol{V}(k)$ and $\tilde{\mathbf{1}}(k)$ are defined in Lemma 6, $\boldsymbol{r}_k(t)=\boldsymbol{\pi}(k+b)^{\mathrm{T}}\boldsymbol{\Phi}_{b-t}^A(k+t+1)\boldsymbol{I}_{k+t}^a\tilde{\boldsymbol{\Gamma}}\boldsymbol{V}(k+t)$ and $\boldsymbol{\eta}_k(t)=\boldsymbol{r}_k(t)\tilde{\mathbf{1}}(k+t)$. Let $\overline{\boldsymbol{\eta}}_k\coloneqq\sum_{t=0}^{b-1}\boldsymbol{\eta}_k(t)$. We have

$$\underline{\gamma}\, n^{-b} \leqslant \boldsymbol{\pi}(k+b)^{\mathrm{T}} \sum_{t=0}^{b-1} \boldsymbol{\Phi}_{b-t}^{A}(k+t) \boldsymbol{I}_{k+t}^{a} \tilde{\boldsymbol{\Gamma}} \boldsymbol{v}(k+t) \leqslant \overline{\eta}_k \leqslant \overline{\gamma}\, nb < \frac{1}{4\beta},\ \forall k. \tag{40}$$

where we used the relation $v_i(k) \geqslant n^{-b}, \forall i \in \mathcal{V}, k \in \mathbb{N}$ from Lemma 2(iii) and $\overline{\gamma} \leqslant \dfrac{1}{4nb\beta}$ from Lemma 7.

By introducing an auxiliary term $\overline{\eta}_k \nabla f\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k)\right)^{\mathrm{T}}$, Eq. (39) becomes

$$\begin{aligned} \boldsymbol{x}_{\boldsymbol{\pi}}(k+b) = {} & \boldsymbol{x}_{\boldsymbol{\pi}}(k) - \overline{\eta}_k \nabla f\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k)\right)^{\mathrm{T}} + \underbrace{\sum_{t=0}^{b-1} \eta_k(t) \left(\nabla f\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k)\right)^{\mathrm{T}} - \frac{1}{n} \mathbf{1}_{\tilde{n}}^{\mathrm{T}} \nabla(k+t) \right)}_{\boldsymbol{h}(k)} \\ & - \sum_{t=0}^{b-1} \boldsymbol{r}_k(t) \left(\boldsymbol{Y}_V(k+t) - \frac{1}{n} \tilde{\mathbf{1}}(k+t) \boldsymbol{v}(k+t)^{\mathrm{T}} \boldsymbol{Y}_V(k+t) \right) \\ = {} & \boldsymbol{x}_{\boldsymbol{\pi}}(k) - \overline{\eta}_k \nabla f\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k)\right)^{\mathrm{T}} + \boldsymbol{h}(k) - \sum_{t=0}^{b-1} \boldsymbol{r}_k(t) \boldsymbol{S}(k+t) \boldsymbol{Y}_V(k+t), \end{aligned} \tag{41}$$

where we have used the relation $\mathbf{1}_{\tilde{n}}^{\mathrm{T}} \nabla(k) = \mathbf{1}_{\tilde{n}}^{\mathrm{T}} \tilde{\boldsymbol{Y}}(k) = \boldsymbol{v}(k)^{\mathrm{T}} \boldsymbol{Y}_V(k)$ in (14).

We now bound $\boldsymbol{h}(k)$ in (41). Recall that $\nabla f(\boldsymbol{x}) = \mathbf{1}_n^{\mathrm{T}} \nabla \mathbf{f}\left(\mathbf{1}_n \boldsymbol{x}^{\mathrm{T}}\right), \forall \boldsymbol{x} \in \mathbb{R}^m$ and $\mathbf{1}_{\tilde{n}}^{\mathrm{T}} \nabla(k) = \left[\mathbf{1}_n; \mathbf{0}\right]^{\mathrm{T}} \nabla(k) = \mathbf{1}_n^{\mathrm{T}} \nabla \mathbf{f}\left(\boldsymbol{X}(k)\right), \forall k$. Let $\tilde{\mathbf{1}} = \left[\mathbf{1}_n; \mathbf{0}_{\tilde{n}-n}\right]$, we have

$$\begin{aligned} \left\| \boldsymbol{h}(k) \right\|_{\mathrm{F}} = {} & \left\| \sum_{t=0}^{b-1} \eta_k(t) \left(\mathbf{1}_n^{\mathrm{T}} \nabla \mathbf{f}\left(\mathbf{1}_n \boldsymbol{x}_{\boldsymbol{\pi}}(k)\right) - \mathbf{1}_n^{\mathrm{T}} \nabla \mathbf{f}\left(\boldsymbol{X}(k+t)\right) \right) \right\|_{\mathrm{F}} \\ \leqslant {} & \sqrt{n}\, \beta\, \overline{\eta}_k \max_{0 \leqslant t \leqslant b-1} \left\| \mathbf{1}_n \boldsymbol{x}_{\boldsymbol{\pi}}(k)^{\mathrm{T}} - \boldsymbol{X}(k+t) \right\|_{\mathrm{F}} \\ \leqslant {} & \sqrt{n}\, \beta\, \overline{\eta}_k \left(\max_{0 \leqslant t \leqslant b-1} \left\| \mathbf{1}_n \boldsymbol{x}_{\boldsymbol{\pi}}(k)^{\mathrm{T}} - \mathbf{1}_n \boldsymbol{x}_{\boldsymbol{\pi}}(k+t-1) \right\|_{\mathrm{F}} \right. \\ & \left. + \max_{0 \leqslant t \leqslant b-1} \left\| \begin{bmatrix} \boldsymbol{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \left(\mathbf{1}_{\tilde{n}} \boldsymbol{\pi}_{k+t-1}^{\mathrm{T}} \tilde{\boldsymbol{X}}(k+t) - \tilde{\mathbf{X}}(k+t) \right) \right\|_{\mathrm{F}} \right) \\ \leqslant {} & n\beta\, \overline{\eta}_k \sum_{t=0}^{b-1} \left(\left\| \tilde{\boldsymbol{X}}(k+t) \right\|_Q + \overline{\gamma} \left\| \tilde{\boldsymbol{Y}}(k+t) \right\|_{\mathrm{F}} \right), \end{aligned} \tag{42}$$

where we defined $\boldsymbol{\pi}_k \coloneqq \boldsymbol{\pi}(k)$ for notional simplicity and used $\boldsymbol{\pi}_{k+t}^{\mathrm{T}} \tilde{\boldsymbol{X}}(k+t) = \boldsymbol{\pi}_k^{\mathrm{T}} \tilde{\boldsymbol{X}}(k) + \sum_{l=0}^{t-1} \boldsymbol{\pi}_{k+l}^{\mathrm{T}} \tilde{\boldsymbol{\Gamma}} \boldsymbol{I}_{k+l}^{a} \tilde{\boldsymbol{Y}}(k+l)$ from (10) and (17) to obtain the last inequality.

Then, it follows from Proposition 2 and (41) that

$$f\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k+b)\right) - f^{\star} \leqslant \sigma^2 \left(f\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k)\right) - f^{\star} \right) + \left(\frac{2}{\overline{\eta}_k} + \beta \right) \left\| \boldsymbol{h}(k) - \sum_{t=0}^{b-1} \boldsymbol{r}_k(t) \boldsymbol{S}(k+t) \boldsymbol{Y}_V(k+t) \right\|_2^2,$$

where $\sigma = \sqrt{1 - \alpha \overline{\eta}_k \left(1 - 2\beta \overline{\eta}_k\right)}$, and $\sigma \leqslant 1 - \dfrac{1}{2} \alpha \overline{\eta}_k \left(1 - 2\beta \overline{\eta}_k\right) < 1 - \dfrac{1}{4} \alpha \overline{\eta}_k < 1$ as $\sqrt{1-a} \leqslant 1 - \dfrac{1}{2} a$, $\forall a \in (0,1)$, and (40). Using $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$ and (42), we obtain

$$\begin{aligned} \tilde{f}(k+b) = {} & \sqrt{f\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k+b)\right) - f^{\star}} \leqslant \sigma \sqrt{f\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k)\right) - f^{\star}} \\ & + \left(\sqrt{\beta} + \sqrt{2}\, \overline{\eta}_k^{-\frac{1}{2}} \right) \left(\left\| \boldsymbol{h}(k) \right\|_{\mathrm{F}} + \sum_{t=0}^{b-1} \left\| \boldsymbol{r}_k(t) \right\|_2 \left\| \boldsymbol{Y}_V(k+t) \right\|_S \right) \\ \leqslant {} & \tilde{\eta} \sum_{t=0}^{b-1} n\beta \left(\left\| \tilde{\boldsymbol{X}}(k+t) \right\|_Q + \overline{\gamma} \left\| \tilde{\boldsymbol{Y}}(k+t) \right\|_{\mathrm{F}} \right) + \left\| \boldsymbol{Y}_V(k+t) \right\|_S + \sigma \tilde{f}(k), \end{aligned} \tag{43}$$

where $\tilde{\eta} \coloneqq \left(\overline{\eta}_k \sqrt{\beta} + \sqrt{2}\, \overline{\eta}_k^{1/2} \right)$ and the two inequalities follow from (42) and $\left\| \boldsymbol{r}_k(t) \right\|_2 \leqslant \overline{\eta}_k, \forall k, t$.

Since $\lambda^b > 1 - \dfrac{1}{8} \alpha \underline{\gamma}\, n^{-b} > 1 - \dfrac{1}{8} \alpha \overline{\eta}_k$, $\sigma < 1 - \dfrac{1}{4} \alpha \overline{\eta}_k$, we obtain from Proposition 1 and (43) that

$$\tilde{f}^{\lambda,k} \leqslant \frac{16b}{\alpha\sqrt{\overline{\eta}_k}}\left(n\beta\left\|\tilde{\boldsymbol{X}}\right\|_Q^{\lambda,k}+\left\|\boldsymbol{Y}_V\right\|_S^{\lambda,k}+n\beta\,\overline{\gamma}\left\|\tilde{\boldsymbol{Y}}\right\|_{\mathrm{F}}^{\lambda,k}\right)+c_3$$
$$\leqslant \frac{16n^{\frac{b}{2}}b}{\alpha\sqrt{\underline{\gamma}}}\left(n\beta\left\|\tilde{\boldsymbol{X}}\right\|_Q^{\lambda,k}+\left\|\boldsymbol{Y}_V\right\|_S^{\lambda,k}+n\beta\,\overline{\gamma}\left\|\tilde{\boldsymbol{Y}}\right\|_{\mathrm{F}}^{\lambda,k}\right)+c_3,$$

where the first inequality used (40) and $\overline{\eta}_k \leqslant 1/(4\beta)$, and

$$c_3=\frac{16n^{\frac{b}{2}}nb}{\alpha\sqrt{\underline{\gamma}}}\sum_{t=0}^{b-1}\lambda^{-t}\left(\beta\left\|\tilde{\boldsymbol{X}}(t)\right\|_Q+\left\|\boldsymbol{Y}_V(t)\right\|_S+\beta\,\overline{\gamma}\left\|\tilde{\boldsymbol{Y}}(t)\right\|_{\mathrm{F}}\right). \tag{44}$$

6.6 Proof of Lemma 8

Since $\tilde{\boldsymbol{Y}}(k)=\boldsymbol{V}(k)\boldsymbol{Y}_V(k)=\boldsymbol{V}(k)\boldsymbol{S}(k)\boldsymbol{Y}_V(k)+\frac{1}{n}\boldsymbol{V}(k)\tilde{\mathbf{1}}(k)\boldsymbol{v}(k)^{\mathrm{T}}\boldsymbol{Y}_V(k)$, we have

$$\left\|\tilde{\boldsymbol{Y}}(k)\right\|_{\mathrm{F}}\leqslant\left\|\boldsymbol{V}(k)\right\|_2\left(\left\|\boldsymbol{Y}_V(k)\right\|_S+\left\|\frac{1}{n}\tilde{\mathbf{1}}(k)\boldsymbol{v}(k)^{\mathrm{T}}\boldsymbol{Y}_V(k)\right\|_{\mathrm{F}}\right)\leqslant n\left\|\boldsymbol{Y}_V(k)\right\|_S+\left\|\tilde{\mathbf{1}}(k)\mathbf{1}_{\tilde{n}}^{\mathrm{T}}\nabla(k)\right\|_{\mathrm{F}}. \tag{45}$$

Note that

$$\begin{aligned}
\left\|\tilde{\mathbf{1}}(k)\mathbf{1}_{\tilde{n}}^{\mathrm{T}}\nabla(k)\right\|_{\mathrm{F}}&=\left\|\tilde{\mathbf{1}}(k)\mathbf{1}_n^{\mathrm{T}}\left(\nabla\mathbf{f}\left(\boldsymbol{X}(k)\right)-\nabla\mathbf{f}\left(\mathbf{1}_n\left(\boldsymbol{x}^\star\right)^{\mathrm{T}}\right)\right)\right\|_{\mathrm{F}}\leqslant\left\|\tilde{\mathbf{1}}(k)\mathbf{1}_n^{\mathrm{T}}\right\|_2\left\|\nabla\mathbf{f}\left(\boldsymbol{X}(k)\right)-\nabla\mathbf{f}\left(\mathbf{1}_n\left(\boldsymbol{x}^\star\right)^{\mathrm{T}}\right)\right\|_{\mathrm{F}}\\
&\leqslant\sqrt{\tilde{n}}\,\beta\left\|\boldsymbol{X}(k)-\mathbf{1}_n\left(\boldsymbol{x}^\star\right)^{\mathrm{T}}\right\|_{\mathrm{F}}=\sqrt{\tilde{n}}\,\beta\left\|\begin{bmatrix}\boldsymbol{I}_n&\mathbf{0}\\\mathbf{0}&\mathbf{0}\end{bmatrix}\tilde{\boldsymbol{X}}(k)-\begin{bmatrix}\mathbf{1}_n\\\mathbf{0}\end{bmatrix}\left(\boldsymbol{x}^\star\right)^{\mathrm{T}}\right\|_{\mathrm{F}}\\
&\leqslant\sqrt{\tilde{n}}\,\beta\left(\left\|\begin{bmatrix}\boldsymbol{I}_n&\mathbf{0}\\\mathbf{0}&\mathbf{0}\end{bmatrix}\tilde{\boldsymbol{X}}(k)-\begin{bmatrix}\mathbf{1}_n\\\mathbf{0}\end{bmatrix}\boldsymbol{\pi}(k-1)^{\mathrm{T}}\tilde{\boldsymbol{X}}(k)\right\|_{\mathrm{F}}+\left\|\begin{bmatrix}\mathbf{1}_n\left(\boldsymbol{\pi}(k-1)^{\mathrm{T}}\tilde{\boldsymbol{X}}(k)-\left(\boldsymbol{x}^\star\right)^{\mathrm{T}}\right)\\\mathbf{0}\end{bmatrix}\right\|_{\mathrm{F}}\right)\\
&\leqslant\sqrt{\tilde{n}}\,\beta\left\|\left(\begin{bmatrix}\boldsymbol{I}_n&\mathbf{0}\\\mathbf{0}&\mathbf{0}\end{bmatrix}-\begin{bmatrix}\mathbf{1}_n\\\mathbf{0}\end{bmatrix}\boldsymbol{\pi}(k-1)^{\mathrm{T}}\right)\boldsymbol{Q}\tilde{\boldsymbol{X}}(k)\right\|_{\mathrm{F}}+\sqrt{\tilde{n}}\,\beta\left\|\begin{bmatrix}\mathbf{1}_n\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k)-\left(\boldsymbol{x}^\star\right)^{\mathrm{T}}\right)\\0\end{bmatrix}\right\|_{\mathrm{F}}\\
&\leqslant\sqrt{\tilde{n}}\,\beta\left(\sqrt{n}\left\|\tilde{\boldsymbol{X}}(k)\right\|_Q+\sqrt{n}\left\|\boldsymbol{x}_{\boldsymbol{\pi}}(k)-\boldsymbol{x}^\star\right\|_{\mathrm{F}}\right)\\
&\leqslant\sqrt{\tilde{n}}\,\beta\left(\sqrt{n}\left\|\tilde{\boldsymbol{X}}(k)\right\|_Q+\frac{\sqrt{2n}}{\sqrt{\alpha}}\sqrt{f\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k)\right)-f^\star}\right),
\end{aligned} \tag{46}$$

where $\boldsymbol{x}^\star=\Pi_{\mathcal{X}^*}\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k)\right)$ and the last inequality follows from that the Polyak-Łojasiewicz condition implies the quadratic growth condition (Theorem 2 (Karimi et al., 2016)), i. e., $f(\boldsymbol{x})-f^\star\geqslant\frac{\alpha}{2}\left\|\boldsymbol{x}-\Pi_{\mathcal{X}^*}(\boldsymbol{x})\right\|_2^2,\ \forall\boldsymbol{x}$.

Substituting (46) into (45) yields

$$\left\|\tilde{\boldsymbol{Y}}(k)\right\|_{\mathrm{F}}\leqslant n\left\|\boldsymbol{Y}_V(k)\right\|_S+\sqrt{\tilde{n}}\,\beta\left(\sqrt{n}\left\|\tilde{\boldsymbol{X}}(k)\right\|_Q+\frac{\sqrt{2n}}{\sqrt{\alpha}}\sqrt{f\left(\boldsymbol{x}_{\boldsymbol{\pi}}(k)\right)-f^\star}\right).$$

The desired result is obtained by the definition of $\left\|\tilde{\boldsymbol{Y}}\right\|_{\mathrm{F}}^{\lambda,k}$.

7 Numerical examples

We use APPG to train a multi-class logistic regression classifier in a distributed manner on the *Covertype* dataset (Dheeru et al., 2017), where the objective function takes the following form

$$f(\boldsymbol{X})=-\sum_{i=1}^{n_s}\sum_{j=1}^{n_c}l_j^i\log\left(\frac{\exp\left(\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{s}^i\right)}{\sum\limits_{j'=1}^{n_c}\exp\left(\boldsymbol{x}_{j'}^{\mathrm{T}}\boldsymbol{s}^i\right)}\right)+\frac{\rho}{2}\left\|\boldsymbol{X}\right\|_{\mathrm{F}}^2.$$

Here $n_s=581\,012$ is the number of training instances, $n_c=7$ is the number of classes, $n_f=55$ is the number of

features, $\boldsymbol{s}^i \in \mathbb{R}^{55}$ is the feature vector of the i-th instance, $\boldsymbol{l}^i = \left[l_1^i, \cdots, l_7^i\right]^{\mathrm{T}}$ is the label vector of the i-th instance using the one-hot encoding, $\boldsymbol{X} = \left[\boldsymbol{x}_1, \cdots, \boldsymbol{x}_7\right] \in \mathbb{R}^{n_f \times n_c}$ is the parameters to be optimized, $\rho = 20$ is a regularization factor.

Environment: APPG is implemented in Python with OpenMPI 1. 10 on Ubuntu 14. 04. The hardware is a server with 28 Xeon E5-2660 cores. Each core serves as a computing node.

Distributed Data: We first normalize non-categorical features by subtracting the mean and dividing by the standard deviation in the whole dataset. Then, we *sort* the data by digit label, and sequentially partition it into n parts (with different sizes), where each node (core) only has *exclusive* access to one part. Thus, we are dealing with distributed datasets.

Topology: The directed network among nodes is as follows: Each node i sends messages to node $\mathrm{mod}\left(2^j + i, n\right)$, where $j \in \mathbb{N} \cap [0, \log_2(n))$ and $\mathrm{mod}\left(a, b\right)$ returns the remainder after division of a by b. Thus, each node has $\mathcal{O}\left(\log(n)\right)$ out-neighbors, which results in a relatively sparse directed networks. Note that gossip-based asynchronous algorithms are generally not applicable over this directed network.

Stepsize: The stepsize of each algorithm is tuned via a grid search around 0. $5/n_s$.

Local Termination Criteria: Node i stops locally if the values of $\boldsymbol{y}_i$ in last n consecutive iterations are less than $300/n_s$.

7. 1 Convergence performance and linear speedup

We implement APPG over $n = 1, 6, 12, 18, 24$ nodes ($n = 1$ is used as a baseline since APPG reduces to standard centralized gradient descent method). The training loss w. r. t. running time is plotted in Fig. 6(a), which validates the linear convergence of APPG, and shows that the training time is significantly reduced with the increase of number of nodes.

Fig. 6(b) depicts the training loss w. r. t. the number of iterations. We find that the number of iterations required to achieve the same accuracy is close to each other for different number of nodes. The time for a node to finish an iteration is proportional to the size of its local dataset, and hence is roughly inversely proportional to the number of nodes, which suggests that using n nodes may reduce $\mathcal{O}(n)$ times of training time than that of one node.

To further illustrate this property, we study the speedup of APPG defined as $S_n := T_n/T_1$, where T_n is the running time of the APPG with n node (s) when the training loss decays to 0. 005. Fig. 7(a) shows that the APPG achieves a roughly linear speedup in convergence rate w. r. t. the number of nodes. Then, we test a synchronized version of APPG, which is done by adding a barrier after each update and is mathematically equivalent to the push-pull method in Pu et al. (2021). One can find that the synchronous version has an approximately linear speedup when the number of cores is small, but it decreases fast when the number of cores is relatively large.

Ideally, the speedup would be n when using n nodes. However, the communication among nodes introduces delays and staleness to the algorithm, which degrades the convergence rate. In practice, a higher speedup can be achieved by using a lower latency network with larger bandwidth.

7. 2 Robustness of APPG to slow cores

We evaluate the robustness of APPG by forcing one core in the network to slow down. This is achieved by adding an artificial waiting time (20 ms, a normal iteration takes about 15 ms with 24 cores) after each local iteration of a node, which simulates either the slow computation or slow communication.

Fig. 7(b) shows the speedup of APPG and the synchronous implementation of APPG in this scenario. It indicates that the synchronous counterpart of APPG has a sharp reduction in convergence rate even when only 1 core slows down. In contrast, APPG still keeps an almost linear speedup. This result is also consistent with that in Lian et al. (2018); Zhang et al. (2020a).

Introducing the slowing core also brings an essential problem of asynchronous algorithms, that is, the cores

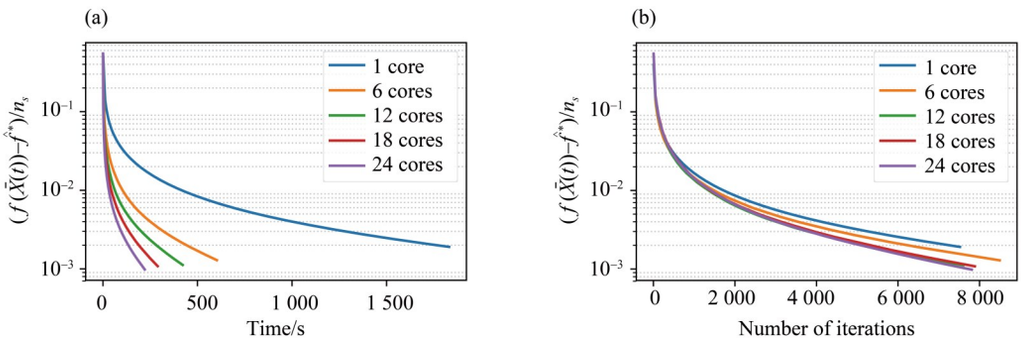

Fig. 6　Convergence performance with different number of nodes. (a) Training loss w. r. t. running time of APPG; (b) Training loss w. r. t. number of iterations of APPG

have *uneven* update rates. To show its effect to the performance, we compare APPG to a gossip-based asynchronous algorithm AD-PSGD (Lian et al. , 2018) with full local gradients. Note that APPG can only work over *undirected* networks, and hence we modify the network by adding a reversed edge to each edge in the directed network, while APPG is still implemented over the directed network. Fig. 8 shows the result over 12 nodes and 24 nodes, where AD-PSGD fails to converge to the exact optimum. In contrast, APPG converges exactly despite a bit slower convergence rate.

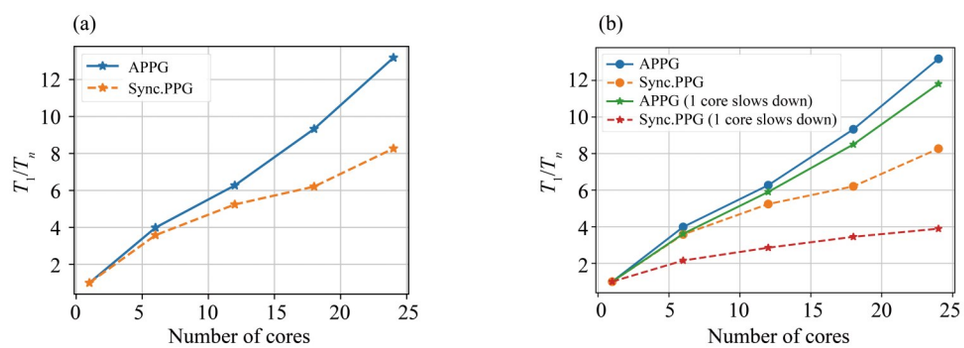

Fig. 7　(a) Speedup of APPG and the synchronous implementation of APPG; (b) Speedup of APPG and the ‘synchronized’ APPG with one slow core

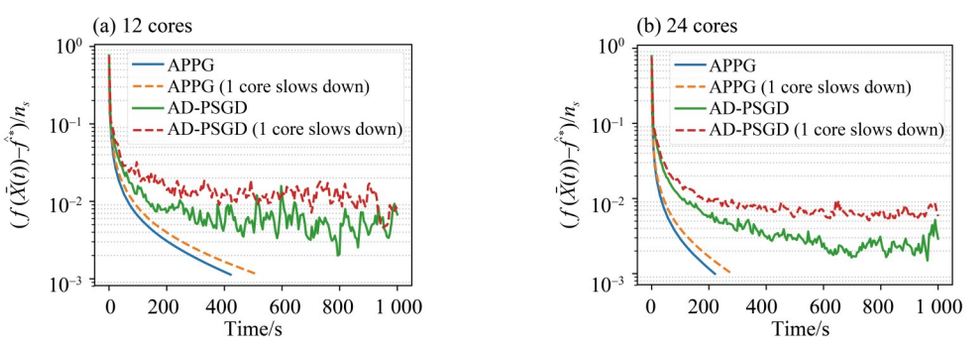

Fig. 8　Convergence of APPG and AD-PSGD with full local gradient when one node is artificially slowed down.

8　Conclusion

This paper has proposed APPG for distributed optimization which allows nodes to connect via a directed communication network and update with uncoordinated computation and stale information from neighbors. Future works may focus on accelerating APPG and extending it to stochastic optimization.

References:

ASSRAN M S，RABBAT M G，2021. Asynchronous gradient push[J]. IEEE Trans Autom Control，66(1)：168−183.

ASSRAN M，LOIZOU N，BALLAS N，et al，2019. Stochastic gradient push for distributed deep learning[C]//Proc Int Conf Mach Learn，97：344−353.

ASSRAN M，AYTEKIN A，FEYZMAHDAVIAN H R，et al，2020. Advances in asynchronous parallel and distributed optimization [J]. Proc IEEE，108(11)：2013−2031.

CHANG T H，HONG M Y，WANG X F，2015. Multi-agent distributed optimization via inexact consensus ADMM[J]. IEEE Trans Signal Process，63(2)：482−497.

CHANG T H，HONG M Y，LIAO W C，et al，2016. Asynchronous distributed ADMM for large-scale optimization—Part I：Algorithm and convergence analysis[J]. IEEE Trans Signal Process，64(12)：3118−3130.

DHEERU D，KARRA TANISKIDOU E，2017. UCI machine learning repository[EB/OL]. University of California，Irvine，School of Information and Computer Sciences. http://archive.ics.uci.edu/ml.

FAZEL M，GE R，KAKADE S M，et al，2018. Global convergence of policy gradient methods for the linear quadratic regulator [C]//Proc Int Conf Mach Learn，80：1467−1476.

JAKOVETIĆ D，2019. A unification and generalization of exact distributed first-order methods[J]. IEEE Trans Signal Inf Process Netw，5(1)：31−46.

KARIMI H，NUTINI J，SCHMIDT M，2016. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition[C]//Joint Eur Conf Mach Learn Knowl Discovery Databases，795−811.

LI H Q，HUANG C C，WANG Z，et al，2020. Computation-efficient distributed algorithm for convex optimization over time-varying networks with limited bandwidth communication[J]. IEEE Trans Signal Inf Process Netw，6：140−151.

LI S，BASAR T，1987. Asymptotic agreement and convergence of asynchronous stochastic algorithms[J]. IEEE Trans Autom Control，32(7)：612−618.

LI Z，SHI W，YAN M，2019. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates[J]. IEEE Trans Signal Process，67(17)：4494−4506.

LIAN X R，ZHANG W，ZHANG C，et al，2018. Asynchronous decentralized parallel stochastic gradient descent[C]//Proc Int Conf Mach Learn，80：3043−3052.

NEDIĆ A，2011. Asynchronous broadcast-based convex optimization over a network[J]. IEEE Trans Autom Control，56(6)：1337−1351.

NEDIĆ A，OLSHEVSKY A，2015. Distributed optimization over time-varying directed graphs[J]. IEEE Trans Autom Control，60(3)：601−615.

NEDIĆ A，OLSHEVSKY A，2016. Stochastic gradient-push for strongly convex functions on time-varying directed graphs[J]. IEEE Trans Autom Control，61(12)：3936−3947.

NEDIĆ A，OZDAGLAR A，2009. Distributed subgradient methods for multi-agent optimization[J]. IEEE Trans Autom Control，54(1)：48−61.

NEDIĆ A，OZDAGLAR A，2010. Convergence rate for consensus with delays[J]. J Glob Optim，47(3)：437−456.

NEDIĆ A，OLSHEVSKY A，SHI W，2017. Achieving geometric convergence for distributed optimization over time-varying graphs[J]. SIAM J Optim，27(4)：2597−2633.

OLSHEVSKY A，TSITSIKLIS J N，2011. Convergence speed in distributed consensus and averaging[J]. SIAM Rev，53(4)：747−772.

PU S，SHI W，XU J M，et al，2021. Push-pull gradient methods for distributed optimization in networks[J]. IEEE Trans Autom Control，66(1)：1−16.

SAADATNIAKI F，XIN R，KHAN U A，2020. Decentralized optimization over time-varying directed graphs with row and column-stochastic matrices[J]. IEEE Trans Autom Control，65(11)：4769−4780.

SCUTARI G，SUN Y，2019. Distributed nonconvex constrained optimization over time-varying digraphs[J]. Math Program，176(1/2)：497−544.

SPIRIDONOFF A，OLSHEVSKY A，PASCHALIDIS I C，2020. Robust asynchronous stochastic gradient-push：Asymptotically optimal and network-independent performance for strongly convex functions[J]. J Mach Learn Res，21(58)：1−47.

SUN H，HONG M Y，2019. Distributed non-convex first-order optimization and information processing：Lower complexity bounds and rate optimal algorithms[J]. IEEE Trans Signal Process，67(22)：5912−5928.

TIAN Y，SUN Y，SCUTARI G，2020. Achieving linear convergence in distributed asynchronous multi-agent optimization[J]. IEEE Trans Autom Control，65(12)：5264−5279.

TOURI B，2012. Product of random stochastic matrices and distributed averaging[M]. Berlin：Springer-Verlag.

TSITSIKLIS J，BERTSEKAS D，ATHANS M，1986. Distributed asynchronous deterministic and stochastic gradient optimization algorithms[J]. IEEE Trans Autom Control，31(9)：803−812.

WU T Y，YUAN K，LING Q，et al，2018. Decentralized consensus optimization with asynchrony and delays[J]. IEEE Trans Signal Inf Process Netw，4(2)：293−307.

XIE P，YOU K Y，TEMPO R，et al，2018. Distributed convex optimization with inequality constraints over time-varying unbalanced digraphs[J]. IEEE Trans Autom Control，63(12)：4331−4337.

XIN R，KHAN U A，2018. A linear algorithm for optimization over directed graphs with geometric convergence[J]. IEEE Control Syst Lett，2(3)：315−320.

XIN R，KAR S，KHAN U A，2020. Decentralized stochastic optimization and machine learning：A unified variance-reduction framework for robust performance and fast convergence[J]. IEEE Signal Process Mag，37(3)：102−113.

XU J M，ZHU S Y，SOH Y C，et al，2018. Convergence of asynchronous distributed gradient methods over stochastic networks[J]. IEEE Trans Autom Control，63(2)：434−448.

XU J M，TIAN Y，SUN Y，et al，2021. Distributed algorithms for composite optimization：Unified framework and convergence analysis[J]. IEEE Trans Signal Process，69：3555−3570.

YI X L，LI X X，XIE L H，et al，2020. Distributed online convex optimization with time-varying coupled inequality constraints[J]. IEEE Trans Signal Process，68：731−746.

YUAN K，YING B C，ZHAO X C，et al，2019. Exact diffusion for distributed optimization and learning—Part I：Algorithm development[J]. IEEE Trans Signal Process，67(3)：708−723.

ZHANG J Q，YOU K Y，2020a. AsySPA：An exact asynchronous algorithm for convex optimization over digraphs[J]. IEEE Trans Autom Control，65(6)：2494−2509.

ZHANG J Q，YOU K Y，CAI K，2020b. Distributed dual gradient tracking for resource allocation in unbalanced networks[J]. IEEE Trans Signal Process，68：2186−2198.

ZHAO X C，SAYED A H，2015. Asynchronous adaptation and learning over networks—Part I：Modeling and stability analysis[J]. IEEE Trans Signal Process，63(4)：811−826.

（责任编辑　冯兆永）