This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

Asynchronous Message-Passing and Zeroth-Order Optimization Based Distributed Learning with a Use-Case in Resource Allocation in Communication Networks

Pourya Behmandpoor, Marc Moonen, Panagiotis Patrinos

***Abstract*—Distributed learning and adaptation have received significant interest and found wide-ranging applications in machine learning and signal processing. While various approaches, such as shared-memory optimization, multi-task learning, and consensus-based learning (e.g., federated learning and learning over graphs), focus on optimizing either local rewards or a global reward, there remains a need for further exploration of their interconnections. This paper specifically focuses on a scenario where agents collaborate towards a common task (i.e., optimizing a global reward equal to aggregated local rewards) while effectively having distinct individual tasks (i.e., optimizing individual local parameters in a local reward). Each agent's actions can potentially impact other agents' performance through interactions. Notably, each agent has access to only its local zeroth-order oracle (i.e., reward function value) and shares scalar values, rather than gradient vectors, with other agents, leading to communication bandwidth efficiency and agent privacy. Agents employ zeroth-order optimization to update their parameters, and the asynchronous message-passing between them is subject to bounded but possibly random communication delays. This paper presents theoretical convergence analyses and establishes a convergence rate for nonconvex problems. Furthermore, it addresses the relevant use-case of deep learning-based resource allocation in communication networks and conducts numerical experiments in which agents, acting as transmitters, collaboratively train their individual policies to maximize a global reward, e.g., a sum of data rates.**

***Index Terms*—Distributed learning and adaptation, asynchronous distributed optimization, zeroth-order optimization, bounded delay, deep learning-based resource allocation**

I. Introduction

A. Distributed learning over message-passing architectures

The recent proliferation of edge devices with computational power, data generation and processing, has motivated research on distributed learning and adaptation. In distributed learning, edge devices, as distributed agents, aim to cooperatively learn a common task or multiple individual tasks. This collaboration occurs within a message-passing architecture, which poses challenges due to heterogeneity in local reward functions/data, processing capabilities, and potential communication delays or failures.

Asynchronous distributed optimization and learning techniques in the literature address this collaboration and consider various message-passing approaches as follows: *Shared-memory approach:* All agents have access to a common shared memory [1]–[5]. Each agent reads the required information from the shared memory and updates the decision variable, either in full or in part, by writing to the shared memory. This model operates asynchronously as the reading and writing processes occur asynchronously. *Local-memory approach:* Each agent possesses a local (private) memory and asynchronously updates a specific part of the decision variable, i.e., the local parameters [4]–[7]. After each update, the agent broadcasts the updated local parameter to other users who require it for their own updates. Due to random communication delays or dropouts, the information received from other agents and stored in the local memories may be outdated. The local-memory approach can be considered, for instance, in multitask learning approaches [8] where each agent optimizes its local parameters based on the sum of its local reward function and a regularizer that captures the relation of the local parameters with other agents' local parameters. In games and multi-agent reinforcement learning, this approach can be considered where each agent optimizes its local reward function while capturing its dependence on other players through interaction [9], [10]. *Consensus-based approach:* Each agent also updates its local parameters asynchronously. However, unlike the previous approaches, agents in this approach aim to achieve parameter consensus. Consensus can be achieved through frequent averaging facilitated by a central entity, as seen in federated learning settings [11]–[13], or through frequent averaging of local parameters among neighboring agents, as seen in learning over graph settings [14]–[19]. In this approach, all agents strive to reach common optimal parameters optimized over a global reward function, such as the sum of local rewards.

While the three aforementioned approaches in distributed optimization and learning have been extensively studied in isolation, their interconnections remain relatively unexplored.

This research work was carried out at the ESAT Laboratory of KU Leuven, in the frame of Research Project FWO nr. G0C0623N 'User-centric distributed signal processing algorithms for next generation cell-free massive MIMO based wireless communication networks' and Fonds de la Recherche Scientifique - FNRS and Fonds voor Wetenschappelijk Onderzoek - Vlaanderen EOS Project no 30452698 '(MUSE-WINET) MUlti-SErvice WIreless NETworks'. The Work is also supported by the Research Foundation Flanders (FWO) postdoctoral grant 12Y7622N and research projects G081222N, G033822N, and G0A0920N; Research Council KU Leuven C1 project No. C14/24/103; European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 953348. The scientific responsibility is assumed by its authors.

Pourya Behmandpoor, Panagiotis Patrinos, and Marc Moonen are with KU Leuven University, Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics (e-mail: pourya.behmandpoor, marc.moonen, panos.patrinos @esat.kuleuven.be).

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

This motivates us to further investigate cases where agents share a common task while they also have different individual tasks. This scenario bridges the gap between the last two mentioned approaches, namely the local-memory and consensus-based approaches.

In this setting, each agent's actions and decisions can potentially impact the performance of other agents, necessitating consideration of interactions during local optimization, similar to the local-memory approach. Furthermore, agents optimize their local parameters not only based on their own local rewards but also with respect to a global function (e.g., sum of local rewards) that reflects the common task, similar to the consensus-based approach.

To the best of our knowledge, this is the first theoretical convergence study addressing such a scenario, where agents asynchronously update their local parameters using zeroth-order oracles by querying only local reward function values. Employing zeroth-order optimization with zeroth-order oracles rather than first-order oracles (i.e., function gradients) in each agent brings multiple benefits. While first- or higher-order oracles may not be available or expensive to calculate in some settings (e.g., simulation-based optimization or bandit optimization [19]), zeroth-order oracles can be available in a more computationally- and memory-efficient manner [20]. Moreover, zeroth-order oracles significantly reduce the communication bandwidth requirements in the proposed distributed learning as it helps agents share only scalars. This feature, sharing scalars unlike sharing local parameters or gradients in federated learning or learning over graph settings, ensures privacy-preserving cooperation [21] (cf. Section V for more details). The relevant problem of RA is examined as a use-case within this framework.

B. Resource allocation in communication networks

RA has been an active research area in both wired and wireless communication networks for several decades. Recently, deep learning (DL)-based approaches have emerged as a promising solution, offering improvements in speed, performance efficiency, and implementation simplicity compared to conventional RA methods [22], [23].

DL-based RA typically consists of two stages: *the training stage* and *the inference stage*. In the training stage, policy (typically a deep neural network (DNN)) parameters are optimized using data-driven approaches. In the inference stage, the trained policies are utilized for performing RA. Both training and inference stages can be categorized into two main scenarios: *(i) centralized, (ii) distributed.*

In the centralized inference scenario, a centralized policy implemented on a server performs the RA by collecting all the relevant global measurements [24]–[28]. On the other hand, in the distributed scenario, each transmitter possesses its own (possibly unique [29]) RA policy, where it allocates resources locally, potentially through message passing with other transmitters [29]–[37]. In both scenarios, however, the training stage is typically performed in a centralized manner [28]–[35].

Distributed training facilitates online training of RA policies meaning that the policies can be trained simultaneously while they are being utilized. This offers several advantages over centralized (offline) training: (1) Online training enables the capture of real-time channel and user state distributions, resulting in more accurate policy training compared to offline training, which may suffer from a mismatch between the training sample distribution and the actual sample distribution [24]. (2) Non-stationary channels can lead to distribution drifts, which are caused by the dynamic nature of communication systems. Factors such as transmitter and receiver dynamic locations, varying environmental conditions (e.g., varying traffic patterns in vehicle-to-vehicle communication), and shifting user demand patterns can contribute to distribution drifts [28], [38]. Distributed training provides a means to address these distribution mismatches and drifts through (online) policy retraining [28]. (3) In practical scenarios, it may be infeasible to have a server with sufficient communication and computational capacity to perform centralized training. Distributed training offers a viable alternative in such cases, as it does not rely on a central server and can distribute the training process across multiple agents [39]. Online (distributed) training can be established by employing zeroth-order or higher-order optimization approaches.

Zeroth-order optimization for RA policy training (also called model-free training) is considered in this paper. During the proposed zeroth-order training, agents approximate utility function gradients by *measuring* their utility values, e.g., data rates, rather than calculating function gradients [24]. This approach can benefit from the advantages of zeroth-order optimization mentioned in the previous subsection. Moreover, it captures the full behavior of the communication system, including nonidealities such as nonlinearities in the modulator and demodulator or antenna setup, which are not typically captured by higher-order approaches due to assumed simplifications in the utility functions, e.g., simplifications made in data rate formulation. The contributions are listed as follows:

- *(i)* A novel distributed learning approach is proposed, where agents share a common task (hence they optimize the sum of local rewards), while each agent has also an individual task (hence agents have local (unique) parameters). Agents impact each others' performance through interactions. They update their local parameters using zeroth-order optimization methods. During the training, agents share scalar quantities, which saves communication bandwidth and preserves their privacy.
- *(ii)* The proposed learning approach operates asynchronously, with agents updating and querying their zeroth-order oracle at the time of interest. Communication between agents is subject to bounded but potentially random delays.
- *(iii)* Theoretical convergence analyses for smooth nonconvex functions are provided, establishing a convergence rate of $\mathcal{O}(1/\sqrt{t})$, where t represents the iteration counter. This convergence rate aligns with the state-of-the-art results found in the zeroth-order optimization literature.
- *(iv)* The relevant problem of DL-based RA is also explored. Building upon the proposed learning approach, agents, acting as transmitters, collaborate to achieve maximum

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

global reward. The cross-talk communication channels naturally link the agents together.

II. System Model

In this paper, we examine the following optimization problem:

$$\underset{\boldsymbol{\theta}=(\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_m)\in\mathbb{R}^n}{\text{minimize}} \; f(\boldsymbol{\theta}) \coloneqq \frac{1}{m}\sum_{\ell=1}^{m} f_\ell(\boldsymbol{\theta}), \tag{1}$$

where the functions $f_\ell : \mathbb{R}^n \to \mathbb{R}$ are continuously differentiable but possibly nonconvex. We consider a scenario with m agents aiming to cooperatively optimize (1) over a message-passing architecture. Each agent ℓ optimizes its local parameter vector $\boldsymbol{\theta}_\ell \in \mathbb{R}^{n_\ell}$, while its local reward function f_ℓ depends on $\boldsymbol{\theta} = (\boldsymbol{\theta}_\ell, \boldsymbol{\theta}_{-\ell})$, and $\boldsymbol{\theta}_{-\ell} \coloneqq \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{\ell-1}, \boldsymbol{\theta}_{\ell+1}, ..., \boldsymbol{\theta}_m\}$ represents local parameters of other agents. Furthermore, motivated by the advantages of zeroth-order optimization mentioned in the previous section, we consider the case where agents access only local stochastic zeroth-order oracles. In other words, each agent ℓ can only measure its noisy reward function value $F_\ell(\boldsymbol{\theta}, \boldsymbol{\xi})$ which is unbiased, i.e., $\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D}_\ell^\xi}[F_\ell(\boldsymbol{\theta}, \boldsymbol{\xi})] = f_\ell(\boldsymbol{\theta})$. In this oracle, $\boldsymbol{\xi} \in \mathbb{R}^s$ is a random variable drawn from local data distribution $\mathcal{D}_\ell^\xi$ and may represent training samples in the learning context (cf. the example of RA in the use-case discussed in Section VII). We consider the following basic assumptions for (1):

Assumption I (basic assumptions)**.**

(i) The functions f_ℓ $\forall \ell \in [m] \coloneqq \{1, 2, ..., m\}$, and consequently f, in (1) are L^0-Lipschitz and L-smooth, i.e.,

$$|f_\ell(\boldsymbol{\theta}) - f_\ell(\boldsymbol{\theta}')| \le L^0 \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|, \quad \forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^n;$$
$$\|\nabla f_\ell(\boldsymbol{\theta}) - \nabla f_\ell(\boldsymbol{\theta}')\| \le L \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|, \quad \forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^n.$$

(ii) The stochastic zeroth-order oracle is unbiased:

$$f_\ell(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D}_\ell^\xi}[F_\ell(\boldsymbol{\theta}, \boldsymbol{\xi})], \; \forall \ell \in [m], \forall \boldsymbol{\theta} \in \mathbb{R}^n.$$

It is worth noting that, for the sake of simplicity, a common Lipschitz constant L is assumed for all functions f_ℓ.

As an example, the optimization problem in (1) deals with a scenario where there are m agents, possibly with different tasks $\boldsymbol{\theta}_i \neq \boldsymbol{\theta}_j, i \neq j$. However, they also share a common task $f(\boldsymbol{\theta})$. Therefore, each agent ℓ seeks to optimize its local parameters $\boldsymbol{\theta}_\ell$ over the sum of all local reward functions. This is different from multi-task or game scenarios where each agent ℓ addresses the problem given as $\text{minimize}_{\boldsymbol{\theta}_\ell} f_\ell(\boldsymbol{\theta})$, with $f_\ell(\boldsymbol{\theta}) = g_\ell(\boldsymbol{\theta}_\ell) + R(\boldsymbol{\theta})$ consisting of the local reward g_ℓ and a regularizer capturing the relation between tasks $\boldsymbol{\theta}_j$ in multi-task scenarios [8] or $f_\ell(\boldsymbol{\theta})$ representing the local reward influenced by other players in the game [9]. In addition, it is different from federated learning [11]–[13] or learning over graphs [14]–[19] where agents seek a consensus in $\text{minimize}_{\boldsymbol{\theta}=(\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_m)\in\mathbb{R}^n} \sum_{\ell=1}^m g_\ell(\boldsymbol{\theta}_\ell)$ by including the constraint $\boldsymbol{\theta}_\ell = \boldsymbol{\theta}_j, \ell \neq j$. As an illustrative use-case, we focus on the relevant problem of resource allocation (RA) in wireless communication systems.

Application in RA: Consider the problem of RA where agents are represented as m transmitters in the communication network, serving $n \le m$ receivers. The set of transmitters and receivers are denoted as $\mathcal{T}$ and $\mathcal{R}$, respectively. In the case where $m = n$, the communication network is considered ad-hoc, i.e., each transmitter serves only one dedicated receiver. In the case where $m > n$, the receivers can be considered as base stations, where multiple transmitters have uplink communication with one base station, possibly causing interference to other base stations. We refer to $r(i) \in \mathcal{R}$ as the receiver dedicated to transmitter $i \in \mathcal{T}$. The communication channel between the ith transmitter and the jth receiver is considered as a random variable, denoted by $h_{ji} \in \mathbb{C}$. The matrix $\boldsymbol{H} \in \mathbb{C}^{|\mathcal{R}|\times|\mathcal{T}|}$, with $|\cdot|$ representing the cardinality of the set, contains all the channels in the communication network, with the channel h_{ji} at the jth row and the ith column of the matrix.

The RA problem aims to find optimal resources (e.g., transmit power) for each user (as an agent) such that the global reward (e.g., sum of user data rates for communication) is maximized. This is done by performing the optimization (1) with $f_\ell(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\xi}=\boldsymbol{H}}[r_\ell^\pi(\boldsymbol{\pi}(\boldsymbol{\theta}, \boldsymbol{H}), \boldsymbol{H})]$ where $r_\ell^\pi : \mathbb{R}^m \times \mathbb{C}^{|\mathcal{R}|\times|\mathcal{T}|} \to \mathbb{R}$ denotes agent ℓ's reward or so-called utility, e.g., its data rate. The vector $\boldsymbol{\pi}(\boldsymbol{\theta}, \boldsymbol{H}) \coloneqq (\pi_1(\boldsymbol{\theta}_1, \boldsymbol{H}), \cdots, \pi_m(\boldsymbol{\theta}_m, \boldsymbol{H})) \in \mathbb{R}^m$ contains individual RA policies $\pi_i : \mathbb{R}^{n_i} \times \mathbb{C}^{|\mathcal{R}|\times|\mathcal{T}|} \to \mathbb{R}$, e.g., deep neural networks (DNNs). These policies map their input to communication resources, e.g., the transmit power of the corresponding agents. Each policy π_i is parametrized by the policy parameter $\boldsymbol{\theta}_i \in \mathbb{R}^{n_i}$. Due to interference between agents through cross-talk channels h_{ji}, $j \neq i$, the utility of each agent, i.e., r_ℓ^π, depends on all the policies $\boldsymbol{\pi}$. Therefore, cooperative learning is required. Otherwise, the utilities are decoupled as $r_\ell^\pi(\pi_\ell(\boldsymbol{\theta}_\ell, \boldsymbol{H}), h_{r(\ell)\ell})$ which can be addressed by non-cooperative learning.

III. Proposed Learning Setup

In this section, we present the setup that governs the local updates and communication among the agents. Frequently used notations are listed in Table I in the Appendix.

A. Agent Activity Time

Let $t \in \mathbb{N}$ represent the global time. The set of time instants at which agent i updates its local parameters is defined as T_u^i. Furthermore, each agent i queries its zeroth-order oracle (i.e., measures its reward function value) $F_i(\boldsymbol{\theta}^t, \boldsymbol{\xi})$ at certain time instants, which define the set T_q^i. At any time, each agent is free to decide whether to update its parameter or keep the current parameter, as well as whether to query its oracle or not. Consequently, we take the following assumption:

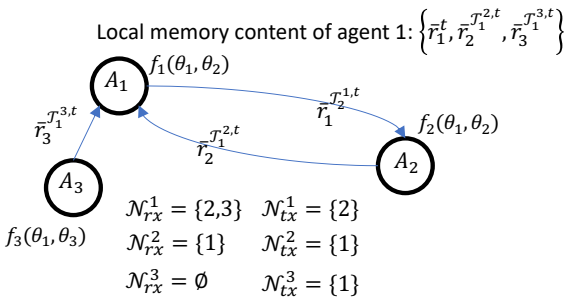
Assumption II (update and query times)**.** *Updates and queries take place at certain time instants such that $T_q^i \subseteq \mathbb{N}$, $T_u^i \subseteq \mathbb{N}$ for all $i \in [m]$.*

B. Message-passing Architecture

The exchange of information among agents depends on the structure of the local reward functions f_i for all $i \in [m]$.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421



(a) *Message-passing architecture among the agents. The quantities $\bar{r}_j^t$ and $\mathcal{T}_i^{j,t}$ are defined in (4) and (5), respectively. Depending on its local reward function, each agent defines the sets of agents $\mathcal{N}_{rx}^i$ and $\mathcal{N}_{tx}^i$ with which it communicates and saves necessary variables in its local memory. Due to asynchrony and random communication delays, the memory contents may be outdated.*



(b) *Update and query time of agent 1. The axis ticks correspond to global time instants $t \in \mathbb{N}$. Agents asynchronously query their reward functions and update their local parameters.*

Figure 1: *An example of a message-passing architecture with asynchronous agents. The communication delays are possibly random and vary over time. The detailed workflow is provided in Algorithm 1.*

To facilitate the presentation of this exchange, we introduce neighbor sets for each agent $i \in [m]$ as follows:

$$\begin{aligned} \mathcal{N}_{rx}^i &:= \{j \in [m] \mid f_j \text{ depends on } \boldsymbol{\theta}_i\}, \\ \mathcal{N}_{tx}^i &:= \{j \in [m] \mid f_i \text{ depends on } \boldsymbol{\theta}_j\}. \end{aligned} \tag{2}$$

The significance of these neighbor sets will become clear in the subsequent sections. The neighbor sets in (2) are application-dependent and agents are aware of them based on system design. An example of a message-passing architecture with asynchronous agent activities is illustrated in Fig. 1.

IV. Zeroth-order optimization

We utilize *randomized smoothing techniques* [40], [41] to leverage zeroth-order optimization methods in the proposed learning approach. These techniques involve introducing random perturbations to smooth out the objective function. Specifically, for a given function f defined in (1), its smooth *approximation* at $\boldsymbol{\theta}$ is defined as

$$f^\mu(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{u}\sim\mathcal{D}_u}[f(\boldsymbol{\theta} + \mu\boldsymbol{u})], \tag{3}$$

where $\mu > 0$, and $\boldsymbol{u}$ is a random perturbation vector with a distribution $\mathcal{D}_u$. Nesterov [41] demonstrated that if the distribution is Gaussian, i.e., $\mathcal{D}_u = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, then f^μ exhibits the following properties:

Proposition IV.1 (the smooth approximation function f^μ [41])**.** *With the constant $\mu > 0$:*

(i) If f is L^0-Lipschitz and L-smooth, f^μ is also L_μ^0-Lipschitz and L_μ-smooth with constants $L_\mu^0 \le L^0$ and $L_\mu \le L$,

(ii) The gradient ∇f^μ is derived as

$$\nabla f^\mu(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\frac{f(\boldsymbol{\theta}+\mu\boldsymbol{u}) - f(\boldsymbol{\theta})}{\mu}\boldsymbol{u}\right],$$

(iii) For any $\boldsymbol{\theta} \in \mathbb{R}^n$,

$$\begin{aligned} |f^\mu(\boldsymbol{\theta}) - f(\boldsymbol{\theta})| &\le \tfrac{\mu^2}{2} Ln, \\ \|\nabla f^\mu(\boldsymbol{\theta}) - \nabla f(\boldsymbol{\theta})\| &\le \tfrac{\mu}{2} L(n+3)^{\frac{3}{2}}, \\ \mathbb{E}_{\boldsymbol{u}}\left[\left\|\tfrac{f(\boldsymbol{\theta}+\mu\boldsymbol{u})-f(\boldsymbol{\theta})}{\mu}\boldsymbol{u}\right\|^2\right] &\le 2(n+4)\|\nabla f(\boldsymbol{\theta})\|^2 \\ &\quad + \tfrac{\mu^2}{2} L^2 (n+6)^3. \end{aligned}$$

The technique is called randomized smoothing, as the approximation function f^μ is smooth (cf. Assumption I for definition) even if the function f is not [41, lem. 2]. The computation of the gradient ∇f^μ in Proposition IV.1*(ii)* constitutes the key computational task for each agent in the proposed learning approach, as elaborated in the following section. Importantly, this gradient derivation solely relies on querying the zeroth-order oracle, i.e., the reward function value, using perturbed and unperturbed decision variables. It is noteworthy that throughout the learning process, the smooth approximation function f^μ is optimized, rather than the original function f. However, as indicated by Proposition IV.1*(iii)*, it becomes evident that by selecting a sufficiently small constant μ, the two functions can be made arbitrarily close to each other.

V. Proposed Distributed learning

In this section, we propose a distributed learning approach within a message-passing architecture. In this framework, agents collaborate to optimize their local policy parameters $\boldsymbol{\theta}_i$ gathered in $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_m)$ in (1).

In accordance with Proposition IV.1*(ii)*, each agent i can perform queries at time $t \in T_q^i$ by measuring its local reward function values and using the following expression:

$$\bar{r}_i^t := \frac{1}{\mu m B}\sum_{j=1}^{B}\left\{F_i(\boldsymbol{\theta}^t, \boldsymbol{\xi}^{t,j}) - F_i(\boldsymbol{\theta}^t + \mu\boldsymbol{u}^t, \bar{\boldsymbol{\xi}}^{t,j})\right\} \in \mathbb{R},$$
$$\text{where} \quad \boldsymbol{u}^t = (\boldsymbol{u}_1^t, \cdots, \boldsymbol{u}_m^t) \sim \mathcal{D}_u = \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{n\times n}). \tag{4}$$

Here, $t \in \mathbb{N}$ represents the global time, F_i is the reward function value (oracle) defined in Assumption I*(ii)*, and the summations are over two minibatches of random variables $\boldsymbol{\xi}^{t,j}$ and $\bar{\boldsymbol{\xi}}^{t,j}$ for $j \in [B]$. While the oracle F_i of agent i relies on the (perturbed) parameters of its neighbors $j \in \mathcal{N}_{tx}^i$, agent i does not require receiving the parameters $\boldsymbol{\theta}_j$. It only needs to query its own oracle, affected by $\boldsymbol{\theta}_j$ through their interaction.

It is essential to highlight that the query in (4) requires a certain level of synchrony among the agents. Specifically, the oracles $F_i(\boldsymbol{\theta}^t, \boldsymbol{\xi}^{t,j})$ and $F_i(\boldsymbol{\theta}^t + \mu\boldsymbol{u}^t, \bar{\boldsymbol{\xi}}^{t,j})$ necessitate the alignment of parameters $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_i + \mu\boldsymbol{u}_i$ in time for all agents $i \in [m]$ to ensure the validity of the queries. This implies that the agents are required to perturb their local parameters synchronously. This requirement arises solely from employing zeroth-order oracles and can be lifted if higher-order oracles (e.g., function gradients) are utilized. Nonetheless, by Assumption II, agents still have the flexibility to asynchronously

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

choose their preferred times T_u^i, T_q^i for updating, querying their oracles, and communicating with other agents, based on their heterogeneous communication and computational capabilities.

To update its parameter, each agent i requires not only its own queries $\bar{r}_i^t$ but also queries from agents whose performance depends on $\boldsymbol{\theta}_i$, namely the agents in the set $\mathcal{N}_{rx}^i$ defined in (2). Parameter sharing is achieved through message-passing, which is susceptible to random (but bounded) communication delays. Therefore, each agent i updates its parameter using potentially outdated queries received from its neighbors $\mathcal{N}_{rx}^i$. To establish the age of the queries, with a maximum $D^{\max} \geq 0$, consistent with the seminal work in [5], we introduce the timestamp $t - D^{\max} \leq \mathcal{T}_i^{j,t} \leq t$ defined as follows:

$$\begin{aligned}&\mathcal{T}_i^{j,t}\text{: the global time at which agent } j \text{ has queried}\\ &\text{its oracle and the corresponding query is stored}\\ &\text{in agent } i\text{'s local memory at time } t.\end{aligned} \tag{5}$$

It is evident that $\mathcal{T}_i^{i,t} = t, \forall i \in [m]$, $\mathcal{T}_i^{j,t} \in T_q^j$, and $0 \leq t - \mathcal{T}_i^{j,t} \leq D^{\max}$ indicates how old agent j's query stored in agent i's memory is, with a maximum age of $D^{\max}$. An example of a message-passing architecture and the involved quantities are depicted in Fig. 1.

Utilizing the quantities $\bar{r}_\ell^{\mathcal{T}_i^{\ell,t}}$ for $\ell \in \mathcal{N}_{rx}^i$ along with the vectors $\boldsymbol{u}_i^\tau$ for $\tau \in \{t - D^{\max}, \cdots, t\}$, stored in the local memories $\mathcal{B}_i$ and $\tilde{\mathcal{B}}_i$ respectively, each agent i performs an asynchronous update at the time of interest $t \in T_u^i$ following Algorithm 1. These updates are executed using stepsize sequences $(\gamma_i^t)_{t\in\mathbb{N}}$. To properly align the queries $\bar{r}_\ell^{\mathcal{T}_i^{\ell,t}}$ with the random perturbations $\boldsymbol{u}_i^{\mathcal{T}_i^{\ell,t}}$ in time, as required in (4), each agent i needs to know the global time $\mathcal{T}_i^{\ell,t}$ associated with the queries $\bar{r}_\ell^{\mathcal{T}_i^{\ell,t}}$ received from its neighbors $\ell \in \mathcal{N}_{rx}^i$. Therefore through message-passing, this global time is also exchanged, along with the corresponding query, to ensure the required alignment.

In Algorithm 1, it is evident that utilizing zeroth-order oracles by each agent has three advantages. First, the proposed distributed learning approach can handle reward functions for which higher-order oracles (e.g., function gradients) may not be available or are expensive to calculate (e.g., in simulation-based optimizations or bandit optimizations [19]). Second, unlike federated learning or learning over graph settings where agents share local parameter or gradient vectors, employing zeroth-order oracles significantly reduces the communication bandwidth requirements in Algorithm 1 as agents only share scalars, by which the required gradients can be estimated (cf. step 1.c and step 2 for message-passing; Proposition IV.1*(ii)* and step 7 for gradient estimation). Third, sharing scalars ensures privacy-preserving cooperation and addresses the known problem of privacy leakage in federated learning settings [42], [43].

An essential requirement for ensuring the convergence of Algorithm 1 is to establish an upper bound for $D^{\max}$, i.e., the maximum age of queries in each agent's local memory. This necessitates imposing appropriate assumptions on the message-passing architecture and the activity time of agents that define $D^{\max}$. We present these crucial assumptions in the subsequent section, along with additional necessary assumptions on the zeroth-order oracles.

Algorithm 1 Proposed asynchronous distributed learning

Initialize: $(\gamma_i^t)_{t=0}^{\bar{t}-1}$, $\boldsymbol{\theta}_i^0 \in \mathbb{R}^{n_i}$, $\bar{r}_\ell^0 = 0, \mathcal{T}_i^{\ell,0} = 0, \ell \in \mathcal{N}_{rx}^i$, $\mathcal{D}_{u_i} = \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \boldsymbol{u}_i^0 \sim \mathcal{D}_{u_i}, \mu > 0, B > 0$, T_q^i, T_u^i, $i \in [m]$

For $t = 0, \cdots, \bar{t} - 1$**, each agent** i **asynchronously:**

► *Quering and broadcasting:*

1: If $t \in T_q^i$:
   a: compute $\bar{r}_i^t$ using (4) with $\boldsymbol{\theta}_i^t$ and $\boldsymbol{u}_i^t$
   b: update the memory $\mathcal{B}_i$ by $\{\bar{r}_i^t, t\}$
   c: send $\{\bar{r}_i^t, t\}$ to the neighbors $j \in \mathcal{N}_{tx}^i$

► *Local memory updates:*

2: Receive new updates from the neighbors $\ell \in \mathcal{N}_{rx}^i$ denoted by $Q_\ell = \{r_\ell, \mathcal{T}_\ell\}$
3: For each $\ell \in \mathcal{N}_{rx}^i$, if $Q_\ell \neq \emptyset$:
   a: update the memory $\mathcal{B}_i$ by $\mathcal{T}_i^{\ell,t} \leftarrow \mathcal{T}_\ell$, $\bar{r}_\ell^{\mathcal{T}_i^{\ell,t}} \leftarrow r_\ell$
4: $\mathcal{T}_i^{\ell,t+1} = \mathcal{T}_i^{\ell,t}, \quad \forall \ell \in \mathcal{N}_{rx}^i$
5: If the memory $\tilde{\mathcal{B}}_i$ is full, omit the oldest entry
6: $\boldsymbol{u}_i^{t+1} \sim \mathcal{D}_{u_i}$ and save it in the local memory $\tilde{\mathcal{B}}_i$

► *Local updates:*

7: Using the locally stored $\boldsymbol{u}_i^\tau$ for $\tau \in \{t - D^{\max}, \cdots, t\}$ and $\bar{r}_\ell^{\mathcal{T}_i^{\ell,t}}$, $\ell \in \mathcal{N}_{rx}^i$ perform

$$\boldsymbol{\theta}_i^{t+1} = \boldsymbol{\theta}_i^t + \gamma_i^t \tilde{\boldsymbol{s}}_i^t \tag{6}$$

where $\tilde{\boldsymbol{s}}_i^t$ is defined by

$$\tilde{\boldsymbol{s}}_i^t := \begin{cases} \sum_{\ell\in\mathcal{N}_{rx}^i} \bar{r}_\ell^{\mathcal{T}_i^{\ell,t}} \boldsymbol{u}_i^{\mathcal{T}_i^{\ell,t}}, & \text{if} \quad t \in T_u^i, \\ \mathbf{0}, & \text{if} \quad t \notin T_u^i, \end{cases} \tag{7}$$

with $\mathcal{T}_i^{\ell,t}$ in (5)

VI. Convergence Study

A. Further definitions and assumptions

To facilitate the convergence study, we define the following vectors associated with agent i at time t:

$$\begin{aligned}\Theta_i^t &:= \text{vec}\{\boldsymbol{\theta}^{\mathcal{T}_i^{\ell,t}} \in \mathbb{R}^n \mid \ell \in \mathcal{N}_{rx}^i\} \in \mathbb{R}^{n|\mathcal{N}_{rx}^i|},\\ \mathbb{U}_i^t &:= \text{vec}\{\boldsymbol{u}^{\mathcal{T}_i^{\ell,t}} \in \mathbb{R}^n \mid \ell \in \mathcal{N}_{rx}^i\} \in \mathbb{R}^{n|\mathcal{N}_{rx}^i|},\end{aligned} \tag{8}$$

where $\text{vec}\{\cdot\}$ denotes the vectorization operator. Additionally, we define the following functions using the variables defined in (8):

$$\begin{aligned}\tilde{f}_i(\Theta_i^t) &:= \tfrac{1}{m} \textstyle\sum_{\ell\in\mathcal{N}_{rx}^i} f_\ell(\boldsymbol{\theta}^{\mathcal{T}_i^{\ell,t}}),\\ \tilde{f}_i^\mu(\Theta_i^t) &:= \tfrac{1}{m} \textstyle\sum_{\ell\in\mathcal{N}_{rx}^i} f_\ell^\mu(\boldsymbol{\theta}^{\mathcal{T}_i^{\ell,t}}),\\ f_\ell^\mu(\boldsymbol{\theta}^{\mathcal{T}_i^{\ell,t}}) &:= \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\Big[F_\ell(\boldsymbol{\theta}^{\mathcal{T}_i^{\ell,t}} + \boldsymbol{u}, \boldsymbol{\xi})\Big].\end{aligned} \tag{9}$$

The functions defined in (9) represent the objective functions by which agent i optimizes its local parameter $\boldsymbol{\theta}_i$. Based on (4), (7), Assumption I*(ii)*, and Proposition IV.1*(ii)*, it can be observed (cf. further discussion in Proposition A.4 in Appendix) that the update $\tilde{\boldsymbol{s}}_i^t$ in Algorithm 1 satisfies

$$\mathbb{E}_{\boldsymbol{\xi},\mathbb{U}}\big[\tilde{\boldsymbol{s}}_i^t\big] = -\nabla_i \tilde{f}_i^\mu(\Theta_i^t), \quad t \in T_u^i, \tag{10}$$

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

where ∇_i represents the gradient with respect to $\boldsymbol{\theta}_i \in \mathbb{R}^{n_i}$ defined as

$$\nabla_i \tilde{f}_i^{\mu}(\Theta_i^t) := \tfrac{1}{m} \textstyle\sum_{\ell \in \mathcal{N}_{rx}^i} \nabla_i f_\ell^{\mu}(\boldsymbol{\theta}^{\mathcal{T}_i^{\ell,t}}). \tag{11}$$

By considering (10), each agent i updates its local parameter $\boldsymbol{\theta}_i$ using the function $\tilde{f}_i^{\mu}(\Theta_i^t)$ with potentially outdated parameters Θ_i^t. In the subsequent section, we demonstrate that these outdated parameters eventually converge to a common parameter $\boldsymbol{\theta}^t$ with possibly different local parameters. To show this convergence, let us introduce another useful parameter associated with agent i:

$$\bar{\Theta}_i^t := (\overbrace{\boldsymbol{\theta}^t, \cdots, \boldsymbol{\theta}^t}^{|\mathcal{N}_{rx}^i| \text{ times}}) \in \mathbb{R}^{n|\mathcal{N}_{rx}^i|}. \tag{12}$$

In an ideal convergence scenario, we seek for $\Theta_i^t \to \bar{\Theta}_i^t$ (cf. Theorem VI.4), where the following holds for any $i \in [m]$:

$$\begin{aligned} \nabla_i \tilde{f}_i(\bar{\Theta}_i^t) &= \nabla_i f(\boldsymbol{\theta}^t), \\ \nabla_i \tilde{f}_i^{\mu}(\bar{\Theta}_i^t) &= \nabla_i f^{\mu}(\boldsymbol{\theta}^t), \end{aligned} \tag{13}$$

with f^{μ} defined in (3). Moreover, leveraging (6) and (7), it is useful to formulate the joint update of local parameters as:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \Gamma^t \tilde{\boldsymbol{s}}^t, \quad t \in \mathbb{N}, \tag{14}$$

where $\tilde{\boldsymbol{s}}^t := (\tilde{\boldsymbol{s}}_1^t, \cdots, \tilde{\boldsymbol{s}}_m^t) \in \mathbb{R}^n$ and $\Gamma^t := \operatorname{diag}\{\gamma_1^t \mathrm{I}_{n_1 \times n_1}, \cdots, \gamma_m^t \mathrm{I}_{n_m \times n_m}\}$ with identity matrices I with appropriate dimensions.

Ensuring a bounded delay is a crucial assumption in our convergence analysis (cf. Theorem VI.2). To establish such a guarantee, we impose the following assumption on the sets T_u^ℓ and T_q^ℓ in consistent with [5, Sec. 7]:

Assumption III (asynchrony level)**.** *There exists a finite $D > 0$ such that for all $\ell \in [m]$ and $t \in \mathbb{N}$, at least one of the elements in the set $\{t, t+1, \cdots, t+D-1\}$ belongs to the sets T_q^ℓ and T_u^ℓ. Moreover, the communication delay is upper bounded by D.*

Assumption III ensures that all agents actively participate in the learning process and that no agent remains idle indefinitely before the learning is completed. Based on this assumption, we can derive the following upper bound for the maximum age of received quantities $\bar{r}_\ell$:

Proposition VI.1 (maximum delay)**.** *There exists a finite $D^{\max} \geq D$ such that for all $i \in [m]$, $\ell \in \mathcal{N}_{rx}^i$, and $t \in \mathbb{N}$, there holds $t - D^{\max} \leq \mathcal{T}_i^{\ell,t} \leq t$.*

Proposition VI.1 states that agents update their parameters $\boldsymbol{\theta}_i^t \in \mathbb{R}^{n_i}$ using the outdated quantities $\bar{r}_\ell^{\mathcal{T}_i^{\ell,t}}, \ell \in \mathcal{N}_{rx}^i$ defined in (4), which are at most $D^{\max}$ time instants old. With $D^{\max}$ in Proposition VI.1, we introduce the next necessary assumption:

Assumption IV (local memory size)**.** *Each agent $i \in [m]$ has the memory $\tilde{\mathcal{B}}_i$ of size $D^{\max}+1$ to save the local perturbation vectors $\boldsymbol{u}_i^\tau \in \mathbb{R}^{n_i}$ for $t - D^{\max} \leq \tau \leq t$.*

Assumption IV ensures that agents can compute $\tilde{\boldsymbol{s}}_i^t$ in (7) using the saved perturbation vectors. It is remarked that the perturbation vectors are generated and saved locally, and they are not shared through message-passing.

Finally, we present the following required assumptions on the function gradients, oracles, and noise vectors:

Assumption V (boundedness)**.**

(i) $\mathbb{E}_{\boldsymbol{\xi}}\left[\|\nabla F_\ell(\boldsymbol{\theta}, \boldsymbol{\xi}) - \nabla f_\ell(\boldsymbol{\theta})\|^2\right] \leq \sigma^2, \forall \ell \in [m], \forall \boldsymbol{\theta} \in \mathbb{R}^n,$

(ii) $\|\nabla f_\ell(\boldsymbol{\theta})\| \leq G, \ \forall \ell \in [m], \forall \boldsymbol{\theta} \in \mathbb{R}^n,$

Assumption VI (noise vector)**.**

(i) $\mathbb{E}_{\boldsymbol{\xi}}\left[\|\boldsymbol{\xi}\|^2\right] \leq v^2;$

(ii) The stochastic zeroth-order oracle is Lipschitz continuous in $\boldsymbol{\xi}$, $\forall \ell \in [m]$ and $\forall \boldsymbol{\theta} \in \mathbb{R}^n$:

$$\left\|F_\ell(\boldsymbol{\theta}, \boldsymbol{\xi}) - F_\ell(\boldsymbol{\theta}, \boldsymbol{\xi}')\right\| \leq L_{\boldsymbol{\xi}} \left\|\boldsymbol{\xi} - \boldsymbol{\xi}'\right\|, \ \forall \boldsymbol{\xi}, \boldsymbol{\xi}' \in \mathbb{R}^s.$$

Assumptions V*(i)* and V*(ii)* are standard in the stochastic optimization literature [44]. Additional Assumptions VI*(i)* and VI*(ii)*, however, is required specifically concerning the noise vector. The additional assumptions are necessary because the quantity $\bar{r}_i^t$ in (4) is obtained using different noise vectors for the perturbed and unperturbed functions. While most existing works in literature, e.g., [19], [24], [44], [45], assume that the oracle $F_\ell(\cdot, \boldsymbol{\xi})$ can be queried at two different points $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ in $\mathbb{R}^n$ with a single noise vector $\boldsymbol{\xi}$, this assumption does not hold in many practical applications, including the use-case discussed in Section VIII. In these applications, allowing different noise vectors $\boldsymbol{\xi}$ at different points $\boldsymbol{\theta}$ makes the implementation of zeroth-order optimization possible, as queries can be taken at separate time slots. We will further examine the impact of this practical limitation on the convergence of Algorithm 1 in the subsequent section (cf. Corollary VI.3), building upon studies of asynchronous parallel and distributed computations presented in [5].

B. Main results

The subsequent results provide proof of convergence and convergence rate for the proposed asynchronous distributed learning approach in Algorithm 1, addressing problem (1). In the first theorem, we show that a subsequence of $(\Theta_\ell^t)_{t \in \mathbb{N}}$, i.e., of the outdated local parameters, converges to a neighborhood of some stationary point of the objective function $\tilde{f}_\ell$ by which agent ℓ optimizes its local parameter.

Theorem VI.2 (subsequential convergence)**.** *Take Assumptions I to VI, the smoothing parameter $\mu > 0$ defined in Proposition IV.1, the batch size B for all users in (4), and the stepsize sequences $(\gamma_i^t)_{t \in \mathbb{N}} \in (0, 1/M), i \in [m]$ where*

$$M := {}^{L(2mD^{\max}+1)}/_{2}.$$

Then for the iterates generated by Algorithm 1 at $\{0, 1, \cdots, \bar{t}\}$, the following holds for all $\ell \in [m]$:

$$\begin{aligned} \min_{t \in \{0, \cdots, \bar{t}\}} \mathbb{E}\left[\left\|\nabla_\ell \tilde{f}_\ell(\Theta_\ell^t)\right\|^2\right] &\leq \frac{1}{2m} \mu^2 L^2 (n+3)^3 \\ &+ \frac{2\mu^2 L n + 2(f(\boldsymbol{\theta}^0) - f^\star) + 2M \frac{\tilde{\sigma}^2}{B} \sum_{t=0}^{\bar{t}} \sum_{\ell=1}^m (\gamma_\ell^t)^2}{\eta \sum_{t=0}^{\bar{t}} \bar{\gamma}^t}, \end{aligned} \tag{15}$$

where

$$\eta := 1 - M\gamma^{\max}, \ \gamma^{\max} := \max_{i \in [m], t \in \mathbb{N}} [\gamma_i^t], \ \bar{\gamma}^t := \min_{i \in [m]} [\gamma_i^t] \forall t,$$

$$\tilde{\sigma}^2 := 4(n+4)(\sigma^2 + G^2) + \mu^2 L^2 (n+6)^3 + 4 \frac{B L_{\boldsymbol{\xi}}^2}{\mu^2} v^2,$$

and $f^\star$ is the minimum of the function f.

Proof: Refer to Appendix B. ∎

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

The same dependency on the problem dimension n is observed in existing works on nonconvex zeroth-order settings, e.g., [19], [41], [44], [45]. Furthermore, the stepsize range provided in Theorem VI.2 is twice as large as the range reported in [44], when there is no delay, i.e., $D^{\max} = 0$. However, it is evident from the defined parameter M that the introduction of delay due to asynchrony and communication delays requires the stepsize to decrease linearly with both the number of agents m and the delay in order to ensure convergence.

An important distinction in the complexity (15) lies in the presence of the last term in the variance term $\tilde{\sigma}^2$ which makes the tuning of variable μ challenging. In (15), it is evident that a smaller μ results in a smaller first term in the right-hand side; however, it also increases the variance $\tilde{\sigma}^2$. In Theorem VI.4, we will see that this results in convergence to a point closer to the stationary point but at a slower rate. It is essential to note that this term arises solely from the practical constraint that prohibits querying the oracle in two different points with the same noise vector. The following corollary establishes the iteration complexity when this constraint is lifted. The proof of this corollary is omitted here due to space limitations.

Corollary VI.3. *Take Assumptions I to V. In addition, it is assumed that the oracles $F_\ell(\cdot, \boldsymbol{\xi}), \forall \ell \in [N]$ can be queried in any two points $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^n$. Hence, the queries in (4) are performed by similar noise vectors in perturbed and unperturbed oracles for each j. Then for the iterates generated by Algorithm 1 the iteration complexity in (15) holds with the noise variance $\tilde{\sigma}^2 := 8(n+4)(\sigma^2 + G^2) + 2\mu^2 L^2 (n+6)^3$.*

We proceed to demonstrate the convergence of the original function f in (1), the outdated parameters $\boldsymbol{\theta}^{\mathcal{T}_\ell^{i,t}}$ to a common parameter $\boldsymbol{\theta}^t$, and the function values $f_i(\boldsymbol{\theta}^{\mathcal{T}_\ell^{i,t}})$.

Theorem VI.4 (asymptotic convergence and convergence rate)**.** *Take the assumptions in Corollary VI.3 and that the steptizes are diminishing such that $\lim_{t\to\infty} \gamma_i^t = 0, i \in [m]$. Then for all $\ell \in [m]$:*

(i) $\lim_{t\to\infty} \mathbb{E}\left[\left\|\bar{\Theta}_\ell^t - \Theta_\ell^t\right\|^2\right] = 0;$

(ii) $\lim_{t\to\infty} \mathbb{E}\left[\left|f_i(\boldsymbol{\theta}^t) - f_i(\boldsymbol{\theta}^{\mathcal{T}_\ell^{i,t}})\right|\right] = 0, \quad \forall i \in \mathcal{N}_{rx}^\ell.$

Moreover, if the diminishing stepsize sequences are defined by $\gamma_i^t = \gamma_i^0/\sqrt{t+r}$ with some $r > 0$, then

$$\min_{t\in\{0,\cdots,\bar{t}\}} \mathbb{E}\left[\left\|\nabla f(\boldsymbol{\theta}^t)\right\|^2\right] \leq \mathcal{O}(\mu^2) + \mathcal{O}(1/\sqrt{\bar{t}}). \tag{16}$$

Proof: Refer to Appendix B. ∎

According to (16), a subsequence of $(\boldsymbol{\theta}^t)_{t\in\mathbb{N}}$ converges to a neighborhood of some stationary point of f, and the size of this neighborhood is determined by the value of μ. This convergence rate is consistent with existing works addressing zeroth-order nonconvex problems with synchronous updates, e.g., [19], [44], [45]. The rate $\mathcal{O}(1/\sqrt{\bar{t}})$ can be established if the parameter μ also diminishes, i.e., is set according to $\mathcal{O}(1/\sqrt{\bar{t}})$ [44], [45]. In addition, if the batch size increases proportionally to $\mathcal{O}(\sqrt{\bar{t}})$, the rate $\mathcal{O}(1/t)$ is achieved [19]. If the assumption that the oracle $F_\ell(\cdot, \boldsymbol{\xi})$ can be queried at two different points $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ in $\mathbb{R}^n$ with a single noise vector $\boldsymbol{\xi}$ does not hold, which is the case in many practical applications, including the use-case considered in Section VIII, the convergence rate in (16) becomes $\mathcal{O}(\mu^2) + \mathcal{O}(1/(\mu^2\sqrt{\bar{t}}))$, which can be concluded from the proof in Appendix B. Hence, in this case, a smaller μ guarantees convergence to a tighter neighborhood of some stationary point; however, the convergence rate becomes slower due to the last term.

VII. Implementation of the Proposed Distributed Learning Approach in RA Problem

In this section, we explore the application of the proposed distributed learning in DL-based RA for communication networks, as an example (cf. Section II for a brief introduction). In the considered RA scenario, we aim to maximize the sum utility of agents (representing the global reward function) in the communication network, which can be formulated by (1). Since typical utility/loss functions in RA, such as data rate, communication latency, and appropriate variants that guarantee fairness, energy efficiency, etc., satisfy the assumptions considered in Assumptions I, V and VI, the learning process can be addressed by the proposed learning approach in a distributed manner, benefiting from the advantages mentioned in Section I-B. It is important to note that for the considered example, in practice, each agent i, acting as a transmitter, can have a significant impact on the utility of a large number of agents due to cross-talk channels. For the purposes of this section, we assume that each agent i influences the utility of all the other agents. Therefore, the neighbor sets defined in (2) are $\mathcal{N}_{rx}^i = [m]$ and $\mathcal{N}_{tx}^i = [m]$, necessitating a fully connected message-passing architecture where each agent communicates with all other agents, which is not always a practical solution. In this case, agent i may choose to communicate with only a few agents (immediate neighbors) to gather the necessary information. Hence, the queries $\bar{r}_\ell, \forall \ell \in [m]$ required by each agent i are passed through intermediate agents and immediate neighbors (possibly with delays) to reach agent i. Consequently, in this application, it is necessary to redefine the message-passing architecture among agents and to establish new sets of neighbors.

A. Setup

The agent activity time is the same as the one defined in Section III-A. We now establish new sets of neighbors.

1) Underlying graph: Let us consider a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where the set of vertices is denoted as $\mathcal{V} := [m]$ and the corresponding edges are represented by $\mathcal{E} := \{a_{ij} \in \{0,1\} \mid i,j \in [m]\}$. In this graph, we set $a_{ij} = 1$ if agent i transmits the necessary quantities to agent j, and $a_{ij} = 0$ if there is no communication from agent i to agent j. It is important to note that a_{ij} can be different from a_{ji}. The neighbor sets of agent i, consisting of the agents that send and receive quantities to and from agent i, respectively, are defined as follows:

$$\begin{aligned} \bar{\mathcal{N}}_{rx}^i &:= \{j \in [m] \mid a_{ji} = 1\}, \\ \bar{\mathcal{N}}_{tx}^i &:= \{j \in [m] \mid a_{ij} = 1\}. \end{aligned} \tag{17}$$

The underlying graph is assumed to be fixed and depends on practical considerations, e.g., existing backhaul communication links, etc. In the proposed learning approach, it is essential for the underlying graph to be connected.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

Assumption VII (connected graph $\mathcal{G}$ [16])**.** *The underlying graph $\mathcal{G}$ is* connected *such that any two vertices are linked in both directions either directly or through other vertices. The two directions possibly are two different paths in the graph.*

2) Communication between the agents: In the considered application, each agent affects the performance of all other agents through interaction (via cross-talk channels in the communication network). Therefore, it is important that each agent gathers the quantities $\bar{r}_\ell$ from all agents. Since each agent communicates only with its immediate neighbors defined in (17), it is crucial for each quantity to be associated with an agent ID as well. Hence, throughout the learning process, each agent i stores and sends its query $\bar{r}_i^t$ along with a timestamp t and the agent ID i. Similarly, the received quantities $\bar{r}_\ell^{\mathcal{T}_i^{\ell,t}}$ from the neighboring agents $j \in \bar{\mathcal{N}}_{rx}^i$ are also stored in the memory $\mathcal{B}_i$ along with their corresponding timestamps $\mathcal{T}_i^{\ell,t}$ and agent IDs $\ell \in [m]$. It is more practical if agent i engages in communication with its neighbors $\bar{\mathcal{N}}_{rx}^i$ and $\bar{\mathcal{N}}_{tx}^i$ only at specific times of interest during the learning process. The time instants for communication are defined by the set T_{tx}^i for agent i. At a given time $t \in T_{tx}^i$, agent i transmits all the quantities $\bar{r}_\ell^{\mathcal{T}_i^{\ell,t}}, \ell \in [m]$ (global information) stored in its memory to its neighbors $j \in \bar{\mathcal{N}}_{tx}^i$. The communication delay within the graph is random and bounded.

Since the graph $\mathcal{G}$ is connected, two agents may be connected through multiple paths. This means that agent i may receive multiple quantities from agent $\ell \neq i$ with possibly different delays. In such cases, agent i has the option to discard the older quantities and retain the most recent ones.

B. Algorithm

The learning process requires the active participation of all agents to ensure that no agent is isolated and that each agent can transmit and receive the necessary information. This cooperative participation is guaranteed by Assumption VII. In addition, it is important to control the level of asynchrony for the newly introduced communication set T_{tx}^ℓ in order to bound the maximum delay $D^{\max}$ during the learning. This can be done, similar to Assumption III, by ensuring that each agent communicates at least every $D > 0$ time instants.

Based on Assumption VII, as well as the introduced communication setup between the agents in Section VII-A, the following proposition is ensured:

Proposition VII.1 (maximum delay and global information)**.** *There exist a finite $D^{\max} \geq D$ such that for $t \geq D^{\max}$, each agent i has $\bar{r}_\ell^{\mathcal{T}_i^{\ell,t}}$ from all agents $\ell \in [m]$ (global information) saved in its local memory, where $t - D^{\max} \leq \mathcal{T}_i^{\ell,t} \leq t$.*

Proposition VII.1 emphasizes that although agents may communicate with only their immediate neighbors, they gather *global* information from all the other agents through message passing. It takes $D^{\max}$ time instants for this global information to propagate throughout the graph. Moreover, similar to Proposition VI.1, the quantities are at most $D^{\max}$ time instants old in each agent's memory.

Algorithm 2 Proposed asynchronous distributed learning in RA

Initialize: $(\gamma_i^t)_{t=0}^{\bar{t}-1}$, $\boldsymbol{\theta}_i^0 \in \mathbb{R}^{n_i}$, $\bar{r}_\ell^0 = 0, \mathcal{T}_i^{\ell,0} = 0, \ell \in [m]$, $\mathcal{D}_{u_i} = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $\boldsymbol{u}_i^0 \sim \mathcal{D}_{u_i}, \mu > 0, B > 0$, T_q^i, T_u^i, T_{tx}^i $i \in [m]$

For $t = 0, \cdots, \bar{t} - 1$, each agent i asynchronously:

► *Quering:*

If $t \in T_q^i$:
- a: compute $\bar{r}_i^t$ using (4) with $\boldsymbol{\theta}_i^t$ and $\boldsymbol{u}_i^t$
- b: update the memory $\mathcal{B}_i$ by $\{\bar{r}_i^t,$ ID: i, time: $t\}$

► *Broadcasting:*

If $t \in T_{tx}^i$:
- a: send the quantities in $\mathcal{B}_i$ to the neighbors $j \in \bar{\mathcal{N}}_{tx}^i$

► *Local memory updates:*

Receive new updates from the neighbors $j \in \bar{\mathcal{N}}_{rx}^i$ denoted by $Q_\ell = \{r_\ell, \ell, \mathcal{T}_\ell\}, \forall \ell \in [m]$

For each $\ell \in [m]$, if $Q_\ell \neq \emptyset$:
- a: update the memory $\mathcal{B}_i$ by $\mathcal{T}_i^{\ell,t} \leftarrow \mathcal{T}_\ell$, $\bar{r}_\ell^{\mathcal{T}_i^{\ell,t}} \leftarrow r_\ell$

$\mathcal{T}_i^{\ell,t+1} = \mathcal{T}_i^{\ell,t}, \quad \forall \ell \in [m]$

If the memory $\tilde{\mathcal{B}}_i$ is full, omit the oldest entry

$\boldsymbol{u}_i^{t+1} \sim \mathcal{D}_{u_i}$ and save it in the local memory $\tilde{\mathcal{B}}_i$

► *Local updates:*

Update $\boldsymbol{\theta}_i^t$ by (6) and (7), using the locally stored $\boldsymbol{u}_i^\tau$ for $\tau \in \{t - D^{\max}, \cdots, t\}$ and global information $\bar{r}_\ell^{\mathcal{T}_i^{\ell,t}}, \forall \ell \in [m]$

Since the underlying graph is not fully connected, each agent i receives quantities $\bar{r}_j, \forall j \in [m]$ with delays as they may pass through intermediate agents who may decide to transmit these quantities only at specific time instants of interest T_{tx}^ℓ. The proposed asynchronous learning approach for DL-based RA is summarized in Algorithm 2.

The sets T_q^i, T_{tx}^i, and T_u^i can be customized by agent i based on factors such as its computational capacity, communication bandwidth, application layer, and other considerations. By specifying these sets, agent i has the flexibility to determine when to query its reward, when to communicate with its neighbors, and when to update its local parameters.

It is important to note that the sets T_q^i, T_{tx}^i, and T_u^i are not required to be known to agent i before the learning process begins. Rather, they are defined to facilitate the study of the proposed learning approach.

VIII. Numerical Experiments

In the numerical experiments, our focus is on the application of power allocation in wireless communication networks, where each agent i, representing a transmitter, has its own DNN, π_i, for adjusting its transmit power, i.e., $p_i = \pi_i(\boldsymbol{\theta}_i, \boldsymbol{\mathcal{H}}_i) \in [0, 10]$, where $\boldsymbol{\theta}_i$ is the local parameter vector and $\boldsymbol{\mathcal{H}}_i$ represents the input of the DNN defined in the sequel. The utility of each agent is defined by $r_i^\pi = R_i(\boldsymbol{p}, \boldsymbol{H})$,

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

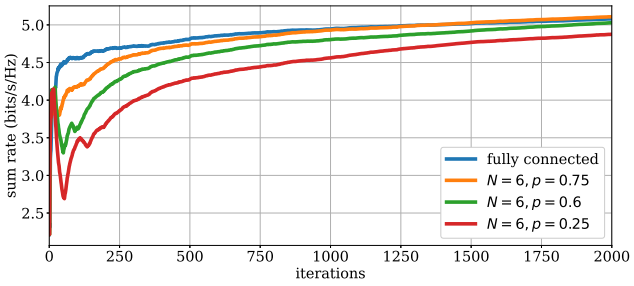
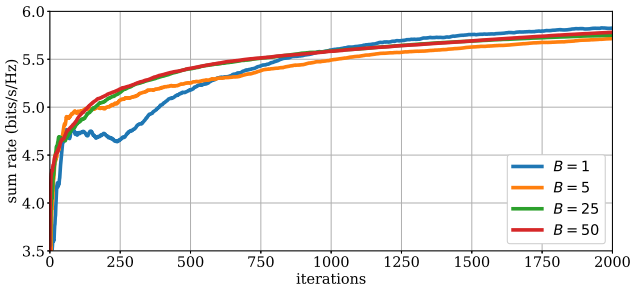
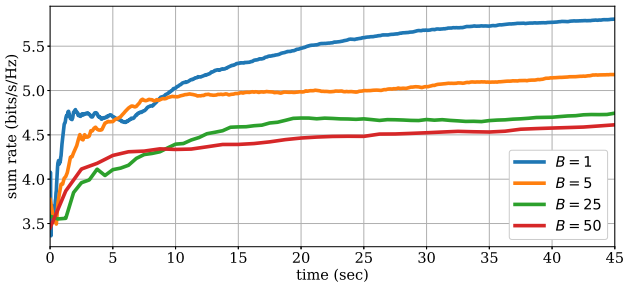


Figure 2: *Convergence of the proposed asynchronous distributed learning approach with different values of maximum delay $D^{\max}$, defined by (5) and Proposition VII.1. There are $m = 24$ agents, each communicating with N neighboring agents, and participating in training with probability p. The comparison is against centralized learning with fully connected agents ($N = m$) in synchronous mode and with no communication delay ($p = 1$).*



(a) *Convergence versus iteration number.*



(b) *Convergence versus global time.*

Figure 3: *Convergence of the proposed asynchronous distributed learning approach with different batch sizes B defined in (4). There are $m = 24$ agents, each communicating with $N = 4$ neighboring agents, and participating in training with a probability of $p = 0.9$. The duration of one iteration in (b) is set equal to the typical average channel coherence time of 25 ms.*

where

$$R_i(\boldsymbol{p}, \boldsymbol{H}) = \log\left(1 + \frac{|h_{r(i)i}|^2 p_i}{1 + \sum_{j\in[m]\setminus i} |h_{r(i)j}|^2 p_j}\right) \tag{18}$$

represents the achievable data rate of agent i, $r(\cdot)$, h_{ji}, and $\boldsymbol{H}$ are defined in Section II, and $\boldsymbol{p} = (p_1, \cdots, p_m)$ containts the transmit power of all agents. The objective of the distributed learning process is to maximize the expected global reward, i.e., the sum data rate in the communication network.

Wireless communication network setup: We consider m agents within the communication network. Each agent i is randomly located within the area $l(i) \in [-m, m]^2$. The corresponding receiver $r(i)$ is then randomly located within the area $l(r(i)) \in [l(i) - m/4, l(i) + m/4]$. The considered channel coefficients $h_{r(j)i}$ consist of both large-scale fading, which includes the pathloss gain $h^p_{r(j)i} = \|l(i) - l(r(j))\|^{-2.2}, \forall i, j \in [m]$, and small-scale fading represented by $h^f_{r(j)i} \sim \mathcal{CN}(0, 1)$, with a circularly-symmetric complex normal distribution, hence $h_{r(j)i} = h^p_{r(j)i} h^f_{r(j)i}$. Transmitter-receiver pairs that are far from each other may have negligible channel gain, and hence, we exclude them from the policy inputs, defined in the next paragraph. For this, we set $\eta = 0.01$ and use the modified channel $\tilde{\boldsymbol{H}} \in \mathbb{C}^{|\mathcal{R}|\times|\mathcal{T}|}$ whose elements are $\tilde{h}_{r(j)i} = h_{r(j)i}$ if $|h_{r(j)i}| \geq \eta$, and $\tilde{h}_{r(j)i} = 0$ otherwise.

RA setup: We employ aggregation graph neural networks (GNNs) proposed in [32]. For each policy i, the input is the aggregation sequence $\boldsymbol{\mathcal{H}}_i(t) := (y^0_i(t), \cdots, y^{K-1}_i(t)) \in \mathbb{R}^K$ where $y^k_i(t)$ is the ith element of the vector $\boldsymbol{y}^k(t) := \left[\prod_{t'=0}^{k-1} |\tilde{\boldsymbol{H}}[t - t']|\right] \mathbf{1} \in \mathbb{R}^m$. Here, $K = 5$ defines the range of neighbors (hops) that each agent considers in its input, $\mathbf{1} \in \mathbb{R}^m$ represents a vector with all ones, and $|\cdot|$ is the absolute value operator. The RA scheme is distributed, meaning that each agent can compute its policy input locally only through message passing with its neighbors (cf. [32, eq. (2)]).

A fully connected DNN with the structure of $\{K, 30, 30, 1\}$ neurons, $\mathrm{relu}(x) = \max\{0, x\}$ activation functions, and sigmoid output function is considered as the policy, hence, $\pi_i(\boldsymbol{\theta}_i, \boldsymbol{\mathcal{H}}_i(t)) = \mathrm{DNN}(\boldsymbol{\theta}_i, \boldsymbol{\mathcal{H}}_i(t))$.

Hyperparameters and the graph $\mathcal{G}^t$: Consider $A_{m\times m}$ as a matrix consisting of a_{ij}, defined in Section VII-A1. To ensure that the agents lie on a connected graph $\mathcal{G}^t$ to meet Assumption VII during the distributed learning, we set a_{ij} such that matrix A is a block diagonal matrix with overlapping blocks. These blocks are square matrices with all ones. The size of the square matrices is κ, with overlaps of $\kappa/2$ elements. Hence, $N := |\bar{\mathcal{N}}^i_{rx}| = |\bar{\mathcal{N}}^i_{tx}| = \kappa + \kappa/2, \forall i \in [m]$. The communication delay between two directly connected agents is assumed to be one time increment. During the numerical experiments, the following parameters in Algorithm 2 are set, unless otherwise specified: $\mu = 2, |\mathcal{B}_i| = m, |\tilde{\mathcal{B}}_i| = 40$, step-sizes $\gamma^t_i = \frac{0.5}{(t+1)^{0.25}}$, and the batch size $B = 20$. The other parameters will be specified in the sequel. Each agent i decides to update its parameter $\boldsymbol{\theta}_i$ with a probability of p_u. Moreover, each agent decides to transmit quantities to each of its neighbors in $\bar{\mathcal{N}}^i_{tx}$ with a probability of p_{tx}.

Centralized learning is considered the benchmark, where individual policies are trained synchronously on a server. It is noted that this scenario can also be considered as decentralized learning where the underlying graph $\mathcal{G}$ is fully connected, there is no communication delay, and the agents are synchronous, i.e., $T^i_{tx} = T^i_u = T^i_q = \mathbb{N}$. The experiments are repeated 5 times with different random seeds and the average is reported.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

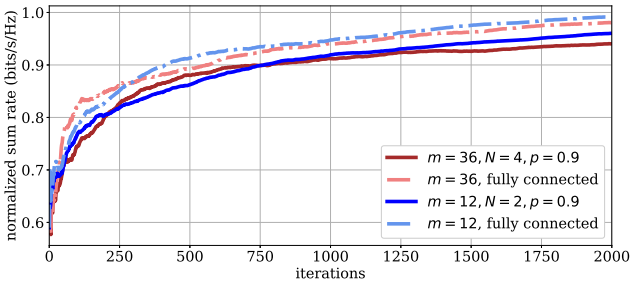


Figure 4: *Convergence of the proposed asynchronous distributed learning approach with different numbers of agents m. Each agent communicates with N neighboring agents and participates in training with probability p.. The performance is normalized by the maximum achieved sum rate for each m. The comparison is against centralized learning with fully connected agents ($N = m$) in synchronous mode and with no communication delay ($p = 1$).*

not.	Description	not.	Description
m	number of agents	γ_ℓ	stepsize
f_ℓ	local reward function	T_q^ℓ	set of query instants
$\boldsymbol{\theta}_\ell$	local parameter vector	T_u^ℓ	set of update instants
F_ℓ	local reward value	$\mathcal{B}_\ell, \tilde{\mathcal{B}}_\ell$	local memories
f_ℓ^μ	smooth approx. function	B	batch size
$\tilde{f}_\ell$	surrogate local function	Θ_ℓ	effective parameters
$\mathcal{N}_{rx}^\ell$	neighbor set	$\tilde{\boldsymbol{s}}_\ell$	estimated gradient
$\mathcal{N}_{tx}^\ell$	neighbor set	$\boldsymbol{u}_\ell$	perturbation vector
$\bar{r}_\ell$	shared quantities	$D^{\max}$	maximum delay time
μ	smoothing parameter	$\boldsymbol{\xi}$	noise/sample vector

Table I: *List of notations*

Results: In the first experiment, we assess the performance of the proposed learning approach under varying probabilities $p := p_u = p_{tx}$, with $m = 24$ agents. This parameter, along with the number of neighboring agents N, directly impacts the maximum delay $D^{\max}$ in Theorem VI.2 which determines the maximum age of the stored quantities $\bar{r}_i$ in each agent's memory. A lower probability p results in a longer propagation time for the quantities $\bar{r}_i$ throughout the network (cf. Section VII-B for further explanations). The observations in Fig. 2, where the performance is reported versus iteration number t, are aligned with the theoretical findings in Theorem VI.2, as increasing $D^{\max}$ leads to slower convergence. In addition, convergence is comparable with centralized learning, even with probabilities as low as $p = 0.25$.

Fig. 3a presents the impact of batch size on the convergence rate. The findings are consistent with theoretical results, as larger batch sizes lead to faster convergence. However, it is important to note that agents require more time to accumulate a larger number of samples for larger batch sizes. As a result, the overall convergence becomes slower over time for larger batch sizes. This phenomenon is demonstrated in Fig. 3b, where the duration of one increment of global time is set equal to the typical average channel coherence time of 25 ms in wireless communication networks.

The convergence of the proposed approach is also depicted for different numbers of agents, specifically $m = 12$ and $m = 36$ in Fig. 4.

IX. Conclusion

In this paper, we have investigated the problem of distributed learning where agents collaborate on a common task while they also have distinct individual tasks. In this context, the performance of each agent is influenced by other agents through interactions. To optimize their local parameters, agents utilize zeroth-order oracles and exchange their local queries through a message-passing architecture, subject to random but bounded communication delays. Importantly, the shared quantities are scalar, ensuring efficient communication bandwidth and preserving agents' privacy. Moreover, the agents engage in distributed learning in an asynchronous manner. We have also established convergence analyses for smooth nonconvex problems, achieving the same convergence rate as existing learning approaches with synchronous updates. According to the results, the stepsize needs to be decreased for larger delays.

As a use-case, the relevant problem of DL-based RA in communication networks is addressed, where agents, acting as transmitters, collaborate in training their local DNNs using the proposed learning approach to maximize the expected global reward in the communication network. An interesting direction for future research is to extend the proposed learning approach to nonsmooth settings and investigate its application in other learning tasks.

Appendix A
Preliminaries

Frequently used notations throughout the paper are listed in Table I. Define the following filtration:

$$\mathcal{F}^t := \text{filtration}\{\boldsymbol{\xi}^k, \boldsymbol{u}^k \mid k \in [t-1]\}. \tag{19}$$

The filtration $\mathcal{F}^t$ includes all the randomness involved up to the time instant t. During the proof, we make use of the following Young's inequality for any $a, b \geq 0$,

$$ab \leq {}^{a^2}\!/_2 + {}^{b^2}\!/_2, \tag{20}$$

and the following lemma and propositions:

Lemma A.1 (descent lemma [46, Prop. A.24])**.** *Given a Lipschitz continuous and L_g-smooth function $g : \mathbb{R}^n \to \mathbb{R}$, the following holds for all $\boldsymbol{\theta}', \boldsymbol{\theta} \in \mathbb{R}^n$:*

$$g(\boldsymbol{\theta}') \leq g(\boldsymbol{\theta}) + \langle \nabla g(\boldsymbol{\theta}), \boldsymbol{\theta}' - \boldsymbol{\theta} \rangle + \frac{L_g}{2} \left\| \boldsymbol{\theta}' - \boldsymbol{\theta} \right\|^2.$$

Proposition A.2 (Lipschitz continuity and differentiability of $\tilde{f}_\ell$)**.** *The function $\tilde{f}_\ell$ defined in (9) is Lipschitz continuous and $\tilde{L}$-smooth with some constants $\tilde{L}^0 \in [0, {}^{L^0}\!/_{\sqrt{m}}]$ and $\tilde{L} \in [0, {}^{L}\!/_{\sqrt{m}}]$, where L^0 and L are defined in Assumption I. Namely, take $\Theta_\ell, \Theta'_\ell \in \mathbb{R}^{n|\mathcal{N}_{rx}^\ell|}$, then*

$$\left\| \nabla_\ell \tilde{f}_\ell(\Theta_\ell) - \nabla_\ell \tilde{f}_\ell(\Theta'_\ell) \right\| \leq \tilde{L} \|\Theta_\ell - \Theta'_\ell\|, \ \forall \Theta_\ell, \Theta'_\ell \in \mathbb{R}^{n|\mathcal{N}_{rx}^\ell|}.$$

Proof: By (9), we rewrite the lhs as:

$$\left\| \frac{1}{m} \sum_{i \in \mathcal{N}_{rx}^\ell} \left(\nabla_\ell f_i(\boldsymbol{\theta}^i) - \nabla_\ell f_i(\boldsymbol{\theta}'^i) \right) \right\|^2$$
$$\leq \frac{1}{m} \sum_{i \in \mathcal{N}_{rx}^\ell} \left\| \nabla_\ell f_i(\boldsymbol{\theta}^i) - \nabla_\ell f_i(\boldsymbol{\theta}'^i) \right\|^2$$

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

$$\leq \frac{L^2}{m} \sum_{i \in \mathcal{N}_{rx}^\ell} \left\| \boldsymbol{\theta}^i - \boldsymbol{\theta}'^i \right\|^2 = \frac{L^2}{m} \|\Theta_\ell - \Theta'_\ell\|^2,$$

where $\boldsymbol{\theta}^i \in \mathbb{R}^n$ and $\boldsymbol{\theta}'^i \in \mathbb{R}^n$ are ith block coordinates of Θ_ℓ and Θ'_ℓ, respectively. The first inequality is due to Jensen's inequality, the second is due to Assumption I(i), and the last equality is due to the definition of Θ_ℓ and Θ'_ℓ. The same reasoning can be leveraged for the Lipschitz constant $\tilde{L}^0$. ∎

Proposition A.3 (Lipschitz differentiability of f_ℓ^μ and $\tilde{f}_\ell^\mu$)**.** *For all $\ell \in [m]$, the functions f_ℓ^μ defined in (9) are L_μ-smooth with $L_\mu \leq L$. Moreover, take Θ_ℓ^t and $\bar{\Theta}_\ell^t$ defined in (8) and (12), respectively. The following is true for the function $\tilde{f}_\ell^\mu$ in (9):*

$$\|\nabla_\ell \tilde{f}_\ell^\mu(\bar{\Theta}_\ell^t) - \nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\| \leq \frac{L}{m} \sum_{i \in \mathcal{N}_{rx}^\ell} \sum_{k \in [m]} \left\| \boldsymbol{\theta}_k^t - \boldsymbol{\theta}_k^{\mathcal{T}_\ell^{i,t}} \right\|, \quad \forall \ell \in [m]$$

Proof: The L_μ-smoothness of the functions f_ℓ^μ can be verified by Assumption I(i) and Proposition IV.1(i). For the inequality, consider

$$\begin{aligned}\|\nabla_\ell \tilde{f}_\ell^\mu(\bar{\Theta}_\ell^t) - \nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\| &\leq \frac{1}{m} \sum_{i \in \mathcal{N}_{rx}^\ell} \left\| \nabla_\ell f_i^\mu(\boldsymbol{\theta}^t) - \nabla_\ell f_i^\mu(\boldsymbol{\theta}^{\mathcal{T}_\ell^{i,t}}) \right\| \\ &\leq \frac{L}{m} \sum_{i \in \mathcal{N}_{rx}^\ell} \left\| \boldsymbol{\theta}^t - \boldsymbol{\theta}^{\mathcal{T}_\ell^{i,t}} \right\|,\end{aligned}$$

where the first inequality uses the definition of the function in (9), and the second inequality considers the L_μ-smoothness of the involved functions and $L_\mu \leq L$ guaranteed by Proposition IV.1(i). ∎

Proposition A.4 (unbiased oracle $\tilde{\boldsymbol{s}}_\ell^t$)**.** *For all $\ell \in [m]$ and $t \in T_u^\ell$, $\tilde{\boldsymbol{s}}_\ell^t$ defined in (7) is unbiased:*

$$\mathbb{E}_{\boldsymbol{\xi},\mathbb{U}}\left[\tilde{\boldsymbol{s}}_\ell^t\right] = -\nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t),$$

where $\tilde{f}_\ell^\mu$ is defined in (9).

Proof: Based on the timestamp $\mathcal{T}_\ell^{i,t}$ stored in $\mathcal{B}_\ell$, user ℓ can determine that the vector $\boldsymbol{u}_\ell^{\mathcal{T}_\ell^{i,t}}$ stored in $\tilde{\mathcal{B}}_\ell$ was used during the measurement of $\bar{r}_i^{\mathcal{T}_\ell^{i,t}}$ by user i. This inference is facilitated by Assumptions II and IV. Consequently, user ℓ can establish (7) by leveraging the definition of $\bar{r}_i$ in (4), Proposition IV.1(ii), and the unbiased nature of the oracle F_i as indicated by Assumption I(ii). This series of reasoning allows for the derivation of the equality. ∎

Proposition A.5 (oracle $\tilde{\boldsymbol{s}}_\ell^t$ with bounded variance)**.** *For all $\ell \in [m]$ and $t \in T_u^\ell$, $\tilde{\boldsymbol{s}}_\ell^t$ defined in (7) is bounded:*

$$\mathbb{E}_{\boldsymbol{\xi},\mathbb{U}}\left[\left\| \tilde{\boldsymbol{s}}_\ell^t + \nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t) \right\|^2\right] \leq {\tilde{\sigma}^2}/{B},$$

where B is the batchsize used by all the users in (4) to derive $\tilde{\boldsymbol{s}}_\ell^t$ and

$$\tilde{\sigma}^2 := 4(n+4)(\sigma^2 + G^2) + \mu^2 L^2 (n+6)^3 + 4\frac{BL_{\boldsymbol{\xi}}^2}{\mu^2} v^2.$$

Proof: The proof follows the proof in [44, thm. 4] with necessary modifications. Considering (7) and (9) and $\mathcal{T}$ instead of $\mathcal{T}_\ell^{i,t}$ for the sake of presentation, we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi},\mathbb{U}}\left[\left\| \tilde{\boldsymbol{s}}_\ell^t + \nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t) \right\|^2\right] \\
&= \mathbb{E}\left[\left\| \sum_{i \in \mathcal{N}_{rx}^\ell} \left(\bar{r}_i^{\mathcal{T}} \boldsymbol{u}_\ell^{\mathcal{T}} + \frac{1}{m} \nabla_\ell f_i^\mu(\boldsymbol{\theta}^{\mathcal{T}}) \right) \right\|^2\right] \\
&\leq m \sum_{i \in \mathcal{N}_{rx}^\ell} \mathbb{E}_{\boldsymbol{\xi},\mathbb{U}}\left[\left\| \bar{r}_i^{\mathcal{T}} \boldsymbol{u}_\ell^{\mathcal{T}} + \frac{1}{m} \nabla_\ell f_i^\mu(\boldsymbol{\theta}^{\mathcal{T}}) \right\|^2\right] \\
&= \frac{1}{m} \sum_{i \in \mathcal{N}_{rx}^\ell} \\
&\mathbb{E}\left[\left\| \frac{\frac{1}{B}\sum_{q=1}^B \left(F_i(\boldsymbol{\theta}^{\mathcal{T}}, \boldsymbol{\xi}^q) - F_i(\boldsymbol{\theta}^{\mathcal{T}} + \mu \boldsymbol{u}^{\mathcal{T}}, \boldsymbol{\xi}^q) \right)}{\mu} \boldsymbol{u}_\ell^{\mathcal{T}} \right.\right. \\
&\left.\left. + \nabla_\ell f_i^\mu(\boldsymbol{\theta}^{\mathcal{T}}) + \frac{\frac{1}{B}\sum_{q=1}^B \left(F_i(\boldsymbol{\theta}^{\mathcal{T}}, \boldsymbol{\xi}'^q) - F_i(\boldsymbol{\theta}^{\mathcal{T}}, \boldsymbol{\xi}^q) \right)}{\mu} \boldsymbol{u}_\ell^{\mathcal{T}} \right\|^2\right] \\
&\leq \frac{2}{B^2 m} \sum_{i \in \mathcal{N}_{rx}^\ell} \sum_{q=1}^B \\
&\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\left[\left\| \frac{F_i(\boldsymbol{\theta}^{\mathcal{T}}, \boldsymbol{\xi}^q) - F_i(\boldsymbol{\theta}^{\mathcal{T}} + \mu \boldsymbol{u}^{\mathcal{T}}, \boldsymbol{\xi}^q)}{\mu} \boldsymbol{u}_\ell^{\mathcal{T}} + \nabla_\ell f_i^\mu(\boldsymbol{\theta}^{\mathcal{T}}) \right\|^2\right] \\
&+ \frac{2}{\mu^2 B m} \sum_{i \in \mathcal{N}_{rx}^\ell} \sum_{q=1}^B \\
&\mathbb{E}_{\boldsymbol{\xi}}\left[\left\| F_i(\boldsymbol{\theta}^{\mathcal{T}}, \boldsymbol{\xi}^q) - F_i(\boldsymbol{\theta}^{\mathcal{T}}, \boldsymbol{\xi}'^q) \right\|^2\right] \mathbb{E}_{\boldsymbol{u}}\left[\left\| \boldsymbol{u}_\ell^{\mathcal{T}} \right\|^2\right] \qquad (21)
\end{aligned}$$

where in the first inequality Jensen's inequality is applied with taking into account $|\mathcal{N}_{rx}^\ell| \leq m$. In the second equality, $\bar{r}_i$ is expanded using (4), where the batch size is B. Also, $F_i(\boldsymbol{\theta}^{\mathcal{T}}, \boldsymbol{\xi}^q)$ is added and subtracted. For the last inequality, Jensen's inequality is applied for both terms, while for the first term Proposition IV.1(ii) is also considered. We continue to bound the first term in the last inequality:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\left[\left\| \frac{F_i(\boldsymbol{\theta}^{\mathcal{T}}, \boldsymbol{\xi}^q) - F_i(\boldsymbol{\theta}^{\mathcal{T}} + \mu \boldsymbol{u}^{\mathcal{T}}, \boldsymbol{\xi}^q)}{\mu} \boldsymbol{u}_\ell^{\mathcal{T}} + \nabla_\ell f_i^\mu(\boldsymbol{\theta}^{\mathcal{T}}) \right\|^2\right] \\
&\leq \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{u}}\left[\left\| \frac{F_i(\boldsymbol{\theta}^{\mathcal{T}} + \mu \boldsymbol{u}^{\mathcal{T}}, \boldsymbol{\xi}^q) - F_i(\boldsymbol{\theta}^{\mathcal{T}}, \boldsymbol{\xi}^q)}{\mu} \boldsymbol{u}_\ell^{\mathcal{T}} \right\|^2\right] \\
&\leq 2(n+4)\mathbb{E}_{\boldsymbol{\xi}}\left[\left\| \nabla F_i(\boldsymbol{\theta}^{\mathcal{T}}, \boldsymbol{\xi}^q) - \nabla f_i(\boldsymbol{\theta}^{\mathcal{T}}) + \nabla f_i(\boldsymbol{\theta}^{\mathcal{T}}) \right\|^2\right] \\
&+ \frac{\mu^2}{2} L^2 (n+6)^3 \\
&\leq 2(n+4)(\sigma^2 + G^2) + \frac{\mu^2}{2} L^2 (n+6)^3,
\end{aligned}$$

where in the first inequality $\nabla_\ell f_i^\mu(\boldsymbol{\theta}^{\mathcal{T}})$ is omitted by the fact that the variance is upperbounded by the second moment. We use Proposition IV.1(iii) in the second inequality, and Assumptions V(i) and V(ii) in the last inequality. The last term in (21) is also bounded by

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\xi}}\left[\left\| F_i(\boldsymbol{\theta}^{\mathcal{T}}, \boldsymbol{\xi}^q) - F_i(\boldsymbol{\theta}^{\mathcal{T}}, \boldsymbol{\xi}'^q) \right\|^2\right] &\leq L_{\boldsymbol{\xi}}^2 \mathbb{E}_{\boldsymbol{\xi}}\left[\left\| \boldsymbol{\xi}^q - \boldsymbol{\xi}'^q \right\|^2\right] \\ &\leq 2L_{\boldsymbol{\xi}}^2 v^2,\end{aligned}$$

leveraging Assumptions VI(i) and VI(ii). Considering $\mathbb{E}\left[\left\| \boldsymbol{u}_\ell^{\mathcal{T}} \right\|^2\right] \leq 1$ due to its distribution and putting the last two bounds back into (21), where $|\mathcal{N}_{rx}^\ell| \leq m$, completes the

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

proof. ■

APPENDIX B
CONVERGENCE STUDY

Proof of Theorem VI.2 *(subsequential convergence):* We make use of the descent lemma (cf. Lemma A.1) on the L_μ-smooth function $f^\mu(\boldsymbol{\theta}^t)$ with an expectation conditioned on the filtration $\mathcal{F}^t$, defined in (19). Note that as conditioned on $\mathcal{F}^t$, all the iterates up to and including $\boldsymbol{\theta}^t$ are deterministic:

$$\begin{aligned}
&\mathbb{E}\big[f^\mu(\boldsymbol{\theta}^{t+1}) \mid \mathcal{F}^t\big] \le f^\mu(\boldsymbol{\theta}^t) + \mathbb{E}\big[\langle \nabla f^\mu(\boldsymbol{\theta}^t), \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t\rangle \mid \mathcal{F}^t\big] \\
&+ \frac{L_\mu}{2}\mathbb{E}\Big[\big\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t\big\|^2 \mid \mathcal{F}^t\Big] \qquad (22)\\
&\le f^\mu(\boldsymbol{\theta}^t) + \sum_{\ell=1}^m \mathbb{E}\big[\langle \nabla_\ell f^\mu(\boldsymbol{\theta}^t), \gamma_\ell^t \tilde{\boldsymbol{s}}_\ell^t\rangle \mid \mathcal{F}^t\big] \\
&+ \frac{L}{2}\sum_{\ell=1}^m (\gamma_\ell^t)^2 \mathbb{E}\Big[\big\|\tilde{\boldsymbol{s}}_\ell^t\big\|^2 \mid \mathcal{F}^t\Big] \\
&= f^\mu(\boldsymbol{\theta}^t) + \sum_{\ell=1}^m \mathbb{E}\Big[\langle \nabla_\ell \tilde{f}_\ell^\mu(\bar{\Theta}_\ell^t), \gamma_\ell^t \tilde{\boldsymbol{s}}_\ell^t\rangle \mid \mathcal{F}^t\Big] \\
&+ \frac{L}{2}\sum_{\ell=1}^m (\gamma_\ell^t)^2 \mathbb{E}\Big[\big\|\tilde{\boldsymbol{s}}_\ell^t\big\|^2 \mid \mathcal{F}^t\Big] \\
&= f^\mu(\boldsymbol{\theta}^t) + \sum_{\ell=1}^m \mathbb{E}\Big[\langle \nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t), \gamma_\ell^t \tilde{\boldsymbol{s}}_\ell^t\rangle \mid \mathcal{F}^t\Big] \\
&+ \frac{L}{2}\sum_{\ell=1}^m (\gamma_\ell^t)^2 \mathbb{E}\Big[\big\|\tilde{\boldsymbol{s}}_\ell^t\big\|^2 \mid \mathcal{F}^t\Big] \\
&+ \sum_{\ell=1}^m \mathbb{E}\Big[\langle \nabla_\ell \tilde{f}_\ell^\mu(\bar{\Theta}_\ell^t) - \nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t), \gamma_\ell^t \tilde{\boldsymbol{s}}_\ell^t\rangle \mid \mathcal{F}^t\Big],
\end{aligned}$$

where in the second inequality $L_\mu \le L$ is considered by Proposition IV.1*(i)* and (14) is employed, in the first equality (13) is used. We proceed by bounding the last term in (22) as

$$\begin{aligned}
&\sum_{\ell=1}^m \mathbb{E}\Big[\langle \nabla_\ell \tilde{f}_\ell^\mu(\bar{\Theta}_\ell^t) - \nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t), \gamma_\ell^t \tilde{\boldsymbol{s}}_\ell^t\rangle \mid \mathcal{F}^t\Big] \qquad (23)\\
&\le \sum_{\ell=1}^m \Big\|\nabla_\ell \tilde{f}_\ell^\mu(\bar{\Theta}_\ell^t) - \nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\Big\| \mathbb{E}\big[\big\|\gamma_\ell^t \tilde{\boldsymbol{s}}_\ell^t\big\| \mid \mathcal{F}^t\big] \\
&\le \frac{L}{m}\sum_{\ell=1}^m \sum_{i\in\mathcal{N}_{rx}^\ell} \sum_{k=1}^m \Big\|\boldsymbol{\theta}_k^t - \boldsymbol{\theta}_k^{\mathcal{T}_\ell^{i,t}}\Big\| \mathbb{E}\big[\big\|\gamma_\ell^t \tilde{\boldsymbol{s}}_\ell^t\big\| \mid \mathcal{F}^t\big] \\
&\le \frac{L}{m}\sum_{\ell=1}^m \sum_{i\in\mathcal{N}_{rx}^\ell} \sum_{k=1}^m \sum_{\tau=\mathcal{T}_\ell^{i,t}}^{t-1} \mathbb{E}\big[\gamma_k^\tau \gamma_\ell^t \|\tilde{\boldsymbol{s}}_k^\tau\| \big\|\tilde{\boldsymbol{s}}_\ell^t\big\| \mid \mathcal{F}^t\big] \\
&\le \frac{L}{m}\sum_{\ell=1}^m \sum_{i\in\mathcal{N}_{rx}^\ell} \sum_{k=1}^m \sum_{\tau=t-D^{\max}}^{t-1} \\
&\mathbb{E}\left[\frac{(\gamma_k^\tau)^2\|\tilde{\boldsymbol{s}}_k^\tau\|^2}{2} + \frac{(\gamma_\ell^t)^2\big\|\tilde{\boldsymbol{s}}_\ell^t\big\|^2}{2} \mid \mathcal{F}^t\right] \\
&\le \frac{mL}{2}\sum_{\ell=1}^m \sum_{\tau=t-D^{\max}}^{t-1} \mathbb{E}\Big[(\gamma_\ell^\tau)^2\|\tilde{\boldsymbol{s}}_\ell^\tau\|^2 + (\gamma_\ell^t)^2\big\|\tilde{\boldsymbol{s}}_\ell^t\big\|^2 \mid \mathcal{F}^t\Big],
\end{aligned}$$

where the second inequality is due to Proposition A.3, in the third inequality (6) is employed, and in the fourth inequality, Young's inequality in (20) and Proposition VI.1 are considered. The last inequality is also due to the bound $|\mathcal{N}_{rx}^\ell| \le m$.

Putting (23) into (22) results in

$$\begin{aligned}
&\mathbb{E}\big[f^\mu(\boldsymbol{\theta}^{t+1}) \mid \mathcal{F}^t\big] \qquad (24)\\
&\le f^\mu(\boldsymbol{\theta}^t) + \sum_{\ell=1}^m \mathbb{E}\Big[\langle \nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t), \gamma_\ell^t \tilde{\boldsymbol{s}}_\ell^t\rangle \mid \mathcal{F}^t\Big] \\
&+ L\frac{mD^{\max}+1}{2}\sum_{\ell=1}^m (\gamma_\ell^t)^2 \mathbb{E}\Big[\big\|\tilde{\boldsymbol{s}}_\ell^t\big\|^2 \mid \mathcal{F}^t\Big] \\
&+ \frac{mL}{2}\sum_{\ell=1}^m \sum_{\tau=t-D^{\max}}^{t-1} (\gamma_\ell^\tau)^2 \|\tilde{\boldsymbol{s}}_\ell^\tau\|^2.
\end{aligned}$$

Note that by (10):

$$\sum_{\ell=1}^m \mathbb{E}\Big[\langle \nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t), \gamma_\ell^t \tilde{\boldsymbol{s}}_\ell^t\rangle \mid \mathcal{F}^t\Big] = -\sum_{\ell=1}^m \gamma_\ell^t \Big\|\nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\Big\|^2$$

Moreover, by Propositions A.4 and A.5,

$$\begin{aligned}
\mathbb{E}\Big[\big\|\tilde{\boldsymbol{s}}_\ell^t\big\|^2 \mid \mathcal{F}^t\Big] &= \mathbb{E}\Big[\Big\|\tilde{\boldsymbol{s}}_\ell^t + \nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t) - \nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\Big\|^2 \mid \mathcal{F}^t\Big] \\
&\le \tilde{\sigma}^2/B + \Big\|\nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\Big\|^2. \qquad (25)
\end{aligned}$$

Taking the total expectation of (24) by applying the tower rule and then utilizing the above relations lead to:

$$\begin{aligned}
&\mathbb{E}\big[f^\mu(\boldsymbol{\theta}^{t+1})\big] \le \mathbb{E}\big[f^\mu(\boldsymbol{\theta}^t)\big] - \sum_{\ell=1}^m \gamma_\ell^t \mathbb{E}\Big[\Big\|\nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\Big\|^2\Big] \\
&+ L\frac{mD^{\max}+1}{2}\sum_{\ell=1}^m (\gamma_\ell^t)^2 \mathbb{E}\Big[\big\|\tilde{\boldsymbol{s}}_\ell^t\big\|^2\Big] \\
&+ L\frac{m}{2}\sum_{\ell=1}^m \sum_{\tau=t-D^{\max}}^{t-1} (\gamma_\ell^\tau)^2 \mathbb{E}\Big[\|\tilde{\boldsymbol{s}}_\ell^\tau\|^2\Big] \\
&\le \mathbb{E}\big[f^\mu(\boldsymbol{\theta}^t)\big] - \sum_{\ell=1}^m \gamma_\ell^t \mathbb{E}\Big[\Big\|\nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\Big\|^2\Big] \\
&+ L\frac{mD^{\max}+1}{2}\sum_{\ell=1}^m (\gamma_\ell^t)^2 \mathbb{E}\Big[\Big\|\nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\Big\|^2\Big] \\
&+ L\frac{m}{2}\sum_{\ell=1}^m \sum_{\tau=t-D^{\max}}^{t-1} (\gamma_\ell^\tau)^2 \mathbb{E}\Big[\Big\|\nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^\tau)\Big\|^2\Big] \\
&+ L\frac{mD^{\max}+1}{2}\frac{\tilde{\sigma}^2}{B}\sum_{\ell=1}^m (\gamma_\ell^t)^2 \\
&+ L\frac{m}{2}\frac{\tilde{\sigma}^2}{B}\sum_{\ell=1}^m \sum_{\tau=t-D^{\max}}^{t-1} (\gamma_\ell^\tau)^2.
\end{aligned}$$

Let us telescope this inequality from $t=0$ until $t=\bar{t}$ to have

$$\begin{aligned}
\mathbb{E}\Big[f^\mu(\boldsymbol{\theta}^{\bar{t}+1})\Big] &\le f^\mu(\boldsymbol{\theta}^0) - \sum_{t=0}^{\bar{t}}\sum_{\ell=1}^m \gamma_\ell^t \mathbb{E}\Big[\Big\|\nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\Big\|^2\Big] \\
&+ M\sum_{t=0}^{\bar{t}}\sum_{\ell=1}^m (\gamma_\ell^t)^2 \mathbb{E}\Big[\Big\|\nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\Big\|^2\Big] \\
&+ M\frac{\tilde{\sigma}^2}{B}\sum_{t=0}^{\bar{t}}\sum_{\ell=1}^m (\gamma_\ell^t)^2,
\end{aligned}$$

where $\sum_{t=0}^{\bar{t}}\sum_{\tau=t-D^{\max}}^{t-1}\{\cdot\} \le D^{\max}\sum_{t=0}^{\bar{t}}\{\cdot\}$ is considered and

$$M := L\frac{2mD^{\max}+1}{2}$$

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

is defined. Also, $\boldsymbol{\theta}^0$ is assumed to be deterministic due to initialization in Algorithm 1. Rearranging the above inequality

$$\eta \sum_{t=0}^{\bar{t}} \bar{\gamma}^t \sum_{\ell=1}^{m} \mathbb{E}\left[\left\|\nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\right\|^2\right] \leq f^\mu(\boldsymbol{\theta}^0) - f^{\mu^\star} + M\frac{\tilde{\sigma}^2}{B}\sum_{t=0}^{\bar{t}}\sum_{\ell=1}^{m}(\gamma_\ell^t)^2, \tag{26}$$

where

$$\eta := 1 - M\gamma^{\max},\ \gamma^{\max} := \max_{i\in[m],t\in\mathbb{N}}[\gamma_i^t],\ \bar{\gamma}^t := \min_{i\in[m]}[\gamma_i^t],\ \forall t,$$

and $\mathbb{E}\left[f^\mu(\boldsymbol{\theta}^{\bar{t}+1})\right] \geq f^{\mu^\star}$ is considered where $f^{\mu^\star}$ minimizes f^μ. In (26), $\eta > 0$ guarantees a descent, hence $\gamma^{\max} < 1/M$ requires to hold. Finally, we devide both sides of (26) by $\sum_{t=0}^{\bar{t}} \bar{\gamma}^t$ and then lower bound the lhs as

$$\begin{aligned}
&\eta \sum_{t=0}^{\bar{t}} \frac{\bar{\gamma}^t}{\sum_{k=0}^{\bar{t}}\bar{\gamma}^k} \sum_{\ell=1}^{m} \mathbb{E}\left[\left\|\nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\right\|^2\right] \\
&\geq \eta \sum_{t=0}^{\bar{t}} \frac{\bar{\gamma}^t}{\sum_{k=0}^{\bar{t}}\bar{\gamma}^k} \mathbb{E}\left[\left\|\nabla_i \tilde{f}_i^\mu(\Theta_i^t)\right\|^2\right] \\
&\geq \min_{t\in\{0,\cdots,\bar{t}\}} \eta\mathbb{E}\left[\left\|\nabla_i \tilde{f}_i^\mu(\Theta_i^t)\right\|^2\right], \quad \forall i \in [m],
\end{aligned}$$

where the weighted average is lowerbounded by the $\min\{\cdot\}$ operator in the last inequality. Hence, by (26), $\forall i \in [m]$

$$\begin{aligned}
&\min_{t\in\{0,\cdots,\bar{t}\}} \mathbb{E}\left[\left\|\nabla_i \tilde{f}_i^\mu(\Theta_i^t)\right\|^2\right] \\
&\leq \sum_{t=0}^{\bar{t}} \frac{\bar{\gamma}^t}{\sum_{k=0}^{\bar{t}}\bar{\gamma}^k} \sum_{\ell=1}^{m} \mathbb{E}\left[\left\|\nabla_\ell \tilde{f}_\ell^\mu(\Theta_\ell^t)\right\|^2\right] \qquad (27)\\
&\leq \frac{f^\mu(\boldsymbol{\theta}^0) - f^{\mu^\star} + M\frac{\tilde{\sigma}^2}{B}\sum_{t=0}^{\bar{t}}\sum_{\ell=1}^{m}(\gamma_\ell^t)^2}{\eta\sum_{k=0}^{\bar{t}}\bar{\gamma}^k}.
\end{aligned}$$

Consider

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla_i \tilde{f}_i(\Theta_i^t) - \nabla_i \tilde{f}_i^\mu(\Theta_i^t) + \nabla_i \tilde{f}_i^\mu(\Theta_i^t)\right\|^2\right] \qquad (28)\\
&\leq \frac{1}{2m}\mu^2 L^2 (n+3)^3 + 2\mathbb{E}\left[\left\|\nabla_i \tilde{f}_i^\mu(\Theta_i^t)\right\|^2\right],
\end{aligned}$$

due to Proposition IV.1*(iii)* and the fact that $\tilde{f}_i$ is $\tilde{L}$-smooth with $\tilde{L} \leq \frac{1}{\sqrt{m}}L$ due to Proposition A.2. Also,

$$|(f^\mu(\boldsymbol{\theta}^0) - f^{\mu^\star}) - (f(\boldsymbol{\theta}^0) - f^\star)| \leq \mu^2 L n,$$

where Proposition IV.1*(iii)* is invoked. Substituting the above inequalities in (27) completes the proof. ∎

Proof of Theorem VI.4 *(asymptotic convergence and convergence rate):*

VI.4*(i)*: For $\ell \in [m]$

$$\begin{aligned}
&\mathbb{E}\left[\left\|\bar{\Theta}_\ell^t - \Theta_\ell^t\right\|^2\right] = \mathbb{E}\left[\sum_{i\in\mathcal{N}_{rx}^\ell}\sum_{k\in[m]}\left\|\boldsymbol{\theta}_k^t - \boldsymbol{\theta}_k^{\mathcal{T}_\ell^{i,t}}\right\|^2\right] \\
&= \mathbb{E}\left[\sum_{i\in\mathcal{N}_{rx}^\ell}\sum_{k\in[m]}\left\|\sum_{\tau=\mathcal{T}_\ell^{i,t}}^{t-1}\gamma_k^\tau \tilde{\boldsymbol{s}}_k^\tau\right\|^2\right] \\
&\leq D^{\max}\sum_{i\in\mathcal{N}_{rx}^\ell}\sum_{k\in[m]}\sum_{\tau=t-D^{\max}}^{t-1}(\gamma_k^\tau)^2\mathbb{E}\left[\left\|\tilde{\boldsymbol{s}}_k^\tau\right\|^2\right] \qquad (29)\\
&\leq m(D^{\max})^2\frac{\tilde{\sigma}^2}{B}\sum_{k\in[m]}(\gamma_k^{t-D^{\max}})^2 \\
&+ mD^{\max}\sum_{k\in[m]}\sum_{\tau=t-D^{\max}}^{t-1}(\gamma_k^\tau)^2\mathbb{E}\left[\left\|\nabla_k \tilde{f}_k^\mu(\Theta_k^\tau)\right\|^2\right],
\end{aligned}$$

where in the first equality, the definitions of $\bar{\Theta}_\ell^t$ and Θ_ℓ^t respectively in (8) and (12) are utilized, and in the first inequality, Proposition VI.1 and Jensen's inequality are invoked. Moreover, in the last inequality, the bound in (25) and $|\mathcal{N}_{rx}^\ell| \leq m$ are used. Also, since the stepsizes are diminishing, $\gamma_k^\tau \leq \gamma_k^{t-D^{\max}}, \forall \tau \in [t-D^{\max}, t-1], \forall k \in [m]$ is considered. The last term can still be upperbounded further by

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla_k \tilde{f}_k^\mu(\Theta_k^\tau)\right\|^2\right] \\
&= \mathbb{E}\left[\left\|\nabla_k \tilde{f}_k^\mu(\Theta_k^\tau) - \nabla_k \tilde{f}_k(\Theta_k^\tau) + \nabla_k \tilde{f}_k(\Theta_k^\tau)\right\|^2\right] \\
&\leq \frac{1}{2m}\mu^2 L^2 (n+3)^3 + 2\mathbb{E}\left[\left\|\nabla_k \tilde{f}_k(\Theta_k^\tau)\right\|^2\right] \\
&\leq \frac{1}{2m}\mu^2 L^2 (n+3)^3 + 2G^2 =: \Delta
\end{aligned}$$

where the first and second inequalities are due to Proposition IV.1*(iii)* and Assumption V*(ii)*, respectively. Putting the last bound back into (29) results in

$$\begin{aligned}
\mathbb{E}\left[\left\|\bar{\Theta}_\ell^t - \Theta_\ell^t\right\|^2\right] &\leq m(D^{\max})^2\tfrac{\tilde{\sigma}^2}{B}\textstyle\sum_{k\in[m]}(\gamma_k^{t-D^{\max}})^2 \qquad (30)\\
&+ m\Delta D^{\max}\textstyle\sum_{k\in[m]}\sum_{\tau=t-D^{\max}}^{t-1}(\gamma_k^\tau)^2
\end{aligned}$$

Since $D^{\max}$ is finite due to Assumption III and Proposition VI.1, and the stepsizes are diminishing, the lhs vanishes when $t \to \infty$.

VI.4*(ii)*: Using the Lipschitz continuity of f_i guaranteed by Assumption I*(i)*,

$$\mathbb{E}\left[|f_i(\boldsymbol{\theta}^t) - f_i(\boldsymbol{\theta}^{\mathcal{T}_\ell^{i,t}})|\right] \leq L^0\mathbb{E}\left[\left\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^{\mathcal{T}_\ell^{i,t}}\right\|\right] \to 0$$

as $t \to \infty$, due to VI.4*(i)*.

Eq. (16): Take

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla f(\boldsymbol{\theta}^t)\right\|^2\right] = \textstyle\sum_{\ell=1}^m \mathbb{E}\left[\left\|\nabla_\ell f(\boldsymbol{\theta}^t)\right\|^2\right] \\
&\leq 2\textstyle\sum_{\ell=1}^m \mathbb{E}\left[\left\|\nabla_\ell \tilde{f}_\ell(\bar{\Theta}_\ell^t) - \nabla_\ell \tilde{f}_\ell(\Theta_\ell^t)\right\|^2\right] \\
&+ 2\textstyle\sum_{\ell=1}^m \mathbb{E}\left[\left\|\nabla_\ell \tilde{f}_\ell(\Theta_\ell^t)\right\|^2\right] \\
&\leq 2\tilde{L}^2\textstyle\sum_{\ell=1}^m \mathbb{E}\left[\left\|\bar{\Theta}_\ell^t - \Theta_\ell^t\right\|^2\right] + 2\textstyle\sum_{\ell=1}^m \mathbb{E}\left[\left\|\nabla_\ell \tilde{f}_\ell(\Theta_\ell^t)\right\|^2\right],
\end{aligned}$$

where in the first inequality, Young's inequality and (13), and in the second inequality, $\tilde{L}$-smoothness of $\tilde{f}_\ell$ established by Proposition A.2 are considered. The bound is also valid if both sides are multiplied by $\alpha^t := \bar{\gamma}^t/\sum_{k=0}^{\bar{t}}\bar{\gamma}^k$ where $\bar{\gamma}^k$ is defined in (26) and are summed over t:

$$\begin{aligned}
&\textstyle\sum_{t=0}^{\bar{t}} \alpha^t \mathbb{E}\left[\left\|\nabla f(\boldsymbol{\theta}^t)\right\|^2\right] \\
&\leq 2\tilde{L}^2\textstyle\sum_{t=0}^{\bar{t}}\alpha^t\sum_{\ell=1}^m \mathbb{E}\left[\left\|\bar{\Theta}_\ell^t - \Theta_\ell^t\right\|^2\right] \\
&+ 2\textstyle\sum_{t=0}^{\bar{t}}\alpha^t\sum_{\ell=1}^m \mathbb{E}\left[\left\|\nabla_\ell \tilde{f}_\ell(\Theta_\ell^t)\right\|^2\right]. \qquad (31)
\end{aligned}$$

With the defined stepsize sequence, the first term in the rhs is diminishing wrt $\bar{t}$ due to (30). Similarly, the last term in the rhs is diminishing due to (27). Inequalities (27) and (30) are considered with $\tilde{\sigma}^2$ defined in Corollary VI.3 here since the assumption that the oracles $F_\ell(\cdot, \boldsymbol{\xi}), \forall \ell \in [N]$ can be queried in any two points holds. As the last term in (31) dominates the first term in the rhs, it can be concluded that the rhs diminishes with a rate of $\mathcal{O}(\mu^2) + \mathcal{O}(1/\sqrt{\bar{t}})$, due to (27) and (28). The lhs

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

can also be lowerbounded by the $\min\{\cdot\}$ operator, similar to (27), completing the proof. ■

REFERENCES

[1] L. Cannelli, F. Facchinei, V. Kungurtsev, and G. Scutari, "Asynchronous parallel algorithms for nonconvex optimization," *Mathematical Programming*, vol. 184, no. 1-2, pp. 121–154, Nov. 2020.

[2] Z. Peng, Y. Xu, M. Yan, and W. Yin, "ARock: An Algorithmic Framework for Asynchronous Parallel Coordinate Updates," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. A2851–A2879, Jan. 2016.

[3] J. Liu and S. J. Wright, "Asynchronous stochastic coordinate descent: Parallelism and convergence properties," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 351–376, 2015.

[4] P. Latafat and P. Patrinos, "Primal-dual algorithms for multi-agent structured optimization over message-passing architectures with bounded communication delays," *Optimization Methods and Software*, vol. 37, no. 6, pp. 2052–2079, Nov. 2022.

[5] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

[6] Y. Zhou, Y. Liang, Y. Yu, W. Dai, and E. P. Xing, "Distributed proximal gradient algorithm for partially asynchronous computer clusters," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 733–764, 2018.

[7] P. Latafat, N. M. Freris, and P. Patrinos, "A new randomized block-coordinate primal-dual proximal algorithm for distributed optimization," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4050–4065, 2019.

[8] R. Nassif, S. Vlaski, C. Richard, J. Chen, and A. H. Sayed, "Multitask learning over graphs: An approach for distributed, streaming machine learning," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 14–25, 2020.

[9] A. Narang, E. Faulkner, D. Drusvyatskiy, M. Fazel, and L. Ratliff, "Learning in stochastic monotone games with decision-dependent data," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 5891–5912.

[10] Y. Yang and J. Wang, "An overview of multi-agent reinforcement learning from game theoretical perspective," *arXiv preprint arXiv:2011.00583*, 2020.

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[12] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[13] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5693–5700.

[14] S. Vlaski and A. H. Sayed, "Distributed learning in non-convex environments—Part I: Agreement at a linear rate," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1242–1256, 2021.

[15] R. Nassif, S. Vlaski, C. Richard, and A. H. Sayed, "Learning Over Multitask Graphs—Part I: Stability Analysis," *IEEE Open Journal of Signal Processing*, vol. 1, pp. 28–45, 2020.

[16] A. Sayed, "Adaptation, Learning, and Optimization over Networks," *Foundations and Trends® in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[17] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed, "Decentralized consensus optimization with asynchrony and delays," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 2, pp. 293–307, 2017.

[18] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.

[19] D. Hajinezhad, M. Hong, and A. Garcia, "Zone: Zeroth-order nonconvex multiagent optimization over networks," *IEEE transactions on automatic control*, vol. 64, no. 10, pp. 3995–4010, 2019.

[20] Y. Zhang, P. Li, J. Hong, J. Li, Y. Zhang, W. Zheng, P.-Y. Chen, J. D. Lee, W. Yin, M. Hong *et al.*, "Revisiting zeroth-order optimization for memory-efficient LLM fine-tuning: A benchmark," *arXiv preprint arXiv:2402.11592*, 2024.

[21] Y. Zhang, Y. Yao, J. Jia, J. Yi, M. Hong, S. Chang, and S. Liu, "How to robustify black-box ML models? a zeroth-order optimization perspective," *arXiv preprint arXiv:2203.14195*, 2022.

[22] F. Zhou, G. Lu, M. Wen, Y.-C. Liang, Z. Chu, and Y. Wang, "Dynamic spectrum management via machine learning: State of the art, taxonomy, challenges, and open research issues," *IEEE Network*, vol. 33, no. 4, pp. 54–62, 2019.

[23] W. Tong and G. Y. Li, "Nine Challenges in Artificial Intelligence and Wireless Communications for 6G," *IEEE Wireless Communications*, pp. 1–10, 2022.

[24] D. S. Kalogerias, M. Eisen, G. J. Pappas, and A. Ribeiro, "Model-Free Learning of Optimal Ergodic Policies in Wireless Systems," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6272–6286, 2020.

[25] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards Optimal Power Control via Ensembling Deep Neural Networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1760–1776, Mar. 2020.

[26] P. Behmandpoor, J. Verdyck, and M. Moonen, "Deep learning-based cross-layer resource allocation for wired communication systems," *IEEE ICASSP International Conference on Acoustics, Speech and Signal Processing*, pp. 4120–4124, 2021.

[27] P. Behmandpoor, P. Patrinos, and M. Moonen, "Learning-based resource allocation with dynamic data rate constraints," *IEEE ICASSP International Conference on Acoustics, Speech and Signal Processing*, pp. 4088–4092, 2022.

[28] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep Learning for Radio Resource Allocation With Diverse Quality-of-Service Requirements in 5G," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2309–2324, Apr. 2021.

[29] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2282–2292, 2019.

[30] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep Reinforcement Learning Based Resource Allocation for V2V Communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[31] Y. S. Nasir and D. Guo, "Multi-Agent Deep Reinforcement Learning for Dynamic Power Allocation in Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Signal and Information Processing over Networks. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2024.3487421

[32] Z. Wang, M. Eisen, and A. Ribeiro, "Learning Decentralized Wireless Resource Allocations With Graph Neural Networks," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1850–1863, 2022.

[33] N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "Learning resilient radio resource management policies with graph neural networks," *IEEE Transactions on Signal Processing*, vol. 71, pp. 995–1009, 2023.

[34] M. Eisen and A. Ribeiro, "Optimal Wireless Resource Allocation With Random Edge Graph Neural Networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2977–2991, 2020.

[35] N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "State-augmented learnable algorithms for resource management in wireless networks," *IEEE Transactions on Signal Processing*, vol. 70, pp. 5898–5912, 2022.

[36] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "Graph Neural Networks for Scalable Radio Resource Management: Architecture Design and Theoretical Analysis," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 101–115, Jan. 2021.

[37] P. Behmandpoor, P. Patrinos, and M. Moonen, "Federated learning based resource allocation for wireless communication networks," *30th European Signal Processing Conference (EUSIPCO)*, pp. 1656–1660, 2022.

[38] M. Eisen, K. Gatsis, G. J. Pappas, and A. Ribeiro, "Learning in wireless control systems over nonstationary channels," *IEEE Transactions on Signal Processing*, vol. 67, no. 5, pp. 1123–1137, 2018.

[39] P. Behmandpoor, P. Patrinos, and M. Moonen, "Model-free decentralized training for deep learning based resource allocation in communication networks," *31th European Signal Processing Conference (EUSIPCO)*, pp. 1494–1498, 2023.

[40] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright, "Randomized smoothing for stochastic optimization," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 674–701, 2012.

[41] Y. Nesterov and V. Spokoiny, "Random Gradient-Free Minimization of Convex Functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, Apr. 2017.

[42] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM conference on computer communications*, 2019, pp. 2512–2520.

[43] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the ACM SIGSAC conference on computer and communications security*, 2017, pp. 603–618.

[44] S. Ghadimi, G. Lan, and H. Zhang, "Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization," *Mathematical Programming*, vol. 155, no. 1, pp. 267–305, Jan. 2016.

[45] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.

[46] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on November 19,2024 at 12:18:05 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.