Joint Resource Allocation to Minimize Execution Time of Federated Learning in Cell-Free Massive MIMO

Tung Thanh Vu, *Member, IEEE*, Duy Trong Ngo, *Senior Member, IEEE*, Hien Quoc Ngo, *Senior Member, IEEE*, Minh Ngoc Dao, Nguyen Hoang Tran, *Senior Member, IEEE*, and Richard H. Middleton, *Fellow, IEEE*

***Abstract*—Due to its communication efficiency and privacy-preserving capability, federated learning (FL) has emerged as a promising framework for machine learning in 5G-and-beyond wireless networks. Of great interest is the design and optimization of new wireless network structures that support the stable and fast operation of FL. Cell-free massive multiple-input–multiple-output (CFmMIMO) turns out to be a suitable candidate, which allows each communication round in the iterative FL process to be stably executed within a large-scale coherence time. Aiming to reduce the total execution time of the FL process in CFmMIMO, this article proposes choosing only a subset of available users to participate in FL. An optimal selection of users with favorable link conditions would minimize the execution time of each communication round while limiting the total number of communication rounds required. Toward this end, we formulate a joint optimization problem of user selection, transmit power, and processing frequency, subject to a predefined minimum number of participating users to guarantee the quality of learning. We then develop a new algorithm that is proven to converge to the neighborhood of the stationary points of the formulated problem. Numerical results confirm that our proposed approach significantly reduces the FL total execution time over baseline schemes. The time reduction is more pronounced when the density of access point deployments is moderately low.**

***Index Terms*—Cell-free massive MIMO, execution time minimization, federated learning (FL).**

I. Introduction

THE NUMBERS of mobile devices and connections have been growing significantly in recent years. According to Cisco [1], while the number of global mobile devices is expected to reach 13.1 billion by 2023, more than 10% of this figure will have 5G connections. These devices generate a vast amount of data, which in turn enable a wide range of on-device artificial intelligence (AI) services, such as traffic navigation, indoor localization, image recognition, natural language processing, and augmented reality [2]–[5]. However, it is impractical to use conventional centralized approaches to train AI models (especially those by deep neural networks) at mobile devices. As such approaches store and process data at distant cloud centers, they find it extremely challenging to support delay-critical applications. Moreover, uploading user raw data to distant cloud servers also raises serious concerns about user's data privacy [6].

Federated learning (FL) has recently emerged as a promising solution for the AI model training at wireless devices with a certain guarantee of data privacy [7]–[12]. An FL process is iterative and involves several communication rounds. In each communication round, users (UEs) compute their local model updates by using their local training data, followed by sending these updates to a central server. The central server aggregates the received local updates to generate a global model update, which is then sent back to the UEs for their subsequent local computation. The FL process terminates when a prescribed level of learning accuracy is attained; at which point, a learning model is established. Here, since only model updates (instead of raw training data) are shared between UEs and a central server, the data privacy of each UE is protected. Furthermore, as a model update is much smaller in size than raw training data, sending a model update requires a much shorter amount of time.

In the literature, there are two main research directions that study FL in wireless network environments. The learning-oriented direction aims to develop FL frameworks that improve learning performance such as test accuracy. They do so by mitigating the detrimental effects of wireless transmissions, such as channel fading and estimation errors, on FL [13]–[16]. On the other hand, the communication-oriented direction aims to develop communication schemes that optimize certain performance metrics for communications. Examples of these metrics include execution time (in seconds) and energy consumption (in Joule) of an FL process executed "over-the-air" [17]–[21].

The focus of this article is on the communication-oriented direction. Here, Yang *et al.* [17], Zeng *et al.* [18], [19], and

Manuscript received 21 January 2022; revised 3 May 2022; accepted 6 June 2022. Date of publication 15 June 2022; date of current version 24 October 2022. The work of Tung Thanh Vu and Hien Quoc Ngo was supported by the U.K. Research and Innovation Future Leaders Fellowships under Grant MR/S017666/1. The work of Minh Ngoc Dao was supported in part by the FMJH Program PGMO and in part by EDF. *(Corresponding author: Tung Thanh Vu.)*

Tung Thanh Vu and Hien Quoc Ngo are with the Institute of Electronics, Communications, and Information Technology, Queen's University Belfast, Belfast BT7 1NN, U.K. (e-mail: t.vu@qub.ac.uk; hien.ngo@qub.ac.uk).

Duy Trong Ngo and Richard H. Middleton are with the School of Engineering, The University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: duy.ngo@newcastle.edu.au; richard.middleton@newcastle.edu.au).

Minh Ngoc Dao is with the School of Engineering, Information Technology and Physical Sciences, Federation University, Ballarat, VIC 3353, Australia (e-mail: m.dao@federation.edu.au).

Nguyen Hoang Tran is with the School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: nguyen.tran@sydney.edu.au).

Digital Object Identifier 10.1109/JIOT.2022.3183295

2327-4662 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

Hu *et al.* [20] developed new wireless network designs that use frequency-division multiple access (FDMA) and time-division multiple access (TDMA) to support FL. In these works, the transmission time of each FL communication round could be significantly prolonged when the number of UEs is large. To enable stable and fast FL over wireless media, Vu *et al.* [21]–[23] proposed to use cell-free massive multiple-input–multiple-output (CFmMIMO) to assist the process of FL. In a CFmMIMO network, UEs are simultaneously served by a large number of distributed access points (APs) over the same frequency band, and hence, CFmMIMO can offer very high macrodiversity and multiplexing gains [24]. As a result, it can uniformly provide very high data rates for all users in the network which enables the stable and fast FL. Paper [21] shows that the CFmMIMO network can achieve a much shorter FL execution time than that of conventional TDMA/FDMA networks. We note that in a wireless network, the information transmitted is not necessarily private. However, in a wireless network with FL, user privacy is better protected since only the model updates (instead of raw data) are sent from devices to a central processing unit (CPU). While the risk of privacy leakage exists in a wireless FL network, it does not overtake the benefits of FL. We also note that more advanced FL algorithms to secure FL in wireless networks have been recently studied (see [25] and references therein).

Scope of Research: In this article, we employ CFmMIMO as the underlying wireless network structure and propose novel resource allocation schemes to support a wireless FL process. Here, we use a conjugate beamforming/matched filtering scheme at the APs. Implemented locally at each AP, this scheme has been shown to perform well when the number of APs is large. The overarching objective is to minimize the total FL execution time, which is a product of the number of FL communication rounds and the execution time of each round.

UE Selection: To achieve the above design objective, we propose allowing only a subset of UEs with favorable links to participate in the learning process. Doing so would reduce the execution time of one FL communication round, albeit at the expense of requiring more FL communication rounds for the FL process to converge. Our aim is to devise a UE selection policy that strikes an optimal balance between these two conflicting outcomes, thereby offering a minimum total FL execution time. This aim is different from that of improving the learning performance (such as robustness, test accuracy, and convergence rate) via UE selection, as advocated by [26]–[30].

It is worth noting that as the UE selection reduces the number of participating UEs, the test accuracy of an FL process is also affected [31]. On the other hand, without UE selection, the FL execution time could potentially be prolonged by the UEs with unfavorable link conditions. These UEs need much more time to transmit. Over a large service area and with a large number of UEs, it is more common to have these vulnerable UEs. Of our particular interest is how to find an optimal set of "sufficiently good" UEs to reduce the FL execution time, without satisfying the test accuracy too much. This important observation leads to two research questions.

TABLE I
FREQUENTLY USED SYMBOLS

Symbols	Definition
$\mathcal{N}, N$	Set of UEs and its cardinality
$\mathcal{M}, M$	Set of APs and its cardinality
N_{QoL}	Minimum number of UEs for quality of learning
T_c	(Small-scale) coherence time
$\widetilde{T}_c$	Large-scale coherence time
a_k	Selection variable of UE k
η_{mk}	Downlink power control coefficient of UE k at AP m
f_k	Processing frequency of UE k
ζ_k	Uplink power control coefficient of UE k
S_d, S_u	Sizes of global and local updates

Q1) What is the minimum number of participating UEs N_{QoL}, above which the test accuracy remains acceptable?

Q2) For a given N_{QoL}, how to select UEs with favorable links to minimize the total FL execution time?

In this article, we focus on Q2) and leave Q1) for future research.

New Optimization-Based Design and Results: UE selection has tight relationships with the allocation of other resources. Specifically, the decision in which UE is selected affects how to allocate transmit power and computing frequency, and *vice versa*. Therefore, it is not clear and difficult to see how to find manually or heuristically the favorable links and determine the exact number of the selected UEs for minimizing the FL execution time. Motivated by this observation, we propose a joint optimization approach for UE selection and resource allocation. In particular, to minimize the execution time of one FL process, the FL execution time is first formulated as a function of UE selection, transmit power, and computation frequency. A mixed-integer two-stage stochastic nonconvex problem of minimizing the FL execution time is then formulated and subjected to the constraints that reflex the practical relationships among all the variables, and the practical constraints on maximum powers at the APs and UEs, imperfect channel estimation, and a minimum number of selected UEs. The considered problem is different from that in [22] and [23] which aim to reduce the execution time in an FL communication round, rather than minimizing the total FL execution time.

To solve the formulated problem, we propose a new algorithm that is proven to converge to the neighborhood of its stationary points. The mathematical challenges lie in the binary nature of UE selection variables and the coupling among the optimizing variables. Structurewise, this optimization problem is much more complex than those considered in [21] and [23]. The algorithm developed in this article is able to handle the binary constraints efficiently, while meeting all the strict conditions for solving two-stage stochastic nonconvex problems [32]. Numerical results with practical parameter settings verify the convergence of the proposed algorithm. They also show that our UE selection approach can cut the total FL execution time by more than half compared to the baseline schemes.

Notation and Paper Organization: In this article, boldfaced symbols are used for vectors and capitalized boldfaced symbols for matrices. $\mathbb{R}^d$ is a space where its elements are real

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

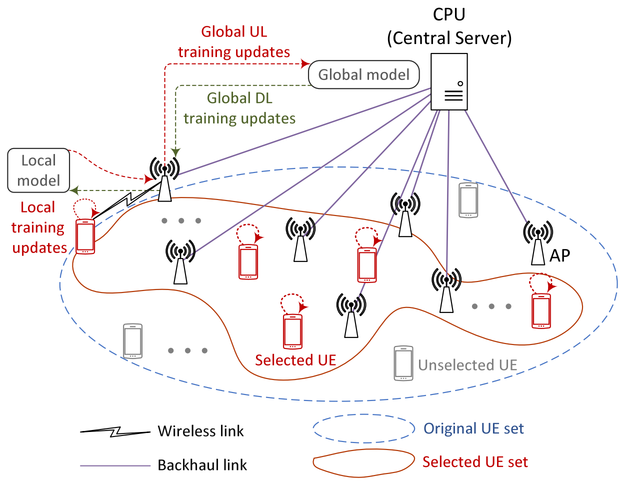

Fig. 1. CFmMIMO network model for wireless FL.

vectors with length d. $\boldsymbol{X}^*$ and $\boldsymbol{X}^H$ are the conjugate and conjugate transposition of a matrix $\boldsymbol{X}$, respectively. $\mathcal{CN}(\mathbf{0}, \boldsymbol{Q})$ denotes the circularly symmetric complex Gaussian distribution with zero mean and covariance $\boldsymbol{Q}$. $\mathcal{N}(0, V)$ denotes the normal distribution with zero mean and variance V. ∇g is the gradient of a function g. $\mathbb{E}\{x\}$ denotes the expected value of a random variable x. For ease of reference, the symbols frequently used throughout this article are listed in Table I.

The remainder of this article is organized as follows. Section II introduces a CFmMIMO system model to support a standard FL framework. Section III formulates the optimization problem of minimizing the FL execution time, whereas Section IV proposes a new algorithm to solve this problem. Section V verifies the performance of the developed algorithm through numerical examples. Finally, Section VI concludes this article.

II. Cell-Free Massive MIMO System Model to Support Wireless Federated Learning

We consider the CFmMIMO network model illustrated in Fig. 1. Let $\mathcal{N} \triangleq \{1, \dots, N\}$ be the UE set in the network. The UEs are served by a set of APs $\mathcal{M} = \{1, \dots, M\}$ via wireless access links with the same time-frequency resource [33], [34]. The transmission between the APs and UEs is under the time-division duplexing operation with channel reciprocity.[1] The APs and UEs are each equipped with a single antenna. These APs are assumed to connect to a CPU via backhaul links with high capacity. As such, the transmission times between the CPU and all the APs are considered negligible [24]. The CPU and UEs act as the central server and the clients in an FL process, respectively. Here, the APs relay the model updates between the CPU and the UEs.

A. UE Selection Model and Standard Federated Learning Framework

First, we propose to select a subset of UEs out of N original UEs to participate in an FL process. Let variable a_k indicate whether a UE $k \in \mathcal{N} \triangleq \{1, \dots, N\}$ is selected to partake in an FL process, i.e.,

$$a_k \triangleq \begin{cases} 1, & \text{if UE } k \text{ is selected} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Let $\widetilde{\mathcal{N}}$ be the set of selected UEs. Denote by $\widetilde{N} \triangleq \sum_{k\in\mathcal{N}} a_k \leq N$ the cardinality of $\widetilde{\mathcal{N}}$ or the number of selected UEs, and by $\boldsymbol{a} \triangleq [a_1, \dots, a_N]^T$ the vector of UE selection. We insist that

$$\sum_{k\in\mathcal{N}} a_k \geq N_{\text{QoL}} \tag{2}$$

where N_{QoL} is a predefined minimum number of participating UEs, chosen such that an acceptable Quality of Learning (QoL) is achieved. As noted in Section I, how to determine the exact value of N_{QoL} is out of the scope of this article.

Next, the above $\widetilde{N}$ selected UEs will participate in an FL process that adopts the standard FL framework with a synchronous aggregation mode[2] [29], [30], [39]–[42]. In general, it is an iterative process consisting of multiple communication rounds. Each communication round involves the following basic steps (S1)–(S4).

(S1) A central server sends the global update to all the selected UEs.

(S2) Upon receiving the global update from the central server, the UEs solve their local learning problems (based on their local data set) and compute their local model updates.

(S3) The UEs send their local updates to the central server.

(S4) Upon receiving all the local updates from UEs, the central server computes the global update by aggregation.

The communication rounds repeat until a certain level of test accuracy is attained at the server. At which point, the server terminates the whole FL process.

B. CFmMIMO System Model for Wireless Federated Learning

Now, we will describe in detail how a communication round outlined in Section II-A is realized by CFmMIMO.

Assumption (A1): All selected UEs execute steps (S2) and (S3) in a synchronized manner. Within a step (S2) or (S3), if a UE completes its operation before other UEs, it must wait until all other UEs complete their respective operation before proceeding to the next step.

[1] In general, channel reciprocity error can be a performance-limiting factor in wireless networks using the time-division duplexing operation. It is because the hardware chains in the transmitter and receivers may not be reciprocal between the uplink and downlink. However, according to [35], the calibration of hardware chains for channel reciprocity is not a serious problem in *massive MIMO systems*. For example, there are calibration solutions that have been tested successfully for practical 64-antenna systems [36], [37].

[2] FL algorithms with the synchronous aggregation mode wait to receive all local model updates sent from UEs before aggregation, while the FL algorithms with an asynchronous aggregation mode do not. The FL algorithms with synchronous aggregation normally offer a better convergence rate and accuracy than the FL algorithms operating with asynchronous aggregation. FL algorithms with synchronous aggregation are well studied [38], while research on the improvement of learning performance of the FL algorithms with asynchronous aggregation is still in its infancy. As such, this article focuses on resource allocation for supporting FL with synchronous aggregation [29], [30], [39]–[42].

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

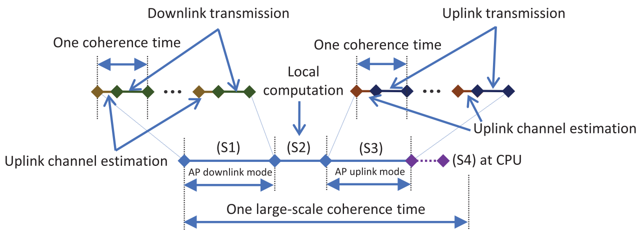

Fig. 2. Details of operation of one FL communication round in the considered CFmMIMO network.

Remark 1: It should be noted that synchronizing the operations of UEs at steps (S2) and (S3) would cause some waste of radio resources because of the wait time among the UEs. Instead, an asynchronous design may potentially reduce the execution time by removing the wait time. However, in the asynchronous design, some UEs may finish step (S2) while other UEs have not even completed step (S1) yet. Therefore, it is difficult for the APs to know exactly when they need to switch to the uplink mode to receive local model updates from the UEs. The solutions for overcoming this difficulty may make the designs of signal processing and signalling much more complex. Based on this observation, this work focuses on the proposed synchronous communication design for ease of implementation and leaves the asynchronous design for future work.

Assumption (A2): Each FL communication round is completely executed within a large-scale coherence time of the wireless channel [21].[3]

Remark 2: Assumption 2 is realistic in many practical scenarios, such as outdoor systems with moderately low user mobility (e.g., dense urban areas) and indoor systems (e.g., office buildings, stadiums, cinema theaters, and factory). An example is a system that supports users' speed of $v \leq 7.5$ m/s $=$ 27 km/h. With a conventional carrier frequency $f_c = 1$ GHz [34], the coherence time of the channels is $T_c \geq (c_\ell/4f_c v) = 10$ ms, where $c_\ell = 3 \times 10^8$ m/s is the speed of light. According to the measurements in [43], the large-scale coefficients can stay relatively unchanged for at least 100 times longer than the coherence time. Therefore, the large-scale coherence time is $\widetilde{T}_c \geq 100T_c \geq 1$ s. Let $S = 4$ KB = 32 Kb be the size of global and local updates [42]. For a 5G system with a user-experienced downlink rate of $R_d = 100$ Mb/s and uplink rate of $R_u = 50$ Mb/s [44], the time durations for downlink and uplink transmissions are $T_d = (S/R_d) = 0.32$ ms and $T_u = (S/R_u) = 0.64$ ms, respectively. Now, let $D_s = 10^6$ be the number of data samples at each UE, $c = 20$ cycles/sample be the number of CPU cycles required to process one data sample, $f = 3 \times 10^9$ cycles/s be the frequency of processing CPU cycles, and $N_L = 5$ be the number of iterations of processing the local data at UEs [42]. Then, the time duration needed to compute the local update is $T_c = N_L(cD_s/f) = 33.3$ ms. Therefore, the execution time of one FL communication round is $T_o = T_d + T_c + T_u = 34.26$ ms $<< 1$ s $\leq \widetilde{T}_c$. Note that the number of collected data samples at UEs in FL applications is normally small (e.g., $10^3 - 10^4$) [45]. Therefore, the time of computing local updates should be much shorter than that in our example. This means in practice, the executive time of one FL communication round can even be much smaller than the large-scale coherence time.

On the other hand, Assumption 2 does not impose any practical challenge for deploying cell-free massive MIMO in indoor scenarios. The issues of deployment complexity and high fronthaul capacity of cell-free massive MIMO in indoor scenarios can be effectively resolved by using an appropriate architecture, namely, *radio stripe system* [46]–[48]. In this system, each antenna element is effectively an AP. Then, a large number of small-size antennas are put in a cable or a radio stripe. Inside the radio stripe, antennas are connected to their associated antenna processing units (APUs). The APUs and antennas are power supplied via a shared bus that is serially located in the radio stripe. Finally, each radio stripe is connected to one or multiple CPUs. The radio stripe system turns the long fronthaul cables into plug-and-play radio stripes, which effectively and flexibly implement the star topology and improve the coverage of cell-free massive MIMO. The required fronthaul capacity of each radio stripe is proportional to the sum rate of data streams for transmitting to or receiving from the users at the maximum network load. Since there is a large number of antennas in the radio stripe system, the required fronthaul capacity can be reduced by reducing the number of users served per radio stripe with a proper design of UE-radio stripe association [46].

With the above assumptions, we are ready to implement steps (S1)–(S4) by CFmMIMO, as illustrated in Fig. 2.

1) Step (S1): This step has the following two phases, both executed within a coherence block.

Uplink Channel Estimation: Uplink pilot sequences are sent by all the UEs to all the APs simultaneously in each small-scale coherence time to help estimate channels. These estimates will later be used to construct downlink signal beams. Here, we assume time-division duplexing where both uplink and downlink channel information are acquired at the APs via uplink channel estimation [24]. Denote by τ_c the number of samples of each coherence block, and by τ_t (samples) the length of one pilot sequence. Let $\sqrt{\tau_t}\boldsymbol{\varphi}_k \in \mathbb{C}^{\tau_t \times 1}$ be the pilot sequence transmitted from a UE $k \in \mathcal{N}$, where $\|\boldsymbol{\varphi}_k\|^2 = 1\ \forall k \in \mathcal{N}$. Denote by $g_{mk} = (\beta_{mk})^{1/2}\tilde{g}_{mk}$ the channel from a UE k to an AP m, where β_{mk} and $\tilde{g}_{mk} \sim \mathcal{CN}(0,1)$ are the large-scale fading and small-scale fading channel coefficients, respectively. At the AP m, g_{mk} is estimated by using the received pilots and the minimum mean-square error (MMSE) estimation. The MMSE estimate $\hat{g}_{mk}$ of g_{mk} is a random variable distributed according to $\mathcal{CN}(0, \hat{\sigma}_{mk}^2)$, where $\hat{\sigma}_{mk}^2 = ([\tau_t\rho_t(\beta_{mk})^2]/[\sum_{\ell\in\mathcal{N}} \tau_t\rho_t\beta_{m\ell}|\boldsymbol{\varphi}_k^H\boldsymbol{\varphi}_\ell|^2 + 1])$ [24].

Downlink Transmission: Denote by S_d (bits) the data size of the (same) global downlink model update for each selected UE. Since the global downlink model update can include thousands of model weights, S_d is typically large. At the CPU, the global model update intended for UE k is encoded into many symbols $s_{d,k,i} \sim \mathcal{CN}(0,1)\ \forall i \in \{1,\ldots,L_k\}$. The number of symbols L_k depends on the data size of the model update as well as the data rate. For ease of presentation, hereafter, we drop the index i from $s_{d,k,i}$ and let $R_{d,k}$ be the data rate of the global update intended for a UE $k \in \mathcal{N}$.

[3] A large-scale coherence time is the period of time where the large-scale fading coefficients are reasonably invariant [21].

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

To transmit the symbols received from the CPU, the APs first use conjugate beamforming to precode these symbols before broadcasting them to all the selected UEs. Specifically, the transmitted signal at an AP m is given as $x_{d,m} = \sqrt{\rho_d}\sum_{k\in\mathcal{N}}\sqrt{\eta_{mk}}(\hat{g}_{mk})^* s_{d,k}$, where ρ_d is the maximum normalized transmit power at each AP and $\eta_{mk}\ \forall m\in\mathcal{M}, k\in\mathcal{N}$, is a power control coefficient. The transmitted power at the AP m is required to meet the average normalized power constraint, i.e., $\mathbb{E}\{|x_{d,m}|^2\}\le\rho_d$, which can also be expressed as the following per-AP power constraint:

$$\sum_{k\in\mathcal{N}}\sigma^2_{mk}\eta_{mk}\le 1 \quad \forall m. \tag{3}$$

Since no power should be allocated to the nonselected UEs, we have

$$\forall k\in\mathcal{N} : \text{if } a_k = 0, \ \text{then} \quad \forall m\in\mathcal{M}, \eta_{mk} = 0. \tag{4}$$

The achievable downlink data rate at a UE k is given in (5), shown at the bottom of the page [24, eq. (24)], where $\boldsymbol{\eta}\triangleq\{\eta_{mk}\}_{m\in\mathcal{M},k\in\mathcal{N}}$, and B is the bandwidth. The numerator in the log function represents the power of the desired signals normalized by noise. The three terms in the denominator in the log function represent the normalized-by-noise power of the pilot contamination, inter-UE interference, and noise, respectively.

The transmission time from the APs to a UE k is given by

$$t_{d,k}(a_k,\boldsymbol{\eta}) = \frac{a_k S_d}{R_{d,k}(\boldsymbol{\eta})} \tag{6}$$

where $a_k\neq 0$. Here, (6) captures the fact that the nonselected UEs do not receive any intended downlink transmission, i.e., $t_{d,k}=0$, for any $k\in\mathcal{N}$ with $a_k = 0$.

2) Step (S2): After receiving the global update, each UE k computes its local model update by using its local data set.

Computation Delay: Let c_k (cycles/sample) be the number of processing cycles required to process one data sample at a UE k. Assume that c_k is known *a priori* by an offline measurement [49]. Denote by D_k (samples) and f_k (cycles/s) the size of the local data set and the processing frequency of the UE k, respectively. The total computing time at the UE k is expressed as [42]

$$t_{c,k} = \frac{a_k L D_k c_k}{f_k} \tag{7}$$

where $a_k\neq 0$, L is the number of local computing iterations, and $(D_k c_k/f_k)$ is the computing time of each iteration over the local training data set at the UE k. Similarly, because the nonselected UEs do not compute their local models, we have

$$\forall k\in\mathcal{N} : a_k = 0, t_{c,k} = 0. \tag{8}$$

3) Step (S3): This step has the following two phases, both executed within a coherence block.

Uplink Channel Estimation: This phase uses the same pilot assignment and channel estimation techniques in the uplink channel estimation of step (S1). The MMSE estimate $\bar{g}_{mk}$ of g_{mk} is thus a random variable distributed according to $\mathcal{CN}(0,\bar{\sigma}^2_{mk})$, where $\bar{\sigma}^2_{mk} = ([\tau_t\rho_t(\beta_{mk})^2]/[\sum_{\ell\in\mathcal{N}}\tau_t\rho_t\beta_{m\ell}|\boldsymbol{\varphi}_k^H\boldsymbol{\varphi}_\ell|^2+1])$.

Uplink Transmission: Similar to step (S1), for ease of presentation, a symbol $s_{u,k}\sim\mathcal{CN}(0,1)$ is encoded for the local model update of a UE k. The symbol $s_{u,k}$ is then allocated a transmit amplitude value $\sqrt{\rho_u\zeta_k}$ to generate a baseband signal $x_{u,k}$ for wireless transmission, i.e., $x_{u,k} = \sqrt{\rho_u\zeta_k}s_{u,k}$. The UE k is subjected to the average transmit power constraint, i.e., $\mathbb{E}\{|x_{u,k}|^2\}\le\rho_u$, which can also be expressed as a per-UE constraint

$$0\le\zeta_k\le 1 \quad \forall k\in\mathcal{N}. \tag{9}$$

Because no power should be allocated to the nonselected UEs, we have

$$\forall k\in\mathcal{N} : \text{if } a_k = 0, \ \text{then } \zeta_k = 0. \tag{10}$$

The achievable uplink data rate $R_{u,k}$ at the CPU for the UE k is then given in (11), shown at the bottom of the next page, where $\boldsymbol{\zeta}\triangleq\{\zeta_k\}_{k\in\mathcal{N}}$ [24, eq. (27)].

Let S_u (bits) and $R_{u,k}$ (bps) be the same data size of the local model updates and the data rate of transmitting the local model update from a UE k to the CPU, respectively. The transmission time from the UE k to the APs is given by

$$t_{u,k}(a_k,\boldsymbol{\zeta}) = \frac{a_k S_u}{R_{u,k}(\boldsymbol{\zeta})} \tag{12}$$

where $a_k\neq 0$. Since the nonselected UEs do not have any uplink transmission, we have $\forall k\in\mathcal{N} : a_k = 0, t_{u,k} = 0$.

4) Step (S4): The CPU computes the global update by using all the received local updates. Since the CPU's computational capability is much higher than that of the UEs, the time required to compute the global update is assumed negligible.

III. Joint UE Selection and Resource Allocation for Minimizing Total FL Execution Time in CFmMIMO: Optimization Problem Formulation

A. Total Execution Time of FL Process

The execution time of one FL communication round comprises the execution times of steps (S1)–(S3) [as step (S4) is assumed not to incur any time delay]. The execution time of each of these steps is the longest execution time within that

$$R_{d,k}(\boldsymbol{\eta}) = \frac{\tau_c-\tau_t}{\tau_c}B\log_2\left(1+\frac{\rho_d\left(\sum_{m\in\mathcal{M}}\eta_{mk}^{1/2}\hat{\sigma}^2_{mk}\right)^2}{\underbrace{\rho_d\sum_{\ell\in\mathcal{N}\setminus k}\left(\sum_{m\in\mathcal{M}}\eta_{m\ell}^{1/2}\hat{\sigma}^2_{m\ell}\frac{\beta_{mk}}{\beta_{m\ell}}\right)^2|\boldsymbol{\varphi}_\ell^H\boldsymbol{\varphi}_k|^2}_{\text{Pilot contamination}}+\underbrace{\rho_d\sum_{\ell\in\mathcal{N}}\sum_{m\in\mathcal{M}}\eta_{m\ell}\hat{\sigma}^2_{m\ell}\beta_{mk}}_{\text{Inter-user interference}}+1}\right) \tag{5}$$

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

step. As such, the execution time of one FL communication round is expressed as follows:

$$\begin{aligned} &T_o(\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta}) \\ &\triangleq \max_k t_{d,k}(a_k, \boldsymbol{\eta}) + \max_k t_{c,k}(a_k, f_k) + \max_k t_{u,k}(a_k, \boldsymbol{\zeta}) \\ &\triangleq \max_k \frac{a_k S_d}{R_{d,k}(\boldsymbol{\eta})} + \max_k \frac{a_k L D_k c_k}{f_k} + \max_k \frac{a_k S_u}{R_{u,k}(\boldsymbol{\zeta})} \end{aligned} \tag{13}$$

where $k \in \{k \in \mathcal{N} | \ a_k \neq 0\}$. Note that $\boldsymbol{a}$ (UE selection) is determined before an FL process is executed, while $\boldsymbol{\eta}$, $\boldsymbol{f}$, and $\boldsymbol{\zeta}$ are optimized in each FL communication round. The mathematical expression of the execution time of one FL communication round must have a two-timescale structure. Therefore, we introduce a new metric named "effective execution time of one FL communication round" $\mathbb{E}\{T_o(\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta})\}$, which is the average of $T_o(\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta})$ over large-scale fading realizations. The values of $\boldsymbol{a}$ in $\mathbb{E}\{T_o(\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta})\}$ remains unchanged, while those of $\boldsymbol{\eta}, \boldsymbol{f}$, and $\boldsymbol{\zeta}$ in $T_o(\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta})$ are optimized in each large-scale coherence time.

Let $G(\boldsymbol{a})$ be the number of communication rounds in an FL process. By using the tightest convergence rates of FL [30, Th. I], we have

$$G(\boldsymbol{a}) = \frac{q}{\sum_{k \in \mathcal{N}} a_k}. \tag{14}$$

Here, q is a known constant which depends on the specific characteristics of the FL learning problems. The total execution time of an FL process is then given by

$$T_e(\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta}) \triangleq G(\boldsymbol{a}) \mathbb{E}\{T_o(\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta})\}. \tag{15}$$

Equations (13)–(15) clearly shows the effects of UE selection on the overall execution time. According to (13), if we only allow users with favorable link conditions (i.e., strong channel gains, weak pilot contamination, and interference) to partake in the FL process, the execution time of one FL communication round is shorter. However, selecting only a subset of all UEs also means increasing the required number of communication rounds as shown in (14). Therefore, minimizing the total execution time in (15) involves finding an optimal set of UEs to be selected, in order to balance between the two contradicting effects.

B. Problem Formulation

We are now ready to formulate the main design problem as the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta}} \quad & T_e(\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta}) && \text{(16a)} \\ \text{s.t.} \quad & (1)-(4), \ (9), (10) \\ & 0 \leq \eta_{mk} \quad \forall m, k && \text{(16b)} \\ & 0 \leq \zeta_k \quad \forall k && \text{(16c)} \\ & 0 \leq f_k \leq f_{k,\max} \quad \forall k. && \text{(16d)} \end{aligned}$$

Problem (16) has a nonconvex stochastic, mixed-integer mixed-timescale structure, along with binary constraints and tight coupling among the optimizing variables. Finding its globally optimal solution is challenging.

IV. Joint UE Selection and Resource Allocation for Minimizing Total FL Execution Time in CFmMIMO: Proposed Algorithm

First, to deal with the binary constraint (1), we observe that $x \in \{0, 1\} \Leftrightarrow x \in [0, 1]$ and $x - x^2 \leq 0$ [50], [51]. Therefore, (1) is equivalent to the following two constraints:

$$\sum_{k \in \mathcal{N}} \left(a_k - a_k^2\right) \leq 0 \tag{17}$$

$$0 \leq a_k \leq 1 \quad \forall k. \tag{18}$$

Since the values of $\{a_k\}$ in (17) and (18) are now real, it is easier to handle (17) and (18) than (1). We then use (17) and (18) to rewrite problem (16) as follows:

$$\begin{aligned} \min_{\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta}} \quad & T_e(\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta}) \\ \text{s.t.} \quad & (2)-(4), \ (9), \ (10), (16\text{b})-(16\text{d}), \ (17), \ (18). \end{aligned} \tag{19}$$

Next, to deal with the max functions in $T_o(\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta})$, we rewrite problem (19) in a more tractable epigraph form as follows:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & G(\boldsymbol{a}) \mathbb{E}\{\widetilde{T}_o(t_d, t_c, t_u)\} && \text{(20a)} \\ \text{s.t.} \quad & (2)-(4), \ (9), (10), \ (16\text{b})-(16\text{d}), \ (17), \ (18) \\ & 0 \leq r_{d,k}, 0 \leq r_{u,k} \quad \forall k && \text{(20b)} \\ & r_{d,k} \leq R_{d,k}(\boldsymbol{\eta}) \quad \forall k && \text{(20c)} \\ & r_{u,k} \leq R_{u,k}(\boldsymbol{\zeta}) \quad \forall k && \text{(20d)} \\ & \frac{a_k S_d}{r_{d,k}} \leq t_d \quad \forall k, a_k \neq 0 && \text{(20e)} \\ & \frac{a_k L D_k c_k}{f_k} \leq t_c \quad \forall k, a_k \neq 0 && \text{(20f)} \\ & \frac{a_k S_u}{r_{u,k}} \leq t_u \quad \forall k, a_k \neq 0 && \text{(20g)} \end{aligned}$$

where $\widetilde{T}_o(t_d, t_c, t_u) = t_d + t_c + t_u$, $\boldsymbol{x} \triangleq \{\boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta}, \boldsymbol{r}_d, \boldsymbol{r}_u, t_d, t_c, t_u\}$; $\boldsymbol{r}_d \triangleq \{r_{d,k}\}$, $\boldsymbol{r}_u \triangleq \{r_{u,k}\}$ $\forall k \in \mathcal{N}$, t_d, t_c, and t_u are additional variables.

Our main problem (16) is now transformed to problem (20) which is a stochastic nonconvex optimization problem. In (20), the UE selection variables $\boldsymbol{a}$ are optimized in a long-term timescale (before any FL process happens), while $\widetilde{\boldsymbol{x}} \triangleq \boldsymbol{x} \backslash \boldsymbol{a}$,

$$R_{u,k}(\boldsymbol{\zeta}) = \frac{\tau_c - \tau_t}{\tau_c} B \log_2 \left(1 + \frac{\rho_u \zeta_k \left(\sum_{m \in \mathcal{M}} \bar{\sigma}_{mk}^2\right)^2}{\underbrace{\rho_u \sum_{\ell \in \mathcal{N} \backslash k} \zeta_\ell \left(\sum_{m \in \mathcal{M}} \bar{\sigma}_{mk}^2 \frac{\beta_{m\ell}}{\beta_{mk}} \right)^2 |\boldsymbol{\varphi}_k^H \boldsymbol{\varphi}_\ell|^2}_{\text{Pilot contamination}} + \underbrace{\rho_u \sum_{\ell \in \mathcal{N}} \zeta_\ell \sum_{m \in \mathcal{M}} \bar{\sigma}_{mk}^2 \beta_{m\ell}}_{\text{Inter-user interference}} + \sum_{m \in \mathcal{M}} \bar{\sigma}_{mk}^2} \right) \tag{11}$$

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

including transmit power and processing frequency are optimized in a short-term timescale (in each FL communication round). To solve this type of optimization problem, we adopt the general framework of [32]. Specifically, we decompose problem (20) into a short-term subproblem and a long-term master problem and solve these resulting problems in an alternating manner. The mathematical derivations are detailed in the following.

For a given $\boldsymbol{a}$, in each large-scale coherence time, the *short-term subproblem* is expressed as follows:

$$\begin{aligned} &\min_{\widetilde{\boldsymbol{x}}} \ \widetilde{T}_o(t_d, t_c, t_u) \\ &\text{s.t.} \ (3),\ (4),\ (9),\ (10),\ (16\text{b})-(16\text{d}),\ (20\text{b})-(20\text{g}) \end{aligned} \tag{21}$$

For given optimal solutions $\widetilde{\boldsymbol{x}}$ to problems (21), we have $t_d = (a_{k^*} S_d / R_{d,k^*})$, where $k^* \triangleq \text{argmax}_{k\in\mathcal{N}}(a_k S_d / R_{d,k})$. Therefore, we can obtain $t_d = \boldsymbol{a}^T \tilde{\boldsymbol{t}}_d$, where $\tilde{\boldsymbol{t}}_d$ is the vector whose elements are 0 except for the k^*th element, and the value of this element is $(S_d / R_{d,k^*})$. Similarly, we have $t_c^* = ([a_{i^*} L D_{i^*} c_{i^*}]/[f_{i^*}])$ and $t_u = (a_{j^*} S_u / R_{u,j^*})$, where $i^* \triangleq \text{argmax}_{k\in\mathcal{N}}(a_k L D_k c_k / f_k)$ and $j^* \triangleq \text{argmax}_{k\in\mathcal{N}}(a_k S_u / R_{u,k})$. Then, we also have $t_c = \boldsymbol{a}^T \tilde{\boldsymbol{t}}_c$ and $t_u = \boldsymbol{a}^T \tilde{\boldsymbol{t}}_u$. Here, $\tilde{\boldsymbol{t}}_c$ is the vector whose elements are 0 except for the i^*th element, and the value of this element is $(L D_{i^*} c_{i^*} / f_{i^*})$. $\tilde{\boldsymbol{t}}_u$ is the vector whose elements are 0 except for the j^*th element, and the value of this element is $(S_u / w_{j^*} R_{u,j^*})$. Then, the *long-term master problem* is expressed as follows:

$$\begin{aligned} &\min_{\boldsymbol{a}} \ g(\boldsymbol{a}) \\ &\text{s.t.} \ (2),\ (17),\ (18) \end{aligned} \tag{22}$$

where

$$\begin{aligned} g(\boldsymbol{a}) &\triangleq T_e(\boldsymbol{a}) = \mathbb{E}\{T(\boldsymbol{a})\} \\ T(\boldsymbol{a}) &\triangleq G(\boldsymbol{a}) \widetilde{T}_o(\boldsymbol{a}) = \frac{q(\boldsymbol{a}^T \tilde{\boldsymbol{t}}_d + \boldsymbol{a}^T \tilde{\boldsymbol{t}}_c + \boldsymbol{a}^T \tilde{\boldsymbol{t}}_u)}{\boldsymbol{a}^T \boldsymbol{1}} \\ \widetilde{T}_o(\boldsymbol{a}) &= \boldsymbol{a}^T \tilde{\boldsymbol{t}}_d + \boldsymbol{a}^T \tilde{\boldsymbol{t}}_c + \boldsymbol{a}^T \tilde{\boldsymbol{t}}_u \\ G(\boldsymbol{a}) &= \frac{q}{\boldsymbol{a}^T \boldsymbol{1}} \end{aligned}$$

and $\boldsymbol{1} \in \mathbb{R}^N$ is an all-one vector.

A. Solving Short-Term Subproblem (21)

First, we rewrite problem (21) as follows:

$$\begin{aligned} &\min_{\widehat{\boldsymbol{x}}} \ \widetilde{T}_o(t_d, t_c, t_u) && (23\text{a}) \\ &\text{s.t.} \ (16\text{b})-(16\text{d}),\ (20\text{b})-(20\text{g}) \\ &\qquad \sigma_{mk}^2 \eta_{mk} \le \tilde{v}_{mk} \quad \forall m, k && (23\text{b}) \\ &\qquad \tilde{v}_{mk} \le a_k \quad \forall m, k && (23\text{c}) \\ &\qquad \sum_{k\in\mathcal{N}} \tilde{v}_{mk} \le 1 \quad \forall m && (23\text{d}) \\ &\qquad \zeta_k \le a_k \quad \forall k && (23\text{e}) \end{aligned}$$

where $\widehat{\boldsymbol{x}} \triangleq \{\widetilde{\boldsymbol{x}}, \tilde{\boldsymbol{v}}\}$ and $\tilde{\boldsymbol{v}} \triangleq \{\tilde{v}_{mk}\}_{m\in\mathcal{M}, k\in\mathcal{N}}$ are additional variables. Here, (23b)–(23d) follow from (3) and (4), while (23e) follows from (9) and (10). Problem (23) is still challenging because of the nonconvex constraints (20c) and (20d). To deal with these constraints, we let $\boldsymbol{v} \triangleq \{v_{mk}\}_{m\in\mathcal{M}, k\in\mathcal{N}}$ and $\boldsymbol{u} \triangleq \{u_k\}_{k\in\mathcal{N}}$ with

$$v_{mk} \triangleq \eta_{mk}^{1/2} \quad \forall m, k \tag{24}$$

$$u_k \triangleq \zeta_k^{1/2} \quad \forall k \tag{25}$$

and rewrite (23) as follows:

$$\begin{aligned} &\min_{\bar{\boldsymbol{x}}} \ \widetilde{T}_o(t_d, t_c, t_u) && (26\text{a}) \\ &\text{s.t.} \ (16\text{d}),\ (20\text{b}),\ (20\text{e})-(20\text{g}),\ (23\text{c}),\ (23\text{d}) \\ &\qquad \sigma_{mk}^2 v_{mk}^2 \le \tilde{v}_{mk} \quad \forall m, k && (26\text{b}) \\ &\qquad 0 \le v_{mk} \quad \forall m, k && (26\text{c}) \\ &\qquad u_k^2 \le a_k \quad \forall m, k && (26\text{d}) \\ &\qquad 0 \le u_k \le 1 \quad \forall k && (26\text{e}) \\ &\qquad 0 \le r_{d,k} \le R_{d,k}(\boldsymbol{v}) \quad \forall k && (26\text{f}) \\ &\qquad 0 \le r_{u,k} \le R_{u,k}(\boldsymbol{u}) \quad \forall k && (26\text{g}) \end{aligned}$$

where $\bar{\boldsymbol{x}} \triangleq \{\widehat{\boldsymbol{x}}, \boldsymbol{v}, \boldsymbol{u}\} \backslash \{\boldsymbol{\eta}, \boldsymbol{\zeta}\}$. Here, (26b)–(26e) follow (16b), (16c), (23b), (23e), (24), and (25), whereas (26f) and (26g) follow (20c) and (20d).

Regarding the nonconvex constraints (26f) and (26g), the concave lower bound $\widetilde{R}_{d,k}(\boldsymbol{v})$ of $R_{d,k}(\boldsymbol{v})$ is given by (27), shown at the bottom of the next page [52], where

$$\Upsilon_k(\{v_{mk}\}_{m\in\mathcal{M}}) = \sqrt{\rho_d} \sum_{m\in\mathcal{M}} v_{mk} \sigma_{mk}^2$$

$$\begin{aligned} \Pi_k(\boldsymbol{v}) = \rho_d &\sum_{\ell\in\mathcal{N}\backslash k} \left(\sum_{m\in\mathcal{M}} v_{m\ell} \sigma_{m\ell}^2 \frac{\beta_{mk}}{\beta_{m\ell}} \right)^2 |\boldsymbol{\varphi}_\ell^H \boldsymbol{\varphi}_k|^2 \\ &+ \rho_d \sum_{\ell\in\mathcal{N}} \sum_{m\in\mathcal{M}} v_{m\ell}^2 \sigma_{m\ell}^2 \beta_{mk} + 1. \end{aligned}$$

Similarly, the concave lower bound $\widetilde{R}_{u,k}(\boldsymbol{u})$ of $R_{u,k}(\boldsymbol{u})$ is given by (28), shown at the bottom of the next page [52], where

$$\Psi_k(u_k) = \sqrt{\rho_u} u_k \left(\sum_{m\in\mathcal{M}} \sigma_{mk}^2 \right)$$

$$\begin{aligned} \Xi_k(\boldsymbol{u}) = \rho_u &\sum_{\ell\in\mathcal{N}\backslash k} u_\ell^2 \left(\sum_{m\in\mathcal{M}} \sigma_{mk}^2 \frac{\beta_{m\ell}}{\beta_{mk}} \right)^2 |\boldsymbol{\varphi}_k^H \boldsymbol{\varphi}_\ell|^2 \\ &+ \rho_u \sum_{\ell\in\mathcal{N}} u_\ell^2 \sum_{m\in\mathcal{M}} \sigma_{mk}^2 \beta_{m\ell} + \sum_{m\in\mathcal{M}} \sigma_{mk}^2. \end{aligned}$$

As such, (26f) and (26g) can be approximated by the following convex constraints:

$$r_{d,k} \le \widetilde{R}_{d,k}(\boldsymbol{v}) \quad \forall k \in \mathcal{N} \tag{29}$$

$$r_{u,k} \le \widetilde{R}_{u,k}(\boldsymbol{u}) \quad \forall k \in \mathcal{N}. \tag{30}$$

At iteration $(\kappa + 1)$, for a given point $\bar{\boldsymbol{x}}^{(\kappa)}$, problem (26) [hence (21)] can finally be approximated by the following convex problem:

$$\min_{\bar{\boldsymbol{x}}\in\widetilde{\mathcal{F}}} \widetilde{T}_o(t_d, t_c, t_u) \tag{31}$$

where $\widetilde{\mathcal{F}} \triangleq \{(16\text{d}), (20\text{b}), (20\text{e})-(20\text{g}), (23\text{c}), (23\text{d}), (26\text{c})-(26\text{e}), (29), (30)\}$ is a convex feasible set. In Algorithm 1, we outline the main steps to solve problem (21).

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Successive Convex Approximation Approach for Solving the Short-Term Subproblem (21)

```
1: Initialization: Set $\kappa = 0$ and choose a random point $\bar{\boldsymbol{x}}^{(0)} \in \mathcal{F}$.
2: repeat
3:     Update $\kappa = \kappa + 1$ Solve the approximated problem (31) of (21) using CVX to obtain its optimal solution $\bar{\boldsymbol{x}}^*$ and update $\bar{\boldsymbol{x}}^{(\kappa)} = \bar{\boldsymbol{x}}^*$
4: until convergence
```

Let $\mathcal{F} \triangleq \{(16d), (20b), (20e) - (20g), (23c), (23d), (26b) - (26g)\}$ be the feasible set of (26). Starting from a random point $\bar{\boldsymbol{x}} \in \mathcal{F}$, we solve (31) using CVX [53] to obtain its optimal solution $\bar{\boldsymbol{x}}^*$. This solution is then used as an initial point in the next iteration. The algorithm terminates when an accuracy level of ε is reached.

Proposition 1: Algorithm 1 converges to a Karush–Kuhn–Tucker (KKT) solution of (21).

Proof: It is true that $\widetilde{R}_{d,k}(\boldsymbol{v})$ and $\widetilde{R}_{u,k}(\boldsymbol{u})$ satisfy the key properties of general inner approximation functions [54, Properties (i)–(iii)]. The feasible set $\widetilde{\mathcal{F}}$ also satisfies Slater's constraint qualification condition for convex programs. Therefore, Algorithm 1 converges to a KKT solution of (26) when starting from a point $\widetilde{\boldsymbol{x}}^{(0)} \in \mathcal{F}$ [54, Th. 1]. By using the variable transformations (24) and (25), it can be seen that the KKT solutions of (26) satisfy the KKT conditions of (23) as well as of (21). ■

B. Solving the Long-Term Master Problem (22)

Let $V(\boldsymbol{a}) \triangleq \sum_{k\in\mathcal{N}}(a_k - a_k^2) = \boldsymbol{a}^T(\boldsymbol{1} - \boldsymbol{a})$, then (17) becomes $V(\boldsymbol{a}) \leq 0$. To deal with this nonconvex constraint, we consider the problem

$$\min_{\boldsymbol{a}} \ \mathcal{L}(\boldsymbol{a}) \triangleq g(\boldsymbol{a}) + \lambda V(\boldsymbol{a}) \quad \text{s.t.} \quad (2),\ (18) \tag{32}$$

where $\mathcal{L}(\boldsymbol{a})$ is the Lagrangian of (22), and $\lambda \geq 0$ is the Lagrangian multiplier corresponding to (17). Let $\mathcal{H} \triangleq \{(2), (18)\}$ be the feasible set of problem (32).

Proposition 2: The following statement holds.

1) The value V_λ of V at the solution of (32) corresponding to λ is decreasing to 0 as $\lambda \to +\infty$.
2) Problem (32) has the following property, i.e.,

$$\min_{\boldsymbol{a}\in\mathcal{H}} g(\boldsymbol{a}) = \sup_{\lambda\geq 0} \min_{\boldsymbol{a}\in\widehat{\mathcal{H}}} \mathcal{L}(\boldsymbol{a}, \lambda) \tag{33}$$

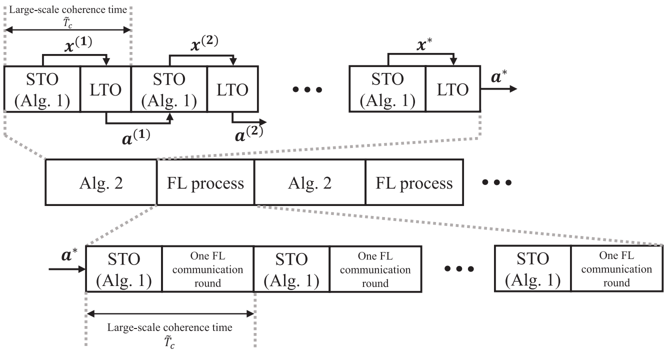

Fig. 3. Overview of Algorithm 2's operation to assist CFmMIMO-based wireless FL.

and is therefore equivalent to (22) at the optimal solution $\lambda^* \geq 0$ of the sup-min problem in (33).

Proof: The proof follows from [51, Proposition 1] and [50, Proposition 1]; hence, omitted for brevity. ■

Theoretically, it is required to have $V_\lambda = 0$ in order to obtain an optimal λ^*. According to Proposition 2, V_λ decreases to 0 as $\lambda \to +\infty$. Since there is always a numerical tolerance in computation, it is sufficient to accept $V_\lambda < \varepsilon$ for some small ε with a sufficiently large value of λ chosen. In our numerical experiment, for $\varepsilon = 10^{-3}$, we see that $\lambda = 1$ is enough to ensure $V_\lambda \leq \varepsilon$. Note that this way of choosing λ has been widely used in the literature, e.g., [50], [51], [55], and [56].

Problem (32) is still challenging due to the expectation operator in $g(\boldsymbol{a})$ in its cost function $\mathcal{L}(\boldsymbol{a})$. Following the procedure proposed in [32], we approximate $\mathcal{L}(\boldsymbol{a})$ by its surrogate function $\widetilde{\mathcal{L}}(\boldsymbol{a})$, which is given as follows:

$$\widetilde{\mathcal{L}}^{(n+1)}(\boldsymbol{a}) \triangleq \widetilde{g}^{(n+1)}(\boldsymbol{a}) + \lambda\widetilde{V}^{(n+1)}(\boldsymbol{a}) \tag{34}$$

$$\widetilde{g}^{(n+1)}(\boldsymbol{a}) \triangleq (1 - \phi^{(n+1)})\widetilde{g}^{(n)}(\boldsymbol{a}) + \phi^{(n+1)}\widetilde{T}(\boldsymbol{a}) \tag{35}$$

$$\widetilde{V}^{(n+1)}(\boldsymbol{a}) \triangleq V^{(n+1)} + \left((\nabla V)^{(n+1)}\right)^T\left(\boldsymbol{a} - \boldsymbol{a}^{(n+1)}\right) + \tau||\boldsymbol{a} - \boldsymbol{a}^{(n+1)}||^2$$

$$(\nabla V)^{(n+1)} = \boldsymbol{1} - 2\boldsymbol{a}^{(n+1)} \tag{36}$$

where $\phi^{(n+1)}$ is a weighting parameter and τ can be any positive constant. Here, the surrogate function $\widetilde{g}^{(n+1)}(\boldsymbol{a})$ at iteration $n+1$ depends on the surrogate function $\widetilde{g}^{(n)}(\boldsymbol{a})$ at iteration n and the approximate functions $\widetilde{T}(\boldsymbol{a})$ of $T(\boldsymbol{a})$. $\widetilde{g}^{(n)}(\boldsymbol{a})$ is approximately updated as follows:

$$\widetilde{R}_{d,k}(\boldsymbol{v}) \triangleq \frac{\tau_c - \tau_t}{\tau_c \log 2}B\left[\log\left(1 + \frac{\left(\Upsilon_k^{(\kappa)}\right)^2}{\Pi_k^{(\kappa)}}\right) - \frac{\left(\Upsilon_k^{(\kappa)}\right)^2}{\Pi_k^{(\kappa)}} + 2\frac{\Upsilon_k^{(\kappa)}\Upsilon_k}{\Pi_k^{(\kappa)}} - \frac{\left(\Upsilon_k^{(\kappa)}\right)^2\left(\Upsilon_k^2 + \Pi_k\right)}{\Pi_k^{(\kappa)}\left(\left(\Upsilon_k^{(\kappa)}\right)^2 + \Pi_k^{(\kappa)}\right)}\right] \leq R_{d,k}(\boldsymbol{v}) \tag{27}$$

$$\widetilde{R}_{u,k}(\boldsymbol{u}) \triangleq \frac{\tau_c - \tau_t}{\tau_c \log 2}B\left[\log\left(1 + \frac{\left(\Psi_k^{(\kappa)}\right)^2}{\Xi_k^{(\kappa)}}\right) - \frac{\left(\Psi_k^{(\kappa)}\right)^2}{\Xi_k^{(\kappa)}} + 2\frac{\Psi_k^{(\kappa)}\Psi_k}{\Xi_k^{(\kappa)}} - \frac{\left(\Psi_k^{(\kappa)}\right)^2\left(\Psi_k^2 + \Xi_k\right)}{\Xi_k^{(\kappa)}\left(\left(\Psi_k^{(\kappa)}\right)^2 + \Xi_k^{(\kappa)}\right)}\right] \leq R_{u,k}(\boldsymbol{u}) \tag{28}$$

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

$$\widetilde{g}^{(n)}(\boldsymbol{a}) = g^{(n)} + (\nabla g)^{(n)}\left(\boldsymbol{a} - \boldsymbol{a}^{(n+1)}\right) \tag{37}$$

and $\widetilde{T}(\boldsymbol{a})$ is updated as follows:

$$\widetilde{T}(\boldsymbol{a}) \triangleq T^{(n+1)} + \left((\nabla T)^{(n+1)}\right)^T\left(\boldsymbol{a} - \boldsymbol{a}^{(n+1)}\right) + \tau||\boldsymbol{a} - \boldsymbol{a}^{(n+1)}||^2. \tag{38}$$

Now, with (37) and (38), (35) becomes

$$\widetilde{g}^{(n+1)}(\boldsymbol{a}) \triangleq g^{(n+1)} + \left((\nabla g)^{(n+1)}\right)^T\left(\boldsymbol{a} - \boldsymbol{a}^{(n+1)}\right) + \tau||\boldsymbol{a} - \boldsymbol{a}^{(n+1)}||^2 \tag{39}$$

where

$$g^{(n+1)} = \left(1 - \phi^{(n+1)}\right)g^{(n)} + \phi^{(n+1)}T^{(n+1)}$$
$$(\nabla g)^{(n+1)} = \left(1 - \phi^{(n+1)}\right)(\nabla g)^{(n)} + \phi^{(n+1)}(\nabla T)^{(n+1)} \tag{40}$$

with $g^{(0)} = 0$, $(\nabla g)^{(0)} = \mathbf{0}$. Here

$$(\nabla T)^{(n+1)} = q\frac{(\tilde{\boldsymbol{t}}_d + \tilde{\boldsymbol{t}}_c + \tilde{\boldsymbol{t}}_u)((\boldsymbol{a}^{(n+1})^T\mathbf{1})) - \mathbf{1}\left((\boldsymbol{a}^{(n+1)})^T(\tilde{\boldsymbol{t}}_d + \tilde{\boldsymbol{t}}_c + \tilde{\boldsymbol{t}}_u)\right)}{\left((\boldsymbol{a}^{(n+1})^T\mathbf{1})\right)^2}. \tag{41}$$

From (36) and (39), (34) can be written as follows:

$$\widetilde{\mathcal{L}}^{(n+1)}(\boldsymbol{a}) \triangleq \mathcal{L}^{(n+1)} + \left((\nabla \mathcal{L})^{(n+1)}\right)^T\left(\boldsymbol{a} - \boldsymbol{a}^{(n+1)}\right) + \tau||\boldsymbol{a} - \boldsymbol{a}^{(n+1)}||^2 \tag{42}$$

where

$$\mathcal{L}^{(n+1)} = g^{(n+1)} + \lambda V^{(n+1)} \tag{43}$$
$$(\nabla \mathcal{L})^{(n+1)} = (\nabla g)^{(n+1)} + \lambda(\nabla V)^{(n+1)}. \tag{44}$$

At the large-scale coherence time or iteration $n + 1$, problem (32) is approximated by the following convex problem:

$$\min_{\boldsymbol{a}\in\mathcal{H}} \widetilde{\mathcal{L}}^{(n+1)}(\boldsymbol{a}). \tag{45}$$

C. Solving the Overall Problem (20)

Algorithm 2 outlines the main steps to solve the overall problem (20) [hence (16)]. In the large-scale coherence time n (i.e., iteration n), for a given random value of $\boldsymbol{a}^{(n+1)} \in \mathcal{H}$, the short-term subproblem (21) is solved by Algorithm 1 after $I_S^{(n)}$ iterations to obtain a KKT solution. This solution is then used to construct the approximate long-term master problem (45). After solving (45) to obtain an optimal solution $(\boldsymbol{a}^*)^{(n+1)}$, we update $\boldsymbol{a}^{(n+2)}$ as follows:

$$a_k^{(n+2)} = \left(1 - \pi^{(n+1)}\right)a_k^{(n+1)} + \pi^{(n+1)}\left(a_k^*\right)^{(n+1)} \quad \forall k \tag{46}$$

where $\pi^{(n+1)}$ is a weighting parameter, and $\{\phi^{(n)}, \pi^{(n)}\}$ are chosen to satisfy the following conditions [32, Assumption 5].

(B1): $\phi^{(n)} \to 0$, $(1/[\phi^{(n)}]) \leq \mathcal{O}(n^{\varsigma})$ for $\varsigma \in (0, 1)$, and $\sum_n(\phi^{(n)})^2 < \infty$.

(B2): $\pi^{(n)} \to 0$, $\sum_n \pi^{(n)} = \infty$, $\sum_n(\pi^{(n)})^2 < \infty$, and $\lim_{n\to\infty}([\pi^{(n)}]/[\phi^{(n)}]) = 0$.

Fig. 3 provides a top-level illustration of the operation of Algorithm 2 within the context of CFmMIMO-based wireless FL. In the beginning, we run Algorithm 2 based on the collected network information. Specifically, we solve the short-term subproblem (21) within a short-term optimization (STO) time block, and the long-term master problem (22) within a long-term optimization (LTO) time block. Every two problems (21) and (22) are solved within a *large-scale coherence time*. Eventually, the converged UE selection solution $\boldsymbol{a}^*$ provided by Algorithm 2 is used within the subsequent FL process as described in Section II-B. In the FL training process, one FL communication round and one STO time block are executed in one large-scale coherence time. Here, the values of transmit power and processing frequency obtained in the STO time block are not those obtained in the STO time block when executing Algorithm 2, but rather are computed by the same Algorithm 1. The detailed execution of one FL communication round is discussed in Section II-B and illustrated in Fig. 2. Once the network information changes, we will re-engage Algorithm 2 to get a new solution, ready for FL to execute again.

Algorithm 2 Online Successive Convex Approximation Approach for Solving the Overall Problem (20)

```
**Initialization**: Set $n = 0$, select a random $\boldsymbol{a}^{(n+1)} \in \mathcal{H}$
**repeat**
    Solve the short-term subproblem (21) to obtain its optimal solution $(\boldsymbol{\eta}^*,\boldsymbol{f}^*,\boldsymbol{\zeta}^*)$ by using Algorithm 1, and update $(\boldsymbol{\eta}^{(n+1)},\boldsymbol{f}^{(n+1)}\boldsymbol{\zeta}^{(n+1)}) = (\boldsymbol{\eta}^*,\boldsymbol{f}^*,\boldsymbol{\zeta}^*)$
    Solve the approximate long-term master problem (45) of (22) using CVX to obtain its optimal solution $(\boldsymbol{a}^*)^{(n+1)}$
    Update $\boldsymbol{a}^{(n+2)}$ by (46) and $n = n + 1$
**until** convergence
**Output**: $\boldsymbol{a}^* = \boldsymbol{a}^{(n+1)}$
```

D. Convergence and Complexity Analyses of Algorithm 2

Definition 1: A solution $(\boldsymbol{a}^*, \boldsymbol{x}^*)$ is called a stationary solution of problem (20) [or (16)] if $\boldsymbol{x}^*$ is a KKT solution of the short-term subproblem (21) for $\boldsymbol{a} = \boldsymbol{a}^*$, and $\boldsymbol{a}^*$ is a KKT solution of the long-term master problem (22) for $\boldsymbol{x} = \boldsymbol{x}^*$.

Proposition 3: Algorithm 2 converges to a neighborhood of the stationary solutions to problem (16).

Proof: See the Appendix. ■

Theoretically, $I_S^{(n)} \to \infty$ and $\lambda \to \infty$ are required for Algorithm 2 to converge to the stationary solutions of problem (16). When $I_S^{(n)}$ and λ are finite, Algorithm 2 converges to approximate stationary solutions of problem (16).

The computational complexities of solving (31) at each iteration of Algorithm 1 and solving (45) at each iteration of Algorithm 2 are polynomial in the number of variables and constraints. In particular, (31) can be transformed into an equivalent optimization problem that involves $N_{v,1} \triangleq (2MN + 4N + 3)$ real-valued scalar decision variables, $N_{l,1} \triangleq (2MN+M+4N+3)$ linear constraints, and $N_{q,1} \triangleq (MN+5N)$ quadratic constraints. Therefore, (31) requires a complexity of $\mathcal{O}(\sqrt{N_{l,1} + N_{q,1}}[N_{v,1} + N_{l,1} + N_{q,1}]N_{v,1}^2)$ [57], [58]. Problem (45) involves $N_{v,2} \triangleq N$ real-valued scalar decision

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

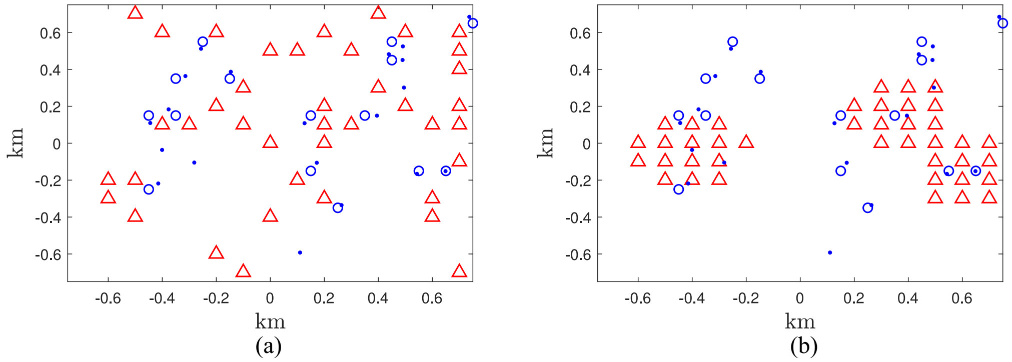

Fig. 4. Examples of the network setup in two cases (a) (C1) and (b) (C2) with 40 APs and 15 UEs. The AP, UE, and fixed locations are represented as the markers in the shapes of triangles, circles, and dots, respectively.

variables, $N_{l,2} \triangleq (N+1)$ linear constraints. As such, (45) requires a complexity of $\mathcal{O}(\sqrt{N_{l,2}}[N_{v,2}+N_{l,2}]N_{v,2}^2)$.

V. Numerical Examples

This section provides numerical results to analyze the effectiveness of the proposed Algorithm 2 in minimizing the execution time of the considered standard FL process [29], [30]. As previously discussed in Section I, the ultimate question is how to find an optimal set of UEs to reduce the FL execution time without satisfying the test accuracy too much. To answer this question, we need to answer the fundamental questions (Q1) and (Q2). Assuming that N_{QoL} is known in advance, our work focuses on answering question (Q2). It is shown in this section that the number of selected UEs in our example is always larger than N_{QoL}. Therefore, the test accuracy obtained with this set of selected UEs is always acceptable. This means with the UE selection of our approach, the test accuracy of an FL process using real data sets is expected to be the same as that in [29] and [30], and hence, not shown in this article. On the other hand, if we try to make an ultimate analysis on the test accuracy of our UE selection scheme, we first have to set N_{QoL} correctly, which leads us back to answering question (Q1). However, answering question (Q1) requires extensive efforts that are out of the scope of this work. Based on this observation, we leave the analysis of the test accuracy of an FL process with our UE selection scheme for future work.

A. Network Setup With Nonuniform UE/AP Distribution

We consider a CFmMIMO network in a square of $D \times D$ km^2 whose edges are wrapped around to avoid the boundary effects. We examine the following two cases.

1) *Case (C1):* The UEs are more likely to stay near some fixed locations (e.g., coffee shops and restaurants) in the considered area; and the APs are uniformly distributed across the considered area.
2) *Case (C2):* Both the APs and UEs are more likely to stay close to some fixed locations in the considered area.

1) Modeling of Case (C1): First, a set of x_p fixed locations are uniformly distributed over the network square. To generate UE locations, we create a grid that has x_l vertical and x_l horizontal lines. Then, N UE locations are chosen as the points on this grid that are closest to x_p fixed locations. To generate AP locations, we create a new grid that is the same as the UE location grid, but the lines of these two grids are interleaved. Finally, M AP locations are uniformly chosen in the points on the latter grid. Here, we consider $x_l = 15$ and $x_p = 20$.

2) Modeling of Case (C2): Case (C2) is modeled in the same way for case (C1), except that the M AP locations are chosen from the points on the AP location grid which are closest to x_{AP} fixed locations. Here, we choose $x_{\text{AP}} = 3$, and the x_{AP} fixed locations are uniformly selected out of x_p fixed locations without replacement. An example of the network setup is shown in Fig. 4.

3) Setup for Each Network Realization: Since each FL communication round happens in one large-scale coherence time (in the order of seconds), the total execution time of an FL process is expected to be around several minutes. Therefore, we assume that the UEs only move around their current locations during the FL process. Here, in each communication round of an FL process, we let each UE move within a circle of radius 5 m around its current location, while the AP locations remain unchanged.

B. Parameter Settings

We model large-scale fading coefficients β_{mk} as [59]

$$\beta_{mk} = 10^{\frac{\text{PL}_{mk}^d}{10}} 10^{\frac{F_{mk}}{10}} \tag{47}$$

where $10^{(\text{PL}_{mk}^d/10)}$ represents the path loss, and $10^{(F_{mk}/10)}$ represents the shadowing effect with $F_{mk} \in \mathcal{N}(0, 4^2)$ (in dB). Here, PL_{mk}^d (in dB) is given by [59]

$$\text{PL}_{mk}^d = -30.5 - 36.7 \log_{10}\left(\frac{d_{mk}}{1\text{ m}}\right) \tag{48}$$

and the correlation among the shadowing terms from the AP m $\forall m \in \mathcal{M}$ to different UEs $k, \ell \in \mathcal{N}$ is expressed as follows:

$$\mathbb{E}\{F_{mk}F_{j\ell}\} \triangleq \begin{cases} 4^2 2^{-\delta_{k\ell}/9\,\text{m}}, & \text{if } j = m \\ 0, & \text{otherwise} \end{cases} \quad \forall j \in \mathcal{M} \tag{49}$$

where $\delta_{k\ell}$ is the physical distance between UEs k and ℓ.

For channel estimation, we use a random pilot assignment scheme. Specifically, the pilot of each user is randomly chosen from a predefined set of $\tau_t = N$ orthogonal pilot sequences,

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

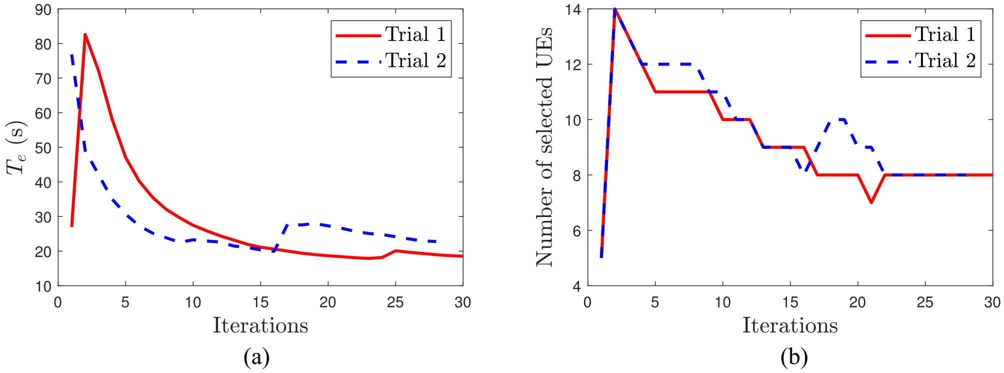

Fig. 5. Convergence process of Algorithm 2: (a) FL execution time and (b) number of selected UEs.

each having a length of τ_t samples. We set $\tau_c=200$ samples, and $S_d = S_u = 5$ MB, noise power $\sigma_0^2 = -92$ dBm, $L = 5$, $f_{\max} = 3 \times 10^9$ cycles/s, $D_k = D = 5 \times 10^6$ samples, $c_k = 20$ cycles/samples [42], for all k, $\alpha = 2 \times 10^{-29}$. We choose $q = 90$. Let $\tilde{\rho}_d = 1$ W, $\tilde{\rho}_u = 0.2$ W, and $\tilde{\rho}_t = 0.2$ W be the maximum transmit powers of the APs, UEs, and uplink pilot sequences, respectively. Here, ρ_d, ρ_u, and ρ_t are the normalized values of $\tilde{\rho}_d$, $\tilde{\rho}_u$, and $\tilde{\rho}_t$ with respect to the noise power. We set $\pi^{(n)} = (1000/1000 + n)$ and $\phi^{(n)} = (1/n^{9/10})$ to satisfy conditions (B1) and (B2) in Section IV-C. Finally, we choose $\lambda = 1$.

C. Results and Discussion

1) Effectiveness of Algorithm 2: First, we evaluate the convergence behavior of the proposed Algorithm 2. As seen from Fig. 5, Algorithm 2 converges within 30 iterations for an arbitrary network realization. Note that each iteration of Algorithm 2 involves solving simple convex programs (31) and (45). It is therefore expected that Algorithm 2 has a low computational complexity.

Next, we compare Algorithm 2 (denoted by **OPT** in the figures) with the following baseline schemes.

1) *Baseline 1 (BL1):* Let $\widehat{N}$ be an integer uniformly drawn from the interval $[N_{\text{QoL}}, N]$. Then, before an FL process is executed, we select participating UEs for this FL process by uniformly selecting $\widehat{N} \le N$ UEs without replacement. The UE selection result of this scheme is presented by a vector $\boldsymbol{a}_{\text{BL1}}$. The FL execution time by *BL1* is thus $T_e(\boldsymbol{a}_{\text{BL1}}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta}) = G(\boldsymbol{a}_{\text{BL1}})\mathbb{E}\{T_o(\boldsymbol{a}_{\text{BL1}}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta})\}$. Since $\boldsymbol{a}_{\text{BL1}}$ is known, solving the problem of minimizing $T_e(\boldsymbol{a}_{\text{BL1}}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta})$ only requires solving the short-term subproblem (21) of optimizing $(\boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta})$ to minimize the execution time of one FL communication round. Here, (21) is solved by using Algorithm 1 for a given $\boldsymbol{a}_{\text{BL1}}$.
2) *Baseline 2 (BL2):* In this baseline, we let all N UEs participate in an FL *process* but only $K \le N$ UEs participate in each FL *communication round*. Here, the integer K is uniformly drawn from the interval $[N_{\text{QoL}}, N]$. Then, K UEs are uniformly chosen out of N original UEs without replacement. The UE selection result of this scheme in each FL communication round is represented by a vector $\boldsymbol{a}_{\text{BL2}}$. Note that *BL2* is identical to the opportunistic UE sampling scheme proposed in [29] and [30]. Since *BL2* performs UE selection in each FL communication round, the number of FL communication rounds is $\widetilde{G} = (q/K) + \tilde{q}(1 - (K/N))$ [30, Th. 1], where q and $\tilde{q}$ depend on the specific characteristics of the FL learning problems and are assumed known in advance. Here, since there is no UE sampling in each FL communication round in our proposed approach, all $\widetilde{N} = \sum_{k\in\mathcal{N}} a_k$ selected UEs participate in an FL process, i.e., $K = \widetilde{N}$. Therefore, the number of FL communication rounds are $(q/\widetilde{N})$ as in (14). In this work, we choose $\tilde{q} = q$ for simplicity. Then, the execution time of an FL process using *BL2* is measured by $T_e(\boldsymbol{a}_{\text{BL2}}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta}) = \widetilde{G}\mathbb{E}\{T_o(\boldsymbol{a}_{\text{BL2}}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta})\}$. Since $\boldsymbol{a}_{\text{BL2}}$ is known in each FL communication round, solving the problem of minimizing $T_e(\boldsymbol{a}_{\text{BL2}}, \boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta})$ only requires solving the short-term subproblem (21) of optimizing $(\boldsymbol{\eta}, \boldsymbol{f}, \boldsymbol{\zeta})$ to minimize the execution time of one FL communication round. Here, (21) is solved by using Algorithm 1 for a given $\boldsymbol{a}_{\text{BL2}}$.

Fig. 6 compares the total execution time of an FL process by all the considered schemes. As seen, our **OPT** scheme is the best performer. In particular, while *BL1* and *BL2* perform quite similarly, **OPT** cuts the execution time by a substantial amount, e.g., by up to 50% in case (C1) and 66% in case (C2) with $M = 20$ and $D = 1.5$ km. These results show the significant advantage of an optimal UE selection over heuristic UE selections.

Fig. 6 also shows the importance of optimal UE selection for reducing the FL execution time when the AP density defined as the number of APs over a geographical area is moderately low. Specifically, in both cases (C1) and (C2) with a large value of D or a low value of M, the reduction in the FL execution time by **OPT** is at least 44%. This is reasonable because in these cases, there is a high probability of having UEs with unfavorable links. This leads to a significantly low execution time of one FL communication time, and hence, the whole FL execution time.

2) Impact of the Number of APs on the Number of Selected UEs: Fig. 7 shows that a larger number of APs corresponds to a larger number of UEs being selected. Here, as the AP-UE distances are smaller, there is potentially more UEs with favorable links hence being selected. The only

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

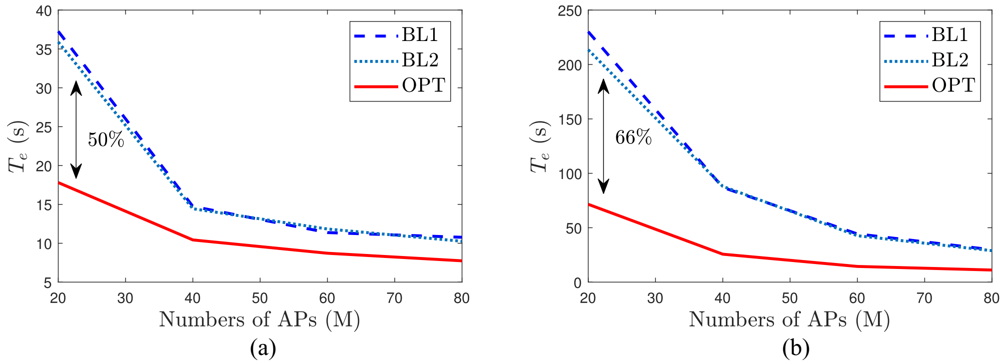

Fig. 6. Effectiveness of the proposed approach. In these examples, we set $N = 15$ and $N_{\text{QoL}} = 5$. (a) Case (C1) with $D = 1.5$ km. (b) Case (C2) with $D = 1.5$ km.

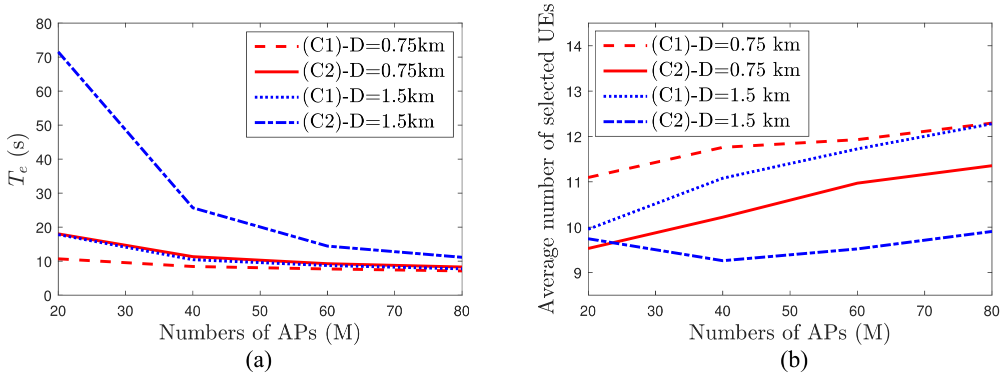

Fig. 7. Impact of the number of APs on: (a) FL execution time and (b) number of selected UEs. In this example, we set $N = 15$ and $N_{\text{QoL}} = 5$.

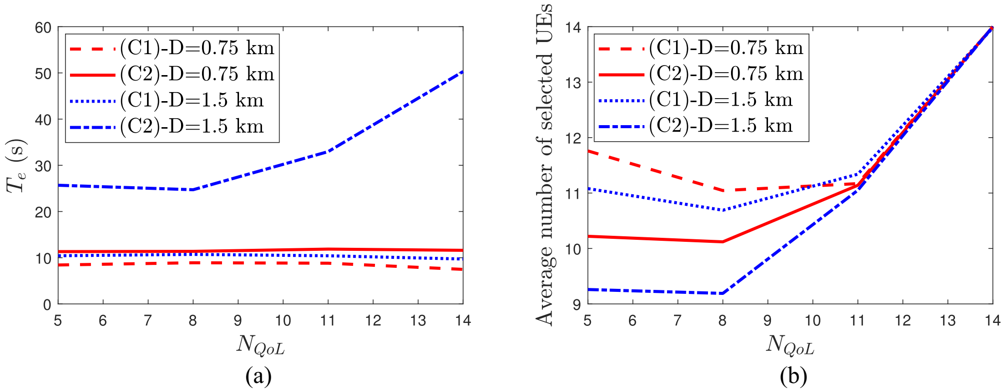

Fig. 8. Impact of the threshold N_{QoL} on: (a) FL execution time and (b) number of UEs being selected. In this example, we set $M = 40$ and $N = 15$.

time the number of UEs being selected decreases is in case (C2) with $M = \{20, 40\}$ and $D = 1.5$ km. In this case, there are a high probability of having UEs with unfavorable links, which leads to the two largest execution times as shown in Fig. 7(a). Therefore, the numbers of selected UEs still need to be reduced to shorten the execution time of one FL communication, and hence, the whole FL execution time.

3) Impact of N_{QoL} on the Execution Time of an FL Process: Fig. 8(a) shows that increasing N_{QoL} leads to a dramatic increase in the FL execution time in a network that has a significantly low density of APs and nonuniformly distributed AP locations. This network is presented in case (C2) with $D = 1.5$ km, where there is a high probability of having UEs with unfavorable links. In this case, for a larger value of N_{QoL}, more UEs are required to participate in an FL process as shown in Fig. 8(b). While this reduces the number of FL communication rounds, a stronger interuser interference is also resulted. In our example, such a reduction in the number of communication rounds is not sufficient to compensate for the increase in the execution time of each FL communication round.

Fig. 8(a) also shows that CFmMIMO networks with a high density of APs potentially provide low-latency FL services for everyone. When there are fewer UEs with unfavorable links as in case (C1) with $D = \{0.75, 1.5\}$ km and case (C2) with $D = 0.75$ km, the FL execution times are nearly the same when N_{QoL} is increased.

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

VI. Conclusion

In this work, we have proposed a novel approach that jointly designs UE selection, transmit power, and processing frequency to minimize the execution time of an FL process in CFmMIMO networks. We formulate a mixed-integer mixed-timescale stochastic nonconvex problem under practical requirements on the maximum transmit powers and the minimum number of selected UEs to guarantee QoL. Utilizing online successive convex approximation, we have successfully developed a novel algorithm to solve the formulated problem. The proposed algorithm has been proved to converge to the neighborhood of stationary points. Numerical results have showed that our approach significantly reduces the FL execution time over the baseline schemes.

Appendix
Proof of Proposition 3

The proof involves two steps. The first step is to prove that the solution $\boldsymbol{x}^*$ obtained from Algorithm 2 is a KKT solution of the short-term subproblem (21). This proof has already been provided in Proposition 1. The second step is to prove that the solution $\boldsymbol{a}^{(n+1)}$ obtained from Algorithm 2 is a KKT solution of the long-term master problem (22). The details of this proof are as follows.

It can be confirmed that problem (20) satisfies the conditions of Assumption 1 on the main problem in the general framework [32]. It is worth noting that we do not verify Assumptions 1–5 and [32, Assumptions 1–6] for the following reasons. Assumptions 1–5 on Mangasarian–Fromovitz constraint qualification is used to ensure the existence of KKT solutions of the short-term subproblem (21). In this work, since Proposition 1 shows that a KKT solution of (21) can be obtained by Algorithm 1, this assumption is unnecessary. Assumptions 1–6 is used to guarantee convergence to an exact stationary point of the short-term subproblem (21). However, Liu *et al.* [32] confirmed that Assumptions 1–6 can be removed when we allow an approximate convergence by solving the short-term subproblem (21) with a finite number of iterations.

From the definitions of $\widetilde{R}_{d,k}(\boldsymbol{v})$, and $\widetilde{R}_{u,k}(\boldsymbol{u})$ in (27) and (28), it can be verified that $\widetilde{R}_{d,k}(\boldsymbol{v})$ and $\widetilde{R}_{u,k}(\boldsymbol{u})$ have the following properties.

1) $\widetilde{R}_{d,k}(\boldsymbol{v}^{(n)}) = R_{d,k}(\boldsymbol{v}^{(n)})$, $\widetilde{R}_{u,k}(\boldsymbol{u}^{(n)}) = h_{u,k}(\boldsymbol{u}^{(n)})$, $\nabla\widetilde{R}_{d,k}(\boldsymbol{v}^{(n)}) = \nabla R_{d,k}(\boldsymbol{v}^{(n)})$, $\nabla\widetilde{R}_{u,k}(\boldsymbol{u}^{(n)}) = \nabla R_{u,k}(\boldsymbol{u}^{(n)})$.
2) $-\widetilde{R}_{d,k}(\boldsymbol{v})$, and $-\widetilde{R}_{u,k}(\boldsymbol{u})$ are strongly convex.
3) $\widetilde{R}_{d,k}(\boldsymbol{v}, \boldsymbol{v}^{(n)})$ and $\widetilde{R}_{u,k}(\boldsymbol{u}, \boldsymbol{u}^{(n)})$ are Lipschitz continuous in both $\boldsymbol{v}, \boldsymbol{v}^{(n)}$ and both $\boldsymbol{u}, \boldsymbol{u}^{(n)}$, respectively.

Algorithm 1 thus satisfies all the conditions of Assumption 2 on the short-term algorithm within the general framework [32]. Since $\{\phi^{(n)}, \pi^{(n)}\}$ are chosen to satisfy conditions (B1) and (B2) in Section IV-C, they satisfy all the conditions of [32, Assumption 5]. When [32, Assumptions 1, 2, and 5] are satisfied, it is confirmed by [32, Corollary 1] that the surrogate function $\widetilde{\mathcal{L}}(\boldsymbol{a})$ in (45) satisfies the [32, Assumptions 3 and 4] on the properties and asymptotic consistency of surrogate functions.

Since [32, Assumptions 1–5] are all satisfied, it follows from [32, Lemma 1] that:

1) the sequence $\{(\boldsymbol{a}^{(n+1)}, (\boldsymbol{a}^*)^{(n+1)})\}_{n=1}^{\infty}$ generated over iterations of Algorithm 2 has the following property:

$$\lim_{n\to\infty}\left\|\boldsymbol{a}^{(n+1)} - (\boldsymbol{a}^*)^{(n+1)}\right\| = 0 \tag{50}$$

2) let $\boldsymbol{a}_\star$ be a limit point of a subsequence $\{\boldsymbol{a}^{(n+1)_j}\}_{j=1}^{\infty}$ and

$$\lim_{j\to\infty}\left|\widetilde{\mathcal{L}}\left(\boldsymbol{a}^{(n+1)_j}\right) - \widetilde{\mathcal{L}}(\boldsymbol{a}_\star)\right| = 0 \tag{51}$$

$$\lim_{j\to\infty}\left|\nabla\widetilde{\mathcal{L}}\left(\boldsymbol{a}^{(n+1)_j}\right) - \nabla\mathcal{L}(\boldsymbol{a}_\star)\right| = 0. \tag{52}$$

Without loss of generality, we assume that $\boldsymbol{a}^{(n+1)} \to \boldsymbol{a}_\star$ as $n \to \infty$. Then, (51) and (52) imply that

$$\lim_{n\to\infty}\left|\widetilde{\mathcal{L}}\left(\boldsymbol{a}^{(n+1)}\right) - \mathcal{L}\left(\boldsymbol{a}^{(n+1)}\right)\right| = 0 \tag{53}$$

$$\lim_{n\to\infty}\left|\nabla\widetilde{\mathcal{L}}\left(\boldsymbol{a}^{(n+1)}\right) - \nabla\mathcal{L}\left(\boldsymbol{a}^{(n+1)}\right)\right| = 0. \tag{54}$$

It can be seen that there always exists one interior point in $\mathcal{H}$. Therefore, the convex problem (45) satisfies Slater's constraint qualification condition. Its optimal solution $(\boldsymbol{a}^*)^{(n+1)}$ is thus a KKT solution to (32), and hence, (22) when $\lambda \to \infty$ (see Proposition 2), i.e.,

$$\nabla\mathcal{L}\left(\left(\boldsymbol{a}^*\right)^{(n+1)}\right) + \sum_{j=1}^{r}\nu_j\nabla\delta_j\left(\left(\boldsymbol{a}^*\right)^{(n+1)}\right) = 0 \tag{55a}$$

$$\nu_j\delta_j\left(\left(\boldsymbol{a}^*\right)^{(n+1)}\right) = 0 \quad \forall j \in \{1, \dots, q\} \tag{55b}$$

where $\delta_j(\boldsymbol{a})$ $\forall j \in \{1, \dots, q\}$ represent the functions in the constraints (2), (17), and (18). It follows from (50) and (54) that the gap between $\boldsymbol{a}^{(n+1)}$ and $(\boldsymbol{a}^*)^{(n+1)}$ and that between $\nabla\widetilde{\mathcal{L}}(\boldsymbol{a}^{(n+1)})$ and $\nabla\mathcal{L}(\boldsymbol{a}^{(n+1)})$ converge to zero as $n \to \infty$. Therefore, (55) implies

$$\nabla\mathcal{L}\left(\boldsymbol{a}^{(n+1)}\right) + \sum_{j=1}^{r}\nu_j\nabla\delta_j\left(\boldsymbol{a}^{(n+1)}\right) = 0 \tag{56a}$$

$$\nu_j\delta_j\left(\boldsymbol{a}^{(n+1)}\right) = 0 \quad \forall j \in \{1, \dots, r\} \tag{56b}$$

which means $\boldsymbol{a}^{(n+1)}$ is a KKT solution of the long-term master problem (22).

As such, the convergence of Algorithm 2 to a stationary point of problem (20) in the sense of Definition 1 is guaranteed if the numbers of iterations of Algorithms 1 and 2 are infinity, i.e., $I_S^{(n)} \to \infty$, $I_L \to \infty$, and $\lambda \to \infty$. In practice, it is acceptable to choose finite $\{I_S^{(n)}\}_{n\in\{1,\dots,I_L\}}$, I_L, and λ for an approximate convergence. Therefore, Algorithm 2 is guaranteed to converge to the neighborhood of the stationary solutions of problem (20), and hence, (16).

References

[1] "Cisco Annual Internet Report." Cisco. Mar. 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/index.html

[2] J. Dong, M. Noreikis, Y. Xiao, and A. Ylä-Jääski, "ViNav: A vision-based indoor navigation system for smartphones," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1461–1475, Jun. 2019.

[3] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

[4] F. D. Calabrese, L. Wang, E. Ghadimi, G. Peters, L. Hanzo, and P. Soldati, “Learning radio resource management in RANs: Framework, opportunities, and challenges,” *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 138–145, Sep. 2018.

[5] Q.-V. Pham, N. T. Nguyen, T. Huynh-The, L. B. Le, K. Lee, and W.-J. Hwang, “Intelligent radio signal processing: A survey,” *IEEE Access*, vol. 9, pp. 83818–83850, 2021.

[6] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, “Toward an intelligent edge: Wireless communication meets machine learning,” *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19–25, Jan. 2020.

[7] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, “Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges,” *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, 3rd Quart., 2021.

[8] S. Niknam, H. S. Dhillon, and J. H. Reed, “Federated learning for wireless communications: Motivation, opportunities, and challenges,” *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, Jun. 2020.

[9] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[10] Z. Du, C. Wu, T. Yoshinaga, K.-L. A. Yau, Y. Ji, and J. Li, “Federated learning for vehicular Internet of Things: Recent advances and open issues,” *IEEE Open J. Comput. Soc.*, vol. 1, pp. 45–61, 2020.

[11] H. Song, J. Bai, Y. Yi, J. Wu, and L. Liu, “Artificial intelligence enabled Internet of Things: Network architecture and spectrum access,” *IEEE Comput. Intell. Mag.*, vol. 15, no. 1, pp. 44–51, Feb. 2020.

[12] M. Chen, H. V. Poor, W. Saad, and S. Cui, “Wireless communications for collaborative federated learning,” *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 48–54, Jan. 2021.

[13] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, “A joint learning and communications framework for federated learning over wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[14] M. M. Amiri and D. Gündüz, “Federated learning over wireless fading channels,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, May 2020.

[15] M. Chen, H. V. Poor, W. Saad, and S. Cui, “Convergence time minimization of federated learning over wireless networks,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[16] V.-D. Nguyen, S. K. Sharma, T. X. Vu, S. Chatzinotas, and B. Ottersten, “Efficient federated learning algorithm for resource allocation in wireless IoT networks,” *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3394–3409, Mar. 2021.

[17] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, “Energy efficient federated learning over wireless communication networks,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[18] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, “Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7947–7962, Dec. 2021.

[19] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, “Energy-efficient radio resource allocation for federated edge learning,” in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6.

[20] Y. Hu, H. Huang, and N. Yu, “Device scheduling for energy-efficient federated learning over wireless network based on TDMA mode,” in *Proc. IEEE Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2020, pp. 286–291.

[21] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, “Cell-free massive MIMO for wireless federated learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6377–6392, Oct. 2020.

[22] T. T. Vu, D. T. Ngo, H. Q. Ngo, M. N. Dao, N. H. Tran, and R. H. Middleton, “Straggler effect mitigation for federated learning in cell-free massive MIMO,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.

[23] T. T. Vu, H. Q. Ngo, T. L. Marzetta, and M. Matthaiou, “How does cell-free massive MIMO support multiple federated learning groups?” in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sep. 2021, pp. 401–405.

[24] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, “Cell-free massive MIMO versus small cells,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[25] Y. Liu, J. Peng, J. Kang, A. M. Iliyasu, D. Niyato, and A. A. A. El-Latif, “A secure federated learning framework for 5G networks,” *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 24–31, Aug. 2020.

[26] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, “On the convergence of FedAvg on non-IID data,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–26.

[27] F. Haddadpour and M. Mahdavi, “On the convergence of local descent methods in federated learning,” 2019. *arXiv:1910.14425*.

[28] W. Xia, T. Q. S. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, “Multi-armed bandit-based client scheduling for federated learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7108–7123, Nov. 2020.

[29] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2017, pp. 1273–1282.

[30] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, “SCAFFOLD: Stochastic controlled averaging for federated learning,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 119, Jul. 2020, pp. 5132–5143.

[31] K. Yang, T. Jiang, Y. Shi, and Z. Ding, “Federated learning via over-the-air computation,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[32] A. Liu, V. K. N. Lau, and M.-J. Zhao, “Online successive convex approximation for two-stage stochastic nonconvex optimization,” *IEEE Trans. Signal Process.*, vol. 66, no. 22, pp. 5941–5955, Nov. 2018.

[33] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, *Fundamentals of Massive MIMO*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[34] E. Björnson, J. Hoydis, and L. Sanguinetti, “Massive MIMO networks: Spectral, energy, and hardware efficiency,” *Found. Trends Signal Process.*, vol. 11, nos. 3–4, pp. 154–655, 2017.

[35] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for next generation wireless systems,” *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[36] C. Shepard *et al.*, “Argos: Practical many-antenna base stations,” in *Proc. ACM Int. Conf. Mobile Comput. Netw.*, 2012, pp. 53–64.

[37] F. Kaltenberger, H. Jiang, M. Guillaud, and R. Knopp, “Relative channel reciprocity calibration in MIMO/TDD systems,” in *Proc. Future Netw. Mobile Summit*, 2010, pp. 1–10.

[38] C. Xu, Y. Qu, Y. Xiang, and L. Gao, “Asynchronous federated learning on heterogeneous devices: A survey,” *ACM Comput. Survey*, vol. 37, no. 4, p. 111, Aug. 2021.

[39] C. T. Dinh *et al.*, “Federated learning over wireless networks: Convergence analysis and resource allocation,” *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, Feb. 2021.

[40] C. T. Dinh, N. Tran, and J. Nguyen, “Personalized federated learning with Moreau envelopes,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 21394–21405.

[41] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, “Robust and communication-efficient federated learning from non-i.i.d. data,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[42] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, “Federated learning over wireless networks: Optimization model design and analysis,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 1387–1395.

[43] I. Viering, H. Hofstetter, and W. Utschick, “Spatial long-term variations in urban, rural and indoor environments,” in *Proc. COST273 5th Meeting*, 2002.

[44] “5G wireless access: An overview,” Ericsson, Stockholm, Sweden, Apr. 2020.

[45] C.-R. Shyu *et al.*, “A systematic review of federated learning in the healthcare area: From the perspective of data properties and applications,” *Appl. Sci.*, vol. 11, no. 23, 2021, Art. no. 11191.

[46] G. Interdonato, E. Björnson, H. Q. Ngo, P. K. Frenger, and E. G. Larsson, “Ubiquitous cell-free massive MIMO communications,” *EURASIP J. Wireless Commun. Netw.*, vol. 2019, p. 197, Aug. 2019.

[47] Z. H. Shaik, E. Björnson, and E. G. Larsson, “MMSE-optimal sequential processing for cell-free massive MIMO with radio stripes,” *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7775–7789, Nov. 2021.

[48] L. Miretti, E. Björnson, and D. Gesbert, “Precoding for scalable cell-free massive MIMO with radio stripes,” in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sep. 2021, pp. 411–415.

[49] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in *Proc. 2nd USENIX Conf. HotCloud*, 2010, p. 4.

[50] T. T. Vu, D. T. Ngo, M. N. Dao, S. Durrani, and R. H. Middleton, “Spectral and energy efficiency maximization for content-centric C-RANs with edge caching,” *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6628–6642, Dec. 2018.

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.

[51] T. T. Vu, D. T. Ngo, M. N. Dao, S. Durrani, D. H. N. Nguyen, and R. H. Middleton, "Energy efficiency maximization for downlink cloud radio access networks with data sharing and data compression," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 4955–4970, Aug. 2018.

[52] V. D. Nguyen, T. Q. Duong, H. D. Tuan, O.-S. Shin, and H. V. Poor, "Spectral and energy efficiencies in full-duplex wireless information and power transfer," *IEEE Trans. Commun.*, vol. 65, no. 5, pp. 2220–2233, May 2017.

[53] M. Grant and S. Boyd. "CVX: MATLAB Software for Disciplined Convex Programming, version 2.1." Mar. 2014. [Online]. Available: http://cvxr.com/cvx

[54] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Oper. Res.*, vol. 26, no. 4, pp. 681–683, 1978.

[55] E. Che, H. D. Tuan, and H. H. Nguyen, "Joint optimization of cooperative beamforming and relay assignment in multi-user wireless relay networks," *IEEE Trans. Wireless Commun*, vol. 13, no. 10, pp. 5481–5495, Oct. 2014.

[56] U. Rashid, H. D. Tuan, H. H. Kha, and H. H. Nguyen, "Joint optimization of source Precoding and relay beamforming in wireless MIMO relay networks," *IEEE Trans. Commun.*, vol. 62, no. 2, pp. 488–499, Feb. 2014.

[57] H. H. M. Tam, H. D. Tuan, D. T. Ngo, T. Q. Duong, and H. V. Poor, "Joint load balancing and interference management for small-cell heterogeneous networks with limited backhaul capacity," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 872–884, Feb. 2017.

[58] Y. Nesterov and A. Nemirovski, *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA, USA: Soc. Ind. Appl. Math., 1994.

[59] E. Björnson and L. Sanguinetti, "Making cell-free massive MIMO competitive with MMSE processing and centralized implementation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 77–90, Jan. 2020.

Tung Thanh Vu (Member, IEEE) received the B.Sc. degree (Hons.) in telecommunications and networking from Ho Chi Minh City University of Science, Ho Chi Minh, Vietnam, in 2012, the M.Sc. degree in telecommunication engineering from Ho Chi Minh City University of Technology, Ho Chi Minh, in 2016, and the Ph.D. degree in wireless communications from The University of Newcastle, Callaghan, NSW, Australia, in 2021. In 2019, he visited the Broadband Communications Research Lab, McGill University, Montreal, QC, Canada. He is currently a Research Fellow with Queen's University Belfast, Belfast, U.K. His research interests include optimization, information theories, and machine-learning applications for 5G-and-beyond wireless networks with advanced technologies, such as massive MIMO, cell-free massive MIMO, federated learning, full-duplex communications, physical-layer security, and low-earth orbit satellite communications.

Dr. Vu received the Best Poster Award at AMSI Optimise Conference in 2018. He is serving as an Editor of *Physical Communication* (Elsevier). He was an IEEE WIRELESS COMMUNICATIONS LETTERS Exemplary Reviewer for 2020 and 2021 and an IEEE TRANSACTIONS ON COMMUNICATIONS Exemplary Reviewer for 2021.

Duy Trong Ngo (Senior Member, IEEE) received the B.Eng. degree (First-Class Hons. and the University Medal) in telecommunication engineering from The University of New South Wales, Sydney, NSW, Australia, in 2007, the M.Sc. degree in electrical engineering (communication) from the University of Alberta, Edmonton, AB, Canada, in 2009, and the Ph.D. degree in electrical engineering from McGill University, Montreal, QC, Canada, in 2013.

He is currently an Associate Professor with the School of Engineering, The University of Newcastle, Callaghan, NSW, Australia. His research interests include wireless communications, intelligent transportation systems, and machine-learning applications.

Hien Quoc Ngo (Senior Member, IEEE) is currently a Lecturer with Queen's University Belfast, Belfast, U.K. He has coauthored many research articles in wireless communications and the Cambridge University Press Textbook *Fundamentals of Massive MIMO* in 2016. His main research interests include massive (large scale) MIMO systems, cell-free massive MIMO, physical-layer security, and cooperative communications.

Dr. Ngo received the IEEE ComSoc Stephen O. Rice Prize in Communications Theory in 2015, the IEEE Sweden VT-COM-IT Joint Chapter Best Student Journal Paper Award in 2015, the IEEE ComSoc Leonard G. Abraham Prize in 2017, and the Best Ph.D. Award from EURASIP in 2018. He was awarded the UKRI Future Leaders Fellowship in 2019. He serves as the Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS LETTERS, the *Digital Signal Processing*, and the *Physical Communication* (Elsevier). He was a Guest Editor of *IET Communications* and IEEE ACCESS in 2017.

Minh Ngoc Dao received the B.Sc. (Hons.) and M.Sc. degrees in mathematics from Hanoi National University of Education, Hanoi, Vietnam, in 2004 and 2006, respectively, and the Ph.D. degree in applied mathematics from the University of Toulouse, Toulouse, France, in 2014. He was a Lecturer with Hanoi National University of Education from 2004 to 2010, a Postdoctoral Fellow with The University of British Columbia, Vancouver, BC, Canada, from 2014 to 2016, and a Research Associate with The University of Newcastle, Callaghan NSW, Australia, and The University of New South Wales, Sydney, NSW, Australia, from 2016 to 2020. He is currently a Lecturer with Federation University, Ballarat, VIC, Australia. His research interests include mathematical optimization, convex and variational analysis, control theory, signal processing, and machine learning.

Dr. Dao received the Annual Best Paper Award from the *Journal of Global Optimization* in 2017.

Nguyen Hoang Tran (Senior Member, IEEE) received the B.S. degree in electrical and computer engineering from the HCMC University of Technology, Ho Chi Minh, Vietnam, in 2005, and the Ph.D. degree in electrical and computer engineering from Kyung Hee University, Seoul, South Korea, in 2011. He was an Assistant Professor with the Department of Computer Science and Engineering, Kyung Hee University, from 2012 to 2017. Since 2018, he has been with the School of Computer Science, The University of Sydney, Camperdown, NSW, Australia, where he is currently a Senior Lecturer. His research interests include distributed computing, machine learning, and networking.

Dr. Tran received the Best KHU Thesis Award in engineering in 2011 and several best paper awards, including IEEE ICC 2016 and ACM MSWiM 2019. He receives the Korea NRF Funding for Basic Science and Research from 2016 to 2023 and ARC Discovery Project from 2020 to 2023. He was the Editor of IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING from 2016 to 2020, and the Associate Editor of IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS from 2020 to 2021 in the area of distributed machine learning/federated learning.

Richard H. Middleton (Fellow, IEEE) received the Ph.D. degree from The University of Newcastle, Callaghan, NSW, Australia, in 1987. He was a Research Professor with Hamilton Institute, The National University of Ireland, Maynooth, Ireland, from May 2007 to 2011, the Head of the School of Electrical Engineering and Computing with The University of Newcastle, Callaghan, NSW, Australia, from 2015 to July 2020, and the Former Director of the ARC Training Centre for Food and Beverage Supply Chain Optimization, and the ARC Centre of Excellence for Complex Dynamic Systems and Control. His research interests include a broad range of control systems theory and applications, including communications systems, control of distributed systems, optimization, and systems biology.

Dr. Middleton has served as the Program Chair (CDC 2006), the Co-General Chair (CDC 2017) of CSS Vice President Membership Activities, and the Vice President of Conference Activities. In 2011, he was the President of the IEEE Control Systems Society. He is a Fellow of IFAC and Engineers Australia.

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 12:36:39 UTC from IEEE Xplore. Restrictions apply.