Straggler Effect Mitigation for Federated Learning in Cell-Free Massive MIMO

Straggler Effect Mitigation for Federated Learning in Cell-Free Massive MIMO

Tung T. Vu*†, Duy T. Ngo*, Hien Quoc Ngo†, Minh N. Dao‡, Nguyen H. Tran§, and Richard H. Middleton*
*The University of Newcastle, Callaghan NSW 2308, Australia
§School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia
†School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT7 1NN, United Kingdom
‡School of Engineering, Information Technology and Physical Sciences, Federation University Australia, Ballarat, VIC 3353, Australia
Email: thanhtung.vu@uon.edu.au, duy.ngo@newcastle.edu.au, hien.ngo@qub.ac.uk,
m.dao@federation.edu.au, nguyen.tran@sydney.edu.au, richard.middleton@newcastle.edu.au

***Abstract*—Straggler effect is the main bottleneck in realizing federated learning (FL) in wireless networks. This work proposes a novel user (UE) selection approach to mitigate this effect with UE sampling in cell-free massive multiple-input multiple-output networks. Our proposed approach selects only a small subset of UEs for participating in one FL process. Importantly, since the UEs are selected before any FL process is executed, the performance of FL during the executing time is not affected by our method. Here, we select UEs by solving an FL transmission time minimization problem that jointly optimizes UE selection, power control, and data rate. The problem is formulated to capture the complex interactions among the FL training time, UE selection, and straggler effect. This mixed-integer mixed-timescale stochastic nonconvex problem is constrained by the minimum number of UEs to guarantee the quality of learning. By employing online successive convex approximation, we propose a novel algorithm to solve the formulated problem with guaranteed convergence to the neighbourhood of their stationary points. Our approach can significantly reduce the FL transmission time over baseline approaches, especially in the networks that experience serious straggler effect due to the moderately low density of access points.**

***Index Terms*—Cell-free massive MIMO, federated learning, user selection.**

I. Introduction

Recently, federated learning (FL) has been considered to be a communication-efficient and privacy-reserved solution to train artificial intelligent models at mobile devices in wireless networks [1]. By definition, an FL process is an iterative process in which users (UEs) use their local data to compute training updates, and send the updates to a central server. The central server then aggregates these updates to compute a global training update, and finally send the global training update back to all the UEs. However, the central server needs to wait until receiving the training updates from all the UEs before processing any next step. As such, some straggler UEs who have unfavorable links may dramatically slow down the whole FL process. This is called "straggler effect", which is the main bottleneck in realizing FL in wireless networks.

For mitigating the straggler effect, several solutions have been proposed in [2]–[5] but not yet sufficiently effective. Specifically, [2], [3] propose heuristic UE sampling schemes to select a subset of UEs for uploading the local updates to the central server. These techniques mitigate the straggler effect by reducing the probability of straggler UEs participating in an FL process. However, they are not always effective because there is a chance that sampled UEs are straggler UEs. On the other hand, the optimal/suboptimal UE selection approaches to mitigate the straggler effect are developed in [4], [5]. In spite of that, the time-division multiple access and frequency-division multiple access networks in [4], [5] might not be suitable to support FL. The FL training time in these networks could be drastically prolonged when the number of UEs is large.

Paper Contribution:[1] This paper consider cell-free massive multiple-input multiple-output (CFmMIMO) networks, which have recently been known as a promising candidate to support FL [7]. Because of their macro-diversity gain, favorable and channel hardening property, the stable operation of an FL process is guaranteed by using a communication scheme proposed in [7]. Here, we propose an UE selection approach to mitigate the straggler effect for a general FL framework with UE sampling [2], [3]. In our approach, UEs are selected by solving an FL transmission time minimization problem that captures the complex interaction among the training time, straggler effect, and UE selection. This mixed-integer mixed-timescale stochastic nonconvex problem jointly optimizes UE selection, power control, and data rate. It is also subjected to the practical constraints on the minimum number of UEs to guarantee the quality of learning. Upon using the online successive convex approximation techniques, we develop a novel algorithm that converges to the neighbourhoods of the stationary points of the formulated problem. Numerical results show that our proposed UE selection approach reduces the FL transmission time more than half, compared to the baseline schemes, especially in the networks that have a moderately low density of access points.

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

A. UE Selection Model

Let N be the total number of UEs. Denote by a_k an indicator variable that shows whether a UE $k \in \mathcal{N} \triangleq \{1,\ldots,N\}$ is selected to take part in an FL process or not, 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 to participate in an FL process, i.e.,

$$\widetilde{\mathcal{N}} = \{k | a_k = 1, \forall k \in \mathcal{N}\}, \tag{2}$$

and $\widetilde{N} \triangleq \sum_{k\in\mathcal{N}} a_k$ is the size of $\widetilde{\mathcal{N}}$. Given the binary values of $\{a_k\}_{k\in\mathcal{N}}$, (2) can be rewritten as:

$$\widetilde{\mathcal{N}} \text{ is the index set of } \mathrm{rnd}(\widetilde{N}) \text{ largest elements of } \boldsymbol{a}, \tag{3}$$

where $\boldsymbol{a} \triangleq [a_1,\ldots,a_N]^T$, and $\mathrm{rnd}(\widetilde{N})$ is the nearest integer of $\widetilde{N}$. Note that when the elements of $\boldsymbol{a}$ are binary, (3) is equivalent to (2). When they are not binary, they can be considered as priority weights for UEs. In the latter case, (3) includes the UEs that have highest priority weights, and hence, is equivalent

[1]The extended version of this work was submitted for publication to the IEEE Transactions on Wireless Communications [6].

to (2) in this sense. Here, (3) is used to assist our UE selection approach, which is discussed later in Section IV-3.

B. The General FL Framework with UE Sampling [2], [3]

After UEs are selected, an FL process starts with selected UEs. Here, we consider the process that has a general FL framework with UE sampling [2], [3]. This process includes the four steps:

(S1) The central server sends the global downlink (DL) training update to all $\widetilde{N}$ selected UEs, and choose randomly a subset $\mathcal{S}^{(n)}$ of $K \leq \widetilde{N}$ UEs with replacement according to the sampling probabilities $\{p_1, \ldots, p_{\widetilde{N}}\}$.

(S2) The UEs in $\mathcal{S}^{(n)}$ update and solve their local machine learning (ML) problems on their local data set and then compute the local uplink (UL) training update

(S3) The UEs in $\mathcal{S}^{(n)}$ send their computed local UL training updates to the central server

(S4) The central server computes the global DL training update by aggregating the received UL training updates.

The process continues and terminates when a global accuracy is obtained.

C. CFmMIMO System Model

To support the FL framework discussed above, we consider a CFmMIMO network where the given set $\mathcal{N}$ of UEs (i.e., the clients) and CPU (i.e, the central server) are connected via a set of APs $\mathcal{M} = \{1, ..., M\}$ [8]. The UEs connect to the APs via wireless links, while the APs connect to the CPU via backhaul links with sufficient capacities.

1) UL channel estimation: UL pilot sequences are sent by all the UEs to all the APs simultaneously. 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 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 UE k to 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 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 distributed according to $\mathcal{CN}(0, \sigma^2_{mk})$, where $\sigma^2_{mk} = \frac{\tau_t \rho_t (\beta_{mk})^2}{\sum_{\ell \in \mathcal{N}} \tau_t \rho_t \beta_{m\ell} |\boldsymbol{\varphi}_k^H \boldsymbol{\varphi}_\ell|^2 + 1}$ [8].

2) Step (S1): The CPU encodes the global DL training update intended for UE k into a symbol $s_{d,k} \sim \mathcal{CN}(0,1)$, and sends all the symbols $s_{d,k}, \forall k \in \mathcal{N}$, to all the APs. Denote by S_d (bits) and $R_{d,k}$ (bps) the data size of the global DL training update and the data rate of sending the global DL training update to UE k, respectively. The transmission time from the CPU to all the APs is given by

$$t_{d,B}(\boldsymbol{a}, \boldsymbol{R}_d) = \frac{\sum_{k \in \mathcal{N}} a_k S_d}{\sum_{k \in \mathcal{N}} a_k R_{d,k}}, \tag{4}$$

where $\boldsymbol{R}_d \triangleq [R_{d,1}, \ldots, R_{d,N}]^T$.

To transmit the symbols received from the CPU, the APs first use conjugate beamforming to precode these symbols. Then, the precoded versions will be broadcasted to all the UEs. Specifically, the transmitted signal at 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 AP m is required to meet the average normalized power constraint, i.e., $\mathbb{E}\{|x_{d,m}|^2\} \leq \rho_d$, which can be expressed as the following per-AP power constraint:

$$\sum_{k \in \mathcal{N}} \sigma^2_{mk} \eta_{mk} \leq 1, \forall m. \tag{5}$$

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

$$\forall k \in \mathcal{N} : \text{if } a_k = 0, \text{ then } \forall m \in \mathcal{M}, \eta_{mk} = 0. \tag{6}$$

At UE k, the achievable DL rate (bps) is [8]

$$R_{d,k} \leq h_{d,k}(\boldsymbol{\eta}), \tag{7}$$

where $\boldsymbol{\eta} \triangleq \{\eta_{mk}\}_{m \in \mathcal{M}, k \in \mathcal{N}}$ and $h_{d,k}(\boldsymbol{\eta})$ is given in (8) shown at the top of the next page [8], and B is the bandwidth. The transmission time from the APs to UE k is given by

$$t_{d,k}(a_k, R_{d,k}) = \frac{a_k S_d}{R_{d,k}}. \tag{9}$$

3) Step (S3): For given $\boldsymbol{a}$, let $\mathcal{S}^{(n)} \subset \widetilde{\mathcal{N}}$ be the set of K UEs randomly sampled from the set $\widetilde{\mathcal{N}}$, using the UE sampling techniques [2], [3]. Here, this set is chosen by randomly sampling UE with replacement according to the sampling probabilities $\{p_1, .., p_{\widetilde{N}}\}$. Denote by b_k an indicator showing whether UE k is sampled or not, i.e.,

$$b_k \triangleq \begin{cases} 1, & \text{if } k \in \mathcal{S}^{(n)} \\ 0, & \text{otherwise,} \end{cases}, \forall k \in \mathcal{N}. \tag{10}$$

After computing a local UL training update, UE $k \in \mathcal{S}^{(n)}$ encodes this update into a symbol $s_{u,k} \sim \mathcal{CN}(0,1)$, and allocates a transmit amplitude value $\sqrt{\rho_u \zeta_k}$ to this symbol. A baseband signal i.e., $x_{u,k} = \sqrt{\rho_u \zeta_k} s_{u,k}$, is then sent to all the APs, and subjected to the average transmit power constraint, i.e., $\mathbb{E}\left\{|x_{u,k}|^2\right\} \leq \rho_u$. This constraint can be expressed in a per-UE constraint as

$$0 \leq \zeta_k \leq 1, \forall k \in \mathcal{N}. \tag{11}$$

Since $\{b_k\}_{k \in \mathcal{S}^{(n)}}$ are only chosen for a given $\boldsymbol{a}$, and no power should be allocated to the unsampled UEs, we have

$$\forall k \in \mathcal{N} : \text{if } a_k b_k = 0, \text{ then } \zeta_k = 0. \tag{12}$$

Let S_u (bits) and $R_{u,k}$ (bps) be the data size of the local UL training updates and the data rate of transmitting the local UL training update from UE k to the CPU, respectively. Here, we assume that S_u is the same for all the UEs. The transmission time from UE $k \in \mathcal{S}^{(n)}$ to the APs is given by

$$t_{u,k}(a_k, R_{u,k}) = \frac{a_k b_k S_u}{R_{u,k}}. \tag{13}$$

Using the signals received from all the UEs, the APs compute and send match-filtered signals to the CPU to detect the UEs' message symbols. The transmission time from the APs to the CPU is thus expressed as

$$t_{u,B}(\boldsymbol{a}, \boldsymbol{R}_u) = \frac{\sum_{k \in \mathcal{N}} a_k b_k S_u}{\sum_{k \in \mathcal{N}} a_k b_k R_{u,k}} \tag{14}$$

where $\boldsymbol{R}_u \triangleq [R_{u,1}, \ldots, R_{u,N}]^T$. At the CPU, the achievable UL rate for UE k is given by

$$R_{u,k} \leq h_{u,k}(\boldsymbol{\zeta}), \tag{15}$$

where $\boldsymbol{\zeta} \triangleq \{\zeta_k\}_{k \in \mathcal{N}}$ and $h_{u,k}(\boldsymbol{\zeta})$ is defined in (16) shown at the top of the next page [8].

III. Problem Formulations

The transmission time of Step (S1) involves the transmission time of sending the global DL training update from the CPU to the APs, and that from the APs to all the UEs, i.e.,

$$T_d(\boldsymbol{a}, \boldsymbol{R}_d) = \frac{\sum_{k \in \mathcal{N}} a_k S_d}{\sum_{k \in \mathcal{N}} a_k R_{d,k}} + \underbrace{\max_{k \in \mathcal{N}} \frac{a_k S_d}{R_{d,k}}}_{\text{Straggler effect in Step (S1)}}. \tag{17}$$

$$h_{d,k}(\boldsymbol{\eta})=\frac{\tau_c-\tau_t}{\tau_c}B\log_2\left(1+\frac{\rho_d\big(\sum_{m\in\mathcal{M}}\eta_{mk}^{1/2}\sigma_{mk}^2\big)^2}{\rho_d\sum_{\ell\in\mathcal{N}\setminus k}\big(\sum_{m\in\mathcal{M}}\eta_{m\ell}^{1/2}\sigma_{m\ell}^2\frac{\beta_{mk}}{\beta_{m\ell}}\big)^2|\boldsymbol{\varphi}_\ell^H\boldsymbol{\varphi}_k|^2+\rho_d\sum_{\ell\in\mathcal{N}}\sum_{m\in\mathcal{M}}\eta_{m\ell}\sigma_{m\ell}^2\beta_{mk}+1}\right) \tag{8}$$

$$h_{u,k}(\boldsymbol{\zeta})=\frac{\tau_c-\tau_t}{\tau_c}B\log_2\left(1+\frac{\rho_u\zeta_k\big(\sum_{m\in\mathcal{M}}\sigma_{mk}^2\big)^2}{\rho_u\sum_{\ell\in\mathcal{N}\setminus k}\zeta_\ell\big(\sum_{m\in\mathcal{M}}\sigma_{mk}^2\frac{\beta_{m\ell}}{\beta_{mk}}\big)^2|\boldsymbol{\varphi}_k^H\boldsymbol{\varphi}_\ell|^2+\rho_u\sum_{\ell\in\mathcal{N}}\zeta_\ell\sum_{m\in\mathcal{M}}\sigma_{mk}^2\beta_{m\ell}+\sum_{m\in\mathcal{M}}\sigma_{mk}^2}\right) \tag{16}$$

The transmission time of Step (S3) consists of the transmission time of transmitting the global UL training update from the UEs to the APs, and that from the APs to the CPU, i.e.,

$$T_u(\boldsymbol{a},\boldsymbol{R}_u)=\underbrace{\max_{k\in\mathcal{N}}\frac{a_kb_kS_u}{R_{u,k}}}_{\text{Straggler effect in Step (S3)}}+\frac{\sum_{k\in\mathcal{N}}a_kb_kS_u}{\sum_{k\in\mathcal{N}}a_kb_kR_{u,k}}. \tag{18}$$

Therefore, the transmission time of one iteration of the FL process is

$$T_o(\boldsymbol{a},\boldsymbol{R}_d,\boldsymbol{R}_u)=T_d(\boldsymbol{a},\boldsymbol{R}_d)+T_u(\boldsymbol{a},\boldsymbol{R}_u). \tag{19}$$

In (17) and (18), the terms of straggler effect can be much larger than the remaining terms, when there are UEs that have highly unfavorable links in a network having a large number of UEs. Therefore, the straggler effect could significantly affect the transmission time of each iteration of the FL process in (19).

On the other hand, as seen from (19), $T_o(\boldsymbol{a},\boldsymbol{R}_d,\boldsymbol{R}_u)$ depends on both $\boldsymbol{a}$ (UE selection) and $(\boldsymbol{R}_d,\boldsymbol{R}_u)$ (rate allocation). However, the UEs are selected before any FL process is executed, while the rates are optimized before each iteration of the FL process happens. Therefore, to measure how efficiently the transmission time is optimized, we introduce a new metric termed "ergodic or effective transmission time of one iteration of an FL process", i.e., $T_e \triangleq \mathbb{E}\{T_o(\boldsymbol{a},\boldsymbol{R}_d,\boldsymbol{R}_u)\}$. Since each iteration of the FL process happens in one large-scale coherence time [7], $\mathbb{E}\{T_o(\boldsymbol{a},\boldsymbol{R}_d,\boldsymbol{R}_u)\}$ is, therefore, the average of $T_o(\boldsymbol{a},\boldsymbol{R}_d,\boldsymbol{R}_u)$ over the large-scale fading and user sampling realizations.

Before any FL process is executed, UEs are selected to mitigate the straggler effect by solving an optimization problem that minimizes the effective transmission time of one iteration of an FL process as

$$\min_{\boldsymbol{a},\boldsymbol{\eta},\boldsymbol{\zeta},\boldsymbol{R}_d,\boldsymbol{R}_u} T_e(\boldsymbol{a},\boldsymbol{R}_d,\boldsymbol{R}_u)=\mathbb{E}\{T_o(\boldsymbol{a},\boldsymbol{R}_d,\boldsymbol{R}_u)\} \tag{20a}$$

$$\text{s.t. } (1),(5),(6),(7),(11),(12),(15)$$

$$0\le\eta_{mk},\forall m,k \tag{20b}$$

$$0\le\zeta_k,\forall k \tag{20c}$$

$$0\le R_{d,k},\forall k \tag{20d}$$

$$0\le R_{u,k},\forall k \tag{20e}$$

$$\sum_{k\in\mathcal{N}}a_k\ge N_{\text{QoL}}, \tag{20f}$$

where $N_{\text{QoL}}\ge K$ is a threshold to ensure the *quality of learning*. The quality of a statistical learning scheme such as FL is defined as the test accuracy obtained after running that scheme. It can be shown in [9] that the test accuracy of FL becomes worse if a number of UEs that are selected to participate in an FL process decreases. In (20f), the quality of FL is thus guaranteed by keeping the number of UEs participating in an FL process to be larger than a certain value N_{QoL}. In practice, N_{QoL} is experimentally chosen according to specific ML models. Problem (20) is challenging because of its nonconvex stochastic nature, mixed-integer mixed-timescale structure, binary constraints and tight coupling among the variables.

IV. Proposed Algorithm

First, we observe that $x\in\{0,1\}\Leftrightarrow x\in[0,1]\,\&\,x-x^2\le 0$ [10]. Therefore, problem (20) is equivalent to

$$\min_{\boldsymbol{a},\boldsymbol{\eta},\boldsymbol{\zeta},\boldsymbol{R}_d,\boldsymbol{R}_u} \mathbb{E}\{T_o(\boldsymbol{a},\boldsymbol{R}_d,\boldsymbol{R}_u)\} \tag{21a}$$

$$\text{s.t. } (5),(6),(7),(11),(12),(15),(20b)-(20f)$$

$$\sum_{k\in\mathcal{N}}(a_k-a_k^2)\le 0 \tag{21b}$$

$$0\le a_k\le 1,\forall k. \tag{21c}$$

We then rewrite (21) in an epigraph form as

$$\min_{\boldsymbol{x}} \mathbb{E}\{\widetilde{T}_o(\boldsymbol{a},\boldsymbol{R}_d,\boldsymbol{R}_u,t_d,t_u)\} \tag{22a}$$

$$\text{s.t. } (5),(6),(7),(11),(12),(15),(20b)-(20f),(21b),(21c)$$

$$\frac{a_kS_d}{R_{d,k}}\le t_d,\forall k \tag{22b}$$

$$\frac{a_kb_kS_u}{R_{u,k}}\le t_u,\forall k, \tag{22c}$$

where $\widetilde{T}_o=\frac{\sum_{k\in\mathcal{N}}a_kS_d}{\sum_{k\in\mathcal{N}}a_kR_{d,k}}+t_d+t_u+\frac{\sum_{k\in\mathcal{N}}a_kb_kS_u}{\sum_{k\in\mathcal{N}}a_kb_kR_{u,k}}$, $\boldsymbol{x}\triangleq\{\boldsymbol{a},\boldsymbol{\eta},\boldsymbol{\zeta},\boldsymbol{R}_d,\boldsymbol{R}_u,t_d,t_u\}$; t_d and t_u are additional variables. (22) can be decomposed into a family of short-term subproblems and a long-term master problem as follows.

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

$$\min_{\widetilde{\boldsymbol{x}}} \widetilde{T}_o(\boldsymbol{R}_d,\boldsymbol{R}_u,t_d,t_u) \tag{23}$$

$$\text{s.t. } (5),(6),(7),(11),(12),(15),(20b)-(20e),(22b),(22c),$$

where $\widetilde{\boldsymbol{x}}\triangleq\boldsymbol{x}\setminus\boldsymbol{a}$. For given optimal solutions $\widetilde{\boldsymbol{x}}$ to problems (23), the *long-term master problem* is expressed as:

$$\min_{\boldsymbol{a}} \hat{g}(\boldsymbol{a})\triangleq\mathbb{E}\{\widetilde{T}_o(\boldsymbol{a})\} \tag{24}$$

$$\text{s.t. } (20f),(21b),(21c),$$

where $\widetilde{T}_o(\boldsymbol{a})$ is rewritten as $\widetilde{T}_o(\boldsymbol{a})=\frac{\boldsymbol{a}^T\boldsymbol{S}_d}{\boldsymbol{a}^T\boldsymbol{R}_d}+t_d+t_u+\frac{\boldsymbol{a}^T\widetilde{\boldsymbol{S}}_u}{\boldsymbol{a}^T\widetilde{\boldsymbol{R}}_u}$, $\boldsymbol{S}_d\in\mathbb{R}^N$ is a vector whose elements are S_d, $\widetilde{\boldsymbol{S}}_u\in\mathbb{R}^N$ is a vector whose k-th element is b_kS_u, and $\widetilde{\boldsymbol{R}}_u\in\mathbb{R}^N$ is a vector whose k-th element is $b_kR_{u,k}$.

1) Solving the Short-term Subproblem (23)*:* Problem (23) can be rewritten as

$$\min_{\widehat{\boldsymbol{x}}} \widetilde{T}_o(\boldsymbol{R}_d,\boldsymbol{R}_u,t_d,t_u) \tag{25a}$$

$$\text{s.t. } (7),(15),(20b)-(20e),(22b),(22c),$$

$$\sigma_{mk}^2\eta_{mk}\le\tilde{v}_{mk},\forall m,k \tag{25b}$$

$$\tilde{v}_{mk}\le a_k,\forall m,k \tag{25c}$$

$$\sum_{k\in\mathcal{N}}\tilde{v}_{mk}\le 1,\forall m \tag{25d}$$

$$\zeta_k\le a_kb_k,\forall k, \tag{25e}$$

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, (25b)-(25d) follow from (5) and (6); (25e) follows from (11) and (12). If 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 $v_{mk}\triangleq\eta_{mk}^{1/2},\forall m,k$, and $u_k\triangleq\zeta_k^{1/2},\forall k$, then (25) can be rewritten as:

$$\min_{\widehat{\boldsymbol{x}}} \widetilde{T}_o(\boldsymbol{R}_d,\boldsymbol{R}_u,t_d,t_u) \tag{26a}$$

$$\text{s.t. } (22b),(22c),(25c),(25d)$$

Algorithm 1 Solving the short-term subproblem (23)

1: **Initialize**: Set $\kappa=1$ and choose a random point $\widetilde{\boldsymbol{x}}^{(0)} \in \mathcal{F}$.
2: **repeat**
3: Update $\kappa = \kappa + 1$
4: Solving (31) to get its optimal solution $\widetilde{\boldsymbol{x}}^*$
5: Update $\widetilde{\boldsymbol{x}}^{(\kappa)} = \widetilde{\boldsymbol{x}}^*$
6: **until** convergence

Output: $(\boldsymbol{\eta}^*, \boldsymbol{\zeta}^*, \boldsymbol{f}^*, \boldsymbol{R}_d^*, \boldsymbol{R}_u^*)$

$$\sigma_{mk}^2 v_{mk}^2 \le \tilde{v}_{mk}, \forall m,k \tag{26b}$$
$$0 \le v_{mk}, \forall m,k \tag{26c}$$
$$u_k^2 \le a_k b_k, \forall m,k \tag{26d}$$
$$0 \le u_k \le 1, \forall k \tag{26e}$$
$$0 \le R_{d,k} \le h_{d,k}(\boldsymbol{v}), \forall k \tag{26f}$$
$$0 \le R_{u,k} \le h_{u,k}(\boldsymbol{u}), \forall k. \tag{26g}$$

where $\bar{\boldsymbol{x}} \triangleq \{\widehat{\boldsymbol{x}}, \boldsymbol{v}, \boldsymbol{u}\} \setminus \{\boldsymbol{\eta}, \boldsymbol{\zeta}\}$.

Regarding the nonconvex constraints (26f) and (26g), the concave lower bound $\widetilde{h}_{d,k}(\boldsymbol{v})$ of $h_{d,k}(\boldsymbol{v})$ is given by [7]

$$\widetilde{h}_{d,k}(\boldsymbol{v}) \triangleq \log_2\Big(1+\frac{(\Upsilon_k^{(\kappa)})^2}{\Pi_k^{(\kappa)}}\Big) - \frac{(\Upsilon_k^{(\kappa)})^2}{\Pi_k^{(\kappa)}} + 2\frac{\Upsilon_k^{(\kappa)}\Upsilon_k}{\Pi_k^{(\kappa)}} - \frac{(\Upsilon_k^{(\kappa)})^2(\Upsilon_k^2+\Pi_k)}{\Pi_k^{(\kappa)}((\Upsilon_k^{(\kappa)})^2+\Pi_k^{(\kappa)})} \le h_{d,k}(\boldsymbol{v}), \tag{27}$$

where $\Pi_k(\boldsymbol{v}) = \rho_d \sum_{\ell\in\mathcal{N}\setminus k}\big(\sum_{m\in\mathcal{M}} v_{m\ell}\sigma_{m\ell}^2 \frac{\beta_{mk}}{\beta_{m\ell}}\big)^2 |\boldsymbol{\varphi}_\ell^H \boldsymbol{\varphi}_k|^2 + \rho_d \sum_{\ell\in\mathcal{N}}\sum_{m\in\mathcal{M}} v_{m\ell}^2\sigma_{m\ell}^2\beta_{mk} + 1$, and $\Upsilon_k(\{v_{mk}\}_{m\in\mathcal{M}}) = \sqrt{\rho_d}\sum_{m\in\mathcal{M}} v_{mk}\sigma_{mk}^2$. Similarly, the concave lower bound $\widetilde{h}_{u,k}(\boldsymbol{u})$ of $h_{u,k}(\boldsymbol{u})$ is

$$\widetilde{h}_{u,k}(\boldsymbol{u}) \triangleq \log_2\Big(1+\frac{(\Psi_k^{(\kappa)})^2}{\Xi_k^{(\kappa)}}\Big) - \frac{(\Psi_k^{(\kappa)})^2}{\Xi_k^{(\kappa)}} + 2\frac{\Psi_k^{(\kappa)}\Psi_k}{\Xi_k^{(\kappa)}} - \frac{(\Psi_k^{(\kappa)})^2(\Psi_k^2+\Xi_k)}{\Xi_k^{(\kappa)}((\Psi_k^{(\kappa)})^2+\Xi_k^{(\kappa)})} \le h_{u,k}(\boldsymbol{u}), \tag{28}$$

where $\Xi_k(\boldsymbol{u}) = \rho_u \sum_{\ell\in\mathcal{N}\setminus k} u_\ell^2 \big(\sum_{m\in\mathcal{M}} \sigma_{mk}^2 \frac{\beta_{m\ell}}{\beta_{mk}}\big)^2 |\boldsymbol{\varphi}_k^H \boldsymbol{\varphi}_\ell|^2 + \rho_u \sum_{\ell\in\mathcal{N}} u_\ell^2 \sum_{m\in\mathcal{M}} \sigma_{mk}^2\beta_{m\ell} + \sum_{m\in\mathcal{M}}\sigma_{mk}^2$, and $\Psi_k(u_k) = \rho_u^{1/2} u_k(\sum_{m\in\mathcal{M}}\sigma_{mk}^2)$. As such, (26f) and (26g) can be respectively approximated by

$$R_{d,k} \le \widetilde{h}_{d,k}(\boldsymbol{v}), \forall k \in \mathcal{N} \tag{29}$$
$$R_{u,k} \le \widetilde{h}_{u,k}(\boldsymbol{u}), \forall k \in \mathcal{N}. \tag{30}$$

At the iteration $\kappa+1$, for a given point $\bar{\boldsymbol{x}}^{(\kappa)}$, problem (26) (hence (23)) can finally be approximated by the following convex problem:

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

where $\widetilde{\mathcal{F}} \triangleq \{(22b), (22c), (25c), (25d), (26b)–(26e), (29), (30)\}$ is a convex feasible set.

In Algorithm 1, we outline the main steps to solve problem (23). Let $\mathcal{F} \triangleq \{(22b), (22c), (25c), (25d), (26b)-(26g)\}$ be the feasible set of (26). Starting from a random point $\bar{\boldsymbol{x}} \in \mathcal{F}$, we solve (31) 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. The converged solution of Algorithm 1 will fulfill the KKT conditions of the main problem (23). The proof of this convergence property follows [11], and hence, omitted.

2) Solving the Long-Term Master Problem (24)*:* Given solution $\widetilde{\boldsymbol{x}}$ to short-term subproblems (23), we have $t_d = \frac{a_{k^*}S_d}{R_{d,k^*}}$, where $k^* \triangleq \underset{k\in\mathcal{N}}{\operatorname{argmax}} \frac{a_k S_d}{R_{d,k}}$. Therefore, we can have $t_d = \boldsymbol{a}^T\tilde{\boldsymbol{t}}_d$, 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 $\frac{S_d}{R_{d,k^*}}$. Similarly, $t_u = \frac{a_{j^*}b_{j^*}S_u}{R_{u,j^*}}$ with $j^* \triangleq \underset{k\in\mathcal{N}}{\operatorname{argmax}} \frac{a_k b_k S_u}{R_{u,k}}$ and $b_{j^*}=1$. It can be rewritten as $t_u = \boldsymbol{a}^T\tilde{\boldsymbol{t}}_u$, where $\tilde{\boldsymbol{t}}_u$ is the vector whose elements are 0 except for the j^*-th element, and the value of this element is $\frac{S_u}{R_{u,j^*}}$. Now, the long-term problem (24) is equivalent to

$$\min_{\boldsymbol{a}} g(\boldsymbol{a}) \triangleq \mathbb{E}\{\frac{\boldsymbol{a}^T\boldsymbol{S}_d}{\boldsymbol{a}^T\boldsymbol{R}_d} + \boldsymbol{a}^T\tilde{\boldsymbol{t}}_d + \boldsymbol{a}^T\tilde{\boldsymbol{t}}_u + \frac{\boldsymbol{a}^T\widetilde{\boldsymbol{S}}_u}{\boldsymbol{a}^T\widetilde{\boldsymbol{R}}_u}\} \tag{32}$$
$$\text{s.t. } (20f), (21b), (21c).$$

Let $V(\boldsymbol{a}) \triangleq \sum_{k\in\mathcal{N}}(a_k - a_k^2) = \boldsymbol{a}^T(\boldsymbol{1}-\boldsymbol{a})$, then (21b) becomes $V(\boldsymbol{a}) \le 0$. We now consider the problem

$$\min_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a},\lambda) \triangleq g(\boldsymbol{a}) + \lambda V(\boldsymbol{a}) \tag{33}$$
$$\text{s.t. } (20f), (21c),$$

where $\mathcal{L}(\boldsymbol{a},\lambda)$ is the Lagrangian of (32), λ is the Lagrangian multiplier corresponding to (21b), and $\boldsymbol{1} \in \mathbb{R}^N$ is an all-one vector. Let $\mathcal{H} \triangleq \{(20f), (21c)\}$ be the feasible set of (24).

Proposition 1. The following statement holds:

(i) The value of V_λ at the solution of (24) corresponding to λ is decreasing to 0 as $\lambda \to +\infty$.
(ii) Problem (33) has the following property
$$\min_{\boldsymbol{a}\in\mathcal{H}} g(\boldsymbol{a}) = \sup_{\lambda\ge 0} \min_{\boldsymbol{a}\in\widehat{\mathcal{H}}} \mathcal{L}(\boldsymbol{a},\lambda), \tag{34}$$
and is therefore equivalent to (32) at the optimal solution $\lambda^* \ge 0$ of the sup-min problem in (34).

The proof of Proposition 1 is rather standard, and follows from [10, Proposition 1]. Theoretically, it is required to have $V_\lambda = 0$ in order to obtain an optimal λ^*. According to Proposition 1, 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 $\varepsilon = 0.001$, we see that $\lambda = 1$ is enough to ensure $V_\lambda \le \varepsilon$. Note that this way of choosing λ has been widely used in the literature, e.g., [10], [12].

At the large-scale coherence time or iteration $n+1$, problem (24) is approximated by the following convex problem:

$$\min_{\boldsymbol{a}\in\mathcal{H}} \bar{\mathcal{L}}(\boldsymbol{a}), \tag{35}$$

where $\bar{\mathcal{L}}(\boldsymbol{a})$ is a surrogate function of $\mathcal{L}(\boldsymbol{a})$, and defined as $\bar{\mathcal{L}}(\boldsymbol{a}) \triangleq \mathcal{L}^{(n+1)} + ((\nabla\mathcal{L})^{(n+1)})^T(\boldsymbol{a}-\boldsymbol{a}^{(n+1)}) + \tau||\boldsymbol{a}-\boldsymbol{a}^{(n+1)}||^2$, $\mathcal{L}^{(n+1)} = \bar{g}^{(n+1)} + \lambda V^{(n+1)}$, $\bar{g}^{(n+1)} = (1-\phi^{(n+1)})\bar{g}^{(n)} + \phi^{(n+1)}T^{(n+1)}$, $(\nabla\mathcal{L})^{(n+1)} = (\nabla\bar{g})^{(n+1)} + \lambda(\nabla V)^{(n+1)}$, and $(\nabla\bar{g})^{(n+1)} = (1-\phi^{(n+1)})(\nabla\bar{g})^{(n)} + \phi^{(n+1)}(\nabla T)^{(n+1)}$. Here, $\bar{g}^{(0)} = 0$, $(\nabla\bar{g})^{(0)} = \boldsymbol{0}$, $\phi^{(n+1)}$ is a weighting parameter,

$$(\nabla T)^{(n+1)} = \frac{\boldsymbol{S}_d((\boldsymbol{a}^{(n+1)})^T\boldsymbol{R}_d) - \boldsymbol{R}_d((\boldsymbol{a}^{(n+1)})^T\boldsymbol{S}_d)}{((\boldsymbol{a}^{(n+1)})^T\boldsymbol{R}_d)^2} + \frac{\widetilde{\boldsymbol{S}}_u((\boldsymbol{a}^{(n+1)})^T\widetilde{\boldsymbol{R}}_u) - \widetilde{\boldsymbol{R}}_d((\boldsymbol{a}^{(n+1)})^T\widetilde{\boldsymbol{S}}_u)}{((\boldsymbol{a}^{(n+1)})^T\widetilde{\boldsymbol{R}}_u)^2} + \tilde{\boldsymbol{t}}_d + \tilde{\boldsymbol{t}}_u,$$

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

3) Solving the Overall Problem (20)*:* Algorithm 2 outlines the main steps to solve the overall problem (20). In the large-scale coherence time n, for a given random value of $\boldsymbol{a}^{(n+1)} \in \mathcal{H}$, the set $\widetilde{\mathcal{N}}^{(n+1)}$ of the selected UEs is constructed by (3). The index set $\mathcal{S}^{(n+1)}$ of sampled UEs in $\widetilde{\mathcal{N}}^{(n+1)}$ is chosen by (10). The short-term subproblem (23) 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 (33). After solving (33) to obtain an optimal solution $(\boldsymbol{a}^*)^{(n+1)}$,

Algorithm 2 UE selection to mitigate the straggler effect for FL in CFmMIMO networks

1: **Initialize**: Set $n=0$, select a random $\boldsymbol{a}^{(n+1)} \in \mathcal{H}$
2: **repeat**
3: Update $\widetilde{\mathcal{N}}^{(n+1)}$ by (3), choose $\mathcal{S}^{(n+1)}$ from $\widetilde{\mathcal{N}}$ by (10)
4: Solve the short-term subproblem (23) to obtain its optimal solution $(\boldsymbol{\eta}^*, \boldsymbol{\zeta}^*, \boldsymbol{R}_d^*, \boldsymbol{R}_u^*)$ by using Algorithm 1, and update $(\boldsymbol{\eta}^{(n+1)}, \boldsymbol{\zeta}^{(n+1)}, \boldsymbol{R}_d^{(n+1)}, \boldsymbol{R}_u^{(n+1)}) = (\boldsymbol{\eta}^*, \boldsymbol{\zeta}^*, \boldsymbol{R}_d^*, \boldsymbol{R}_u^*)$
5: Solve the approximate long-term master problem (35) to obtain its optimal solution $(\boldsymbol{a}^*)^{(n+1)}$
6: Update $\boldsymbol{a}^{(n+2)}$ by (36)
7: Update $n = n+1$
8: **until** convergence

Output: $\boldsymbol{a}^* = \boldsymbol{a}^{(n+1)}$

we update $\boldsymbol{a}^{(n+2)}$ as

$$a_k^{(n+2)} = (1-\pi^{(n+1)})a_k^{(n+1)} + \pi^{(n+1)}(a_k^*)^{(n+1)}, \forall k, \quad (36)$$

where $\pi^{(n+1)}$ is a weighting parameter; $\{\phi^{(n)}, \pi^{(n)}\}$ is chosen to satisfy the following conditions [13, Assumption 5].

(A1) $\phi^{(n)} \to 0$, $\frac{1}{\phi^{(n)}} \leq \mathcal{O}(n^{\varsigma})$, $\varsigma \in (0,1)$, and $\sum_n (\phi^{(n)})^2 < \infty$;

(A2) $\pi^{(n)} \to 0$, $\sum_n \pi^{(n)} = \infty$, $\sum_n (\pi^{(n)})^2 < \infty$, and $\lim_{n\to\infty} \frac{\pi^{(n)}}{\phi^{(n)}} = 0$.

Once Algorithm 2 converges, the FL process is then executed using the solution $\boldsymbol{a}$ obtained by Algorithm 2.

Definition 1. *A solution $(\boldsymbol{a}^*, \boldsymbol{x}^*)$ is called a stationary solution of problem* (21) *(or* (20)*) if $\boldsymbol{x}^*$ is a KKT solution of the short-term subproblem* (23) *for $\boldsymbol{a} = \boldsymbol{a}^*$, and $\boldsymbol{a}^*$ is a KKT solution of the long-term master problem* (24) *for $\boldsymbol{x} = \boldsymbol{x}^*$.*

As discussed in Section IV-1, the solution $\boldsymbol{x}^*$ obtained from Algorithm 2 is a KKT solution of the short-term subproblem (23). By a similar argument as in [13], the solution $\boldsymbol{a}^{(n+1)}$ obtained from Algorithm 2 is proved to be a KKT solution of the long-term master problem (24). 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., $I_S^{(n)} \to \infty$, $I_L \to \infty$. In practice, it is acceptable to choose finite $\{I_S^{(n)}\}_{n\in\{1,\ldots,I_L\}}$ and I_L for an approximate convergence. Therefore, Algorithm 2 is guaranteed to converge to the neighbourhood of the stationary solutions of problem (20).

V. Numerical Examples

A. Network Setup and Parameter Setting

Consider a CFmMIMO network where the APs and UEs are randomly located in a square of $D \times D$ km^2. The locations of APs and UEs are generated with the following practical properties:

(P1) The UEs are more likely to stay close to some local points.
(P2) Some local areas attract more UEs than other local areas.
(P3) The APs are more likely to locate close to some local points.
(P4) Some local areas have more APs than other local areas.

We adopt the method in [14] to model the property (P1). First, let Φ be a homogeneous Poisson Point Process (PPP) in Q with a density μ, and ϕ be the number of the Poisson Points (PPs) in Φ. Denote by $\mathcal{V} \triangleq \{\mathcal{V}_1, \ldots, \mathcal{V}_\phi\}$ the set of all Voronoi cells that are generated from these PPs. Φ is then thinned by retaining points in Φ independently with probability p and removing the rest. The thinned version Φ_p of Φ models the local points that attract UEs and the Voronoi cells corresponding to the PPs of Φ_p models their local areas. Let $\mathcal{I}$ be the set of indices of the PPs retained in Φ. Here, the UEs are then uniformly distributed in the Voronoi cells of these PPs, i.e., $\{\mathcal{V}_i\}_{i\in\mathcal{I}}$. Since the Voronoi cells of Φ_p is larger than those of the retained points of Φ, the UEs are thus pushed towards the interior of the local areas, which captures the property (P1). Here, the larger thinning probability p implies the higher probability of a UE in a Voronoi cell close to its local point.

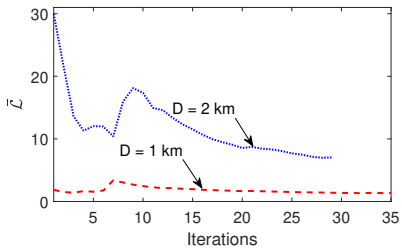


Fig. 1. The convergence of Algorithm 2 ($M=10, N=6, N_{\text{QoL}}=3, K=3$).

To capture the property (P2), we set a probability p_i for the local area $i \in \mathcal{I}$ that is chosen to come by each UE. We assume that $\{p_i\}_{i\in\mathcal{I}}$ are the same for all the UEs. In each realization, we randomly choose a set $\{p_i\}_{i\in\mathcal{I}}$ such that $\sum_{i\in\mathcal{I}} p_i = 1$ and $\max_{i\in\mathcal{I}} p_i - \min_{i\in\mathcal{I}} p_i = \Delta$, where Δ controls the difference in the attraction of the local areas. Each UE $k \in \mathcal{N}$ is then selected to the local area i with probabilities $\{p_i\}_{i\in\mathcal{I}}$. Finally, the selected UEs in each local area $i \in \mathcal{I}$ is uniformly distributed in the corresponding Voronoi cells $\mathcal{V}_i$. To capture the properties (P3) and (P4), the APs are non-uniformly distributed using the same method that captures the properties (P1) and (P2).

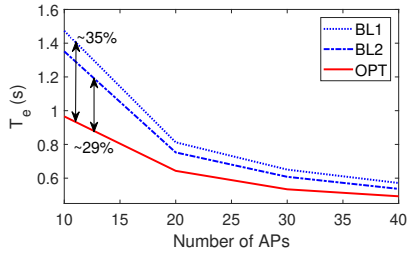
In each network realization, the locations of APs are fixed and those of the UEs change over the iterations of the FL process. Since each iteration of the FL process happens in one large-scale fading coherence time (in the order of seconds), the total running time of an FL process is expected to be around several minutes. Therefore, we assume that the UEs only move around their current local areas during the FL process. Here, in each iteration of the FL process of each network setup realization, the locations of UEs are uniformly distributed in the Voronoi cells that those UEs belong to.

We set $\tau_c = 200$ samples. The large-scale fading coefficients, e.g., β_{mk}, are modeled in the same manner as [15, (37), (38)]. To estimate channels, a random pilot assignment is used as in [7]. We choose a thinning probability $p = 0.3$, $\Delta = 1/4$ for UEs and $\Delta = 1/10$ for APs, $\tau_t = 10$, $S_d = S_u = 0.5$ MB, and noise power $\sigma_0^2 = -92$ dBm. Let $\tilde{\rho}_d = 1$ W, $\tilde{\rho}_u = 0.2$ W and $\tilde{\rho}_t = 0.2$ W be the maximum transmit power of the APs, UEs and UL pilot sequences, respectively. The maximum transmit powers ρ_d, ρ_u and ρ_t are normalized by the noise power. We set $\pi^{(n)} = \frac{100}{100+n}$ and $\phi^{(n)} = \frac{1}{n^{9/10}}$ which satisfy conditions (A1) and (A2) in Section IV-3.

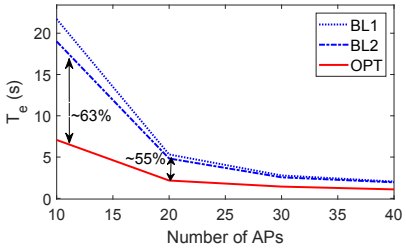
B. Results and Discussions

First, we evaluate the convergence behavior of the proposed Algorithm 2. As seen from Fig. 1 with an arbitrary network realization, Algorithm 2 converges in around 30 iterations.

To further evaluate the effectiveness of Algorithm 2, we consider the following baseline schemes:



(a) Case (C2) with $D = 1$ km



(b) Case (C2) with $D = 2$ km

Fig. 2. Comparison among the proposed approach and baselines ($N = 6$, $N_{\text{QoL}} = 3$ and $K = 3$).

- Baseline 1 (BL1): UE selection is not optimized. The transmitted powers and rates of Steps (S1) and (S3) for all N original UEs are optimized by using a slightly modified version of Algorithm 1.
- Baseline 2 (BL2): This baseline is similar to BL1 except that the UEs are selected by randomly choosing $\hat{N}$ UEs from the original N UEs, where $\hat{N} \in [N_{\text{QoL}}, N-1]$ is a random number.

In each network realization, the effective transmission times T_e obtained by baselines (BL1) and (BL2) are the average times over the large-scale fading and user sampling realizations. For ease of presentation, our "optimal" UE selection approach is denoted by "OPT".

Fig. 2 shows the comparison among the considered schemes in terms of the effective transmission time T_e of each iteration of an FL process. As seen, OPT gives the best performance. In particular, while BL1 and BL2 achieve nearly the same performance, OPT provides substantial time reductions over these schemes, e.g., 63% with $M = 10$ and $D = 2$ km.

The figure also shows the importance of UE selection in reducing the FL training time, especially in the networks that have a moderately low density of APs, i.e., having a large value of D and a small/moderate number of APs. This is because the straggler effect becomes serious in such these cases. Specifically, compared to the transmission time T_e obtained by BL1, the amount of time reduction by OPT with $M = 10$ and $D = 2$ km is approximately twice that with $D = 1$ km. The amount of time reduction by OPT also increases when the number of APs decreases. This is because the APs are located close to some local points which may be far from the UE locations. When D is large, there are more UEs that have unfavorable links for a larger area. Moreover, due to the smaller array gain, the data rates of UEs decrease when the number of APs decreases.

VI. Conclusion

This work has proposed an UE selection approach to mitigate the straggler effect for FL in CFmMIMO. Targeting the transmission time minimization for the general FL framework with UE sampling [2], [3], we have jointly designed UE selection, power control, and data rate under practical requirements on the maximum transmit powers of APs and UEs, and the minimum number of UEs to guarantee the quality of learning. A mixed-integer mixed-timescale stochastic nonconvex problems were formulated with the objectives of minimizing the transmission time of each iteration of an FL process. Utilizing online successive convex approximation techniques, we have successfully developed a novel algorithm to solve the formulated problem. The proposed algorithm has been proved to converge to the neighbourhood of stationary points. Numerical results have showed that our UE selection approach significantly reduces the FL transmission time over the baselines under comparison, especially in networks with moderately low AP density.

Acknowledgment

This work is supported in part by an ECR-HDR scholarship from The University of Newcastle, in part by the Australian Research Council Discovery Project grant DP170100939, in part by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant numbers 102.02-2018.320 and 102.02-2019.321, in part by the U.K. Research and Innovation Future Leaders Fellowships under Grant MR/S017666/1, and in part by Vietnam National University HoChiMinh City (VNU-HCM) under grant number DS2020-28-01.

References

[1] 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, 2020.

[2] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on Non-IID data," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020.

[3] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," 2019. [Online]. Available: https://arxiv.org/abs/1910.14425

[4] W. Shi, S. Zhou, and Z. Niu, "Device scheduling with fast convergence for wireless federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[5] 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.

[6] T. T. Vu, D. T. Ngo, H. Q. Ngo, M. N. Dao, N. H. Tran, and R. H. Middleton, "User selection approaches to mitigate the straggler effect for federated learning on cell-free massive MIMO networks," 2020. [Online]. Available: https://arxiv.org/abs/2009.02031

[7] 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, 2020.

[8] 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.

[9] 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, 2020.

[10] 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.

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

[12] 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.

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

[14] H. S. Dhillon, R. K. Ganti, and J. G. Andrews, "Modeling non-uniform UE distributions in downlink cellular networks," *IEEE Wireless Commun. Lett.*, vol. 2, no. 3, pp. 339–342, Jun. 2013.

[15] 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.