Federated Learning for Massive MIMO Power Allocation

Yushu Yan*, Xiangyu Chang*, Basak Guler*,
Amit K. Roy-Chowdhury*, Srikanth V. Krishnamurthy*, Ananthram Swami**
* University of California, Riverside, USA ** DEVCOM Army Research Laboratory, USA
{yyan064@, cxian008@, bguler@ece., amitrc@ece., krish@cs.}ucr.edu, ananthram.swami.civ@army.mil

***Abstract*—This work proposes the use of federated learning (FL) for deep learning-based power allocation in massive MIMO systems. The conventional approach for learning-based power allocation builds on centralized training, where training data is collected at a single node which performs training. On the other hand, doing so requires extensive data exchange and processing at the central node, leading to high communication and computational overhead. In contrast, FL enables distributed training without the need to centralize training data, allowing for parallelization of the training load (hence speeding up training), as well as efficient use of the available communication resources. In this work we study FL for massive MIMO power allocation with heterogeneity-aware user sampling, and show that FL-based power allocation can achieve performance comparable to that of the centralized approach while significantly reducing communication overhead. Our experiments demonstrate the effectiveness of the proposed distributed learning strategies in terms of both convergence rate and spectral efficiency.**

I. INTRODUCTION

Massive MIMO is a promising solution for improving spectral efficiency in future wireless networks, through the use of a large number of antennas at the base station to support multiple user equipments (UEs) [1] [2]. Massive MIMO has been studied extensively in recent years, most notably due to two key properties: 1) channel hardening, allowing one to rely on large scale fading for design specifications, and 2) favorable propagation, allowing the use of simple beamforming methods such as maximum ratio combining. A major challenge in the design of scalable massive MIMO frameworks is developing fast and efficient power allocation strategies [3] [4], as the number of UEs increases in the system.

To that end, deep learning based power allocation schemes have been proposed recently to automate power allocation decisions [5] [6]. In [5] and [6], UE positions and large-scale fading information have been shown to be sufficient for deep learning based power allocation, respectively. In doing so, a centralized supervised learning strategy is proposed in [5] for automating downlink power allocation, in which a fully-connected neural network is trained by using UE positions as the input features, and the output corresponds to the power allocation policy. Similarly, a deep neural network is employed in [6], to learn the unknown mapping between large-scale fading coefficients and optimal power allocation for cell-free massive MIMO. Both centralized and decentralized supervised learning strategies have been explored; in the latte, each party performs training locally, without cooperation across the parties. It has been observed that there is a substantial performance gap between centralized and distributed approaches [6].

Current literature assumes that a single party holds the entire dataset for training, which may hinder applicability in real-world settings. In such settings, the data samples are typically collected by multiple parties, and collecting the dataset at a single party may incur high communication cost. In addition, during dataset collection, different parties may be located at different geographical environments, such as rural vs. urban areas, in which the UE distributions, and accordingly, power allocation strategies may be different. This will cause heterogeneity among local datasets at different parties.

To address the above challenges, this work explores the benefits of cooperation across distributed parties during training, to improve the predictive performance of the trained model, by leveraging FL principles [7]. Our goal is to bridge the gap between fully centralized and decentralized (non-cooperative) training paradigms, by leveraging cooperation across distributed parties during training, while avoiding the substantial communication and computational overhead of fully centralized training. Traditional FL, e.g., federated averaging (FedAvg) algorithm [7], selects a subset of parties uniformly at random during training, which can slow down convergence in heterogeneous settings. In order to speed-up the model convergence of FL and improve model performance, we leverage a contextual multi-armed bandit (MAB) mechanism for user selection, without any prior knowledge about the heterogeneity patterns of the datasets.

II. SYSTEM MODEL

In this work, we consider downlink communication in a hierarchical massive MIMO network as illustrated in Fig. 1. We assume that the network is supported by N edge processors (EPs), connected to a central processor. Each EP covers an area of L cells, and each cell is supported by an M-antenna access point (AP). For simplicity, we use the same index for a cell and its corresponding AP. Each AP $j \in \{1, \dots, L\}$ communicates with K_j single-antenna UEs within its cell. The

Research was sponsored in part by the OUSD(R&E)/RT&L under Cooperative Agreement Number W911NF-20-2-0267, NSF CAREER Award CCF-2144927, NSF CNS grant 2106982, and the UCR OASIS Funding Award. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ARL and USD(R&E)/RT&L or the U.S. Government.

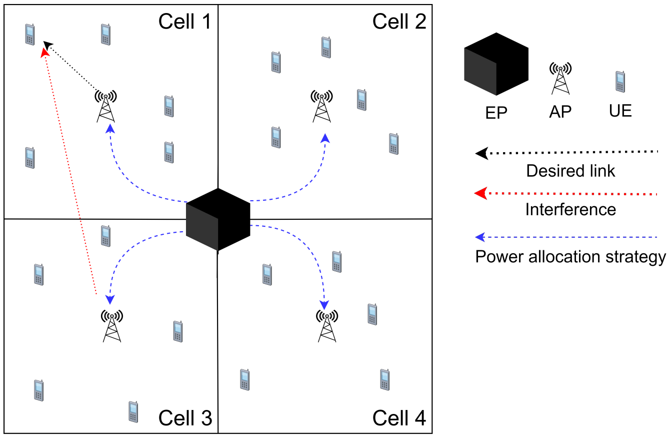

Fig. 1. System model

UEs within a given cell are subject to the interference caused by the neighboring APs (served by the same EP). We further assume that EPs are located sufficiently distant from each other, and the interference between the APs belonging to the cells served by different EPs is negligible. There are K UEs in each EP, i.e., $\sum_{j=1}^{L} K_j = K$. The distribution of UEs can be different across the EPs, representing heterogeneity across different environments.

We let $\mathbf{h}^j_{jk_j}$ denote the channel gain between the j-th AP and k_j-th UE in cell j, where $\mathbf{h}^j_{jk_j} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}_M, \mathbf{R}^j_{jk_j})$ such that $\mathbf{R}^j_{jk_j} \in \mathbb{C}^{M\times M}$ is the spatial correlation matrix. The normalized trace of $\mathbf{R}^j_{jk_j}$ determines the average channel gain from one antenna at the j-th AP to the k_j-th UE in cell j, i.e., $\frac{1}{M} tr(\mathbf{R}^j_{jk_j}) = \beta^j_{jk_j}$, where $\beta^j_{jk_j}$ is the large-scale fading coefficient modeled as [4],

$$\beta^j_{jk_j} = \beta_0 - 10\alpha \log_{10}(d^j_{jk_j}) + F^j_{jk_j} \quad \text{dB} \tag{1}$$

where β_0 is the median channel gain at 1 km reference distance, $d^j_{jk_j}$ is the distance between the j-th AP and the k_j-th UE in cell j, α is the path-loss exponent, and $F^j_{jk_j} \sim \mathcal{N}(0, \sigma^2_{sf})$ denotes the shadow fading.

A. Uplink Training

To estimate the channel between the AP and UEs, we consider pilot-based channel estimation with time-division duplexing (TDD). The total number of samples in one coherence block is τ_c. The length τ_p of the pilot signals is determined by the maximum number of UEs among the L cells, i.e., $\tau_p = \max_{j=1,\ldots,L} K_j$. All UEs transmit the pilot signal with equal power ρ. The same set of orthogonal pilot signals is used in each cell. By applying minimum mean square error (MMSE) channel estimation [4], the j-th AP obtains the estimated channel of $\mathbf{h}^j_{jk_j}$,

$$\hat{\mathbf{h}}^j_{jk_j} = \mathbf{R}^j_{jk_j} \mathbf{Q}^{-1}_{jk_j} \boldsymbol{\psi}_{jk_j}, \tag{2}$$

in which $\mathbf{Q}_{jk_j} = \sum_{j'=1}^{L} \mathbf{R}^j_{j'k_j} + \frac{\sigma^2}{\rho\tau_p}\mathbf{I}_M$, $\boldsymbol{\psi}_{jk_j} = \sum_{j'=1}^{L} \mathbf{h}^j_{j'k_j} + \sqrt{\frac{\sigma^2}{\rho\tau_p}}\mathbf{n}_{jk_j}$, $\mathbf{n}_{jk_j} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{I}_M)$, and σ^2 is the noise power.

B. Downlink Communication

As shown in Fig. 1, APs only communicate with the UEs within its own cell. The desired signal transmitted from the j-th AP to the k_j-th UE in cell j is denoted as $a_{jk_j} \sim \mathcal{N}_{\mathbb{C}}(0, p_{jk_j})$, where p_{jk_j} is the downlink transmit power allocated to UE k_j in cell j. The downlink signal sent by the j-th AP can then be represented as $\mathbf{x}_j = \sum_{k_j=1}^{K_j} \mathbf{w}_{jk_j} a_{jk_j}$ where $\mathbf{w}_{jk_j}$ is the precoding vector assigned to UE k_j in cell j, where $\mathbb{E}\left\{\|\mathbf{w}^H_{jk_j}\mathbf{w}_{jk_j}\|^2\right\} = 1$, such that $\mathbb{E}\left\{\|\mathbf{w}_{jk_j} a_{jk_j}\|^2\right\} = p_{jk_j}$. Maximum ratio (MR) precoding is applied, i.e., $\mathbf{w}_{jk_j} = \frac{\hat{\mathbf{h}}^j_{jk_j}}{\|\hat{\mathbf{h}}^j_{jk_j}\|}$. The received signal at the k_j-th UE in cell j can then be written as,

$$\begin{aligned} y_{jk_j} = & \left(\mathbf{h}^j_{jk_j}\right)^H \mathbf{w}_{jk_j} a_{jk_j} + \sum_{i=1, i\neq k_j}^{K_j} \left(\mathbf{h}^j_{jk_j}\right)^H \mathbf{w}_{ji} a_{ji} \\ & + \sum_{l=1, l\neq j}^{L} \sum_{i=1}^{K_l} \left(\mathbf{h}^l_{jk}\right)^H \mathbf{w}_{li} a_{li} + n_{jk_j}, \end{aligned} \tag{3}$$

where $n_{jk} \sim \mathcal{N}(0, \sigma^2)$ denotes the noise in the downlink.

Then, the downlink spectral efficiency of the k_j-th UE in cell j can be written as,

$$\mathrm{SE}_{jk_j} = \frac{\tau_c - \tau_p}{\tau_c} \log_2\left(1 + \frac{p_{jk_j} c_{jk_j}}{\sum_{l=1}^{L}\sum_{i=1}^{K_l} p_{li} d_{lijk_j} + \sigma^2}\right) \tag{4}$$

where $\frac{\tau_c - \tau_p}{\tau_c}$ is the fraction of samples used for downlink communication per coherence block,

$$c_{jk_j} = \left|\mathbb{E}\left\{\mathbf{w}^H_{jk_j}\mathbf{h}^j_{jk_j}\right\}\right|^2 \tag{5}$$

and

$$d_{lijk_j} = \begin{cases} \mathbb{E}\left\{\left|\mathbf{w}^H_{jk_j}\mathbf{h}^l_{jk_j}\right|^2\right\}, & (l,i) \neq (j,k_j) \\ \mathbb{E}\left\{\left|\mathbf{w}^H_{li}\mathbf{h}^l_{jk_j}\right|^2\right\} - \left|\mathbb{E}\left\{\mathbf{w}^H_{li}\mathbf{h}^j_{jk_j}\right\}\right|^2, & (l,i) = (j,k_j) \end{cases} \tag{6}$$

The expectations are computed with respect to the channel realizations.

C. Power Allocation

Our focus is on power allocation for downlink communication, under the max-min fairness setting [5]. To this end, we consider power allocation strategies to maximize the minimum spectral efficiency across all UEs for a given EP.

Then, the optimization problem for any given EP can be formulated as,

$$\begin{aligned} \mathcal{P}^* = \arg & \max_{\{p_{jk_j}, \forall j, k_j\}} \min_{j,k_j} \mathrm{SE}_{jk_j} \\ & \text{subject to} \quad \sum_{k_j=1}^{K_j} p_{jk_j} \leq p_{\max}, \forall j = 1, \ldots, L \end{aligned} \tag{7}$$

where $\mathcal{P}^* = \{p^*_{jk_j}, \forall j, k_j\}$ denotes the optimal power allocation for all APs in a given EP. A bisection method can be used to solve (7) [4]. On the other hand, the time cost of doing so can be prohibitive in real-world settings when the number of UEs is large. To address this challenge, deep learning-based solutions have been proposed to alleviate the computational overhead of solving (7) during test time [5]. To do so, a dataset $\mathcal{D}$ can be generated using the UE locations as the features, and the theoretical solution from (7) can be used as the labels. Specifically, for each data point $\{\mathbf{V}, \mathcal{P}^*\} \in \mathcal{D}$, the features represent one realization of the UE positions denoted by $\mathbf{V} \in \mathbb{R}^{2\times K}$ at the given EP. To obtain the labels $\mathcal{P}^*$, APs associated with the same EP estimate the channel coefficients and send them to the EP. The EP then obtains the labels $\mathcal{P}^*$ by solving the max-min fairness problem from (7). At the end of training, the model can be used for power allocation in previously unseen environments directly without estimating c_{jk_j} and d_{lijk_j} in (4).

III. FL BASED POWER ALLOCATION

In this work, we consider a supervised learning task to determine the power allocation strategy. Different from the centralized training architecture in [5], where the dataset is stored at a central party, we assume that the dataset is distributed across a large number of EPs, with a local dataset stored locally at each EP. When the dataset is distributed across a large number of EPs, collecting the datasets at a central party may incur high communication cost in real world settings, in which case centralized training may not be feasible. Our goal is to accelerate the training process for deep learning based power allocation in distributed settings, by leveraging FL to enable multiple EPs to collaborate with each other, coordinated by a central processor, but without sharing their local datasets directly.

A. FL Framework

The goal of FL is to train a global model $\mathbf{W}^*$ to minimize a loss function [7],

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \sum_{n=1} \frac{|\mathcal{D}_n|}{\sum_{n=1}^{N}|\mathcal{D}_n|} F_n(\mathbf{W}_n^t, \mathcal{D}_n). \tag{8}$$

where $\mathcal{D}_n$ denotes the local dataset of EP n with $|\mathcal{D}_n|$ samples, $F_n(\mathbf{W}^t, \mathcal{D}_n) = \ell(\mathcal{P}_n^* - f(\mathcal{D}_n, \mathbf{W}^t))$ is the local loss function (MSE) of EP n, $f(\mathcal{D}_n, \mathbf{W}^t)$ represents the output of the neural network $\mathbf{W}_n^t$ to learn the mapping between UE positions $\mathbf{V}_n$ and the optimal power allocation $\mathcal{P}_n^*$ in dataset $\mathcal{D}_n$. Each EP deploys a local model $\mathbf{W}_n$ with the same architecture. FL training process consists of iterative global and local training rounds. At the beginning of training ($t = 0$), a global model $\mathbf{W}^0$ is initialized at the central processor. Next, at the t-th global round, the central processor first distributes $\mathbf{W}^t$ to all EPs. Each EP then updates the received model using their local dataset $\mathcal{D}_n$. This process is called local training, where each EP generates a local model $\mathbf{W}_n^t \leftarrow \mathbf{W}^t$, which is then updated through multiple local training rounds as,

$$\mathbf{W}_n^t \leftarrow \mathbf{W}_n^t - \eta \nabla F_n(\mathbf{W}_n^t, \mathcal{D}_n) \tag{9}$$

where η is the learning rate, and $\nabla F_n(\cdot)$ denotes the gradient. After local training is finished, the EPs transmit the trained local models $\{\mathbf{W}_n^t, n = 1, \ldots, N\}$ back to the central processor. We consider a synchronous FL setting where the central processor waits until the local models are received from all designated EPs. We further assume that the uploaded models can be received perfectly and that there are no stragglers in the system. Then the central processor aggregates the received local models to generate a global model $\mathbf{W}^{t+1}$ with better representation capability. The model aggregation process is represented as,

$$\mathbf{W}^{t+1} = \sum_{n=1} \frac{|\mathcal{D}_n|}{\sum_{n'=1}^{N}|\mathcal{D}_{n'}|} \mathbf{W}_n^t \tag{10}$$

The training process iteratively repeats the above stages until convergence. In bandwidth-limited real world settings, not all EPs can participate in FL when the number of EPs is large. A common approach is then to select a small subset of EPs $\mathcal{N}^t$ in each global round to save communication resources, and a popular mechanism for user selection is FedAvg [7], where EPs are selected uniformly at random. FedAvg has been shown to have comparable performance to full participation with a sufficient number of global rounds, when the dataset is distributed i.i.d. among the EPs. Data heterogeneity, on the other hand, can severely degrade model performance [7]. In our work, the dataset distributions are heterogeneous across the EPs, as the wireless environment observed by each EP can be different; some EPs might support an urban area, where UEs are densely distributed, while some EPs may serve a rural area where UEs are sparse. In such settings, a robust EP selection mechanism is critical for training a reliable power allocation model, as selecting the EPs uniformly at random may degrade the performance of the global model. We will next demonstrate a contextual multi-armed bandit mechanism for EP selection for federated power allocation, to address this challenge.

B. Contextual MAB-Based EP Selection

To address this challenge, we leverage a contextual MAB algorithm for user selection for sample-efficient FL in heterogeneous networks. The bandit is implemented at the central processor. Each EP is treated as an arm, with its own context. The goal is to select the optimal set of EPs to maximize the reward respect to the actions taken by the central processor, where the action is the selection of S arms with the highest scores, where $S = |\mathcal{N}^t|$.

We use the local training performance of the EPs as context information to guide the user selection strategy. Specifically, the context vector for EP n at global round t is represented as

$$\mathbf{b}_n^t = \left[r^{t-1}, \bar{\ell}^t(\mathcal{T}_n), \bar{\ell}^t(\mathcal{V}_n) \right] \tag{11}$$

where the local dataset $\mathcal{D}_n = \{\mathcal{T}_n, \mathcal{V}_n\}$ of EP n is partitioned into a local training set $\mathcal{T}_n$ and a local validation set $\mathcal{V}_n$, $\bar{\ell}(\mathcal{T}_n)$ is the normalized training loss, $\bar{\ell}(\mathcal{V}_n)$ is the normalized local validation loss, and r^{t-1} is the reward from the last global round. The normalization for both training and validation loss is operated in the same way, i.e., by using the current training (validation) loss divided by the first round training (validation) loss. In doing so, the key intuition is to make the central processor select the EPs with a larger training and validation loss, to maximize the worst-case performance.

Since our goal is to achieve max-min fairness, i.e., maximize the minimum spectral efficiency among the EPs, the global reward for the contextual MAB algorithm is designed as follows,

$$r^t = \overline{\mathrm{SE}}^t - \overline{\mathrm{SE}}^{t-1} \tag{12}$$

where $\overline{\mathrm{SE}}^t$ is the minimum average downlink spectral efficiency in the test set using the global model at global round t. Then, Thompson Sampling is utilized to select the arms that correspond to the top S scores. We denote $\mathsf{B}^t \in \mathsf{R}^{(S\cdot t)\times 3}$ as the collection of the context history for all selected EPs, and $\mathsf{r}^t \in \mathsf{R}^t$ as the reward history across t global rounds. We further assume that $\boldsymbol{\theta} \in \mathsf{R}^3$ denotes the parameters of the contextual bandit which is unknown to the EPs. At each global round t, the parameters are determined based on the selection history (contexts and rewards history), sampled as,

$$\boldsymbol{\theta}^t \sim \mathcal{N}\left(\left(\left(\mathsf{B}^t\right)^\top \mathsf{B}^t + \lambda \mathsf{I}\right)^{-1} \mathsf{B}^t \mathsf{r}^t, \delta^2\left(\left(\mathsf{B}^t\right)^\top \mathsf{B}^t + \lambda \mathsf{I}\right)^{-1}\right),$$

where λ is a regularization parameter, and δ denotes the exploration strength. Then the score for EP n at round t is given by,

$$\mathrm{Score}_n^t = \left(\boldsymbol{\theta}^t\right)^\top \mathsf{b}_n^t \tag{13}$$

At the end of each round, the context vectors of the selected EPs along with the reward will be appended to the context history B^t and reward history r^t, respectively. By doing so, EPs with higher training loss are given a higher priority in the selection process, which can significantly speed up the training as will be demonstrated in our experiments.

IV. Numerical Evaluation

We consider a hierarchical massive MIMO network with 15 EPs, where each EP supports 4 adjacent cells. Each cell is a $500m \times 500m$ square region with the AP located at the center. Each AP is equipped with 100 transmit antennas. We consider communication over a bandwidth of 20 MHz with a noise power of -94 dBm at the receiver. There are 8 UEs distributed within the coverage area of each EP. We assume that different EPs serve different geographical regions, modeled by the three topologies shown in Fig. 2, where the number indicated in each cell represents the number of UEs within the corresponding cell. For the first topology, all UEs are distributed uniformly at random within the coverage area. For the second topology, most UEs are concentrated within a single cell, whereas the remaining cells have relatively fewer UEs. As for the third topology, all UEs are distributed within two cells, with the remaining cells being empty. The total dataset size is $30,000$, with $10,000$ samples realized from each topology. The central coordinator is assigned a test set of 3000 samples (with 1000 samples selected uniformly at random from each topology). The remaining $27,000$ samples

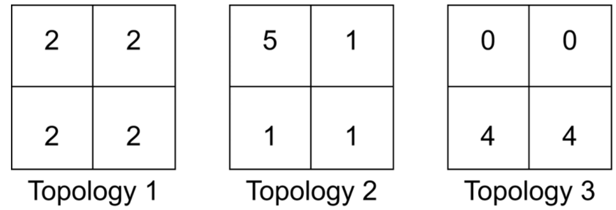

Fig. 2. Heterogeneous UE distribution topologies.

are distributed non-i.i.d. across 15 EPs, such that each EP is randomly assigned realizations from two different topologies with an equal number of data points. Specifically, EPs 1-5 are assigned 900 samples from topologies 1 and 2, EPs 6-10 are assigned 900 samples from topologies 2 and 3, whereas EPs 11-15 are assigned 900 samples from topologies 1 and 3 (uniformly random without replacement). Accordingly, we consider a heterogeneous network with three different groups of EPs, where five EPs observe realizations from topologies 1 and 2, five EPs from with topologies 2 and 3, and the remaining EPs observe topologies 1 and 3. Then, the local dataset of each EP is partitioned into a local training set and local validation set, respectively, with a $9:1$ split ratio. At the end, each EP has 810 random samples per topology within the local training set and 90 samples within the local validation set. We consider a severely resource-limited setting in which only a single EP can participate in FL at each global round.

We then consider a 1-dimensional (1D) convolutional neural network for power allocation. The input features are the 2D Cartesian coordinates of UE positions. The input layer is followed by three 1D convolutional layers, with a kernel size of 3. The number of neurons in the convolutional layers are 128, 64 and 32. After each convolutional layer, we consider a 1D batch normalization layer. The convolutional layers are followed by three fully connected layers with the number of neurons 128, 64 and 8, respectively. The last layer is the output layer, where the output denotes the power allocation for all UEs for the given EP. Mean squared error (MSE) is used as the loss function $\ell(\cdot)$ for training. The learning rate is 0.001 and we apply the Adam optimizer from [8].

To evaluate the performance of the proposed FL framework with context-based EP selection, we compare performance in terms of the minimum spectral efficiency with respect to the following baselines:

- *Centralized training*: The entire training set is available at the central processor who performs centralized training, adapted from the centralized training mechanism from [6] to our setting,
- *Global aggregation*: Federated learning with all EPs participating in model aggregation at each training round during the global model update,
- *FedAvg* (federated averaging) [7]: FL with uniformly random EP selection (1 EP is selected uniformly at random per global round),
- *Contextual MAB* (our work): FL with contextual MAB-based EP selection (1 EP is selected per global round),
- *Decentralized*: Each EP trains its own model locally