IEEE ComSoc | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS | IEEE Transactions on Machine Learning in Communications and Networking

Received 18 October 2022; revised 10 February 2023 and 4 April 2023; accepted 27 May 2023.
Date of publication 5 June 2023; date of current version 3 July 2023.

The associate editor coordinating the review of this article and approving it for publication was M. Chen.

Digital Object Identifier 10.1109/TMLCN.2023.3283228

Access Point Clustering in Cell-Free Massive MIMO Using Conventional and Federated Multi-Agent Reinforcement Learning

BITAN BANERJEE[1] (Graduate Student Member, IEEE), ROBERT C. ELLIOTT[1] (Senior Member, IEEE), WITOLD A. KRZYMIEŃ[1] (Fellow, IEEE), AND MOSTAFA MEDRA[2] (Member, IEEE)

[1]Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada
[2]Huawei Technologies Canada Company Ltd., Ottawa, ON K2K 3J1, Canada

CORRESPONDING AUTHOR: W. A. KRZYMIEŃ (krzymien@ualberta.ca)

This work was supported in part by Huawei Technologies Canada Company Ltd. and in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada. The work in this paper was presented in part at the 2022 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC).

ABSTRACT Cell-free massive multiple-input multiple-output (MIMO) systems consist of geographically-distributed multi-antenna access points (APs) that form a virtual massive MIMO array. To make the network arbitrarily scalable in size, each user should be served by the best possible personalized user-centric cluster of nearby APs. Unfortunately, determining that cluster is a combinatorially-complex problem made even harder when the users are in motion. Therefore, in this work, we develop a multi-agent reinforcement learning (MARL) algorithm for AP selection and clustering. Each AP is an agent in the MARL algorithm and it is trained to near-optimally select for itself which users to serve. Conventional MARL algorithms require a centralized reward system to train the agents, and the agents' neural network weights tend to strongly depend on their locations during training. To counteract these problems, we also consider a federated MARL framework. Simulation results demonstrate both our conventional and federated MARL algorithms outperform existing published AP selection algorithms, and also provide performance comparable to the case of all APs serving all users. The results also show the conventional algorithm has somewhat superior performance in the environment it was trained in, but the federated algorithm transfers its learning to changed environments much better, with very little performance loss.

INDEX TERMS Access point clustering, cell-free massive MIMO, centralized critic, decentralized actors, federated reinforcement learning, multi-agent reinforcement learning, user association

I. INTRODUCTION

MASSIVE multiple-input multiple-output (MIMO) antenna systems are a key component of 5th generation (5G) and beyond cellular networks in order to achieve high spectral efficiency (SE), data rate, and throughput requirements [1]. Most typically, massive MIMO antenna arrays are assumed to have all their elements co-located at a base station (BS) [2], [3], [4]. However, in such cases, the massive MIMO cellular network is normally limited by inter-cell interference, resulting in poor cell-edge performance. Therefore, for beyond-5G cellular networks, which put more emphasis on equitable service for all pieces of user equipment (UE) within the coverage area, modifications to the network architecture are necessary. To overcome this shortcoming of conventional massive MIMO, distributed massive MIMO architectures have been studied. In the literature, the core idea of distributed MIMO has been examined under various names, including distributed antenna system (DAS) [5], [6], network MIMO [7], [8], [9], coordinated multipoint (CoMP) transmission [10], [11], [12], [13], [14], or cloud radio access network (C-RAN) [15], [16], [17], [18], [19].

More recently, distributed architecture has again appeared in the ''massive''-sized array regime with the name of cell-free massive MIMO [20]. In cell-free massive MIMO, the access points (APs), each with one or more antennas, are distributed over a geographical area, and multiple APs

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

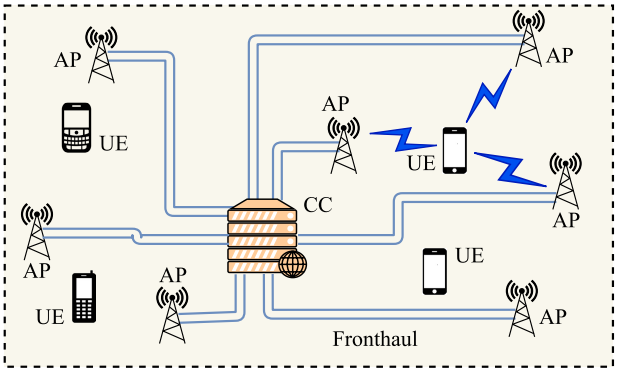


FIGURE 1. Illustration of a cell-free massive MIMO network. (AP: access point, CC: central controller, UE: user equipment.)

coordinate to form a virtual massive MIMO array to serve UEs [21]. Cell-free massive MIMO conceptually removes cell boundaries and therefore suppresses the inter-cell interference problem. This creates more uniformity of service and fairness to UEs over the entire network area [21]. It has been observed in [20] that in a cell-free massive MIMO network, the APs are closer to the UEs than in conventional massive MIMO, yielding higher diversity gain, lower path loss, and better throughput. Reference [20] has also shown that cell-free massive MIMO has significantly better performance than small-cell systems where each UE is served by a single BS.

A. BACKGROUND AND MOTIVATION

We illustrate a typical cell-free massive MIMO network in Fig. 1. In canonical cell-free massive MIMO, UEs are served by all the APs [20], [22], which are connected to a central controller (CC) using fronthaul connections. Therefore, the number of fronthaul connections increases proportionally with the number of APs in the network. Furthermore, as each AP serves all UEs, the overall fronthaul capacity requirement also increases, along with computing requirements for signal processing. These factors give rise to scalability issues, and thus the canonical form of cell-free massive MIMO is impractical for an arbitrarily large number of APs in the network.

In contrast, in scalable cell-free massive MIMO systems, each UE is served by a subset of APs [23], [24], [25], [26], [27]. Importantly, the cluster of serving APs should be user-centric and individualized for each UE.[1] This challenge was initially addressed in [23], wherein the authors proposed a user-centric AP cluster solution; other works since have also examined the problem (see Section II). However, two major issues still remain open: 1) How should the system select which APs to serve a UE in real-time in an environment where the UEs are in motion, where AP selection may need to be updated often? 2) How should the system support the significant fronthaul and computational load in such an environment? We focus mainly on the first question, and address the second through the use of localized precoding.

To tackle these challenges, this paper focuses on developing machine learning (ML) methods for AP clustering such that each AP can determine the UEs it serves mostly independently of the others. To support the dynamic nature of a mobile environment, reinforcement learning (RL) is a natural choice. In a recent article [28], multi-agent RL (MARL) techniques have been applied to a canonical cell-free massive MIMO network to solve the power allocation problem in a mobile environment, and the performance of the MARL algorithm therein is promising. Those results suggest that MARL algorithms would be suitable for cell-free massive MIMO with mobile UEs, which served as one of our initial motivations for applying MARL to AP clustering. In our earlier work [29], we developed an actor-critic MARL framework that trains the APs to select which UEs to serve; each AP is a distributed agent/actor in the system, and the centralized critic that judges the agents' performance is located at the CC of the network. Because the agents are distributed, with the use of localized precoding, the fronthaul load has been reduced. However, during training, conventional MARL systems require regular information updates regarding rewards from the CC, which can result in significant communication overhead [30]. As an alternative, to train the agents with limited interactions between the CC and agents, federated learning (FL) [31] is a promising technique. Under FL, distributed agents train their neural networks (NNs) locally. The CC periodically requests the NN weights from the agents and uses those local weights to compute and distribute a global NN weight update for all agents. FL was initially developed and deployed by Google in their predictive keyboard feature [32]. Later, it has been observed that combining features of FL with those of RL can help reduce the number of interactions between the CC and agents [30], [33]. This motivates us to modify our previous MARL system to a multi-agent federated reinforcement learning (MAFRL) system, and study its performance. Among various distributed ML methods, both MARL and FL have been deemed to be key techniques for wireless communication problems [34].

B. CONTRIBUTIONS

The specific contributions of this paper are:

- We develop a MARL framework for AP clustering in an environment with mobile (i.e., non-stationary) UEs. We consider UE mobility at pedestrian speeds when creating the simulation environment. We formulate the problem as a Markov game and then solve it using the "decentralized actor, centralized critic" variant of reinforcement learning. We develop multiple reward policies to incorporate fair performance.
- We extend the MARL system to a MAFRL system by introducing FL features. We describe how implementing a MAFRL-based solution can further reduce the communication overhead fronthaul load.

[1] This is an important and notable difference from earlier work on distributed architecture, where the clusters of BSs/APs/antennas were typically centric to the serving nodes rather than the UEs.

- We examine the performance of the proposed actor-critic MARL and MAFRL algorithms for UE association and AP clustering via simulations. We also compare the SE of the MARL and MAFRL algorithms with those of greedy-based AP clustering, ML-based clustering algorithms proposed in [25] and [35], and a modified RL-based downlink (DL) power control algorithm from [36]. As part of this examination, we illustrate the differences in the performance obtained by extending our MARL algorithm to a MAFRL algorithm. We demonstrate that the MAFRL performance is somewhat inferior to that of the MARL algorithm in the trained environment, but the MAFRL algorithm also transfers its learning to new environments more readily and without a notable performance loss, in contrast to the MARL algorithm.

C. ORGANIZATION

The rest of the paper is organized as follows. We briefly provide an overview of related work in Section II. In Section III, we describe the model of the cell-free massive MIMO network, the precoding method, and the calculation of SE. The framework and details of our MARL and MAFRL techniques are discussed in Sections IV and V, respectively. We evaluate and discuss the simulated performance of our proposed algorithms in Section VI. Finally, we conclude the paper in Section VII.

Notation: Italic variables like a or A denote scalars, whereas boldface uppercase ($\mathbf{A}$) and lowercase ($\mathbf{a}$) variables denote matrices and vectors, respectively. Calligraphic variables like $\mathcal{A}$ and $\boldsymbol{\mathcal{A}}$ represent sets and families of sets, respectively, with $|\mathcal{A}|$ being the cardinality. $\mathbf{M}^H$ denotes the Hermitian (conjugate) transpose of matrix $\mathbf{M}$. If $\mathbf{M}$ is square, $\mathbf{M}^{-1}$ and tr($\mathbf{M}$) respectively denote its inverse and trace. $\mathbf{I}_n$ is the $n \times n$ identity matrix, and $\mathbf{0}_{m \times n}$ is an $m \times n$ matrix containing all zeros. $\mathbb{1}(x)$ is an indicator function that equals 1 if condition x is true, and 0 otherwise.

II. RELATED WORK

To the best of our knowledge, distributed architecture specifically in the context of massive MIMO was initially investigated in [37], where a BS selection procedure was developed; the selected BSs coordinated using either maximum ratio combining or minimum mean square error (MMSE) combining to serve the UEs on the uplink (UL) of hexagonal cells. The authors of [20], [38] first gave the name "cell-free massive MIMO" to the core idea of distributed massive MIMO architecture.

The benefits of cell-free massive MIMO come at the price of increased fronthaul capacity requirements [39]. Existing literature typically assumes infinite-capacity fronthaul links, e.g., [20], [23]. However, prior work in similar contexts has shown that limited fronthaul capacity has a significant performance impact, e.g., for CoMP [40] or C-RAN [41]. The performance of cell-free massive MIMO with capacity-constrained fronthaul links has been studied for some specific scenarios in [25], [39], and [42]. Distributed precoding [21] also helps address the issue; we use this approach herein.

Although there are relatively few works on AP clustering or selection, several works on the related problem of antenna selection for massive MIMO are available in the literature, e.g., [43], [44], [45], [46]. Antenna selection and AP clustering are fundamentally the same type of problem. However, solutions to the former are most typically centric to the transmit nodes, whereas user-centric solutions are best for the latter. In [44] and [45], the authors have proposed greedy selection algorithms; [44] has maximized the incremental sum rate with each selected antenna, whereas [45] has used the technique of matching pursuits. The authors of [43] have proposed a branch-and-bound selection algorithm based on the largest minimum singular value of channel submatrices. An ML method for joint antenna selection and user scheduling to maximize the energy efficiency of a single-cell massive MIMO system has been proposed in [46]. The authors of [47] have investigated the related problem of antenna clustering in distributed antenna systems, and [8] has considered cell clustering for network MIMO. In the context of C-RANs, the authors of [15] have considered joint user clustering and sparse beamforming under the constraints of finite-capacity backhaul links, and have obtained a solution by optimizing a weighted MMSE problem. The authors of [19] have framed the user clustering problem as one of a cooperative bargaining game, whose Nash equilibrium has been found in part by a Hungarian method to pair bargaining users.

In the context of cell-free massive MIMO, [25] has proposed two strategies for AP clustering: 1) minimize the number of UE-AP associations subject to the signal-to-interference-plus-noise ratio (SINR) being greater than a threshold, and 2) maximize the minimum SINR subject to a maximum allowable number of APs associated with a UE. More recently, an AP selection method using an ML algorithm based on κ-means clustering has been proposed in [35], a multiple user access scheme using deep RL has been investigated in [48], and a distributed beamforming technique using deep RL has been considered in [49]; [35] has considered DL transmissions, whereas [48] and [49] have considered data transmissions on the UL. Additionally, cell-free massive MIMO DL power control/allocation schemes using deep RL have been developed in [28] and [36].

None of the above antenna selection algorithms considers an environment with UEs in motion. In such a dynamic environment, the association problem needs to be re-solved periodically. Typical deep neural networks (DNNs) face another challenge in that the input or output state size may vary with the number of nearby and/or active UEs. Therefore, we consider the use of RL to solve the AP clustering problem, as it is suited to handle dynamic environments. Recently, the authors of [28] have developed RL-based power allocation strategies for a mobile environment. In our MARL algorithm, which we first investigated in [29], we take a more

distributed approach with decentralized actors and a centralized critic, inspired by the work in [50]. In this approach, each agent (actor) only has localized environmental information for its AP, whereas the critic has global information. However, a conventional MARL approach such as this has a couple of shortcomings. First, the agents need frequent feedback from the central critic to get their rewards and accordingly update their NN weights. Thus, conventional MARL increases the communication overhead, and to some degree contradicts the basic philosophy of distributed operation in cell-free massive MIMO. Second, the overall policy learned by each agent is strongly contingent on the location of that agent. Therefore, the learned policies are somewhat dependent on the environment, which makes transferring the agents and their policies to a new environment problematic [33], [51].

To overcome these issues, we additionally consider a MAFRL algorithm. A key consideration in the development of FL was maintaining data privacy between different agents in a system. Agents are only allowed to share learned information (most typically their local NN weights), but not the data with which they train [31]. In the context of communication systems, FL has been used in a variety of scenarios ranging from resource allocation and optimization problems, edge caching and computing, vehicular networks (whether road-based or unmanned aerial vehicles), health care, and the Internet of Things [52], [53], [54]. General frameworks for using FL in beyond-5G networks have been proposed in [55] and [56], while [34] has surveyed numerous distributed ML techniques for wireless communications, including RL, FL, and other methods that operate in a completely distributed manner with no central coordination. In [57], the authors have considered how best to use the APs of a cell-free massive MIMO system to support and optimize training of an FL framework, where the local NNs being trained are located at the UEs. A related problem has been examined in [58], where massive MIMO and compressive sensing have been used to help reconstruct sparse gradient vectors used for the FL updates. References [59] and [60] have considered FL methods for channel estimation, whereas [61] has used a mixture of deep FL and game theory for dynamic frequency allocation in multicell massive MIMO networks. In [30], the authors have used federated deep RL to tackle the problem of user access control in open radio access networks. However, to the best of our knowledge, our work is the first to combine the advantages of both reinforcement learning and federated learning in the context of optimizing AP clustering in a cell-free massive MIMO network, while also considering mobility of UEs.

III. CELL-FREE MASSIVE MIMO SYSTEM MODEL

Consider the DL of a cell-free massive MIMO system with L APs that serve a total of K single-antenna UEs, where each AP is equipped with N antennas. We assume that $L \times N \gg K$, which is the typical operating regime for massive MIMO. Each AP can serve any of the UEs, and theoretically can serve any number of them. However, as mentioned earlier, the more UEs served, the more significant the fronthaul load will be. The APs are connected to a CC that forwards UE data symbols to the APs and coordinates the training of ML.

Time-division duplex (TDD) mode is used to alternate between UL and DL transmission. As such, DL channel state information (CSI) may be obtained from the assumption of UL/DL radio channel reciprocity. The UL channel $\mathbf{h}_{k\ell} \in \mathbb{C}^{N\times 1}$ between UE k and AP ℓ is distributed $\sim \mathcal{CN}(0, \mathbf{R}_{k\ell})$, which models correlated Rayleigh fading; $\mathbf{R}_{k\ell} \in \mathbb{C}^{N\times N}$ is the channel covariance matrix. $\beta_{k\ell} = \text{tr}(\mathbf{R}_{k\ell})/N$ is the large-scale fading parameter of the channel, incorporating path loss and shadow fading [21]. The APs make an estimate $\widehat{\mathbf{h}}_{k\ell}$ of the UL channels based on pilot sequences sent by the UEs, as follows [21]:

$$\widehat{\mathbf{h}}_{k\ell} = \sqrt{\rho_p \tau_p}\, \mathbf{R}_{k\ell} \boldsymbol{\Psi}_{k\ell}^{-1} \mathbf{y}_{k\ell}^{p}, \tag{1}$$

where

$$\boldsymbol{\Psi}_{k\ell} = \mathbb{E}\left\{ \mathbf{y}_{k\ell}^{p} (\mathbf{y}_{k\ell}^{p})^{H} \right\} = \rho_p \tau_p \mathbf{R}_{k\ell} + \sigma^2 \mathbf{I}_N \tag{2}$$

is the $N \times N$ covariance matrix of the received pilot signal $\mathbf{y}_{k\ell}^{p} \in \mathbb{C}^{N\times 1}$ from UE k at AP ℓ. ρ_p and τ_p are respectively the power and the length of the transmitted pilot sequence, and σ^2 is the variance of the noise (assumed to be distributed $\sim \mathcal{CN}(0, \sigma^2 \mathbf{I}_N)$). We also denote the covariance matrix of the error between $\mathbf{h}_{k\ell}$ and $\widehat{\mathbf{h}}_{k\ell}$ as $\mathbf{C}_{k\ell} \in \mathbb{C}^{N\times N} = \mathbf{R}_{k\ell} - \rho_p \tau_p \mathbf{R}_{k\ell} \boldsymbol{\Psi}_{k\ell}^{-1} \mathbf{R}_{k\ell}$ [21]. Here, we assume for simplicity that every UE has its own orthogonal pilot sequence, so interference between pilots does not exist. We also assume the noise variance is the same on the UL and DL and the same[2] for all UEs and APs.

Let us assume that UE k is served by the APs in set $\mathcal{L}_k$. We define an $N \times N$ binary diagonal matrix $\mathbf{D}_{k\ell}$ to represent if UE k is associated with AP ℓ:

$$\mathbf{D}_{k\ell} = \begin{cases} \mathbf{I}_N, & \ell \in \mathcal{L}_k; \\ \mathbf{0}_{N\times N}, & \ell \notin \mathcal{L}_k. \end{cases} \tag{3}$$

The effective DL channel vector between AP ℓ and UE k can then be considered to be $\mathbf{h}_{k\ell}^{H} \mathbf{D}_{k\ell}$. We assume that distributed DL precoding is performed, i.e., precoding is done locally at each AP. The data symbol for UE k is given by ζ_k (with $\mathbb{E}\{|\zeta_k|^2\} = 1$), which is sent from the CC to the serving APs over the fronthaul. The received DL signal at UE k is given by [21]

$$y_k = \left(\sum_{\ell=1}^{L} \mathbf{h}_{k\ell}^{H} \mathbf{D}_{k\ell} \mathbf{w}_{k\ell} \right) \zeta_k + \sum_{\substack{i=1, \\ i\neq k}}^{K} \left(\sum_{\ell=1}^{L} \mathbf{h}_{k\ell}^{H} \mathbf{D}_{i\ell} \mathbf{w}_{i\ell} \right) \zeta_i + n_k. \tag{4}$$

$\mathbf{w}_{k\ell} \in \mathbb{C}^{N\times 1}$ is the precoding vector that AP ℓ uses for UE k, and n_k is the noise. The double summation in the second term of (4) represents interference from signals for other UEs, sent from both the serving APs for UE k and the other APs.

[2]Even if the noise variances are not the same, due to the network being interference-limited, differences in the variances have a negligible impact on the performance of the system.

To reduce the fronthaul load, we consider localized precoding, where each AP only has the knowledge of its own CSI for the UEs it serves. Thus, no CSI from other APs needs to be exchanged over the fronthaul; only data symbols need to be forwarded. However, with only local CSI knowledge, an AP can't coordinate with any other to serve its UEs. Therefore, it can only create at most N independent spatial streams for its N antennas, meaning it can serve up to N UEs simultaneously.[3] Specifically, we consider local partial MMSE (LP-MMSE) precoding [21]. Let the set of UEs served by AP ℓ be denoted by $\mathcal{D}_\ell$. The (arbitrarily scaled) LP-MMSE precoding vector for UE k at AP ℓ is given by

$$\overline{\mathbf{w}}_{k\ell} = p_{k\ell}\left(\sum_{i\in\mathcal{D}_\ell} p_{i\ell}\left(\widehat{\mathbf{h}}_{i\ell}\widehat{\mathbf{h}}_{i\ell}^H + \mathbf{C}_{i\ell}\right) + \sigma^2\mathbf{I}_N\right)^{-1}\mathbf{D}_{k\ell}\widehat{\mathbf{h}}_{k\ell}. \quad (5)$$

$p_{k\ell}$ ($p_{i\ell}$) is the transmit power assigned by AP ℓ for UE k (i). To normalize the total transmit power, the AP uses the precoding vector $\mathbf{w}_{k\ell} = \overline{\mathbf{w}}_{k\ell}\sqrt{p_{k\ell}}/\sqrt{\mathbb{E}\{\|\overline{\mathbf{w}}_{k\ell}\|^2\}}$. Typically, the transmit power for each UE is determined by a power allocation algorithm, such as in [28] and [36]. However, for simplicity, in this work we have used equal power allocation, i.e., $p_{i\ell} = P_t/|\mathcal{D}_\ell|, \forall i \in \mathcal{D}_\ell$, where P_t is the total transmit power available at the AP (assumed to be the same for all APs). LP-MMSE precoding is scalable to arbitrary network sizes, since the maximum data volume transferred over the fronthaul to AP ℓ is $|\mathcal{D}_\ell| \leq N$ data symbols, which is independent of both K and L.

The effective DL SINR of UE k is given by [21, Eq. (6.22)]

$$\Upsilon_k = \frac{\left|\sum_{\ell=1}^{L}\mathbb{E}\{\mathbf{h}_{k\ell}^H\mathbf{D}_{k\ell}\mathbf{w}_{k\ell}\}\right|^2}{\sum_{i=1}^{K}\mathbb{E}\left\{\left|\sum_{\ell=1}^{L}\mathbf{h}_{i\ell}^H\mathbf{D}_{k\ell}\mathbf{w}_{i\ell}\right|^2\right\} - \left|\sum_{\ell=1}^{L}\mathbb{E}\{\mathbf{h}_{k\ell}^H\mathbf{D}_{k\ell}\mathbf{w}_{k\ell}\}\right|^2 + \sigma^2}. \quad (6)$$

An achievable SE for UE k may then be defined as:

$$\eta_k = \log_2(1+\Upsilon_k). \quad (7)$$

This SE of the UEs is used to define the reward functions in our RL algorithms.

IV. REINFORCEMENT LEARNING FRAMEWORK

In this section, we develop the RL framework for solving our AP clustering problem. RL is a very effective ML technique for dynamic environments such as real-time strategic games and autonomous driving [62], [63], [64]. To implement the MARL algorithm, first we define the AP clustering problem as a Markov game [65], represented as a tuple $(\mathcal{L}, \mathcal{S}, \boldsymbol{\mathcal{A}}, \mathcal{P}, r, \gamma)$ [66]. $\mathcal{L} = \{1, 2, \ldots, |\mathcal{L}|\}$ is the set of agents, which in our case are the APs. The state of the environment or state space is represented by $\mathcal{S}$. In our case, the state is based on the received signal strength (RSS) of each UE at the APs. The RSS between AP ℓ and UE k is calculated from the received pilot signal as follows:

$$\text{RSS}_{k\ell} = ||\mathbf{y}_{k\ell}^p||^2. \quad (8)$$

We note that the RSS is directly proportional to $\beta_{k\ell}$, as $\mathbb{E}\{||\mathbf{y}_{k\ell}^p||^2\} = \rho_p\tau_p N\beta_{k\ell} + \sigma^2$. Thus, the pilot signals sent by the UEs are used by each AP both to calculate that AP's RSS values for all UEs, and to estimate the UL CSI for the set of UEs that AP serves. The joint action space $\boldsymbol{\mathcal{A}}$ is the Cartesian product of the action spaces $\mathcal{A}_\ell$ for all agents. The variable $\mathcal{P} : \mathcal{S}\times\boldsymbol{\mathcal{A}}\times\mathcal{S} \to \mathbb{R}$ represents the transition probability kernel of moving from one state to another. The reward function is represented by $r : \mathcal{S}\times\boldsymbol{\mathcal{A}} \to \mathbb{R}$, and $\gamma \in [0, 1)$ is called the discount factor. In the MARL algorithm, each agent ℓ has its own parameter vector $\boldsymbol{\theta}_\ell$ (which is basically its NN weights); concatenating $\boldsymbol{\theta}_\ell$ of all agents forms a joint parameter vector $\boldsymbol{\theta}$. In step $t \in \mathbb{N}$, the environment is in state $s_{(t)}$; agent ℓ takes an action $a_{\ell,(t)} \in \mathcal{A}_{\ell,(t)}$ based on the policy $\pi_{\boldsymbol{\theta}_\ell}(a_{\ell,(t)}|s_{(t)})$, where $\mathcal{A}_{\ell,(t)}$ is the action space of agent ℓ at step t. The joint policy of all the agents is

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(a_{1,(t)}, a_{2,(t)}, \ldots, a_{L,(t)}|s_{(t)}) = \prod_{\ell=1}^{L}\pi_{\boldsymbol{\theta}_\ell}(a_{\ell,(t)}|s_{(t)}). \quad (9)$$

Training the MARL algorithm consists of groups of steps called "episodes"; the weights of the NNs are updated after each episode following a policy gradient approach [67]. The agents (or actors, in an actor-critic framework) aim to find the optimal policy that, on average, will maximize the cumulative reward in step t, i.e., $R_{(t)} = \sum_{i=0}^{\infty}\gamma^i r_{(t+i)}$. The performance of agent ℓ's policy is evaluated using the centralized action-value function $\text{Q}_\ell^{\boldsymbol{\pi}_{\boldsymbol{\theta}}} = (\boldsymbol{x}, a_1, a_2, \cdots, a_L)$, where $\boldsymbol{x}$ contains the relevant information about the state of the environment. $\text{Q}_\ell^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ defines the algorithm's critic; essentially, it determines the rewards given by the critic to agent ℓ depending on the actions of all agents [50]. For a more detailed description of these parameters, we refer the reader to [50], [65], [66], [67], and [68].

In our considered system, the UEs are mobile, and therefore, the AP clustering should focus on long-term rewards for optimal AP-UE association. The value of the discount factor γ determines over how long of a period an agent's actions affect its rewards during training [28]. An exponentially-decaying weight γ^i is applied to future rewards; the larger the value of γ, the more emphasis that is placed on long-term rewards. Each AP needs to decide whether it is better to serve a given UE now or wait until later, based on the movement of all UEs. For example, the RSS and SE for a UE and thus the reward for serving that UE will increase over time if said UE is moving towards the AP, and decrease if it is moving away. The emphasis of "waiting until later" on this decision (and how long to wait) depends on the value of γ. Thus, in mobile environments, the discount factor indirectly helps APs learn about possible UE movement and whether serving a specific UE at a given time is good for the cumulative reward.

[3] This is in contrast to $C\times N$ UEs that can be jointly served if C APs coordinate with centralized precoding.

In this work, we have implemented an actor-critic policy gradient MARL-assisted approach, which is efficient in dealing with high-dimensional action spaces [67], [69]. In our case, the size of the action space for each AP is 2^K, as the output state for each UE is either 1 or 0 (i.e., associated with that AP or not). To improve the speed of convergence, we reduce the size of the state space for each agent by first selecting a pool of only Φ UEs with the highest RSS at that AP from the available K. Use of the pool also ensures the algorithm is scalable to arbitrarily large K. This furthermore largely solves the problem of potentially inactive UEs, which would cause the length of the input vector for the NNs to not be constant. We vary the value of Φ to examine its effect on the MARL algorithm's performance.

However, this approach by itself does not guarantee service to all UEs. For instance, a UE may not be associated with any AP if that UE's RSS is not within the top Φ RSSs for any AP. To address this issue, at each AP, two additional UEs that are not yet in the AP's pool are chosen in a round-robin[4] fashion and added to the pool; the AP then serves up to N UEs from that enlarged pool. Additionally, we introduce a global penalty to all APs if not all UEs are served. The penalty at time step t is

$$\mathrm{P}_{G(t)}\Big(\Big|\bigcup_{\forall \ell}\mathcal{D}_\ell\Big|\Big) = \big(\varrho + \mathrm{P}_{G(t-1)}\big)\cdot \mathbb{1}\Big(\Big|\bigcup_{\forall \ell}\mathcal{D}_\ell\Big| < K\Big). \quad (10)$$

$\mathrm{P}_{G(0)}$ is initialized to 0, and a value of ϱ is progressively added to the penalty for each time step that all K UEs are not served. The argument of the indicator function checks if all the UEs in the coverage area have been served. If so, the indicator function resets the penalty to 0. The CC applies the global penalty to the reward of every AP, then forwards the resulting rewards to their corresponding APs over the fronthaul. Enlarging each AP's pool with unserved UEs, combined with the penalty, helps the agents to learn within a few time steps that all the UEs should be served. Overall, in the MARL implementation, the additional overhead is the information shared between the APs and the CC, i.e., the reward for each AP in every time step. The interaction between the cell-free massive MIMO network and the agents' NNs in the MARL algorithm is shown in Fig. 2.

The NN for each agent consists of an input layer (Φ+2 nodes containing RSS values of Φ+2 UEs), a hidden layer (20 neurons), and an output layer (Φ+2 neurons that determine the action $a_{\ell,(t)}$ of the agent). The NN weights are initialized randomly with the distribution $\sim \mathcal{N}(0, 0.03^2)$. The activation function of the hidden layer neurons is tanh(·), whereas for the output layer, it is the softmax(·) function[5] [66]. The output of each output node n is the probability

[4]In our earlier work [29], we used uniformly random selection rather than round-robin selection for the two additional UEs, which also worked well. However, random selection does not completely guarantee that all UEs will be considered, although the probability of some UE not being considered eventually is quite low. The choice to consider specifically two additional UEs was made heuristically.

[5]For $\mathbf{z} = [z_1, z_2, \ldots, z_N] \in \mathbb{R}^N$, $\mathrm{softmax}(\mathbf{z}) = \dfrac{[e^{z_1}, e^{z_2}, \ldots, e^{z_N}]}{\sum_{i=1}^{N} e^{z_i}}$.

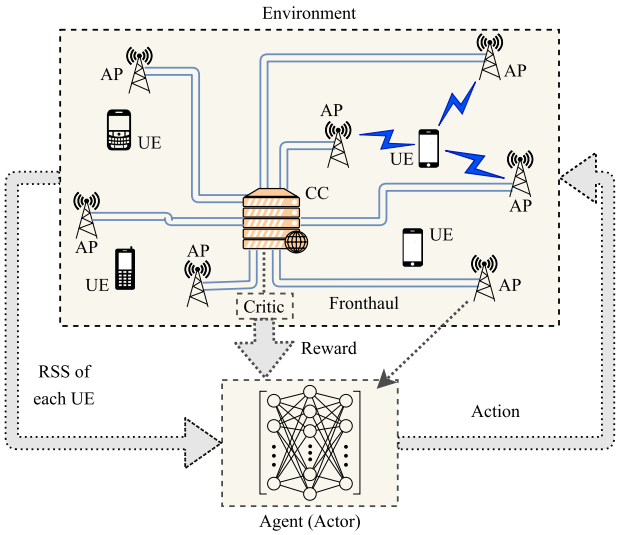


FIGURE 2. Illustration of the decentralized actor, centralized critic MARL algorithm's interactions between the environment and agents.

χ_n of serving the UE corresponding to input node n. At each time step, using $\{\chi_1, \chi_2, \ldots, \chi_{\Phi+2}\}$, the agent calculates the probability of each action from the set of the possible ones. The UE for node n can either be served (with probability χ_n) or not served (with probability $1-\chi_n$), making for $2^{\Phi+2}$ possible actions in total. The agent's action (the set of UEs to be served) is then chosen at random as weighted by the action probabilities. This method of choosing an action by weighted random sampling from the set of possible actions is known as stochastic policy gradient-based action selection [50]; it allows for exploration as well as exploitation of acquired knowledge from earlier training.

The (non-convex) optimization problem of determining which APs should serve which UEs in order to maximize the achievable sum SE can be formulated as

$$\max_{\mathcal{D}_1,\mathcal{D}_2,\ldots,\mathcal{D}_L} \sum_{k=1}^{K} \eta_k \quad (11a)$$

$$\text{subject to: } \Big|\bigcup_{\forall \ell}\mathcal{D}_\ell\Big| = K, \quad (11b)$$

$$\big|\mathcal{D}_\ell\big| \le N,\ \forall \ell\,. \quad (11c)$$

Similarly, the optimization problem to maximize the minimum UE SE would replace (11a) by

$$\max_{\mathcal{D}_1,\mathcal{D}_2,\ldots,\mathcal{D}_L} \min_{k\in\{1,2,\ldots,K\}} \eta_k \quad (12)$$

with the same constraints as in (11b) and (11c). However, finding the globally optimum solution for either optimization problem would need to be done in a centralized manner. Implementing such a solution in a cell-free scenario would result in higher fronthaul loads because the CC would have to transmit its resulting solution to each AP. Instead, a distributed solution can be found at each AP. Moreover, in a mobile environment, the system should in general optimize

the objective function over some interval (such as several sequential time steps t in the set $\mathcal{T}$) in order to account for the movement of the UEs. Therefore, the optimization problems should be modified. We adjust the max sum SE problem as follows:

$$\max_{\mathcal{D}_{\ell,(t)},\forall \ell,\,\forall t\in\mathcal{T}} \sum_{t\in\mathcal{T}}\sum_{\ell=1}^{L}\sum_{k\in\mathcal{D}_{\ell,(t)}} \eta_{k,(t)} \tag{13a}$$

$$\text{subject to: } \Big|\bigcup_{\forall \ell}\mathcal{D}_{\ell,(t)}\Big| = K, \quad \forall t\in\mathcal{T}, \tag{13b}$$

$$\big|\mathcal{D}_{\ell,(t)}\big| \le N, \quad \forall \ell,\ \forall t\in\mathcal{T}, \tag{13c}$$

whereas for the modified max min SE problem, (13a) is replaced by

$$\max_{\mathcal{D}_{\ell,(t)},\forall \ell,\,\forall t\in\mathcal{T}} \min_{k\in\mathcal{D}_{\ell,(t)}} \eta_{k,(t)}. \tag{14}$$

It is important to note that solutions to these optimization problems do not depend solely on the decisions taken by an individual AP, but rather on the decisions made by the cluster of APs that serve a given UE. Observation of the optimization problem in (13) indicates that these decisions depend on the state of the cellular network at the time steps in $\mathcal{T}$. (If some of these time steps are in the future, the actual state may be replaced by the predicted or expected/average state at that time.) Also, the objective function of the optimization problem exhibits an episodic nature, i.e., the objective function depends on values obtained at multiple time steps. Thus, the overall optimization problem can be extended to a Markov game or Markov decision process (MDP). It has been known for some time that RL can be an efficient methodology to solve MDP problems [70]. This was one of our motivations for using RL in the first place.

We consider four reward policies for our MARL algorithm when evaluating its performance:

Policy 1 — Max sum SE: In this case, the reward function for agent ℓ at time t is defined as

$$r_{\ell,(t)} = \sum_{k\in\mathcal{D}_\ell} \eta_{k,(t)} + \mathrm{P}_{\ell,(t)}(|\mathcal{D}_\ell|) + \mathrm{P}_{G(t)}\Big(\Big|\bigcup_{\forall \ell}\mathcal{D}_\ell\Big|\Big), \tag{15}$$

where η_k is given by (7). $\mathrm{P}_{\ell,(t)}(|\mathcal{D}_\ell|)$ is a local penalty function that applies if AP ℓ attempts to serve more than N UEs; if so, a penalty of -10 is incurred. $\mathrm{P}_{G(t)}(|\cup_{\forall \ell}\mathcal{D}_\ell|)$ is the global penalty described in (10); we use $\varrho = -20$. The purpose of the penalties is to prevent illegal or undesirable actions by the agents when creating their policies. The penalty values are thus somewhat arbitrary; any large negative value that negates the potential reward of such actions will suffice.

Understandably, since the goal of Policy 1 is to maximize the sum SE of all UEs, the APs will be biased towards associating with the highest RSS (i.e., nearest) UEs. This reduces the system fairness and more distant UEs might not obtain high quality service. Thus, we also consider another reward function that incorporates fairness.

Policy 2 — Max min SE: In this policy, the agents try to maximize the minimum SE of their served UEs, thus providing fairness in the performance. The reward function in this case is expressed as

$$r_{\ell,(t)} = \min_{k\in\mathcal{D}_\ell} \eta_{k,(t)} + \mathrm{P}_{\ell,(t)}(|\mathcal{D}_\ell|) + \mathrm{P}_{G(t)}\Big(\Big|\bigcup_{\forall \ell}\mathcal{D}_\ell\Big|\Big). \tag{16}$$

The penalties in (16) are the same as in (15). However, if the reward function is expressed as above, without additional constraints such as are typically seen in optimization problems (for example, a constraint that every UE be guaranteed some minimum quality of service), then the agents generally do not learn to each serve multiple UEs. (APs instead prefer serving only one UE each if possible, since that maximizes their minimum, i.e., only, UE SE, although the single UE each AP serves is generally a different one.) To overcome the shortfall of the traditional max min policy, we thirdly use a modified max min SE policy.

Policy 3 — Modified max min SE: The reward function is modified as follows:

$$r_{\ell,(t)} = |\mathcal{D}_\ell| \times \min_{k\in\mathcal{D}_\ell} \eta_{k,(t)} + \mathrm{P}_{\ell,(t)}(|\mathcal{D}_\ell|) + \mathrm{P}_{G(t)}\Big(\Big|\bigcup_{\forall \ell}\mathcal{D}_\ell\Big|\Big). \tag{17}$$

By weighting the minimum UE SE with the number $|\mathcal{D}_\ell|$ of served UEs, the APs learn to serve multiple UEs while still maximizing the minimum SE of the UEs they serve. The penalties in (17) are the same as in (15).

Policy 4 — Hybrid policy[6]: For the sake of interest, we also examine a policy that is a heuristic hybrid of the max SE and max min SE policies. In this policy, the minimum UE SE is weighted by the sum SE of all the agent's served UEs. Because of the presence of the sum SE, the agents still learn to serve multiple UEs. The reward function is as follows:

$$r_{\ell,(t)} = \min_{k\in\mathcal{D}_\ell} \eta_{k,(t)} \times \sum_{k\in\mathcal{D}_\ell} \eta_{k,(t)} + \mathrm{P}_{\ell,(t)}(|\mathcal{D}_\ell|) + \mathrm{P}_{G(t)}\Big(\Big|\bigcup_{\forall \ell}\mathcal{D}_\ell\Big|\Big). \tag{18}$$

The penalties in (18) are the same as in (15). It is expected that this reward function should yield a performance somewhere between the performance of the max SE reward and the performance of the max min SE reward by themselves.

At the completion of training, based upon the final probabilities at the NN output nodes, there may remain a very small but non-zero possibility of choosing an action that serves more than N UEs. To ensure that an agent does not take such an action, we force the probability of those actions to be zero.[7]

V. FEDERATED REINFORCEMENT LEARNING FRAMEWORK

In this section, we develop the MAFRL framework to solve the same AP clustering problem. Unlike in the conventional MARL algorithm, in the MAFRL algorithm the interaction

[6]We called this policy the "max min SE" policy in our previous work [29]. We have renamed it to be a "hybrid" policy here, since, as we will show in the simulation results, the reward function of Policy 3 does a much better job of satisfying the max min SE criterion.

[7]We did not encounter any such actions in our simulations, even without forcing the probabilities to be zero. The enforcement therefore mainly guarantees that such actions will not occur over the long-term timescale of the network operation.

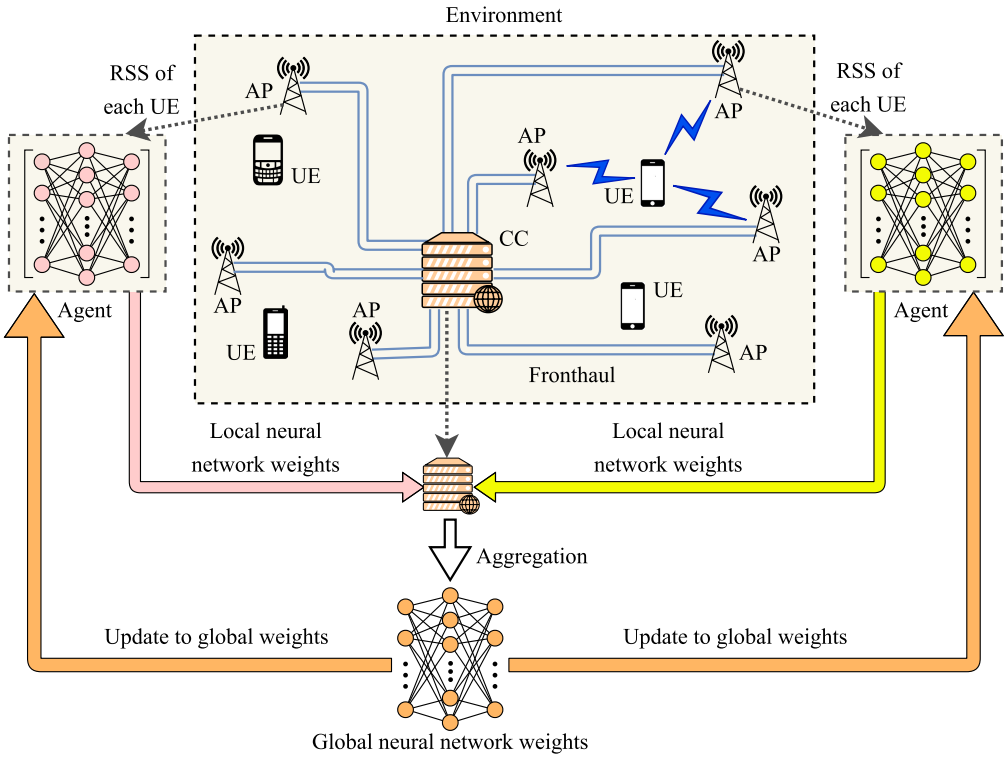


FIGURE 3. Illustration of the MAFRL algorithm's interactions between the central controller, environment, and agents.

between the CC and the APs is now limited to a periodic exchange of NN weights. Therefore, one can no longer use a centralized critic type of reinforcement learning. Instead, we consider a policy-gradient approach [69] to train the agents. We assume that there is a set $\mathcal{L}$ of agents, with each agent having a local state space and action space and using the same reward function. Although the state and action spaces may be different for each agent, the dimensions of the state spaces are the same for every agent, as are the dimensions of the action spaces. The structure of the NN of each agent is the same as in the previous section. Similar to the conventional MARL problem, we formulate the MAFRL problem as a Markov game, represented as a tuple[8] $(\mathcal{S}, \boldsymbol{\mathcal{A}}, \mathcal{P}, r, \gamma)$ [71]. The goal of the MAFRL algorithm is to have the $|\mathcal{L}|$ agents jointly learn a policy function $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ that they all use and that performs as close to optimally as possible and uniformly well across the entire environment. This differs from the MARL algorithm, in which each agent has its own (location-dependent) policy. To reduce the communication overhead, agents do not communicate between themselves; instead, they share their parameter vector $\boldsymbol{\theta}_\ell$ (i.e., their NN weights) only with the CC.

Similar to the previous section, the agents update the weights of their local NNs after each training episode. Each agent aims to find the optimal policy to maximize its cumulative reward in step t, i.e., $r_{\ell,t} = \sum_{t=0}^{\infty} \gamma^i r_{\ell,t+i}$. The state value function of agent ℓ is defined as $V_{\boldsymbol{\theta}_\ell}(s) = \mathbb{E}_{\mathcal{A}_\ell, s_\ell}\{r_{\ell,0} \,|\, s_0{=}s\}$. Mathematically speaking, the goal of each agent is find the policy $\pi^{\star}_{\boldsymbol{\theta}_\ell}$ that maximizes the expected state value function:

$$\pi^{\star}_{\boldsymbol{\theta}_\ell} = \arg\max_{\pi_{\boldsymbol{\theta}_\ell}} \mathbb{E}\{V_{\boldsymbol{\theta}_\ell}(s)\}. \tag{19}$$

Each AP forms a pool of $\Phi{+}2$ UEs as the input to its NN in the same fashion as in the MARL algorithm. Furthermore, each agent in the MAFRL algorithm receives information about Υ_k as feedback from the UEs it is serving. Thereafter, it calculates the SE of each UE using (7). However, it is not possible for the agents themselves to determine if all UEs have been served or not. In the event one or more UEs have not been served, the CC broadcasts a global penalty to all the APs.

For the MAFRL algorithm, we consider an additional alternative max sum SE reward function, which has the local penalty removed compared to Policy 1. We make the assumption here that if an AP serves more than N UEs, then the resulting inter-user interference will increase significantly, resulting in smaller SEs for the UEs and thus a lower reward for the AP. Formally, the alternative reward policy is defined as follows:

Policy 5 — Max sum SE for MAFRL: The reward function for agent ℓ at time t is

$$r_{\ell,(t)} = \sum_{k \in \mathcal{D}_\ell} \eta_{k,(t)} + \mathrm{P}_{G(t)}\Big(\Big| \bigcup_{\forall \ell} \mathcal{D}_\ell \Big|\Big). \tag{20}$$

[8] $\mathcal{L}$ no longer appears in the tuple since every agent is playing a copy of the same game.

After T_{FL} episodes, each agent shares its parameter vector $\boldsymbol{\theta}_\ell$ (NN weights) with the CC. The CC then aggregates the agents' parameter vectors and uses them to calculate updated global NN weights. There are various possible methods of doing this (see e.g., [52], [53], and [54]), but a common way is simply to average $\boldsymbol{\theta}_\ell$ over all agents; we use this average in our work.[9] The global NN weight update is transmitted back to the APs via fronthaul links. The interaction of different components of the MAFRL system is illustrated in Fig. 3.

Even though our MARL and MAFRL algorithms have many similarities, there are several significant differences between them as well, mostly during training. Notably, APs trained using our MAFRL algorithm will all end up with the same NN weights, whereas each AP trained using our MARL algorithm will end up with different localized NN weights. During MARL training, the CC distributes individual rewards (including possible penalties) to each AP. In contrast, during MAFRL training, in every episode the CC broadcasts the global penalty to all APs. Every T_{FL} episodes, the agents sent their NN weights to the CC, which aggregates them and then broadcasts the updated weights to be used by all APs.

A. COMPLEXITY OF MARL AND MAFRL DECISIONS

Concerning the complexity of an AP making a decision on which UEs to serve, we note again that the NN of each agent has only three layers: the input and output layers and a single hidden layer. The number of floating point operations (FLOPs) for each AP to make a decision can be calculated as follows. In the hidden layer, at each of the 20 neurons, the Φ+2 input values are multiplied by a weight, then the weighted values are summed. The sum then passes through the tanh$(\cdot)$ activation function, which requires a bit shift operation (which is simpler than a FLOP) and c_e+3 FLOPs, where c_e is the complexity of calculating e^x of a scalar x (an $\mathcal{O}(1)$ operation). In the output layer, at each of the Φ+2 neurons, the 20 outputs of the hidden layer are again weighted and summed. Then, the vector of those Φ+2 sums is input into the softmax$(\cdot)$ activation function, which uses $(\Phi+2)(c_e+2)$ FLOPs. Thus, in total, $\Phi(c_e+82)+22c_e+224$ FLOPs are required to make a decision. As this number of FLOPs is quite low, the proposed MARL and MAFRL algorithms should not be a challenge for practical implementation. Additionally, with only three layers, the delay involved in computation should be sufficiently small for real-time operation. There is of course additional complexity that occurs during training, but this would happen off-line and not during the regular operation of the network.

[9]Averaging may not be the best choice in certain scenarios, such as if there are significant differences in the distribution of data each agent trains with or in the computing capabilities of each agent. In the case of agents training very large DNNs, they may instead send only a portion of their weights to the CC, e.g., for the last few layers. However, since the NNs in our agents are quite small and the network architecture quite homogeneous, we simply average the entire NN weight vectors for the agents at the CC.

VI. PERFORMANCE EVALUATION

In this section, we examine the simulation results of our MARL and MAFRL algorithms for a cell-free massive MIMO network. We consider $L = 40$ 10-m-tall APs with $N = 4$ antennas each that are uniformly distributed over a geographical area of 1 km $\times$ 1 km. $K = 20$ single-antenna UEs have their locations initialized uniformly over the area. Unless otherwise indicated, we assume that the UEs move around the simulation area at a speed[10] of $v = 1$ m/s. The direction of each UE is initially selected at random isotropically within the range of angles $[0, 2\pi)$ radians; the UEs move in a straight line afterwards, with the movement wrapped around the edges of the simulation area.

We consider a carrier frequency of 2 GHz and channel bandwidth of 20 MHz. The elements of each AP's antenna array are spaced at half a wavelength at the carrier frequency. We neglect any spatial correlation between the antenna elements, i.e., $\mathbf{R}_{k\ell} = \beta_{k\ell}\mathbf{I}_N, \forall k, \ell$. We set (in dB) $\beta_{k\ell} = -30.5 - 36.7\log_{10}(d_{k\ell}) + \Omega_{k\ell}$, where the distance $d_{k\ell}$ (in m) accounts for the AP height of 10 m, and $\Omega_{k\ell} \sim \mathcal{N}(0, 4^2)$ is log-normal shadowing [22]. When UEs are initialized at a distance δ from one another, and whenever a UE moves a distance δ, $\Omega_{k\ell}$ is created/updated with a correlation of $2^{-\delta/(9\text{ m})}$ with the earlier value [22]. The transmitted power of each AP is $P_t = 38$ dBm and the noise power is assumed to be $\sigma^2 = -94$ dBm. K orthogonal pilot sequences are available to the UEs, each with length $\tau_p = K$ and power $\rho_p = 100$ mW. The discount factor for the MARL algorithm is set to $\gamma = 0.95$, which is a typically-used value (e.g., [68], [69]).

We consider a discrete-time system where for the purpose of AP association, the UEs' positions and channels are updated and sampled[11] every 63 ms. Thus, the sampling interval is about the same as the channel coherence time $t_c = 0.423\lambda/v$ [73, Eq. (5.40.c)], where λ is the carrier wavelength. In RL terminology, these samples are the steps, and we consider 80 steps during one episode of training. This corresponds to a UE travel distance of 5.04 m at 1 m/s speed. For this relatively small distance, the assumption of UEs moving in a straight line is reasonable.[12] After each episode, the UEs' locations and directions are reset randomly, but the AP locations stay the same. The NN weights for each agent

[10]In this work, we limit the examination to pedestrian speeds, because considering vehicular speeds would result in the channel estimates becoming increasingly inaccurate. Depending on the carrier frequency and UE speed, the channel coherence time could diminish sufficiently so that the channel could no longer be considered constant within a TDD frame. As such, channel prediction would be needed along with CSI estimation. We have begun to investigate networks with UEs moving at vehicular speeds in some of our other work, e.g., [72].

[11]With a sample period of 63 ms and a UE speed of 1 m/s, the UEs thus move a distance δ = 63 mm when updating $\Omega_{k\ell}$ between samples.

[12]If the distance traveled per UE per episode was longer, alternative models for the UE movement could be more appropriate, such as along a grid in an urban area, along some predefined paths, according to a random walk model (see e.g., [74]), or by a machine-learned model [75]. However, such more complicated UE movement models are not necessary in this article and are outside of its main focus.

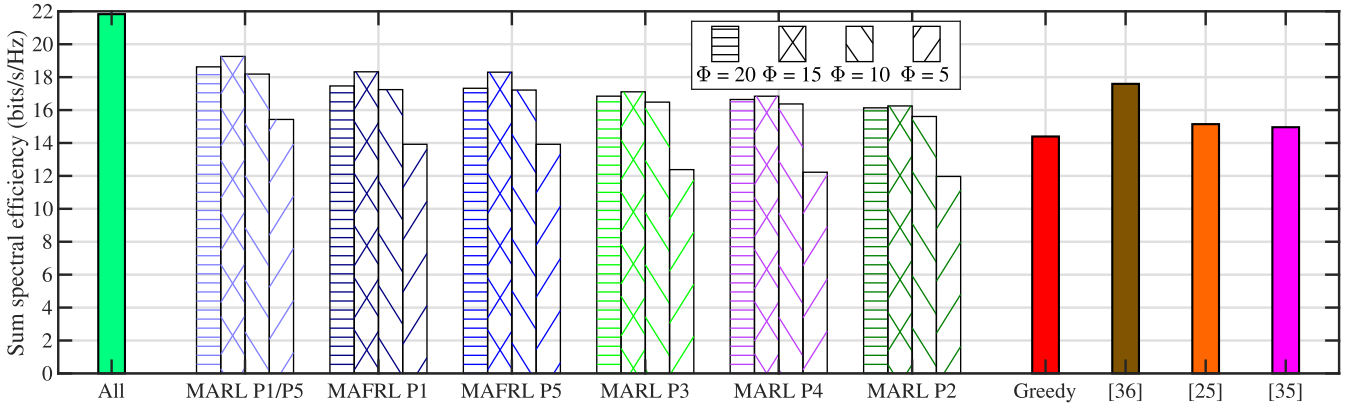


FIGURE 4. Average sum SE performance of MARL and MAFRL algorithms with several values of Φ, compared against "All" and "Greedy" strategies, max min SINR strategy from [25], κ-means clustering ML algorithm from [35], and modified RL-based power control algorithm from [36]. "P1": Max sum SE policy having local and global penalties, "P2": Max min SE policy, "P3": Modified max min SE policy, "P4": Hybrid policy, "P5": Max sum SE policy having global penalty only.

are updated after each episode. For the MAFRL algorithm, the global update of weights at the CC occurs every $T_{\text{FL}} = 20$ episodes. Agents are trained for 4000 episodes and thereafter their performance is evaluated for 40 test cases (each being a new episode with the NN weights fixed). We repeat this procedure for 10 independent simulation runs.

The performance results averaged over the $40 \times 10 = 400$ total test cases are compared against five existing strategies: 1) "All": UEs are served by all the APs, and coordinated centralized precoding is done rather than distributed precoding, thus representing the maximum possible performance; 2) "Greedy": each AP serves the N highest-RSS (nearest) UEs; 3) the max min SINR method proposed in [25]; 4) the κ-means clustering ML algorithm proposed in [35]; 5) a modified version of the RL-based power control method proposed in [36]. In the case of [36], the authors had originally considered APs equipped with a single antenna each, and UEs served by all APs with centralized precoding. For a fair performance comparison with the other schemes, we modified the method from [36] for multi-antenna APs with LP-MMSE precoding first by adding the global penalty from our reward policies to its reward function. The use of localized precoding implies that each AP should serve no more than N UEs. However, it does not by itself ensure that the APs learn to serve a maximum of N UEs, because if the power allocated to some UE is very small, the resulting effect on the sum SE would be negligible. Hence, the algorithm would not be able to learn properly whether that action is better or worse. Therefore, we additionally defined a threshold such that if the power allocated to a given UE is less than 1% of the AP's total transmit power, then it is considered that the given UE and that AP are not associated. This threshold for the scheme modified from [36] helps limit the number of served UEs to be at most N.

The average sum SE performance of the MARL and MAFRL algorithms with several values of Φ and the five policies is illustrated in Fig. 4. As expected, if UEs are served by all the APs with centralized precoding, then the sum SE is maximum (21.8 bits/s/Hz), but so too is the fronthaul load. Our MARL algorithm under Policy 1 with $\Phi = 10$ and LP-MMSE precoding achieves about 18.2 bits/s/Hz, or about 83.3% of the max SE; this increases to about 88.3% (19.3 bits/s/Hz) using $\Phi = 15$. As seen, increasing Φ improves the sum SE performance, but our algorithms require more training episodes to converge properly. This can be seen in the $\Phi = 20$ result; in this case, 4000 episodes is insufficient for training because of the large action space for $\Phi = 20$.

The results of the MAFRL algorithm are similar to, though marginally less than, those provided by MARL algorithm. We first observe that the MAFRL algorithm's results using Policy 1 and Policy 5 are nearly identical, confirming that the MAFRL algorithm does not need the local penalty as in Policy 1 when maximizing the sum SE. We also observe that the MAFRL algorithm's sum SE is about 90−95% of (or about 0.95−1.5 bits/s/Hz less than) that of the MARL algorithm. The main reason for the worse MAFRL performance is because the NN weights of the MAFRL agents are not optimized to their individual locations; rather, the global average is optimized. Thus, the MAFRL algorithm trades off some locally optimized higher performance in favor of consistently good performance over the entire coverage area.

Given the similarity in MAFRL performance between Policies 1 and 5, as an additional test, we also checked the performance of Policy 5 when used with the MARL algorithm, even though that policy was designed for the MAFRL algorithm. We found the MARL algorithm performance is also virtually identical for both Policies 1 and 5, which demonstrates that with localized precoding, the sum SE reduces when an AP serves more than N UEs. Thus, the agents can learn to serve only N UEs even without the local penalty in the reward when the goal is to maximize the sum SE. Since the performances of the MARL and MAFRL algorithms are nearly identical under Policy 1 as they are under Policy 5, hereafter we will just

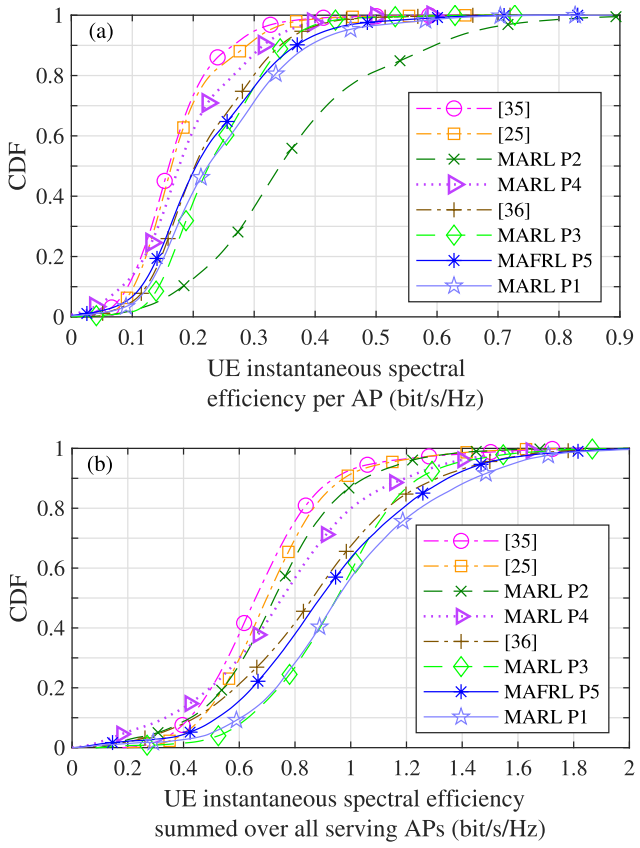


FIGURE 5. CDFs of UEs' instantaneous SEs for MARL and MAFRL algorithms with $\Phi = 15$ under five reward policies, compared with schemes from [25], [35], and [36]. "P1": Max sum SE policy having local and global penalties, "P2": Max min SE policy, "P3": Modified max min SE policy, "P4": Hybrid policy, "P5": Max sum SE policy having global penalty only. (a) UE SEs per AP. (b) UE SEs summed over their serving APs.

depict results for MARL using Policy 1 and MAFRL using Policy 5.

Considering the reference algorithms, we observe that the modified RL-based power control algorithm from [36] performs the best and achieves a sum SE of about 17.6 bits/s/Hz, or about 81% of the "All" case. However, we note that our MARL and MAFRL algorithms employ equal power allocation to all the UEs, yet with $\Phi = 15$ they outperform [36]. This is because the APs learn better to account for UE mobility, whereas the method from [36] does not account for mobility. If we were to incorporate power allocation along with our MARL and MAFRL algorithms, it can be expected that their performance would be further improved. We furthermore note that the higher performance of our algorithms comes via considerably less complex NNs than the NNs used with the method from [36]. For instance, our agent NNs have a single hidden layer with 20 neurons, whereas the (multiple) layers in [36] are of size 400×300.

We next note that considering just the UEs with the highest RSS values or serving the closest UEs together is far from an optimal AP clustering solution to maximize the sum SE, as observed from the performance for the schemes from [25] and [35], and the $\Phi = 5$ results for MARL under Policy 1 and MAFRL under Policy 5. These respectively achieve only about 68.5%, 69.4%, 70.7%, and 63.8% of the sum SE of the "All" case. "Greedy" selection performs the worst of the reference schemes, as it is not optimized to maximize the SE globally,[13] though the MAFRL algorithm's performance with $\Phi = 5$ is the lowest overall among the algorithms intended to maximize the sum SE. The lower the value of Φ, the more localized the selection of UEs is around a given AP, and thus in some sense the whole performance is also based on more localized conditions.

Interestingly, Policies 2 and 3 provide sum SEs not much below that of Policies 1 and 5. They also provide a considerably better sum SE than the scheme from [25], despite all three nominally having a "max min" goal. With $\Phi = 15$, Policy 3 yields a sum SE of about 18.3 bits/s/Hz, while Policy 2 yields a sum SE of about 17.3 bits/s/Hz. The sum SE of Policy 4 is in between that of Policies 2 and 3, about 17.8 bits/s/Hz. It should be noted, though, that Policies 2–4 perform particularly poorly with low values of Φ. For these three policies, Φ must be large enough so that the APs can find the UE with the minimum SE in a wider area around their vicinity. Φ being too small (e.g., $\Phi = 5$) results in the APs considering too small of a neighborhood around their respective locations for the algorithm to properly increase the minimum UE SEs, and thus the sum SE by extension. We lastly note that we also tested Policies 2, 3, and 4 with the MAFRL algorithm, and they displayed a similar small drop in performance relative to the MARL algorithm as was seen with Policies 1 and 5. Therefore, we do not depict the MAFRL results for Policies 2–4, since the drop in performance with Policies 1 and 5 is representative of all the policies.

Fig. 4 does not provide insight about the fairness of the MARL and MAFRL algorithms. Therefore, in Fig. 5 we examine the cumulative distribution functions (CDFs) of the UEs' instantaneous SEs for the five reward policies with $\Phi = 15$, compared with the CDFs for the schemes from [25], [35], and [36]. Fig. 5(a) shows the UE SEs per AP, whereas Fig. 5(b) shows the UE SEs summed over their respective serving APs. We differentiate between the two because our RL reward policies apply separately at each individual AP, as do the power allocations from [36], whereas [25] and [35] apply more to the system as a whole. We observe that Policies 1 and 5, which maximize the sum SE, provide the

[13]The "Greedy" algorithm selects UEs with the highest RSS values for a given AP. This selection would be the optimal one in terms of maximizing the sum SE if each AP only had one antenna and served a single UE each [76]. However, it is no longer optimal when each AP has multiple antennas and serves multiple UEs, and each UE is served by multiple APs. Instead, other factors such as the orthogonality between the channel vectors of UEs must be considered, on account of the interference their signals cause on each other. A greedy algorithm that maximizes the incremental SE provided by each UE it selects in succession can still provide a near-maximal sum SE. However, using only RSS values, serving the UEs with highest RSS often results in higher multiuser interference, and correspondingly lower sum SE.

highest median and 95th percentile total UE SEs in Fig. 5(b), as expected. When each individual AP maximizes the SE of the UEs it serves, that also maximizes the total SE they receive from all their serving APs. The MAFRL algorithm's CDF is slightly to the left of the one for the MARL algorithm, reflecting its slightly worse performance seen in Fig. 4. The MARL algorithm under Policy 1 provides median and 95th percentile UE SEs per AP of about 0.22 bits/s/Hz and 0.45 bits/s/Hz, respectively; the median and 95th percentile of the UE total SEs are about 0.95 bits/s/Hz and 1.59 bits/s/Hz, respectively. In comparison, the MAFRL algorithm under Policy 5 yields median and 95th percentile UE SEs per AP of about 0.20 bits/s/Hz and 0.42 bits/s/Hz, respectively; for UE total SEs, they are about 0.89 bits/s/Hz and 1.48 bits/s/Hz, respectively. We also note Policy 3 yields about the same median SE as Policy 1.

The results for Policy 2 are rather unusual compared to the other schemes. Fig. 5(a) indicates that this policy results in the highest UE SEs per AP out of any of the schemes. However, this is because under Policy 2, the APs only learn to serve a single UE each at a time, and UEs are served by only one or two APs in total. This behaviour is largely due to the fact that, although the reward of Policy 2 is based on the total SE a UE receives, each AP does not know the specific actions taken at other APs, only the net result of all those actions (including its own). It therefore cannot properly differentiate whether or not the reward is solely due to its own actions. This distinction is irrelevant when maximizing the sum SE as in Policies 1 and 5, but is more important when maximizing the minimum SE. Note that the policy does indeed achieve its goal of each AP maximizing the minimum SE provided to its served UEs (its 5th percentile UE SE per AP is 0.14 bits/s/Hz), but it does so by allocating all its resources to that single UE. This could potentially be problematic with more UEs in the system. In examining Fig. 5(b), it can be observed that Policy 2 generally yields the lowest total UE SEs out of our five policies. However, the CDF of UE total SEs for Policy 2 is also one of the steepest out of our five policies, meaning that it provides lower variation/more uniformity in total SEs among the UEs.

In comparison, the modification made in Policy 3 allows the agents to learn to serve multiple UEs, and also results in the highest of the minimum total UE SEs out of all the examined schemes. The 5th percentile total UE SE of Policy 3 is about 0.55 bits/s/Hz. This performance is achieved by trading off the SE given to the higher SE UEs; the upper percentiles are worse for Policy 3 than for Policies 1 and 5. Like Policy 2, Policy 3 also has a steep CDF, meaning that there again is lower variation/higher uniformity in total SEs among the UEs.

The results for Policy 4 indicate that the hybrid reward function does not end up working particularly well at either the lower or the upper end of the CDFs. Neither the sum SE nor the minimum SE ends up maximized. However, Policy 4 does in general perform better than Policy 2 in terms of total UE SEs; their CDFs cross each other at about the 35th percentile. Policy 4 provides better performance at the middle and upper end of the CDFs, whereas Policy 2 provides better performance at the lower end. We finally note that in regard to our policies, much like for Policies 1 and 5, the MAFRL algorithm provides slightly worse performance than the MARL algorithm under Policies 2–4 as well. (We do not depict the MAFRL algorithm's performance with Policies 2–4 in Fig. 5 in order to avoid obscuring the other results.)

In terms of instantaneous SE performance, among the reference algorithms, the modified RL-based algorithm from [36] provides the highest SEs. Its performance for instantaneous SE per AP in Fig. 5(a) is quite close to that of MAFRL under Policy 5, but with a somewhat fairer (less varied) distribution of SEs among UEs. This is likely an indication of the power allocation part of the scheme from [36] diverting power from certain UEs and APs to other ones. The results for SE summed over all APs in Fig. 5(b) show more of a difference between the MAFRL algorithm and the modified method from [36]; the summed SEs provided by the former are larger than those of the latter, even for low-SE UEs. This indicates that the modified scheme from [36] likely allocates most power to a UE at a single "best" AP, and significantly less at other APs. In both Figs. 5(a) and (b), the SEs provided by the MARL algorithm under Policy 1 are higher than those provided by the modified scheme from [36] across almost the entire distribution. It can lastly be observed from Fig. 5 that the CDFs for the schemes from [25] and [35] lie mostly to the left of those of our MARL and MAFRL algorithms. The exception is at the bottom-left of the UE total SE CDFs; those two schemes provide better lower total UE SEs than our Policies 2 and 4. Hence, the schemes from [25] and [35] provide the lowest SEs to most of the UEs out of any of the examined schemes. This reflects the fact that those two schemes also provide the lowest sum SE for the system, as seen in Fig. 4.

The convergence of the MARL algorithm can be proven following steps similar to those in [67]. Similarly, [77] provides upper bounds on the convergence rate of FL when global NN updates are calculated as the average of the local NN weights. It is therefore unnecessary to duplicate such proofs of convergence here. Instead, in Table 1, we compare the MARL and MAFRL performance with $\Phi = 10$ and their max sum SE policies when varying the number of training episodes, to investigate how many episodes are required for the NNs to converge. From the results, we observe that the MARL sum SE oscillates initially, but slowly it stabilizes around 18.2−18.3 bits/s/Hz after about 4000 episodes. The same oscillatory behaviour is seen with the MAFRL algorithm, but it too stabilizes after roughly the same amount of training, this time to around 17.2−17.3 bits/s/Hz. The convergence with Policies 2–4 is similar for both algorithms.

We have also observed that a UE is served by a mean of 3.82 APs with a standard deviation of 0.52 by the MARL algorithm under Policy 1; the results for the MAFRL algorithm under Policy 5 are similar. Under Policy 3, UEs are served by a mean of 3.8 APs with a standard deviation of

TABLE 1. Sum SE for $\Phi = 10$, max sum SE policies, and varying number of training episodes

Number of training episodes	MARL P1 Sum SE (bits/s/Hz)	MAFRL P5 Sum SE (bits/s/Hz)
500	10.6	11.1
1000	12.5	12.7
1500	14.2	14.2
2000	14.7	14.5
2500	17.7	16.3
3000	16.9	16.8
3500	17.9	17.2
4000	18.2	17.2
4500	18.2	17.3
5000	18.3	17.3

0.35 by the MARL algorithm, whereas under Policy 4, UEs are served by a mean of 3.76 APs with a standard deviation of 0.68 by the MARL algorithm. Hence, Policies 3 and 4 are comparable in this regard to Policies 1 and 5, but with Policy 3 having a bit less variation, and Policy 4 a bit more. In sharp contrast, under Policy 2, UEs are only served by a mean of 1.71 APs, with a standard deviation of 0.24, for the reasons explained earlier. We have additionally confirmed that the set of APs that serves each UE changes as the UEs move through the coverage area. This further demonstrates that the proposed MARL and MAFRL algorithms can indeed properly handle UE mobility while ensuring near-optimal performance, enabling UEs to be connected to 4 APs most of the time (with the exception of Policy 2).

Up to now, the results have suggested that the MARL algorithm outperforms the MAFRL algorithm. This is indeed true when the algorithms are used in the exact same environment they are trained in. However, we next consider a scenario where, after training, the algorithms are transferred to a different environment where the APs' locations are different than those during training. Specifically, both the APs' and UEs' initial locations are randomized, with the results averaged over 400 test cases. This lets us examine how well the algorithms transfer their learning. Fig. 6 shows the performance of the MARL and MAFRL algorithms with max sum SE policies, where it can be observed that the MAFRL algorithm now has a significant advantage. The performance of the MARL algorithm drops considerably in the new environment, between about 1.9 to 4.2 bits/s/Hz, or by 10−27%, compared to its performance in the training environment. In contrast, there is almost no change in the MAFRL performance. Its sum SE drops by at most about 0.2 bits/s/Hz, which is too small for a clear depiction in the figure. Thus, the MARL algorithm's performance can be highly dependent on the agents' locations. The $\Phi = 5$ case is particularly vulnerable, because the chosen UEs in that case tend to be those closest to the AP. As such, the agents do not get as broad a sense of the overall UE conditions as they do with higher values of Φ. For the reference algorithms, the scheme from [25] experiences no change in the different environment, which

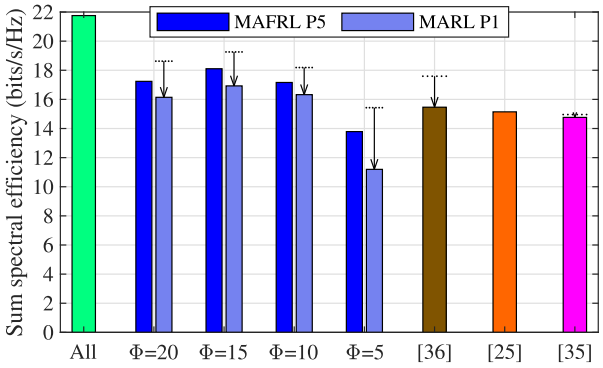


FIGURE 6. Average sum SE performance of MARL and MAFRL algorithms under max sum SE policies with several values of Φ, max min SINR strategy from [25], κ-means clustering ML algorithm from [35], and modified RL-based power control algorithm from [36], when the AP locations during testing differ from those during training. Arrows show drop in MARL performance compared to when the training and testing environments match; performance of MAFRL algorithm is mostly unaffected. Performance of "All" strategy also shown for reference. "P1": Max sum SE policy having local and global penalties, "P5": Max sum SE policy having global penalty only.

is understandable — its max min SINR metric considers the global environment to begin with. There is also a very small drop in the performance of the scheme from [35] in the new environment, just sightly larger than for our MAFRL algorithm. However, there is a significant dependency[14] of the modified deep RL-based algorithm from [36] on the training environment similar to that of our MARL algorithm. Consequently, that scheme also sees a similar loss in performance in the new environment like our MARL algorithm does.

We next are interested in examining how fast the UEs can move before the performance of our algorithms significantly deteriorates. The sum SE performance for UE speeds of $v = \{1, 1.5, 2, 2.5, 5\}$ m/s is depicted in Fig. 7. The first four values cover the range of walking to jogging, while the highest speed that we considered corresponds to a fast run or leisurely bicycle ride. Importantly, the results shown are all for our algorithms having been trained using a UE speed of 1 m/s, but tested on different speeds. We do not show results for the other reference algorithms in this case, because as they do not explicitly account for UE mobility, their performance does not change significantly at the tested speeds; there is just a slight degradation in performance as v increases. From Fig. 7, it can be observed that there is no significant change in the performance of our algorithms up to $v = 2.5$ m/s. At 2.5 m/s, the drop in performance relative to 1 m/s is less than 1%. It is only at $v = 5$ m/s that a notable deterioration in performance can be seen. In this case, the sum SE drops by about 5−6% for all of the reward policies. Even still, the performance of our MARL algorithm under Policy 1 (18.2 bits/s/Hz at 5 m/s) remains higher than that of

[14]The authors of [36] have noted this dependency on the training environment in their paper. Their results have circumvented the issue by using data from multiple environments when training.

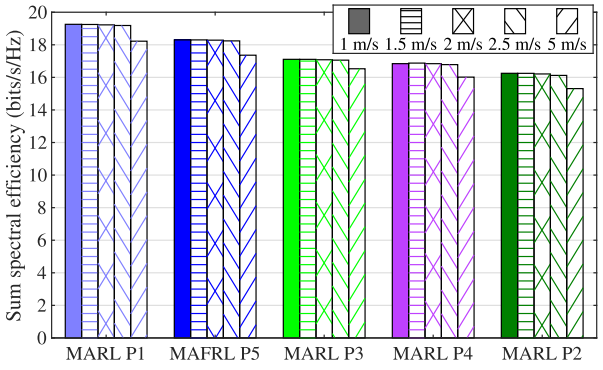


FIGURE 7. Average sum SE performance of MARL and MAFRL algorithms with $\Phi = 15$ and several values of UE speed v, having been trained at 1 m/s. "P1": Max sum SE policy having local and global penalties, "P2": Max min SE policy, "P3": Modified max min SE policy, "P4": Hybrid policy, "P5": Max sum SE policy having global penalty only.

the next-best algorithm from [36] (17.6 bits/s/Hz, as seen in previous figures.) The MAFRL performance at 5 m/s under Policy 5 (17.4 bits/s/Hz) is also only slightly worse. We have additionally examined the case of our proposed algorithms being both trained and tested at $v = 5$ m/s. In this event, the algorithms' performance returns to the same as if they were both trained and tested at 1 m/s. Moreover, training at 5 m/s and testing at 1 m/s again results in a decline in performance; the MARL performance under P1 drops to 18.9 bits/s/Hz, for example. This indicates that the degradation in performance is a result of the mismatch between the training and testing environments (much like what was seen in Fig. 6), rather than an inability of the proposed algorithms to handle higher UE speeds. It also suggests that training with a variety of UE speeds ought to result in a bit better performance.

Next, we investigate the impact of varying the ratio of the total number of antenna elements at the APs to the number of UEs on the sum SE performance. For this, we examine the sum SE performance for two different scenarios: a) varying the number of antennas N per AP while keeping the number of UEs fixed, and b) varying the number of UEs K while keeping the number antennas per AP fixed. The performance when varying N is illustrated in Fig. 8(a). We vary N from 2 to 6, and compare the performance of the MARL and MAFRL algorithms with the five reward policies, both using $\Phi = 15$, against the same existing strategies as in Fig. 4. We observe that the performance of both our proposed algorithms under Policies 1 and 5 is very close to the maximum performance of the "All" case when $N = 2$. This is understandable, because for smaller antenna array sizes, the agents/APs serve fewer UEs; thus, the likelihood of making an optimal UE selection is considerably higher since the search space is much smaller. For $N = 2$, the MAFRL and MARL algorithms obtain about 97−98% of the maximum possible SE of the "All" strategy, whereas for $N = 3$, they obtain about 91−92%. For larger values of N, the slopes of the curves stabilize, such that the SE obtained by our algorithms is consistently about 83−89% of the maximum.

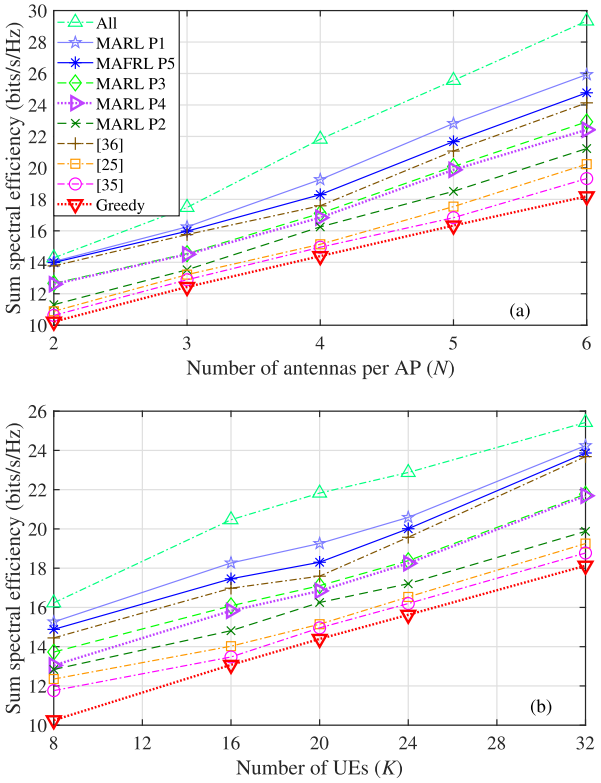


FIGURE 8. Average sum SE performance of MARL and MAFRL algorithms with $\Phi = 15$ and varying ratios of total number of AP antennas to number of UEs. Performance is compared against "All" and "Greedy" strategies, max min SINR strategy from [25], κ-means clustering ML algorithm from [35], and modified RL-based power control algorithm from [36]. "P1": Max sum SE policy having local and global penalties, "P2": Max min SE policy, "P3": Modified max min SE policy, "P4": Hybrid policy, "P5": Max sum SE policy having global penalty only. (a) $K = 20$ UEs, varying numbers of antennas per AP (N). (b) $N = 4$, varying number of UEs (K).

Compared to the modified algorithm from [36], the performance of MARL algorithm with Policy 1 is about 7−9% better for $N \geq 4$. We again note that our better performance is with equal power distribution among the UEs; the addition of power allocation should further enhance the MARL (and MAFRL) performance. At the same time, for $N \geq 4$, the MARL algorithm's SE under Policy 1 is at least about 21−23% larger than the other three reference algorithms, while the MAFRL algorithm's SE under Policy 5 is about 17−19% larger. Policies 3 and 4 provide about the same sum SE as each other (Policy 3 is marginally better) and their performance improves with N at about the same rate as Policies 1 and 5. Policy 2 continues to have the worst performance among our reward policies, though it remains better than that of all the reference schemes other than the modified one from [36]. Interestingly, although not shown in the figure, we have observed that the performance of Policy 2 improves slower with N when $\Phi = 10$ is used, such that the scheme from [25] catches up at $N = 6$ in that case.

The performance when varying the number of UEs K is illustrated in Fig. 8(b). In this case, we vary the number of

UEs from 8 to 32 in steps of 8; we also include the previous results for 20 UEs. The relative performance of all the compared algorithms remains the same as in Fig. 8(a). Moreover, the rate in increase of SE with K for the SE-maximizing algorithms is roughly the same among that group, as is the rate of increase among the max-min SE algorithms. Unsurprisingly, the SE grows slower with K for the latter group than it does for the former. This simply reflects that the SE-maximizing algorithms can better exploit multiuser diversity. In contrast, the max-min SE algorithms have to trade off some total SE to improve the performance of additional UEs in the system in relatively poorer channel conditions; hence, their sum SE cannot increase as quickly with K.

VII. CONCLUSION

In this work, we have proposed MARL and MAFRL AP clustering algorithms for cell-free massive MIMO systems. We have described the mathematical details for obtaining the CSI and precoding vectors for each AP. The proposed algorithms' performance has been examined for five reward policies and compared with several existing strategies. It has been demonstrated that our MARL algorithm outperforms the other AP clustering strategies, and achieves up to 88.3% of the maximum possible sum SE achievable if all APs were to serve all UEs using centralized precoding. Our MAFRL algorithm performs slightly worse than our MARL algorithm (about 5−10% lower SE) on account of trading off some localized performance gains in favor of uniformly good performance across the entire coverage area. However, that tradeoff also means the MAFRL algorithm can transfer its learning to different environments much better than the MARL algorithm; the latter instead tends to develop dependencies on the training environment. When the AP locations are different during testing than they were during training, the MARL algorithm performance drops significantly (up to 27% lower SE in one case), whereas the MAFRL performance is almost unchanged. A similar but much smaller drop in performance (about 5−6%) occurs for both algorithms when the UE speeds during testing differ significantly from those used for training (e.g., 5 m/s vs. 1 m/s or vice versa). The relative performances of all the examined algorithms also remain about the same when the number of antennas per AP is equal to or greater than 4 and when the number of UEs varies between 8 and 32.

Several extensions of this work are possible, such as tweaking the hyperparameters of the agents' NNs for better performance. The operation of the MARL and MAFRL algorithms using measured channel data or a ray-traced environment model could be examined. The performance of other types of ML algorithms can also be studied. In our simulations, the ability to adapt the UE SEs to maximize the rewards of the RL policies is somewhat lessened by the use of equal power allocation for the UEs. It would therefore be useful to examine implementing a power allocation method along with our proposed AP clustering algorithms, possibly together within the same type of MARL/MAFRL framework. We also plan on investigating a higher-mobility environment, i.e., with much higher UE speeds than the pedestrian speeds considered herein.

ACKNOWLEDGMENT

The authors would like to thank Dr. Hamid Farmanbar, who was their Huawei collaborator and coauthor of their earlier conference paper [29] on this topic. They would also like to thank Dr. Emil Björnson for the open-source code repository for [21], which helped them to develop the channel model and precoding calculations.

REFERENCES

[1] 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.

[2] 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.

[3] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct. 2014.

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

[5] W. Choi and J. Andrews, "Downlink performance and capacity of distributed antenna systems in a multicell environment," *IEEE Trans. Wireless Commun.*, vol. 6, no. 1, pp. 69–73, Jan. 2007.

[6] X.-H. You, D.-M. Wang, B. Sheng, X.-Q. Gao, X.-S. Zhao, and M. Chen, "Cooperative distributed antenna systems for mobile communications," *IEEE Wireless Commun.*, vol. 17, no. 3, pp. 35–43, Jun. 2010.

[7] D. Gesbert, S. Hanly, H. Huang, S. Shamai (Shitz), O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: A new look at interference," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 9, pp. 1380–1408, Dec. 2010.

[8] J.-M. Moon and D.-H. Cho, "Efficient cell-clustering algorithm for intercluster interference mitigation in network MIMO systems," *IEEE Commun. Lett.*, vol. 15, no. 3, pp. 326–328, Mar. 2011.

[9] K. Hosseini, W. Yu, and R. S. Adve, "Large-scale MIMO versus network MIMO for multicell interference mitigation," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 930–941, Oct. 2014.

[10] R. Irmer et al., "Coordinated multipoint: Concepts, performance, and field trial results," *IEEE Commun. Mag.*, vol. 49, no. 2, pp. 102–111, Feb. 2011.

[11] D. Lee et al., "Coordinated multipoint transmission and reception in LTE-advanced: Deployment scenarios and operational challenges," *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 148–155, Feb. 2012.

[12] J. Lee et al., "Coordinated multipoint transmission and reception in LTE-advanced systems," *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 44–50, Nov. 2012.

[13] V. Jungnickel et al., "The role of small cells, coordinated multipoint, and massive MIMO in 5G," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 44–51, May 2014.

[14] D. Jaramillo-Ramírez, M. Kountouris, and E. Hardouin, "Coordinated multi-point transmission with imperfect CSI and other-cell interference," *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 1882–1896, Apr. 2015.

[15] B. Dai and W. Yu, "Sparse beamforming and user-centric clustering for downlink cloud radio access network," *IEEE Access*, vol. 2, pp. 1326–1339, 2014.

[16] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, "Cloud radio access network (C-RAN): A primer," *IEEE Netw.*, vol. 29, no. 1, pp. 35–41, Jan./Feb. 2015.

[17] A. Checko et al., "Cloud RAN for mobile networks—A technology overview," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 405–426, 1st Quart., 2015.

[18] C. Pan, M. Elkashlan, J. Wang, J. Yuan, and L. Hanzo, "User-centric C-RAN architecture for ultra-dense 5G networks: Challenges and methodologies," *IEEE Commun. Mag.*, vol. 56, no. 6, pp. 14–20, Jun. 2018.

[19] M. M. Abdelhakam, M. M. Elmesalawy, K. R. Mahmoud, and I. I. Ibrahim, "A cooperation strategy based on bargaining game for fair user-centric clustering in cloud-RAN," *IEEE Commun. Lett.*, vol. 22, no. 7, pp. 1454–1457, Jul. 2018.

[20] 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.

[21] Ö. T. Demir, E. Björnson, and L. Sanguinetti, "Foundations of user-centric cell-free massive MIMO," *Found. Trends Signal Process.*, vol. 14, nos. 3–4, pp. 162–472, 2021.

[22] 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.

[23] S. Buzzi and C. D'Andrea, "Cell-free massive MIMO: User-centric approach," *IEEE Wireless Commun. Lett.*, vol. 6, no. 6, pp. 706–709, Dec. 2017.

[24] G. Interdonato, E. Björnson, H. Q. Ngo, P. Frenger, and E. G. Larsson, "Ubiquitous cell-free massive MIMO communications," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 197, pp. 1–13, Dec. 2019.

[25] C. F. Mendoza, S. Schwarz, and M. Rupp, "Cluster formation in scalable cell-free massive MIMO networks," in *Proc. 16th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2020, pp. 62–67.

[26] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4247–4261, Jul. 2020.

[27] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. V. Srinivas, "User-centric cell-free massive MIMO networks: A survey of opportunities, challenges and solutions," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 611–652, 1st Quart., 2022.

[28] Y. Zhao, I. G. Niemegeers, and S. M. H. De Groot, "Dynamic power allocation for cell-free massive MIMO: Deep reinforcement learning methods," *IEEE Access*, vol. 9, pp. 102953–102965, 2021.

[29] B. Banerjee, R. C. Elliott, W. A. Krzymień, and H. Farmanbar, "Access point clustering in cell-free massive MIMO using multi-agent reinforcement learning," in *Proc. IEEE 33rd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2022, pp. 1086–1092.

[30] Y. Cao, S.-Y. Lien, Y.-C. Liang, K.-C. Chen, and X. Shen, "User access control in open radio access networks: A federated deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 3721–3736, Jun. 2022.

[31] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *Proc. NIPS Workshop Private Multi-Party Mach. Learn.*, Barcelona, Spain, Dec. 2016, pp. 1–10.

[32] A. Hard et al., "Federated learning for mobile keyboard prediction," Feb. 2019, *arXiv:1811.03604*.

[33] X. Wang, R. Li, C. Wang, X. Li, T. Taleb, and V. C. M. Leung, "Attention-weighted federated deep reinforcement learning for device-to-device assisted heterogeneous collaborative edge caching," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 154–169, Jan. 2021.

[34] M. Chen et al., "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3579–3605, Dec. 2021.

[35] S. Biswas and P. Vijayakumar, "AP selection in cell-free massive MIMO system using machine learning algorithm," in *Proc. 6th Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2021, pp. 158–161.

[36] L. Luo, J. Zhang, S. Chen, X. Zhang, B. Ai, and D. W. K. Ng, "Downlink power control for cell-free massive MIMO with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 6772–6777, Jun. 2022.

[37] K. T. Truong and R. W. Heath Jr., "The viability of distributed antennas for massive MIMO systems," in *Proc. 47th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2013, pp. 1318–1323.

[38] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO: Uniformly great service for everyone," in *Proc. IEEE 16th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2015, pp. 201–205.

[39] G. Femenias and F. Riera-Palou, "Cell-free millimeter-wave massive MIMO systems with limited fronthaul capacity," *IEEE Access*, vol. 7, pp. 44596–44612, 2019.

[40] P. Marsch and G. Fettweis, "Uplink CoMP under a constrained backhaul and imperfect channel knowledge," *IEEE Trans. Wireless Commun.*, vol. 10, no. 6, pp. 1730–1742, Jun. 2011.

[41] J. Kang, O. Simeone, J. Kang, and S. Shamai, "Layered downlink precoding for C-RAN systems with full dimensional MIMO," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2170–2182, Mar. 2017.

[42] H. Masoumi and M. J. Emadi, "Performance analysis of cell-free massive MIMO system with limited fronthaul capacity and hardware impairments," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1038–1053, Feb. 2020.

[43] Y. Gao, W. Jiang, and T. Kaiser, "Bidirectional branch and bound based antenna selection in massive MIMO systems," in *Proc. IEEE 26th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Aug. 2015, pp. 563–568.

[44] R. Hamdi, E. Driouch, and W. Ajib, "Resource allocation in downlink large-scale MIMO systems," *IEEE Access*, vol. 4, pp. 8303–8316, 2016.

[45] M. O. K. Mendonça, P. S. R. Diniz, T. N. Ferreira, and L. Lovisolo, "Antenna selection in massive MIMO based on greedy algorithms," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1868–1881, Mar. 2020.

[46] M. Guo and M. C. Gursoy, "Statistical learning based joint antenna selection and user scheduling for single-cell massive MIMO systems," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 1, pp. 471–483, Mar. 2021.

[47] Y. Xin, R. Zhang, D. Wang, J. Li, L. Yang, and X. You, "Antenna clustering for bidirectional dynamic network with large-scale distributed antenna systems," *IEEE Access*, vol. 5, pp. 4037–4047, 2017.

[48] Y. Al-Eryani, M. Akrout, and E. Hossain, "Multiple access in cell-free networks: Outage performance, dynamic clustering, and deep reinforcement learning-based design," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1028–1042, Apr. 2021.

[49] F. Fredj, Y. Al-Eryani, S. Maghsudi, M. Akrout, and E. Hossain, "Distributed beamforming techniques for cell-free wireless networks using deep reinforcement learning," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 2, pp. 1186–1201, Jun. 2022.

[50] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. 31st Conf. Neural Inform. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 1–16.

[51] L. Canese et al., "Multi-agent reinforcement learning: A review of challenges and applications," *Appl. Sci.*, vol. 11, no. 11, pp. 1–25, 2021.

[52] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1342–1397, 2nd Quart., 2021.

[53] S. Hu, X. Chen, W. Ni, E. Hossain, and X. Wang, "Distributed machine learning for wireless communication networks: Techniques, architectures, and applications," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1458–1493, 3rd Quart., 2021.

[54] 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.

[55] R. Ali, Y. B. Zikria, S. Garg, A. K. Bashir, M. S. Obaidat, and H. S. Kim, "A federated reinforcement learning framework for incumbent technologies in beyond 5G networks," *IEEE Netw.*, vol. 35, no. 4, pp. 152–159, Jul./Aug. 2021.

[56] Y. Mu, N. Garg, and T. Ratnarajah, "Federated learning in massive MIMO 6G networks: Convergence analysis and communication-efficient design," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 6, pp. 4220–4234, Nov./Dec. 2022.

[57] 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.

[58] Y.-S. Jeon, M. M. Amiri, J. Li, and H. V. Poor, "A compressive sensing approach for federated learning over massive MIMO communication systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1990–2004, Mar. 2021.

[59] A. M. Elbir and S. Coleri, "Federated learning for channel estimation in conventional and RIS-assisted massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4255–4268, Jun. 2022.

[60] Y. Guo, Z. Qin, and O. A. Dobre, "Federated generative adversarial networks based channel estimation," in *Proc. IEEE Int. Conf. Commun. (ICC) Workshops*, May 2022, pp. 61–66.

[61] K.-K. Wong, G. Liu, W. Cun, W. Zhang, M. Zhao, and Z. Zheng, "Truly distributed multicell multi-band multiuser MIMO by synergizing game theory and deep learning," *IEEE Access*, vol. 9, pp. 30347–30358, 2021.

[62] D. Silver et al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.

[63] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[64] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," 2016, *arXiv:1610.03295*.

[65] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. 11th Int. Conf. Mach. Learn.*, W. W. Cohen and H. Hirsh, Eds. San Mateo, CA, USA: Morgan Kaufmann, 1994, pp. 157–163.

[66] J. G. Kuba et al., "Settling the variance of multi-agent policy gradients," 2021, *arXiv:2108.08612*.
[67] A. Grosnit, D. Cai, and L. Wynter, "Decentralized deterministic multi-agent reinforcement learning," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Dec. 2021, pp. 1548–1553.
[68] W. Zhang, X. Wang, J. Shen, and M. Zhou, "Model-based multi-agent policy optimization with adaptive opponent-wise rollouts," 2021, *arXiv:2105.03363*.
[69] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, "Bridging the gap between value and policy based reinforcement learning," 2017, *arXiv:1702.08892*.
[70] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, Jan. 1996.
[71] H. Jin, Y. Peng, W. Yang, S. Wang, and Z. Zhang, "Federated reinforcement learning with environment heterogeneity," in *Proc. 25th Int. Conf. Artif. Intell. Statist.*, vol. 151, G. Camps-Valls, F. J. R. Ruiz, and I. Valera, Eds., Mar. 2022, pp. 18–37. [Online]. Available: https://proceedings.mlr.press/v151/jin22a.html
[72] B. Banerjee, R. C. Elliott, W. A. Krzymień, and M. Medra, "Machine learning assisted DL CSI estimation for high-mobility multi-antenna users with partial UL CSI availability in TDD massive MIMO systems," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2022, pp. 1579–1585.
[73] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[74] S. Hanna, "Random walks in urban graphs: A minimal model of movement," *Environ. Planning B, Urban Analytics City Sci.*, vol. 48, no. 6, pp. 1697–1711, Jul. 2021.
[75] R. Korbmacher and A. Tordeux, "Review of pedestrian trajectory prediction methods: Comparing deep learning and knowledge-based approaches," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24126–24144, Dec. 2022.
[76] R. Knopp and P. A. Humblet, "Information capacity and power control in single-cell multiuser communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 1, Jun. 1995, pp. 331–335.
[77] B. Woodworth et al., "Is local SGD better than minibatch SGD?" in *Proc. 37th Int. Conf. Mach. Learn.*, Jul. 2020, pp. 10334–10343.

BITAN BANERJEE (Graduate Student Member, IEEE) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, India, in 2015, and the M.Sc. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2017, where he is currently pursuing the Ph.D. degree in communications with the Department of Electrical and Computer Engineering.

He has published more than 20 papers in several prestigious conferences and journals, including IEEE TRANSACTIONS ON COMPUTERS, IEEE COMMUNICATIONS LETTERS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON MACHINE LEARNING IN COMMUNICATIONS AND NETWORKING, IEEE WCNC, IEEE GLOBECOM, IEEE PIMRC, and *Computer Networks* (Elsevier). His research interests include heterogeneous cellular networks, massive MIMO systems, machine learning, reinforcement learning, information-centric networking, and radio resource management. He won the Best Student Paper Award from IEEE PIMRC 2021.

ROBERT C. ELLIOTT (Senior Member, IEEE) received the B.Sc. (cooperative) degree in electrical engineering and the M.Sc. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2000 and 2003, respectively, and the Ph.D. degree in communications from the Department of Electrical and Computer Engineering, University of Alberta, in 2011.

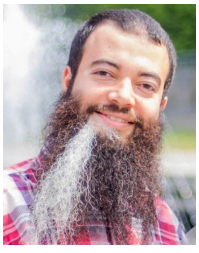
During his B.Sc. studies, he held several cooperative work experience positions. In 1998, he was with Computing Devices Canada (now General Dynamics Mission Systems–Canada), Calgary, AB, Canada, and in 1999, he was with Nortel Networks, Ottawa, ON, Canada. From 2001 to 2016, he was also affiliated with Telecommunications Research Laboratories (TRTech), Edmonton. In 2005, he was a Visiting Researcher with the German Aerospace Center (DLR), Oberpfaffenhofen, Germany. He has also done collaborative research with Huawei Technologies and TELUS Communications, in part as a Post-Doctoral Fellow. He is currently a Research Associate with the Department of Electrical and Computer Engineering, University of Alberta. His research interests include heterogeneous cellular networks, coordinated transmission techniques in broadband multiuser multiple-input multiple-output wireless systems, massive MIMO systems, machine learning, and radio resource management.

Dr. Elliott received the Governor General's Silver Academic Medal and the APEGGA Medal in Electrical Engineering in 2000 for having the highest overall undergraduate academic standing at the University of Alberta. He has also held numerous scholarships and fellowships during his academic studies.

WITOLD A. KRZYMIEŃ (Fellow, IEEE) received the M.Sc. (Eng.) and Ph.D. degrees in electrical engineering from the Poznań University of Technology, Poznań, Poland, in 1970 and 1978, respectively. He received a Polish national award of excellence for his Ph.D. thesis.

Since April 1986, he has been with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada, where he is currently the endowed Rohit Sharma Professor in communications and signal processing. In 1986, he was one of the key research program architects of the newly launched TRLabs, which for a long time was Canada's largest industry-university-government pre-competitive research consortium in the area of information and communication technology. Over the years, he has also done collaborative research work with TELUS Communications, Huawei Technologies, Nortel Networks, Ericsson, German Aerospace Centre (DLR—Oberpfaffenhofen), and the University of Padova, Italy. His current research interests include radio resource management and transceiver signal processing for broadband heterogeneous cellular networks employing machine learning and massive MIMO antenna techniques.

Dr. Krzymień is a fellow of the Engineering Institute of Canada and a licensed Professional Engineer in the Province of Alberta, Canada. Since 2007, he has been an Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. From 1999 to 2005, he was the Chairman of Commission C (Radio Communication Systems and Signal Processing) of the Canadian National Committee of Union Radio Scientifique Internationale (URSI). He has chaired or co-chaired technical program committees for numerous IEEE conferences in wireless communication systems and communication theory areas.



MOSTAFA MEDRA (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Alexandria University, Alexandria, Egypt, in 2009 and 2013, respectively, and the Ph.D. degree in electrical engineering from McMaster University, Hamilton, ON, Canada, in 2017.

From 2017 to 2019, he was a Post-Doctoral Researcher with The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. He is currently with Huawei Technologies Canada Company Ltd., Ottawa, ON, Canada, working on 6G radio access network technologies. His current research interests include MIMO communications, optimization, wireless communications, and signal processing.