Multi-Agent Reinforcement Learning-Based Pilot Assignment for Cell-Free Massive MIMO Systems

MOSTAFA RAHMANI1, MOHAMMAD JAVAD DEHGHANI1, (Member, IEEE), PEI XIAO2, (Senior Member, IEEE), MANIJEH BASHAR3, (Member, IEEE), AND MÉROUANE DEBBAH4, (Fellow, IEEE)

1Department of Electrical and Electronic Engineering, Shiraz University of Technology, Shiraz 71557-13876, Iran
2GIC & GIC, Institute for Communication Systems (ICS), University of Surrey, GU2 7XH Guildford, U.K.
3British Telecommunications (BT) Company, IP5 3RE Ipswich, U.K.
4Technology Innovation Institute, Abu Dhabi, United Arab Emirates

Corresponding author: Mostafa Rahmani (m.rahmani@sutech.ac.ir)

This work was supported in part by the U.K. Engineering and Physical Sciences Research Council under Grant EP/P03456X/1.

ABSTRACT

Cell-free massive multiple-input multiple-output (CF-mMIMO) has been considered as one of the potential technologies for beyond-5G and 6G to meet the demand for higher data capacity and uniform service rate for user equipment. However, reusing the same pilot signals by several users, owing to limited pilot resources, can result in the so-called pilot contamination problem, which can prevent CF-mMIMO from unlocking its full performance potential. It is challenging to employ classical pilot assignment (PA) methods to serve many users simultaneously with low complexity; therefore, a scalable and distributed PA scheme is required. In this paper, we utilize a learning-based approach to handle the pilot contamination problem by formulating PA as a multi-agent static game, developing a two-level hierarchical learning algorithm to mitigate the effects of pilot contamination, and presenting an efficient yet scalable PA strategy. We first model a PA problem as a static multi-agent game with P teams (agents), in which each team is represented by a specific pilot. We then define a multi-agent structure that can automatically determine the most appropriate PA policy in a distributed manner. The numerical results demonstrate that the proposed PA algorithm outperforms previous suboptimal algorithms in terms of the per-user spectral efficiency (SE). In particular, the proposed approach can increase the average SE and 95%-likely SE by approximately 2.2% and 3.3%, respectively, compared to the best state-of-the-art solution.

INDEX TERMS

Cell-free massive MIMO, deep reinforcement learning, pilot assignment, pilot contamination, spectral efficiency.

I. INTRODUCTION

As service demands in wireless communications increase, each new generation of cellular networks must determine new ways to multiplex more devices and enhance the spectral efficiency (SE) per device. Network densification is a common method for increasing the SE in wireless networks [1]. This can be achieved either by adding more base stations (BSs) or increasing the number of antennas on each BS to serve numerous user equipments (UEs) within the same time and frequency resource block, which is known as massive multiple-input multiple-output (mMIMO) [2]. Each of these techniques has certain drawbacks: placing a large number of BSs increases inter-cell interference and consequently decreases the service quality of the UEs, whereas with mMIMO, UEs positioned at the edge of the cell experience substantial propagation loss because of their long distance from the BS. Cell-free mMIMO (CF-mMIMO) was recently suggested as a solution to address the deficiencies of the aforementioned technologies by combining macro-diversity and multi-user interference reduction to provide a uniform user experience [3]. In CF-mMIMO systems, a massive number of distributed access points (APs) simultaneously service a relatively small number of UEs without imposing cell borders, which improves both SE and energy efficiency [4]. The implementation of the MIMO approach requires precise channel state information (CSI) at the transmitter (or receiver). Owing to the scalability of uplink pilot
transmissions, CSI acquisition in CF-mMIMO systems can be performed via uplink pilot transmissions. However, because of the limited coherence time, we are only allowed to have a limited number of orthogonal pilots, which is often less than the total number of UEs. Consequently, we are forced to reuse the same pilots for multiple UEs, introducing undesirable effects known as pilot contamination. As such, the fading channel cannot be accurately estimated because of the co-pilot interference that occurs between UEs. It is worth mentioning that under the assumptions of suboptimal linear precoders, pilot contamination becomes the only capacity-limiting factor for both cellular and CF-mMIMO networks [3], [5]. Consequently, a proper pilot assignment (PA) is vital for minimizing the effects of pilot contamination, which is the primary focus of this research.

A. RELATED WORKS
The optimal PA problem for CF-mMIMO system is the inherently nondeterministic polynomial time (NP). Consequently, the computational complexity required to obtain an optimal solution increases exponentially with the number of UEs. Therefore, the vast majority of published studies have concentrated on heuristic-based solutions. These algorithms can be classified into centralized and distributed schemes. Random PA (RPA) is a technique presented in [3] that enables UE to randomly choose a pilot sequence. The complexity of this strategy is minimal and it can be implemented in a distributed manner. However, the RPA has the worst performance because it is possible for two close UEs to select the same pilot signal, which might result in significant pilot contamination. Additionally, a greedy PA (GPA) approach was suggested in [3] to iteratively update the minimum rate of all UEs. Unfortunately, this approach can only enhance the performance of the weakest UE and not that of the entire system. Using an iterative application of the K-means clustering technique, the authors of [6] proposed a structured PA that seeks to maximize the minimum distance between co-pilot UEs. In [7], the MIMO network was modeled by artificially imposing topological structures on the UE-AP connectivity and considering a partially connected interference network. Thus, the PA problem can be considered a topological interference management problem with multiple groupcast messages. Subsequently, depending on whether the channel connectivity pattern is known a priori, the topological PA problem is formulated in two ways. Another approach that can be employed to address the PA problem is using a graph-theoretic framework. In [8], a conflict graph was developed by forming an edge connecting UEs that are dominant interferers with one another. Subsequently, a greedy approach was employed to solve the graph coloring problem. The authors of [9] mapped the PA problem to the max K-cut problem and solved it using a heuristic algorithm. In [10], the PA problem was formulated as a graph matching problem and a Hungarian algorithm was applied to alleviate it. A novel approach based on tabu search was proposed in [11] to alleviate the PA problem and maximize the sum-user SE. The majority of these studies are based on the similar premise that geographically far enough apart UEs can use the same pilot. This principle also serves as the impetus behind the primary PA scheme that was proposed in this work, along with the additional objective that it should be distributed and scalable while providing competitive performance in terms of the SE for the average UE. The following section will provide a summary of our contributions.

B. MOTIVATION AND CONTRIBUTIONS
Recently, learning-based approaches have demonstrated considerable promise in various applications [12], specifically for addressing a range of resource-allocation problems in wireless communication networks [13], [14], [15], [16]. The authors in [13] proposed a deep supervised learning approach to reduce pilot contamination for a multi-user mMIMO system. In [14] and [15], deep reinforcement learning (DRL) was employed for pilot design to alleviate the pilot contamination problem in cellular mMIMO systems. The authors of [16] developed a deep supervised learning-based PA algorithm for CF-mMIMO systems with massive access. In our previous work [17], we used a single-agent DRL to solve the power allocation problem in CF-mMIMO systems. To the best of our knowledge, this study is the first attempt to use a multi-agent DRL approach to solve a PA problem in CF-mMIMO. The main contributions of this study are summarized as follows:

1. We develop an optimization problem to maximize the uplink sum-user SE by considering the per-user power constraints, channel estimation error, multi-antenna APs, AP selection, and pilot contamination effects.
2. An appropriately designed PA algorithm improves system performance by alleviating the pilot contamination effect. We address the PA problem in an uplink CF-mMIMO system by modeling the PA problem as a diverse clustering problem.
3. We formulate the PA problem as a multi-agent static game. We then employ hierarchical multi-agent DRL to propose an efficient yet distributed PA scheme to mitigate the effects of pilot contamination on CF-mMIMO.
4. The computational complexity and convergence of the proposed approach are analyzed.

The rest of the paper is organized as follows: In Section II, the system model for CF-mMIMO is introduced. In this section, we formulate the PA problem as an optimization problem. Preliminaries on reinforcement learning are studied in Section III. The proposed DRL-based PA approach and its experimental results are presented and discussed in Sections IV and V. Finally, conclusions are drawn, and future work is discussed in Section VI.
is given by $M$, where $M = LN$. It is also assumed that all APs are connected to the central unit via error-free fronthaul links. We employ a UE-centric technique to handle the scalability problem in which each UE is served by a subset of APs. Throughout this paper, $M_k$ denotes the subset of APs serving UE $k$ and $D_l$ denotes the subset of UEs serviced by AP $l$ [18]. We also consider a conventional time-division duplex (TDD) protocol for downlink and uplink data transmissions. Let $\tau_c$ be the length of the coherence block in samples, which is divided into three parts: $\tau_c = \tau_p + \tau_T + \tau_d$; where $\tau_p$, $\tau_T$, and $\tau_d$ samples are used for uplink data transmissions, uplink pilot training, and downlink data transmissions, respectively [2]. In this paper, we focus on uplink training and uplink data transmissions and assume that $\tau_d = 0$. The channel coefficient between UE $k$ and AP $l$ is given by $g_{kl} = \sqrt{p_{kl}} h_{kl} \in \mathbb{C}^N$, where $p_{kl}$ indicates the large-scale fading coefficient (both path loss and shadowing), and $h_{kl} \in \mathbb{C}^N$ are independent and identically distributed random variables that represent the small-scale fading vector. In our model, it is presumed that the system has access to all deterministic information, including the large-scale fading coefficient and the geographic locations of the APs.

**A. UPLINK PILOT TRAINING AND CHANNEL ESTIMATION**

During the channel estimation phase, each UE uses a pilot from a set $P$ orthogonal pilot signal $P = \{\phi_1, \phi_2, \ldots, \phi_P\}$, where $\phi_i$ denotes the $i$-th pilot signal with the length of the $\tau_p$ samples. We assume that $P$ pilot signals are mutually orthogonal, $P \leq \tau_p$. Since the pilot resources are limited, the pilot set must be reused throughout the network. As a result, many UEs share the same pilot; we refer to these UEs as co-pilot UEs (pilot-sharing UEs). We indicate the index of the pilot assigned to UE $k$ by $p_k = \{1, 2, \ldots, P\}$ and $C_k$ as the set of co-pilot UEs, including UE $k$ itself. After correlating the received signal at AP $l$ with the pilot $p_k$, the estimate of channel $y_{pkl}$ is given by:

$$y_{pkl}^p = \sum_{k \in C_k} \sqrt{p_{pkl}} g_{kl} + n_l,$$

where $p_{pkl}^p$ denotes the normalized transmit signal-to-noise ratio of the pilot symbol for UE $k$, and $n_l$ is an additive Gaussian noise at the $l$-th AP whose elements are i.i.d. $\mathcal{CN}(0, 1)$. The minimum mean-squared-error estimate of the channel coefficient between the $k$-th UE and the $l$-th AP is given by

$$\hat{g}_{kl} = \alpha_{kl} \left( \frac{1}{\tau_p} \sum_{k \in C_k} g_{kl} + W_{p,l} \phi_k \right),$$

where $W_{p,l} \in \mathbb{C}^{N \times \tau_p}$ denotes the noise at the $l$-th AP, whose elements are i.i.d. $\mathcal{CN}(0, 1)$, and $\alpha_{kl}$ is given by

$$\alpha_{kl} = \frac{\sqrt{p_{kl}} \beta_{kl}}{\tau_p \sum_{k \in C_k} p_{kl} + 1}.$$  

It is worth mentioning that $g_{kl}$ is uncorrelated with the $\hat{g}_{kl}$ and distributed as follows:

$$g_{kl} \sim \mathcal{CN}(0, \gamma_{kl}); \quad \hat{g}_{kl} \sim \mathcal{CN}(0, \beta_{kl} - \gamma_{kl}),$$

where $\gamma_{mk} = \sqrt{p_{mk} p_{mk} \alpha_{mk}}$.

**B. UPLINK DATA TRANSMISSION**

AP $l$ physically receives the signal from all UEs during uplink data transmission, this is,

$$y_l = \sum_{k=1}^{K} g_{kl} s_k + n_l, \quad (5)$$

where $s_k (\mathbb{E}|s_k|^2 = p_k)$ is the symbol transmitted from UE $k$ with power $\sqrt{p_k}$, and $n_l \in \mathbb{C}^N$ denotes the noise at AP $l$, and its elements are i.i.d. $\mathcal{CN}(0, 1)$. However, to estimate $\hat{s}_k$, only the APs in $M_k$ can be used, and $\hat{s}_k$ is given as:

$$\hat{s}_k = \sum_{l=1}^K a_{hl}^H y_l,$$

where $a_{hl} \in \mathbb{C}^N$ is the combining vector selected by AP $l$ for UE $k$. Here, we consider the maximum ratio combining (MRC) method with $a_{hl} = \hat{g}_{kl}$. By substituting (5) and $g_{kl} = (g_{kl} - \hat{g}_{kl})$ into (6), we obtain

$$\hat{s}_k = \sum_{l=1}^K a_{hl}^H g_{kl} s_k + \sum_{l=1}^K \sum_{k' \neq k} a_{hl}^H g_{kl' \hat{s}_{k'}} + \sum_{l=1}^K a_{hl}^H n_l + \sum_{k'=1}^K \sum_{k \neq k'} a_{hl}^H g_{kl' \hat{s}_{k'}}.$$  

where $\text{DS}_k$, $\text{IUI}_{kk'}$, $\text{TEE}_{kk'}$, and $\text{TN}_k$ are the desired signal, interuser interference, total estimation error, and total noise, respectively. The achievable SE for UE $k$ is obtained by utilizing the so-called use-and-then-forget bound because the CPU is unaware of the channel estimations [19]:

$$\text{SE}_k = \frac{\tau_d}{\tau_c} \mathbb{E}_{\text{SSF}} \left\{ \log_2 (1 + \text{SINR}_k) \right\},$$

where $\text{SSF}$ denotes the sum of squared functions.
where SINR is given by
\[
\text{SINR}_k = \frac{Np_k (\sum_{M_k} \gamma_{kl})^2}{D_k},
\]
where \( D_k \) is obtained as follow:
\[
N \sum_{k'=1}^{K} p_{k'} \left( \sum_{M_k} \gamma_{kl} \beta_{k'l} \right)^2 + \sum_{k'=1}^{K} \sum_{k'=1}^{K} p_{k'} \beta_{k'l} + \sum_{M_k} \gamma_{kl} = (10)
\]
In the above equation, the first term denotes the interference caused by co-pilot UEs, the second term corresponds to multi-user interference, and the last term contains the noise power.

C. ACCESS POINT SELECTION

In the proposed approach, unlike [18], which first selects the appropriate APs for each UE through a competitive mechanism and then implements a PA approach, we first apply the PA algorithm and then perform AP clustering. Our main idea is based on the fact that all APs physically receive the signal of all UEs, and therefore, proper PA schemes must be performed before AP clustering. The proposed PA scheme is described in detail in Sections IV. Here, we cover the AP selection algorithm in which the pilot is already assigned to each UE. The following assumption was considered when developing an algorithm for AP selection:

Assumption: Each AP serves most of the \( \tau_p \) UEs (one UE per pilot) to prevent severe pilot contamination.

Our proposed AP selection method consists of the following steps: Step 1) UE \( k \) finds the AP \( l \) with the strongest large-scale fading (LSF) coefficient as its master AP, and two co-pilot UEs are not able to choose one specific AP for their Master AP; in this case, the latter UEs select the second-strongest LSF channel coefficient as its Master AP. In this step, we ensure that each UE is served by at least one AP. Step 2) Each AP decides to serve \( P \) UEs. Each AP then selects one UE per pilot with the strongest LSF coefficient.

D. UPLINK DATA POWER CONTROL

Before delving into the proposed PA scheme, it is pertinent to explore a scalable power control with low complexity. Here, we utilize the fractional power (FrP) algorithm to determine the appropriate power control for UE \( k \), which is given by
\[
p_k = \min_{1 \leq i \leq K} \left( \frac{\sum_{m=1}^{M} \beta_{mi}}{\sum_{m=1}^{M} \beta_{mk}} \right)^{\theta} \tilde{p}, \quad (11)
\]
where \( \theta \in [0, 1] \) controls how the range of power coefficients is compressed and larger values of \( \theta \) indicate better fairness.

E. PROBLEM FORMULATION

The objective of our PA algorithm is to maximize the sum SE by properly selecting the set of co-pilot UEs subject to certain constraints owing to the limited availability of pilots. In the literature on linear programming, this type of problem is often written up using the column generation method, in which each possible group of co-pilot UEs is thought of as a column. Let \( A \) represent all possible co-pilot UE sets that do not include null and singleton sets. As a result, \( A \) has the cardinality of \( 2^K - 1 \), which is equal to the number of columns. The matrix \( A \) is defined, in which each column represents a set in \( A \), and the corresponding column for the co-pilot UEs is indicated by \( x_j \). We define the cost of \( x_j \) as
\[
C(x_j) = \begin{cases} 
- \sum_{k \in x_j} \log_2 (1 + \text{SINR}_k), & \text{SINR}_k \geq \Gamma_{min} \\
\infty, & \text{SINR}_k < \Gamma_{min}
\end{cases} \quad (12)
\]

According to the above definition, the set of co-pilot UEs that do not fulfill the minimum SINR threshold, even for a single UE, is never chosen, and \( \Gamma_{min} \) denotes the minimum SINR threshold. The optimization problem is formulated as follows
\[
P_1 : \max_{\Lambda} \sum_{s=1}^{\vert A \vert} C(x_s) \lambda_s \quad (13a)
\]
\[\text{s.t.} \quad \Lambda \lambda = 1 \quad (13b)
\]
\[\|\Lambda\|_1 = P \quad (13c)
\]
where \( \Lambda = [\lambda_1, \ldots, \lambda_{\vert A \vert}] \). Equation (13a) guarantees that each UE is assigned one pilot, and (13b) guarantees that exactly \( P \) pilots are used in the PA scheme. Problem P1 is NP-hard in nature, whereas the feasible solution space is denoted by \( \binom{\vert A \vert}{P} \). For example, the feasible solution space for a moderately small system comprising 20 UEs and 10 pilots is approximately equal to \( 4.4 \times 10^{33} \). Owing to the complexity of the problem, it is reasonable to explore suboptimal solutions that can be effectively implemented in a network. In Sections IV, we describe a multi-agent-based PA scheme that considers only UE’s location to cluster the co-pilot UEs in a way that mitigates pilot contamination-induced interference and indirectly enhances the sum SE. This approach is scalable and can be implemented in a distributed manner as network size increases.

III. PRELIMINARIES ON DEEP REINFORCEMENT LEARNING MODEL

Reinforcement learning is a type of sequential decision making in which the goal is to learn a policy in a given environment whose dynamics are unknown. Reinforcement learning requires an interactive environment in which an agent can choose from a variety of activities that affect the environment. DRL, which effectively solves complex and high-dimensional environments, was created by fusing a reinforcement learning structure with a deep neural network (DNN) as a function approximation [20]. In this section, we review the mathematical background and preliminaries of both single-agent and multi-agent DRL.
The Markov decision process (MDP) formulation develops a mathematical foundation for modeling a single-agent DRL environment that includes tuple \((S, U, r, \eta)\). In the MDP model, \(S\) is a set of possible states \((s_t)\), \(U\) is a set of possible actions \((u)\), and \(r\) is the decision time point. The transition probability of performing action \(u\) in state \(s_t\), resulting in state \(s_{t+1}\), is denoted as \(P_a : S \times A \times S \rightarrow [0, 1]\). The expected immediate reward for taking action \(u\) and the transition from state \(s_t\) to state \(s_{t+1}\) are defined by \(r_u\), and \(\eta\) determines the discount factor [21]. Different DRL systems have varying specifications regarding how data are gathered and how their performance is measured. The dynamic relationship between the agent and environment is depicted in Fig. 2. After carrying out the action requested by the agent, the environment always delivers the corresponding state and reward to the agent at the end of each iteration. Finding an optimal policy, \((\pi^*(s))\), which takes action depending on the present state in order to maximize the discounted reward over time, is the fundamental objective of the RL agent. This can be obtained by optimally solving the Bellman equation:

\[
Q^\pi_n(s, u) = E \left[ r(s, u) + \eta \sum_{u'} \pi(u|s) Q_n(s', u') \right],
\]

(14)

Q-learning is one of the most influential developments in reinforcement learning, and is a specific implementation of the Bellman equation based on the Q-table as follows:

\[
Q(s_t, u_t) \leftarrow Q(s_t, u_t) + \alpha [r_t + \eta \max_a Q(s_{t+1}, u) - Q(s_t, u_t)]
\]

(15)

Q-table computes the estimated maximum future reward for each action in each state. Subsequently, the proper action is selected to yield the greatest expected reward. The optimal state-action value function can be obtained by updating (15). However, obtaining a Q-table is tedious task in many cases. A Deep Q-network (DQN) expands the Q-learning concept to alleviate this problem. The DQN utilizes a DNN instead of a Q-table to estimate the nonlinear Q-values for each state-action [22]. The optimal action policy is obtained based on every action that the agent can perform in the environment, as follows:

\[
\pi^*(s) = \arg \max_{u \in A} \{Q(s, u; \theta^*)\}.
\]

(16)

The DQN agent aims to maximize the return function by determining the optimal weight vector, that is, \(\theta\). Owing to the ambiguous transition probability, the DQN agent uses an epsilon-greedy algorithm to balance the exploitation and exploration. Each experience is stored in a first-in-first-out replay buffer, which is always accessible. Finally, the DQN agent updates \(\theta\) to minimize the loss function by choosing a mini-batch of events from the experience replay buffer and applying a proper optimizer (such as Adam or SGD). For further information, refer to [23] which provides a detailed examination of DRL as well as various types of neural networks, DRL architectures, and the applications of these technologies in the real world.

B. MULTI-AGENT DRL (MADRL)

In practical situations, multi-agent systems have inspired the development of distributed solutions that are likely to be less expensive and more effective than centralized single-agent alternatives. Sequential decision-making problems involving a number of agents are addressed by MADRL, in which all agents work together to influence the dynamics of the system. In particular, the reward that an agent receives now depends on the actions of all other agents rather than just its own. Consequently, a specific agent should consider the policies of other agents to maximize the long-term reward. The stochastic game (SG) is a generalization of the MDP to the multi-agent scenario that includes a tuple \((S, U^1, \ldots, U^n, r^1, \ldots, r^n, P)\) [24], where \(n\) is the number of agents, \(U^i, i = 1, 2, \ldots, n\) denotes the finite sets of actions available to the agents, resulting in the joint action set \(U = U^1 \times U^2 \times U^n\). In the state transition probability function. The Q-function of each agent depends on the joint action and policy. Furthermore, in fully cooperative SG, all the agents have the same reward function. There is a specific SG with no state signal known as a static (stateless) game. A static game is defined by a tuple \((U^1, \ldots, U^n, r^1, \ldots, r^n, P)\) in which all agents make decisions simultaneously, without knowledge of the policies that are being chosen by other agents.

1) MULTI-AGENT DQN (MADQN)

In an SG, the joint optimal policy is known as the Nash equilibrium (NE) and can be defined as \(\pi^* = (\pi^*_1, \pi^*_2, \ldots, \pi^*_n)\). The NE is an integration of all agents’ policies, with each policy being the optimal reaction to the policies of the other agents. In NE, each agent’s action is an optimal reaction to the choice of another agent’s action. Specifically, an agent’s policy must be compatible with that of other agents; otherwise, no actor may gain an advantage by modifying its policy. Consequently, each duty of learning agent is to investigate an NE for any given condition in the environment. Each agent in
MADQN comprises a primary (online) network, target network, and replay memory. In the training phase, the learnable parameters of the DNN are updated to boost the accuracy of the Q-function approximation in accordance with the system transition history. In the learning step, \( t \), each agent inputs the current state into the DQN and outputs Q-values for each action. Agents in DQN employ the same Q-values for action selection and assessment, leading to a Q-value overestimation problem. To address this problem and enhance the learning efficiency of the agents, the following two enhanced versions of DQN are presented.

2) MULTI-AGENT DOUBLE DQN (MADDQN)

The DDQN overcomes the overestimation problem by separating the maximum procedure in the target network’s action selection from the action assessment of the target network. In particular, two networks, DQN1 and DQN2, have been utilized, where DQN1 is employed to select actions and DQN2 is employed to assess the Q-value of the corresponding action.

3) MULTI-AGENT DUELING DOUBLE DQN (MAD3QN)

The standard DQN method computes the value of each action in a given state. However, in certain states, different policies can lead to the same value function. This behavior can hinder the ability to learn an optimal response to a specific condition. To alleviate this problem, dueling of a DDQN has been proposed. The dueling DDQN is an improved variant of the DQN in which the Q-network includes two streams of Q-functions, namely, the state value function \( V(s) \) and the advantage function \( A(s, a) \), where the advantage function is used to assess the relative significance of one action in relation to other actions in a given situation. The output of the dueling network is derived by merging these two streams into a single-output Q-function and an aggregate module to accelerate convergence and enhance the efficiency.

IV. PROPOSED MADRL-BASED PILOT ASSIGNMENT SCHEME

A. MODELING THE PROBLEM AS A STOCHASTIC GAME

The interactions of the agents are modeled by SG, where the environment is altered in response to player actions. Here, we show that the PA in CF-mMIMO systems can be modeled as a SG. The detailed definitions are provided below.

1. Set of agents: In a proper PA, each UE attempts to choose an appropriate pilot sequence from \( P \) orthogonal pilots. We define a new strategy in which each pilot, as an agent, interacts with the wireless communication environment and selects the UEs that can be assigned to it.

2. State space: All agents collectively probe the environment by inspecting various states. Specifically, the location and SINR of the co-pilot UEs are considered to be the current state.

3. Action space: We define the action as reassigning a new pilot to one of the UEs in each agent. A set of discrete pilot indices is considered as the action space. Each agent is allowed to select only one UE in each time slot and to reassign the new pilot index to the selected UE. All agents have the same action space.

4. Reward function: It evaluates an agent’s actions as either favorable or unfavorable. Consequently, a reward function must be constructed that corresponds to the objective function described in (12). The reward function of each agent at learning step \( t \) is defined as

\[
r(t) = C(x_j, t) - C(x_j, t + 1)
\]

B. PROPOSED PA APPROACH

In the proposed method, we provide a new perspective on the PA problem. An effective PA policy requires only location of the UE and does not require any additional signaling overhead to mitigate the effects of pilot contamination. This is because the distance between the UEs has the most significant impact on co-pilot interference. Here, we model the PA as a diverse clustering problem. To this end, a static multi-agent game is defined with \( P \) teams (agents), in which each team is represented by a specific pilot. Each team is interested in connecting with an \( \lceil \frac{K}{P} \rceil \) UEs, where each team is supposed to select a number of UEs in which the least pilot contamination effect occurs. It is worth mentioning that the main difference between our approach and the previous one is that, instead of assigning the pilot to the UEs, we cluster the UEs into teams specified by the pilot. As shown in Fig. 3, the implementation of the proposed approach is a two-level hierarchical MADRL. The steps of the proposed approach are as follows.

1. Define \( P \) agents (\( P \) teams), each represented by a specific pilot.

2. Randomly set \( \lceil \frac{K}{P} \rceil \) UEs for each pilot as the initialization phase. Each team has at most \( \lceil \frac{K}{P} \rceil \) co-pilot UEs. This is the starting point and the input of the proposed algorithm. The proposed method attempts to maximize...
3. In each agent, the UE that causes the worst pilot contamination effect by the pre-trained DNN is selected and expelled. Simply put, the agents learn to select the best UE candidate at a low level through independent learning. This step is executed in a distributed manner and the agents are executed independently of each other. The output of this step is an exclusion list containing all the UEs that have been expelled from their group. In this step, we design one DNN and train it in an unsupervised manner to learn to select the UE that has the least sum distance to other co-pilot UEs. Here, we do not require any ground-truth outputs for training, which makes this step more flexible for practical implementation. In Fig. 4, the proposed DNN for the worst UE selection is illustrated. The network comprises four layers that are consecutively interconnected to select one UE from the \(P\) UEs. The input of the network is the UEs’ location, the number of neurons in the input layer is equal to \(2P\), and the number of neurons in the output layers is equal to \(\lceil P/\pi \rceil\) to construct a one-hot vector to determine the index of the expelled UE. The number of hidden layers and nodes within the layer is a hyperparameter, and in our proposed structure, we consider two hidden layers with \(\lceil K/4 \rfloor\) and \(\lceil K/2 \rfloor\) nodes in each hidden layer. Finally, the following loss function is considered for unsupervised training to select the UE that causes the most pilot contamination.

\[
\text{Loss} = \sum_{k=1}^{\lceil P/\pi \rceil} \text{dist}(\text{UE}_k, \text{UE}^\star), \quad (18)
\]

where

\[
\text{dist}(\text{UE}_k, \text{UE}^\star) = \sqrt{\sum_{i=1}^{\lceil P/\pi \rceil} (\text{UE}_k[i] - \text{UE}^\star[i])^2}, \quad (19)
\]

computes the distance between the UE\(k\) and UE\(^\star\), where \(F\) denotes the geographical coordinates of the UEs. Because the distance between UEs has the greatest influence on co-pilot interference, we only consider (19) as a loss function. However, for more sophisticated loss functions, we also need to consider shadowing and fast-fading effects, which will be addressed in future work.

4. Implement cooperative agents with centralized training and decentralized execution to learn how to reassign UEs in the exclusion list into different teams so that the final objective function is improved. During the learning phase, the cost function constructed in (12) is made available to each agent as a reward. Each agent then adjusts its actions to remain close to an optimal policy by updating the deep Q-network. In this step, each agent learns to connect its expelled UE to the best possible team (assigning a new pilot for the expelled UE). During the execution phase, each agent receives local observations of the environment (which is the co-pilot UEs’ location and UEs’ SINR) and then chooses an action (reassigning a new pilot to expelled UE) based on its trained DQN. The NN used in this step is similar to that used in the previous step. The only difference is the number of neurons in the output layer, which is \(P\) to construct a one-hot vector to determine a new pilot for the UE.

5. Repeat Steps 3-4 until the stopping condition is met, that is, the difference between two consecutive objective functions is less than the threshold or the number of iterations specified in advance.

C. COMPLEXITY ANALYSIS

This section presents a computational complexity analysis of the proposed PA scheme. The computational complexity of the MADQL-based PA is determined by the number of floating operations per second (FLOPS) in the neural network. The number of FLOPS for each NN used in Step 3 is determined by matrix multiplication in each layer [25] as:

\[
\text{FLPOS}_{\text{DNN}} \approx \left(\frac{10 + P}{8P}\right)K^2 + \left(\frac{3P + 4}{4P}\right)K, \quad (20)
\]

and the number of FLOPS for each NN used in Step 4 is given by

\[
\text{FLPOS}_{\text{DQN}} \approx \left(\frac{P + 1}{8P}\right)K^2 + \left(\frac{3}{4}\right)K + 1 \quad (21)
\]

During the inference, \(P\) DNN in Step 3 and \(P\) DQN in Step 4 should be performed to ensure proper action. Therefore, the total number of FLOPS for our approach is computed as follows:

\[
\text{FLPOS} = P \times \text{FLPOS}_{\text{DNN}} + P \times \text{FLPOS}_{\text{DQN}} \\
\approx \left(\frac{1}{4}\right)PK^2 + \left(\frac{11}{8}\right)K^2 + \left(\frac{3}{2}\right)KP + K + P. \quad (22)
\]

As shown in (22), the complexity of the proposed approach is of the polynomial order \((PK^2)\).
TABLE 1. Simulation parameters.

| Parameter          | Value   |
|--------------------|---------|
| Carrier frequency  | 1.9 GHz |
| Bandwidth          | 20 MHz  |
| Noise figure       | 9 dB    |
| Pilot transmit power ($P_t$) | 100 mW |
| Data transmit Power ($P_d$) | 100 mW |

V. NUMERICAL RESULT AND DISCUSSION

A. SIMULATION SETUP

A typical CF-mMIMO with $L = 100$ APs equipped with $N = 2$ antennas and $K = 50$ UEs is considered, where all APs and UEs are independently and uniformly distributed in a simulation area of size $1km \times 1 km$. We utilize the wrap-around technique to prevent boundary effects at the edge and imitate network behavior over an infinite area. We formulate the large-scale propagation parameters to generate the path loss and the shadow fading coefficients using the 3GPP Urban Microcell model [3]. We also consider that each coherence block has $\tau_c = 200$ samples, where $P$ samples are used for uplink pilots and the remainder is used for uplink data. Other simulation parameters are summarized in Table 1, which are the same parameters as used in [4]. We also employed Frp power control expressed by (11) to further improve the system performance.

B. BENCHMARK ALGORITHMS

In this section, we evaluate the performance of our proposed PA scheme. We compared the performance of the proposed approach with that of the benchmark solutions listed below.

1. RPA: Each UE is randomly assigned one pilot from $P$ orthogonal pilots.
2. GPA: This approach considers a simple greedy algorithm that iteratively refines the PA. GAP is explained in detail in [3].
3. Repulsive clustering based PA (RCBPA) [26]: In this approach, a repulsive clustering-based PA scheme is employed to mitigate the effects of pilot contamination.
4. No-pilot contamination (NPC): In this approach, we consider that there is no limitation on pilot resources, and each UE uses an orthogonal pilot. It should be mentioned that in this approach, the number of pilots is equal to the number of UEs, $\tau_p = K$.

C. SIMULATION SETUP

We implemented our MADRL approach in the TensorFlow 2 framework and ran our simulation on a PC with a Core(TM)i7 CPU @ 4 GHz and 32 GB of installed memory (RAM).

D. RESULT AND DISCUSSION

Fig. 5 illustrates the convergence behavior of the MADRL-based PA schemes in CF-mMIMO system. This figure describes the performance of the proposed PA approach with different agents (MADQN, MADDQN, MAD3QN) in terms of the total reward. It is worth mentioning that we implement the MAD3QN and MADDQN algorithms for PA with the same DNN architecture as the one used in MADQN. The first observation is that all three agents eventually converge to the same amount of reward, which confirms that the final performance of the proposed PA algorithm remains unchanged by utilizing different agents, and the only difference is the speed of convergence. More specifically, MAD3QN and MADDQN converge quickly, and favorable results are obtained after 95 and 145 iterations, respectively. Furthermore, it can be seen that the MADQN approach learns the right policy to gain positive rewards after 174 episodes, and it can achieve complete convergence after 400 episodes. Fig. 6 illustrates the cumulative distribution function of the SE per UE. We compare the proposed schemes with the four above-mentioned benchmarks. The first observation is that the proposed MADRL-based PA performs better than other approaches. As shown in Fig. 6, there are 40%, 29% and 16% improvements in the median of per-user uplink SE with

FIGURE 5. The convergence of the proposed MADRL-based PA algorithm with various agents.

FIGURE 6. Cumulative distribution of the per-user SE of CF-mMIMO with MRC receiver with $L = 100$, $N = 2$, $K = 50$, and $P = \tau_p = 10$ for different PA strategies.
the proposed scheme compared to the RPA, GPA, RCBPA algorithm, respectively. When comparing with the NPC, the proposed method performs better by a small margin. It should be considered that although there is no pilot contamination in NPC approach, assigning $\tau_p = K$ samples to pilots degrades the SE by a factor of $1 - \frac{\tau_p}{P}$. The 95%-likely of the uplink SE by varying the number of pilots for different PA schemes is depicted in Fig. 7, in which we can see the impact of the number of pilots in 95%-likely SE. In the NPC scheme, the number of pilots is constant and equal to $\tau_p = 50$, which we use in the figure for comparison. It is obvious that for $\tau_p = 1$ and $\tau_p = 50$ the 95%-likely of the uplink SE for all PA schemes are the same. For the RPA scheme, although increasing the number of pilots improves the system performance (the 95%-likely of the uplink SE), there is always a possibility that two UEs that are not far from each other will be assigned the same pilot and then cause strong mutual interference. For GPA, the same trend can be observed. It has slightly better performance due to the greedy search. For RCBPA and our scheme, adding more pilots will only enhance the 95%-likely of the uplink SE up to a certain point, and then it will begin to decrease once that threshold has been reached. Specifically, Fig. 7 shows that the maximum 95%-likely SE of CF-mMIMO with the MADRL-PA scheme is about 1.45 bits/s/Hz, and for the RCBPA scheme it is approximately 1.33 bits/s/Hz. It is also obvious that for both algorithms (proposed approach and RCBPA) the maximum 95%-likely SE is achieved at the $\tau_p = 16$. This figure also demonstrates how important it is to determine the optimum number of pilots. This issue will be the subject of our future research.

In this study, we proposed a MADRL-based PA scheme for CF-mMIMO systems with the objective of maximizing the sum SE in the uplink. For this purpose, we considered the PA problem as a multi-agent static game and developed a two-level hierarchical MADRL algorithm to mitigate the effects of pilot contamination by assigning the same pilot to UEs that are geographically far apart. At a low level, agents learned to select the worst UE among the co-pilot UEs in an unsupervised manner, whereas they

| Authors, year | Average SE  | 95%-likely SE |
|---------------|------------|---------------|
| Liu, et al., 2019 [11] | 230.8 | 1.04 |
| Buzzi, et al., 2020 [10] | 233.1 | 0.91 |
| Liu, et al., 2020 [8] | 231.5 | 1.21 |
| Zeng, et al., 2021 [9] | 223.8 | 0.98 |
| Yu, et al., 2022 [7] | 237.8 | 1.12 |
| Mohebi, et al., 2022 [26] | 237.2 | 1.20 |
| Ours | 243.1 | 1.25 |

VI. CONCLUSION

In particular, compared to the existing competing alternative, our PA approach improves the average SE by 2.2% and the second-best performance are highlighted and underlined.
learned to reassign expelled UEs with new pilots at a higher level via centralized training and decentralized execution. In addition, the complexity and convergence of the proposed scheme were demonstrated. Furthermore, the superiority of MADRL-based PA over prior algorithms, such as RPA, GPA, and RCBPA, was validated by simulation results. It has been shown that compared to the existing competing alternatives, the proposed MADRL-based PA approach achieved approximately 2.2% and 3.3% improvements in the average SE and the 95%-likely SE, respectively. In future work, we will expand our MADRL-based approach by jointly optimizing the PA, power control, and subcarrier allocation to enhance CF-mMIMO system performance.

REFERENCES

[1] J. Liu, M. Sheng, L. Liu, and J. Li, “Network densification in 5G: From the short-range communications perspective,” IEEE Commun. Mag., vol. 55, no. 12, pp. 96–102, Dec. 2017.
[2] 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.
[3] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, “Cell-free massive MIMO versus small cell,” IEEE Trans. Wireless Commun., vol. 16, no. 3, pp. 1834–1850, Mar. 2017.
[4] 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.
[5] T. L. Marzetta and H. Q. Ngo, Fundamentals of Massive MIMO, Cambridge, U.K.: Cambridge Univ. Press, 2016.
[6] M. Attarfar, A. Abbasfam, and A. Lozano, “Random vs structured pilot assignment in cell-free massive MIMO wireless networks,” in Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops), May 2018, pp. 1–6.
[7] H. Yu, Y. Yi, and G. Caire, “Topological pilot assignment in large-scale distributed MIMO networks,” IEEE Trans. Wireless Commun., vol. 21, no. 8, pp. 6141–6155, Aug. 2022.
[8] H. Liu, J. Zhang, S. Jin, and B. Ai, “Graph coloring based pilot assignment for cell-free massive MIMO systems,” IEEE Trans. Veh. Technol., vol. 69, no. 6, pp. 9180–9184, Aug. 2020.
[9] W. Zeng, Y. He, B. Li, and S. Wang, “Pilot assignment for cell free massive MIMO systems using a weighted graphic framework,” IEEE Trans. Veh. Technol., vol. 70, no. 6, pp. 6190–6194, Jun. 2021.
[10] S. Buzzi, C. D’Andrea, M. Fresia, Y.-P. Zhang, and S. Feng, “Pilot assignment in cell-free massive MIMO based on the Hungarian algorithm,” IEEE Wireless Commun. Lett., vol. 10, no. 1, pp. 34–37, Jan. 2021.
[11] H. Liu, J. Zhang, X. Zhang, A. Kurniawan, T. Juhana, and B. Ai, “Tabu-search-based pilot assignment for cell-free massive MIMO systems,” IEEE Trans. Veh. Technol., vol. 69, no. 2, pp. 2286–2290, Feb. 2020.
[12] K. Aggarwal, M. M. Mijwal, A. H. Al-Mistarehi, S. Alomari, M. Gök, A. M. Z. Alaabdin, and S. H. Abdulrhman, “Has the future started? The current growth of artificial intelligence, machine learning, and deep learning,” Iraqi J. Comput. Sci. Math., vol. 3, no. 1, pp. 115–123, 2022.
[13] J. Xu, P. Zhu, J. Li, and X. You, “Deep learning-based pilot design for multi-user distributed massive MIMO,” IEEE Trans. Commun. Lett., vol. 8, no. 4, pp. 1016–1019, Aug. 2019.
[14] Y. Omid, S. M. Hosseini, S. M. Shahabi, M. Shikh-Bahaei, and A. Nallanathan, “AoA-based pilot assignment in massive MIMO systems using deep reinforcement learning,” IEEE Commun. Lett., vol. 25, no. 9, pp. 2948–2952, Sep. 2021.
[15] S. Luo, C. Zhang, Y. Duan, and J. Chen, “Pilot allocation game: A Monte Carlo tree based method,” in Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC), Oct. 2019, pp. 277–282.
[16] J. Li, Z. Wu, P. Zhu, D. Wang, and X. You, “Scalable pilot assignment scheme for cell-free large-scale distributed MIMO with massive access,” IEEE Access, vol. 9, pp. 122107–122112, 2021.
[17] M. Rahmani, M. Bashar, M. J. Dehghani, P. Xiao, R. Tafazoli, and M. Debbah, “Deep reinforcement learning-based power allocation in uplink cell-free massive MIMO,” in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Apr. 2022, pp. 459–464.
[18] S. Chen, J. Zhang, E. Björnson, J. Zhang, and B. Ai, “Structured massive access for scalable cell-free massive MIMO systems,” IEEE J. Sel. Areas Commun., vol. 39, no. 4, pp. 1086–1100, Apr. 2021.
[19] T. C. Mai, H. Q. Ngo, M. Egan, and T. Q. Duong, “Pilot power control for cell-free massive MIMO,” IEEE Trans. Veh. Technol., vol. 67, no. 11, pp. 11264–11268, Nov. 2018.
[20] Y. S. Nasir and D. Guo, “Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks,” IEEE J. Sel. Areas Commun., vol. 37, no. 10, pp. 2239–2250, Oct. 2019.
[21] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2018.
[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, 2015.
[23] S. Balharra, N. Gupta, A. Alkhayyat, I. Bharti, R. Q. Malik, S. N. Mahmood, and F. Abedi, “A survey on deep reinforcement learning architectures, applications and emerging trends,” IET Commun., pp. 1–16, Jul. 2022.
[24] L. Buşoniu, R. Babuś, and B. De Schutter, “Multi-agent reinforcement learning: An overview,” in Innovations in Multi-Agent Systems and Applications-I. Berlin, Germany: Springer, 2010, pp. 183–221.
[25] W. Li, W. Ni, H. Tian, and M. Hua, “Deep reinforcement learning for energy-efficient beamforming design in cell-free networks,” in Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW), Mar. 2021, pp. 1–6.
[26] S. Mohebi, A. Zanella, and M. Zorzi, “Reactive clustering based pilot assignment for cell-free massive MIMO systems,” 2022, arXiv:2203.12403.

MOSTAFA RAHMANI was born in Iran, in 1986. He received the B.Sc. degree in electrical engineering from Shiraz University, Iran, in 2009, and the M.Sc. degree in communication systems engineering from Tarbiat Modares University, Tehran, Iran, in 2012. He is currently pursuing the Ph.D. degree in communication systems engineering with the Shiraz University of Technology, Shiraz, Iran. His research interests include massive MIMO systems and machine learning for signal processing in communications.

MOHAMMAD JAVAD DEHGHANI (Member, IEEE) received the B.S. and M.S. degrees from the Isfahan University of Technology, Iran, in 1998 and 1992, respectively, and the Ph.D. degree from IIT Madras, in 2003, all in communication engineering. In 1993, he joined Shiraz University as a Faculty Member. Since 2003, he has been a Faculty Member of the Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz, Iran. His activities included wireless communications, MIMO, resource allocation, underwater acoustic signal processing, and source localization.

PEI XIAO (Senior Member, IEEE) received the Ph.D. degree from the Chalmers University of Technology, Gothenburg, Sweden, in 2004. He is currently a Professor in wireless communications at the Institute for Communication Systems and the Home of 5GIC and 6GIC, the University of Surrey. He is also the Technical Manager of 5GIC/6GIC, leading the research team in the new physical layer work area and coordinating/supervising research activities across all the work areas (https://www.surrey.ac.uk/institute-communication-systems/5g-6g-innovation-centre). Prior to this, he worked at Newcastle University and Queen’s University Belfast. He also held positions at Nokia Networks, Finland. He has published extensively in the fields of communication theory, RF and antenna design, and signal processing for wireless communications, and is an inventor on over 15 recent 5GIC patents addressing bottleneck problems in 5G systems.
M. Rahmani et al.: Multi-Agent Reinforcement Learning-Based PA for CF-mMIMO Systems

MANIJEH BASHAR (Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Guilan, Iran, in 2009, the M.Sc. degree (Hons.) in communication systems engineering from the Shiraz University of Technology, Iran, in 2013, and the Ph.D. degree in communications engineering from the University of York, U.K., in 2019. In 2017, she was an Academic Visitor with the Department of Electronics and Nanoengineering, Aalto University, Espoo, Finland, with a Short Term Scientific Mission (STSM) Scholarship Award from European COST-IC1004 “Cooperative Radio Communications for Green Smart Environments.” She was with the Institute for Communication Systems (ICS) and the Home of 5G Innovation Centre (5GIC), University of Surrey, at the time of this work. She is currently with British Telecommunications (BT) Company. She received the K. M. Stott Prize for excellent Ph.D. research in electronics engineering from the University of York, in 2019. She has been awarded first place (based on jury) in the IEEE WCNC’18 three-minute Ph.D. thesis competition for her research in cell-free massive MIMO. She has been nominated for Departmental Prize for Excellence in Research at the University of Surrey, in 2019. She has been a member of Technical Program Committees (TPC) for the IEEE ICC 2020.

MÉROUANE DEBBAH (Fellow, IEEE) received the M.Sc. and Ph.D. degrees from the Ecole Normale Supérieure Paris-Saclay, France. He is currently a Chief Researcher at the Technology Innovation Institute, Abu Dhabi. He is also a Professor at CentraleSupélec and an Adjunct Professor with the Department of Machine Learning, Mohamed bin Zayed University of Artificial Intelligence. He was with the Motorola Laboratories, Saclay, France, from 1999 to 2002, and also with the Vienna Research Center for Telecommunications, Vienna, Austria, until 2003. From 2003 to 2007, he was an Assistant Professor with the Mobile Communications Department, Institut Eurecom, Sophia Antipolis, France. In 2007, he was appointed as a Full Professor at CentraleSupélec, Gif-sur-Yvette, France. From 2007 to 2014, he was the Director of the Alcatel-Lucent Chair on Flexible Radio. From 2014 to 2021, he was the Vice President of the Huawei France Research Center. He was jointly the Director of the Mathematical and Algorithmic Sciences Laboratory and the Director of the Lagrange Mathematical and Computing Research Center. Since 2021, he has been leading the AI & Digital Science Research Centers, Technology Innovation Institute. He has managed eight EU projects and more than 24 national and international projects. His research interests include fundamental mathematics, algorithms, statistics, information, and communication sciences research. He is a fellow of WWRF, Eurasip, AAIA, and Institut Louis Bachelier, and a Membre Emerite of SEE. He was a recipient of the ERC Grant MORE (Advanced Mathematical Tools for Complex Network Engineering), from 2012 to 2017. He was a recipient of the Mario Boella Award in 2005, the IEEE Glavieux Prize Award in 2011, the Qualcomm Innovation Prize Award in 2012, the 2019 IEEE Radio Communications Committee Technical Recognition Award, and the 2020 SEE Blondel Medal. He received more than 20 best paper awards, among which the 2007 IEEE GLOBECOM Best Paper Award, the Wi-Opt 2009 Best Paper Award, the 2010 Newcom++ Best Paper Award, the WUNCogCom Best Paper 2012 and 2013 Award, the 2014 WCNC Best Paper Award, the 2015 ICC Best Paper Award, the 2015 IEEE Communications Society Leonard G. Abraham Prize, the 2015 IEEE Communications Society Fred W. Ellersick Prize, the 2016 IEEE Communications Society Best Tutorial Paper Award, the 2016 European Wireless Best Paper Award, the 2017 EURASIP Best Paper Award, the 2018 IEEE Marconi Prize Paper Award, the 2019 IEEE Communications Society Young Author Best Paper Award, the 2021 EURASIP Best Paper Award, the 2021 IEEE Marconi Prize Paper Award, the 2022 IEEE Communications Society Outstanding Paper Award, the 2022 ICC Best Paper Award, and the Valuetools 2007, Valuetools 2008, CrownCom 2009, Valuetools 2012, SAM 2014, and 2017 IEEE Sweden VT-COM-IT Joint Chapter best student paper awards. He is an Associate Editor-in-Chief of the journal Random Matrix: Theory and Applications. He was an Associate Area Editor and a Senior Area Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, from 2011 to 2013 and from 2013 to 2014, respectively. From 2021 to 2022, he serves as an IEEE Signal Processing Society Distinguished Industry Speaker.