

Deep Reinforcement Learning Based Joint Cooperation Clustering and Downlink Power Control for Cell-Free Massive MIMO

Du Mingjun¹, Sun Xinghua^{1,*}, Zhang Yue², Wang Junyuan³, Liu Pei^{4,5}

¹ School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 518107, China

² Department of Electronic and Information Engineering, Shantou University, Shantou 515063, China

³ College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China

⁴ School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China

⁵ Zhongshan Institute of Advanced Engineering Technology of Wuhan University of Technology, Zhongshan 528437, China

* The corresponding author, email: sunxinghua@mail.sysu.edu.cn

Cite as: M. Du, X. Sun, *et al.*, “Deep reinforcement learning based joint cooperation clustering and downlink power control for cell-free massive mimo,” *China Communications*, 2024, vol. 21, no. 11, pp. 1-14. **DOI:** 10.23919/JCC.fa.2024-0217.202411

Abstract: In recent times, various power control and clustering approaches have been proposed to enhance overall performance for cell-free massive multiple-input multiple-output (CF-mMIMO) networks. With the emergence of deep reinforcement learning (DRL), significant progress has been made in the field of network optimization as DRL holds great promise for improving network performance and efficiency. In this work, our focus delves into the intricate challenge of joint cooperation clustering and downlink power control within CF-mMIMO networks. Leveraging the potent deep deterministic policy gradient (DDPG) algorithm, our objective is to maximize the proportional fairness (PF) for user rates, thereby aiming to achieve optimal network performance and resource utilization. Moreover, we harness the concept of “divide and conquer” strategy, introducing two innovative methods termed alternating DDPG (A-DDPG) and hierarchical DDPG (H-DDPG). These approaches aim to decompose the intricate joint optimization problem into more manageable sub-problems, thereby facilitating a more efficient resolution process. Our findings unequivocally

showcase the superior efficacy of our proposed DDPG approach over the baseline schemes in both clustering and downlink power control. Furthermore, the A-DDPG and H-DDPG obtain higher performance gain than DDPG with lower computational complexity.

Keywords: cell-free massive MIMO; clustering; deep reinforcement learning; power control

I. INTRODUCTION

Cell-free massive multiple-input multiple-output (CF-mMIMO) technology [1] has emerged as a highly promising approach for the next-generation communication systems. It offers seamless handover-free connectivity and consistently high-quality service for all users, marking a significant advancement in wireless communication technology. In the paradigm of CF-mMIMO, a multitude of distributed access points (APs) collaboratively serve a large number of users simultaneously, leveraging the shared time-frequency resources. Unlike traditional cellular networks with rigid cell boundaries, this approach allows for flexible resource allocation and efficient utilization [2–4]. Each AP is linked to a central processing unit (CPU)

Received: May 13, 2024

Revised: Aug. 18, 2024

Editor: Li Changle

through a backhaul connection, employing conjugate beamforming techniques to transmit data to all users.

It is widely recognized that the scenario where all APs provide service to every user will lead to a significant backhaul overhead. To make backhaul burden acceptable, several methods have been proposed. The concept of user-centric was first introduced in [5] and had been applied in CF-mMIMO [6], where APs exclusively serve the users exhibiting the strongest channel gains. Similarly, [7, 8] improved user association and AP selection methods to reduce interference. Another approach is the clustered networking scheme, which arranges users and APs into distinct subnetworks [9, 10]. In particular, the study presented in [9] explored clustering methods aimed at maximizing the quantity of subnetworks while concurrently constraining interference among them. Furthermore, [10] took into account the minimum per-user rate constraints and introduced a novel clustering algorithm to meet this criterion. The concept of clustering was further utilized in [11] for pilot allocation, introducing a revised K-means clustering scheme. Moreover, in [12], to augment the efficacy of the overall clustering procedure, a learning-based algorithm grounded in K-means++ was devised.

Besides, downlink power control also plays a pivotal role in upgrading user rates. The power control algorithm delineated in [1] maximized the minimum user rate and guaranteed high-quality service across the entire network. Additionally, [13] proposed both alternating and sequential optimization methods to maximize either the sum-rate of the whole system or the minimum user rate. Furthermore, a fractional power control policy for uplink CF-mMIMO systems was outlined in [14]. This method can be completely distributed and managed because it relies solely on large-scale quantities. Moreover, machine learning-based schemes, like supervised learning, have demonstrated the capability to closely approximate the outcomes of bisection algorithm with high precision while concurrently decreasing computational complexity [15].

Yet, the majority of power control schemes overlook the clustering strategies. These two aspects are deeply interconnected and complement each other. Clustering reduce the scope of power control and APs don't need to allocate power to all users (UEs). Power control provides better energy allocation policy within each cluster, aiming at attaining spatial multiplexing gain

for all the clusters. Ignoring either one can lead to significant performance degradation. Thus, both power control and clustering play vital roles in enhancing the performance for networks, underscoring the necessity for their integrated design. This joint optimization problem poses a significant challenge due to the combination of continuous power control space and discrete clustering space. It is known to be NP-Hard [16] and cannot be efficiently solved in polynomial time. In [17], a joint AP selection and power control approach was developed to minimize total energy consumption using successive convex optimization. Such iteration-based algorithms can be used to obtain suboptimal solutions, but they entail high computational complexity, which need to be approximated by learning-based approaches [18].

To tackle the joint optimization problem, recent studies have introduced reinforcement learning (RL) based approaches [19] to capitalize on their predictive capabilities. As an illustration, [20] formulated a scheme employing deep reinforcement learning (DRL) to jointly optimize content caching and clustering in CF-mMIMO. [21] performed dynamic clustering of APs and flexible channel assignment in Wi-Fi networks using RL. A recent paper [22] developed a multi-agent DRL approach for clustering and AP selection. Additionally, [23] introduced a model-free DRL approach to solve the max-sum, max-min, and max-product power control problem for CF-mMIMO. [24] formulated a joint power allocation and flexible duplexing problem and proposed an asynchronous advantage actor-critic scheme. Our previous work [25] developed a deep deterministic policy gradient (DDPG) based algorithm to solve the joint clustering and downlink power control problem in CF-mMIMO.

While DRL excels at solving optimization problems, its limitations become apparent as the quantity of users and APs increases. With the state space and action space growing exponentially, the curse of dimensionality significantly reduces the performance gain and convergence efficiency of DRL. To tackle such issue, we harness the concept of “divide and conquer”, and develop two frameworks to break down the initial joint optimization problem into independent two sub-problems. In the first framework, we adopt the methodology of alternating optimization [26], wherein clustering and power control are optimized alternately and iteratively. This approach aims

to mitigate the complexity arising from the simultaneous optimization of both tasks. In the second one, we employ hierarchical RL (HRL) [27] to divide the action space into upper-level clustering and lower-level power control layers. HRL offers a promising avenue for reducing the complexity of the joint optimization problem and enhances the learning efficiency through the reuse of lower-layered policies [28–30].

In particular, in this study, we consider CF-mMIMO systems and address the challenge of jointly optimizing clustering and downlink power control. Building upon our previous work [25], we introduce two novel methods named alternating DDPG (A-DDPG) and hierarchical DDPG (H-DDPG) based on the aforementioned frameworks. The main contributions of this paper are summarized as follows:

- We formulate a mixed-integer non-convex problem for maximizing proportional fairness (PF) of user rate. Both A-DDPG and H-DDPG partition the initial problem into more manageable sub-problems, thereby enhancing adaptability and flexibility in handling complex scenarios.
- In A-DDPG, we adopt an alternating optimization approach where clustering (or power control) is iteratively optimized while the other task remains fixed. This alternation between the two sub-problems allows for efficient optimization.
- H-DDPG implements a two-layer control system operating on different time scales. It treats clustering and power control asymmetrically by running an upper-level clustering process less frequently than the lower-level power control. The reward for system-level clustering is derived from the link-level power control.
- Our findings indicate that A-DDPG and H-DDPG achieve better performance compared to DDPG with lower computational complexity, and H-DDPG slightly outperforms A-DDPG.

The subsequent sections of this paper are outlined as follows. Section II delineates the system model and formulates the joint optimization problem. The DDPG-based approaches are introduced in Section III. Numerical results and discussion are presented in Section IV. Lastly, Section V provides the conclusion for the whole paper.

Notations: The superscripts $()^*$, $()^T$, and $()^H$ denote conjugate, transpose, and conjugate-transpose, respectively.

$\mathcal{CN}(0, \sigma^2)$ denotes circularly symmetric complex Gaussian distribution and $\mathcal{N}(0, \sigma^2)$ stands for real-valued Gaussian distribution. $\mathbb{E}\{\cdot\}$ denotes the statistical expectation.

II. SYSTEM MODEL

We focus on the downlink CF-mMIMO network including K users (or user equipments, UEs) and M APs. Let $\mathcal{K}$ and $\mathcal{M}$ represent the sets of all UEs and APs, respectively. UEs and APs are each equipped with a single antenna and are randomly positioned. All APs connect to a CPU via the backhaul network and APs collaboratively provide service to UEs within a subnetwork (or a cluster). The channel coefficient between the k -th UE and the m -th AP can be described as

$$g_{mk} = \beta_{mk}^{1/2} h_{mk}, \quad (1)$$

where β_{mk} and h_{mk} represent the large-scale fading and the small-scale fading, respectively. Each coherence interval includes three main phases: uplink training, downlink data transmission, and uplink data transmission. In this work, we only consider the downlink transmission, and hence, the uplink transmission is ignored. We assume that β_{mk} is priorly known and h_{mk} is static during each coherence interval.

2.1 Uplink Training

During the period of uplink training, all K UEs transmit dedicated pilot signals to APs. Subsequently, each AP employs these received pilot signals to estimate the channel for all UEs. Let τ_p indicate the duration of uplink training (measured in samples) of every coherence interval, thus the pilot vector utilized by the k -th UE can be expressed as $\sqrt{\tau_p} \boldsymbol{\varphi}_k \in \mathbb{C}^{\tau_p \times 1}$, where $\|\boldsymbol{\varphi}_k\|^2 = 1$. Then, the $\tau_p \times 1$ vector of pilot signals received at the m -th AP is given by

$$\mathbf{y}_{p,m} = \sqrt{\tau_p \rho_p} \sum_{k=1}^K g_{mk} \boldsymbol{\varphi}_k + \mathbf{w}_{p,m}, \quad (2)$$

where $\mathbf{w}_{p,m}$ represents the vector of additive noise at the m -th AP and ρ_p indicates the normalized signal-to-noise ratio (SNR) of each pilot symbol. The elements of $\mathbf{w}_{p,m}$ are independent and identically distributed $\mathcal{CN}(0, 1)$ random variables. In order to esti-

mate the channel g_{mk} for all $k = 1, 2, \dots, K$, each AP projects $\mathbf{y}_{p,m}$ onto $\boldsymbol{\varphi}_k^H$ and has:

$$\begin{aligned}\tilde{y}_{p,mk} &= \boldsymbol{\varphi}_k^H \mathbf{y}_{p,m} \\ &= \sqrt{\tau_p \rho_p} g_{mk} + \sqrt{\tau_p \rho_p} \sum_{k' \neq k}^K g_{mk'} \boldsymbol{\varphi}_k^H \boldsymbol{\varphi}_{k'} + \boldsymbol{\varphi}_k^H \mathbf{w}_{p,m}.\end{aligned}\quad (3)$$

Considering the minimum mean-square error (MMSE) estimation of g_{mk} given $\tilde{y}_{p,mk}$, we need to minimize the expected distance between the estimated channel and the true channel, i.e., $\mathbb{E}\{|c_{mk}\tilde{y}_{p,mk} - g_{mk}|^2\}$, where c_{mk} is the linear coefficient to be optimized. Taking the derivative with respect to c_{mk} , we can obtain

$$\begin{aligned}\frac{\partial \mathbb{E}\{|c_{mk}\tilde{y}_{p,mk} - g_{mk}|^2\}}{\partial c_{mk}} \\ = 2c_{mk}\mathbb{E}\{|\tilde{y}_{p,mk}|^2\} - 2\mathbb{E}\{\tilde{y}_{p,mk}^* g_{mk}\}.\end{aligned}\quad (4)$$

Let the result of the above formula be 0, we can obtain c_{mk} and estimated channel $\hat{g}_{mk}$

$$\begin{aligned}\hat{g}_{mk} = c_{mk}\tilde{y}_{p,mk} &= \frac{\mathbb{E}\{\tilde{y}_{p,mk}^* g_{mk}\}}{\mathbb{E}\{|\tilde{y}_{p,mk}|^2\}} \tilde{y}_{p,mk} \\ &= \frac{\sqrt{\tau_p \rho_p} \beta_{mk}}{\tau_p \rho_p \sum_{k'=1}^K \beta_{mk'} |\boldsymbol{\varphi}_k^H \boldsymbol{\varphi}_{k'}|^2 + 1} \tilde{y}_{p,mk}.\end{aligned}\quad (5)$$

2.2 Downlink Data Transmission

Let us denote $\mathcal{M}(k)$ as the set of APs serving the k -th UE, and $\mathcal{K}(m)$ as the set of UEs served by the m -th AP. We can utilize the $M \times K$ adjacency matrix $\mathbf{A} = (A_{mk})$ to describe the connection status between APs and UEs:

$$A_{mk} = \begin{cases} 1, & \text{If AP } m \text{ serves UE } k, \\ 0, & \text{otherwise.} \end{cases}\quad (6)$$

It is obvious that $A_{mk} = 1$ if AP $m \in \mathcal{M}(k)$ for all $k = 1, 2, \dots, K$, or UE $k \in \mathcal{K}(m)$ for all $m = 1, 2, \dots, M$. We assume that all APs provide service to UEs and all UEs are ensured to be served within the clustered CF-mMIMO networks. Particularly, a cluster comprises multiple UEs and APs, and APs collaboratively serve UEs within a cluster as shown in

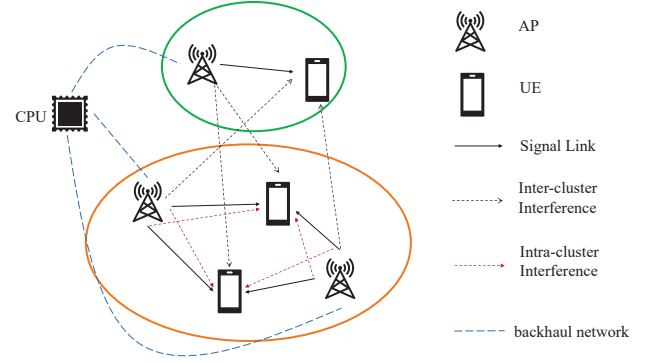


Figure 1. Architecture of the CF-mMIMO networks, APs collaboratively provide service to UEs within the same cluster. Both inter-cluster interference and intra-cluster interference are included.

Figure 1. Signals only come from within the cluster, but interference includes both intra-cluster interference and inter-cluster interference. Let N indicate the quantity of clusters and C_n be the set of UEs and APs in the n -th cluster, we can obtain

$$\bigcup_{n=1}^N C_n = \mathcal{M} \cup \mathcal{K} = \bigcup_{k=1}^K \mathcal{M}(k) \cup \bigcup_{m=1}^M \mathcal{K}(m), \quad (7)$$

and

$$C_n \cap C_{n'} = \emptyset, \forall n' \neq n. \quad (8)$$

A clustering scheme $\mathcal{C}$ can be expressed as $\{C_1, C_2, \dots, C_N\}$. It is straightforward to verify that if UE k , UE k' , AP m , AP m' are in the same cluster, then we have $\mathcal{M}(k) = \mathcal{M}(k')$, $\mathcal{K}(m) = \mathcal{K}(m')$. We assume that the channels are reciprocal and APs regard the uplink channel estimation as the true channel. The signal transmitted by the m -th AP using conjugate beamforming can be expressed as

$$x_m = \sqrt{\rho_d} \sum_{k \in \mathcal{K}(m)} \sqrt{\eta_{mk}} \hat{g}_{mk}^* q_k, \quad (9)$$

where q_k represents the unit symbol designated for the k -th UE and it follows $\mathbb{E}\{|q_k|^2\} = 1$. Let ρ_d represent the normalized transmit SNR by the m -th AP and η_{mk} indicates the power control coefficients which satisfies $\mathbb{E}\{|x_m|^2\} \leq \rho_d$. Optimizing the power control coefficients is crucial. The signal received at the k -th

UE can be described as follows:

$$\begin{aligned}
y_k &= \sum_{m=1}^M g_{mk} x_m + w_k \\
&= \sqrt{\rho_d} \sum_{m=1}^M \sum_{k' \in \mathcal{K}(m)} \sqrt{\eta_{mk'}} g_{mk} \hat{g}_{mk'}^* q_{k'} + w_k \\
&= \sqrt{\rho_d} \sum_{k'=1}^K \sum_{m \in \mathcal{M}(k')} \sqrt{\eta_{mk'}} g_{mk} \hat{g}_{mk'}^* q_{k'} + w_k \\
&= \underbrace{\sqrt{\rho_d} \sum_{m \in \mathcal{M}(k)} \sqrt{\eta_{mk}} g_{mk} \hat{g}_{mk}^* q_k}_{\text{desired signal}} \\
&\quad + \underbrace{\sqrt{\rho_d} \sum_{k' \neq k}^K \sum_{m \in \mathcal{M}(k')} \sqrt{\eta_{mk'}} g_{mk} \hat{g}_{mk'}^* q_{k'}}_{\text{multiuser interference}} + w_k,
\end{aligned} \tag{10}$$

where $w_k \sim \mathcal{CN}(0, 1)$ represents the additive noise at the k -th UE. The downlink Signal-to-Interference plus Noise-Ratio (SINR) for the k -th UE can be denoted as

$$\gamma_k = \frac{\rho_d \left| \sum_{m \in \mathcal{M}(k)} \sqrt{\eta_{mk}} g_{mk} \hat{g}_{mk}^* \right|^2}{\rho_d \sum_{k' \neq k}^K \left| \sum_{m \in \mathcal{M}(k')} \sqrt{\eta_{mk'}} g_{mk} \hat{g}_{mk'}^* \right|^2 + 1}. \tag{11}$$

According to Shannon's formula, the downlink achievable rate for the k -th UE R_k is denoted by $\log_2(1 + \gamma_k)$. Mathematically, the joint optimization problem can be formulated as follows:

$$\begin{aligned}
&\max_{\mathcal{C}, \boldsymbol{\eta}} \quad \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{k=1}^K \log_2 R_{k,t} \\
&\text{s.t.} \quad \bigcup_{n=1}^N C_n = \mathcal{M} \cup \mathcal{K}, \\
&\quad C_n \cap C_{n'} = \emptyset, \forall n' \neq n, \\
&\quad \sum_{k \in \mathcal{K}(m)} \eta_{mk} |\hat{g}_{mk}|^2 \leq 1, m = 1, 2, \dots, M, \\
&\quad \eta_{mk} \geq 0, m = 1, 2, \dots, M, k = 1, 2, \dots, K.
\end{aligned} \tag{12}$$

With PF as our optimization objective, our aim is to collaboratively design a cooperation clustering method $\mathcal{C} = \{C_1, C_2, \dots, C_N\}$ and a power con-

trol matrix $\boldsymbol{\eta} = (\eta_{mk})$, ensuring that the long-term global average logarithmic sum of user rates is maximized. $R_{k,t}$ represents the downlink achievable rate at time step t for the k -th UE. It's worth noting that $\eta_{mk} = 0$ if $A_{mk} = 0, \forall m, k$. Employing the sum of logarithms as the objective function instead of a direct summation helps mitigate traffic starvation for edge users. Attributed to the non-convex feature for the above optimization problem, traditional optimization methods become intractable, especially for large networks. Therefore, we develop DDPG-based strategies in Section III to address this challenge.

III. DDPG-BASED ALGORITHM

3.1 Naive DDPG Approach

3.1.1 State, Action, and Reward

The action at time step t involves clustering action a_t^C and power control action a_t^P simultaneously. Within this scheme, the action directly influences both the power control matrix and the adjacency matrix, which can be denoted as

$$a_t = \{a_t^C, a_t^P\} = \{\mathbf{A}_t, \boldsymbol{\eta}_t\}, \tag{13}$$

where the subscript t denotes the matrix at time step t . The state ought to be a comprehensive set of information stored within the CPU, capable of characterizing the network effectively. Since the accurate channel gain is not available, we utilize large-scale fading instead of real channel gain in [25]. In this scheme, the state is composed of large-scale fading $\boldsymbol{\beta}_t = \{\beta_{mk,t} : m = 1, 2, \dots, M, k = 1, 2, \dots, K\}$, clustering and power control actions in the last time step, and each UE's SINR $\Gamma_t = \{\gamma_{k,t} : k = 1, 2, \dots, K\}$, i.e.,

$$s_t = \{\boldsymbol{\beta}_t, a_{t-1}, \Gamma_t\}. \tag{14}$$

The reward can be defined as follows based on the objective function outlined in (12):

$$r_t = \sum_{k=1}^K \log_2 R_{k,t}. \tag{15}$$

3.1.2 DDPG Approach

Given the extensive and continuous nature of the action space, a value-based algorithm such as deep Q-network (DQN) is considered unsuitable for addressing this joint optimization problem. Hence, we opt for the DDPG algorithm, categorized as an actor-critic (AC) method, as it is better suited for handling continuous spaces.

The actor networks with weights θ leverage the policy function $\mu(s_t|\theta)$ to generate an action. The critic networks with weights ϕ utilize the action-value function $Q(s_t, a_t|\phi)$ to assess the actor's performance. The Q function depicts the anticipated cumulative reward and conforms to the Bellman equation:

$$\begin{aligned} Q(s_t, a_t) &= \mathbb{E}[G_t | s_t, a_t] = \mathbb{E}\left[\sum_{k=0}^{T-t} \gamma^k r_{t+k} | s_t, a_t\right] \\ &= \mathbb{E}\left[r_t + \gamma \max_{a'} q_*(s_{t+1}, a') | s_t, a_t\right], \end{aligned} \quad (16)$$

where γ denotes the discount factor and G_t is the cumulative discounted reward. As the agent's goal is to maximize G_t , parameters of the actor networks should be adjusted in the direction of the gradient of the Q function to obtain the best weights. Utilizing a deterministic policy $a_t = \mu(s_t|\theta)$ and the performance $J(\theta)$ is measured by action-value function Q , the gradient of actor can be denoted as:

$$\nabla_{\theta} J(\theta) = \nabla_{\theta} \mathbb{E}[Q(s_t, \mu(s_t|\theta)|\phi)]. \quad (17)$$

The parameters θ of the actor networks are updated using gradient descent:

$$\theta \leftarrow \theta + \alpha^{\theta} \nabla_{\theta} J(\theta), \quad (18)$$

where α^{θ} denotes the learning rate of the actor networks. As the critic networks are responsible for evaluating actions, the output of critic networks is Q function, which needs to approximate the genuine value of the action. However, during parameter updates, the iteration for Q function in (16) tends to be unstable and challenging to converge. To tackle this issue, DDPG imitates DQN and employs target networks for both actor and critic with parameters θ' and ϕ' respectively to facilitate convergence. The critic networks' loss function takes the form of temporal difference error.

The Q function of the next state is forecasted by the target networks, while the Q function of the current state is obtained from the critic networks. Therefore, the loss function and updating mechanism of the critic networks can be expressed as:

$$L(\phi) = \mathbb{E}[(r_t + \gamma Q'(s_{t+1}, \mu'(s_{t+1}|\theta'))|\phi') - Q(s_t, a_t|\phi)]^2, \quad (19)$$

and

$$\phi \leftarrow \phi - \alpha^{\phi} \nabla_{\phi} L(\phi), \quad (20)$$

similar to actor networks, α^{ϕ} denotes the learning rate of the critic networks. In practical implementation, the expectations of $L(\phi)$ and $J(\theta)$ can be approximated through sampled averages, as acquiring the true probability distribution of states poses challenges. At each time step, the four-element tuple (s_t, a_t, r_t, s_{t+1}) is preserved within the experience buffer $\mathcal{D}$, and a mini-batch of tuples $\mathcal{B}$ is randomly selected from $\mathcal{D}$ for updating the networks. The experience buffer has a capacity of D and older experiences are substituted with new transitions if the buffer reaches full capacity. Finally, target networks undergo a gradual update process:

$$\theta' \leftarrow \tau \theta + (1 - \tau) \theta', \quad (21)$$

$$\phi' \leftarrow \tau \phi + (1 - \tau) \phi', \quad (22)$$

where $\tau \ll 1$, indicating a gradual change for target networks. The entire learning procedure for DDPG is outlined in Algorithm 1.

Algorithm 1. DDPG process.

- 1: Initialize weights θ for actor networks and weights ϕ for critic networks randomly.
 - 2: Initialize the weights of the actor target network $\theta' = \theta$ and weights for the critic target network $\phi' = \phi$.
 - 3: Initialize the experience buffer $\mathcal{D}$ with a capacity of D .
 - 4: **for** episode = 1 to MAX-EPISODE **do**
 - 5: Establish the initial state s_1 .
 - 6: Adjust the exploration noise $\mathcal{N}$.
 - 7: **for** $t = 1$ to T **do**
 - 8: Perform the action $a_t = \mu(s_t|\theta) + \mathcal{N}_t$.
 - 9: Obtain the reward r_t and next state s_{t+1} .
 - 10: **if** $|\mathcal{D}| < D$ **then**
 - 11: Store experience (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$.
 - 12: **else**
-

```

13:     Substitute the old experience with the new
        experience.
14:   end if
15:   if  $|\mathcal{D}| \geq |\mathcal{B}|$  then
16:     Sample a mini-batch  $\mathcal{B}$  from  $\mathcal{D}$  randomly.
17:     Update the actor network weights  $\theta$  and
        critic network weights  $\phi$  using equations
        (18) and (20).
18:     Update the weights for target actor network
         $\theta'$  and target critic network  $\phi'$  using equa-
        tions (21) and (22).
19:   end if
20:    $s_t \leftarrow s_{t+1}$ .
21: end for
22: end for

```

3.2 A-DDPG Approach

The naive DDPG approach incorporates both clustering and power control into its action, leading to a complex search space. An efficient idea is alternating optimization, wherein we optimize clustering and power control alternately exploiting two agents corresponding to these actions. We name it alternating DDPG (A-DDPG), which is shown in Figure 2. Specifically, we fix the power control scheme and continuously optimize clustering until a satisfactory reward is achieved. Then, we fix the clustering scheme and optimize power control until the reward exceeds a certain threshold. These two steps are iterated repeatedly until convergence is reached. It's worth noting that when we fix the power control scheme, we indicate that power control matrix $\boldsymbol{\eta} = (\eta_{mk})$ is fixed. While optimizing clustering scheme, the agent can select different power control coefficients η_{mk} in the fixed $\boldsymbol{\eta}$ by changing A_{mk} in the adjacency matrix.

The state and reward are the same with naive DDPG for each agent, i.e., $s_t^C = s_t^P = \{\beta_t, a_{t-1}^C, a_{t-1}^P, \Gamma_t\}$ and $r_t^C = r_t^P = \sum_{k=1}^K \log_2 R_{k,t}$. Clustering action a^C and power control action a^P are determined by two agents, respectively. At the beginning of the algorithm, clustering and power control are initialized and clustering agent continuously selects a^C for Δn steps. Throughout this interval, the reward for each step is deposited into reward buffer $\mathcal{D}^R$. Then, the clustering agent persists until the reward reaches a certain threshold $\mathcal{D}_\alpha^R$ (upper α percentile of $\mathcal{D}^R$), ensuring the effectiveness of the clustering scheme. Analogously,

the power control agent iterates through similar steps. Both agents operate alternately until the predefined step limit is surpassed. The complete A-DDPG learning process is outlined in Algorithm 2 (Lines 7-20 for clustering optimization and Lines 22-35 for power control optimization).

Algorithm 2. A-DDPG process.

```

1: Randomly initialize critic network weights  $\phi^C$ 
   and actor network weights  $\theta^C$  for clustering, and
   critic network weights  $\phi^P$  and actor network
   weights  $\theta^P$  for power control.
2: Initialize target network  $\theta^{C'}, \phi^{C'}, \theta^{P'}$  and  $\phi^{P'}$ 
   with parameters  $\theta^{C'} \leftarrow \theta^C, \phi^{C'} \leftarrow \phi^C, \theta^{P'} \leftarrow$ 
    $\theta^P$ , and  $\phi^{P'} \leftarrow \phi^P$ .
3: Initialize the experience buffer  $\mathcal{D}^C$  and  $\mathcal{D}^P$  with
   capacity  $D$ .
4: Set  $t = 1$  and the initial state  $s_1$ .
5: while  $t < T \times \text{MAXEPISODE}$  do
6:   Initialize the reward buffer  $\mathcal{D}^R$ .
7:   for  $n = 1$  to  $\Delta n$  do
8:     Obtain current clustering state  $s_t^C$  and update
       exploration noise  $\mathcal{N}$ .
9:     Execute clustering action  $a_t^C = \mu(s_t^C | \theta^C) +$ 
        $\mathcal{N}$ .
10:    Obtain the clustering reward  $r_t^C$  and next state
        $s_{t+1}^C$ .
11:    Store reward  $r_t^C$  in  $\mathcal{D}^R$ .
12:    Store experience  $(s_t^C, a_t^C, r_t^C, s_{t+1}^C)$  in  $\mathcal{D}^C$ .
13:    Sample a mini-batch  $\mathcal{B}^C$  from  $\mathcal{D}^C$  randomly.
14:    Update the weights  $\theta^C$  and  $\phi^C$  using equation
       (18) and (20).
15:    Update the target network weights  $\theta^{C'}$  and
        $\phi^{C'}$  using equation (21) and (22).
16:     $t \leftarrow t + 1$ 
17:   end for
18:   repeat
19:     Step 8-16
20:   until  $r_t^C > \mathcal{D}_\alpha^R$ 
21:   Initialize the reward buffer  $\mathcal{D}^R$ .
22:   for  $n = 1$  to  $\Delta n$  do
23:     Obtain current power control state  $s_t^P$  and up-
       date exploration noise  $\mathcal{N}$ .
24:     Execute clustering action  $a_t^P = \mu(s_t^P | \theta^P) +$ 
        $\mathcal{N}$ .
25:     Obtain the clustering reward  $r_t^P$  and next state
        $s_{t+1}^P$ .

```

for upper-level clustering is lower than power control. In particular, the clustering agent selects a^C for one step and the power control agent continuously optimizes a^P with clustering policy fixed. Then, clustering agent receives a reward and updates parameters, and selects a new action.

The clustering state and power control state can be denoted as $s_t^C = \{\beta_t, a_{t-1}^C, \Gamma_t\}$ and $s_t^P = \{\beta_t, a_{t-1}^P, \Gamma_t\}$, respectively. Power control reward $r_t^P = \sum_{k=1}^K \log_2 R_{k,t}$ is consistent with naive DDPG and A-DDPG while clustering reward is the maximum value during lower-level power control steps, i.e., $r^C = \max(\mathcal{D}^R)$. The pseudocode for H-DDPG is outlined in Algorithm 3 (Line 5 for the outer clustering stage and Line 9 for the inner power control stage).

Algorithm 3. *H-DDPG process.*

- 1: Randomly initialize critic network weights ϕ^C and actor network weights θ^C for clustering, and critic network weights ϕ^P and actor network weights θ^P for power control.
- 2: Initialize target network $\theta^{C'}, \phi^{C'}, \theta^{P'}$ and $\phi^{P'}$ with parameters $\theta^{C'} \leftarrow \theta^C, \phi^{C'} \leftarrow \phi^C, \theta^{P'} \leftarrow \theta^P$, and $\phi^{P'} \leftarrow \phi^P$.
- 3: Initialize the experience buffer $\mathcal{D}^C$ and $\mathcal{D}^P$ with capacity D .
- 4: Set the initial clustering state s_1^C .
- 5: **for** $n = 1$ to MAX-EPISODE **do**
- 6: Obtain current clustering state s_n^C and update exploration noise $\mathcal{N}$.
- 7: Execute clustering action $a_n^C = \mu(s_n^C | \theta^C) + \mathcal{N}$.
- 8: Initialize the power control reward buffer $\mathcal{D}^R$.
- 9: **for** $t = 1 + (n - 1)T$ to nT **do**
- 10: Obtain the power control state s_t^P .
- 11: Execute power control action $a_t^P = \mu(s_t^P | \theta^P) + \mathcal{N}_t$.
- 12: Obtain the reward r_t^P and next state s_{t+1}^P .
- 13: Store reward r_t^P in $\mathcal{D}^R$.
- 14: Store experience $(s_t^P, a_t^P, r_t^P, s_{t+1}^P)$ in $\mathcal{D}^P$.
- 15: Sample a mini-batch $\mathcal{B}^P$ from $\mathcal{D}^P$ randomly.
- 16: Update the weights θ^P and ϕ^P using equation (18) and (20).
- 17: Update the target network weights $\theta^{P'}$ and $\phi^{P'}$ using equation (21) and (22).
- 18: **end for**

- 19: Obtain the clustering reward $r_n^C = \max(\mathcal{D}^R)$ and next state s_{n+1}^C .
 - 20: Store experience $(s_n^C, a_n^C, r_n^C, s_{n+1}^C)$ in $\mathcal{D}^C$.
 - 21: Sample a mini-batch $\mathcal{B}^C$ from $\mathcal{D}^C$ randomly.
 - 22: Update the weights θ^C and ϕ^C using equations (18) and (20).
 - 23: Update the target network weights $\theta^{C'}$ and $\phi^{C'}$ using equations (21) and (22).
 - 24: **end for**
-

3.4 Computational Complexity

In the case of naive DDPG, the actor networks' input layer consists of $3MK + K$ neurons and the actor networks' output layer consists of $N(M + K) + MK$ neurons. Reshaping the output vector into a $(M + K) \times N$ matrix $\mathbf{X}$ and a $M \times K$ matrix $\mathbf{Y}$, $\mathbf{X}$ determines clustering scheme, i.e., AP (or UE) $j \in C_{\arg\max_k X_{jk}}$, and $\mathbf{Y}$ determines power control, i.e., $\eta_{mk} = \frac{1}{\sum_{k' \in \mathcal{K}(m)} \frac{\exp(Y_{mk'})}{\sum_{j \in \mathcal{K}(m)} \exp(Y_{mj})} |\hat{g}_{mk'}|^2 \sum_{j \in \mathcal{K}(m)} \exp(Y_{mj})}$. We guarantee that each AP (or UE) belong to only one cluster and clusters will not overlap. By taking the softmax operation on $\mathbf{Y}$, we normalize power control coefficients and η satisfies restrictions in (12). The number of the hidden layer neurons is the average of the input and output layers and the overall architecture of the network is the same as [25]. Hence, the computational complexity is the product of dimensions for three layers, i.e., $\mathcal{O}((3MK + K)(N(M + K) + MK)(3MK + K + N(M + K)))$. For A-DDPG, the input layer of the actor networks is the same as naive DDPG. The output dimensions of the actor networks are $N(M + K)$ and MK for clustering and power control, respectively. Since the network structure is the same with naive DDPG, the computational complexity for A-DDPG actor networks is the sum of clustering actor networks and power control actor networks, i.e., $\mathcal{O}((3MK + K)(N(M + K))(3MK + K + N(M + K)) + (3MK + K)MK(4MK + K))$. It is easy to prove that the computational complexity of A-DDPG is lower than that of naive DDPG. Besides, the clustering state's dimension for H-DDPG is smaller, leading to the lowest complexity. The conclusion for critic networks is similar to actor networks, so we neglect it for simplicity.

IV. SIMULATION RESULTS

We consider a CF-mMIMO network comprising 20 APs and 10 UEs randomly dispersed within a 1×1 km² square area. We evaluate our algorithm across 10 distinct random scenarios with varying topologies. Particularly, the large-scale fading coefficient β_{mk} includes shadow fading and path loss, which can be described as

$$\beta_{mk} = 10^{\frac{PL_{mk} + \sigma_{sh} z_{mk}}{10}}, \quad (23)$$

where $z_{mk} \sim \mathcal{N}(0, 1)$ denotes the shadow fading with standard deviation σ_{sh} , and PL_{mk} denotes the path loss model characterized by three slopes [1] which is specified below in dB:

$$PL_{mk} = \begin{cases} -140.7 - 35 \log_{10}(d_{mk}), & \text{if } d_{mk} > d_1, \\ -140.7 - 15 \log_{10}(d_1) - 20 \log_{10}(d_{mk}), & \text{if } d_0 < d_{mk} \leq d_1, \\ -140.7 - 15 \log_{10}(d_1) - 20 \log_{10}(d_0), & \text{if } d_{mk} \leq d_0, \end{cases} \quad (24)$$

where d_0 and d_1 are different reference distances, and d_{mk} is the distance between m -th AP and k -th UE. Note that there is no shadowing when $d_{mk} \leq d_1$. Besides, we employ the following process to characterize the time-varying channel [20]

$$h_{mk}(t+1) = \sqrt{1 - \epsilon^2} h_{mk}(t) + \epsilon \Delta, \quad (25)$$

where Δ is presumed to be independent and identically distributed as $\mathcal{CN}(0, 1)$ and $\epsilon \in [0, 1]$ represents the channel variation coefficient. The key parameters are outlined in Table 1. Since uplink training duration τ_p is less than K , τ_p mutually orthogonal pilots are assigned to τ_p users, and the left users are assigned with $K - \tau_p$ pilots which are not orthogonal with previous ones. The quantities $\bar{\rho}_p, \bar{\rho}_d$ are the transmit power for pilot symbols and downlink data, respectively. The corresponding normalized transmit SNRs ρ_p, ρ_d can be computed by dividing these powers by the noise power, which is the product of bandwidth, Boltzmann constant, noise temperature and noise figure.

We compare the proposed DDPG, A-DDPG and H-DDPG approaches with another three methods named K-means, Uniform, and UC-DDPG. In the case of K-

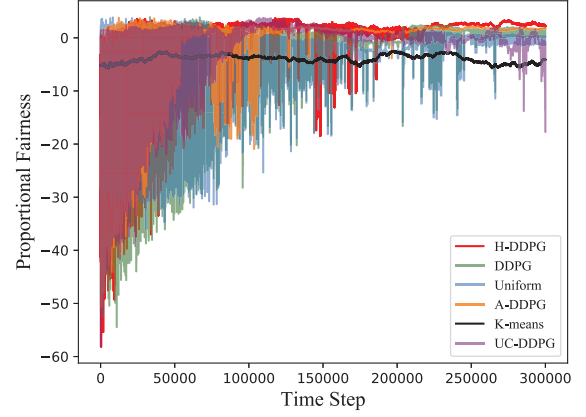


Figure 3. Training process across various methods.

means baseline, UEs are grouped together utilizing the K-means algorithm [11]. APs are allocated to the UE clusters according to their proximity to the centroid of the UE cluster, with preference given to the AP-cluster pair where the distance from the AP to the cluster centroid is the shortest. The power control method in K-means baseline follows the specific rule:

$$\eta_{mk} = \frac{1}{\sum_{k' \in \mathcal{K}(m)} |\hat{g}_{mk'}|^2}. \quad (26)$$

In this methodology [6], the power control coefficients for a single AP remain consistent across all served UEs. The transmit power for m -th AP to k -th UE is $\frac{\rho_d |g_{mk}|^2}{\sum_{k' \in \mathcal{K}(m)} |\hat{g}_{mk'}|^2}$, which indicates that power control is directly proportional to channel gains. In the case of the Uniform baseline, clustering approach remains consistent with DDPG, while the power control scheme aligns with (26). In the UC-DDPG approach, clustering scheme follows the user-centric approach while power control follows the DDPG-based method in [23]. The reward function in UC-DDPG is the product of each UE's SINR and each UE is served by APs with the largest channel gains. As the reward function in UC-DDPG is similar to our optimization goal, it is relatively fair to compare UC-DDPG with our proposed schemes. The numerical outcomes presented in this paper represent the average of the data collected during the last ten percent of the simulation duration.

Figure 3 illustrates the training curves across various methods in a specific network topology. Initially, all

Table 1. System parameters.

Parameters	Value
Uplink training duration τ_p	5
Pilot transmit power $\bar{p}_p$	200 mW
Downlink transmit power $\bar{p}_d$	200 mW
Reference distance d_0	10 m
Reference distance d_1	50 m
Shadow fading standard deviation σ_{sh}	8 dB
Noise figure	9 dB
Boltzmann constant	1.381×10^{-23} J/K
Noise temperature	290 K
Bandwidth	20 MHz
Channel variation coefficient ϵ	0.001
Discount factor γ	0.99
Actor learning rate α^θ	10^{-4}
Critic learning rate α^ϕ	10^{-3}
Update rate τ	0.001
Replay buffer capacity	25000
Batch size	128
T	3
MAX EPISODE	100000

methods (except K-means) exhibit poor performance as certain UEs experience traffic starvation. However, as the time steps progress, the learning-based methods gradually enhance their performance. In the last ten percent of the simulation duration, it's noticeable that H-DDPG and A-DDPG achieve the highest PF, while DDPG outperforms Uniform. UC-DDPG achieves good performance at about 100000-th time step but fails to keep up with environmental changes. It can also be observed that H-DDPG and DDPG converge at about 200000-th time step, and Uniform converges at about 250000-th time step. A-DDPG achieves the fastest convergence and can converge at approximately 150000-th time step, which can be attributed to the effectiveness of alternate optimization. In particular, alternating optimization effectively handles dependencies between multiple variables, ensuring high computational efficiency. Additionally, the jitter for H-DDPG and A-DDPG is smaller compared to DDPG, indicating better convergence.

Figure 4 displays the cumulative distribution functions (CDFs) across various approaches with different cluster numbers N . While comparing Uniform with K-mean, we observe that our DDPG-based clustering scheme surpasses the baseline K-means clustering scheme due to both Uniform and K-means exploit the same power control rule in (26). Likewise, upon comparing Uniform and DDPG, it is evident that the DDPG-based power control scheme outperforms the baseline in (26), because clustering approaches

are the same for Uniform and DDPG. Besides, our DDPG-based approaches outperform UC-DDPG, ensuring better fairness. Furthermore, it can be observed that the curves representing A-DDPG and H-DDPG roughly lie further to the right compared to DDPG, signifying superior performance. This can be attributed to the decomposition of the problem into two levels by A-DDPG and H-DDPG. The layered based approach allows for more flexible and efficient optimization decisions at different layers. Overall, A-DDPG strategy outperforms DDPG and UC-DDPG by up to 5.72%, 4.98%, 15.0% and 54.2%, 47.0%, 22.3% respectively in terms of average PF when $N = 2, 3, 4$. These data come to 5.20%, 6.54%, 19.7% and 53.5%, 49.2%, 27.4% when comparing H-DDPG with DDPG and UC-DDPG.

Figure 5 indicates that as N increases, the performance degrades for H-DDPG, A-DDPG and DDPG, and the gaps between Uniform and these methods would narrow. This is because each AP serves fewer UEs when N increases, leading to a smaller degree of freedom of power control and a diminished performance gap. The proportional fairness for UC-DDPG is a fixed value because no clusters are formed adopting user-centric approach, and cluster number N is not considered in UC-DDPG. If N continues to increase, the number of APs and UEs within a cluster will decrease. In an extreme case, each AP will serve a single UE, resulting in a performance similar to the user-centric approach, and power control is not needed in such case. These five lines will converge at one point and exhibit similar performance if N reaches K . When $N = 1$, the network reverts to traditional CF-mMIMO, as discussed in [1], where only power control is considered. In this scenario, H-DDPG, A-DDPG, and DDPG outperform multi-cluster cases because there are no clustering restrictions, leading to the largest solution space for power control. Furthermore, the performance gaps for H-DDPG across different cluster numbers are the smallest, indicating robustness. The disparity between H-DDPG and A-DDPG arises from the fact that alternating optimization may lead to suboptimal solutions, while the hierarchical approach is more suitable for this problem due to clustering and power control operating at different layers. H-DDPG with 1 cluster outperforms 2, 3, and 4 clusters by up to 5.43%, 8.45%, and 27.0% respectively in terms of average PF. These data come

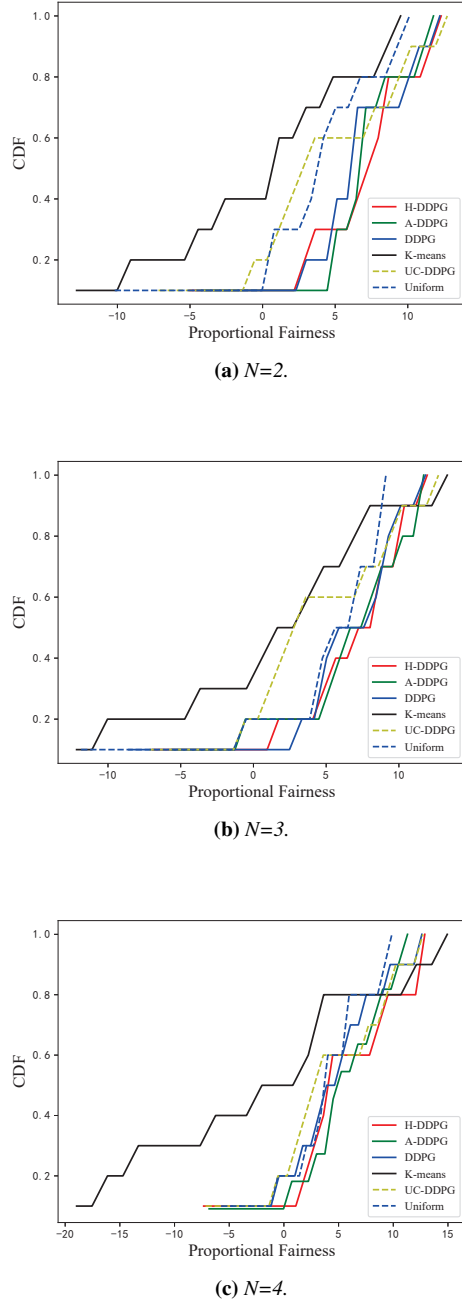


Figure 4. CDF for proportional fairness under different cluster numbers.

to 5.86%, 11.1%, and 33.4% and 12.6%, 17.3%, and 54.4% for A-DDPG and DDPG, respectively.

V. CONCLUSION

In this work, we present DDPG-based approaches to tackle the joint clustering and downlink power control problem for CF-mMIMO networks, with proportional

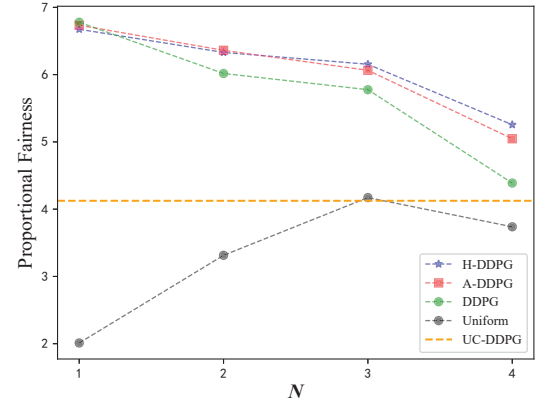


Figure 5. Average proportional fairness for different methods with different cluster numbers N .

fairness as the optimization objective. Furthermore, by decomposing the original problem into two sub-problems and training two separate agents, we introduce alternating DDPG and hierarchical DDPG to establish a collaborative optimization framework that effectively addresses the joint problem. The simulation results demonstrate that H-DDPG and A-DDPG outperform naive DDPG with lower computational complexity. H-DDPG is advantageous for its better effects, while A-DDPG stands out for its rapid convergence. Additionally, H-DDPG exhibits robustness in maintaining good performance across various cluster numbers. For future research, we will consider more APs and UEs with multiple antennas. Besides, we are interested in setting overlapped clusters and deactivating some APs for energy efficiency, which is more practical in real-world applications.

ACKNOWLEDGEMENT

The work of Sun Xinghua was supported by Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515012015. The work of Zhang Yue was supported in part by the National Natural Science Foundation of China under Grant 62201336, and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515011541. The work of Wang Junyuan was supported in part by the National Natural Science Foundation of China under Grant 62371344 and in part by the Fundamental Research Funds for the Central Universities. The work of Liu Pei was supported in part by Knowledge In-

novation Program of Wuhan-Shuguang Project under Grant 2023010201020316 and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515010247.

References

- [1] H. Q. Ngo, A. Ashikhmin, *et al.*, "Cell-free massive mimo versus small cells," *IEEE Transactions on Wireless Communications*, 2017, vol. 16, no. 3, pp. 1834–1850.
- [2] P. Liu, K. Luo, *et al.*, "Spectral efficiency analysis of cell-free massive mimo systems with zero-forcing detector," *IEEE Transactions on Wireless Communications*, 2020, vol. 19, no. 2, pp. 795–807.
- [3] D. Wang, M. Wang, *et al.*, "Performance of network-assisted full-duplex for cell-free massive mimo," *IEEE Transactions on Communications*, 2019, vol. 68, no. 3, pp. 1464–1478.
- [4] C. Liu, W. Feng, *et al.*, "Fairness-oriented hybrid precoding for massive mimo maritime downlink systems with large-scale csit," *China Communications*, 2018, vol. 15, no. 1, pp. 52–61.
- [5] L. Dai, S. Zhou, *et al.*, "Capacity with mrc-based macro-diversity in cdma distributed antenna systems," in *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*, 2002, vol. 1, pp. 987–991.
- [6] S. Buzzi and C. D'Andrea, "Cell-free massive mimo: User-centric approach," *IEEE Wireless Communications Letters*, 2017, vol. 6, no. 6, pp. 706–709.
- [7] C. D'Andrea and E. G. Larsson, "User association in scalable cell-free massive mimo systems," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 826–830.
- [8] H. Dao and S. Kim, "Effective channel gain-based access point selection in cell-free massive mimo systems," *IEEE Access*, 2020, vol. 8, pp. 108 127–108 132.
- [9] L. Dai and B. Bai, "Optimal decomposition for large-scale infrastructure-based wireless networks," *IEEE Transactions on Wireless Communications*, 2017, vol. 16, no. 8, pp. 4956–4969.
- [10] J. Wang, L. Dai, *et al.*, "Clustered cell-free networking: A graph partitioning approach," *IEEE Transactions on Wireless Communications*, 2023, <https://arxiv.org/abs/2207.11641>.
- [11] F. Riera-Palou, G. Femenias, *et al.*, "Clustered cell-free massive mimo," in *Proc. IEEE GLOBECOM*, 2018.
- [12] Q. Le, V. Nguyen, *et al.*, "Learning-assisted user clustering in cell-free massive mimo-noma networks," *IEEE Transactions on Vehicular Technology*, 2021, vol. 70, no. 12, pp. 12 872–12 887.
- [13] S. Buzzi and A. Zappone, "Downlink power control in user-centric and cell-free massive mimo wireless networks," in *Proc. IEEE PIMRC*, 2017.
- [14] R. Nikbakht and A. Lozano, "Uplink fractional power control for cell-free wireless networks," in *2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–5.
- [15] Y. Zhao, I. Niemegeers, *et al.*, "Power allocation in cell-free massive mimo: A deep learning method," *IEEE Access*, 2020, vol. 8, pp. 87 185–87 200.
- [16] M. Mahajan, P. Nimbhorkar, *et al.*, "The planar k-means problem is np-hard," *Theoretical Computer Science*, 2012, vol. 442, pp. 13–21.
- [17] T. Vu, S. Chatzinotas, *et al.*, "Joint power allocation and access point selection for cell-free massive mimo," in *ICC 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.
- [18] C. Hao, T. Vu, *et al.*, "Joint user association and power control for cell-free massive mimo," *IEEE Internet of Things Journal*, 2024.
- [19] P. Henderson, R. Islam, *et al.*, "Deep reinforcement learning that matters," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, vol. 32, no. 1.
- [20] R. Chang, S. Han, *et al.*, "Reinforcement learning-based joint cooperation clustering and content caching in cell-free massive mimo networks," in *Proc. IEEE VTC-Fall*, 2021.
- [21] N. N. Krishnan, E. Torkildson, *et al.*, "Optimizing throughput performance in distributed mimo wi-fi networks using deep reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, 2020, vol. 6, no. 1, pp. 135–150.
- [22] B. Banerjee, R. C. Elliott, *et al.*, "Access point clustering in cell-free massive mimo using conventional and federated multi-agent reinforcement learning," *IEEE Transactions on Machine Learning in Communications and Networking*, 2023.
- [23] L. Luo, J. Zhang, *et al.*, "Downlink power control for cell-free massive mimo with deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, 2022, vol. 71, no. 6, pp. 6772–6777.
- [24] D. Deng, C. Wang, *et al.*, "Joint flexible duplexing and power allocation with deep reinforcement learning in cell-free massive mimo system," *China Communications*, 2023, vol. 20, no. 4, pp. 73–85.
- [25] M. Du, X. Sun, *et al.*, "Joint cooperation clustering and downlink power control for cell-free massive mimo with deep reinforcement learning," in *2023 IEEE 23rd International Conference on Communication Technology (ICCT)*, 2023, pp. 618–623.
- [26] J. C. Bezdek and R. J. Hathaway, "Some notes on alternating optimization," in *Advances in Soft Computing—AFSS 2002: 2002 AFSS International Conference on Fuzzy Systems Calcutta, India, February 3–6, 2002 Proceedings*, 2002, pp. 288–300.
- [27] S. Pateria, B. Subagdja, *et al.*, "Hierarchical reinforcement learning: A comprehensive survey," *ACM Computing Surveys (CSUR)*, 2021, vol. 54, no. 5, pp. 1–35.
- [28] T. Ren, J. Niu, *et al.*, "Enabling efficient scheduling in large-scale uav-assisted mobile-edge computing via hierarchical reinforcement learning," *IEEE Internet of Things Journal*, 2021, vol. 9, no. 10, pp. 7095–7109.
- [29] Y. Geng, E. Liu, *et al.*, "Hierarchical reinforcement learning for relay selection and power optimization in two-hop cooperative relay network," *IEEE Transactions on Communications*, 2021, vol. 70, no. 1, pp. 171–184.
- [30] F. Tan, Q. Deng, *et al.*, "Energy-efficient access point clustering and power allocation in cell-free massive mimo networks: A hierarchical deep reinforcement learning ap-

Biographies



Du Mingjun received the B.E. degree in communication engineering from the School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen, China, in 2022, where he is currently pursuing the M.E. degree in electronic information engineering with the School of Electronics and Communication Engineering. His research interests include MIMO, reinforcement learning and combination of Wi-Fi and large language models.



Sun Xinghua received the B.S. degree from the Nanjing University of Posts and Telecommunications (NJUPT), China, in 2008, and the Ph.D degree from the City University of Hong Kong (CityU), Hong Kong, China, in 2013. Since 2018, he has been an Associate Professor with Sun Yat-sen University, China. He served as the Technical Program Committee Member and the Organizing Committee Member for numerous conferences. His research interests include stochastic modeling of wireless networks and machine learning for networking.



Zhang Yue received his B.S. and M.S. degrees from South China University of Technology, Guangzhou, China, in 2012 and 2015, respectively, and the Ph.D degree from City University of Hong Kong, Hong Kong, China, in 2021. He is currently an assistant professor at the Department of Electronic and Information Engineering, Shantou University. His research interests include resource allocation and access point placement optimization of wireless communication networks.



Wang Junyuan received the B.S. degree in Communications Engineering from Xidian University, Xi'an, China, in 2010, and the Ph.D degree in Electronic Engineering from City University of Hong Kong, Hong Kong, China, in 2015. She is currently a Research Professor with the College of Electronic and Information Engineering and the Institute for Advanced Study, Tongji University, Shanghai, China. Her research mainly focuses on wireless communications and networking, and artificial intelligence.



Liu Pei received the B.E. degree of Telecommunication Engineering from Huazhong University of Science and Technology (HUST), China, in 2014, and the Ph.D degree of Information and Communication Engineering from HUST, China, in 2019. He joined the School of Information Engineering, Wuhan University of Technology, in 2019, where he is currently a Special Professor and a Master's Supervisor. He performs research on Massive MIMO, Low-Cost ADC, Random Access, and Physical Layer Security.