Tan *et al.*
EURASIP Journal on Advances in Signal Processing (2024) 2024:18
https://doi.org/10.1186/s13634-024-01111-9

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Energy-efficient access point clustering and power allocation in cell-free massive MIMO networks: a hierarchical deep reinforcement learning approach

Fangqing Tan[1*], Quanxuan Deng[1] and Qiang Liu[2]

*Correspondence:
tfqing@guet.edu.cn

[1] Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing, Guilin University of Electronic Technology, Guilin 541004, China
[2] College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, China

Abstract

Cell-free massive multiple-input multiple-output (CF-mMIMO) has attracted considerable attention due to its potential for delivering high data rates and energy efficiency (EE). In this paper, we investigate the resource allocation of downlink in CF-mMIMO systems. A hierarchical depth deterministic strategy gradient (H-DDPG) framework is proposed to jointly optimize the access point (AP) clustering and power allocation. The framework uses two-layer control networks operating on different timescales to enhance EE of downlinks in CF-mMIMO systems by cooperatively optimizing AP clustering and power allocation. In this framework, the high-level processing of system-level problems, namely AP clustering, enhances the wireless network configuration by utilizing DDPG on the large timescale while meeting the minimum spectral efficiency (SE) constraints for each user. The low layer solves the link-level sub-problem, that is, power allocation, and reduces interference between APs and improves transmission performance by utilizing DDPG on a small timescale while meeting the maximum transmit power constraint of each AP. Two corresponding DDPG agents are trained separately, allowing them to learn from the environment and gradually improve their policies to maximize the system EE. Numerical results validate the effectiveness of the proposed algorithm in term of its convergence speed, SE, and EE.

Keywords: Cell-free massive MIMO, Access points clustering, Power allocation, Energy efficiency, hierarchical deep deterministic policy gradient

1 Introduction

With the rapid development in mobile communication technology, wireless communication systems are continuously moving toward higher data rates and energy efficiency (EE). In order to fulfill the high-rate, ultra-reliable, and low-latency communication requirements of future applications, massive multiple-input multiple-output (MIMO) systems has emerged as an effective paradigm to support the growing popularity of mobile applications, in which the base station implements hundreds of antennas to efficiently serve a large number of user equipments (UEs) [1]. In practical real-world

applications of massive MIMO systems, a substantial quantity of antennas can be deployed either centrally or in a distributed fashion. In the former scenario, all antennas are positioned within a confined space, eliminating the need for fronthaul. Leveraging channel hardening and favorable propagation characteristic, massive MIMO reduces the transmission energy consumption while mitigating interference among UEs, thereby supporting greater throughput and more connectivity [2–5]. In the latter scenario, on the other hand, distinct antennas are geometrically spaced apart but linked to a central processing unit (CPU) through a fronthaul network. Given the proximity of UEs to antennas, cell-free massive MIMO (CF-mMIMO) can reduce interference, enhance system capacity, and more effectively utilize radio resources, compared with the central massive MIMO. Furthermore, CF-mMIMO technology fully leverages macro-diversity and multi-UE interference suppression, offering a nearly consistent quality of service (QoS) to UEs. This, in turn, enhances system performance and extends coverage [6–8].

For CF-mMIMO systems, access point (AP) clustering and power allocation play an important role in enhancing the system performance. Because proper AP clustering and power allocation strategies can effectively reduce multi-UE interference, improve the system EE, and ensure UE's QoS [9]. In recent years, there have been many works focused on AP clustering and power allocation. For example, the work [10] introduced a joint optimization approach encompassing cell, channel, and power allocation to maximize the overall UE throughput. In [11], an iterative optimization was conducted for AP clustering, linear least mean square error precoding, and power allocation to maximize the sum-rate in CF-mMIMO systems. Additionally, the authors of the work [12] introduced distributed algorithms to address the joint problem of AP selection and power allocation in a multi-channel, multi-AP network. Furthermore, the authors of [13] devised a joint optimization approach for pilot allocation, AP selection, and power control, to enhance the throughput of CF-mMIMO systems. However, these algorithms' computational complexity increase exponentially with the increase of the number of AP and UE in the above-mentioned works, which results in real-time processing limitations that may be difficult to apply in CF-mMIMO systems. Therefore, it is urgent to develop a low-complexity and good-scalability method for large-scale networks.

On the other hand, to perform the AP clustering and power allocation, an alternative approach is to utilize the "learn to optimize" methodology. This approach leverages the capability of deep neural networks (DNNs) to acquire intricate patterns and approximate complex function mappings. Deep learning (DL) has excelled in perceptual capabilities but has certain limitations in decision making. In contrast, reinforcement learning (RL) possesses decision-making abilities but has constraints in perception. Therefore, the integration of DL and RL, known as deep reinforcement learning (DRL), aims to overcome each other's shortcomings [14]. It enables the direct learning of control strategies from high-dimensional raw data. This approach closely resembles human thinking and has shown immense potential in enhancing wireless communication performance. DRL operates based on rewards, allowing it to find solutions to convex and non-convex problems without the need for extensive training datasets. Additionally, the computational complexity required for generating DRL outputs is low, involving only a small number of simple operations. These characteristics make DRL a powerful technology poised to make significant advancements in the field of wireless communication.

There are many works on resource allocation of CF-mMIMO systems based on DRL [15–24]. In [15], a power allocation method based on deep Q-network (DQN) was examined, taking into account the presence of imperfect channel state information (CSI). The authors of [16] introduced a distributive dynamic power allocation scheme, employing model-free DRL, to maximize a utility function based on the weighted sum-rate. The work [17] investigated the joint pilot and data power control, as well as the design of receiving filter coefficients in cell-free systems. It also presented a decentralized solution utilizing the actor-critic (AC) approach. Moreover, in [18], feed-forward neural networks were integrated into the output layer of the deep deterministic policy gradient (DDPG) algorithm. This integration aimed to mitigate the issue of over-fitting in DRL. In [19], a power control algorithm, utilizing the twin delayed deep deterministic policy gradient (TD3), was introduced. Furthermore, to concurrently optimize compute and radio resources, [20] tackled a multi-objective problem through the application of a distributed RL-based method and a heuristic iterative algorithm. In [21], the authors introduced a framework for the joint channels clustering and power allocating using a multi-agent DRL approach, known as double DQN (DDQN). In [22], the research focused on optimizing relay clustering and power level allocation in device-to-device transmissions. To address this challenge, a centralized hierarchical DRL (HDRL) approach was introduced with the aim of discovering the most effective solution to the problem. [23] introduced a multi-agent RL (MARL) algorithm to tackle complex signal processing problems in high-dimensional scenarios. This approach incorporates predictive management and distributed optimization, all the while taking into account a dual-layer power control architecture that leverages large-scale fading coefficients between antennas to reduce interference. The work of [24] introduced HDRL to achieve automatic unbalanced learning. At a higher level, decisions regarding the quantity of synthetic samples to generate are made, while at a lower level, the determination of sample locations is influenced by the high-level decision, considering that the optimal locations may vary depending on the sample quantity. [25] explored DRL within a hierarchical framework and presented the hierarchical DQN (H-DQN) model. This approach breaks down the primary problem into autonomous sub-problems, with each one being handled by its dedicated RL agent. [26] demonstrated tasks that H-DQN cannot solve, highlighted limitations in such hierarchical frameworks, and described the recursive hierarchical framework that summarizes an architecture using recursive neural networks at the meta-level.

In a word, many research works demonstrated the excellent performance of DRL-based methods in resource allocation in CF-mMIMO systems. However, while DRL is excellent at solving the optimization problems, its limitations become apparent when dealing with some problems with hierarchy and complexity. This provides an opportunity for the introduction of HDRL. HDRL uses a hierarchical framework that breaks down problems into independent sub-problems and is handled by a dedicated DRL agent, allowing for better challenges when dealing with complex problems. Especially when the joint optimization problem of AP clustering and power allocation is conducted, HDRL can efficient handle multiple levels of information and decision making. This is mainly because HDRL can flexibly deal with hierarchical relationships, the original optimization problem is divided into more manageable sub-problems, so as to better adapt to the complexity and uncertainty in the actual scenario. Therefore, although DRL

is excellent in solving most of optimization problems for some specific joint optimization problems, HDRL's hierarchical structure and flexibility can provide a more efficient solution. In this paper, how to make better use of the advantages of DRL and HDRL, and how to choose appropriate methods, is an important direction of resource allocation in CF-mMIMO systems.

More recently, it is important to emphasize that the AP-UE association network structure is tailored to the channel statistics to harness array gain, while power allocation is customized based on the instantaneous effective CSI to attain spatial multiplexing gain. Furthermore, AP clustering constitutes a global decision for the entire system, while power allocation involves local decisions for each specific link or connection after connection establishment. Therefore, two sub-problems need to be addressed at different timescales and levels since they have distinct scopes of influence and optimization objectives. Fortunately, the HDRL algorithm can decompose this joint optimization problem into two sub-problems, allowing optimization at different levels. Moreover, for CF-mMIMO systems, the total number of APs and UEs is very large and thus the joint optimization of AP clustering and power allocation would induce prohibitively high complexity. However, in practical applications, especially for mobile communication systems, the joint optimization needs to be completed in real-time. This requires the algorithm to have fast convergence speed and low latency to adapt to the network dynamics and user mobility.

To address the above challenges, we propose a hierarchical depth deterministic strategy gradient (H-DDPG) framework for joint optimization of AP clustering and power allocation in CF-mMIMO systems using the idea of "divide and conquer." The framework employs a two-layer control network operating on different timescales to improve the system EE performance. The main contributions of this paper can be summarized as follows.

- In this paper, we proposed a joint optimization framework for resource allocation of CF-mMIMO systems, taking into account AP clustering and power allocation. Unlike traditional methods, this integrated approach helps to coordinate system resources, reduce multi-user interference and power consumption, while meeting the constraints of the minimum spectral efficiency (SE) per UE and the transmitted power budget per AP, and thus achieve a better EE performance.
- To solve the formulated joint optimization problem, we propose a H-DDPG algorithm. The algorithm combines the principles of DRL and hierarchical optimization, and creates a collaborative optimization framework that combines AP clustering and power allocation. By running two-layer control networks on different time scales, the decomposition and processing of system-level and link-level problems are realized, with the goal of maximizing the EE of the system. For system-level problems, namely AP clustering, the implement of DDPG on a large timescale enhances the configuration of wireless networks and helps to improve the overall performance of the network. For link-level problem, i.e., power allocation, by utilizing DDPG on a small timescale, interference between APs is successfully reduced and the transmission performance is improved.

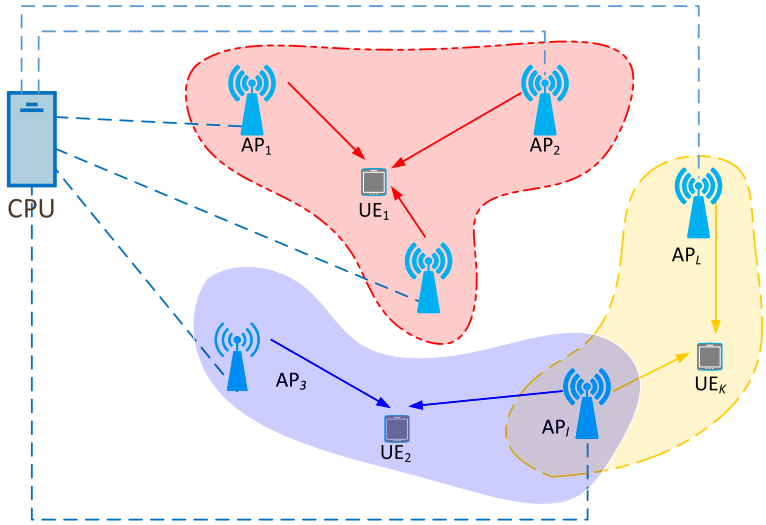

Fig. 1 CF-mMIMO system model

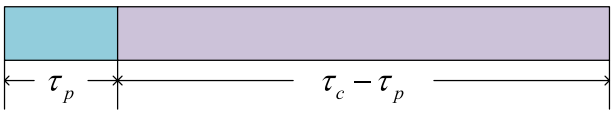

Fig. 2 Structure of resource block of length τ_c time instants

- The efficiency of the proposed approach is verified through numerical simulations. The simulation results demonstrate that compared to single-layer DDPG approaches, the proposed two-tier H-DDPG method, exhibits substantial enhancement in terms of the convergence speed, EE, and SE performance.

The following sections of this paper are organized as follows: Section 2 provides a detailed description of the system model. In Section 3, the power consumption model is elaborated. Section 4 is dedicated to formulating the optimization problem. Section 5 expounds on the principles and design of the H-DDPG algorithm. Section 6 discusses the simulation results. Finally, Section 7 offers the paper's conclusion.

2 System model

As illustrated in Fig. 1, we consider a CF-mMIMO system, comprising K arbitrarily located single-antenna UEs and L geographically distributed APs, with each AP being equipped with N antennas. In the context of implementing the UE-centric cell-free architecture, where each UE is catered to by a specific subset of APs. All APs are connected to the CPU via fronthaul links.

In this paper, time division duplex (TDD) protocol and a block fading model are adopt, where time–frequency resources are divided into coherent blocks so that the channel coefficients in each block can be assumed to be fixed. The coherent block consists of τ_c symbols, as shown in Fig. 2, and the channels are independently and randomly distributed in each coherent block. In the TDD-based protocol, each coherent interval τ_c is

divided into three stages: (i) the uplink channel estimation phase, (ii) the downlink data transmission phase, and (iii) the uplink data transmission phase. In this paper, we only focus on resource allocation to maximize the system EE of the downlink in CF-mMIMO systems. The proposed algorithms in this work can be readily applied to the uplink scenario, after some straightforward simplification. Assuming the duration of uplink training phase is τ_{p}, the remaining coherence interval $(\tau_{\mathrm{c}} - \tau_{\mathrm{p}})$ is used for the downlink data transmission.

In the context of spatially correlated Rayleigh fading, the channel linking the k-th UE and the l-th AP is denoted as $\boldsymbol{h}_{kl} \in \mathbb{C}^N$and is characterized as $\boldsymbol{h}_{kl} \sim \mathcal{N}_{\mathbb{C}}(0, \boldsymbol{R}_{kl})$ with $\boldsymbol{R}_{kl} \in \mathbb{C}^{N\times N}$ signifying the spatial correlation matrix. Conversely, when transmitting from the l-th AP to the k-th UE, the average channel gain for any specific antenna is determined through the normalized trace $\beta_{kl} = {}^{1}\!/_{N}\mathrm{tr}(\boldsymbol{R}_{kl})$

2.1 Channel estimation

In the channel estimation phase, each AP independently obtains the CSI through local estimation, using the uplink pilot transmissions from UEs. Consider a collection of mutually orthogonal τ_{p} pilot sequences allocated to UEs, where t_k denotes the index of the pilot assigned to the k-th UE, and S_{t_k} represents the group of UEs sharing pilot t_k. When a UE from set S_{t_k} transmits pilot t_k, the received signal $\boldsymbol{y}^p_{t_k l} \in \mathbb{C}^N$ at the l-th AP is determined by computing the inner product between the received signal and pilot sequence t_k as follows:

$$\boldsymbol{y}^p_{t_k l} = \sum_{i\in S_{t_k}} \sqrt{\tau_p p_p}\boldsymbol{h}_{il} + \boldsymbol{n}_{t_k l}, \tag{1}$$

where p_p represents the pilot transmitting power per UE, and $\boldsymbol{n}_{t_k l} \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma^2 \boldsymbol{I}_N\right)$ denotes the additive Gaussian noise vector at the l-th AP. By utilizing the MMSE estimator at the l-th AP, the channel estimation between the k-th UE and the l-th AP can be expressed as

$$\hat{\boldsymbol{h}}_{kl} = \sqrt{\tau_p p_p}\boldsymbol{R}_{kl}\boldsymbol{\Psi}^{-\mathbf{1}}_{t_k l}\boldsymbol{y}^p_{t_k l} \sim \mathcal{N}_{\mathbb{C}}\Big(\mathbf{0}, \tau_p p_p \boldsymbol{R}_{kl}\boldsymbol{\Psi}^{-\mathbf{1}}_{t_k l}\boldsymbol{R}_{kl}\Big), \tag{2}$$

where $\boldsymbol{\Psi}_{t_k l} = \mathbb{E}\left\{\boldsymbol{y}^p_{t_k l}\left(\boldsymbol{y}^p_{t_k l}\right)^H\right\} = \sum_{i\in S_{t_k}} \tau_p p_p \boldsymbol{R}_{il} + \sigma^2 \boldsymbol{I}_N$ represents the received pilot signal's correlation matrix.

2.2 Data transmission

Let $\varsigma_i \in \mathbb{C}$ symbolize the downlink data signal with unit power for the i-th UE, where $E\left\{|\varsigma_i|^2\right\} = 1$ is independent of each other. For the l-th AP, the CPU encodes the associated data symbol and transmits it to the l-th AP over fronthaul links. Then, the transmitted signal at the l-th AP is given as

$$\boldsymbol{x}_l = \sum_{i=1}^{K} \sqrt{\rho_{il}}\boldsymbol{w}_{il}\varsigma_i, \tag{3}$$

where $\boldsymbol{w}_{il} \in \mathbb{C}^N$ represents the normalized precoding vector at the l-th AP for the i-th UE, i.e., $\|\boldsymbol{w}_{il}\|^2 = 1$, and $\rho_{il} \geq 0$ signifies the power allocated to the i-th UE by the l-th AP.

Suppose that each UE is served by all APs over the same time–frequency resources. Accordingly, the received signal at the k-th UE can be expressed as

$$\begin{aligned} y_k^{\mathrm{dl}} &= \sum_{l=1}^{L} \boldsymbol{h}_{kl}^{\mathrm{H}} \boldsymbol{x}_l + n_k \\ &= \sum_{l=1}^{L} \boldsymbol{h}_{kl}^{\mathrm{H}} \sqrt{\rho_{kl}} \boldsymbol{w}_{kl} \varsigma_k + \sum_{\substack{i=1 \\ i \neq k}}^{K} \left(\sum_{l=1}^{L} \boldsymbol{h}_{kl}^{\mathrm{H}} \sqrt{\rho_{kl}} \boldsymbol{w}_{il} \varsigma_i \right) + n_k, \end{aligned} \tag{4}$$

where $n_k \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma^2\right)$ represents the noise at the k-th UE.

As stated in [27], the achievable SE of the k-th UE is defined as

$$SE_k = \frac{\tau_d}{\tau_c} \log_2 \left(1 + SINR_k\right), \tag{5}$$

where

$$SINR_k = \frac{\left| \sum_{l=1}^{L} E\left\{ \mathbf{h}_{kl}^H \rho_{kl} \mathbf{w}_{kl} \right\} \right|^2}{\sum_{i=1}^{K} E\left\{ \left| \sum_{l=1}^{L} \mathbf{h}_{kl}^H \rho_{il} \mathbf{w}_{il} \right|^2 \right\} - \left| \sum_{l=1}^{L} E\left\{ \mathbf{h}_{kl}^H \rho_{kl} \mathbf{w}_{kl} \right\} \right|^2 + \sigma^2}, \tag{6}$$

This SE expression is applicable to any choice of the transmission precoding vector $\boldsymbol{w}_k$. In fact, it also applies to any channel allocation, not just the associated Rayleigh fading. The expression can be calculated using Monte Carlo methods, which involves approximating each expectation with a substantial number of randomly generated sample averages. More precisely, we can generate an implementation of the channel estimate in a large set of coherent blocks. The normalized precoded vector is denoted as $\boldsymbol{w}_{il} = \overline{\boldsymbol{w}}_{il} / \|\overline{\boldsymbol{w}}_{il}\|$. This paper adopts the linear precoding minimum mean square error precoding scheme.

3 Power consumption model and energy efficiency

This section presents the definition of a general power consumption model. The main elements of the power consumption model are given as follows: (1) the power consumption at the radio site, encompassing power utilization by UEs $\left\{P_k^{ue} : \forall k\right\}$APs $\left\{P_l^{ap} : \forall l\right\}$, and fronthaul links $\left\{P_l^{fh} : \forall l\right\}$(2) the CPU power consumption P_{cpu} [28]. The total power consumption is represented as

$$P_{total} = \sum_{k=1}^{K} P_k^{ue} + \sum_{l=1}^{L} P_l^{ap} + \sum_{l=1}^{L} P_l^{fh} + P_{cpu}, \tag{7}$$

The power consumption at the k-th UE is given as

$$P_k^{ue} = P_k^{c,ue} + \frac{\tau_p p_p}{\tau_c \eta_{ue}}, \tag{8}$$

where $P_k^{c,ue}$ is the power consumption of the internal circuitry, while the second term accounts for the power usage during uplink transmission, and $0 < \eta_{ue} \leq 1$ denotes the power amplifier efficiency at the UE. $\frac{\tau_p}{\tau_c}$ is represented as the fractions of the uplink pilot transmission, the power consumption at the l-th AP is

$$P_l^{ap} = NP_l^{c,ap} + N|D_l| \cdot P_l^{pro} + \frac{\tau_d}{\tau_c \eta_{ap}} \sum_{k \in D_l} \rho_{kl}, \tag{9}$$

wherein the internal circuit power of each AP antenna is represented by $P_l^{c,ap}$ $D_l \subset \{1, \ldots, K\}$ is the subset of UEs served by the l-th AP, P_l^{pro} as the power consumed to process the received/transmitted signals of each UE in D_l, and $0 < \eta_{ap} \leq 1$ as the power amplifier efficiency at the AP. For each fronthaul link, the power consumption is

$$P_l^{fh} = P_l^{fix} + \frac{\tau_u + \tau_d}{\tau_c} |D_l| \cdot P_l^{sig}, \tag{10}$$

where P_l^{fix} represents the fixed power consumption, and the remainder describes the load-related uplink and downlink signaling, with P_l^{sig} representing the signaling power for each UE. The CPU is charge of processing all of the UE's signals and has power consumption

$$P_{cpu} = P_{cpu}^{fix} + B \sum_{k=1}^{K} SE_k \cdot P_{cpu}^{cod}, \tag{11}$$

where P_{cpu}^{fix} represents the fixed power consumption, B stands for the system bandwidth, and P_{cpu}^{cod} denotes the energy consumption per bit for the initial encoding at the CPU.

As per the established power consumption model, the overall system EE in the considered CF-mMIMO system (in bits/joules) is

$$EE = B \cdot \frac{\sum_{k=1}^{K} SE_k}{P_{total}}, \tag{12}$$

4 Problem formulation

The focus of this paper is on maximizing the overall system EE in the considered CF-mMIMO system through a joint optimization of AP clustering and power allocation, while adhering to constraints associated with the SE requirements of individual UEs and the transmit power budgets of each AP [29]. The problem is formulated as

$$\begin{aligned} &\underset{(\rho_{kl}:\forall k,l)}{\text{Maximize}} \; EE \\ &\text{s.t.} \qquad \text{C1} : SE_k \geq SE_k^{\min}, \; \forall k \\ &\qquad\quad \text{C2:} \sum_{k=1}^{K} \rho_{kl} \leq P_{\max}, \; \forall l \end{aligned} \tag{13}$$

where $SE_k^{\min}$ represents the minimum spectral efficiency required for the k-th UE, and $P_{\max}$ represents the maximum transmit power of each AP.

It is evident that the optimization problem in (13) is non-convex, making it challenging to obtain a global optimal solution that adheres to the non-convex constraint C1 and C2. Furthermore, in CF-mMIMO systems, the number of APs and UEs is very large, leading to a high complexity when we jointly optimize AP clustering and power allocation. To address these challenges, in the following section, the HDRL algorithm is developed to obtain a optimal solution.

5 Joint optimization of AP clustering and power allocation based on H-DDPG algorithm

This section introduces a two-layer DRL framework for addressing the optimization problem (13) by employing the HDRL framework. This framework incorporates both high-level and low-level control strategies to achieve the network optimization through hierarchical coordination.

5.1 HDRL framework

Since the joint optimization of AP clustering and power allocation usually involves a large number of nonlinear constraints. Traditional optimization methods cannot effectively solve nonlinear problems, and non-convex optimization algorithms result in cubic-order complexity, which is impractical in CF-mMIMO systems. The DRL algorithm has some advantages in dealing with complex nonlinear problems. Moreover, HDRL is a DRL framework composed of two distinct levels of DRL. This hierarchical approach aims to address complex decision tasks by partitioning them into high-level and low-level control layers. The advantages of HDRL compared to DRL are as follows.

1. HDRL uses a hierarchy to break down complex tasks into smaller sub-tasks with their own policies and rewards, improving efficiency in solving complex problems.
2. HDRL promotes knowledge sharing between different levels, allowing low-level strategies to benefit from higher-level guidance, accelerating effective strategy learning.
3. HDRL balances exploration and utilization by involving different levels of strategy. High-level policies focus on global decision making, while low-level policies provide detailed exploration and adjustment, enhancing environmental exploration and strategic refinement.
4. Because HDRL can learn strategies more efficiently, especially in the case of reasonably divided hierarchies, it usually has faster convergence. This means it can find a better strategy in a relatively short period of time.

Therefore, this paper proposes a distributed two-layer DRL framework, as shown in Fig. 3, in which the high-level control strategy and the low-level control strategy work successively in a distributed manner. In each layer of HDRL, agents learn their dynamic environment through repeated sequences of observations, actions, and rewards. In slot t, by observing state s_t, the agent takes action $a_t \in A$ based on some strategy π, and then receives a reward r_t from the environment and advances to the following state s_{t+1}.

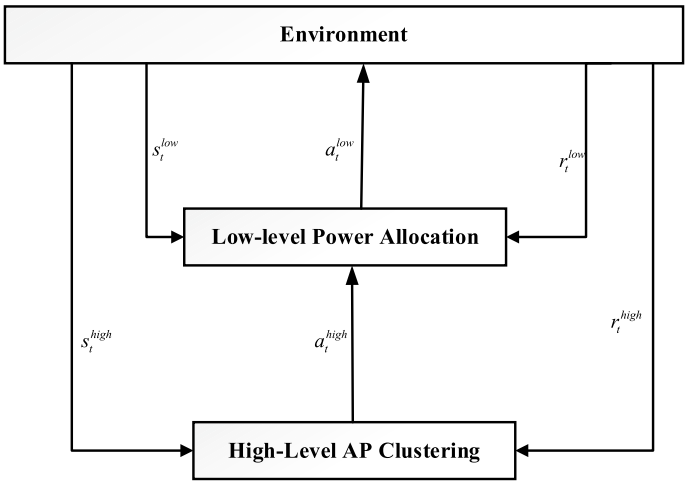

Fig. 3 HDRL algorithm framework

The quadruple (s_t, a_t, r_t, s_{t+1})represents a single interaction with the environment [30]. Define e_t as an empirical sequence, where $e_t = (s_t, a_t, r_t, s_{t+1})$. To maximize cumulative rewards, the agent seeks an optimal strategy.

The two primary methods for determining the optimal strategy are the value-based approach, DQN, and the policy-based approach, DDPG, respectively. Significant spatial capacity is necessary to accommodate the potentially substantial quantity of UEs and APs within the CF-mMIMO system, and continuous motion space due to power allocation issues. Traditional RL methods, such as Q-learning and DQN, are commonly used to deal with discrete action spaces where agents choose one of a limited set of discrete actions. DDPG and its enhanced iterations are well-suited for addressing challenges in scenarios with continuous action spaces. It combines deterministic strategy gradient method and DNN, which can effectively learn and optimize continuous action strategies. Hence, the DDPG algorithm can obtain rapid and consistent learning rate.

The DDPG algorithm consists of four networks: the actor network, the critic network, and their respective target networks, each with specific relationships between them. The role of the actor component is to produce an action for each time step using a deterministic strategy denoted as $\mu(s_t|\theta^\mu)$, learned by a DNN with weight . Update the weight θ^μ to find the best deterministic strategy $\mu(s_t|\theta^\mu)$ using the action value function, with the anticipated long-term reward defined as:

$$Q(s_t, a_t) = E[R_t|s_t, a_t], \tag{14}$$

Here, R_t represents the cumulative discounted future reward. In general, θ^μ is updated by gradient ascent:

$$\theta^\mu \leftarrow \theta^\mu + \alpha^\mu \nabla_\theta J(\theta)|_{\theta=\theta^\mu}, \tag{15}$$

Here, α^μ represents the learning rate, and $J(\theta)$ stands for

$$J(\theta) = E[Q(s_t, \mu(s_t|\theta))], \tag{16}$$

$J(\theta)$ represents the anticipated target value of s_t. It is worth noting that the action value function in Eq. (14) permits recursive relationships

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma E[Q(s_{t+1}, a_{t+1})], \tag{17}$$

This is what is required to determine $Q(s_t, \mu(s_t|\theta^\mu))$ given by (15) in the critic section. More specifically, the critic uses a separate DNN with weight θ^Q to evaluate the action value function $Q\big(s_t, \mu(s_t|\theta^\mu)|\theta^Q\big)$. Typically, weight θ^Q is updated using the following method:

$$\theta^Q \leftarrow \theta^Q - \alpha^Q \nabla_\theta L(\theta)|_{\theta=\theta^Q}, \tag{18}$$

Here, α^Q represents the learning rate, and $L(\theta)$ stands for

$$L(\theta) = E\Big[\big(Q(s_t, a_t|\theta) - y_t\big)^2\Big], \tag{19}$$

where $L(\theta)$ represents the mean squared Bellman error function of the target.

The target actor network is essentially a duplicate of the actor network, but it undergoes gradual parameter updates aimed at stabilizing the training process. The primary role of the target actor network is to supply target actions, thereby minimizing the training error of the actor network. The target critic network is a duplicate of the critic network, and it also undergoes gradual updates. Its function is to provide target value function to reduce the training error of critic network. The target critic network's output is utilized to compute the critic network's error, facilitating the back-propagation of the error signal and, consequently, updating the critic network. In the course of training, the parameters of the target actor network and target critic network are methodically aligned with those of the corresponding actor network and critic network at specific intervals. This process aids in stabilizing the training, minimizing fluctuations, and enhancing the algorithm's convergence. The structure of the DDPG algorithm is illustrated in Fig. 4.

5.2 Action dichotomy scheme based on DDPG

The advantage of splitting DDPG actions into AP clustering and power allocation actions is that it allows the agent to consider these two critical factors simultaneously in a single time step and optimize system performance in a more integrated manner. This design helps to reduce communication latency, increase network capacity, reduce interference levels, and provide a better user experience [31].

1. State s_t

obtained from the low-level AP clustering,The state space comprises the AP cluster AP_{t-1} selected by the AP in the previous time slot, the channel's large-scale fading coefficient β_{lk}, the transmit power p_{t-1} from the previous slot, and the UE's SINR. Defined as follows is the state of the t-th time slot:

$$s_t = \{AP_{t-1}, \beta_{lk}, p_{t-1}, SINR\}, \tag{20}$$

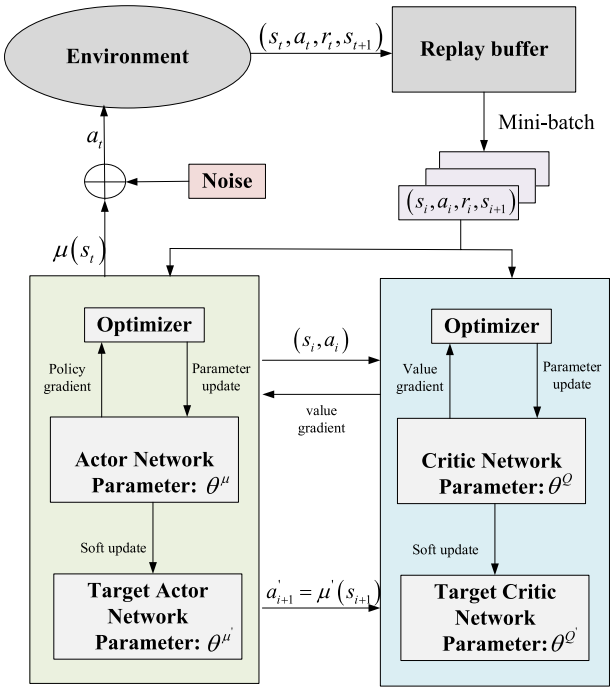

Fig. 4 Structure of the DDPG algorithm

2. Action a_t.

The agent's action is divided into AP clustering and power allocation; thus, the action of the t-th time slot is defined as follows:

$$\begin{aligned} a_t &= \left\{ a_t^{AP}, a_t^{PA} \right\} \\ a_t^{AP} &= \{\alpha_{11}, \alpha_{12}, \alpha_{22}, \ldots, \alpha_{LK}\}, \\ a_t^{PA} &= \{p_{11}, p_{12}, p_{22}, \ldots, p_{LK}\} \end{aligned} \tag{21}$$

3. Reward $r(s_t, a_t)$.

Defining (22) is the reward function of the t-th time slot according to the goal in the problem.

$$r(s_t, a_t) = B \cdot \frac{\sum_{k=1}^{K} SE_k}{P_{total}} - \sum_{k=1}^{K} u\left(SE_k^{\min} - SE_k \right), \tag{22}$$

where $u(x)$ is a step function, with a value of zero for $x \leq 0$ and one otherwise. The steps of the DDPG algorithm closely resemble those of the H-DDPG algorithm presented below, and will not be introduced here.

5.3 H-DDPG-based method

Considering the complexity of AP clustering and power allocation, a HDRL model with DDPG is proposed to reduce the search space. The layered model consists of two DRL agents and is optimized with power allocation as the meta-controller. The power allocation agent $PA\left(s_t^{PA}, a_t^{PA}, r_t^{PA}\right)$ is responsible for assigning transmission power to the UE with the objective of optimizing the signal quality, such as the signal-to-noise ratio, at the UE terminal, and enhancing the overall system throughput, all while minimizing interference. The AP clusters the agent $AP\left(s_t^{AP}, a_t^{AP}, r_t^{AP}\right)$ to select the service AP for all UEs and makes it meet the constraint of minimum SE. Typically, the AP clustering agent depends on the power allocation agent for transmitting packets, and the reward for the AP clustering agent actions depends on the actions of the power allocation agent. Moreover, the input s_t^{PA} for the power allocation agent depends on the actions of the AP clustering agent and the environmental state.

Specifically, two DDPG agents are designed, corresponding to AP clustering and power allocation, respectively. In the first layer DDPG, a state space is established, which includes CSI of different base stations, UE equipment location and channel quality. Based on this state information, a strategy network is trained to cluster the appropriate set of APs to maximize the data transfer rate. Further considered in the second layer of DDPG is the power allocation problem. Using the current network state and the AP set clustered in the first layer of DDPG, the second policy network is trained to optimize the power allocation for each AP, aiming to achieve maximum EE while considering power consumption. They evolve their strategies through constant interaction. Verified by a large number of numerical simulation experiments are the effectiveness and performance advantages of the proposed method. The definitions of state, action, and reward functions related to the H-DDPG algorithm are as follows.

- AP Clustering

Because AP clustering is a global decision for the entire system, and power allocation is a local decision for each specific link or connection after the connection is established. These two problems need to be solved at different timescales and levels because they have different spheres of influence and optimization goals. Therefore, AP is clustered as high-level DDPG, and its DDPG algorithm is designed as follows:

1. State s_t.

   The state space S is a collection of states observed by the agent within the environment. It includes the AP cluster AP_{t-1} from the previous time slot, the large-scale fading coefficient β_{lk} of the channel, and the SINR of the UE. Defined as follows is the state of the t-th time slot:

$$s_t = \{AP_{t-1}, \beta_{lk}, SINR\}, \tag{23}$$

2. Action a_t

The action space A is a collection of actions undertaken by the agent in response to each state. In the context of the AP clustering problem, an assignment variable $\alpha_{lk} \in \{0, 1\}$ is defined, where $k = 1, \ldots, K$, $l = 1, \ldots L$. When the k-th UE is served by the l-th AP, it is equal to 1; otherwise, $\alpha_{lk} = 0$. The joint clustering of all α_{lk} forms an action space composed of discrete points of 2^{kl}. The action for the t-th time slot is defined as follows:

$$a_t = \{\alpha_{11}, \alpha_{12}, \alpha_{22}, \ldots, \alpha_{LK}\}, \tag{24}$$

3) Reward $r(s_t, a_t)$.

According to the goal in the problem, the reward function of the t-th time slot is defined as:

$$r(s_t, a_t) = B \cdot \frac{\sum_{k=1}^{K} SE_k}{P_{total}} - \sum_{k=1}^{K} u\left(SE_k^{\min} - SE_k\right), \tag{25}$$

Through the step function, a uniform penalty is created for all UEs who do not achieve the minimum SE [32].

The steps of high-level DDPG algorithm are shown in Algorithm 1. Initially, the actor network and critic network, along with their respective target networks, are initialized. Simultaneously, the experience replay buffer and noise are also initialized. Using the actor network, an action is selected based on the current state, if a_t^H is greater than zero, then consider action $a_t^H = 1$, otherwise set $a_t^H = 0$. Execute the selected action in the environment, then record the reward and the next state. Storing the current state, action, reward, and the next state within the experience replay buffer. Randomly selecting a batch of experiences from the replay buffer for training the actor and critic networks. The target value is computed with the assistance of the target actor network and target critic network, aiming to minimize the error of the critic network. By applying the target actor network to the next state, the target value is estimated. To minimize the critic network error, the critic network error is calculated, and the gradient descent method is employed to update the critic network weights. The policy gradient method updates the weights of the actor network by maximizing the value function of the critic network associated with the actor network's output. Soft updates are employed to enhance training stability by gradually adjusting the parameters of the target actor and critic networks to align with those of the actor and critic networks. Repeat the training cycle until the predetermined number of training cycles is reached.

6 Power allocation

As the low level of the H-DDPG algorithm, power allocation is taken, and the DDPG algorithm is designed as follows.

1. State s_t

Consisting of the AP clusters AP_t obtained from the low-level AP clustering, the large-scale fading coefficient β_{lk} of the channel, the transmit power p_{t-1} of the.

Algorithm 1 The main step of the high-level DDPG algorithm

Initialize: High level network parameter
for episode = 1 to EP **do**
  for t = 1 to T_1 **do**
    Initialize the exploration noise $\mathcal{N}_t^H$ at time t
    Add noise to the action, $a_t^H = \mu\left(s_t^H \mid \theta_H^\mu\right) + \mathcal{N}_t^H$
    Apply tanh to each element of a_t^H, projected to a finite range [-1, 1]
    If a_t^H is greater than zero, consider action $a_t^H = 1\sqrt{a^2 + b^2}$, otherwise set $a_t^H = 0$
    Evaluate action a_t^H taken by the actor
    Obtain the reward r_t^H and next state s_{t+1}^H
    Store $b_t^H = \left(s_t^H, a_t^H, r_t^H, s_{t+1}^H\right)$ in replay buffer D_H
    Sample a mini-batch B from D_H randomly
    Evaluate target $y_t^H = r\left(s_t^H, a_t^H\right) + \gamma Q'\left(s_{t+1}^H, \mu'\left(s_{t+1}^H \mid \theta^{\mu'}\right) \mid \theta^{Q'}\right)$
    Determine the gradient:

$$\nabla_{\theta_H^\mu} J\left(\theta_H^\mu\right) = \nabla_{\theta_H^\mu} \frac{1}{|\mathrm{B}|} \sum_{b_i \in \mathrm{B}} Q_H\left(s_t^H, \mu\left(s_t^H \mid \theta_H^\mu\right) \mid \theta_H^Q\right)$$

$$\nabla_{\theta_H^Q} L\left(\theta_H^Q\right) = \nabla_{\theta_H^Q} \frac{1}{|\mathrm{B}|} \sum_{b_i \in \mathrm{B}} \left(Q_H\left(s_t^H, a_t^H \mid \theta_H^Q\right) - y_t^H\right)^2$$

    Update the weights in the evaluation network
    $\theta_H^\mu \leftarrow \theta_H^\mu + \alpha^\mu \nabla_{\theta_H^\mu} J\left(\theta_H^\mu\right), \theta_H^Q \leftarrow \theta_H^Q - \alpha^Q \nabla_{\theta_H^Q} L\left(\theta_H^Q\right)$
    Update the weights in the target network
    $\theta_H^{\mu'} \leftarrow \tau\theta_H^{\mu'} + (1-\tau)\theta_H^\mu, \theta_H^{Q'} \leftarrow \tau\theta_H^{Q'} + (1-\tau)\theta_H^Q$
  end
end

previous time slot, and the SINR of the UE, the state space is defined. Defined as follows is the state of the t-th time slot:

$$s_t = \{AP_t, \beta_{lk}, p_{t-1}, SINR\}, \tag{26}$$

2. Action a_t

The action of the agent is defined as the transmit power assigned by the AP to the UE. The action of the t-th time slot is defined as follows:

$$a_t = \{\rho_{11}, \rho_{12}, \rho_{22}, \ldots, \rho_{LK}\}, \tag{27}$$

3. Reward $r(s_t, a_t)$.

In accordance with the problem's objective, the reward function for the t-th time.

Algorithm 2 The main steps of low-level DDPG algorithm

Initialize: Low-level network parameters
for episode = 1 to EP **do**
  if t can be divided by ΔT **or (** $T_1 < t < T_2$ **) do**
    Initialize the exploration noise $\mathcal{N}_t^L$ at time t
    Add noise to the action, $a_t^L = \mu\left(s_t^L \mid \theta_L^\mu\right) + \mathcal{N}_t^L$
    Apply sigmoid to each element of a_t^L, projected to a finite range [0,1].
    a_t^L is processed to satisfy the maximum transmit power constraint for each AP.
    Evaluate action a_t^L taken by the low-level actor and get a_t^H based on the high-level actor
    Obtain the reward r_t^L and next state s_{t+1}^L
    Store $b_t^L = \left(s_t^L, a_t^L, r_t^L, s_{t+1}^L\right)$ in replay buffer D_L
    Sample a mini-batch B from D_L randomly
    Evaluate target $y_t^L = r\left(s_t^L, a_t^L\right) + \gamma Q'\left(s_{t+1}^L, \mu'\left(s_{t+1}^L \mid \theta^{\mu'}\right) \mid \theta^{Q'}\right)$
    Determine the gradient:

$$\nabla_{\theta_L^\mu} J\left(\theta_L^\mu\right) = \nabla_{\theta_L^\mu} \frac{1}{|\mathrm{B}|} \sum_{b_j \in \mathrm{B}} Q_L\left(s_t^L, \mu\left(s_t^L \mid \theta_L^\mu\right) \mid \theta_L^Q\right)$$

$$\nabla_{\theta_L^Q} L\left(\theta_L^Q\right) = \nabla_{\theta_L^Q} \frac{1}{|\mathrm{B}|} \sum_{b_j \in \mathrm{B}} \left(Q_L\left(s_t^L, a_t^L \mid \theta_L^Q\right) - y_t^L\right)^2$$

    Update the weights in the evaluation network

$$\theta_L^\mu \leftarrow \theta_L^\mu + \alpha^\mu \nabla_{\theta_L^\mu} J\left(\theta_L^\mu\right), \theta_L^Q \leftarrow \theta_L^Q - \alpha^Q \nabla_{\theta_L^Q} L\left(\theta_L^Q\right)$$

    Update the weights in the target network

$$\theta_L^{\mu'} \leftarrow \tau\theta_L^{\mu'} + (1-\tau)\theta_L^\mu, \theta_L^{Q'} \leftarrow \tau\theta_L^{Q'} + (1-\tau)\theta_L^Q$$

  end
end

slot is defined as follows:

$$r(s_t, a_t) = B \cdot \frac{\sum_{k=1}^{K} SE_k}{P_{total}}, \tag{28}$$

It is important to emphasize that although the high-level and low-level rewards are expressed differently, they both constitute integral components of their respective global rewards. Furthermore, the constraints on high-level and low-level strategies are identical,

ensuring a monotonically improving EE throughout the system. The procedure of low-level DDPG algorithm is shown in **Algorithm 2**.

The most significant distinction between Algorithm 2 and Algorithm 1 lies in the.

Algorithm 3 he main steps of H-DDPG algorithm

Initialize: High-level and low-level network parameters;
for episode = 1 to EP **do**
  for t = 1 to T_1 **do**
    Algorithm 1: The main step of the high-level DDPG algorithm
  if t can be divided by ΔT **or** ($T_1 < t < T_2$) **do**
    Algorithm 2: The main steps of low-level DDPG algorithm
  end
end

action design. Algorithm 2 handles action a_t^L to adhere to the maximum transmit power constraint of each AP. The idea of H-DDPG is to update the high-level and low-level networks sequentially, where the agents continuously adjust the high-level and low-level decisions to improve the transmission performance of the whole system. In particular, the high-level network policy is adjusted using a stable low-level policy to reduce inter-AP interference, whereas the low-level policy is updated with a more convergent high-level network policy to enhance system performance. The two networks continually update each other to attain their respective optimal strategies [33].

In this paper, we focus on the convergence and cumulative returns of the proposed framework H-DDPG in different training periods. During each training session, the two agents utilize the policy with the maximum reward to select the AP and assign power to each UE in the network, and then update their policies at the end of each training session. The complete H-DDPG algorithm steps are summarized in **Algorithm 3**.

6.1 Computational complexity

H-DDPG consists of two layers of DDPG, so its complexity is the sum of the superposition of two layers. In the DDPG layer for AP clustering, the input dimension is $L(2K-1)+K$, and the output dimension is LK. Hence, the computational complexity of both the actor networks and their corresponding target networks is represented as $O(a1) = (3LK - L + K)^2$, and the complexity of the critic networks and their target networks is denoted as $O(c1) = (8LK - 2L + 2K)(2LK - L + K) + 2L^2K^2 + 2LK$. The overall complexity is represented as $O(ap) = O(a1) + O(c1)$. Moreover, in the DDPG layer for power allocation, the input dimension is $L(3K-1)+K$ and the output dimension is LK, the complexity of the actor networks and their target networks is symbolized as $O(a2) = (4LK - L + K)^2$. Similarly, the complexity of the critic networks and their target networks is expressed as $O(c2) = (10LK - 2L + 2K)(3LK - L + K) + 2L^2K^2 + 2LK$. The overall complexity is represented as $O(pa) = O(a2) + O(c2)$, resulting in the complexity of the H-DDPG algorithm being denoted as $O(H) = O(ap) + O(pa)$.

7 Numerical results

In this section, the implementation details of the HDRL method for solving the optimization problem given by (13) are presented, followed by evaluating its performance and comparing it with other existing methods.

7.1 Simulation parameters

The network topology comprises $L = 12$ APs and $K = 8$ UEs randomly distributed within a 0.5 km radius. The number of antennas per AP is denoted as $N = 4$. In the case of each randomly generated topology as described above, the positions of UEs and APs remain constant during the evaluation phase. A random pilot assignment is employed, in which each UE randomly selects a pilot sequence from an orthogonal pilot pool with a length of τ_p. Considering the communicate operates on a 20 MHz channel. The total receiver noise power is -94 dBm. Each AP has the maximum downlink transmit power of $p_{\max} = 1\,W$, while each UE has an uplink power of $p_i = 100\,mW$ during the pilot transmit phase. The coherent block length is set to $\tau_c = 200$ [34–36]. The path loss model used to calculate the LSF coefficient is defined as follows:

$$\beta_{kl} = -35.3 - 37.6 \log_{10}\left(\frac{d_{kl}}{1m}\right) dB, \tag{29}$$

where d_{kl} represents the distance between the k-th UE and the l-th AP. Table 1 summarizes the simulation parameters and represents the urban micro-area environment.

In this algorithm, the high and low layer networks have the same structure. The actor neural network consists of a fully connected layer of input layer and average dimension hidden layer, and it is connected to a second fully connected hidden layer of output action space dimension by batch normalization and layer normalization. The number of hidden layer neurons is the average over the input dimension and output dimension, respectively. Moreover, critic neural network is divided into two parts:

Table 1 Environmental parameters

Regional area	**0.5 km × 0.5 km**
Bandwidth	20 MHz
AP number	$L = 12$
Number of AP antennas	$N = 4$
Number of UEs	$K = 8$
Downlink noise coefficient	-94 dBm
Maximum transmit power of a single AP	$P_{\max}^{\mathrm{dl}} = 1$ W
Coherent block length	$\tau_c = 200$
Pilot length	$\tau_p = 10$
Internal circuit power	$P_k^{c,ue} = 0.1\,W$
Internal circuit power	$P_l^{c,ap} = 0.2$ W
Fixed power consumption	$P_l^{fix} = 0.825$ W
Fixed power consumption	$P_{cpu}^{fix} = 5$ W
Signaling power	$P_l^{sig} = 0.01$ W
Energy consumption per bit	$P_{cpu}^{cod} = 0.1$ mW

state and action. The state part includes the input layer, the full connection layer of the average dimension hidden layer, which is processed by batch normalization and layer normalization. The action part connects the status part output and the action input with the full connection layer. The number of hidden layer neurons is twice that of the average input dimension. The critic network finally outputs Q-value through the output layer. ReLU is chosen as the activation function for the high-level aspect, while the Sigmoid function is selected for the low-level aspect.

The learning rate for training the actor network is set to $\delta = 1e-4$, the learning rate for training the critic network is $\delta = 1e-3$, the discount factor is 0.99, and the soft update parameter, which governs the gradual update of target network parameters, is 0.001 [37]. The target network is created as a duplicate of both the actor network and the critic network to ensure training stability. The neural network was trained using mini-batch stochastic gradient descent (mini-batch SGD) with a mini-batch size of 128, and an experience replay buffer with a maximum capacity of 10,000. It is assumed that all UEs have the same $SE_k^{\min}$, as CF-MIMO systems offer the advantage of delivering a consistently satisfactory service to all UEs in the network.

7.2 Simulation results discussion

To evaluate the performance of the proposed H-DDPG-based methods, three other algorithms were used for comparison. One approach is to employ a single-layer DDPG algorithm to address the challenge of jointly optimizing AP clustering and power allocation. The single-layer DDPG framework generates an action corresponding to AP clustering and power allocation at each time slot, which is represented as "S-DDPG." The second approach involves applying the DDPG algorithm to AP clustering, while power allocation is addressed using a conventional method, referred to as "AP-DDPG". The third is the traditional joint optimization scheme [38], which is denoted as "LP-MMSE." Compared to H-DDPG, S-DDPG is used to compare the advantages of hierarchical structure and provides an alternative method to solve the joint optimization problem. Moreover, the AP-DDPG scheme is chosen to independently study the effects of AP clustering and power allocation, which facilitates to understand the advantages of DRL in specific problems and the applicability of traditional methods for large-scale problems. As a traditional optimization scheme, LP-MMSE provides a benchmark against DRL-based methods. For S-DDPG, the total complexity of the actor network and its corresponding target network is denoted as $O(a) = (5LK - L + K)^2$. Similarly, the complexity of the critic network and its associated target network is represented as $O(c) = (14LK - 2L + 2K)(3LK - L + K) + 8L^2K^2 + 4LK$, whose complexity is $O(s) = O(a) + O(c)$. In AP-DDPG, the complexity is $O(AP) = O(a1) + O(c1) + LK$.

Since SE and EE affect each other, when SE is increased, EE may decrease. It is evident from Figs. 5 and 6 that when subjected to similar SE constraints, the CF-mMIMO system employing H-DDPG attains the highest EE. It improves EE by nearly 35% and 19% compared to S-DDPG and AP-DDPG, respectively. Compared with traditional optimization schemes, the EE performance gains are more significant. Hence, while the computational complexity of the S-DDPG and AP-DDPG algorithms is lower compared to the H-DDPG algorithm, but it is evident that the optimization performance of the H-DDPG algorithm is significantly superior to both of these algorithms.

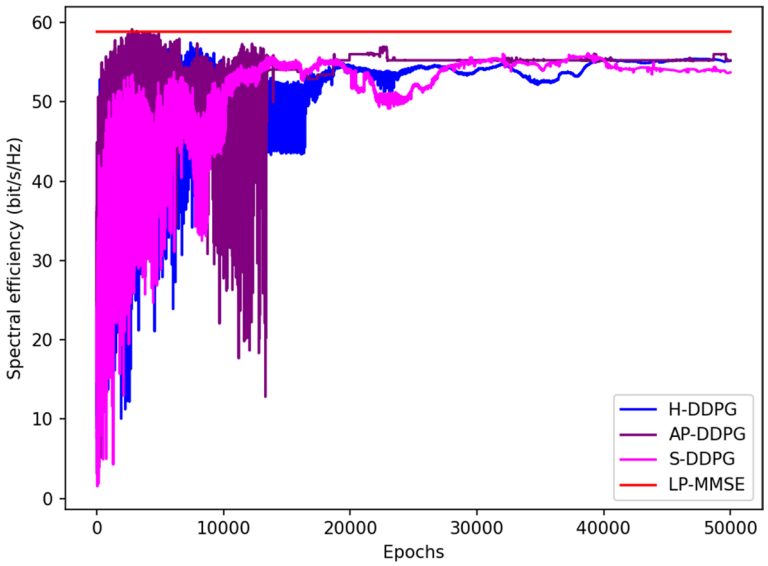

Fig. 5 Comparison of SE with Epochs increase under different schemes with $L = 12$, $K = 8$

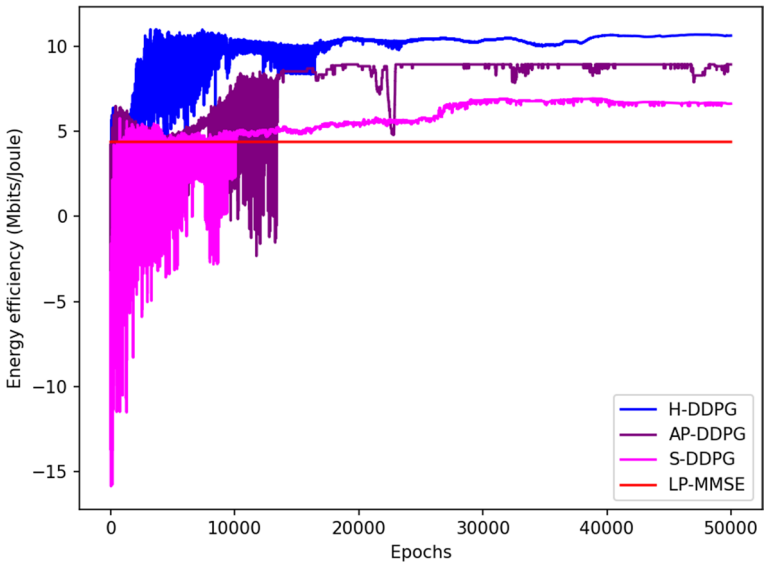

Fig. 6 Comparison of EE with Epochs increase under different schemes with $L = 12$, $K = 8$

This is attributed to the fact that H-DDPG decomposes the problem into two levels: a high level, which addresses AP clustering, and a low level, focusing on power allocation, with each level having its dedicated policy network. This layered approach makes the optimizing decisions at different levels more flexible and efficient. In addition, the hierarchical approach helps the policy network reach convergence more easily because the action space and state space at each stage are relatively small. Simultaneously, the high-level and low-level policies can be optimized to enhance system-level and link-level performance, thereby elevating the overall system's performance. Moreover, it is important to note that the curve performs poorly at the beginning, but converge rapidly as the

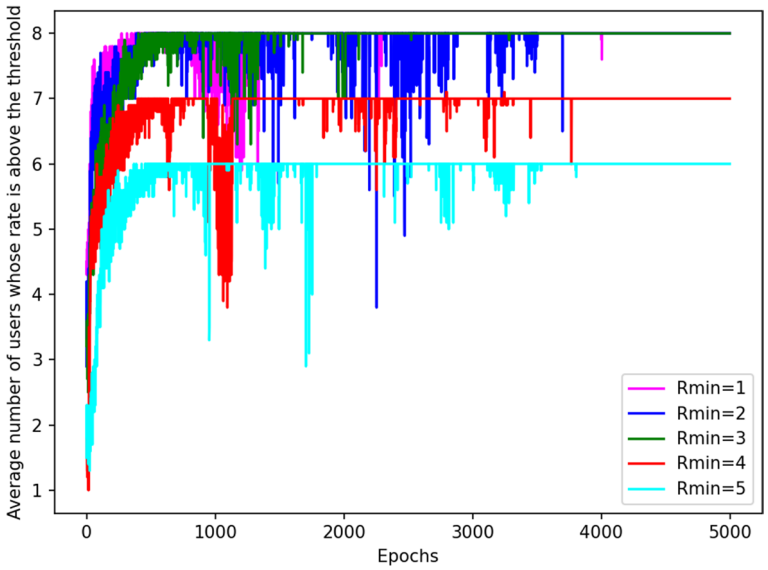

Fig. 7 Average number of UEs whose SE exceeds the threshold

training rounds increase, because the DDPG algorithm needs to interact with the environment several times to gradually adjust the neural network weight.

As depicted in Fig. 7, it is evident that within the H-DDPG algorithm, the number of UEs meeting the minimum SE requirement gradually rises with the advancement of iterations. Furthermore, it is evident that even when the minimum SE threshold is set at a high value and the EE does not exhibit significant differences, the majority of UEs can still satisfy this threshold. This highlights the advantages of H-DDPG algorithms in the tradeoff of SE and EE, which can flexibly adjust system performance under different requirements to achieve better performance optimization.

The hierarchical structure enables H-DDPG to dynamically adjust its policy to adapt to changes in different environments and network states. This flexibility can be important in real networks, especially when network conditions are constantly changing, and H-DDPG can respond to those changes, thus achieves an improved EE performance. It adjusts the power allocation strategy according to the current network state and UE needs and gradually improves the SE of each UE. Moreover, the algorithm efficiently controls the transmission power of each AP to guarantee system stability and feasibility, consequently mitigating potential interference issues. H-DDPG algorithms consider power allocation collaboratively on a global scale to maximize total EE, rather than just local optimization, which helps to make more efficient use of limited resources.

Figure 8 illustrates the cumulative distribution function of EE per UE for various methods. From Fig. 8, it is evident that the H-DDPG method exhibits the outstanding EE performance, the AP-DDPG method shows a slightly inferior performance, and the S-DDPG method demonstrates the poorest EE performance. This is mainly because the H-DDPG method makes full use of the knowledge sharing and reuse mechanism brought by the hierarchical structure, so that it can make more accurate decisions. In the hierarchical structure, agents at different levels can share information, and

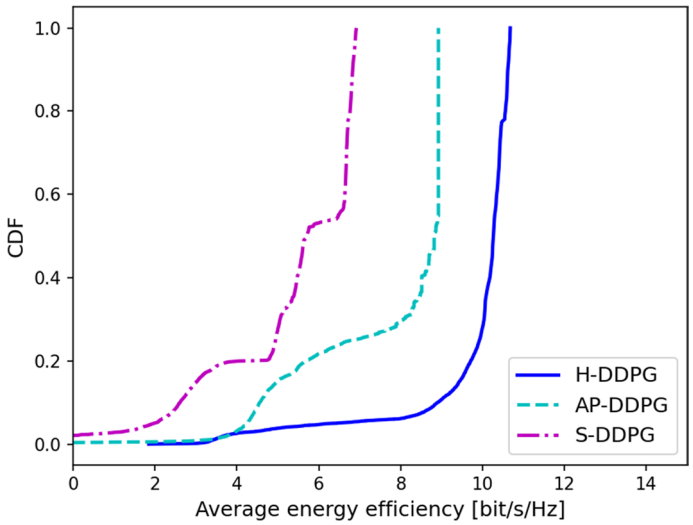

Fig. 8 CDF of the EE per UE under different schemes with $L = 12$ and $K = 8$

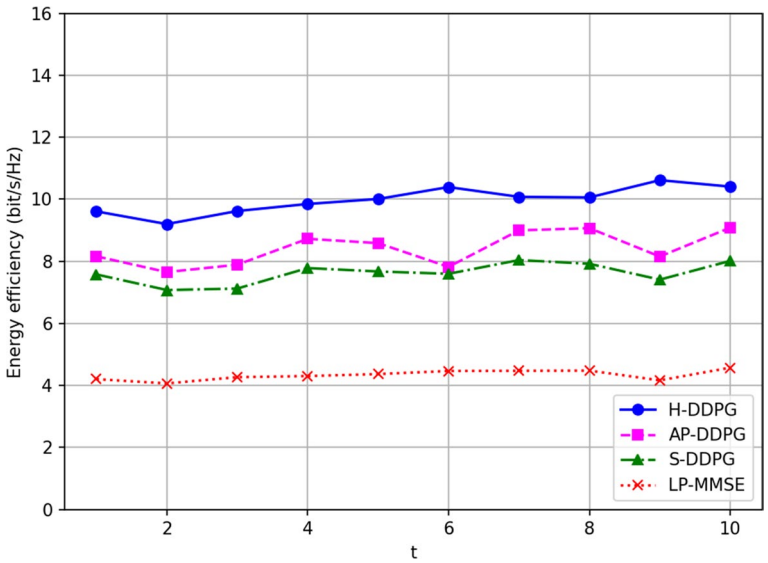

Fig. 9 EE versus times

the policies at lower levels can be guided by the policies at higher levels, thus the knowledge of the whole system can be transferred between different levels, effective policies can be learned faster. This provides a more comprehensive understanding of the entire system state, enabling more precise optimization. Therefore, H-DDPG can learn strategies more efficiently, especially in the case of a reasonably divided hierarchy, and it usually has faster convergence.

As shown in Fig. 9, we can observe the effect of the relative position between the AP and the UE on the system EE over time. It is worth noting that the H-DDPG method consistently performs well in terms of the EE performance, consistently outperforming other methods. In comparison, AP-DDPG method, although slightly inferior to

H-DDPG, still ranks second and shows the good performance. In addition, S-DDPG method is inferior to H-DDPG and AP-DDPG in the EE performance, it is still superior to the traditional algorithm.

These observations highlight the effectiveness of the H-DDPG method. The reason for this difference is that H-DDPG breaks down the joint optimization problem into multiple levels, each focused on a different task or goal. This hierarchical control strategy enables agents to deal with multi-objective problems more effectively, and decomposes complex tasks into simpler subtasks, thus improving the manageability and execution efficiency of tasks. Moreover, H-DDPG can balance the capacity of exploration and utilization. Each level of strategy can play a role in different stages of exploration and utilization. High-level policies are generally more focused on the global exploration and general decision making, while low-level policies can be explored and adjusted in more detail. This facilitates to explore the environment during the learning process, while allowing for more refined strategic adjustments.

8 Conclusion

In this paper, we designed an energy-efficient AP clustering and power allocation for CF-mMIMO systems. To deal with the non-convex optimization problem, a H-DDPG algorithm is proposed. This algorithm takes the advantage of both DRL and hierarchical optimization to create a collaborative optimization framework which effectively handle AP clustering and power allocation. In particular, the original optimization problem first was divided into two sub-problems, and then the corresponding DDPG agents are trained separately. During the training phase, these agents learn from the environment, gradually improving their strategies to maximize the system EE. Simulation results clearly demonstrate that the proposed H-DDPG method leads to a substantial enhancement in the system's EE, compared to benchmark methods. Moreover, this study provides a new research perspective for resource allocation in CF-mMIMO systems, that is, by dividing the problem into small sub-problems in multiple stages, the problems at different levels can be optimized more flexibly. For the further research, in light of the centralized information exchange in single-agent DRL-based algorithms, it is essential to develop multi-agent DRL algorithms for resource allocation in CF-mMIMO systems. Also, the joint design of AP clustering and UE scheduling, as well as beamforming is promising to satisfy future massive connectivity in real-world applications.

Abbreviations

AMP	Approximate message passing
BER	Bit error rate
CS	Compressed sensing
CIR	Channel impulse response
EM	Expectation maximization
FB-Kalman	Forward–backward Kalman filter
GM	Gaussian mixture
IN	Impulse noise
LMMSE	Linear mean squared estimator
NMSE	Normalized mean square error
OFDM	Orthogonal frequency division multiplexing
PLC	Power line communication
SBL	Sparse Bayesian learning
SNR	Signal-to-noise Ratio

Acknowledgements
This work has been supported by the National Natural Science Foundation of China under Grants 62261013, and in part by Director Foundation of Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing under Grants GXKL06220104.

Author contributions
All the authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no Competing interests.

Received: 1 November 2023 Accepted: 11 January 2024
Published online: 26 January 2024

References

1. T.L. Marzetta, Noncooperative cellular wireless with unlimited numbers of base station antennas. IEEE Trans. Wireless Commun.Commun. **9**(11), 3590–3600 (2010)
2. L. Daza, S. Misra, Fundamentals of massive MIMO. IEEE Wirel. Commun.Wirel. Commun. **25**(1), 9–9 (2018)
3. E. Björnson, J. Hoydis, L. Sanguinetti, Massive MIMO has unlimited capacity. IEEE Trans. Wireless Commun.Commun. **17**(1), 574–590 (2018)
4. E. Björnson, J. Hoydis, L. Sanguinetti, Massive MIMO networks Spectral, energy, and hardware efficiency. Found. Trends Signal Process. **11**(3–4), 154–655 (2017)
5. M. Matthaiou, O. Yurduseven, H.Q. Ngo, D. Morales-Jimenez, S.L. Cotton, V.F. Fusco, The road to 6G: ten physical layer challenges for communications engineers. IEEE Commun. Mag.Commun. Mag. **59**(1), 64–69 (2021)
6. H.Q. Ngo, A. Ashikhmin, H. Yang, E.G. Larsson, T.L. Marzetta, Cell-free massive MIMO versus small cells. IEEE Trans. Wireless Commun.Commun. **16**(3), 1834–1850 (2017)
7. H. He, X. Yu, J. Zhang, S. Song, K.B. Letaief, Cell-free massive MIMO for 6G wireless communication networks. J. Commun. Inf. Netw. **6**(4), 321–335 (2021)
8. Ö.T. Demir, E. Björnson, L. Sanguinetti, Foundations of user-centric cell-free massive MIMO. Found. Trends Signal Process. **14**(3–4), 162–472 (2021)
9. T.X. Doan, T.Q. Trinh, An overview of emerging technologies for 5G: full-duplex relaying cognitive radio networks, device-to-device communications and cell-free massive MIMO. J. Thu Dau Mot Univ. **2**, 348–364 (2020)
10. M. Fallgren, G. Fodor and A. Forsgren, "Optimization approach to joint cell, channel and power allocation in wireless communication networks," *2012 In: Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pp. 829–833, 2012.
11. V. M. T. Palhares, R. C. de Lamare, A. R. Flores and L. T. N. Landau, "Iterative MMSE precoding and power allocation in cell-free massive MIMO systems," *2021 IEEE Statistical Signal Processing Workshop (SSP)*, pp.181–185, 2021.
12. M. Hong, A. Garcia, J. Barrera, S.G. Wilson, Joint access point selection and power allocation for uplink wireless networks. IEEE Trans. Signal Process. **61**(13), 3334–3347 (2013)
13. X. Lin, F. Xu, J. Fu, Y. Wang, Resource allocation for TDD cell-free massive MIMO systems. Electronics **11**(12), 2022 (1944)
14. W. Jess, K. Arulkumaran, M. Crosby, The societal implications of deep reinforcement learning. J. Artif. Intell. Res. **70**, 1003–1030 (2021)
15. J. Jang, H.J. Yang, Deep reinforcement learning-based resource allocation and power control in small cells with limited information exchange. IEEE Trans. Veh. Technol.Veh. Technol. **69**, 13768–13783 (2020)
16. Y.S. Nasir, D. Guo, Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks. IEEE J. Sel. Areas Commun.Commun. **37**, 2239–2250 (2018)
17. I.M. Braga, R.P. Antonioli, G. Fodor, Y.C.B. Silva, W.C. Freitas, Decentralized joint pilot and data power control based on deep reinforcement learning for the uplink of cell-free systems. IEEE Trans. Veh. Technol.Veh. Technol. **72**(1), 957–972 (2023)
18. L. Luo, J. Zhang, S. Chen, X. Zhang, B. Ai, D.W.K. Ng, Downlink power control for cell-free massive MIMO with deep reinforcement learning. IEEE Trans. Veh. Technol.Veh. Technol. **71**(6), 6772–6777 (2022)
19. M. Rahmani, M. Bashar, M. J. Dehghani, P. Xiao, R. Tafazolli and M. Debbah, "Deep reinforcement learning-based power allocation in uplink cell-free massive MIMO," *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 459–464, 2022.

20. Z. Li, C. Hu, W. Wang, Y. Li, G. Wei, Joint access point selection and resource allocation in MEC-assisted network: a reinforcement learning based approach. China Commun. **19**(6), 205–218 (2022)
21. J. Tan, L. Zhang and Y. -C. Liang, "Deep reinforcement learning for channel selection and power control in D2D networks," *2019 IEEE Global Communications Conference (GLOBECOM)*, pp.1–6, 2019.
22. H. Zhang, S. Chong, X. Zhang, N. Lin, A deep reinforcement learning based D2D relay selection and power level allocation in mmwave vehicular networks. IEEE Wireless Commun. Lett. **9**(3), 416–419 (2020)
23. Z. Liu, J. Zhang, Z. Liu, H. Xiao and B. Ai, "Double-layer power control for mobile cell-free XL-MIMO with multi-agent reinforcement learning," *IEEE Transactions on Wireless Communications*, 2023, early access.
24. D. Zha, K.H. Lai, Q. Tan, S. Ding, N. Zou, X. Hu, "Towards automated imbalanced learning with deep hierarchical reinforcement learning," In: *Proceedings of the 31st ACM International Conference on Information,* 2022.
25. S. Liu, J. Wu, J. He, Dynamic multichannel sensing in cognitive radio: hierarchical reinforcement learning. IEEE Access **9**, 25473–25481 (2021)
26. W. Yuan, H. Muñoz-Avila, "Hierarchical reinforcement learning for deep goal reasoning: an expressiveness analysis," *ArXiv*, 2020.
27. F. Tan, P. Wu, Y.-C. Wu, M. Xia, Energy-efficient non-orthogonal multicast and unicast transmission of cell-free massive MIMO systems with SWIPT. IEEE J. Sel. Areas Commun.Commun. **39**(4), 949–968 (2021)
28. S. Chen, J. Zhang, E. Björnson, Ö. Demir, B. Ai, "Energy-efficient cell-free massive MIMO through sparse large-scale fading processing," *IEEE Transactions on Wireless Communications*, pp. 1–37, May. 2023, Early Access.
29. H.Q. Ngo, L.-N. Tran, T.Q. Duong, M. Matthaiou, E.G. Larsson, On the total energy efficiency of cell-free massive MIMO. IEEE Trans. Green Commun. Netw. **2**(1), 25–39 (2018)
30. Y. Zhao, I.G. Niemegeers, S.M.H. De Groot, Dynamic power allocation for cell-free massive MIMO: deep reinforcement learning methods. IEEE Access **9**, 102953–102965 (2021)
31. R. Y. Chang, S. -F. Han and F. -T. Chien, "Reinforcement learning-based joint cooperation clustering and content caching in cell-free massive MIMO networks," *2021 IEEE 94th vehicular technology conference (VTC2021-Fall)*, Norman, OK, USA, 2021, pp. 1-7
32. N. Ghiasi, S. Mashhadi, S. Farahmand, S.M. Razavizadeh, I. Lee, Energy efficient AP selection for cell-free massive MIMO systems: Deep reinforcement learning approach. IEEE Trans. Green Commun. Netw. **7**(1), 29–41 (2023)
33. K. Yu, C. Zhao, G. Wu, and G. Y. Li, "Distributed two-tier DRL framework for cell-free network: Association, beamforming and power allocation," *arXiv e-prints*, 2023.
34. A. Zhou, J. Wu, E.G. Larsson, P. Fan, Max-min optimal beamforming for cell-free massive MIMO. IEEE Commun. Lett. Commun. Lett. **24**(10), 2344–2348 (2020)
35. J. Zhang, J. Zhang, E. Björnson, B. Ai, Local partial zero-forcing combining for cell-free massive MIMO systems. IEEE Trans. Commun.Commun. **69**(12), 8459–8473 (2021)
36. Ö. Özdogan, E. Björnson, J. Zhang, Performance of cell-free massive MIMO with rician fading and phase shifts. IEEE Trans. Wireless Commun.Commun. **18**(11), 5299–5315 (2019)
37. Y. Zhang, J. Zhang, Y. Jin, S. Buzzi and B. Ai, "Deep learning-based power control for uplink cell-free massive MIMO systems," *2021 IEEE global communications conference (GLOBECOM)*, Madrid, Spain, 2021, pp. 1-6
38. E. Björnson, L. Sanguinetti, Scalable cell-free massive MIMO systems. IEEE Trans. Commun.Commun. **68**(7), 4247–4261 (2020)

Publisher's Note