Received July 1, 2021, accepted July 12, 2021, date of publication July 14, 2021, date of current version July 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3097243

Dynamic Power Allocation for Cell-Free Massive MIMO: Deep Reinforcement Learning Methods

YU ZHAO, (Graduate Student Member, IEEE), IGNAS G. NIEMEGEERS, AND SONIA M. HEEMSTRA DE GROOT

Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands

Corresponding author: Yu Zhao (y.zhao3@tue.nl)

ABSTRACT Power allocation plays a central role in cell-free (CF) massive multiple-input multiple-output (MIMO) systems. Many effective methods, e.g., the weighted minimum mean square error (WMMSE) algorithm, have been developed for optimizing the power allocation. Since the state of the channels evolves in time, the power allocation should stay in tune with this state. The present methods to achieve this typically find a near-optimal solution in an iterative manner at the cost of a considerable computational complexity, potentially compromising the timeliness of the power allocation. In this paper we address this problem by exploring the use of data-driven methods since they can achieve near-optimal performance with a low computational complexity. Deep reinforcement learning (DRL) is one such method. We explore two DRL power allocation methods, namely the deep Q-network (DQN) and the deep deterministic policy gradient (DDPG). The objective is to maximize the sum-spectral efficiency (SE) in CF massive MIMO, operating in the microwave domain. The numerical results, obtained for a 3GPP indoor scenario, show that the DRL-based methods have a competitive performance compared with WMMSE and the computational complexity is much lower. We found that the well-trained DRL methods achieved at least a 33% higher sum-SE than the WMMSE algorithm. The execution times of the DRL methods in our simulation platform are 0.1% of the WMMSE algorithm.

INDEX TERMS Cell-free massive MIMO, deep reinforcement learning, deep Q-network, deep deterministic policy gradient, power allocation.

I. INTRODUCTION

In cell-free (CF) massive multiple-input multiple-output (MIMO) systems, a large number of access points (APs) are distributed over the coverage area. The concept of a cell, where all user equipments (UEs) within a particular cell are served exclusively by a particular AP, no longer exists. Each UE is jointly served by all APs using the same time-frequency resource in a time-division duplex (TDD) mode [1]. One advantage of such a system is that because the antennas are geographically distributed, it is less affected by shadow fading than a centralized system. Another advantage is that the average distance from a UE to its nearest AP is smaller. The drawback is the need for an optical or wireless fronthaul to connect the APs with a central controller (CC), see Fig. 1.

CF massive MIMO can reap all benefits of massive MIMO [2], i.e., favorable propagation and channel hardening when using many antennas at the APs [3]. Favorable propagation means that the UE's channel vectors are almost orthogonal. Channel hardening means that the beamforming

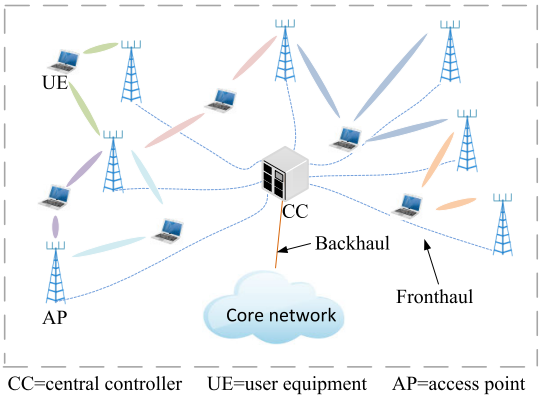


FIGURE 1. Cell-free massive MIMO system.

The associate editor coordinating the review of this manuscript and approving it for publication was M. Shamim Kaiser.

transforms the fading multi-antenna channel into an almost deterministic scalar channel. These properties simplify the signal processing and resource allocation. Since in massive MIMO, and in particular in CF massive MIMO, different UEs are simultaneously served by the same time-frequency resource block, controlling the inter-user interference is important. Power allocation plays a crucial role in controlling this interference and in optimizing the performance of the downlink [4, Chapter 7]. This is the issue we are concerned with in this paper. Unless otherwise stated, the power allocation in this paper refers to the downlink power allocation.

Although many existing power allocation algorithms show excellent performance in simulations and theoretical analysis, their implementation in real systems faces serious obstacles [5]. In particular, the computational complexity is problematic. Since the state of the channels evolves in time, the power allocation should stay in tune with this state. The present methods to achieve this typically find a near-optimal solution in an iterative manner, at the cost of a considerable computational complexity, potentially compromising the timeliness of the power allocation. For example, the popular weighted minimum mean square error (WMMSE) algorithm requires complex operations such as matrix inversion and bisection in each iteration [6]. Even if the power allocation in CF massive MIMO can be performed at the CC and on the large-scale fading time scale [1], it is still challenging to make an implementation that meets the real-time constraints.

Recently, data-driven methods, in particular deep learning (DL) [7] methods, have been proposed for this task since they achieve near-optimal performance with a low computational complexity. Several DL-based methods have already been proposed for power allocation in wireless communication systems. There are two main branches of DL-based power allocation methods, namely supervised learning and reinforcement learning. For supervised learning, the key idea is to use a deep neural network (DNN) to approximate the results of a known but computationally complex algorithm [8]. Specifically, in [5] a DNN was trained to approximate the WMMSE power-allocation algorithm to manage the interference in multi-cell networks; in [9] a deep convolutional neural network (CNN) was proposed to approximate an iterative algorithm for power allocation in cellular massive MIMO. In [10] a fully connected neural network and a recurrent neural network (RNN) were proposed to maximize the SE and implement the max-min power policy respectively for cellular massive MIMO. None of the above works considered CF massive MIMO. In [11], a deep CNN (DCNN) was proposed to determine the mapping from the large-scale fading coefficients to the optimal uplink power through solving the sum-rate maximization problem using the quantized channel. In [12], we proposed a two-stage DNN to approximate the bisection algorithm for power allocation in CF massive MIMO. In reinforcement learning, the idea is to determine how agents ought to take actions by observing an environment such that a cumulative reward is maximized [13]. Several deep reinforcement learning (DRL)-based methods have been applied to solve power optimization problems in cellular networks, e.g., [14]–[17]. However, none of these works considered massive MIMO systems. [18] proposed a DRL method for handover and power optimization for massive-antenna base stations (BSs). However, in the case a UE is only served by one BS, which is very different from CF massive MIMO.

In this paper, we propose two DRL-based power allocation methods, namely the deep Q-network (DQN) [13] and the deep deterministic policy gradient (DDPG) [19], which have the merits of low computational complexity. Unlike traditional supervised learning, which needs a huge training dataset generated by a computationally complex algorithm [8], in DRL the training is done by means of rewards obtained in trial-and-error interactions with the environment. This requires only a small number (e.g., 4 in this paper) of matrix multiplications for performing the power allocation. Moreover, we will find that the performance, in terms of the downlink sum-spectral efficiency (SE), is competitive compared with the WMMSE algorithm. To the best of our knowledge, there is no published work considering using DRL-based methods for power allocation in CF massive MIMO.

The objective of the power allocation we choose is to maximize the downlink sum-SE, corresponding to maximizing the overall network capacity. The normalized conjugate beamforming (NCB) [20] is used for analysis because it can be performed locally at each AP, which implies that there is no overhead for sending channel state information (CSI) from the APs to the CC via the fronthaul. Moreover, it has been shown that NCB benefits more from channel hardening than classic conjugate beamforming (CB), in the sense that it has a tighter lower bound for the sum-SE [3], [20]. Our contributions are threefold:

(1). The formalization of the sum-SE maximization problem for CF massive MIMO with multi-antenna APs and single-antenna UEs, taking into account imperfect CSI and pilot contamination during uplink channel estimation.

(2). The proposition of two DRL-based power allocation methods, namely the DQN and the DDPG, for CF massive MIMO systems.

(3). The evaluation of the sum-SE obtained by the two DRL-based power allocation methods, by comparing them with the WMMSE algorithm for a 3GPP indoor mixed office scenario [21] with mobile UEs. The simulation results show that the DQN-based method and the DDPG-based method have competitive performance with the WMMSE algorithm in terms of sum-SE and have much lower computational complexity.

The rest of the paper is structured as follows. Section II describes the system models on which we base our analysis. Section III formalizes the max sum-SE power allocation problem and the WMMSE algorithm. Section IV describes the two DRL-based power allocation methods. Section V shows the simulation results and their interpretation. Finally, Section VI draws the conclusions.

II. SYSTEM MODEL

Consider a microwave CF massive MIMO system where N APs serve K UEs using the same time-frequency resource under TDD operation. Each AP has M antennas, whereas each UE has a single antenna. All APs are connected to a CC through a fronthaul, see Fig. 1. As in [2] and [3], we define the channel vector between AP n and UE k as:

$$\mathbf{g}_{n,k} = \beta_{n,k}^{1/2}\mathbf{h}_{n,k} \tag{1}$$

where $\beta_{n,k}$ is the large-scale fading between AP n and UE k, $\mathbf{h}_{n,k}$ is an $M \times 1$ small-scale fading vector whose elements are assumed to be Rayleigh fading, i.e., the elements of $\mathbf{h}_{n,k}$ are i.i.d. $\mathcal{CN}(0, 1)$ random variables. One should notice that the channels between UEs and AP antennas are in general not the same. Some may be modelled by the Rayleigh model, others by the Rician model, and some might have several dominant multi-path components. Since we concentrate on the power allocation method, we simplify our task by assuming Rayleigh fading for all channels, as is done in [2] and [3]. However, for assessing the performance gains obtained by the methods more realistically, one would need to consider channel models that are based on measurements. Let τ_c be the coherence time, during which the channel can be regarded as constant [4, Chapter 2]. A part of the coherence time, τ_p, with $\tau_p < \tau_c$, is used for channel estimation, the remaining part, $\tau_c - \tau_p$, is used for downlink data transmission. The beamforming is valid for approximately the coherence time. It is performed based on the CSI estimated by the APs. This implies that there are real-time constraints on the computation of the beamforming: the results should be available before the start of each data transmission phase.

If one has the benefit of channel hardening, i.e., small-scale fading can be neglected, due to the spatial diversity created by the large number of antennas [3], power allocation can be done on a longer time scale, commensurate with the large-scale fading time τ_l. τ_l is defined as the time during which the effect of large-scale fading can be considered to be constant. It typically is around 40 times the coherence time τ_c [1]. Fig. 2 shows the time scales for beamforming and power allocation.

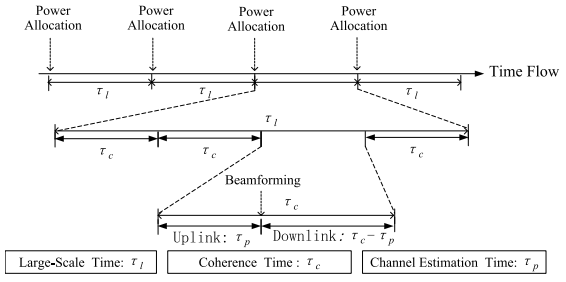


FIGURE 2. Time scales for beamforming and power allocation.

A. CHANNEL ESTIMATION

The APs estimate the channels via the uplink pilots. By using the minimum mean-square error (MMSE) estimation [4, Chapter 3], the estimate $\hat{\mathbf{g}}_{n,k}$ includes M independent and identical Gaussian components. The mean-square of the m-th component, denoted as $\gamma_{n,k}$, is given by [2]:

$$\gamma_{n,k} = \frac{\tau_p p_p \beta_{n,k}^2}{\tau_p p_p \sum_{k'=1}^{K} \beta_{n,k'} |\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H|^2 + 1} \tag{2}$$

where p_p is the normalized pilot power, $\boldsymbol{\psi}_k$ is the time sequence with elements 0-1 and $|\boldsymbol{\psi}_k|^2 = 1$. One remark is that if $\tau_p \geq K$, one can choose $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_K$ to be pairwise orthogonal, i.e., $\boldsymbol{\psi}_k \boldsymbol{\psi}_k'^H = 0$ for $k \neq k'$. However, because of the limited length of the coherence time τ_c, in general $\tau_p < K$, some pilot sequences are reused, leading to pilot contamination in CF massive MIMO [1]. Pilot contamination degrades the performance of the CF massive MIMO system in terms of SE.

B. DOWNLINK DATA TRANSMISSION

Based on the estimated channels, the APs use NCB to transmit their signals to the UEs. Let q_k, with $\mathbb{E}\{|q_k|^2\} = 1$, be the intended signal for user k, then the transmitted signal from AP n is:

$$\begin{aligned}\mathbf{x}_n &= \sum_{k'=1}^{K} \sqrt{p_{n,k'}} \frac{\hat{\mathbf{g}}_{n,k'}^*}{\sqrt{\mathbb{E}\{||\hat{\mathbf{g}}_{n,k'}||^2\}}} q_{k'} \\ &= \sum_{k'=1}^{K} \sqrt{p_{n,k'}} \frac{\hat{\mathbf{g}}_{n,k'}^*}{\sqrt{M\gamma_{n,k'}}} q_{k'}\end{aligned} \tag{3}$$

where $p_{n,k'}$ is the normalized downlink transmission power from AP n to user k' satisfying $p_{n,k'} \leq p_{\max}$, where $p_{\max}$ is the transmission power limit. UE k will receive the signal y_k, which is the superposition of the signals from all APs in the whole system:

$$y_k = \sum_{n=1}^{N}\sum_{k'=1}^{K} \sqrt{p_{n,k'}} \frac{\mathbf{g}_{n,k}^T \hat{\mathbf{g}}_{n,k'}^*}{\sqrt{M\gamma_{n,k'}}} q_{k'} + w_k \tag{4}$$

where $w_k \sim \mathcal{CN}(0, 1)$ is the additive noise at UE k.

C. SPECTRAL EFFICIENCY

The intended signal for UE k can be detected from y_k as follows [1]:

$$\begin{aligned} y_k = &\underbrace{\mathbb{E}\{\sum_{n=1}^{N} \sqrt{p_{n,k}} \frac{\mathbf{g}_{n,k}^T \hat{\mathbf{g}}_{n,k}^*}{\sqrt{M\gamma_{n,k}}} q_k\}}_{\text{Intended}} \\ &+ \underbrace{(\sum_{n=1}^{N} \sqrt{p_{n,k}} \frac{\mathbf{g}_{n,k}^T \hat{\mathbf{g}}_{n,k}^*}{\sqrt{M\gamma_{n,k}}} - \mathbb{E}\{\sum_{n=1}^{N} \sqrt{p_{n,k}} \frac{\mathbf{g}_{n,k}^T \hat{\mathbf{g}}_{n,k}^*}{\sqrt{M\gamma_{n,k}}}\})q_k}_{\text{Fluctuation}} \\ &+ \underbrace{\sum_{k'\neq k}\sum_{n=1}^{N} \sqrt{p_{n,k'}} \frac{\mathbf{g}_{n,k}^T \hat{\mathbf{g}}_{n,k'}^*}{\sqrt{M\gamma_{n,k'}}} q_{k'}}_{\text{Interference}} + \underbrace{w_k}_{\text{Noise}} \end{aligned} \tag{5}$$

There are four parts in (5), namely, the intended signal, the fluctuation caused by uncertain channel gains, the interference from other UEs, and the noise. Then SE_k, the downlink SE for user k, is given by (6), as shown at the bottom of the page, [4, Chapter 4].

III. MAX SUM-SE POWER ALLOCATION

We consider the max sum-SE power allocation policy, which can be formulated as follows:

$$\max_{p_{n,k}} \sum_{k=1}^{K} SE_k$$
$$s.t.\ p_{n,k} \leq p_{\max}, \quad \forall n, k \tag{7}$$

The problem in (7) is non-convex and NP-hard, since the computational complexity increases exponentially as N and K increase. A well-adopted method to solve (7) is the WMMSE algorithm, which converts the sum-SE maximization problem to an equivalent minimization problem of the mean square error in the data detection [6]. Specifically, the algorithm works as follows:

From an initial point $\{v_{n,k}^0\}$ satisfying the constraints, the optimal power allocation is obtained by updating $\{v_{n,k}, u_{n,k}, w_{n,k}\}$ in an iterative manner, where $v_{n,k}, u_{n,k}, w_{n,k}$ are optimization variables. The variables $\{v_{n,k}, u_{n,k}, w_{n,k}\}$, for all n, k in iteration I, are updated using (8), (9), and (10), as shown at the bottom of the page, where (10) implies that the variable $v_{n,k}$ should be in range from 0 to $\sqrt{p_{\max}}$. The details of the WMMSE algorithm in solving (7) are given by Algorithm 1. The algorithm stops when the condition $w_{n,k} < \epsilon$ is fulfilled. The value of ϵ depends on the convergence behavior of the WMMSE algorithm. Similarly to [6] we set $\epsilon = 0.01$.

The computational complexity mainly lies in steps 3 and 5. The calculation of $\gamma_{n,k}$ has a complexity of $O(K)$. For the denominator of (8) and (10), the complexity is $O(NK^2)$. The computational complexity for updating $p_{n,k}$, is $O(INK^2)$. Finally, the total computational complexity to update NK links is $O(IN^2K^3)$.

IV. DRL-BASED POWER ALLOCATION METHODS

In this section we propose two DRL-based power allocation methods, namely the DQN and the DDPG.

Algorithm 1 Pseudo Code of WMMSE for (7)

1: Initialize $v_{n,k}^0$ such that $(v_{n,k}^0)^2 \leq p_{\max}, \forall k, n$.
2: Set $I = 1$, repeat:
3: Update the variables $u_{n,k}^I$ for all n, k, by using (8).
4: Update the variables $w_{n,k}^I$ for all n, k, by using (9).
5: Update the variables $v_{n,k}^I$ for all n, k, by using (10).
6: Set $I = I + 1$.
7: Until $w_{n,k} < \epsilon$.
8: Output: $p_{n,k} = (v_{n,k})^2$.

A. BACKGROUND OF DRL

DRL is a category of machine learning where an agent learns by interacting with its dynamic environment through a repeated sequence of observations, actions and rewards [13]. At time slot t, where t is an integer, by observing the state s^t, the agent takes action $a^t \in \boldsymbol{A}$ according to a certain policy π, then gets the reward r^t from the environment and enters the next state s^{t+1}. The four-tuple (s^t, a^t, r^t, s^{t+1}), describes a single interaction with the environment. We define e^t, where $e^t = (s^t, a^t, r^t, s^{t+1})$ as an experience sequence. The agent aims to find an optimal policy to maxmize the cumulative reward:

$$R^t = r^t + \omega r^{t+1} + \omega^2 r^{t+2} + \dots \tag{11}$$

where $\omega \in [0, 1)$ is a discount factor that trades off the importance of immediate and future rewards. To evaluate a policy, the action value is defined:

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\{R^t | s^t = s, a^t = a\} \tag{12}$$

$$SE_k = (1 - \frac{\tau_p}{\tau_c}) \log_2(1 + \frac{M(\sum_{n=1}^{N} \sqrt{p_{n,k}\gamma_{n,k}})^2}{M \sum_{k' \neq k} (\sum_{n=1}^{N} \sqrt{p_{n,k'}\gamma_{n,k'}} \frac{\beta_{n,k}}{\beta_{n,k'}})^2 |\boldsymbol{\psi}_k \boldsymbol{\psi}_{k'}^H|^2 + \sum_{k'=1}^{K} \sum_{n=1}^{N} p_{n,k'} \beta_{n,k} + 1}) \tag{6}$$

$$u_{n,k}^I = \frac{\sqrt{M} v_{n,k}^{I-1} \sqrt{\gamma_{n,k}}}{M \sum_{k'=1}^{K} (\sum_{n'=1}^{N} v_{n',k'}^{I-1} \sqrt{\gamma_{n',k'}} \frac{\beta_{n',k}}{\beta_{n',k'}})^2 |\boldsymbol{\psi}_k \boldsymbol{\psi}_{k'}^H|^2 + \sum_{k'=1}^{K} \sum_{n'=1}^{N} (v_{n',k'}^{I-1})^2 \beta_{n',k} + 1} \tag{8}$$

$$w_{n,k}^I = \frac{1}{1 - u_{n,k}^{I-1} \sqrt{M\gamma_{n,k}} v_{n,k}^{I-1}} \tag{9}$$

$$v_{n,k}^I = [\frac{w_{n,k}^I u_{n,k}^I \sqrt{M\gamma_{n,k}}}{\sum_{k'=1}^{K} \sum_{n'=1}^{N} w_{n',k'}^I (u_{n',k'}^I)^2 M \gamma_{n',k'}}]_0^{\sqrt{p_{max}}} \tag{10}$$

where the right side of (12) is the expected reward once action a^t is taken under state s^t and policy π. Reinforcement learning makes use of the Bellman equation [22] for the cumulative calculation:

$$Q^{\pi}(s,a)=\mathbb{E}\{r^t+\omega Q^{\pi}(s',\pi(s'))|s^t=s,a^t=a,s^{t+1}=s'\} \tag{13}$$

The optimal policy op, is the one that maximizes (13), i.e.,

$$op=\arg\max_{\pi} Q^{\pi}(s,a) \tag{14}$$

For convenience, we use $Q^{op}(s,a)$ to represent the action value of the optimal policy. The main two methods to find the optimal policy are: the value-based method DQN and the policy-based method DDPG.

B. DQN-BASED METHOD

The key to find the optimal policy for the DQN-based method is to obtain the action-value function of $Q^{\pi}(s,a)$. It is common to use a function approximator to estimate the action-value function, typically a lookup table or a linear function. If this approximator is a DNN, it is called DQN. The DQN is defined as $Q(s,a,\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the parameters (weights between neurons). The DQN is trained to approximate the action-value function, i.e., $\boldsymbol{\theta}$ is updated to approximate the action-value function of $Q^{\pi}(s,a)$. The agent stores the experiences in a data set $D=\{e^1,e^2,\dots e^t\}$, which is used to train the DQN by the gradient descent algorithm [7]. Ideally the training of DQN should use all data in each iteration, however, this is very expensive when the training set is huge. An efficient way is to use a random subset of the training set D^t, called mini-batch, to evaluate the gradients in each iteration [23]. Then the loss function, which is used to calculate the gradients for updating $\boldsymbol{\theta}$, is defined as:

$$L(\boldsymbol{\theta})=\sum_{e\in D^t}(\underbrace{r+\omega\max_{a'}Q(s',a',\hat{\boldsymbol{\theta}})}_{\text{target}}-Q(s,a,\boldsymbol{\theta}))^2 \tag{15}$$

where (15) is the loss function of the DQN for a random mini-batch D^t at time slot t, $e=(s,a,r,s')$, and $\hat{\boldsymbol{\theta}}$ represents the quasi-static 'target' parameters, which are updated to be equal to $\boldsymbol{\theta}$ once every C time slots [13]. Finally the optimal parameters $\boldsymbol{\theta}^{op}$ are:

$$\boldsymbol{\theta}^{op}=\arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \tag{16}$$

To use the DQN for solving the power allocation problem in CF massive MIMO, we define the duration of each time slot t as the time unit during which the large-scale fading stays constant, i.e., the duration of each time slot is τ_l in Fig. 2. The power allocation is done at the CC and on the large-scale fading time [1] (τ_l in Fig.2). As in [14] and [15], we define for each AP-UE link an agent, thus the power allocation is performed by a multi-agent system, where each agent contains a DQN. The agents interact with the environment to collect data (s^t,a^t,r^t,s^{t+1}) and store it in a dataset at the CC, then by the mini-batch sampling the DQN is trained by gradient descent algorithm, as shown in Fig. 3. There are in total NK agents in the whole system. During time slot t, each agent (n,k) allocates power from AP n to UE k. One should note that all agents use the same DQN parameters, i.e., after the DQN is trained by the experience of all agents, the DQN shares its parameters with all other agents to allocate power.

We define $e^t_{n,k}=(s^t_{n,k},a^t_{n,k},r^t_{n,k},s^{t+1}_{n,k},)$ as the experience sequence of agent (n,k) at time slot t. The DQN is trained by the dataset $D=\{e^1_{1,1},e^1_{1,2},\dots,e^t_{n,k}\dots\}$, which describes the agents' relation with their environment. The key to use the DQN for solving (7) is to model the decision variables as the action of agents. Obviously the normalized downlink transmission power $p_{n,k}$ is the decision variable for SE, therefore the action of agent (n,k) is $p_{n,k}$. As a time series we define $p^t_{n,k}$ as the action of agent (n,k) at time slot t. The agent (n,k) takes action according to the current state $s^t_{n,k}$, which features the independent variables. From (6) we find that the large-scale fading is the independent variable for SE, therefore the large-scale fading information is a key element for $s^t_{n,k}$. The objective function, which describes the target of the agents, can be defined as the reward in each time slot. Based on the above analysis, the elements of the experience $e^t_{n,k}$. for CF massive MIMO power allocation are defined as following:

1) STATE $s^t_{n,k}$

The signal-to-interference-plus-noise ratio (SINR) is the key element of the SE. The *signal* in SINR of UE k comes from the agent set $\{(1,k),(2,k),\dots,(N,k)\}$, while the *interference* in SINR of UE k comes from the agent set $\{(1,k'),(2,k'),\dots,(N,k')\}$, where $k'=1,2,\dots,K$ and $k'\neq k$. The agent (n,k) allocates the power considering the *signal* of its own link and the *interference* caused by other links. Therefore, the information $\beta^t_{n,k}$ and $\boldsymbol{\zeta_1}=\{\beta^t_{1,k'},\beta^t_{2,k'},\dots,\beta^t_{N,k'}\}$ are the components of the state $s^t_{n,k}$. Referring to [15], the information of power allocation and SE at time slot $t-1$ can improve the sum-SE performance of the DQN, therefore we also consider $p^{t-1}_{n,k}$, $\boldsymbol{\zeta_2}=\{p^{t-1}_{1,k'},p^{t-1}_{2,k'},\dots,p^{t-1}_{N,k'}\}$ and $\boldsymbol{\zeta_3}=\{SE^{t-1}_1,SE^{t-1}_2,\dots,SE^{t-1}_K\}$ as the components of the state $s^t_{n,k}$. Finally, $s^t_{n,k}$ is formalized as follows:

$$s^t_{n,k}=\{\beta^t_{n,k},\boldsymbol{\zeta_1},p^{t-1}_{n,k},\boldsymbol{\zeta_2},\boldsymbol{\zeta_3}\} \tag{17}$$

The size of $s^t_{n,k}$, i.e., the input dimension of the DQN, is $2N(K-1)+K+2$. To reduce the input dimension and the complexity, the elements in $\boldsymbol{\zeta_1}$ are sorted in decreasing order and only the first L components remain. The corresponding links of components are also reduced in $\boldsymbol{\zeta_2}$. Therefore, there are $2(1+L)+K$ components remaining in $s^t_{n,k}$.

2) ACTION $a^t_{n,k}$

The allocated power is a continuous value, while the agent's action is a discrete value. Therefore, we should discretize the transmission power as follows:

$$\boldsymbol{A}=\{0,\frac{p_{\max}}{|\boldsymbol{A}|-1},\frac{2p_{\max}}{|\boldsymbol{A}|-1},\dots,p_{\max}\} \tag{18}$$

where $|\boldsymbol{A}|$ represents the number of power levels.

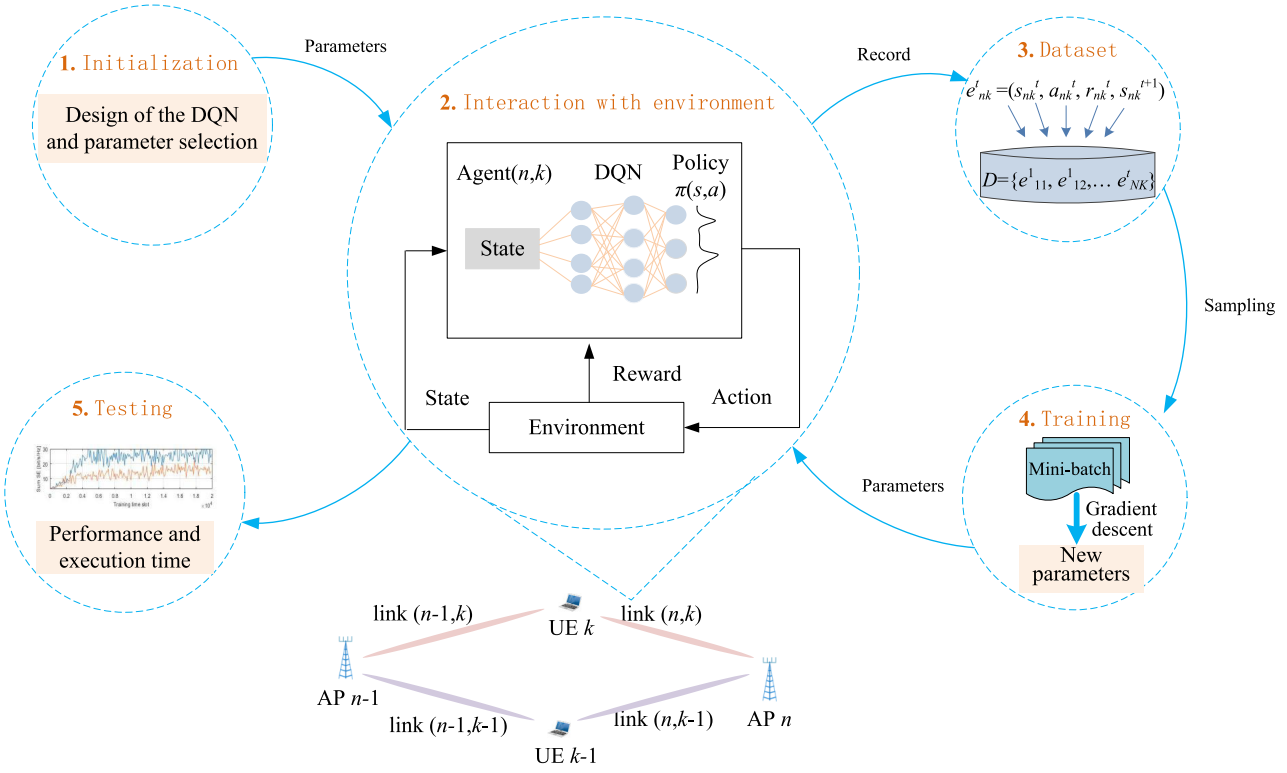


FIGURE 3. DQN-based power allocation method: a multi-agent system where agents interact with the environment to collect training data and each agent contains a DQN, which outputs the Q-value of the action. The optimal policy follows the maximization of the Q-value.

3) REWARD $r_{n,k}^t$

The target is to maximize the sum-SE. Therefore, the reward is the sum-SE at time slot t:

$$r_{n,k}^t = \sum_{k=1}^{K} SE_k^t \tag{19}$$

C. DDPG-BASED METHOD

For the DDPG-based method, a DNN is used as a policy network, that is, the output of the DNN is the action. DDPG is an actor-critic method [19]: an actor $Ac(\boldsymbol{\delta}_a)$ takes the action a by observing the state s, where $Ac(\boldsymbol{\delta}_a)$ is the policy network and $\boldsymbol{\delta}_a$ represents the parameters of this network. A critic $Cr(\boldsymbol{\delta}_c)$ evaluates the action a with the critic state, where $Cr(\boldsymbol{\delta}_c)$ is the critic network (a different DNN) and $\boldsymbol{\delta}_c$ represents the parameters of this network, see Fig. 4. The optimal policy by DDPG is:

$$op_{DDPG} = \arg \max_{Ac(\boldsymbol{\delta}_a^{op})} Cr(\boldsymbol{\delta}_c^{op}) \tag{20}$$

The actor $Ac(\boldsymbol{\delta}_a)$ and the critic $Cr(\boldsymbol{\delta}_c)$ work cooperatively to get the optimal parameters of $\boldsymbol{\delta}_a^{op}$ and $\boldsymbol{\delta}_c^{op}$. Similarly to the DQN-based method, a random mini-batch D^t is used to train $Ac(\boldsymbol{\delta}_a)$ and $Cr(\boldsymbol{\delta}_c)$. The loss functions are defined as follows:

$$L(\boldsymbol{\delta}_a) = Cr(\boldsymbol{\delta}_c)|_{a=Ac(\boldsymbol{\delta}_a)} \tag{21}$$

$$L(\boldsymbol{\delta}_c) = Cr(\boldsymbol{\delta}_c)|_{a=Ac(\boldsymbol{\delta}_a)} - r \tag{22}$$

Finally the optimal parameters $\boldsymbol{\delta}_a^{op}$ and $\boldsymbol{\delta}_c^{op}$ are:

$$\boldsymbol{\delta}_a^{op} = \arg \max_{\boldsymbol{\delta}_a} L(\boldsymbol{\delta}_a) \tag{23}$$

$$\boldsymbol{\delta}_c^{op} = \arg \min_{\boldsymbol{\delta}_c} L(\boldsymbol{\delta}_c) \tag{24}$$

(23) implies that the actor $Ac(\boldsymbol{\delta}_a)$ strives to get the maximum evaluation from the critic; (24) aims to get the precise evaluation.

Similarly to the DQN-based method, the DDPG-based method uses the multi-agent system to train $Ac(\boldsymbol{\delta}_a)$ and $Cr(\boldsymbol{\delta}_c)$. The state s and reward r are the same as in the DQN-based method. The only difference between the DQN-based method and the DDPG-based method is the action a. The output of the DQN is the action value of different actions and is determined by (14). The output of actor $Ac(\boldsymbol{\delta}_a)$ is

$$a_{n,k}^t = \left[Ac(\boldsymbol{\delta}_a)|_{s_{n,k}^t} \right]_0^{\sqrt{p_{max}}} \tag{25}$$

resulting in a continuous value, which is different from the action generating discrete values obtained by the DQN-based method in (18).

D. COMPLEXITY OF THE DRL-BASED POWER ALLOCATION METHODS

The computational complexity considers the operational phase, i.e., the time period that a trained DRL performs the

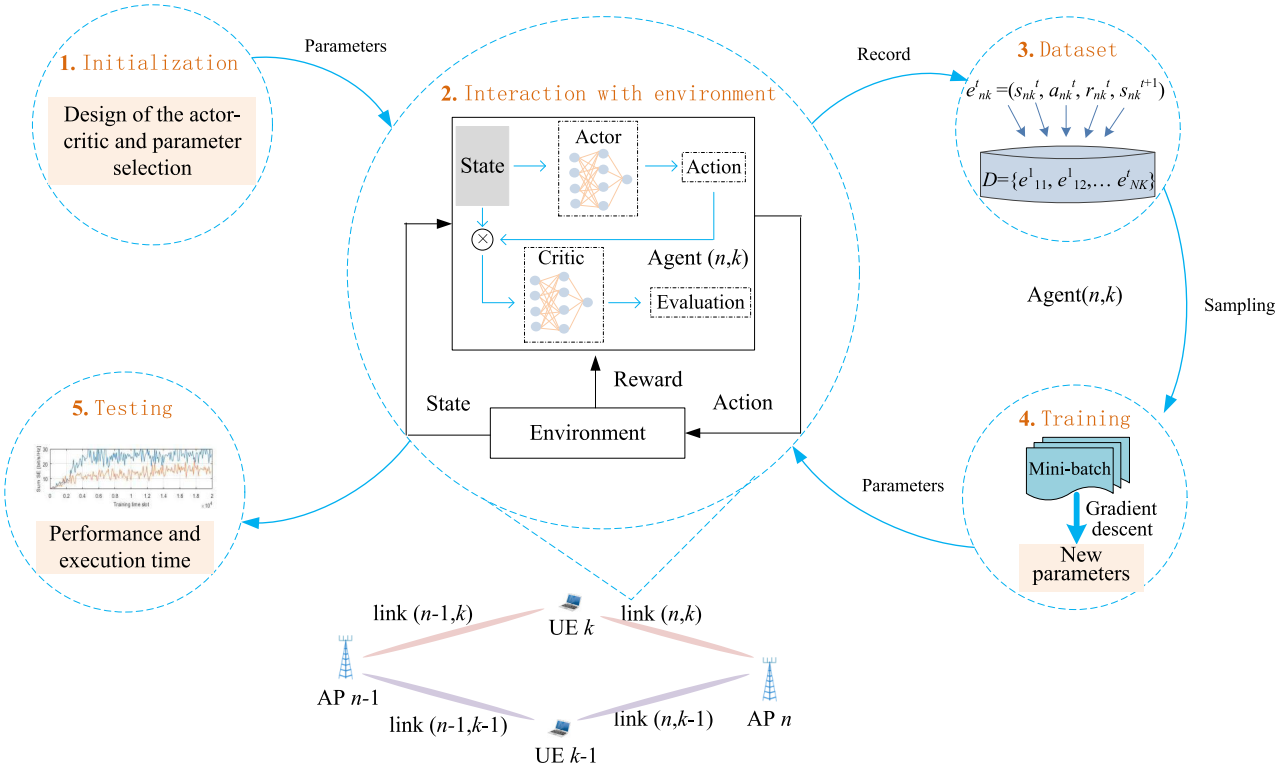


FIGURE 4. DDPG-based power allocation method: a multi-agent system where agents interact with the environment to collect training data and each agent contains an actor network and a critic network. The actor network outputs the action; the critic network evaluates the action. The optimal policy follows the action that has the maximum evaluation.

power allocation. Therefore the computational complexity of the DRL-based methods is only determined by the DNN. Typically, for a dense neural network, the complexity is $O(\nu\mu^2)$, where ν is the number of layers and μ is the number of neurons for the widest layer, i.e., the layer with the most neurons. The number of neurons in each layer depends on the dimension of the input layer, i.e., $O(\mu) = O(N + K + 1)$ in our case. The number of layers for a neural network is independent of the scale of the problem. Therefore, the computational complexity of DRL is $O(N^2 + K^2)$. Compared to the WMMSE algorithm $O(IN^2K^3)$ in Section III, the DRL-based power allocation methods have a much lower computational complexity. This implies less execution time on a given processing platform. It means that the DRL-methods will have no problem achieving a timely adaptation of the power allocation to the channel dynamics, which is not guaranteed when iterative algorithms such as WMMSE are used.

V. SIMULATION RESULTS

In this section we show by simulations that the DRL-based power allocation methods in CF massive MIMO are competitive in terms of performance and complexity, compared to the non-DL WMMSE algorithm.

A. SCENARIO

We consider the 3GPP TR 38.901 indoor mixed office scenario ($120m \times 50m \times 3m$) [21] with 12 APs, positioned as shown in Fig. 5. Each AP contains 10 antennas in a horizontally mounted and downward radiating 2×5 uniform planar array (UPA) at a height of 3m. We assume $K = 10$ single-antenna UEs moving within the coverage area. Each UE moves in a random direction (up, down, left, and right) with a randomly chosen velocity distributed uniformly between 0 and 1m/s. We consider a discrete time system where the duration of each time slot is 3 s, corresponding to 40 coherence times, as discussed in Section IV-B. The carrier frequency is 2 GHz, the bandwidth is 20 MHz. For a UE velocity of 1m/s, the channel coherence time τ_c is about

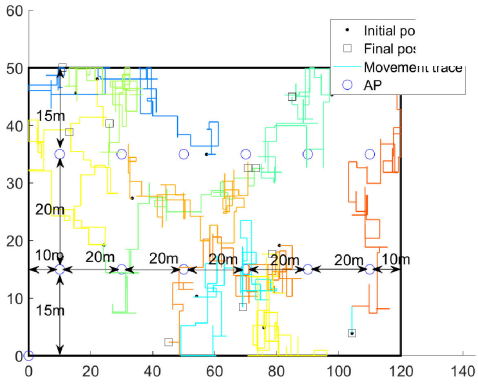


FIGURE 5. Example of UE movement traces in a 3GPP TR 38.901 scenario for 100 time slots.

75 ms, calculated as $\tau_c = \lambda/2v$, where λ is the wavelength and v is the velocity [24, Chapter 2]. Each UE maintains its speed and direction in each second before selecting a new speed and direction. The initial positions of the UEs at time $t = 0$, are uniformly distributed over the coverage area (Fig. 5). We model the large-scale fading as the combination of pathloss and shadowing, as in [21].

We set the coherence time to 200 modulation symbols as in [1]. The length of the uplink pilot is 5 symbols for channel estimation in our simulations. As discussed in Section II-A, when $\tau_p < K$ some pilot sequences are reused, hence the simulations take the pilot contamination into consideration. Other parameters used in the simulations are listed in Table 1.

TABLE 1. Parameters used in simulations.

Parameter	value
Coverage volume	120m×50m×3m
K, number of UEs	10
M, number of antennas per AP	10
N, number of APs	12
$p_{\max}$, maximum power constraint	13 dBm
p_p, pilot power	20 dBm
τ_c, length of coherence time in symbols	200
τ_p, length of pilot in symbols	5
Carrier frequency	2 GHz
Bandwidth	20 MHz
Noise power	-74dBm
Distribution of UE velocity	U(0, 1) m/s
Timeslot duration	3 s

B. HYPER-PARAMETER SELECTION

We next describe the hyper-parameters used for the DQN-based method and the DDPG-based method. We first tried three DNN candidate architectures to investigate the impact of the number of hidden layers and number of neurons per hidden layer on the training process of both the DQN-based method and the DDPG-based method.

For the DRL-based power allocation in cellular networks, [14] proposed a DNN using two fully-connected hidden layers with 128 and 64 neurons, respectively; [15] proposed a DNN using three fully-connected hidden layers with 200, 100 and 40 neurons, respectively. Based on this, we chose the candidate architectures of the DNNs as follows:

(1) DNN_1 with two fully-connected hidden layers, with 128 and 64 neurons, respectively;

(2) DNN_2 with two fully-connected hidden layers, with 256 and 128 neurons, respectively;

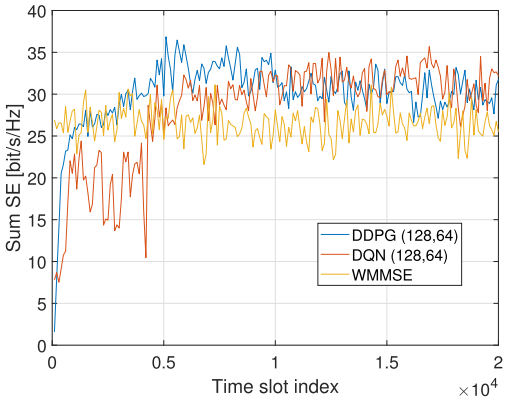
(3) DNN_3 with three fully-connected hidden layers, with 256, 128 and 64 neurons, respectively.

The number of neurons in the input layers for the DNNs are $2(1 + L) + K$ as we explained in the description of state $s^t_{n,k}$, see (17). We set $L = 20$ therefore there are 52 neurons in the input layer. The selection of L affects the number of interfering links for a given agent. The value of L is in the range $[0, NK - 1]$. A larger value of L makes the power allocation of the agent closer to the optimal solution but results in a higher complexity. Referring to [14] using $L = 16$ and [15] using $L = 5$, we chose $L = 20$ to gain a near-optimal power allocation.

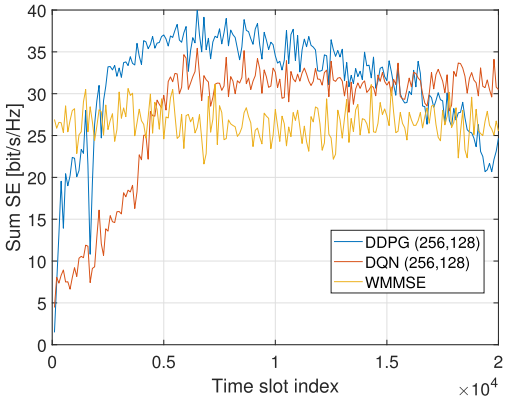
For the DQN-based method, the number of power levels is set to 10, hence the number of neurons in the output layer for $Q(s, a, \boldsymbol{\theta})$ is also 10. For the DDPG-based method, since the output of the actor network is an action, i.e., the allocated power, therefore there is one neuron in the output of $Ac(\boldsymbol{\delta}_a)$; the critic network produces an evaluation value of the action, therefore there is one neuron in the output of $Cr(\boldsymbol{\delta}_c)$. All the activation functions of the DNNs use the rectifier linear unit (ReLU), except for the output layer of $Ac(\boldsymbol{\delta}_a)$, which uses the Sigmoid function to obtain the normalized allocated power.

There are other hyper-parameters affecting the training process of the DQN-based method and the DDPG-based method, namely, the discount factor ω, the training interval C, the initial adaptive learning rate lr, the adaptive ϵ-greedy algorithm and the mini-batch size $|D^t|$. Adaptive learning means that the learning rate decays with the number of training time slots. Generally, a large learning rate allows the model to learn faster but may end up with a sub-optimal final set of weights. A smaller learning rate may allow the model to learn a more optimal or even globally optimal set of weights but may take significantly longer. Adaptive learning balances the training time and performance. The ϵ-greedy algorithm is a learning method that makes use of the exploration-exploitation tradeoff, in which the agent takes a random action with probability ϵ or takes an action using the policy of DQN or DDPG with probability $1 - \epsilon$. A random action may lead the training 'jumps' out of a local optimum and explores new convergence regions. In the adaptive ϵ-greedy algorithm the value of ϵ decays each training time slot. A large ϵ avoids the training ending up in local optima during the initial training time slots, a small value of ϵ makes sure that the training will converge in the later training time slots.

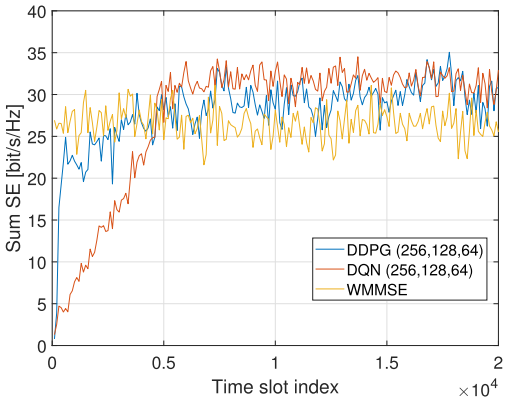
Referring to the literature, we choose:
$\omega \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ [13], [14];
$C \in \{10, 50, 100, 200, 500\}$ [13], [19];
$lr \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$ [5], [7];
$\epsilon \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ [14], [15];
$|D^t| \in \{500, 1000, 2000, 5000, 10000\}$ [5]
to find the optimal hyper-parameters. We found that for different values of ω, C and $|D^t|$, the behavior of the sum-SE as a function of the training time-slot is very similar, implying our proposed DRL-based methods are not very sensitive to these hyper-parameters. However, for different values of lr and ϵ, we get very different behaviors of the sum-SE as a function of the training time-slot. This is because the learning rate and ϵ-greedy algorithm affect the exploration-exploitation tradeoff, which is important for the updating of the DNN parameters, i.e., weights and bias. We chose the parameters $\omega = 0.1, C = 10, lr = 0.005, \epsilon = 0.1$ and $|D^t| = 500$, which gave us the highest sum-SE values, in our simulations.



(a) Training process of DQN and DDPG with DNN_1.



(b) Training process of DQN and DDPG with DNN_2.



(c) Training process of DQN and DDPG with DNN_3.

FIGURE 6. Sum-SE as a function of time during the training of the DQN-based method and the DDPG-based method for three candidate architectures of DNN.

Fig. 6 shows the training process, i.e., the sum-SE as a function of the training time slot, where the DQN and DDPG use the three DNN candidate architectures. The length of the training period we chose is 20,000 time slots, corresponding to roughly 16.6 hours. It was determined by the observation, by the naked eye, of the time it takes for the sum-SE to converge, i.e., a longer training period does not result in fluctuations around a significantly different value of sum-SE. The fluctuations of the sum-SE in Fig. 6 are caused by the random mobility of the UEs.

Fig. 6(a) shows the training process of DQN and DDPG for the DNN_1 architecture: the DQN-based method and the DDPG-based method converge to fluctuations around 30 bit/s/Hz during 5,000 training time slots. It is obvious that the DDPG-based method converges faster than the DQN-based method, by observing that the blue line is always higher than the red line during most of the training time slots between 0 and 5,000. Afterwards, the DQN-based method and the DDPG-based method achieve similar sum-SE values, that are higher than the values obtained by the WMMSE algorithm which is shown as a reference.

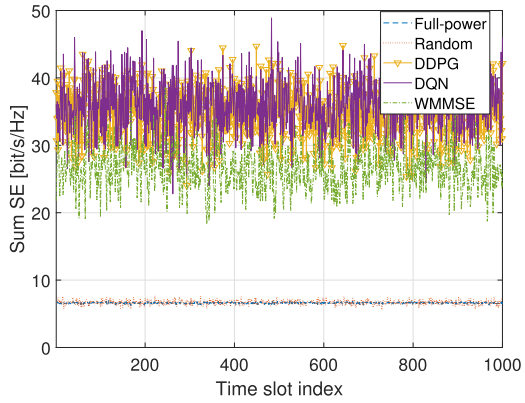
Fig. 6(b) shows the training process of DQN and DDPG for the DNN_2 architecture: the DQN-based method converges to fluctuations around 32 bit/s/Hz during 5,000 training time slots, achieving higher values than the WMMSE algorithm. For the DDPG-based method, it converges to fluctuations around almost 40 bit/s/Hz through 7,000 training time slots. Afterwards however, the sum-SE decreases at around the 15,000th time slot. Moreover, the sum-SE values of the DDPG-based method become even lower than the WMMSE algorithm.

Fig. 6(c) shows the training process of DQN and DDPG for the DNN_3 architecture: similar observations can be made as for Fig. 6(a), the DQN-based method and the DDPG-based method converge to fluctuations around 32 bit/s/Hz through 5,000 training time slots. Afterwards, the DQN-based method and the DDPG-based method achieve similar sum-SE values, higher than the WMMSE algorithm. In addition, the DDPG-based method converges faster than the DQN-based method.

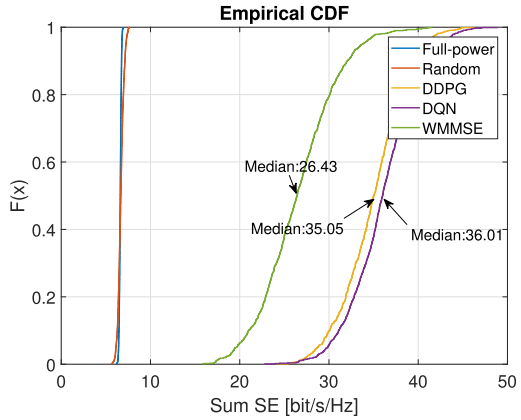
Based on the above observations, we choose DNN_1 as the architecture for the DQN-based method and the DDPG-based method for the following reasons: (1) the DDPG-based method have higher sum-SE values for DNN_1 than for DNN_2, (2) DNN_1 achieves the similar performance as DNN_3, which has one more layer implying more execution time.

C. SUM-SE PERFORMANCE

To evaluate the performance of the DQN-based method and the DDPG-based method, we have used three benchmark algorithms. The first benchmark is the WMMSE algorithm. The second is random power allocation where $p_{n,k} \sim U(0, p_{\max})$ for all n and k. The third one is full-power allocation, i.e., $p_{n,k} = p_{\max}$ for all n and k. For the DQN-based method and the DDPG-based method we used the DNN_1 architecture. The training period is 20,000 time slots. After training we run the systems for 1,000 time slots, during which we record the sum-SE values obtained by the five power allocation methods, as shown in Fig. 7. We chose 1000 time slots for testing and comparing our proposed DRL-based methods, based on the fact that in [14], for the same purpose,



(a) Sum-SE as a function of time slot.



(b) CDF of the sum-SE.

FIGURE 7. Comparison of the sum-SE over 1,000 time slots with $N = 12$, $K = 10$ and $M = 10$.

500 time slots was estimated to be sufficient to have a reliable comparison.

Fig. 7(a) shows the sum-SE over a period of 1,000 time slots, Fig. 7(b) shows the empirical CDF of the sum-SE. As expected, the DQN-based method and the DDPG-based method have better performance than the three benchmark algorithms. The random mobility of UEs causes the fluctuations of the sum-SE for the above methods in Fig. 7(a). Although the differences are not pronounced in Fig. 7(a), it is clear that the DQN-based method slightly outperforms the DDPG-based method, by observing that the purple line is to the right of the yellow line in Fig. 7(b). Nevertheless, we can conclude that the DQN-based method and the DDPG-based method achieve a similar performance, which results in fluctuations around a value about 10bit/s/Hz higher than the WMMSE algorithm.

To explore the robustness of the DRL-based power allocation methods, DQN and DDPG, with respect to the traffic assumptions, we ran some experiments with different values of N, K and M. We first set $N = 8$ while keeping $K = 10$ and $M = 10$. The APs are positioned as shown in Fig. 8. The UE mobility model is kept the same as in Fig. 5. Fig. 9 shows the simulation result over a period of 1000 time slots.

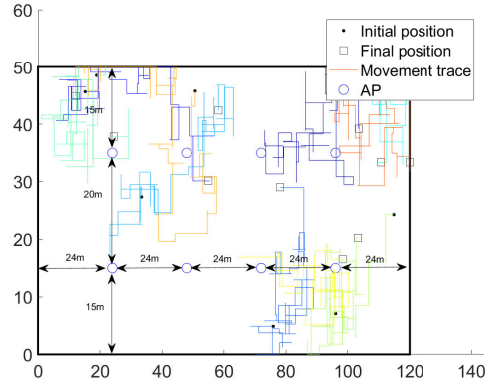
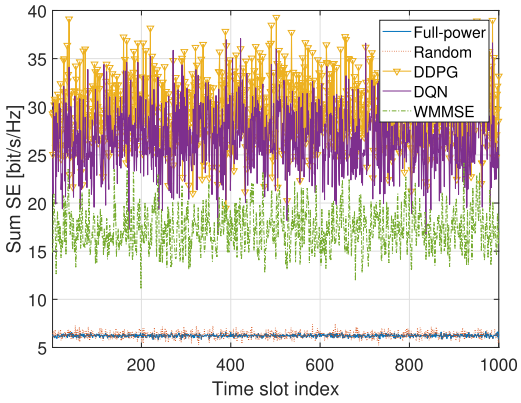
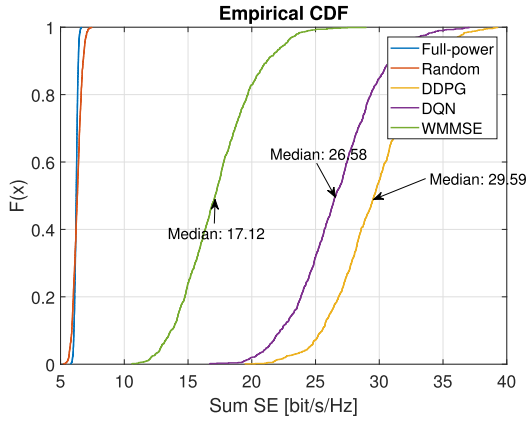


FIGURE 8. Example of UE movement traces in a 3GPP TR 38.901 scenario for 100 time slots with $N = 8$ APs.



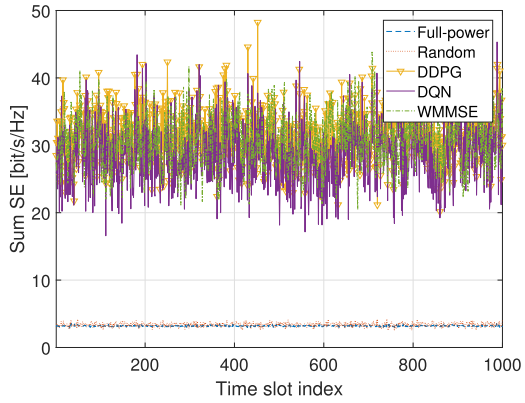
(a) Sum-SE as a function of time slot.



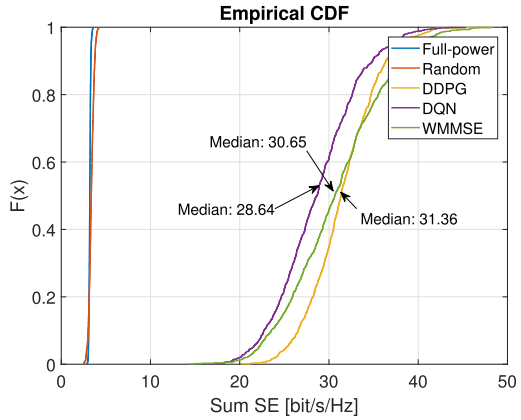
(b) CDF of the sum-SE.

FIGURE 9. Comparison of the sum-SE over 1,000 time slots with $N = 8$, $K = 10$ and $M = 10$.

Observe from Fig. 9(a) that DQN and DDPG still have much better performance than the WMMSE algorithm. The DQN-based method has a sum-SE average that is 9 bit/s/Hz higher than the value for the WMMSE algorithm. For DDPG the difference is 13 bit/s/Hz. It is also clear that DDPG works better than DQN, by observing that the yellow line is to the right of the purple line in Fig. 9(b).

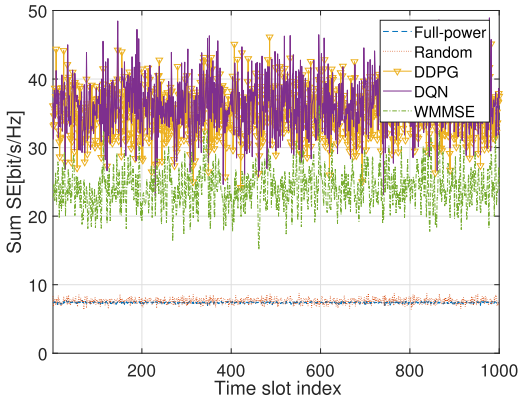


(a) Sum-SE as a function of time slot.

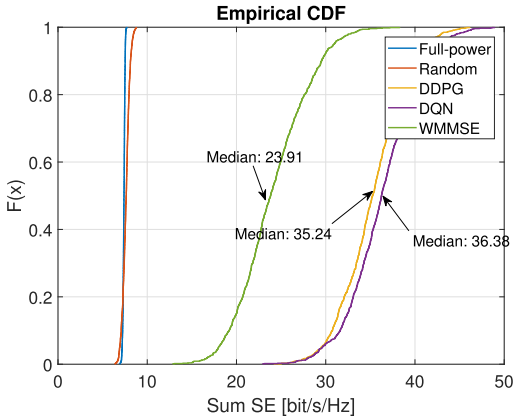


(b) CDF of the sum-SE.

FIGURE 10. Comparison of the sum-SE over 1,000 time slots with $N = 12$, $K = 5$ and $M = 10$.



(a) Sum-SE as a function of time slot.



(b) CDF of the sum-SE.

FIGURE 11. Comparison of the sum-SE over 1,000 time slots with $N = 12$, $K = 10$ and $M = 15$.

Next we set $K = 5$ while keeping $N = 12$ and $M = 10$. Fig. 10 shows the simulation result over a period of 1000 time slots. Observe from Fig. 10(a), we find that the DQN-based method, the DDPG-based method and the WMMSE algorithm achieve similar sum-SE performance. However, the empirical CDF in Fig. 10(b) shows that the DQN-based method has lower sum-SE than the WMMSE algorithm and DDPG-based method. From Fig. 10 we conclude that the DRL-based power allocation methods are sensitive to the number of UEs. This implies that, when the number of UEs changes, the DQN- and DDPG-based methods are expected to be continuously trained to get better performance. The length of the training process depends on the convergence of the sum-SE.

Finally, we set $M = 15$ while keeping $N = 12$ and $K = 10$. Fig. 11 shows the simulation result over a period of 1000 time slots. The result in Fig. 11 is very similar to Fig. 7. Because the power allocation in CF massive MIMO is optimized on the basis of the AP-UE links, which does not change when the number of antennas per AP is varied, so the number of antennas per AP does not affect the power allocation. Therefore, when M changes, the power allocation solution still holds.

D. COMPUTATIONAL COMPLEXITY COMPARISON

To get an indication of the difference in computational complexity of the DRL-based methods and the WMMSE algorithm, we measured the execution time in each of the 1000 time slots. We ran the algorithms on a personal laptop with CPU Intel i5-7300. The programs are coded in Python 3.7.2. Table. 2 shows the statistical characteristics of the execution time for a simulation over 1000 time slots.

TABLE 2. Execution time of the DRL-based power methods and the WMMSE algorithm (in ms).

Method	Mean	Max	Min	Standard deviation
DQN	0.66	0.97	0.52	0.03
DDPG	0.63	0.99	0.51	0.04
WMMSE	621.23	759.63	592.16	16.35

From Table. 2, it is obvious that the DQN-based method and the DDPG-based methods require much less processing time than the WMMSE algorithm and have less variation. For the DQN-based method and the DDPG-based method, the number of calculations is constant, as the number of neurons and layers does not change over 1000 time slots. Observe that there are still some slight fluctuations of execution time for DQN and DDPG, which come from the calculation of

different floating-point numbers and the inaccuracy of reading the system time. For the WMMSE algorithm, the time fluctuation mainly comes from different initializations, i.e., a different initial point of the algorithm can make a large difference in the time needed to find the optimum.

VI. CONCLUSION

In this paper, we studied the use of two DRL-based power allocation methods, namely the DQN and the DDPG, in microwave CF massive MIMO systems with mobile UEs, with the objective of maximizing the sum-SE in the downlink. Our models take into account the imperfect CSI and uplink pilot contamination. Unlike supervised learning that needs a huge training data set, the DRL-based methods are trained by interacting with the environment. The objective function is directly defined as the reward to train the neural network. We found that for the scenario we considered, the performance (in terms of sum-SE) of the DRL-based methods are competitive with the popular and well-adopted WMMSE algorithm. For the same configuration and scenario, the well-trained DRL-based methods achieved an around 38% higher sum-SE than the WMMSE algorithm. For the different configuration of scenarios, the well-trained DRL-based methods still achieved an average 33% higher sum-SE than the WMMSE algorithm. In addition, the computational complexity of the DRL-based power allocation methods is significantly lower than the WMMSE algorithm. On our simulation platform, only 0.1% of the execution time of the WMMSE algorithm is needed. There are still some issues that should be addressed in further studies:

1. Our results are based on a particular 3GPP indoor mixed office scenario and a particular user mobility model. Different scenarios and mobility models should be considered in further studies.

2. Since we were focusing on determining the merits and drawbacks of the power allocation methods, we used a uniform Rayleigh model for all channels. Different and non-uniform channel models should be considered in further studies to assess the performance gains for a wider set of cases.

3. Our focus was on maximizing the sum-SE. One should also look at different objectives, e.g., max-min power control which maximizes the minimum SE achieved at each UE, aiming at a form of fairness.

REFERENCES

[1] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[2] T. C. Mai, H. Q. Ngo, and T. Q. Duong, "Downlink spectral efficiency of cell-free massive MIMO systems with multi-antenna users," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4803–4815, Aug. 2020.

[3] Z. Chen and E. Björnson, "Channel hardening and favorable propagation in cell-free massive MIMO with stochastic geometry," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5205–5219, Nov. 2018.

[4] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Found. Trends Signals Process.*, vol. 11, nos. 3–4, pp. 154–655, 2017.

[5] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.

[6] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] E. Bjornson and P. Giselsson, "Two applications of deep learning in the physical layer of communication systems," *IEEE Signal Process. Mag.*, vol. 37, no. 5, pp. 134–140, Sep. 2020.

[9] T. Van Chien, T. Nguyen Canh, E. Björnson, and E. G. Larsson, "Power control in cellular massive MIMO with varying user activity: A deep learning solution," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5732–5748, Sep. 2020.

[10] L. Sanguinetti, A. Zappone, and M. Debbah, "Deep learning power allocation in massive MIMO," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Oct. 2018, pp. 1257–1261.

[11] M. Bashar, A. Akbari, K. Cumanan, H. Q. Ngo, A. G. Burr, P. Xiao, M. Debbah, and J. Kittler, "Exploiting deep learning in limited-fronthaul cell-free massive MIMO uplink," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1678–1697, Aug. 2020.

[12] Y. Zhao, I. G. Niemegeers, and S. H. D. Groot, "Power allocation in cell-free massive MIMO: A deep learning method," *IEEE Access*, vol. 8, pp. 87185–87200, 2020.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 56, no. 518, pp. 529–533, 2015.

[14] F. Meng, P. Chen, and L. Wu, "Power allocation in multi-user cellular networks with deep *Q* learning approach," in *Proc. ICC*, Shanghai, China, May 2019, pp. 1–6.

[15] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

[16] F. Meng, P. Chen, L. Wu, and J. Cheng, "Power allocation in multi-user cellular networks: Deep reinforcement learning approaches," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6255–6267, Oct. 2020.

[17] U. F. Siddiqi, S. M. Sait, and M. Uysal, "Deep reinforcement based power allocation for the max-min optimization in non-orthogonal multiple access," *IEEE Access*, vol. 8, pp. 211235–211247, 2020.

[18] D. Guo, L. Tang, X. Zhang, and Y.-C. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13124–13138, Nov. 2020.

[19] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*. [Online]. Available: http://arxiv.org/abs/1509.02971

[20] A. A. Polegre, F. Riera-Palou, G. Femenias, and A. G. Armada, "New insights on channel hardening in cell-free massive MIMO networks," in *Proc. ICC Workshops*, Dublin, Ireland, Jun. 2020, pp. 1–7.

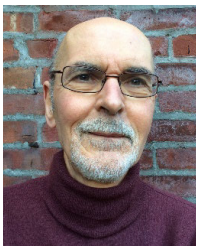
[21] *Study on Channel Model for Frequencies From 0.5 to 100 GHz (Release 15)*, document 3GPP TR 38.901, 3GPP Radio Access Network Working Group, 2018.

[22] E. N. Barron and H. Ishi, "The Bellman equation for minimizing the maximum cost," *Nonlinear Anal., Theory, Methods Appl.*, vol. 13, no. 9, pp. 1067–1090, 1989.

[23] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 661–670.

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

YU ZHAO (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in control science and engineering from Xi'an Jiaotong University, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands. His current research interests include massive MIMO, deep learning, and artificial intelligent algorithms.



IGNAS G. NIEMEGEERS received the M.Sc. degree in electrical engineering from the University of Gent, Belgium, in 1970, and the M.Sc. and Ph.D. degrees in computer engineering from Purdue University, USA, in 1972 and 1978, respectively. From 1978 to 1981, he was a Designer of one of the first commercial packet switching networks with Bell Telephone Mfg Company, Antwerp, Belgium. From 1981 to 2002, he was a Professor with the University of Twente, The Netherlands. From 1995 to 2002, he was the Scientific Director of the Centre for Telematics and Information Technology (CTIT), University of Twente. From 2002 to 2012, he chaired the Telecommunications Department, Delft University of Technology. Since August 2012, he has been an Emeritus Professor with the Delft University of Technology, and an Advisor of the Centre for Wireless Technology, Eindhoven University of Technology, The Netherlands. Since March 2016, he has been a part-time Guest Professor in future optically-supported wireless network with the Eindhoven University of Technology. He was involved in many European research projects and reviewer for many projects. He has supervised 42 Ph.D. students and has authored or coauthored around 300 articles. His research interests include machine learning in radio access network architectures, radio-over-fiber, mmWave technologies, energy-harvesting IoT, and V2X communication.

SONIA M. HEEMSTRA DE GROOT received the M.Sc. degree in electrical engineering from the Universidad Nacional de Mar del Plata, Argentina, and the Philips International Institute/NUFFIC, The Netherlands, and the Ph.D. degree in electrical engineering from the University of Twente, The Netherlands, in 1990. She was an Assistant Professor and an Associate Professor with the University of Twente. She was a Full Professor of personal and ambient networking with the Delft University of Technology. After having worked some years as a Senior Researcher with Ericsson EuroLab, The Netherlands, she co-founded the Twente Institute for Wireless and Mobile, where she was the Chief Scientist, from 2003 to 2014. Since 2012, she has been a Full Professor with the Eindhoven University of Technology, where she holds the part-time Chair in heterogeneous network architectures. She became the Director of the Center for Wireless Technology Eindhoven, in September 2016. Her research interests include wireless and mobile communications, 5G, vehicular networks, wireless indoor communications, the Internet of Things, and wireless security.