Power Allocation in Dual Connectivity Networks Based on Actor-Critic Deep Reinforcement Learning

Elham Moein*, Ramin Hasibi*, Matin Shokri†, and Mehdi Rasti*
*Department of Computer Engineering and Information Technology
Amirkabir University of Technology
{moein.e, r.hasibi.94, rasti}@aut.ac.ir
†Department of Electrical and Computer Engineering
K. N. Toosi University of Technology
{shokri}@email.kntu.ac.ir

***Abstract*—Dual Connectivity (DC) has been proposed by Third Generation Partnership Project (3GPP), in order to address the small coverage areas and outage of users and improve the mobility robustness and rate of users in Heterogeneous Networks (HetNets). In the HetNet with DC, each user is assigned a Macro eNode Base Station (MeNB) and a Small eNode Base Station (SeNB) and transmits data to both eNode Base Stations (eNBs), simultaneously. In this paper, we present a power splitting scheme for the HetNet with DC; to maximize the total rate of the users while not exceeding the maximum transmit power of each user. In our proposed power splitting scheme, a Deep Reinforcement Learning (DRL) approach is taken based on the actor-critic model on continuous state-action spaces. Simulation results demonstrate that our power splitting scheme outperforms the baseline approaches in terms of total rate of users and fairness.**

***Index Terms*—Dual connectivity, heterogeneous networks, power allocation, deep reinforcement learning**

I. Introduction

Recently, with the increase of mobile users' demands, there has been an enormous growth of load in cellular networks. Therefore, in order to increase the capacity of cellular networks, several solutions have been proposed. One approach is cell densification i.e., deploying Small eNode Base Stations (SeNBs) in order to offload the load from Macro eNode Base Stations (MeNBs). The SeNBs have smaller coverage and lower transmit power compared to MeNBs and can provide service for the users in their proximity [1]. However, offloading the users to SeNBs can have some drawbacks such as increase in control overhead and outage of mobile users [2]. These flaws can be addressed by Dual Connectivity (DC) Heterogeneous Networks (HetNets). As shown in Fig. 1, in this type of network a user can be assigned to both an SeNB and an MeNB in micro and macro tiers, respectively, and transmit data to both of them, simultaneously [3]. This approach has several advantages compared to traditional HetNets such as increased frequency spectrum, spatial diversity [4], achieving wider bandwidth, and robust management of users [5].

One of the key research challenges in DC HetNet is Radio Resource Allocation (RRA) e.g., power control. The traditional

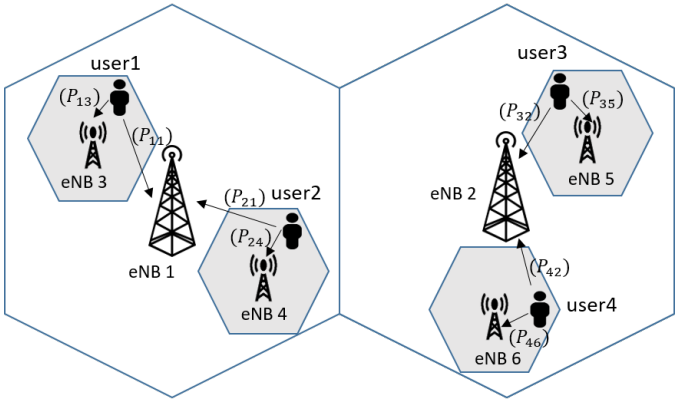

Fig. 1. A dual connectivity HetNet

RRA approaches in HetNets are not usually as efficient in DC HetNets since in these networks, users are assigned to two eNode Base Stations (eNBs) at the same time and are able to take advantage of radio resources provided by both eNBs. One of the main issues discussed in DC HetNets in the uplink is the problem of user power allocation. If the allocated power for transmitting to each eNB is not large enough, the total rate in the network will not reach its maximum potential; on the other hand, if the allocated power for transmitting to each eNB exceeds a certain level, the total rate of the network is susceptible to degradation due to high levels of interference. As a result, the power of users should be assigned to maximizes the throughput of the DC HetNet while not exceeding the maximum power constraint of each user [3].

There have been several schemes proposed to allocate transmit power of each user for transmitting to MeNB and SeNB. One approach is power splitting [6], [7] which is a direct approach that guarantees the maximum power of users will not be exceeded. However, the main drawback of this scheme is the fact that the redundant power for transmitting to one eNB is not sufficiently used in the other eNB. In [8], a solution is proposed to overcome this drawback which

aims to utilize the redundant power of a vacant eNB by another eNB. While this improves the power utilization, it does not consider the interference caused by excessive use of the power, and therefore the total rate of the network is prone to degradation. In [3], an uplink power control scheme has been suggested in order to enhance the system performance by considering system traffic demand and Interference of Thermal (IoT). The flaw of this approach arises from the fact that the redundant power not utilized in transmission to one eNB, is not reused for another eNB, leading to a decrease rate in certain circumstances.

Reinforcement Learning (RL) is a category of learning algorithms that involves an agent that can interact with an environment and receive certain rewards for every action it takes. The mentioned agent gradually learns how to take the best set of actions in order to reach an optimal solution [9]. The main drawback of traditional RL methods such as Q-learning is that the feature extraction must be done by an expert. Thus, if this procedure is not performed sufficiently well, the algorithm may not obtain the best possible results. Additionally, these algorithms are also prone to slow convergence in continuous state-action spaces. This category of solutions have been deployed in many works for power allocation. In [10], the use of cooperative Q-learning has been proposed for the users in order to learn the optimal answer considering a Quality of Service (Qos) for every user in their reward function. In [11], the authors have suggested a power allocation scheme using Deep RL (DRL) architecture called Deep Q-Network (DQN) in which the power levels are learned using a distributed approach. This method is extended in [12] by investigating the case of multiple users.

In all the aforementioned RL works, the action space is discrete which is not efficient in real world situations requiring continuous actions. Additionally, so far, the problem of power allocation is discussed in single connectivity (SC) HetNets. In addressing both of these shortcomings, in this paper we propose a deep reinforcement learning scheme in order to solve the problem of uplink user power allocation in DC HetNets using continuous state action spaces and evaluate the outcome of this scheme by comparing its performance with baseline methods.

This paper is organized as follows. In section II the preliminary concepts of this paper are presented. In section III, the system model and the optimization problem are formulated. The simulation results and conclusion are illustrated in section IV and section V, respectively.

II. PRELIMINARIES

In this section we explain the concepts of RL, deep learning, and DRL.

A. Reinforcement Learning

RL is one of the methods that is widely used in order to estimate optimal policies in model-less environments. These methods introduce techniques through which a learning agent tries to find the best actions in every state of the environment and obtains the optimal policy that gains the maximum reward. Q-learning is one of the most popular methods in RL in which the learning agent updates its action-value function of the current state-action pair according to the action-value function of the next state-action pair. This value is called Q-value and is updated as

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha^{'}[r^{'}_{t+1} + \gamma^{'} \max_a Q(S_{t+1}, a) - Q(S_t, A_t)], \quad (1)$$

in which S_t and A_t denote state and action at time step t, respectively. Furthermore, $r^{'}_{t+1}$ is the reward value at time step $t+1$. In addition, $\alpha^{'}$ and $\gamma^{'}$ are used as the learning rate and the discount factor, respectively [9].

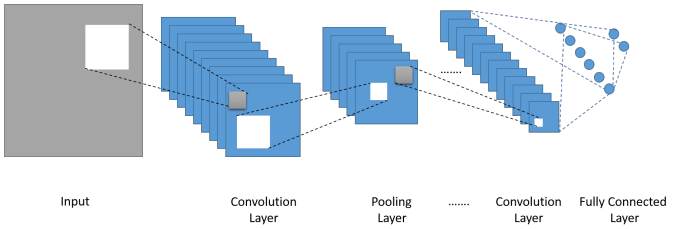

Fig. 2. Different parts of a CNN

B. Deep Learning

Deep learning is used to enable computers to learn from experience and comprehend complex concepts that are defined through their relation to simpler concepts. Supervised learning is heavily supported by deep learning as a powerful framework. In this framework, a high complexity can be represented by adding more layers and more units within a layer in a deep neural network [13].

Deep learning solutions usually consist of two major steps:

- **Automatic feature extraction:** This step aims to extract features and hand them to the classification in order to make proper decisions.
- **Classification or regression:** This step makes decisions about the inputs' class or the output value based on the features of the previous step.

A state of the art deep learning architecture is Convolutional Neural Network (CNN). The main purpose of these types of networks is to solve a supervised classification problem, but these methods have also been proven effective in regression tasks. Three main components of this architecture are:

- **Convolutional layer:** This layer comprises a set of learnable filters that are responsible for extracting spatial features from the input.
- **Pooling layer:** This layer performs down-sampling operations along the spatial dimensions.
- **Fully connected layer:** This layer is responsible for deciding the class of the input features [14].

Fig. 2 illustrates the different parts of a deep CNN and how these components interact with one another.

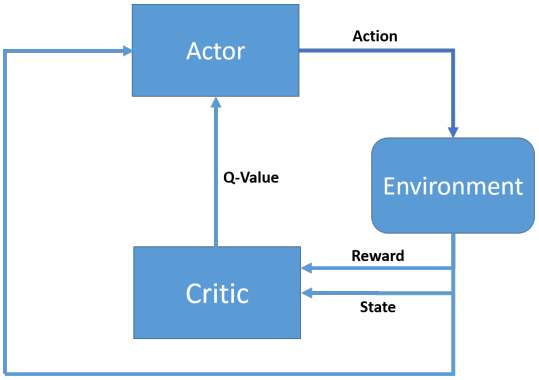

Fig. 3. Actor-critic model

C. Deep Reinforcement Learning

When the number features in an environment becomes too large for traditional RL methods to estimate the Q-value, a subset of RL algorithms called DRL is used. Although several approximation estimators have been introduced for these type of problems, all of these estimators need an expert in order to define the value function as a linear function of the environment features, which is subject to divergence in the case of inappropriate features [9]. Therefore, we can benefit from the automatic feature extraction of deep learning architectures to prevent this issue [15].

Many problems in real world situations require continuous and high-dimensional action spaces. Traditional DRL methods are not able to find the actions that maximize the action-value function in continuous domains. An obvious solution to this problem is to discretize the action space. However, the drawback of this approach is exponential increase in dimension when increasing the degree of freedom i.e., curse of dimensionality. In order to overcome this issue, actor-critic based models have been introduced. In this type of models, two functions named actor and critic are responsible for estimating the continuous action and the Q-value, respectively. Actor receives the state and decides the action to be taken and critic receives this action with the current state and decides whether this particular action is best suited for the agent. In DRL, the two functions that serve as actor and critic are one of the deep neural network architectures e.g., CNN, Long Short Term Memory (LSTM) networks, etc. In this approach, the actor network models the action prediction task as a regression problem, thus, it does not suffer from the curse of dimensionality. [16]. Fig. 3 depicts the actor-critic components.

III. System Model and Problem Formulation

In this section we present our system model and formally state the optimization problem.

A. System Model

In this paper, a two tier DC HetNet is considered which comprises M MeNBs, S SeNBs, and N users. The bandwidth is orthogonally divided between SeNB and MeNB, which results in zero inter-tier interference. The sets of MeNBs, SeNBs, and users are denoted by $\mathcal{M} = \{1, 2, ..., M\}$, $\mathcal{S} = \{M+1, M+2, ..., M+S\}$, and $\mathcal{N} = \{1, 2, ..., N\}$, respectively. Each MeNB is located at the center of every cell. Furthermore, SeNBs and users are randomly distributed in the DC network.

We assume that every user is assigned to its corresponding MeNB and SeNB based on pathloss. Thus, the eNB assignment is already performed. Let $x_{ij} \in \{0, 1\}$ denote the set of assignments of users to eNBs where x_{ij} is set to 1, if user i is assigned to eNB j, otherwise it is set to 0. Additionally, it is assumed that

$$\begin{aligned} \sum_{j \in \mathcal{M}} x_{ij} = 1 \quad \forall i \in \mathcal{N}, \\ \sum_{j \in \mathcal{S}} x_{ij} = 1 \quad \forall i \in \mathcal{N}. \end{aligned} \tag{2}$$

Parameters P_{ij} and L_{ij} denote the transmit power of user i while transmitting to eNB j and the pathloss between user i and eNB j, respectively. The interference of each user on each eNB is calculated as

$$I_{ij} = \begin{cases} P_{ij} L_{ij}^{-1} & \text{if} \quad x_{ij} = 0 \\ 0 & \text{else if} \quad x_{ij} = 1. \end{cases} \tag{3}$$

The received signal-to-interference-plus-noise-ratio (SINR) of MeNB j due to the transmission power of user i is determined as

$$\gamma_{ij} = \frac{P_{ij} L_{ij}^{-1}}{\sum_{k \in \mathcal{N}} I_{kj} + \sigma_j} \quad \forall i \in \mathcal{N}, j \in \mathcal{M}, \tag{4}$$

where σ_j represent noise power at MeNB j which is calculated by $\sigma_j = W_j n_j$, where W_j and n_j denote the bandwidth and power density at MeNB j, respectively.

Likewise, the SINR of SeNB j as a result of the transmission power of the user i is given by

$$\gamma_{ij} = \frac{P_{ij} L_{ij}^{-1}}{\sum_{k \in \mathcal{N}} I_{kj} + \sigma_j} \quad \forall i \in \mathcal{N}, j \in \mathcal{S}, \tag{5}$$

where σ_j shows the noise power in SeNB j which is obtained through $\sigma_j = W_j n_j$. Additionally, W_j and n_j represent the bandwidth and power density of SeNB j, respectively.

According to Shannon's theory, the data rate of user i to eNB j is calculated as

$$r_{ij} = x_{ij} W_j log_2(1 + \gamma_{ij}) \quad \forall i \in \mathcal{N}, j \in \mathcal{M} \cup \mathcal{S}, \tag{6}$$

Moreover, the total rate of user i while transmitting to eNB j is obtained through

$$R_i = \sum_{j \in \mathcal{M} \cup \mathcal{S}} r_{ij} \quad \forall i \in \mathcal{N} \tag{7}$$

The set of total rates of all the users is expressed as

$$\mathcal{R} = \bigcup_{i=1}^{N} R_i. \tag{8}$$

Additionally, the set of interferences of all users on the eNBs is denoted by

$$\mathcal{I} = \bigcup_{i=1,j=1}^{i=N,j=M+S} \{I_{ij}\} \tag{9}$$

The set of powers allocated to user i for transmitting to eNB j is denoted by

$$P_i = \{\sum_{j\in\mathcal{M}} x_{ij}P_{ij}, \sum_{j\in\mathcal{S}} x_{ij}P_{ij}\} \quad \forall i \in \mathcal{N}. \tag{10}$$

In addition, we assume that all the users have an equal constant maximum transmit power that we aim to split between MeNB and SeNB in the uplink.

B. Power Splitting

According to Third Generation Partnership Project (3GPP) [17], the transmit power of users for the Physical Uplink Shared Channel (PUSH) is calculated as follows

$$\begin{aligned} &P_{PUSCH} = \\ &\min\{P_{max}, 10log_{10}RB + P_0 + \alpha P_L + \Delta_{TF} + f(i)\}, \end{aligned} \tag{11}$$

where P_{max} and RB represent the maximum transmit power of users and the number of resource blocks assigned to each user, respectively. Additionally, P_0 and P_L express the power offset that controls the SINR target and pathloss from a given user to its assigned eNB, respectively. Furthermore, Δ_{TF} denotes an offset that depends on transport format (TF) scheme and $f(i)$ denotes the correction value, which is based on the Transmit Power Control (TPC) command. Finally, α is the compensation factor of pathloss that is usually set in the range of $[0, 1]$. This power control scheme is also known as Fractional Power Control (FPC) and is widely used in LTE networks. In this scheme, which is a combination of open-loop and closed-loop control, the user is in charge of measuring signal quality in order to compensate for pathloss and shadowing in the open-loop power control. In the closed-loop power control, eNB generates the power control command based on its measurements and feeds it back to the user using the downlink control signaling channel. The parameter α in (11) specifically makes an equilibrium between cell edge throughput and cell capacity in the open-loop power control.

In order to avoid exceeding the limit of power, the maximum transmit power of each user is split into two proportions between MeNB and SeNB based on the following

$$P_{MeNB} + P_{SeNB} \leq P_{max}, \tag{12}$$

where P_{MeNB} and P_{SeNB} denote the maximum transmit power of the user when transmitting to MeNB and SeNB, respectively. Parameter P_{max} in (11) is replaced by these two values in each tier. Two basic solutions given below are employed in order to tackle the power splitting problem.

1) Splitting Equally: The maximum transmit power of each user is equally split between MeNB and SeNB as

$$P_{MeNB} = P_{SeNB} = \frac{P_{max}}{2}. \tag{13}$$

Although this approach is fairly simple without any overhead, it does not take into account the pathloss difference between MeNB and SeNB and therefore is susceptible to rate deterioration of the users with higher pathloss. This is due to the fact that the users at the cell edge but in close proximity to SeNBs, do not use their power resources for transmitting to MeNB efficiently while the power transmitted to SeNB is wasted.

2) Pathloss Based Splitting: In order to guarantee the users' QoS, a pathloss based power splitting has been proposed. This approach is described as

$$P_{MeNB} = \frac{L_{ij}}{L_{ij} + L_{ik}} P_{max} \quad \forall i \in \mathcal{N}, j \in \mathcal{M}, k \in \mathcal{S} \tag{14}$$

$$P_{SeNB} = \frac{L_{ik}}{L_{ij} + L_{ik}} P_{max} \quad \forall i \in \mathcal{N}, j \in \mathcal{M}, k \in \mathcal{S}. \tag{15}$$

Although as opposed to the previous method, the cell edge users can achieve a desirable rate at the eNB, this method imposes a large amount of interference on the neighboring cells since a large proportion of the maximum power is allocated to the eNB located further from the user compared to the other eNB. Thus, the interference level raises and the performance of the whole system is negatively affected.

C. Problem formulation

The uplink power allocation optimization problem to maximize the total rate of the DC HetNet can be formally stated as

$$\begin{aligned} &\max_{P} \sum_{i\in\mathcal{N}} x_{ij}R_i \\ &\text{s.t.} \\ &C_1 : \sum_{j\in\mathcal{M}\cup\mathcal{S}} x_{ij}P_{ij} \leq P_{max} \quad \forall i \in \mathcal{N}, \end{aligned} \tag{16}$$

In problem (16), we aim to maximize the total transmit power of users. In addition, constraint C_1 implies that the total transmit power of each user when transmitting to its assigned MeNB and SeNB should be less than or equal to the maximum transmit power of users.

IV. Power allocation based on Actor-Critic Deep Deterministic Policy Gradient

In this section, we present our DRL setting to solve the optimization problem (16). We adjust the parameters of the actor-critic based Deep Deterministic Policy Gradient (DDPG) model of [16] such that the DC property of the HetNet is realized through proper choice of state, action, and reward. Moreover, to analyse these parameter, the CNN concepts are adopted in the actor and critic functions. Below, we explain our novel scheme in details. Three main components of each RL problem are state, action, and reward, which are describes below

Algorithm 1: DDPG Algorithm [16]

Initialize weight of actor $\pi(s|\theta^{\pi})$ with θ^{π}
Initialize weight of critic $Q(s|\theta^{Q})$ with θ^{Q}
Initialize target network $Q^{'}$ with weights $\theta^{Q^{'}} \leftarrow \theta^{Q}$
Initialize target network $\pi^{'}$ with weights $\theta^{\pi^{'}} \leftarrow \theta^{\pi}$
Initialize replay buffer R
for $e \leftarrow 1$ **to** $episodes$ **do**
  Initialize the exploration noise Z_0 for action exploration
  Receive initial state s_1
  for $t \leftarrow 1$ **to** $steps$ **do**
    Select action $a_t = \pi(s_t|\theta^{\pi}) + Z_t$ based on the current policy and exploration noise
    Perform action a_t and get reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in replay buffer R
    Randomly select a mini-batch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i^{'} + \gamma^{'} Q^{'}(s_{i+1}, \pi^{'}(s_{i+1}|\theta^{\pi^{'}})|\theta^{Q^{'}})$
    Update critic by minimizing $L(\theta^{Q}) = \mathbb{E}_{s_t \sim \rho^{\beta}, a \sim \beta, r' \sim E}[(Q(s_t, a_t|\theta^{Q}) - y_t)^2]$
    Update the actor policy using randomly selected sample policy gradient
    $$\nabla_{\theta^{\pi}} J = \mathbb{E}_{s_t \sim \rho^{\beta}}[\nabla_a Q(s, a|\theta^{Q})|_{s=s_t, a=\pi(s_t)} \nabla_{\theta^{\pi}} \pi(s|\theta^{\pi})|_{s=s_t}]$$
    Update the target networks
    $$\theta^{Q^{'}} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q^{'}}$$
    $$\theta^{\pi^{'}} \leftarrow \tau\theta^{\pi} + (1-\tau)\theta^{\pi^{'}}$$
  end
end

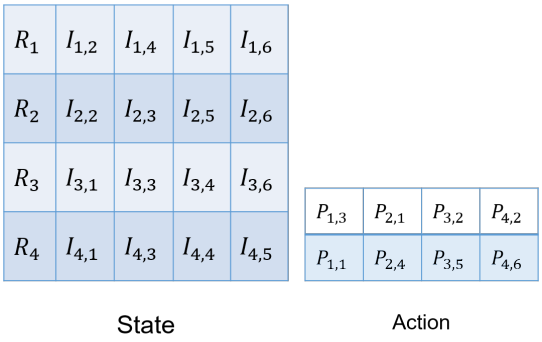

Fig. 4. State and action of the scenario in Fig. 1

- **State:** The state of the environment at time step t is

$$s_t = \mathcal{R} \cup \mathcal{I} \tag{17}$$

- **Action:** The action set of the agent at time step t is

$$a_t = \bigcup_{i=1}^{N} P_i. \tag{18}$$

If the sum of the powers allocated for each user to the eNBS are higher than the maximum power of the user, the powers are scaled such that their sum is equal to the maximum power of each user.

- **Reward:** The reward function for each action at time step t is

$$r_t^{'} = \sum_{i=1}^{N} R_i \tag{19}$$

Therefore, the state of the environment at each time step is an array of size $N(M+S-1)$ with each row corresponding to the rate of each user and the interference on the eNBs that are not assigned to this user. Furthermore, the actions include an array of size $2 \times N$ with each row corresponding to the power set of each user in the macro and micro tiers. Fig. 4 shows the state and action arrays of the scenario depicted in Fig. 1.

In DDPG model of [16], same as every RL setting, an agent interacts with the environment E with a certain action $a_t \in \mathbb{R}^N$ and receives a reward $r^{'}(s_t, a_t)$ and state s_{t+1} at time step t. The actor determines the deterministic policy function $\pi : \mathcal{S} \rightarrow \mathcal{A}$ in which $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces, respectively. The action-value function is defined as:

$$\begin{aligned} Q^{\pi}(s_t, a_t|\theta^{Q}) = \mathbb{E}_{s,r' \sim E, a \sim \pi}[r^{'}(s_t, a_t) \\ + \gamma^{'} Q^{\pi}(s_{t+1}, \pi(s_{t+1})|\theta^{Q})] \end{aligned} \tag{20}$$

in which, $\mathbb{E}$ denotes the expectation value. Furthermore, $\gamma^{'} \in [0, 1]$ and θ^{Q} denote the discount factor and the critic parameters, respectively. Additionally, this expectation depends solely on E and therefore, can be learned by an off-policy approach based on the trajectories generated through a behaviour policy β. The loss function considered for optimizing the critic function is

$$L(\theta^{Q}) = \mathbb{E}_{s_t \sim \rho^{\beta}, a \sim \beta, r' \sim E}[(Q(s_t, a_t|\theta^{Q}) - y_t)^2], \tag{21}$$

in which y_t is obtained by

$$y_t = r^{'}(s_t, a_t) + \gamma^{'} Q(s_{t+1}, \pi(s_{t+1})|\theta^{Q}). \tag{22}$$

The expected sum of discounted future rewards J from the starting state is defined as

$$J = \mathbb{E}_{s,r^{'}\sim E,a\sim\pi}\left[\sum_{i=1}^{T}\gamma^{'(i-1)}r^{'}(s_i,a_t)\right] \quad (23)$$

Assuming the actor function $\pi(s|\theta^{\pi})$, the critic function can be learned by (1). Furthermore, the actor function can be updated by applying the chain rule on J with respect to the actor parameters θ^{π} as

$$\nabla_{\theta^{\pi}}J = \mathbb{E}_{s_t\sim\rho^{\beta}}[\nabla_a Q(s,a|\theta^{Q})|_{s=s_t,a=\pi(s_t)}\nabla_{\theta^{\pi}}\pi(s|\theta^{\pi})|_{s=s_t}], \quad (24)$$

in which ρ^{β} represents the discounted state visitation distribution of behaviour policy β. It is proved in [18] that (24) is in fact the policy gradient.

The exploration noise function Z_t is used to create trajectories in this off-policy method. Similar to [16] we employ Ornstein-Uhlenbeck [19] process in order to produce trajectories based on exploration policy $\pi^{'}(s)$ which is calculated as

$$\pi^{'}(s) = \pi(s|\theta^{\pi}) + Z_t. \quad (25)$$

In order to avoid learning divergence in this approach, [16] has proposed the use of two methods:

- **Replay buffer**: in this method, a finite sized buffer which stores the transition by tuples of (s_t, a_t, r_t, s_{t+1}) is employed. In order to update the actor and critic functions, a mini-batch is selected from the buffer based on uniform distribution.
- **Soft target update**: in this approach, initially, the parameters of the actor and critic networks are copied in networks $Q^{'}(s_t, a_t|\theta^{Q^{'}})$ and $\pi^{'}(s|\theta^{\pi^{'}})$ and the target value is obtained by these two networks. Afterwards, in each update, the parameters of these networks are calculated by $\theta^{'} \leftarrow \tau\theta + (1-\tau)\theta^{'}$ with $\tau << 1$.

The pseudo-code of the DDPG is provided in Algorithm 1.

V. Simulation Results

In this section, we demonstrate simulation results to evaluate our proposed approach of DRL based uplink power control in a DC HetNet. There are two MeNBs in a $500m \times 500m$ coverage area. MeNBs are placed in the center of each cell and four SeNBs and N users are distributed randomly. Additionally, the pathloss of each user i to MeNB j and SeNB k is calculated by $L_{ij}(d_{ij}) = 34 + 40log_2(d_{ij})$ where d_{ij} denotes the distance between user i and MeNB j and $L_{ik}(d_{ik}) = 37 + 30log_2(d_{ik})$ where d_{ik} denotes the distance between user i and SeNB k, respectively. The rest of the simulation parameters are provided in table III. Additionally, the architectures of both actor and critic networks for different number of users are given in Tables I and II, respectively, with abbreviations defined below. Conv(x, y, z, v) denotes a convolution layer with kernel size of $[x, y]$ and z filters with valid padding. Furthermore, FC(n) denotes a fully connected layer with n number of neurons. Additionally, BN stands for batch normalization layer [20]. Moreover, LReLU and ReLU stand for the Leaky Rectifying Linear Unit and Rectifying Linear Unit activation functions, respectively [21], [22]. Finally, Out stands for the linear output of the network. In order to train the RL agent for allocating the power to users, each episode was run on different scenarios with random user locations. Maximum of 20000 episodes were run on each scenario. Fig. 5 depicts the average obtained reward in each episode on all the scenarios. As shown in Fig. 5, the agent is able to converge to an optimum policy for allocating the users' power.

TABLE I
Actor network models

Number of users	Actor
12	State(12,5) - Conv(32,3,2,v) - BN - LReLU - Conv(32,3,2,v) - BN - LReLU - Conv(64,3,2,v) - BN - FC(256) - Sigmoid
16	State(16,5) - Conv(32,5,2,v) - BN - LReLU - Conv(32,5,2,v) - BN - LReLU - Conv(64,3,2,v) - BN - FC(256) - Sigmoid
20	State(20,5) - Conv(32,6,2,v) - BN - LReLU - Conv(32,6,2,v) - BN - LReLU - Conv(64,5,2,v) - BN - LReLU - FC(256) - Sigmoid
24	State(24,5) - Conv(32,6,2,v) - BN - LReLU - Conv(32,5,2,v) - BN - LReLU - Conv(64,5,2,v) - BN - LReLU - FC(512) - LReLU - FC (256) - Sigmoid
28	State(24,5) - Conv(32,9,2,v) - BN - LReLU - Conv(32,6,2,v) - BN - LReLU - Conv(64,6,2,v) - BN - LReLU - FC(512) - LReLU - FC (256) - Sigmoid

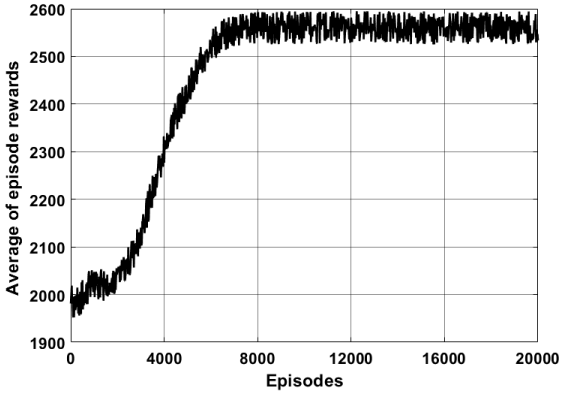

Fig. 5. Average reward of training the agent

In order to evaluate our proposed approach in a DC HetNet, we compare it with methods described in III-B1, III-B2, and Genetic Algorithm (GA) for optimizing (16) in which each gene comprises of a proportion of the power that is allocated to a user. In [23], GA is considered to be a near-optimum method in the non-convex problems such as (16).

In Fig. 6, we compare the achieved total rate of users in the network through our DRL approach with the total rate obtained by splitting equally, pathloss based splitting, and GA. It can

TABLE II
CRITIC NETWORK MODELS

Number of users	Critic
12	State(12,5) - Conv(32,3,2,v) - BN - LReLU - Conv(32,3,2,v) - BN - LReLU - Conv(64,3,2,v) - BN - LReLU - FC (256) - Out1 Action(12,2) - Conv(64,3,2,v) - ReLU - FC(256) - Out2 Out1 * Out2 - ReLU
16	State(16,5) - Conv(32,5,2,v) - BN - LReLU - Conv(32,5,2,v) - BN - LReLU - Conv(64,3,2,v) - BN - LReLU - FC (256) - Out1 Action(16,2) - Conv(64,5,2,v) - ReLU - FC(256) - Out2 Out1 * Out2 - ReLU
20	State(20,5) - Conv(32,6,2,v) - BN - LReLU - Conv(32,6,2,v) - BN - LReLU - Conv(64,5,2,v) - BN - LReLU - FC (512) - Out1 Action(20,2) - Conv(64,6,2,v) - ReLU - FC(512) - Out2 Out1 * Out2 - ReLU
24	State(24,5) - Conv(32,6,2,v) - BN - LReLU - Conv(32,5,2,v) - BN - LReLU - Conv(64,5,2,v) - BN - LReLU - FC(1024) - LReLU - FC (512) - Out1 Action(24,2) - Conv(64,6,2,v) - ReLU - FC(512) - Out2 Out1 * Out2 - ReLU
28	State(28,5) - Conv(32,9,2,v) - BN - LReLU - Conv(32,6,2,v) - BN - LReLU - Conv(64,6,2,v) - BN - LReLU - FC(1024) - LReLU - FC (512) - Out1 Action(24,2) - Conv(64,6,2,v) - ReLU - FC(512) - Out2 Out1 * Out2 - ReLU

TABLE III
SIMULATION PARAMETERS

Parameter	***value***
α	0.8
MeNB total noise power spectral density	-174 dBm/Hz
SeNB total noise power spectral density	-104 dBm/Hz
MeNB shadowing-fading deviation	10 dB
SeNB shadowing-fading deviation	8 dB
MeNB bandwidth (W_M)	2 GHz
SeNB bandwidth (W_S)	3.5 GHz
User transmit power ($P_{N(M\cup S)}$)	25 dBm
P_0	-75 dBm
$\gamma^{'}$	0.99
τ	0.001
Replay buffer size	300000
Mini-batch size	128
Maximum number of episodes	20000

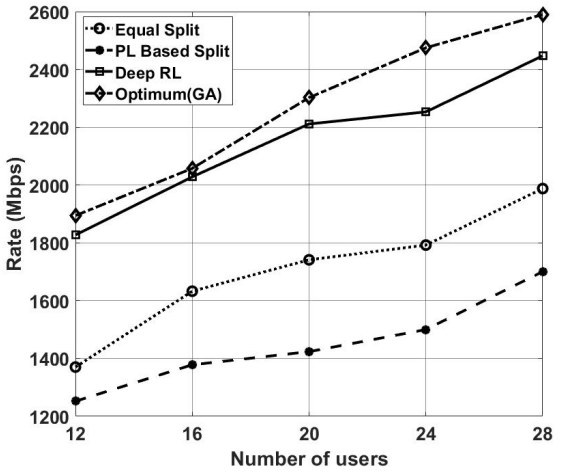

Fig. 6. Total rate versus different number of users

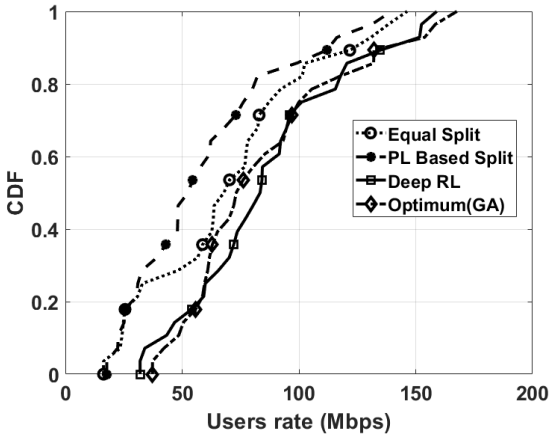

Fig. 7. The CDF of total rate of users

TABLE IV
AVERAGE CPU TIME (SECONDS)

Number of users	Genetic	DRL	Equal Split	PL Split
12	40.2	4e-4	8e-4	9e-4
16	50	1.6e-1	1.4e-3	1.4e-3
20	57	2.2e-2	2e-3	2e-3
24	61	3e-1	2.7e-3	2.7e-3
28	100	5.4e-1	1e-2	2e-2

be seen that our DRL approach is able to allocate power for transmitting to MeNBs and SeNBs in a way that maximizes the total rate of the DC HetNet and is close to the near-optimum answer (GA). Furthermore, it is able to outperform the two basic methods of splitting equally and pathloss based splitting in terms of total rate of users.

Fig. 7 illustrates the Cumulative Distribution Function (CDF) for the rate of 28 users in our DRL approach, the two baseline methods, and the near-optimum answer of GA. As can be seen, our approach is able to obtain higher rate for all of the users compared to splitting equally and all of the users achieve higher data rate than the pathloss based splitting. Furthermore, the CDF of our method demonstrates that for some of the users the total rate is higher than those of the GA answer.

Table IV demonstrates the average CPU time required for obtaining the final answer in each of the methods. As can be seen, the CPU time of DRL agent is marginally less than GA

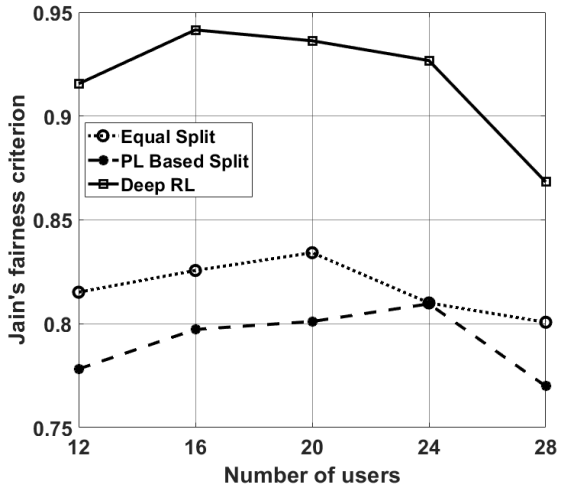

Fig. 8. Jain's fairness criterion

and is close to the base line methods which can be appropriate in practical real world scenarios.

Jain's fairness index [24] is a reliable criterion for measuring the rate fairness in a network. This index is defined as

$$f(R_1, R_2, ..., R_N) = \frac{(\sum_{i=1}^{n} R_i)^2}{N \sum_{i=1}^{N} (R_i^2)}, \tag{26}$$

in which $0 \leq f(R_1, R_2, ..., R_N) \leq 1$ and if this index is equal to 1, all the rates of users are the same. In Fig. 8, Jain's fairness index of users' rate are compared in 3 practical approaches of DRL, PL based split, and equal split. As illustrated, our proposed method is able to present a power allocation in which the users' rate are fairly distributed and a fairness measure of nearly 95% is achieved for the case of 16 users. Additionally, the fairness measure is higher than those of baseline approaches across different number of users. Although with the increasing amount of users in the network, this measure will be decreased, our proposed scheme is still able to outperform the two methods of splitting equally and pathloss based splitting.

VI. CONCLUSION

In this paper, we proposed a power allocation scheme for DC HetNets based on DRL that utilizes the continuous state-action space in order to maximize the total rate of the network. Our simulation results show that our suggested scheme outperforms the baseline methods in terms of total rate.

REFERENCES

[1] J. G. Andrews. "Seven ways that HetNets are a cellular paradigm shift". *IEEE Communications Magazine*, 51(3):136–144, March 2013.

[2] X. Wang, M. Xiao, J. Yi, C. Feng, and F. Jiang. "On the performance analysis of downlink heterogeneous networks with dual connectivity". In *2016 8th International Conference on Wireless Communications Signal Processing (WCSP)*, pp. 1-6, Oct 2016.

[3] S. Lv, Y. Chang, Y. Sun, and M. Hu. "A novel dynamical uplink power control scheme for dual connectivity". In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1-6, Sep. 2016.

[4] G. Pocovi, S. Barcos, H. Wang, K. I. Pedersen, and C. Rosa. "Analysis of Heterogeneous Networks with Dual Connectivity in a Realistic Urban Deployment". In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1-5, May 2015.

[5] S. Barbera, K. Pedersen, P. H. Michaelsen, and C. Rosa. "Mobility Analysis for Inter-Site Carrier Aggregation in LTE Heterogeneous Networks". In *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, pages 1–5, pp. 1-5, Sep. 2013.

[6] 3GPP R1-140625. "Views on open issues for dual connectivity". Ntt docomo, RAN1 #76, 02 2014.

[7] 3GPP R1-140455. "Physical layer aspects for dual connectivity". Qualcomm, RAN1 #76, 02 2014.

[8] J. Liu, J. Liu, and H. Sun. "An Enhanced Power Control Scheme for Dual Connectivity". In *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, pp. 1-5, Sep. 2014.

[9] R. S.Sutton. and A. G.Barto. *"Introduction to Reinforcement Learning"*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[10] R. Amiri, H. Mehrpouyan, L. Fridman, R. K. Mallik, A. Nallanathan, and D. Matolak. "A Machine Learning Approach for Power Allocation in HetNets Considering QoS". In *2018 IEEE International Conference on Communications (ICC)*, pp. 1-7, May 2018.

[11] Y. Sinan Nasir and D. Guo. "Deep Reinforcement Learning for Distributed Dynamic Power Allocation in Wireless Networks". *arXiv e-prints*, August 2018.

[12] F. Meng, P. Chen, and L. Wu. "Power Allocation in Multi-user Cellular Networks With Deep Q Learning Approach". *CoRR*, abs/1812.02979, 2018.

[13] I. Goodfellow, Y. Bengio, and A. Courville. *"Deep Learning"*. The MIT Press, 2016.

[14] Y. LeCun, K. Kavukcuoglu, and C. Farabet. "Convolutional networks and applications in vision". In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 253–256, pp. 253-256, May 2010.

[15] 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, Ch. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, Sh. Legg, and D. Hassabis. "Human-level control through deep reinforcement learning". *Nature*, 518(7540):529–533, February 2015.

[16] 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". *CoRR*, abs/1509.02971, 2015.

[17] 3GPP. "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 12)". Technical specification (ts), 3rd Generation Partnership Project (3GPP), 12 2015. Version 12.8.0.

[18] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. "Deterministic Policy Gradient Algorithms". In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14. pp. I-387–I-395, JMLR.org, 2014.

[19] G. E. Uhlenbeck and L. S. Ornstein. "On the Theory of the Brownian Motion". *Phys. Rev.*, 36:823–841, Sep 1930.

[20] I. Sergey and S. Christian. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". *CoRR*, abs/1502.03167, 2015.

[21] A. L.Maas, A. Y.Hannun, and A Y.Ng. "Rectifier nonlinearities improve neural network acoustic models". In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[22] V. Nair and G. E.Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 807–814, USA, 2010. Omnipress.

[23] K. I. Ahmed, H. Tabassum, and E. Hossain. Deep learning for radio resource allocation in multi-cell networks. *CoRR*, abs/1808.00667, 2018.

[24] R. Jain, D. Chiu, and W. Hawe. "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems". *CoRR*, cs.NI/9809099, 1998.