This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2933417, IEEE Transactions on Wireless Communications

Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Cellular Networks

Nan Zhao, *Member, IEEE*, Ying-Chang Liang, *Fellow, IEEE*, Dusit Niyato, *Fellow, IEEE*, Yiyang Pei, *Member, IEEE*, Minghu Wu, Yunhao Jiang

***Abstract*—Heterogeneous cellular networks can offload the mobile traffic and reduce the deployment costs, which have been considered to be a promising technique in the next-generation wireless network. Due to the non-convex and combinatorial characteristics, it is challenging to obtain an optimal strategy for the joint user association and resource allocation issue. In this paper, a reinforcement learning (RL) approach is proposed to achieve the maximum long-term overall network utility while guaranteeing the quality of service requirements of user equipments (UEs) in the downlink of heterogeneous cellular networks. A distributed optimization method based on multi-agent RL is developed. Moreover, to solve the computationally expensive problem with the large action space, multi-agent deep RL method is proposed. Specifically, the state, action and reward function are defined for UEs, and dueling double deep Q-network (D3QN) strategy is introduced to obtain the nearly optimal policy. Through message passing, the distributed UEs can obtain the global state space with a small communication overhead. With the double-Q strategy and dueling architecture, D3QN can rapidly converge to a subgame perfect Nash equilibrium. Simulation results demonstrate that D3QN achieves the better performance than other RL approaches in solving large-scale learning problems.**

***Index Terms*—Heterogeneous cellular networks, user association, resource allocation, multi-agent deep reinforcement learning.**

This work was supported by the National Natural Science Foundation of China (No. 61501178, No. 61571100, No. 61631005, No. 61628103, and No. 61771187) and the Natural Science Foundation of Hubei Province (No. 2018CFB698). Part of this paper was presented at the IEEE Global Communications Conference, UAE, December 2018 [1]. (Corresponding author: Ying-Chang Liang)

N. Zhao is with the Hubei Collaborative Innovation Center for High-efficiency Utilization of Solar Energy, Hubei University of Technology, Wuhan 430068, China, and also with the Center for Intelligent Networking and Communications (CINC), University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: nzhao@mail.hbut.edu.cn).

Y.-C. Liang is with the Center for Intelligent Networking and Communications (CINC), University of Electronic Science and Technology of China (UESTC), Chengdu, China (e-mail: liangyc@ieee.org).

D. Niyato is with the Nanyang Technological University, Singapore (e-mail: dniyato@ntu.edu.sg).

Y. Pei is with the Singapore Institute of Technology, Singapore (e-mail: yiyang.pei@singaporetech.edu.sg).

M. Wu and Y. Jiang are with the Hubei Key Laboratory for High-efficiency Utilization of Solar Energy and Operation Control of Energy Storage System, Hubei University of Technology, Wuhan 430068, China (e-mail: wuxx1005@mail.hbut.edu.cn, jyh6858@126.com).

I. Introduction

With the rapid growth in mobile devices and the emergence of Internet of Things, the next-generation wireless network faces the enormous challenge to deal with the surging increase of wireless applications [2]. The most promising solution is to use picocells and femtocells with various transmission powers and coverage ranges to densify the existing cellular networks. These heterogeneous networks (HetNets) can offload user equipments (UEs) from macro base stations (MBSs) to pico BSs (PBSs) and femto BSs (FBSs), which are different in the transmission power and coverage size [3]. Moreover, to achieve the high spectrum efficiency of the HetNets, PBSs and FBSs can reuse and share the same channels with MBSs. Therefore, HetNets have been regarded to be a good strategy to increase system capacity for the future wireless communication.

There are some optimization issues in such HetNets, such as user association and resource allocation (UARA). The user association problem has been investigated in [4] and [5]. Joint UARA issue has been discussed in [6]–[9]. However, considering the non-convex and combinatorial characteristics, it is challenging to obtain a globally optimal strategy of the joint optimization problem. Some studies have recently proposed new methods, such as game-theoretic approach [10], linear programming method [11], and Markov approximation strategy [12]. However, these approaches need nearly complete information, which may not be typically available. Therefore, it is challenging for above approaches to achieve the optimal solutions without such complete information. In this paper, we concentrate on reinforcement learning (RL) method to solve such challenging problem.

In RL approach, RL agents consider the maximum long-term rewards, rather than simply getting the current optimal rewards. This is important for the time-varying dynamic system [13], [14], especially in wireless networks [15], [16]. Q-learning is a RL method used extensively [17]. In a single-agent RL framework, the independent agents may change their respective actions without collaboration, resulting in fluctuating actions on the strategy for learning process [18]. Additionally, since one agent's action may influence other agents' rewards, multi-agent RL (MARL) should be considered. The authors in [19] investigated a docitive Q-learning method to achieve the optimal joint strategy of resource allocation and power control. An autonomous Q-learning method was

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2933417, IEEE Transactions on Wireless Communications

studied in [20] to solve the optimal resource allocation issue in the HetNets. In [21], an online RL-based user association strategy was proposed in vehicular networks. Ghadimi *et al.* [22] studied a RL-based approach to achieve power control and rate adaptation in cellular networks. A learning method was investigated in [23] to obtain the optimal resource allocation and network access strategy in LAA-LTE based HetNets. However, none of the existing RL works investigated the UARA coordination in the downlink of the HetNets.

Moreover, considering the huge state and action spaces in the above joint optimization issue, it is challenging to obtain the optimal strategy using Q-learning approach. Fortunately, the emerging deep reinforcement learning (DRL) [24] is considered as a promising technique to solve the complex control issues, especially for the high-dimension solutions. Deep Q-Network (DQN) can achieve the better performance than Q-learning method with the help of deep neural network (DNN). Recently, DRL-based approach has been applied to several research fields, such as resource management and allocation [25]–[31], mobile offloading [32], dynamic channel access [33], fog radio access networks [30], mobile edge computing and caching [34]. Furthermore, multi-agent DRL method has also be considered in the above related areas [35], [36]. However, there is very little research to develop a DRL-based approach to solve the joint optimization issue.

In this paper, the joint UARA optimization issue is investigated in the downlink of the HetNets. The distributed optimization algorithm based on DRL is proposed to obtain the joint UARA strategy. In particular, this paper has the following main contributions:

- *New Solution Technique:* A multi-agent DRL-based method is proposed to solve the joint UARA optimization problem. The optimization issue is investigated to obtain the optimal long-term network utility while guaranteeing UEs' QoS requirements. The optimal solution is obtained by jointly associating UEs to BSs and allocating channels to UEs.
- *Optimal Algorithm Design:* Considering the non-convex and combinatorial characteristics of the joint optimization issue, the multi-agent DRL method is proposed. Specifically, the state, action and reward function are defined for the UEs. Then, dueling-double DQN (D3QN) approach is developed to solve the joint optimization issue. Through message passing, all UEs obtain the global state information with a little communication overhead. With the help of the double-Q strategy and dueling architecture, D3QN can achieve a near-optimal strategy by approximating the action-value functions from the current state.
- *Performance Analysis:* Simulation results are given to indicate that our proposed method achieves faster convergence speed and better learning performance than other MARL approaches. Instead of a large number of steps required to reach the near-optimal strategy in Q-learning method, the proposed D3QN approach achieves a subgame perfect Nash equilibrium (SPNE) with less learning steps.

The remainder of this paper unfolds as follows. We introduce the system model and problem formulation in Section II. Multi-agent RL and DRL methods are investigated in Section III and Section IV, respectively. Simulation results are presented and discussed in Section V, followed by conclusion in Section VI.

II. System Model and Problem Formulation

In this section, the system model and problem formulation are described in this paper.

A. System Model

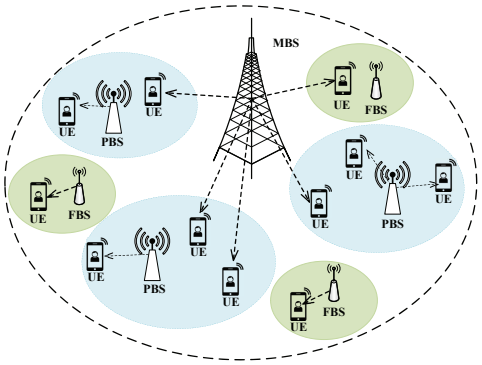


Fig. 1. Three-tier heterogeneous cellular network.

Consider a typical HetNet composed of N_m MBSs, N_p PBSs, N_f FBSs and N randomly located UEs, as shown in Fig. 1. PBSs are usually deployed to offload traffic from the macrocell with medium transmission power. FBSs are commonly used for small coverage areas with better Qos and higher data rates. We denote the set of all BSs by $\mathcal{BS} = \{mbs_1, \ldots, mbs_{n_m}, pbs_1, \ldots, pbs_{n_p}, fbs_1, \ldots, fbs_{n_f}\}$, with the set of the BSs' indices $\mathcal{B} = \{0, 1, \ldots, L-1\}$, where $L = N_m + N_p + N_f$. Assume that the BSs operate on K shared orthogonal channels.

Considering that different BSs may have different sets of UEs, the binary *user-association* vector is given by $b_i^l(t) = (b_i^0(t), \ldots, b_i^{L-1}(t)), i \in \mathcal{N}, l \in \mathcal{B}$, where $\mathcal{N} = \{1, \ldots, N\}$, $b_i^l(t) = 1$ when the i^{th} UE chooses to associate with the $\mathcal{BS}_l$, and $b_i^l(t) = 0$ otherwise. Assume that each UE can only choose at most one BS at any time. That is,

$$\sum_{l=0}^{L-1} b_i^l(t) \leq 1, \forall i \in \mathcal{N}. \tag{1}$$

Moreover, spectrum resource of each BS is assigned to its associated UEs. For the i^{th} UE, the binary *channel-allocation* vector is given by $c_i^k(t) = (c_i^1(t), \ldots, c_i^K(t)), k \in \mathcal{K}, i \in \mathcal{N}$, where $\mathcal{K} = \{1, \ldots, K\}$. If the i^{th} UE utilizes the channel C_k at time t, we have $c_i^k(t) = 1$, and $c_i^k(t) = 0$ otherwise. Although the number of UEs operating simultaneously on the same channel is unlimited, for the simplicity of discussion, we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2933417, IEEE Transactions on Wireless Communications

assume that each UE can only choose at most one channel at any time[1]. That is,

$$\sum_{k=1}^{K} c_i^k(t) \leq 1, \forall i \in \mathcal{N}. \tag{2}$$

Since PBSs and FBSs are deployed within the radio coverage of the MBSs, co-channel interference should be considered. Let $\mathbf{p_{li}}(t) = (p_{li}^1(t), \ldots, p_{li}^K(t)), l \in \mathcal{B}, i \in \mathcal{N}, k \in \mathcal{K}$ be a vector of the transmit power used on the channel C_k between the i^{th} UE and its associated $\mathcal{BS}_l$ at time t. Then, the signal to interference plus noise ratio (SINR) at the i^{th} UE from $\mathcal{BS}_l$ using C_k is

$$\Gamma_{li}^k(t) = \frac{b_i^l(t) h_l^{i,k}(t) c_i^k(t) p_{li}^k(t)}{\sum\limits_{j \in B \setminus \{l\}} b_i^j(t) h_j^{i,k}(t) c_i^k(t) p_{ji}^k(t) + W N_0}, \tag{3}$$

where $h_l^{i,k}(t)$ is the channel gain between the $\mathcal{BS}_l$ and the i^{th} UE operating on the channel C_k at time t, W denotes the channel bandwidth, and N_0 denotes the noise power spectral density.

The downlink capacity of the i^{th} UE from $\mathcal{BS}_l$ on the channel C_k can be achieved as

$$r_{li}^k(t) = W \log_2(1 + \Gamma_{li}^k(t)). \tag{4}$$

Thus, we have the total transmission capacity of the i^{th} UE, that is,

$$r_i(t) = \sum_{l=0}^{L-1} \sum_{k=1}^{K} r_{li}^k(t) = \sum_{l=0}^{L-1} \sum_{k=1}^{K} W \log_2(1 + \Gamma_{li}^k(t)). \tag{5}$$

B. Problem Formulation

Consider that all UEs want to obtain their maximum transmission capacity from their selected BSs while meeting a minimum QoS requirement Ω_i. Thus, we assume that the SINR of the i^{th} UE $\Gamma_i(t)$ should be not less than the minimum QoS requirement Ω_i,

$$\Gamma_i(t) = \sum_{l=0}^{L-1} \sum_{k=1}^{K} \Gamma_{li}^k(t) \geq \Omega_i. \tag{6}$$

Moreover, considering that $p_{li}^k(t)$ is the transmit power of $\mathcal{BS}_l$, the total transmission cost associated with the i^{th} UE can be given by

$$\varphi_i(t) = \sum_{l=0}^{L-1} \lambda_l b_i^l(t) \sum_{k=1}^{K} c_i^k(t) p_{li}^k(t), \tag{7}$$

where λ_l is the unit price of the $\mathcal{BS}_l$'s transmit power.

Then, we define the i^{th} UE's utility $w_i(t)$ as the difference between the achieved profit and the transmission cost, which is given by

$$\begin{aligned} w_i(t) &= \rho_i r_i(t) - \varphi_i(t) \\ &= \sum_{l=0}^{L-1} \left[\sum_{k=1}^{K} \left[\rho_i r_{li}^k(t) - \lambda_l b_i^l(t) c_i^k(t) p_{li}^k(t) \right] \right], \end{aligned} \tag{8}$$

[1]This assumption can be relaxed. In this case, besides choosing one BS, each UE needs to choose k_i channels from K channels. Then, the action space of each UE will increase from LK to $LC_K^{k_i}$. Our proposed multi-agent DRL method can also be applied to solve the joint user association and channel allocation problem.

where ρ_i is the profit per transmission rate.

Consequently, considering that action selection may consume some cost, the reward of the i^{th} UE can be given as the utility minus the action-selection cost Ψ_i, that is,

$$\mathcal{R}_i(t) = w_i(t) - \Psi_i, \tag{9}$$

where $\Psi_i > 0$. Note that the negative reward $(-\Psi_i)$ acts as a punishment. If the value of the negative reward is small, the UE may tolerate it. Therefore, to guarantee the minimum QoS of all UEs, this negative reward should be set big enough.

In this HetNet, the joint UARA optimization problem is to maximize the long-term reward. Here, we define the long-term reward Φ_i as the weighted sum of the instantaneous rewards over a finite period T, that is,

$$\Phi_i = \sum_{t=0}^{T-1} \gamma^t \mathcal{R}_i(t), \tag{10}$$

where $\gamma \in [0, 1)$ is the discount rate to determine the weight of the future reward. $\gamma = 0$ means that we only care about the immediate reward. When $\gamma < 1$, the rewards in the future are less than the rewards of the earlier periods. Note that, when the UEs choose the associated BS and the accessed channels, the action space of joint UARA issue increases exponentially with the number of UEs.

Note that due to the non-convex and combinatorial characteristics, it is challenging to obtain a globally optimal strategy of the joint UARA optimization issue. Moreover, the traditional method may need the nearly complete information to achieve the optimal solution. In the following sections, we first propose MARL method for joint optimization strategy. The state, action and reward of MARL are defined for the UEs. Each UE can get the global state information with message passing. Multi-agent Q-learning (MAQL) approach is presented in Section III. Moreover, in order to handle the large action space of the joint optimization problem, with the help of double-Q strategy and dueling architecture, multi-agent D3QN is investigated to solve the joint optimization issue in Section IV.

III. Multi-Agent Cooperative RL for the Joint UARA Optimization Problem

Here, we first present the formulation of a stochastic game. Then, the MARL method is investigated to obtain the optimal solution to the optimization issue.

A. Game Formulation

Assumed that all UEs do not know the network environment and the quality of operating channel. UEs are selfish and rational and each UE selects the BSs and channels to obtain the maximum long-term reward Φ_i. At any time t, each UE's reward relies on network environment's current state and other UEs' actions in the HetNets. At next time, the game turns into a completely new stochastic state which is influenced by the previous state and the selected actions of all UEs. Consequently, we formulate this joint optimization issue as a stochastic game $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$ [37],

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2933417, IEEE Transactions on Wireless Communications

- $\mathcal{N}$ is the set of N UEs;
- $\mathcal{S}$ is the set of possible states;
- $\mathcal{A}_i$ is the set of the i^{th} UE's action, and $\overrightarrow{a} = (a_1, \ldots, a_N)^T \in \times_i \mathcal{A}_i$ is a vector of the joint action of all UEs;
- $\mathcal{P}$ is the sate transition probability. $\mathcal{P}_{ss'}(\overrightarrow{a})$ is the state transition probability from state s to state s' by taking joint action;
- $\mathcal{R}_i$ is the reward function of the i^{th} UE.

We define the state $s(t)$ to indicate whether each UE meets its QoS demand at time t, that is,

$$s(t) = \{s_1(t), s_2(t), \ldots, s_N(t)\}, \tag{11}$$

where $s_i(t) \in \{0, 1\}$. $s_i(t) = 0$ means that the i^{th} UE cannot meet its the minimum QoS requirement, that is, $\Gamma_i(t) < \Omega_i$, and $s_i(t) = 1$ means that $\Gamma_i(t) \geq \Omega_i$.

Notice that the number of possible states is 2^N, and this number can be very large for large N.

Moreover, consider that all UEs need to choose a BS and transmission channel at the time t. Then, the action space $\mathcal{A}_i$ of each UE can be defined as

$$a_{li}^k(t) = \{b_i^l(t), c_i^k(t)\}, \tag{12}$$

where $b_i^l(t) \in \{0, 1\}$, $\boldsymbol{b}_i^l(t) \in \{b_i^0(t), \ldots, b_i^{L-1}(t)\}$, $c_i^k(t) \in \{0, 1\}$ and $\boldsymbol{c}_i^k(t) \in \{c_i^1(t), \ldots, c_i^K(t)\}$.

The number of possible actions of each UE is LK, and this number can be large as L and K increase. Furthermore, the action generally affects the evolution of the state. At any time t, the action vector of the other $N-1$ UEs is defined as $\mathcal{A}_{-i}(t) = \{a_1(t), \ldots, a_{i-1}(t), a_{i+1}(t), \ldots, a_N(t)\}$.

Furthermore, when the i^{th} UE takes the action $a_{li}^k(t)$, it can achieve the immediate reward $\mathcal{R}_i$. Considering the actions of the other UEs $\boldsymbol{a}_{-i}(t)$, the reward of the i^{th} UE can be expressed as $\mathcal{R}_i(t) = \mathcal{R}_i(s, a_i^*, \boldsymbol{a}_{-i}^*)$.

Here, we define action vector $(a_i, \boldsymbol{a}_{-i}) \in \mathcal{A}$ as the possible solution to this game. When the following inequalities hold true for every UE in any $s \in \mathcal{S}$, the game obtains the NE state [38]:

$$\mathcal{R}_i(s, a_i^*, \boldsymbol{a}_{-i}^*) \geq \mathcal{R}_i(s, a_i, \boldsymbol{a}_{-i}^*), \forall a_i \in \mathcal{A}_i. \tag{13}$$

Note that each UE's action can be considered as the best response to other UEs' actions. No UEs can benefit from unilateral deviation [38].

Assume that this stochastic game is considered to be episodic. At the end of every episode, we reset the state. The policy in every episode is composed of the state, actions and rewards. We can use the policy to obtain the accumulative rewards from the environment. If each UE obtains the information about reward function and state transition, the NE strategy can be found with integer programming methods. However, in this non-cooperative stochastic game, such information is unknown to each UE. To address this challenge, we resort to reinforcement learning method to obtain a NE strategy π_i^* at any state s by interacting with the environment repeatedly.

B. Multi-Agent Q-learning Method

Here, a finite-state Markov decision process (MDP) is introduced to describe this stochastic game. Then, MAQL method is proposed to solve the joint UARA problem. Since UEs' cumulative rewards may be influenced by the current state and the UEs' actions, learning tasks of UEs can be satisfied the Markov property. We model the joint optimization issue as a MDP $(\mathcal{S}; \mathcal{A}_i; \mathcal{R}_i; \mathcal{P}_{ss'})$, where $\mathcal{S}$ is a discrete set of environment states, $\mathcal{A}_1, \ldots, \mathcal{A}_N$ define a discrete set of possible actions of UEs, $\mathcal{R}_1, \ldots, \mathcal{R}_N$ are the reward functions of UEs and $\mathcal{P}_{ss'}$ means the state transition probability.

In an unknown stochastic environment, by using RL method, the optimal policy $\pi_i^* : \mathcal{S} \to \mathcal{A}_i$ is obtained to maximize the long-term reward for each agent [39]. Considering its simplicity and distributed characteristic, collaborative MARL is considered with local states. More precisely, all UEs try to learn the optimal policies $a_i^* = \pi_i^*(s) \in \mathcal{A}_i$ based on the state space through message passing. Each UE iteratively sends its own optimal policy and state information to its associated BS with only one bit (0 or 1). By the message passing between the BSs through a backhaul communication link, the global state information and joint policies of all UEs is obtained. This incurs negligible communication overhead [40]. For the i^{th} UE, an optimal policy π_i^* should be obtained at each state to maximize its value-state function, which can be defined as

$$V_i(s, \pi_i, \pi_{-i}) = E\left[\sum_{t=0}^{T-1} \gamma^t \mathcal{R}_i(s(t), \pi_i(t), \pi_{-i}(t)) | s(0) = s\right], \tag{14}$$

where $E[\cdot]$ denotes the expectation operator, π_{-i} denotes the vector of strategies of the other $N-1$ agents, i.e., $\pi_{-i} = (\pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_N)$.

Considering the Markov property, the value function can be rewritten as

$$\begin{aligned} V_i(s, \pi_i, \pi_{-i}) &= u_i(s, \pi_i, \pi_{-i}) \\ &\quad + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}(\pi_i, \pi_{-i}) V_i(s', \pi_i, \pi_{-i}), \end{aligned} \tag{15}$$

where $u_i(s, \pi_i, \pi_{-i}) = E\left[\mathcal{R}_i(s, \pi_i, \pi_{-i})\right]$, and $P_{ss'}(\pi_i, \pi_{-i})$ is the state transition probability.

Notice that the accumulative expected discounted rewards of each UE is determined by $V_i(s, \pi_i, \pi_{-i})$. And the immediate reward can be obtained from each state on every possible policy.

Consequently, a strategy tuple (π_i^*, π_{-i}^*), with $\pi_{-i}^* = (\pi_1^*, \ldots, \pi_{i-1}^*, \pi_{i+1}^*, \ldots, \pi_N^*)$, is an NE if the following inequalities hold for any π_i:

$$V_i(s, \pi_i^*, \pi_{-i}^*) \geq V_i(s, \pi_i, \pi_{-i}^*), \forall s \in \mathcal{S}. \tag{16}$$

Considering that any finite games have the mixed-strategy equilibrium, there always exists an NE to satisfy the following Bellman optimality equation [41]:

$$\begin{aligned} V_i^*(s, \pi_i, \pi_{-i}) &= V_i(s, \pi_i^*, \pi_{-i}^*) \\ &= \max_{a_i \in \mathcal{A}_i} \left[u_i(s, a_i, \pi_{-i}^*) + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}(a_i, \pi_{-i}^*) V_i(s', \pi_i^*, \pi_{-i}^*)\right]. \end{aligned} \tag{17}$$

To deal with such MDP problem, Q-learning is one of the popular RL methods [17]. The optimal Q-value function $Q_i^*(s, a_i)$ is achieved from the following Bellman's equation:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2933417, IEEE Transactions on Wireless Communications

Algorithm 1 MAQL method for the joint UARA optimization problem in the HetNets

- **Input:** List of allowed actions to be taken by all UEs.
- **Output:** Optimal sequence of actions required to achieve the QoS requirement of all UEs, if feasible.
- Each UE initializes Q-values $Q_i(s, a_i)$, e.g., to 0 for all states.
- **for** *each episode* to EP episodes **do**
- Initialize the network state s through message passing.
- **for** each step of an episode to T steps **do**
- Each UE chooses an action a_i at the state s using ϵ-greedy policy from $Q_i(s, a_i)$.
- Each UE sends its request to the certain BS to access the selected channel. If the BS sends feedback signal to indicate the available channel frequency to the UE, then the UE obtains the immediate reward $u_i(s, a_i)$. Otherwise, the BS will not reply anything and the UE will obtain a negative reward.
- Each UE obtains the network state s' through message passing. Set $s \leftarrow s'$.
- Each UE updates Q-values based on (21).
- **if** the current state is $s = \{1, \ldots, 1\}$, **then**
- break.
- **end if**
- **end loop for** T **steps.**
- **end loop for** EP **episodes.**

$$Q_i^*(s, a_i) = u_i(s, a_i, \pi_{-i}^*) + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}(a_i, \pi_{-i}^*) V_i(s', \pi_i^*, \pi_{-i}^*). \tag{18}$$

Although there may exist more than one NE strategy $\pi_i^*(s)$ for each UE, all UEs may have the same action-values $Q_i^*(s, a_i)$ and the optimal policy $\pi_i^*(s)$ could be obtained from $Q_i^*(s, a_i)$ [42],

$$V_i(s, \pi_i^*, \pi_{-i}^*) = \max_{a_i \in \mathcal{A}_i} Q_i^*(s, a_i). \tag{19}$$

Then, combining (18) and (19), we have

$$\begin{aligned} Q_i^*(s, a_i) &= u_i(s, a_i, \pi_{-i}^*) \\ &+ \gamma \sum_{s' \in \mathcal{S}} P_{ss'}(a_i, \pi_{-i}^*) \max_{a_i' \in \mathcal{A}_i} Q_i^*(s', a_i'). \end{aligned} \tag{20}$$

Generally, it is difficult to get the information about the transition probability $P_{ss'}(a_i, \pi_{-i})$. However, through Q-learning method, the optimal strategy can be found based on the available information $(s, a_i, s', u_i(s, a_i, \pi_{-i}))$ in a recursive manner. The update equation of Q-value function is given by

$$\begin{aligned} Q_i(s, a_i) &= Q_i(s, a_i) \\ &+ \delta \left[u_i(s, a_i, \pi_{-i}) + \gamma \max_{a_i' \in \mathcal{A}_i} Q_i(s', a_i') - Q_i(s, a_i) \right], \end{aligned} \tag{21}$$

where δ denotes the learning rate to determine the update of $Q_i(s, a_i)$. It has been proven that by appropriately setting δ, Q-learning method can tend to converge when updating $Q_i(s, a_i)$ [43].

Furthermore, during the learning procedure, the tradeoff of *exploration-exploitation* should be considered in the *action selection* mechanism. In order to balance the exploitation of the current best Q-value function with the exploration of the better option, we adopt ϵ-greedy policy in the *action selection* mechanism. In ϵ-greedy policy, an action is selected at random with probability ϵ, and the best action a_i^* is selected with probability $1 - \epsilon$.

Algorithm 1 describes MAQL method for the joint optimization problem. At the beginning of each training episode, the network state is initialized through message passing. Each UE is connected to the neighboring BS with the maximum received signal power. By using a pilot signal, each UE can measure the received power from the associated BS and the randomly-selected channel. Then, each UE reports its own current state to its current associated BS. By the message passing among the BSs through the backhaul communication link, the global state information of all UEs is obtained. Then, the BSs send this global state information s to all UEs.

Each episode lasts T steps (500 steps in our experiments). In each step of an episode, the execution action a_i is selected with the ϵ-greedy policy from the estimated Q-value $Q_i(s, a_i)$. Then, each UE i sends its request to BS l with which the UE has chosen to associate. The request contains the index of the required channel. Then, the BS l either accepts or rejects the request from the i^{th} UE depending on its available resources. If the BS l accepts the request of the i^{th} UE, the BS l will send feedback signal to indicate the available channel and the time-slot to the i^{th} UE. Otherwise, the BS l will not reply anything. Then, after obtaining the immediate reward $u_i(s, a_i)$ and next state s', each UE updates Q-values $Q_i(s, a_i)$. Each episode ends when the QoS of all UEs is satisfied or when the maximum step T is reached. The total episode reward is the accumulation of instantaneous rewards of all steps within an episode.

IV. Multi-Agent DRL for the Joint UARA Optimization Problem

Generally, Q-learning method performs well for small state and action spaces. However, when the spaces become huge, it is challenging to find the optimal strategy in the huge Q-value table. Therefore, multi-agent DRL is considered to solve the joint optimization issue in this section. Recently, to deal with the issue with large spaces, a DNN has been introduced into the framework of Q-learning. DQN is the most well adopted method. In DQN, DNN is used to represent action and state spaces. Moreover, the Q-value function $Q_i(s, a_i)$ is approximated by mapping from a state to an action. A NN function approximator $Q_i(s, a_i; \theta) \approx Q_i^*(s, a_i)$ with weights θ is used as an *online network*. The DQN utilizes a *target network* alongside the *online network* to stabilize the overall network performance. The Q-network updates its weights to minimize the loss function defined as

$$L_i(\theta) = E_{s, a_i, u_i(s, a_i), s'}[(y_i^{DQN} - Q_i(s, a_i; \theta))^2], \tag{22}$$

where $y_i^{DQN} = u_i(s, a_i) + \gamma \max_{a_i' \in \mathcal{A}_i} Q_i(s', a_i'; \theta^-)$, and θ^- represents the weights of a *target network*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2933417, IEEE Transactions on Wireless Communications

Algorithm 2 Multi-agent D3QN Algorithm for the joint UARA problem in HetNets

- **Input:** List of allowed actions to be taken by all UEs.
- **Output:** Optimal sequence of actions required to achieve the QoS requirement of all UEs, if feasible.
- Initialize the replay memory $\mathcal{D}$, DQN network parameters θ, and the target network replacement frequency N^-.
- Initialize the *online network* $Q(s,a;\theta)$ with weights θ.
- Initialize the *target network* $Q_i(s',a_i';\theta^-)$ with weights $\theta^- = \theta$.
- **for** *each episode* to EP episodes **do**
- Initialize the network state s through message passing.
- **for** each step of an episode to T steps **do**
- Each UE chooses an action a_i at the state s using ϵ-greedy policy from $Q_i(s,a_i;\theta)$,
- Each UE sends its request to the certain BS to access the selected channel. If the BS sends feedback signal to indicate the available channel frequency to the UE, then the UE obtains the immediate reward $u_i(s,a_i)$. Otherwise, the BS will not reply anything and the UE will obtain a negative reward.
- Each UE obtains the network state s' through message passing. Set $s \leftarrow s'$.
- Each UE stores transition $(s,a_i,u_i(s,a_i),s')$ in $\mathcal{D}$.
- Each UE samples random mini-batch of transitions $(s,a_i,u_i(s,a_i),s')$ from $\mathcal{D}$.
- Each UE sets y_i^{DDQN} based on (23).
- Each UE performs the gradient descent step on $(y_i^{DDQN} - Q_i(s,a_i;\theta))^2$.
- In every N^- steps, each UE replaces target parameters, $\theta^- = \theta$.
- **if** the current state is $s = \{1,\ldots,1\}$, **then**
- break.
- **end if**
- **end loop for** T **steps.**
- **end loop for** EP **episodes.**

The action a_i can be chosen from the *online network* $Q_i(s,a_i;\theta)$ with a simple ϵ-greedy policy. Although the *target network* is a duplicate of $Q_i(s,a_i;\theta)$, the weights of the *target network* are fixed for a number of iterations while the weights are updated in the *online network*.

In DQN, in order to overcome learning instability, the *experience replay* strategy is used. The transition of the form $(s,a_i,u_i(s,a_i),s')$ is stored in the experience replay memory $\mathcal{D}$. During learning, instead of using only the current experience $(s,a_i,u_i(s,a_i),s')$, the NN can be trained through sampling mini-batches of experiences from replay memory $\mathcal{D}$ uniformly at random. By reducing the correlation among the training examples, the *experience replay* strategy ensures that the optimal policy cannot be driven to a local minima.

Moreover, since the same values are used to select and evaluate an action in Q-learning and DQN methods, Q-value function may be over-optimistically estimated. Thus, double DQN (DDQN) [44] is used to mitigate the above problem by replacing the target y_i^{DQN} by the following target y_i^{DDQN} defined as

$$y_i^{DDQN} = u_i(s,a_i) + \gamma Q_i\left(s', \arg\max_{a_i' \in \mathcal{A}_i} Q_i(s',a_i';\theta);\theta^-\right). \tag{23}$$

The procedure of RL with DDQN strategy is shown in Fig. 2. More specifically, the next state s' is employed by both the *online network* and *target network* to compute the optimal value $Q_i(s',a_i';\theta)$. Then, with the discount factor γ and the current reward $u_i(s,a_i)$, the target value y_i^{DDQN} is obtained. Finally, the error is calculated by subtracting the target value with the optimal value $Q_i(s,a_i;\theta)$ predicted by the *online network*, and then is backpropagated to update the weights.

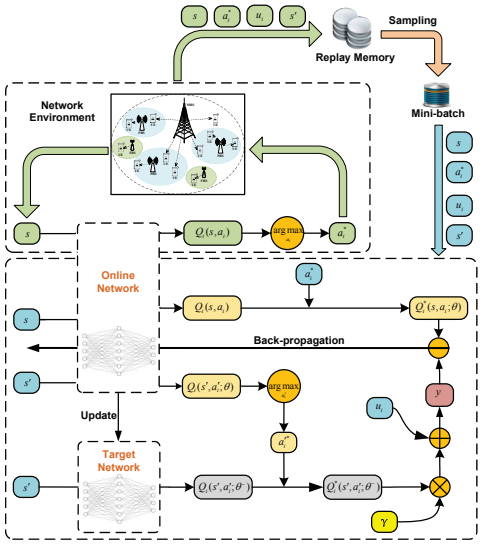


Fig. 2. Reinforcement learning with DDQN strategy.

Furthermore, considering that Q-value function can describe how beneficial an action a_i is taken at a state s, the dueling neural network [45] is introduced to obtain the estimation of a value function $V(s)$ and an *advantage function* $A(s,a_i) = Q_i(s,a_i) - V(s)$. The *advantage function* $A(s,a_i)$ describes the advantage of the action a_i compared with the other possible actions. Therefore, in the dueling architecture, the last layer of the DDQN is split into two subnetworks to estimate $V(s)$ and $A(s,a_i)$ separately. Then, by combining $V(s)$ and $A(s,a_i)$, the *action value function* $Q_i(s,a_i)$ can be estimated. This dueling architecture can lead to better policy evaluation.

Then, D3QN is extended to a multi-agent DRL. Similar to Section III-B, the multi-agent D3QN method is proposed by combining D3QN with MAQL. The detailed procedure of multi-agent D3QN is presented in Algorithm 2. More specifically, in each step, all UEs estimate the Q-values with (s,a_i) for action a_i. The ϵ-greedy policy can be utilized to select the action a_i. Next, each UE i sends its request about the index of the required channel to BS l with which the UE has chosen to associate. If the BS l accepts the request of the i^{th} UE, the BS l will send feedback signal to indicate the available channel and the time-slot to the i^{th} UE. Otherwise, the BS l will not reply anything. Then, when obtaining the immediate reward $u_i(s,a_i)$ and next state s', the information $(s,a_i,u_i(s,a_i),s')$ can be stored at the replay memory. At the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2933417, IEEE Transactions on Wireless Communications

end of each step, all UEs update the weights θ of the DDQN by randomly sampling mini-batch from $\mathcal{D}$. Each episode ends when the QoS of all UEs is satisfied or when the maximum step T is reached. Experimentally we have found $T = 500$ to be a good choice.

Since deep learning algorithm is mainly dependent on the hyperparameters, it is challenging to guarantee the convergence of the proposed multi-agent DRL method by using analytical approaches. Moreover, considering that it may be impossible to know in advance the optimal configuration of hyperparameters for the specific problem, a trial and error procedure is required generally. This is a typical problem in the literature to prove the optimality and convergence analytically. Therefore, we limit the convergence analysis by providing simulation results in Section V, which is also employed in the similar literatures [25], [27], [30], [34]. The learning rate, the hidden layer structure and the training optimization algorithm are the important hyperparameters for the convergence. By choosing the reasonable hyperparameters, convergence can be achieved in the proposed multi-agent DRL method.

Next, we analyze the convergence of our proposed algorithm. In our proposed game, each UE's reward is bounded. Both UEs' number and state-action space are finite. Thus, this game is a finite game. Moreover, since the proposed multi-agent DRL method uses replay memory strategy, necessary historical state information can be stored. Through message passing, each UE can obtain the global state information and other UEs' actions and then choose suitable action accordingly. Therefore, from Seltens theorem, we can derive that an SPNE exists in each finite game with perfect memory [38]. In the proposed game, all UEs want to obtain their own maximum expected discounted rewards at each step. Then, our proposed MARL optimization algorithm can guarantee to converge to the SPNE in our proposed game.

V. Performance Evaluation

A. Simulation Settings

In the simulation, the network comprises 2 MBSs, 8 PBSs, 16 FBSs and 50 UEs with the radius of BSs $500m$, $100m$ and $30m$, respectively, as shown in Fig. 3. The simulation parameters to validate the proposed solution are defined in Table I. Assume that the equivalent profit ρ_i follows the uniform distribution in $[0, 1]$. We first evaluate the training efficiency of our learning method with the fixed location of UEs. Then, the proposed learning method is evaluated in the mobile scenario by comparing with other optimization methods.

In our simulations, the structure of our D3QN is shown in Fig. 4. The structure is composed of an input layer (50 neurons), three hidden layers (64, 32 and 32 neurons), and an output layer (LK neurons). The hyperparameters of D3QN are given in Table II. We use ReLU function as the activation function. The ϵ-greedy policy is used with ϵ linearly chosen from 0 to 0.9. In the weight updating process, the optimal stochastic gradient descent is obtained with the RMSProp optimization approach [46].

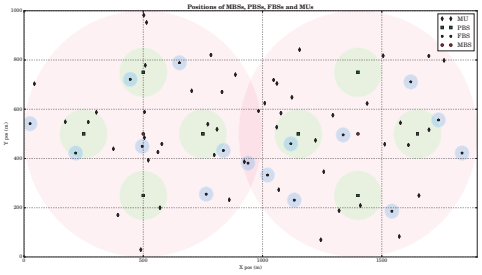


Fig. 3. Network layout with MBSs, PBSs, FBSs and UEs.

TABLE I
Basic Radio Environment Parameters

Parameter	Value
Channel bandwidth W	180 kHz
Downlink center frequency f	2 GHz
Number of UEs N	50
Number of channels K	30
Transmit power of BSs	$\{40, 30, 20\}$ dBm
Radius of cells	$\{500, 100, 30\}$ m
Path loss model for MBS/PBS	$34 + 40\log(d)$
Path loss model for FBS	$37 + 30\log(d)$
Noise power density N_0	-174 dBm/Hz
Action-selection cost Ψ_i	10^{-3}
Minimum QoS requirement Ω_i	5dB
Unit price per transmit power λ_l	$5 * 10^{-4}$

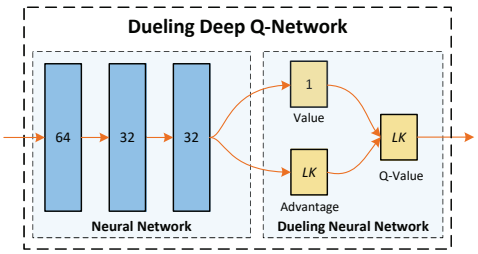


Fig. 4. Three hidden layers with dueling architecture used in our D3QN.

TABLE II
Hyperparameters of D3QN

Parameter	Value
Episodes EP	500
Steps T	500
Mini-batch size	8
Discount rate γ	0.9
Maximum ϵ	0.9
ϵ*-greedy increment*	0.003
Learning rate δ	0.1
Replay memory $\mathcal{D}$ *size*	500
Optimizer	RMSProp
Activation Function	ReLU

B. Simulation Results

1) Training Efficiency with Different Learning Hyperparameters: In this section, D3QN method is evaluated with various learning rates δ, shown in Fig. 5. At the beginning of the learning process, training steps are very large in all cases.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2933417, IEEE Transactions on Wireless Communications

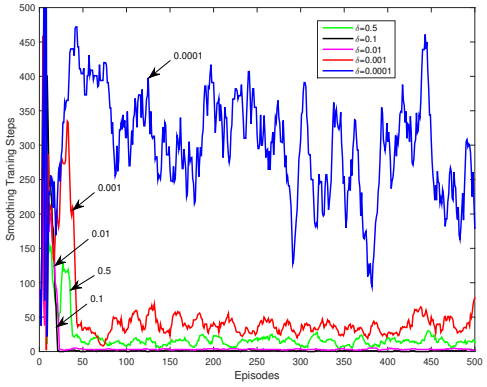


Fig. 5. Smoothing training steps with different learning rates δ.

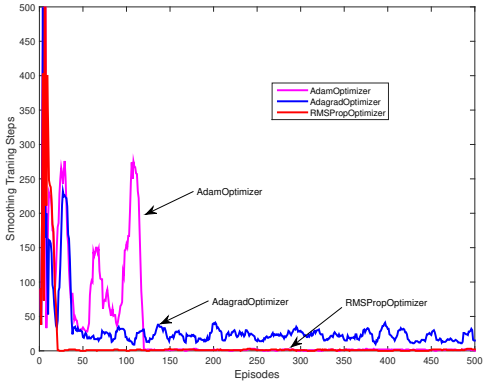
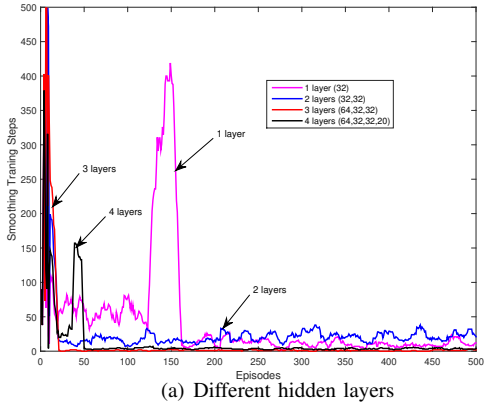


Fig. 6. Smoothing training steps with different optimization strategies.

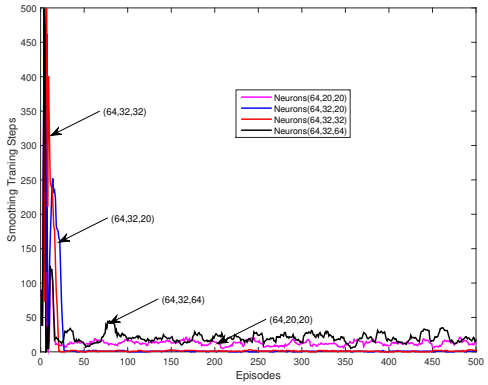
With the number of episodes increasing, the convergence speed tends to increase. Moreover, as δ increases, fewer training steps are required to meet all UEs' QoS requirements. The learning convergence is faster for $\delta = 0.01$ than that of $\delta = 0.001$ and $\delta = 0.0001$. The smoothing training steps converge to less than 5 within 30 episodes when $\delta = 0.1$. However, a larger learning rate may result in local optimum instead of global optimum. When $\delta = 0.5$, the learning convergence is slower than that of $\delta = 0.1$. Thus, considering practical real-time execution of the algorithm, δ is hence chosen to be 0.1.

Then, the performance of D3QN method is evaluated with various optimization strategies. Fig. 6 shows the smoothing training steps with the various optimization strategies. At the beginning of the learning process, training steps are very large in all the three cases. With the number of episodes increasing, the speed of convergence tends to increase. The learning convergence is fastest with the RMSProp optimization algorithm. The learning curve converges to less than 5 steps after 30 episodes. Therefore, the RMSProp optimization strategy is chosen in our D3QN method.

Figure 7 shows the smoothing training steps with the various DNN structure layers. As the number of hidden layers increases, we can obtain the faster convergence speed. The convergence is the fastest with the 3 layers. However, when the number of the hidden layers becomes too large, the optimization problem tends to terminate at a local optimum. Thus, we need to choose an appropriate hidden layers in a specific problem. The 3-hidden-layer is chosen in the D3QN structure. Moreover, with an increase of the number of neurons, the convergence speed tends to increase. However, when the number of neurons becomes too large, the optimization problem may result in overfitting and more training time. Therefore, 3 layers with 64, 32 and 32 neurons are chosen in the D3QN structure.



(a) Different hidden layers



(b) Different neurons.

Fig. 7. Learning curves with different DNN structure.

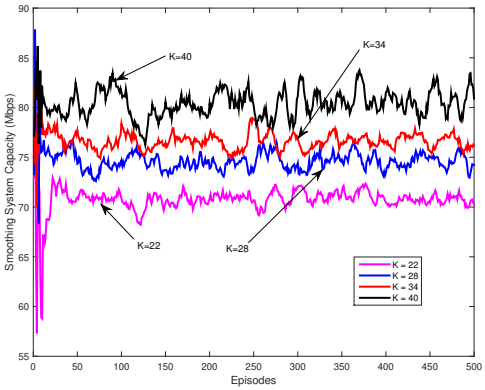


Fig. 8. Smoothing training steps with different numbers of channels K.

2) Training Efficiency with Different Scenarios: Next, the performance of D3QN method is evaluated with the different numbers of channels K. Figure 8 demonstrates that the number of smoothing training steps is decreasing in the number of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2933417, IEEE Transactions on Wireless Communications

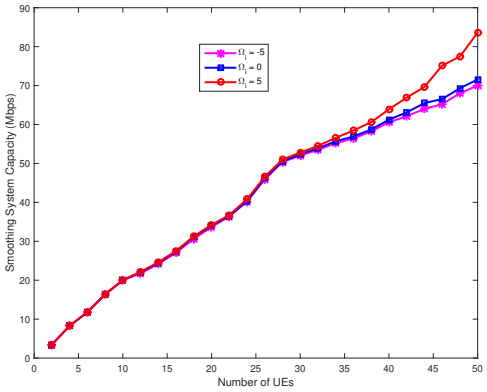
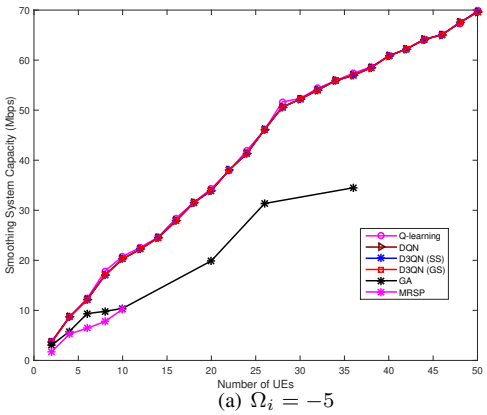
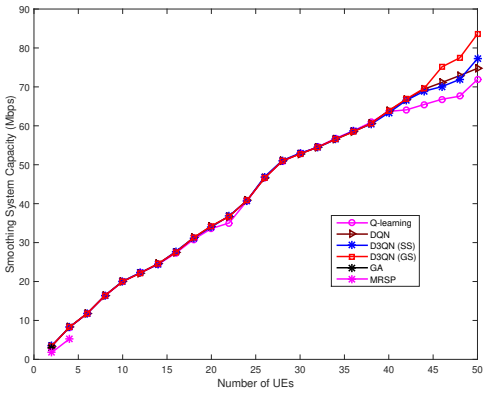


Fig. 9. Smoothing system capacity with different numbers of UEs N.



(a) $\Omega_i = -5$



(b) $\Omega_i = 5$

Fig. 10. Smoothing system capacity with different numbers of UEs N.

channels K. Specifically, the convergence is faster when $K = 26$ than that when $K = 22$ and $K = 24$. The learning curve converges within less than 5 steps after 20 episodes when $K = 26$. As K increases, the average number of UEs per transmission channel decreases and the total SINR of each UE increases. Thus, it will be easier to meet the UEs' QoS requirement with fewer training steps. However, when the number of channels K becomes too large, the size of action space increases. Then, it will require much more training steps to meet all UEs' QoS requirements.

Figure 9 plots the learning curves of D3QN method with the various numbers of UEs N and different minimum QoS requirement of UEs Ω_i. With the growing number of UEs N, we achieve the higher system capacity. Moreover, when N is small, fewer training steps are needed to meet all UEs' QoS requirements, which results in the similar system capacity in the three cases of Ω_i. However, as the number of UEs N increases, the co-channel interference increases. Thus, more training steps are required to meet all UEs' QoS requirements. At this time, the most system capacity is achieved with the largest QoS requirement of all UEs. This explains why the lower three curves in Fig. 8 first coincide with each other and then increase in Ω_i.

Then, the training performance with various MARL methods is analyzed. For comparison, the common MAQL and MA-DQN methods are considered. The multi-agent D3QN strategy without message passing is also considered, which is denoted as $D3QN(SS)$. Our proposed D3QN method is denoted as $D3QN(GS)$. Moreover, genetic algorithm (GA) is considered, which is the common method to solve nonlinear integer programming problem. The traditional maximum received signal power (MRSP) criterion is also used as the joint UARA strategy.

Figure 10 plots the learning curves of the six methods with the various numbers of UEs N and different minimum QoS requirement of UEs Ω_i. As the number of UEs N and the minimum QoS requirement of UEs Ω_i increase, the smoothing system capacity increases in all optimization methods. Compared with the four learning methods ($D3QN(GS)$, $D3QN(SS)$, DQN and Q-learning), GA and MRSP approaches can performance well in certain case of N and small Ω_i. For example, GA method only finds the suitable joint UARA strategy to meet all UEs' QoS requirements $\Omega_i = 5$ when $N = 2$.

Moreover, by comparing the four learning methods, it can be seen that, when N or Ω_i is small, fewer training steps are needed to meet all UEs' QoS requirements, which results in the similar system capacity in the four MARL methods. However, with the growing number of UEs N, since the QoS requirements of all UEs are guaranteed, we can achieve the higher system capacity with the four MARL methods. The $D3QN(GS)$ strategy also achieves the highest system capacity among the four MARL methods with different numbers of UEs N.

3) Optimization Performance with Different Strategies in Mobile HetNets Scenario: Finally, the optimization performance in the mobile HetNets is evaluated. In each episode, each UE walks randomly with a normal distribution movement. The small-scale fading is modeled as the Rayleigh fading with unit scale. For comparison, the above six optimization methods are considered. Moreover, we compare the CPU time (second) in computations with the different optimization algorithms. The simulation is implemented on a Windows 10 with Intel Core i7 CPU and 8GB memory. We evaluate the GA's performance in Matlab, and the other five methods in python.

The performance of average system capacity (ASC), average network utility (ANU) and average computational time (ACT) with the above six optimization algorithms are presented in Table III ($\Omega_i = -15$), Table IV ($\Omega_i = -5$) and Table V

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2933417, IEEE Transactions on Wireless Communications

TABLE III
AVERAGE SYSTEM CAPACITY, AVERAGE NETWORK UTILITY AND COMPUTATIONAL PERFORMANCE ($\Omega_i = -15$)

Method	$N = 30$			$N = 40$			$N = 50$		
	ASC (Mbps)	ANU	ACT (sec.)	ASC (Mbps)	ANU	ACT (sec.)	ASC (Mbps)	ANU	ACT (sec.)
D3QN (GS)	52.51	23.53	11.73	**61.26**	**27.00**	19.40	69.97	30.67	31.95
D3QN (SS)	52.51	23.53	12.66	**61.26**	**27.00**	19.19	69.97	30.67	29.91
DQN	52.51	23.53	11.84	**61.26**	**27.00**	**17.50**	69.97	30.67	**27.52**
Q-Learning	52.83	23.68	27.00	61.12	26.83	40.13	**70.49**	**30.97**	61.69
GA	38.11	19.38	$3.75e^3$	42.95	21.99	$4.61e^3$	48.39	24.50	$1.39e^4$
MRSP	**68.15**	**31.71**	**10.76**	–	–	–	–	–	–

TABLE IV
AVERAGE SYSTEM CAPACITY, AVERAGE NETWORK UTILITY AND COMPUTATIONAL PERFORMANCE ($\Omega_i = -5$)

Method	$N = 30$			$N = 40$			$N = 50$		
	ASC (Mbps)	ANU	ACT (sec.)	ASC (Mbps)	ANU	ACT (sec.)	ASC (Mbps)	ANU	ACT (sec.)
D3QN (GS)	53.04	**23.73**	14.85	62.38	27.60	36.29	72.03	31.67	100.69
D3QN (SS)	53.04	**23.73**	15.38	62.38	27.60	33.93	72.03	31.67	103.46
DQN	53.04	**23.73**	**13.35**	62.38	27.60	**32.74**	72.03	31.67	**94.53**
Q-learning	**53.05**	23.67	33.94	**62.89**	**27.98**	79.00	**72.19**	**31.74**	318.94
GA	36.51	18.33	$3.96e^3$	–	–	–	–	–	–
MRSP	–	–	–	–	–	–	–	–	–

TABLE V
AVERAGE SYSTEM CAPACITY, AVERAGE NETWORK UTILITY AND COMPUTATIONAL PERFORMANCE ($\Omega_i = 5$)

Method	$N = 30$			$N = 40$			$N = 50$		
	ASC (Mbps)	ANU	ACT (sec.)	ASC (Mbps)	ANU	ACT (sec.)	ASC (Mbps)	ANU	ACT (sec.)
D3QN (GS)	54.74	24.60	45.93	**68.42**	**30.74**	**187.96**	**81.30**	**36.33**	$\mathbf{3.59e^3}$
D3QN (SS)	54.74	24.60	46.87	67.30	30.09	217.20	79.09	35.01	$3.78e^3$
DQN	54.74	24.60	**42.08**	67.46	30.15	432.80	78.43	34.96	$3.60e^3$
Q-learning	**54.82**	**24.78**	259.26	66.69	29.67	$3.94e^3$	20.32	9.03	$2.09e^5$
GA	–	–	–	–	–	–	–	–	–
MRSP	–	–	–	–	–	–	–	–	–

($\Omega_i = 5$). As the number of UEs N and the minimum QoS requirement of UEs Ω_i increase, ASC, ANU and ACT increase in all optimization methods. Compared with the four learning methods ($D3QN(GS)$, $D3QN(SS)$, DQN and Q-learning), GA and MRSP approaches can only performance well in the case of small N and Ω_i. When N or Ω_i is large, these two methods may not find the suitable joint UARA strategy to meet all UEs' QoS requirements. In this case, we denote it as "–" in Table III-Table V. Even though GA can find the suitable strategy shown in Table III, the computational complexity is great when $\Omega_i = -15$.

Moreover, it can be observed that the learning methods in the fixed network environment can also be used for the joint UARA issue in the mobile scenario. Compared with the performance in the fixed scenario, all the four learning methods ($D3QN(GS)$, $D3QN(SS)$, DQN and Q-learning) achieve the less system capacity and network utility in the mobile scenario. The proposed learning strategy shows good generalization ability and low computational time cost.

Furthermore, by comparing the four learning methods, when N or Ω_i is small, less learning time is required to meet all UEs' QoS requirements, which results in similar system capacities in all four learning approaches. Especially, due to similar DNN models, $D3QN(GS)$, $D3QN(SS)$ and DQN approaches achieve the similar performance in the case of small N and Ω_i. This is similar to the performance in Fig. 9 and Fig. 10. However, when N or Ω_i is large, our proposed $D3QN(GS)$ method can achieve better performance on system capacity and network utility with lower computational time. For example, $D3QN(GS)$ method obtains the highest system capacity and network utility when $N = 40$ and $N = 50$ with the lowest computational time cost in Table V. This indicates that the proposed $D3QN(GS)$ method has an advantage in solving large-scale learning problems.

VI. CONCLUSION

In this paper, the distributed multi-agent DRL method has been proposed to obtain the jointly optimal UARA strategy of the HetNets. The optimization problem has been formulated to achieve the maximum the long-term downlink utility while guaranteeing the UEs' QoS requirements. Considering the non-convex and combinatorial characteristics of this joint optimization problem, we have proposed the MARL method by jointly associating UEs to BSs and allocating channels to UEs. Moreover, considering the large action space in the MARL approach, D3QN has been proposed to obtain the optimal policy with little computation complexity. Through message passing, the distributed UEs obtain information about network state with a little communication overhead. With the help of double-Q strategy and dueling architecture, D3QN can converge to a SPNE with a small number of iterations. Simulation results are given to indicate that the proposed method has outperformed the other reinforcement learning methods with faster convergence speed and better generalization ability.

REFERENCES

[1] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, UAE, Dec. 2018, pp. 1-6.

[2] Y. Huang, J. Tan, and Y.-C. Liang, "Wireless big data: transforming heterogeneous networks to smart networks," *J. Commun. and Inf. Networks*, vol. 2, no. 1, pp. 19-32, 2017.

[3] S. Y. Lien, S. C. Hung, K. C. Chen, and Y.-C. Liang, "Ultra-low-latency ubiquitous connections in heterogeneous cloud radio access networks," *IEEE Wireless Commun.*, vol. 22, no. 3, pp. 22-31, 2015.

[4] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706-2716, 2013.

[5] K. Shen and W. Yu, "Distributed pricing-based user association for downlink heterogeneous cellular networks," *IEEE J. Sel. Areas in Commun.*, vol. 32, no. 6, pp. 1100-1113, 2014.

[6] D. Fooladivanda and C. Rosenberg, "Joint resource allocation and user association for heterogeneous wireless cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 1, pp. 248-257, 2013.

[7] Y. Lin, W. Bao, W. Yu, and B. Liang, "Optimizing user association and spectrum allocation in HetNets: A utility perspective," *IEEE J. Sel. Areas in Commun.*, vol. 33, no. 6, pp. 1025-1039, 2014

[8] Y. Chen, J. Li, W. Chen, Z. Lin, and B. Vucetic, "Joint user association and resource allocation in the downlink of heterogeneous networks," *IEEE Trans. Veh. Tech.*, vol. 65, no. 7, pp. 5701-5706, 2016.

[9] Q. Han, B. Yang, G. Miao, C. Chen, X. Wang, and X. Guan, "Backhaul-aware user association and resource allocation for energy-constrained HetNets," *IEEE Trans. Veh. Tech.*, vol. 66, no. 1, pp. 580-593, 2017.

[10] S. Bayat, R. Louie, Z. Han, B. Vucetic, and Y. Li, "Distributed user association and femtocell allocation in heterogeneous wireless networks," *IEEE Trans. Commun.*, vol. 62, no. 8, pp. 3027-3043, 2014.

[11] A. Elsherif, W.-P. Chen, A. Ito, and Z. Ding, "Resource allocation and inter-cell interference management for dual-access small cells," *IEEE J. Sel. Areas in Commun.*, vol. 33, no. 6, pp. 1082-1096, 2015.

[12] M. Chen, S. C. Liew, Z. Shao, and C. Kai, "Markov approximation for combinatorial network optimization," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6301-6327, 2013.

[13] M. Simsek, M. Bennis, and A. Czylwik, "Dynamic inter-cell interference coordination in HetNets: A reinforcement learning approach," *in Proc. IEEE Global Commun. Conf. (GLOBECOM),* 2012, pp. 5446-5450.

[14] N. Zhao, Y.-C. Liang, and Y. Pei, "Dynamic contract incentive mechanism for cooperative wireless networks," *IEEE Trans. Veh. Tech.*, vol. 67, no. 11, pp. 10970-10982, 2018.

[15] Y. Sun, G. Feng, S. Qin, Y.-C. Liang, and T. P. Yum, "The SMART handoff policy for millimeter wave heterogeneous cellular networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 6, pp. 1456-1468, 2018.

[16] N. Zhao, Y. Chen, R. Liu, M. Wu, and W. Xiong, "Monitoring strategy for relay incentive mechanism in cooperative communication networks," *Computers & Electrical Engineering*, vol. 60, pp. 14-29, 2017.

[17] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279-292, 1992.

[18] K.-L. A. Yau, P. Komisarczuk, and P. D. Teal, "Reinforcement learning for context awareness and intelligence in wireless networks: Review, new features and open issues," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 253-267, 2012.

[19] A. Shahid, S. Aslam, H. S. Kim, and K. G. Lee, "A docitive Q-learning approach towards joint resource allocation and power control in self-organised femtocell networks," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 2, pp. 216-230, 2015.

[20] A. Asheralieva and Y. Miyanaga, "An autonomous learning-based algorithm for joint channel and power level selection by D2D pairs in heterogeneous cellular networks," *IEEE Trans. Commun.*, vol. 64, no.9, pp. 3996-4012, 2016.

[21] Z. Li, C. Wang, and C. J. Jiang, "User association for load balancing in vehicular networks: an online reinforcement learning approach," *IEEE Trans. Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2217-2228, 2017.

[22] E. Ghadimi, F. D. Calabrese, G. Peters, and P. Soldati, "A reinforcement learning approach to power control and rate adaptation in cellular networks," *in Proc. IEEE Int. Conf. Commun. (ICC),* 2017, pp. 1-7.

[23] J. Tan, S. Xiao, S. Han, and Y.-C. Liang, "A learning-based coexistence mechanism for LAA-LTE based HetNets" *in Proc. IEEE Int. Conf. Commun. (ICC),* 2018, pp. 1-6.

[24] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, et al, and S. Petersen, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, 2015.

[25] U. Challita, L. Dong, and W. Saad, "Proactive resource management for LTE in unlicensed spectrum: A deep learning perspective," IEEE Trans. Wireless Commun., vol. 17, no. 7, pp. 4674-4689, 2018.

[26] X. Li, J. Fang, W. Cheng, H. Duan, Z. Chen, and H. Li, "Intelligent power control for spectrum sharing in cognitive radios: A deep reinforcement learning approach," *IEEE Access*, vol. 6, pp. 25463-25473, 2018.

[27] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in hetnets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 680-692, 2018.

[28] H. Ye, Y. G. Li and, B. F. Juang, "Deep reinforcement learning for resource allocation in V2V communications," *IEEE Trans. Veh. Tech.*, vol. 68, no. 4, pp. 3163-3173, 2019.

[29] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs," in *IEEE International Conference on Communications (ICC)*, Paris, 2017, pp. 1-6.

[30] Y. Sun, M. Peng, and S. Mao, "Deep reinforcement learning based mode selection and resource management for green fog radio access networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1960-1971, 2019.

[31] L. Xiao, Y. Li, C. Dai, H. Dai, and H. V. Poor, "Reinforcement learning-based NOMA power allocation in the presence of smart jamming," *IEEE Trans. Veh. Tech.*, vol. 67, no. 4, pp. 3377-3389, 2018.

[32] L. Xiao, Y. Li, X. Huang, and X. Du, "Cloud-Based malware detection game for mobile devices with offloading," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2742-2750, 2017.

[33] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 257-265, 2018.

[34] Y. He, F. R. Yu, N. Zhao, V. C. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Commun. Magazine*, vol. 55, no. 12, pp. 31-37, 2017.

[35] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310-323, 2019.

[36] Y. S. Nasir and D. Guo, "Multi-Agent deep reinforcement learning for dynamic power allocation in wireless networks," arXiv:1808.00490.

[37] A. Neyman and S. Sorin, Eds., Stochastic games and applications. Dordrecht, The Netherlands: Kluwer, 2003.

[38] M. J. Osborne, An introduction to game theory. Oxford University Press, 2004.

[39] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT Press Cambridge, 1998.

[40] Y. Fan and H. Li, "Distributed approximating global optimality with local reinforcement learning in HetNets," *in Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2017, pp. 1-7.

[41] G. Tesauro, "Extending Q-learning to general adaptive multi-agent systems," *in Advances in Neural Information Processing Systems*, 2004, pp. 871-878.

[42] L. Busoniu, R. Babuska, and B. D Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 2, pp. 156-172, 2008.

[43] C. Szepesvri, and M. L. Littman, "A unified analysis of value-function based reinforcement-learning algorithms," *Neural Comput.*, vol. 11, no. 8, pp. 2017-2060, 1999.

[44] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," *arXiv preprint arXiv:1509.06461*, 2015.

[45] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," *arXiv preprint arXiv:1511.06581*, 2015

[46] T. Tieleman and G. Hinton, "Lecture 6.5 RmsProp: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, 2012.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2019.2933417, IEEE Transactions on Wireless Communications

Nan Zhao (M'17) received the B.S., M.S. and Ph.D. degrees from Wuhan University, Wuhan, China in 2005, 2007 and 2013, respectively. She is currently an Associate Professor in the Hubei University of Technology, Wuhan, China, and also works as a postdoctoral research fellow at the the University of Electronic Science and Technology of China. Her current research involves machine learning in wireless communications and cognitive radio.

Ying-Chang Liang (F'11) is currently a Professor with the University of Electronic Science and Technology of China, China, where he leads the Center for Intelligent Networking and Communications and serves as the Deputy Director of the Articial Intelligence Research Institute. He was a Professor with The University of Sydney, Australia, a Principal Scientist and Technical Advisor with the Institute for Infocomm Research, Singapore, and a Visiting Scholar with Stanford University, USA. His research interests include wireless networking and communications, cognitive radio, symbiotic radio, dynamic spectrum access, the Internet-of-Things, articial intelligence, and machine learning techniques.

Dr. Liang has been recognized by Thomson Reuters (now Clarivate Analytics) as a Highly Cited Researcher since 2014. He received the Prestigious Engineering Achievement Award from The Institution of Engineers, Singapore, in 2007, the Outstanding Contribution Appreciation Award from the IEEE Standards Association, in 2011, and the Recognition Award from the IEEE Communications Society Technical Committee on Cognitive Networks, in 2018. He is the recipient of numerous paper awards, including the IEEE Jack Neubauer Memorial Award, in 2014, and the IEEE Communications Society APB Outstanding Paper Award, in 2012. He is a Fellow of the IEEE, and a Foreign Member of Academia Europaea.

He is the Founding Editor-in-Chief of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS: COGNITIVE RADIO SERIES, and the Key Founder and now the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He is also serving as an Associate Editor-in-Chief for China Communications. He served as a Guest/Associate Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS, the IEEE Signal Processing Magazine, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORK. He was also an Associate Editor-in-Chief of the World Scientic Journal on Random Matrices: Theory and Applications. He was a Distinguished Lecturer of the IEEE Communications Society and the IEEE Vehicular Technology Society. He was the Chair of the IEEE Communications Society Technical Committee on Cognitive Networks, and served as the TPC Chair and Executive Co-Chair of the IEEE Globecom'17.

Dusit Niyato (M'09-SM'15-F'17) received the B.Eng. degree from the King Mongkuts Institute of Technology Ladkrabang in 1999 and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Canada, in 2008. He is currently a Full Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in the areas of green communications, the Internet of Things, and sensor networks.

Yiyang Pei (S'09-M'12) received her B.E. degree and PhD degree in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2007 and 2012, respectively. She is currently an Assistant Professor in Singapore Institute of Technology. She was a research scientist in Institute for Infocomm Research, Singapore from 2012 to 2016. Her research interests are in the general area of wireless communications, with current emphasis on resource allocation in cognitive radio networks.

Minghu Wu received the B.S. degree in electronic information engineering from Communication University of China, Beijing, China, the M.S. degree in communication information system from Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree in Nanjing University of Posts and Telecommunications, Nanjing, China, in 1998, 2002 and 2013, respectively. He is currently a Professor in Hubei University of Technology. His major research interests include communication signal processing and video coding.

Yunhao Jiang received the M.S. and Ph.D. degrees in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2010, respectively. From 2011 to 2014, he was a Lecturer at the Nanjing University of Information Science and Technology, Nanjing, China. He is currently a Associate Professor at the Hubei University of Technology, Wuhan. His research interests include electromagnetic compatibility of power conversion and interference suppression.