Deep Reinforcement Learning Based Beamforming for Throughput Maximization in Ultra-Dense Networks

Huihan Yu, Yang Xiao, Jiawei Wu, Zilong He, Fang Liu, and Jun Liu
School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China
Email: {yhh1881021, zackxy, cloudsae, hznsmi, lindaliu, liujun}@bupt.edu.cn

2022 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-6654-4266-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/WCNC51071.2022.9771887

***Abstract*—Ultra-dense network (UDN) is a promising technology for 5G and beyond communication systems to meet the requirements of explosive data traffic. However, the dense distribution of wireless terminals potentially leads to severe interference and deteriorate network performance. To address this issue, beamforming is widely used to coordinate the interference in UDNs and improve receive gains by controlling the phase of multiple antennas. In this paper, we propose a multi-agent deep reinforcement learning (DRL) based beamforming algorithm to achieve more dynamic and fast beamforming adjustment. In the proposed algorithm, the agents inside beamforming controllers are distributively trained while exchanging partial channel state information (CSI) for better optimizing beamforming vectors to achieve maximized throughputs in UDNs. The evaluation results demonstrate that the proposed algorithm significantly improves the computation efficiency, as well as achieves the highest network throughput compared to several baselines.**

***Index Terms*—Ultra-dense network (UDN), deep reinforcement learning (DRL), multi-agent learning, beamforming.**

I. Introduction

The ultra-dense network (UDN) is regarded as one of the most promising technologies to satisfy the rapidly increasing demands for modern communications. However, cellular users will suffer from severe inter-cell interference when the network devices are deployed densely. To address the issue, the beamforming technology is introduced to coordinate and manage the interference. Beamforming has been applied to relieve the inter-cell interference and improve the network throughput [1]. In general, the problem of making optimal beamforming decision to achieve maximized network throughput is NP hard [2]. To tackle the problem, a number of sub-optimal methods were proposed, i.e., the fractional programming (FP) [3]. However, these methods require the real-time global channel state information (CSI), which are computationally expensive owing to the large number of antennas in typical UDNs. In addition, these methods adopt centralized structures, where a central controller collects the global CSI and designates the beamformers to base stations (BSs). The denser the UDN is, the stricter constraints are imposed on the beam selection scheme [4]. Therefore, these traditional beamforming algorithms are difficult to agilely adapt the dynamic network environment owing to the low computation efficiency.

Meanwhile, deep reinforcement learning (DRL) has been proved to be an effective technology for solving decision making problem in complex environment. Therefore, some researchers try to utilize DRL to improve the adaptability and computation efficiency of beamforming algorithm. In [4], Sun *et al.* firstly proposed a single-agent DRL-based algorithm for beamforming that aimed at maximizing the throughput of the UDN. However, with the increase of micro cell density, the performance would go down rapidly because of the increase of computation workload. To further reduce the computation workload, Zhang *et al.* [5] proposed a multi-agent DRL-based algorithm for beamforming that improves the network throughput and the computation efficiency. Inspired by above works, we design and implement a multi-agent DRL based algorithm for beamforming in UDNs. To the best of our knowledge, it is the first multi-agent DRL beamforming algorithm in UDNs. The proposed algorithm can guarantee the throughput for the primary users, which is not considered in prior research works. Apart from providing guaranteed communication services for the primary users, our proposed algorithm effectively improves the throughput of the UDN and the computation efficiency. The main contributions of this work are summarized as follows:

- We proposed a muti-agent DRL-based algorithm to improve the beamforming computation efficiency drastically. Agents train their deep Q-networks (DQNs) and execute the actions distributedly. Each agent only needs to exchange a part of the global CSI during the entire training phase.
- We designed a specific reward function to make sure the transmission rates of primary users are larger than a given threshold, and avoid to generate too much interference to secondary users.

The remainder of this paper is organized as follows: Section II introduces the system model and expounds the problem formulation. Section III presents the fundamental RL concepts and our proposed distributed DRL-based beamforming algorithm. In Section IV, we evaluate the performance of the proposed algorithm. Finally, Section V concludes the paper.

II. System Model and Problem Formulation

A. System Model

In this paper, we take a downlink orthogonal frequency division multiple access (OFDMA) UDN scenario as an example,

(*Corresponding author: Fang Liu.*)

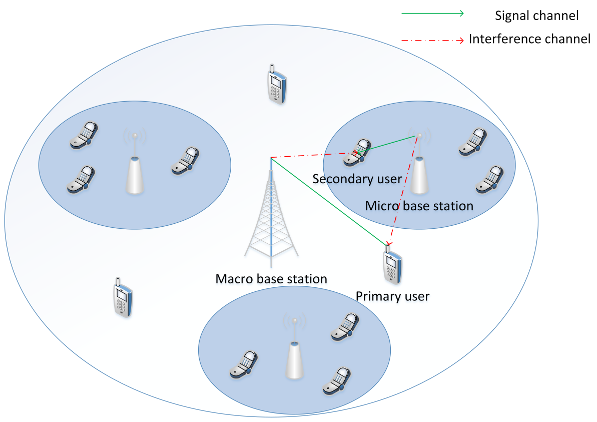


Fig. 1. System model of the example UDN.

which is shown in Fig. 1. In the example UDN, L micro cells are randomly distributed in a macro-cell coverage area. The resource blocks (RBs) assigned to users in the same cell are orthogonal to each other. Therefore, there is no interference between users in the same cell. As a result, we simplify the model by assigning only one user in each cell. We assume that all the users share the same RB, so the inter-cell interference is inevitable. The system can be modeled as a UDN multiple-input single-output (MISO) model, where BSs equipped with N antennas serving their associated users each with a single antenna. Hence, the received signal of the b-th receiver at time t can be written as follows:

$$y_b(t) = \mathbf{h}^{\dagger}_{b,b}(t)\mathbf{w}_b(t)x_b(t) + \sum_{c\neq b}\mathbf{h}^{\dagger}_{c,b}(t)\mathbf{w}_c(t)x_c(t) + z_b(t), \quad b, c \in \mathcal{B} = \{0, ..., L\}, \tag{1}$$

where $\mathbf{h}^{\dagger}$ denotes the Hermitian transposition of $\mathbf{h}$. $\mathbf{h}_{b,b}(t) \in \mathcal{C}^{N\times 1}$ denotes the direct channel gain between BS b and its associated user equipment (UE) b. Note that $\mathbf{h}_{b,b}(t)$ includes both the large-scale fading factor and the small-scale fading factor. $\mathbf{h}_{c,b}(t) \in \mathcal{C}^{N\times 1}$ denotes the interference channel between BS c and UE b. $\mathbf{w}_b(t) \in \mathcal{C}^{N\times 1}$ denotes the beamformer of BS b. $x_b(t)$ denotes the signal transmitted by BS b, and $z_b(t) \sim \mathcal{CN}(0, \sigma^2)$ denotes the Gaussian noise at UE b. The instantaneous achievable rate under the unit bandwidth of UE b at time t is expressed as:

$$R_b(\mathbf{W}(t)) = \log(1 + \frac{|\mathbf{h}^{\dagger}_{b,b}(t)\mathbf{w}_b(t)|^2}{\sum\limits_{c\neq b}|\mathbf{h}^{\dagger}_{c,b}(t)\mathbf{w}_c(t)|^2 + \sigma^2}), \quad b, c \in \mathcal{B} = \{0, ..., L\}, \tag{2}$$

where $\mathbf{W}(t) = [\mathbf{w}_0(t), \mathbf{w}_1(t), ..., \mathbf{w}_L(t)] \in \mathcal{C}^{N\times B}$ is the vector of the beamformers chosen by all BSs.

B. Problem Formulation

A number of works like [6], [7] have illustrated that the average transmission rate of users will go down when the density of micro cells is too high. However, users may have distinct tolerabilities of the communication quality. The primary users generally have the minimum transmission rate limit, while secondary users do not. Therefore, we consider the throughput maximization problem of micro cells while guaranteeing communication quality of the primary user as the optimization goal in this paper. Specifically, the optimization goal in time slot t can be formulated as follows:

$$\max_{\mathbf{W}(t)} \quad \sum_{d=1}^{L} R_d(\mathbf{W}(t)), \tag{3a}$$

$$\begin{aligned} s.t. \quad & R_0(\mathbf{W}(t)) \geq R_{\psi}, \\ & \|\mathbf{w}_0(t)\|^2 \leq P_{max}, \\ & \|\mathbf{w}_d(t)\|^2 \leq p_{max}, \quad d \in \{1, ..., L\}, \end{aligned} \tag{3b}$$

where P_{max} and p_{max} denote the maximum transmission power conducted by the BSs in the macro cell and micro cells, respectively. $R_d(\mathbf{W}(t))$ and $R_0(\mathbf{W}(t))$ denote the transmission rates of the primary user and the secondary user, respectively. R_{ψ} denotes the minimum transmission rate of the primary user. The defined problem is NP-hard, which is difficult for heuristic methods to find the optimal solution under the time constraints. To solve this problem, we design and implement a multi-agent DRL-based algorithm, which will be detailedly introduced in the next section.

III. Design of Multi-agent DRL Based Beamforming Algorithm

A. Deep Reinforcement Learning

Deep reinforcement learning (DRL) has following fundamental concepts: state, action, policy, and reward. In each time step t, the agent (i.e., the decision-maker) in state $s \in S$ takes an action $a \in A$ according to the policy π: $s \rightarrow a$. Then, the agent receives a reward r while the state s transfers to the next state $s' \in S$. Q-learning is one of the most classic and widely used reinforcement learning algorithms. In Q-learning, the agent learns the action-value function $Q_{\pi}(s, a)$, which satisfies the Bellman equation:

$$Q^*_{\pi}(s_t, a_t) = \mathbf{E}_{s'}\left[r_{s,s',a} + \gamma \max_{a'} Q^*_{\pi}(s', a') \middle| s_t, a_t \right], \tag{4}$$

where $\gamma : 0 < \gamma < 1$ is the discount factor that determines the importance of the predicted future rewards. a' is the next action. During the training phase, the agent learns the optimal policy π^*. The update rule of Q-learning in its general form is given by

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r_t + \gamma \max_{a'} Q(s', a') - Q(s, a)], \tag{5}$$

where α denotes the learning rate. However, when the state space is large, the vanilla Q-learning algorithm is not applicable owing to the curse of dimensionality [8]. To address this issue, the deep Q-network (DQN) is proposed by introducing deep neural network for action-value function approximation. DQN updates the weights of deep neural networks by learning from the experience. Generally, the parameter set ξ can be optimized by minimizing the following loss function,

$$L_t(\xi_t) = \mathbf{E}[y_t - Q(s_t, a_t; \xi_t)]^2, \tag{6}$$

where $y_t = r_t + \gamma \max_a Q(s', a; \xi_t^-)$ is the objective function, and ξ_t^- is copied from ξ_t per T_{step} steps.

B. Distributed DRL-Based Algorithm

In this paper, we propose a distributed-training-distributed-executing (DTDE) DRL algorithm for beamforming to maximize the throughput of the UDN. Each agent trains its individual DQN, and chooses the optimal beamformer according to its own policy. Note that the environment of each agent will change if other agents execute their actions in a multi-agent system. This non-stationary situation is obviously detrimental to the policy convergence of each agent. One way to solve this problem is to allow the agents to communicate with each other [9]. In our proposed algorithm, historical actions, channel measurements, and transmission rates are exchanged between the agents to help them make appropriate decisions by predicting the behaviors of other agents.

In a DTDE algorithm, each agent will go through two stages: offline training stage and online decision-making stage. In the offline training stage, the agent takes a minibatch from the experience pool to calculate the loss function. Then, the agent utilizes an optimizer to minimize the loss function, and updates the weights of the training DQN. After every T_{step} steps, the weights of the DQN are copied to the target network. As for the online decision-making stage, the agent firstly obtains the state of the environment. Then, the agent takes an action based on its own policy. The pseudocode of the proposed distributed DRL-based DTDE algorithm is shown in Algorithm 1. The key concepts of the proposed algorithm are defined as below.

1) **State**: Inspired by [10], we feed agents with the following input states, which are divided into three categories, i.e., local information, interferers' information and interfered neighbors' information. Taking the *b*th channel directly connecting BS *b* and UE *b* as an example, its interferers are defined as channels whose BSs transmit interference signals to UE *b*. And its interfered neighbors are channels whose UEs are interfered by BS b.

- Local information: There are eight components in local information. We take BS b as an example. The first three components are the transmit power $p_b(t-1)$, the code of beam codebook selected by the BS b e_b, and the transmission rate of UE *b* in time slot $t-1$ $R_b(\mathbf{W}(t-1))$. The fourth and fifth components are the equivalent channel gains, $|\mathbf{h}^{\dagger}_{b,b}(t)\overline{\mathbf{w}}_b(t-1)|^2$ and $|\mathbf{h}^{\dagger}_{b,b}(t-1)\overline{\mathbf{w}}_b(t-2)|^2$, respectively. The sixth and seventh components are the total interference-plus-noise power at UE *b*, $\sum\limits_{c\neq b} |\mathbf{h}^{\dagger}_{c,b}(t)\mathbf{w}_c(t-1)|^2 + \sigma^2$ and $\sum\limits_{c\neq b} |\mathbf{h}^{\dagger}_{c,b}(t-1)\mathbf{w}_c(t-2)|^2 + \sigma^2$, respectively. The eighth one is the channel index in the Jakes model J_b [11].

Algorithm 1 The proposed multi-agent DRL-based algorithm for beamforming in UDN

```
Initialize a replay memory $M_b$ of DQN to capacity $M$ for BS $b$, $\forall b \in \mathcal{B}$;
Initialize a trained network with random parameter $\xi^b$, and initialize target nework with parameter $\xi^{b-} = \xi^b$;
repeat
    Obersve the state $s_b$ in time slot $t$, $\forall b \in \mathcal{B}$;
    Choose an action $a_b$ according to $\epsilon - greedy$ policy:
    if $p > \epsilon$ then
        choose an action $a_b = argmax_{a \in A} Q(s_b, a; \xi^b)$;
    else
        Randomly choose an action $a_b \in A$;
    Perform the action $a_b$ and gets the reward $r_b$;
    Observe next state $s'_b$ in time slot $t+1$;
    Store experience $e[t] = (s_b, a_b, r_b, s'_b)$ into its own experience pool $M_b$;
    Minibatch sample from $M_b$, and update the parameter $\xi^b$ of its trained DQN using Adam optimizer;
    Update $\xi^{b-}$ with $\xi^b$ every $T_{step}$ time slots.
until convergence
```

- Interferers' information: BS b gets the interference information from its interferers. For each interferer $c \in I_b(t)$, there are four components of interference information:

(i) The BS that generates interference to UE *b*, c.
(ii) The interference generated by the BS c, $p_c|\mathbf{h}^{\dagger}_{c,b}\overline{\mathbf{w}}_c|^2$.
(iii) The beam code chosen by BS c, e_c.
(iv) The transmission rate of UE c, $R_c(\mathbf{W})$.

We consider those information in time slot t and $t-1$. There are 8*K* components describing interferers' information since UE *b* has *K* interferers, i.e., $|I_b(t)| = K$.

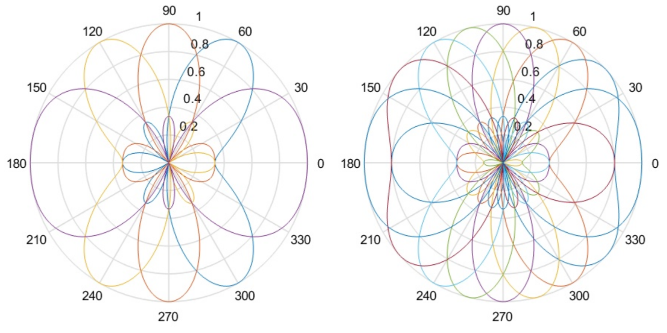
- Interfered neighbors' information: For each interfered neighbor $a \in O_b(t)$, there are three components of interfered neighbors' information, which are $|\mathbf{h}^{\dagger}_{a,a}(t-1)\overline{\mathbf{w}}_a(t-1)|^2$, $R_a(\mathbf{W}(t-1))$, and $\frac{|\mathbf{h}^{\dagger}_{b,a}(t-1)\mathbf{w}_b(t-1)|^2}{\sum\limits_{c\neq a}|\mathbf{h}^{\dagger}_{c,a}(t-1)\mathbf{w}_c(t-1)|^2+\sigma^2}$. Hence, there are 3*K* components describing interfered neighbors' information since BS *b* has *K* interfered neighbors, i.e., $|O_b(t)| = K$.

2) **Action**: To improve the computation efficiency, discrete actions are used instead of continuous actions. Firstly, we divide the beamformer of BS *b* in time slot *t* into two parts,

$$\mathbf{w}_b(t) = \sqrt{p_b(t)}\overline{\mathbf{w}}_b(t), \tag{7}$$

where $p_b(t) = \|\mathbf{w}_b(t)\|^2$ denotes the transmit power of BS *b* in time slot *t*, subject to $0 \leq p_b(t) \leq \mathbf{P_{max}}$ and $[p_{max}, P_{max}] \in \mathbf{P_{max}}$. $\overline{\mathbf{w}}_b(t) \in [0, 2\pi)$ indicates the direction of the transmit beam.

We divide the values from $\frac{1}{E_{pow}}\mathbf{P_{max}}$ to $\mathbf{P_{max}}$ into E_{pow} parts evenly, and define $\mathbf{P}$ as the set of transmit power. We also consider that codebook $\mathbf{C}$ is composed of E_{code} code vectors, $\mathbf{c}_e \in \mathcal{C}^{N\times 1}$ for $\overline{\mathbf{w}}_b(t)$, where $e \in \{0, 1, ..., E_{code}-1\}$. For an agent, an action is choosing an appropriate transmission power



(a) $E_{code} = 4$ (b) $E_{code} = 8$

Fig. 2. The proposed codebooks when N = 3 and S = 16.

and code. Therefore, the actions set at each BS can be simply represented as follows:

$$\mathbf{A} = \{(\mathbf{c}, p), \mathbf{c} \in \mathbf{C}, p \in \mathbf{P}\}, \tag{8}$$

where $\mathbf{P} = \left\{ \frac{1}{E_{pow}} \mathbf{P_{max}}, \frac{2}{E_{pow}} \mathbf{P_{max}}, ..., \mathbf{P_{max}} \right\}$ and $\mathbf{C} = \{\mathbf{c}_0, \mathbf{c}_1, ..., \mathbf{c}_{E_{code}-1}\}$. We design the codebook following the guidelines in [12]. $\mathbf{C}(n, q)$ refers to the phase shift of the nth antenna element in the qth code. The codebook matrix is designed as follows:

$$\mathbf{C}(n, q) = \frac{1}{\sqrt{N}} \exp\left(j \frac{2\pi}{S} \left\lfloor \frac{n mod(q + \frac{E_{code}}{2}, E_{code})}{E_{code/S}} \right\rfloor \right), \tag{9}$$

where S denotes the number of available phase values for each antenna element, and $\lfloor . \rfloor$ denotes the function of rounding down. We set N = 3 and S = 16. The beam patterns are shown in Fig. 2 when $E_{code} = 4$ and 8. In this condition, the size of possible actions set is $E = E_{code} \cdot E_{pow}$.

3) **Reward**: When an agent chooses an optimal action to maximize its achievable transmission rate, it will inevitably generate interference to other users. Obviously, the interference causes a decline of the overall throughput of the whole UDN. Thus, a well-designed reward could help the agent improve its achievable transmission rate with acceptable interference to other users. Towards this end, we define the reward of agents in micro cells as:

$$r_d(t) = R_d(\mathbf{W}(t)) - P_d(\mathbf{W}(t)), \quad d \in \{1, ..., L\}, \tag{10}$$

where $R_d(\mathbf{W}(t))$ and $P_d(\mathbf{W}(t))$ denote the achievable transmission rate and penalty of BS d, respectively. The reward function is the net profit of the transmission rate and penalty value of an individual agent. BS d will transmit interference signals to UE $f \in O_d(t+1)$, thus reducing the achievable transmission rate of UE f. Therefore, we define the sum of the achievable transmission rate losses of UE $f \in O_d(t+1)$ as the penalty value of agent d:

$$\begin{aligned} P_d(\mathbf{W}(t)) = &\sum_{f \in O_d(t+1)} \log\left(1 + \frac{|\mathbf{h}^\dagger_{f,f}(t)\overline{\mathbf{w}}_f(t)|^2 p_f(t)}{\sum\limits_{g \neq d, f} |\mathbf{h}^\dagger_{g,f}(t)\overline{\mathbf{w}}_g(t)|^2 p_g(t) + \sigma^2}\right) \\ - \quad &R_f(\mathbf{W}(t)), \qquad\qquad d \in \{1, ..., L\}, \end{aligned} \tag{11}$$

where g denotes the interferers of UE f, $g \in I_f(t+1)$. Note that the reward $r_d(t)$ is calculated in time slot $t+1$, i.e., after agent executes an action $a(t)$ in time slot t. As for an agent in the macro cell, the minimum transmission rate of the primary user should be given priority. On this basis, the agent of macro station tries to reduce the interference to secondary users as much as possible. We design the reward of an agent in the macro cell as a piecewise function:

$$r_0(t) = \begin{cases} (\exp(-R_0(\mathbf{W}(t)) + R_\psi) + \delta_1)\beta_1 - P_0(\mathbf{W}(t)), \\ \qquad\qquad\qquad if \quad R_0(t) \geq R_\psi, \\ (\exp(R_0(\mathbf{W}(t)) - R_\psi) - \delta_2)\beta_2, \\ \qquad\qquad\qquad if \quad R_0(t) < R_\psi, \end{cases} \tag{12}$$

where R_ψ denotes the minimum transmission rate of the primary user. δ_1, δ_2, β_1, β_2 denote the parameters that are set according to experience for achieving better performance. $P_0(\mathbf{W}(t))$ denotes the penalty value of the agent in the macro cell. The calculation of $P_0(\mathbf{W}(t))$ is similar to (11). When the primary user's transmission rate is less than the minimum transmission rate, its reward will become smaller dramatically. Thus, the agent is more inclined to make the primary user's transmission rate greater than the minimum. At the same time, the penalty $P_0(\mathbf{W}(t))$ in the reward guarantees the interference would not be too enormous.

IV. Performance Evaluation

A. Simulation Setup

As described in Section II, we consider a UDN in which L micro cells are randomly distributed within the coverage of a macro cell. To ensure the communication quality of the primary user, the minimum transmission rate R_ψ is set to 1 *bps*. The radii of the macro cell coverage and the micro cell coverage are 1000 *m* and 50 *m*, respectively. In each cell, there is a BS with three antennas located at the center. The maximum transmit power of macro BS is 46 *dBm*, and that of micro BS is 21 *dBm*. The large-scale fading factor between BS b and UE c can be modeled as $\beta_{b,c} = 120.9 + 37.6 \log_{10} d_{b,c} + 10 \log_{10}(z)$ *dB*, where $d_{b,c}$ denotes the distance between BS b and UE c in kilometer. z denotes the shadow fading factor, whose mean value is zero and standard deviation is 8 *dB*. Besides, the small-scale fading factor adopts the Jakes model following the Rayleigh distribution. And the total number of channel paths M is four. The noise power is set to $\sigma^2 = -114$ *dBm*, and the angular spread Δ is 3°.

The DQN hyperparameters are set as below. We employ the Adam optimizer to update the weights of DQNs. In addition, the architecture of each DQN consists of an input layer, an output layer and three hidden layers. The number of interferers is set to $K = 5$. The total number of input ports is 63, i.e., $8 + 8K + 3K$. We set the number of available power levels and the codebook size as E_{pow} = 7 and $E_{code} = 8$, respectively. Thus, the total number of output ports is $E = E_{pow} \cdot E_{code} = 56$. We also apply the $\epsilon-greedy$ policy with $\epsilon(t) = \max\{(1 - \lambda_\epsilon)\epsilon(t-1), \epsilon_{min}\}$, in which

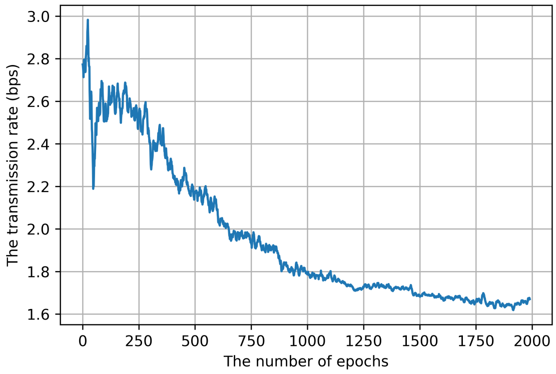


Fig. 3. The transmission rate of the primary user when the number of micro cells L is 11.

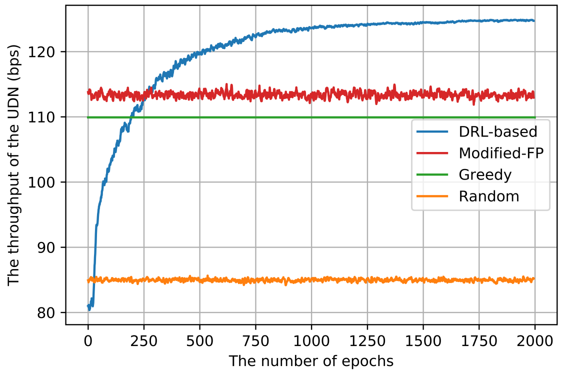


Fig. 4. The throughput comparison of several algorithms including Modified-FP algorithm, DRL-based algorithm, random algorithm and greedy algorithm when the number of micro cells L is 11.

the exploration probability ϵ decreases steadily. The rest of hyperparameters are shown in Table I.

B. Performance Evaluation of the Proposed Algorithm

We perform a set of experiments by setting the number of micro cells as 5, 7, 11, 13, 15, 17 and 19. We can have similar conclusions from the evaluation results of these experiments. In order to facilitate the elaboration of the performances of our proposed algorithm, we utilize the data collected when the number of micro cells is 11 to illustrate some experimental results. To satisfy the constraints in (3), the agent of macro station needs to guarantee the high quality communication for the primary user. However, while the macro BS sends data to the primary user, it also generates interference to the secondary users. In addition, because the transmit power of the macro BS is larger than that of other BSs, the interference can not be ignored. In most cases, the larger the transmission rate of the primary user, the greater the interference of macro station to other micro cells. Fig. 3 illustrates the primary user's transmission rate of the proposed DRL-based algorithm when the number of micro cells L is 11. It can be observed that the transmission rate descents steadily, but still greater than the minimum transmission rate R_ψ. Note that the transmission rate of the primary user does not converge to the minimum transmission rate R_ψ. The reason is that the codebook and transmission power are discrete, which is designed to to improve the computation efficiency.

TABLE I
THE HYPERPARAMETERS OF DQNS

Hyperparameters	DQN for macro cell	DQN for micro cell
The number of neurons in the hidden layers	128, 64, 64	64, 32, 32
Initial exploration rate	0.6	0.6
Minimum exploration rate	0.001	0.001
Decay exploration rate	$8e^{-5}$	$1.2e^{-4}$
Initial learning rate	1e^{-4}	1e^{-4}
The size of minibatch	256	128

To evaluate the performance of our proposed algorithm, we take the following three algorithms as the baselines.

• **Greedy**: Each agent only chooses the strategy to maximum the transmission rate of its own cell user.

• **Random**: Each agent randomly selects an action in all time slots.

• **Modified-FP**: In [3], Shen *et al.* proposed an iterative near optimal approach based on FP to solve the interference channel problems. Each BS is able to collect instantaneous global CSI. In this paper, we modified the FP algorithm in order to guarantee the communication quality of the primary user. When the transmission rate of the primary user is less than the minimum transmission rate R_ψ, the macro BS changes the strategy at this time to greedy algorithm.

Fig. 4 illustrates the average transmission rates of ten random distributions to rule out the influence of the distribution of micro cells. The modified-FP algorithm achieves the highest transmission rate among the three baselines, while the random algorithm achieves the lowest. In the simulation, the transmission rate achieved by DRL-based algorithm goes up rapidly firstly. After nearly 400 epochs, the performance of DRL-based algorithm is better than that of modified-FP algorithm. After 1000 epochs, the performance of the DRL-based algorithm converges. After converged, the throughput achieved by DRL-based algorithm outperforms modified-FP algorithm by almost 8%. In fact, modified-FP algorithm could get close to the optimal result. However, when the transmission rate of the primary user is less than R_ψ, the mechanism for guaranteeing the communication quality of the primary user works. As a result, the transmission rate of the primary user is much higher than R_ψ. It brings interference to secondary users, and limits the achieved throughput of the entire network.

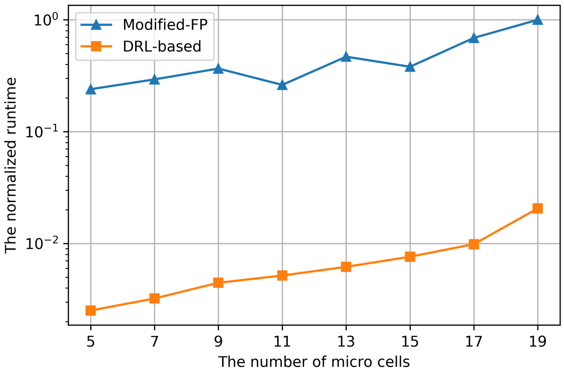


Fig. 5. The runtime comparison between modified-FP and DRL-based algorithm with the increase of the number of micro cells.

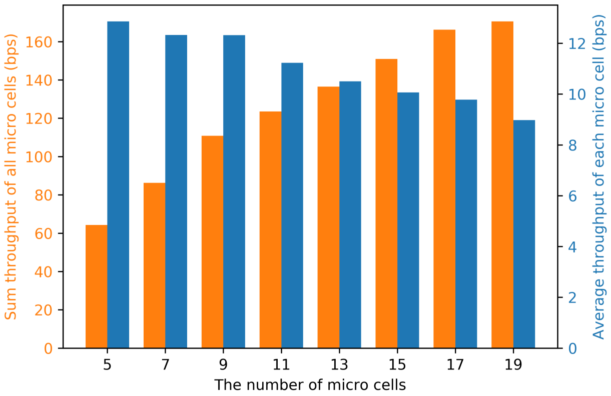


Fig. 6. Sum throughput of all micro cells and average throughput of each micro cell with the increase of the number of micro cells.

The computation efficiency of the algorithm can be judged by the runtime of the algorithm. Fig. 5 illustrates the improvement of runtime. The runtime of the DRL-based algorithm is greatly less than that of the modified-FP algorithm. Specifically, when the number of micro cells is 11, the runtime of the DRL-based algorithm only requires 2% of the modified-FP algorithm. In fact, the DRL-based algorithm only uses partial CSI and adopts distributed structure, which leads to the improvement of computation efficiency. Therefore, the proposed algorithm greatly reduces the decision-making time, and is applicable to the dynamic environment.

To eliminate the influence of the random distribution of micro cells, we carried out experiments with ten different random distributions. The results are shown in Fig.6, in which each point of the histogram is the average value of the collected data. When the number of micro cells increases, the throughput of the all micro cells increases, and the average throughput of each micro cell decreases. Obviously, the more micro cells deployed, the larger throughput achieved. However, with the increase of micro cell density, the distance between micro cells will be much closer. In this condition, users will suffer from enormous interference from other cells, which leads to the reduction of the average throughput. Note that the beam codebook chosen by our proposed algorithm is discrete and finite. It can not increase transmission rates of all users. In addition, each agent only communicates with a part of other agents. It is impossible for each agent to predict the behaviors of all other agents. With the increase of micro cell density, it becomes more difficult to coordinate all the interference in the UDN. Therefore, the marginal benefit of deploying additional BS is small when the number of micro cells is large.

V. Conclusion

In this paper, we investigated the beamforming strategy for throughput maximization in UDNs, and proposed a multi-agent DRL-based beamforming algorithm. The algorithm trains each BS to select an appropriate beamformer with the transmit power level to coordinate the interference. The simulation results show that the proposed algorithm can effectively maximize the overall throughput of the UDN while guarantees the communication quality of the primary user. In addition, the proposed algorithm improves the computation efficiency, which makes it achieve the highest performance compared to the baselines with reasonable time resource consumption.

References

[1] I. Ahmed, H. Khammari, A. Shahid, A. Musa, K. S. Kim, E. De Poorter, and I. Moerman, “A Survey on Hybrid Beamforming Techniques in 5G: Architecture and System Model Perspectives,” *IEEE Commun. Surv. Tutor.*, vol. 20, no. 4, pp. 3060–3097, 2018.

[2] Y. Du, W. Zhang, S. Wang, J. Xia, and H. A. Mohammad, “Joint Resource Allocation and Mode Selection for Device-to-Device Communication Underlying Cellular Networks,” *IEEE Access*, vol. 9, pp. 29 020–29 031, 2021.

[3] K. Shen and W. Yu, “Fractional programming for communication systems—Part I: Power control and beamforming,” *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, 2018.

[4] C. Sun, Z. Shi, and F. Jiang, “A Machine Learning Approach for Beamforming in Ultra Dense Network Considering Selfish and Altruistic Strategy,” *IEEE Access*, vol. 8, pp. 6304–6315, 2020.

[5] X. Wang and M. C. Gursoy, “Multi-Agent Double Deep Q-Learning for Beamforming in mmWave MIMO Networks,” in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2020, pp. 1–6.

[6] J. Zhang, X. Zhang, Z. Yan, Y. Li, W. Wang, and Y. Zhang, “Social-aware cache information processing for 5G ultra-dense networks,” in *2016 8th International Conference on Wireless Communications Signal Processing (WCSP)*, 2016, pp. 1–5.

[7] G. P. Koudouridis and P. Soldati, “Capacity model for network density scheduling in small cell networks,” in *2016 23rd International Conference on Telecommunications (ICT)*, 2016, pp. 1–6.

[8] Y. Xiao, J. Liu, J. Wu, and N. Ansari, “Leveraging deep reinforcement learning for traffic engineering: A survey,” *IEEE Commun. Surv. Tutor.*, vol. 23, no. 4, pp. 2064–2097, 2021.

[9] G. Papoudakis, F. Christianos, A. Rahman, and S. V. Albrecht, “Dealing with non-stationarity in multi-agent deep reinforcement learning,” *arXiv preprint arXiv:1906.04737*, 2019.

[10] J. Ge, Y.-C. Liang, J. Joung, and S. Sun, “Deep Reinforcement Learning for Distributed Dynamic MISO Downlink-Beamforming Coordination,” *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 6070–6085, 2020.

[11] E. U. T. R. Access, “Radio Frequency (RF) system scenarios,” *Document 3GPP TR*, vol. 36, 2011.

[12] W. Zou, Z. Cui, B. Li, Z. Zhou, and Y. Hu, “Beamforming codebook design and performance evaluation for 60 GHz wireless communication,” in *2011 11th International Symposium on Communications & Information Technologies (ISCIT)*. IEEE, 2011, pp. 30–35.