Multi-agent Deep Reinforcement Learning Based Adaptive User Association in Heterogeneous Networks

Weiwen Yi(✉), Xing Zhang, Wenbo Wang, and Jing Li

Wireless Signal Processing and Network Laboratory, Beijing University of Posts and Telecommunications, Beijing 100876, People's Republic of China
yww2013@bupt.edu.cn

Abstract. Nowadays, lots of technical challenges emerge focusing on user association in ever-increasingly complicated 5G heterogeneous networks. With distributed multiple attribute decision making (MADM) algorithm, users tend to maximize their utilities selfishly for lack of cooperation, leading to congestion. Therefore, it is efficient to apply artificial intelligence to deal with these emerging problems, which enables users to learn with incomplete environment information. In this paper, we propose an adaptive user association approach based on multi-agent deep reinforcement learning (RL), considering various user equipment types and femtocell access mechanisms. It aims to achieve a desirable trade-off between Quality of Experience (QoE) and load balancing. We formulate user association as a Markov Decision Process. And a deep RL approach, semi-distributed deep Q-network (DQN), is exploited to get the optimal strategy. Individual reward is defined as a function of transmission rate and base station load, which are adaptively balanced by a designed weight. Simulation results reveal that DQN with adaptive weight achieves the highest average reward compared with DQN with fixed weight and MADM, which indicates it obtains the best trade-off between QoE and load balancing. Compared with MADM, our approach improves by $4\% \sim 11\%$, $32\% \sim 40\%$, 99% in terms of QoE, load balancing and blocking probability, respectively. Furthermore, semi-distributed framework reduces computational complexity.

Keywords: Heterogeneous networks · User association · Multi-agent Deep Q-network

1 Introduction

In order to meet the demand of surging traffic, 5G heterogeneous networks (HetNets) have emerged as an essential solution, especially through the deployment of lower-power small cell base stations (BSs). Compared with traditional cellular networks, HetNets differ primarily in maximum transmit power, coverage

X. Liu et al. (Eds.): ChinaCom 2018, LNICST 262, pp. 57–67, 2019.
https://doi.org/10.1007/978-3-030-06161-6_6

area and spatial density. A survey demonstrates serious penetration losses of the buildings degrade quality of service (QoE) [1]. Hence, femtocells with different access mechanisms have been proposed, where subscribers of femtocells are the users registered in it and nonsubscribers are the users not registered in it [2].

- Closed access: Closed access femtocells only provide services for subscribers, which guarantee privacy and security.
- Hybrid access: Resources of hybrid access femtocells are reserved for subscribers, who may get higher rate than nonsubscribers.
- Open access: Open access femtocells are available to all users.

It is hard to cope with user association because of network heterogeneity and limited resources, which leads to user competitions and network congestion [3]. Due to incomplete information interactions and dynamic environment changes, emerging artificial intelligence method turns into an efficient tool for user association. A network-assisted approach was proposed with Q-learning to derive network information and satisfaction-based multi-criteria decision-making method was used to guide user behavior [4]. In [5], context-aware multiple radio access technology (multi-RAT) was studied. It made double decision on which exact RAT and access point to occupy with ant colony algorithm. However, complicated centralized algorithms have high requirements for the central controller's computational ability. In [3], the evolutionary game and Q-learning were implemented to help distributed individuals make decisions independently. It pursues high QoE without taking load balancing into consideration, which may bring about congestion. Moreover, users tend to maximize their utilities selfishly for lack of cooperation, such as distributed multiple attribute decision making (MADM), which results in the one-sidedness of user decisions [6]. The above related works didn't take into account QoE, load balancing and computational complexity simultaneously when dealing with user association. Therefore, one of the crucial goals for user association in HetNets is to achieve a desirable tradeoff between QoE and load balancing with an appropriate user association algorithm.

In [7], a deep RL method, termed a deep Q-network (DQN), was proposed. In complex and dynamic HetNets, users can learn optimal strategy from high-dimensional state and action space using DQN. In this paper, we propose an adaptive user association approach based on multi-agent DQN. The main contributions include:

- Our approach aims to obtain the desirable trade-off between QoE and load balancing. Considering user equipment (UE) types and femtocell access mechanisms, we exploit semi-distributed multi-agent DQN framework to achieve the optimal strategy. It can transfer the main calculations from central controller to UEs and reduce computational complexity.
- We formulate user association as a Markov decision process (MDP). And we define the individual reward as a weighted function of transmission rate and BS load. The weight is designed into the action. Such reward provides evaluative feedback for each user to make decision adaptively.

– Simulation results show that the proposed approach converges well and achieves the best trade-off between QoE and load balancing. It yields gains in terms of QoE and load balancing and significantly decreases the blocking probability compared with MADM.

2 System Model

We focus on the downlink (DL) transmission scenario of two-tier HetNet. The system model, including information sharing and distributed association scheme, is shown in Fig. 1. We consider a macrocell and N femtocells. The set of users is denoted as $\mathcal{U} = \{u|u = 1, 2, \ldots, K\}$. And the set of BSs is denoted as $\Phi = \{m|m = 0, 1, \ldots, N\}$, where macrocell is indexed by 0.

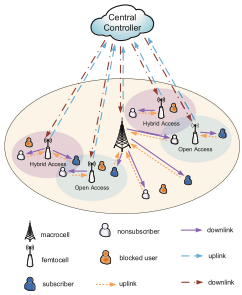


Fig. 1. System model.

The two-tier HetNet uses orthogonal spectrum with an assumption of co-tier interference [2,8]. Every femtocell is equipped with open access or hybrid access signed by 0 and 1 respectively. Therefore, the set of access mechanisms for BSs is $\mathcal{X} = \{0, 1\}$. Each BS consists of M sub-bands with bandwidth b, which are referred to time-frequency radio blocks (RBs). Hence the total bandwidth for BS is denoted as $W = Mb$. Besides, transmission power is uniformly allocated to each sub-band [8].

The spatial distribution of femtocells and users is modeled by homogeneous Poisson Point Process (PPP) with density λ_f and λ_u respectively [9]. Each user can be associated with one BS simultaneously. UE type includes registration attribute and service type. The registration attribute set is $\mathcal{A} = \{0, 1\}$, where subscribers are marked by 0 and nonsubscribers by 1. We consider two kinds of service types as $\mathcal{V} = \{0, 1\}$, where data traffic is indexed by 0 and voice calls by 1. The set of required RBs for different service types is denoted as $\mathcal{B} = \{\beta_s|s \in \mathcal{V}\}$. Therefore, the bandwidth that BS m allocates to each user u with service type s can be denoted as $\varphi_{m,u} = \eta(x, y)b\beta_s$, where $\eta(x, y) \in (0, 1]$. $\eta(x, y)$ is the match factor between registration attribute x and access mechanism y, and $x \in \mathcal{A}$ and $y \in \mathcal{X}$. If nonsubscribers associate with hybrid access femtocells, resources allocated to them will be reduced by $\eta(x, y) < 1$.

Load factor υ_m is defined as the ratio of the allocated bands to the total bandwidth in Eq. (1), which indicates the BS load. $\mathcal{I}_m$ is the initial resource utilization of BS m. BS is overloaded when $\upsilon_m \geq 1$ and under-loaded when $\upsilon_m < 1$. When BS is overloaded, it will randomly block some users until it is under-loaded. Such users are regarded as blocked users, marked by set $\mathcal{O}$.

$$\upsilon_m = \frac{\mathcal{I}_m + \sum\limits_{u \in \mathcal{U}} \varphi_{m,u}}{W}. \tag{1}$$

The received signal-to-noise-plus-interference-ratio (SINR) is formulated as

$$\gamma_{m,u} = \frac{\frac{\varphi_{m,u}}{W} P_m g_{m,u} |x_{m,u}|^{-\alpha}}{\varphi_{m,u} N_0 + I_{\bar{\Phi}_u^f}}, \tag{2}$$

where $g_{m,u}$ is the exponentially distributed channel power with unit mean. $|x_{m,u}|$ indicates the distance from BS m to user u. P_m is the transmit power of BS m, α denotes the path loss exponent, and N_0 is regarded as the power spectral density of white Gaussian noise. The interference of user u is

$$I_{\bar{\Phi}_u^f} = \sum_{n \in \Phi_u^f \setminus m} \delta_n \frac{\varphi_{n,u}}{W} P_n g_{n,u} |x_{n,u}|^{-\alpha}. \tag{3}$$

When $I_{\bar{\Phi}_u^f} = 0$, the SINR degenerates into signal-to-noise-ratio (SNR). And the feasible BS set of user u is $\Phi_u^f = \{m | SNR_{m,u} \geq \gamma_{th}\}$, where γ_{th} is the SNR threshold. The interference probability of BS n detected by an arbitrary user is scaled by a thinning factor $\delta_n = \min\left(\frac{l_n}{W}, 1\right)$, where l_n is the resource utilization of BS n [9]. δ_n indicates that the interference probability is related with the sub-bands occupied. That is, if sub-bands are fully occupied, $\delta_n = 1$, and the interference from BS becomes larger than that of $\delta_n < 1$.

3 Adaptive User Association Based on Multi-agent DQN

In this section, we first formulate the problem as a MDP and elaborate the state, action and reward. Next, we review the basic conception of DQN adopted in this paper. Finally, we show the semi-distributed multi-agent DQN framework, then we get the optimal strategy using our proposed approach.

3.1 Problem Formulation

The BS environment consists of macrocell and femtocells in HetNet. In our proposed approach, users play the role of agents and interact with the BS environment. The parameters are defined as follows.

State. $\mathbf{s}_u$ indicates the state of agent (user) u with BS m selected, which is defined as $\mathbf{s}_u = (w_u, g_{m,u}, \varphi_{m,u}, \upsilon_m)$. $w_u \in \Omega$ is the weight of transmission rate discretized into F levels. $\Omega = \{\omega | \omega = 1\Delta, 2\Delta, \ldots, (F-1)\Delta\}$ is the set of weight and $\Delta = \frac{1}{F}$. And the state profile can be formulated as $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K)$.

Action. Due to the indeterminacy of the weight, $w_u \in \Omega$ has been designed into the action. Current action of agent u can be denoted as $\mathbf{a}_u = (c_u, w_u)$, where $c_u \in \Phi_u^f$ and $w_u \in \Omega$. The action profile can be formulated as $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_K)$.

Reward. $R_u(\mathbf{s}_u, \mathbf{s}'_u, \mathbf{a}_u)$ indicates the feedback received when agent u takes the action $\mathbf{a}_u$ and turns out to be state $\mathbf{s}'_u$ from $\mathbf{s}_u$ [10]. The transmission rate of agent u refers to Shannon formula, which is formulated in Eq. (4).

$$U_u(\mathbf{s}_u, \mathbf{s}'_u, \mathbf{a}_u) = \begin{cases} \varphi_{m,u}(\mathbf{s}'_u) log(1 + \gamma_{m,u}(\mathbf{s}'_u)), u \notin \mathcal{O}(\mathbf{s}'_u) \\ 0, u \in \mathcal{O}(\mathbf{s}'_u) \end{cases}. \tag{4}$$

Conclusions as a result, we draw the following reward as a function of transmission rate and BS load as shown in Eq. (5).

$$R_u(\mathbf{s}_u, \mathbf{s}'_u, \mathbf{a}_u) = w_u(\mathbf{s}'_u) \frac{U_u(\mathbf{s}_u, \mathbf{s}'_u, \mathbf{a}_u)}{\sum\limits_{u \in \mathcal{U}} U_u(\mathbf{s}_u, \mathbf{s}'_u, \mathbf{a}_u)} + (1 - w_u(\mathbf{s}'_u))(1 - \upsilon_m(\mathbf{s}'_u)). \tag{5}$$

There is a trade-off problem between transmission rate and BS load, which are balanced by the designed weight w_u. To seek high transmission rate, agent sets large w_u, which negatively affects BS load. Therefore, by such reward, each agent can discover actions in a more effective way, in order to contribute to the trade-off between QoE and load balancing.

3.2 Deep Q-Network

The main modification to online Q-learning in DQN module is to use a separate target network $\hat{\mathcal{Q}}_u$ with weight θ_u^- for generating the target action-value in learning update [7]. Evaluation network $\mathcal{Q}_u$ with weight θ_u is updated every step while $\hat{\mathcal{Q}}_u$ is assigned by θ_u every H step. DeepMind has proposed the DQN with the temporal-difference goal

$$y_u^t = R_u(\mathbf{s}_u, \mathbf{s}'_u, \mathbf{a}_u) + \tau \max_{\mathbf{a}'_u} \hat{\mathcal{Q}}_u(\mathbf{s}'_u, \mathbf{a}'_u; \theta_u^-), \tag{6}$$

where agent takes action $\mathbf{a}'_u$ in the next step. t indicates current training step and τ is a discounted factor. Therefore the update of θ_u can be formulated as

$$\theta_u^{t+1} = \theta_u^t + \rho\{y_u^t - \mathcal{Q}_u(\mathbf{s}_u, \mathbf{a}_u; \theta_u)\} \nabla \mathcal{Q}_u(\mathbf{s}_u, \mathbf{a}_u; \theta_u), \tag{7}$$

where ρ is the learning rate.

3.3 Proposed Algorithm

The proposed semi-distributed multi-agent DQN framework is illustrated in Fig. 2. This figure shows the interactions between agents and BS environment. After agents take actions, the information sharing scheme is executed. Then agents transform to next states, get the reward feedbacks and perform updates.

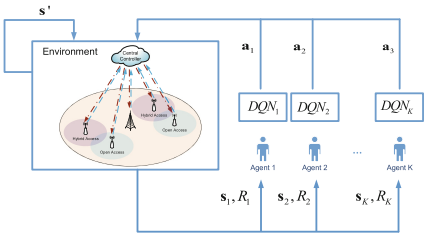


Fig. 2. Semi-distributed multi-agent DQN framework.

Algorithm 1 Multi-agent DQN Based Adaptive User Association

Initialize:

τ, ρ, ε, K, $\mathbf{D}$ with capacity M for every agent, replace iter H, training steps T, initial state profile $\mathbf{s}$, $\mathbf{Q}$ with random weights $\boldsymbol{\theta}$, $\hat{\mathbf{Q}}$ with weights $\boldsymbol{\theta}^- = \boldsymbol{\theta}$

Output:

Optimal strategy $\boldsymbol{\pi_{opt}}$

1: **for** $t = 1$ **to** T **do**
2: **for** $u = 1$ **to** K **do**
3: Observe state $\mathbf{s}_u$
4: **if** $rand() < \varepsilon$ **then**
5: Select a random action $\mathbf{a}_u$
6: **else**
7: Select $\mathbf{a}_u = \arg\max_{\mathbf{a}_u} \mathcal{Q}_u(\mathbf{s}_u, \mathbf{a}_u; \theta_u)$
8: **end if**
9: Execute $\mathbf{a}_u$, share $\mathbf{\Gamma}_u^{UL}$ and acquire $\mathbf{\Gamma}_u^{DL}$
10: Share U_u and acquire $\mathbf{U}_u^{DL}$
11: Observe $\mathbf{s}'_u$ and acquire $R_u(\mathbf{s}_u, \mathbf{s}'_u, \mathbf{a}_u)$
12: Store transition $(\mathbf{s}_u, \mathbf{a}_u, \mathcal{R}_u, \mathbf{s}'_u)$ in $\mathcal{D}_u$, then sample minibatch from $\mathcal{D}_u$
13: Set y_u^t according to Eq. (6) and perform a gradient descent on $(y_u^t - \mathcal{Q}_u(\mathbf{s}_u, \mathbf{a}_u; \theta_u))^2$ with respect to θ_u according to Eq. (7)
14: Set $\mathbf{s}_u = \mathbf{s}'_u$ and reset $\hat{\mathcal{Q}}_u = \mathcal{Q}_u$ every H step
15: **end for**
16: Decrease ε
17: **end for**

Make a final optimal strategy $\boldsymbol{\pi_{opt}} = \mathbf{a}$

The pseudo-code of multi-agent DQN based adaptive user association algorithm is shown in Algorithm 1. $\mathbf{D} = (\mathcal{D}_u, u \in \mathcal{U})$ are the replay memories for users. $\mathbf{Q} = (\mathcal{Q}_u, u \in \mathcal{U})$ with weights $\boldsymbol{\theta} = (\theta_u, u \in \mathcal{U})$ are evaluation networks for users. And target networks for users are $\hat{\mathbf{Q}} = (\hat{\mathcal{Q}}_u, u \in \mathcal{U})$ with weights $\boldsymbol{\theta}^- = (\theta_u^-, u \in \mathcal{U})$.

At decision epochs, after current state $\mathbf{s}_u$ observed, every agent takes action $\mathbf{a}_u$, by exploration or exploitation (Line 3–8). In exploration mode, agent takes action randomly with probability ε (Line 4–5). However, in exploitation mode, agent takes action by maximum Q-value based on the previously learned $\mathcal{Q}_u$ (Line 6–7). Once agents take actions, they share $\mathbf{\Gamma}_u^{UL} = (c_u, \varphi_{m,u})$ on the UL and

acquire others' information $\mathbf{\Gamma}_u^{DL} = (\mathbf{\Gamma}_i^{UL}, i \in \bar{\mathcal{U}}_u)$ on the DL, where $\bar{\mathcal{U}}_u$ indicates users except for current agent u (Line 9). Next, agents share U_u on the UL and acquire others' transmission rates $\mathbf{U}_u^{DL} = (U_i, i \in \bar{\mathcal{U}}_u)$ on the DL (Line 10). After that, agent u transforms to next state $\mathbf{s}'_u$ and gets evaluation feedback $R_u(\mathbf{s}_u, \mathbf{s}'_u, \mathbf{a}_u)$ to drive the next more correct decision (Line 11).

By experience replay, we store the agent experiences, $(\mathbf{s}_u, \mathbf{a}_u, \mathcal{R}_u, \mathbf{s}'_u)$ transition, into memory $\mathcal{D}_u$ with finite capacity M. If the memory buffer of $\mathcal{D}_u$ is full, we overwrite with recent transitions. Next, with full replay memory, sample uniformly minibatch from $\mathcal{D}_u$ (Line 12). Then, with temporal-difference goal y_u^t, perform a gradient descent step on evaluation network $\mathcal{Q}_u$ by RMSProp algorithm (Line 13). It's important to copy $\mathcal{Q}_u$ to target network $\hat{\mathcal{Q}}_u$ every H step. $\hat{\mathcal{Q}}_u$ is used for calculating y_u^t for the following H steps (Line 14). The policy during training is ε-greedy with ε annealed linearly. ε decreases with training steps until there is no exploration process (Line 16). Finally, after each agent repeats the above procedures T times, we get the optimal strategy $\boldsymbol{\pi_{opt}}$ for all users.

4 Performance Evaluation

Simulation results are presented in this section. The details of parameter setting are shown in Table 1. The access mechanisms of femtocells, registration attributes and service types of users are assigned randomly. If x=1 and y=1, match factor $\eta(x, y) = 0.6$, otherwise $\eta(x, y) = 1$. We consider MADM as baseline approach. Its utility function is formulated in Eq. (8) with fixed weight and users take actions by maximum $R_{u,m}$.

$$R_{u,m} = w_u \frac{U_{u,m}}{\sum\limits_{j \in \Phi_u^f} U_{u,j}} + (1 - w_u)(1 - \upsilon_m). \tag{8}$$

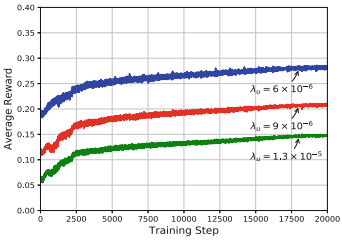
All results are averaged with P Monto Carlo simulation epochs and evaluated by four metrics. They are average reward, average transmission rate, standard deviation of resource utilization rate and blocking probability, respectively. And we consider fixed weights, $w_{1,u} = 0.2, 0.5, 0.8$, in order to investigate the effects of adaptive weight.

Figure 3a plots the convergency under user density λ_u=6×10^{-6}, 9×10^{-6} and 1.3×10^{-5}. It shows the average reward varying with the training steps. Fluctuation of average reward indicates that the exploration probability ε works. When ε decreases with training steps, reward tends to rise first and then converges to a relative stable value within a certain range. It suggests that our proposed approach converges well.

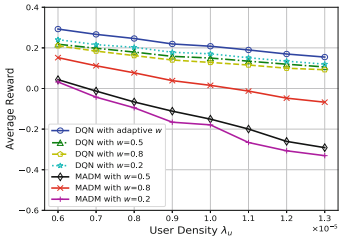
Figure 3b plots average reward varying against user density. The trade-off performance is evaluated by average reward. When user density increases, average reward decreases because of higher blocking probability. As seen in this figure, the proposed approach achieves the highest average reward compared with any other approach, which indicates it obtains the desirable tradeoff between

Table 1. Parameter setup.

Parameter	Value
Area radius	500 m
Bandwidth W	20 MHz
Transmit power of two-tier HetNet	{46, 20} dBm
Power spectral density of white Gaussian noise N_0	-174 dBm
Path loss α	4
Femtocell density λ_f	4×10^{-6}
Initial resource utilization of network	Uniform distribution
Location of N femtocells	PPP
Location of K users	PPP
Sub-band bandwidth b	180 kHz
Required RBs $\mathcal{B}$	$\mathcal{B} = \{10, 20\}$
Weight discretized level F	5
Monte Carlo simulation epochs P	300
Training steps T	20 K
SNR threshold γ_{th}	9.56 dB
Discounted factor τ	0.9
Learning rate ρ	0.05
Replace iter H	200
Exploration probability ε	0.2
Capacity M of replay memory $\mathcal{D}_u$	2000



(a) Average reward vs. training step under DQN with adaptive weight.



(b) Average reward vs. user density under different approaches.

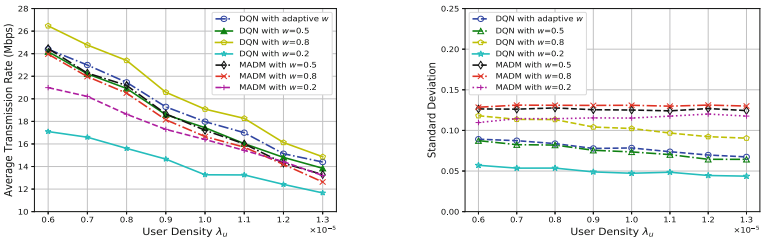
Fig. 3. Average reward.

QoE and load balancing. DQN approaches have better performance than MADM approaches. It shows that by information sharing and learning, users make better decisions. MADM gets optimal strategy according to current network situation

without cooperation, which leads users to simultaneously select low-load BSs that can provide high transmission rate.

Figure 4 shows the comparison of QoE and load balancing. Figure 4a plots average transmission rate against user density, which reflects QoE of users. Figure 4b investigates the standard deviation of resource utilization rate among BSs, which reflects load balancing of network. In Fig. 4a, QoE decreases when user density increases. It is due to limited resources BSs can offer and larger interference probability from other BSs. This figure shows that DQN with $w = 0.8$ gains the best QoE because of large weight of transmission rate. Our approach gains the second best QoE, by $4 \sim 11\%$ improvement than MADM. In Fig. 4b, as the user density rises, the standard deviation decreases among DQN approaches while increases slowly among MADM approaches. Lower standard deviation represents better load balancing. DQN with adaptive weight outperforms MADM approaches from the perspective of load balancing by $32 \sim 40\%$ improvement.

Comparing Fig. 4a with Fig. 4b, for DQN with fixed weight, weight can control the optimal strategy to focus more on QoE or load balancing. It can be seen that DQN with $w = 0.2$ gets the worst QoE in Fig. 4a. However, In Fig. 4b, DQN with $w = 0.2$ has the lowest standard deviation, which suggests that it performs well in load balancing because of large weight of the load. For DQN with $w = 0.8$, we observe that seeking high QoE has a negative impact on load balancing. Thus, we can infer that DQN with adaptive weight intelligently selects the appropriate weight and gets a desirable trade-off strategy. Moreover, the QoE of MADM with $w = 0.2$ decreases with user density more slowly than MADM with $w = 0.5$ and $w = 0.8$. And for DQN approaches, the gap of QoE is decreasing with user density. It shows that we urgently need to consider load balancing in the case of high user density, in order to maintain the QoE level.



(a) Average transmission rate vs. user density.

(b) Standard deviation of resource utilization rate among BSs vs. user density.

Fig. 4. Comparison of QoE and load balancing under different approaches.

In Fig. 5, the ordinate axis is logarithmic. As user density increases, blocking probability rises because BSs with limited resources could not accept more requests from users. MADM approaches get the worse blocking probability owing

to its decision way, while DQN approaches improve by 99% compared with MADM.

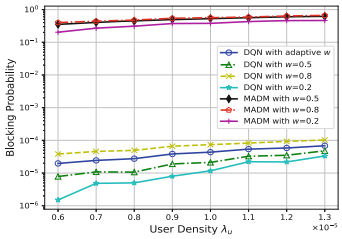


Fig. 5. Blocking probability vs. user density under different approaches.

The computational complexity of our approach depends on the number of state, action of each UE and the amount of information sharing, while by centralized algorithm it depends on the number of the cartesian product of state and action among users. It offloads the main calculations to UEs, which reduces computational complexity.

5 Conclusion

In this paper, we have studied user association problem in HetNets, considering femtocell access mechanisms and UE types. We have proposed multi-agent DQN based adaptive user association approach, aiming to jointly solve the trade-off problem from the perspective of QoE and load balancing. We formulate the problem as a MDP and adopt semi-distributed multi-agent DQN to get the optimal strategy. The reward is defined as a weighted function of transmission rate and BS load, which enables users to maintain QoE and contribute to load balancing. Therefore, by our approach, users can set their weights adaptively and select BSs intelligently to obtain the desirable trade-off strategy. Simulation results verify that the average reward of our approach outperforms DQN with fixed weight and MADM, which indicates it obtains the best trade-off between QoE and load balancing. In terms of QoE, load balancing and blocking probability, our approach improves by $4\% \sim 11\%$, $32\% \sim 40\%$, 99% respectively, compared with MADM. This is because our approach addresses user association adaptively by cooperation. The computational complexity depends on the number of state, action of each UE and the amount of information sharing. It is a relatively significantly improvement over centralized algorithms.

Acknowledgements. This work is supported by the National Science Foundation of China (NSFC) under grant 61771065, 61571054 and 61631005.

References

1. Chandrasekhar, V., Andrews, J.G., Gatherer, A.: Femtocell networks: a survey. IEEE Commun. Mag. **46**(9), 59–67 (2008)
2. De La Roche, G., Valcarce, A., López-Pérez, D., Zhang, J.: Access control mechanisms for femtocells. IEEE Commun. Mag. **48**(1), 33–39 (2010)
3. Feng, Z., Song, L., Han, Z., Zhao, X., et al.: Cell selection in two-tier femtocell networks with open/closed access using evolutionary game. In: Wireless Communications and Networking Conference (WCNC), pp. 860–865. IEEE (2013)
4. El Helou, M., Ibrahim, M., Lahoud, S., Khawam, K., Mezher, D., Cousin, B.: A network-assisted approach for rat selection in heterogeneous cellular networks. IEEE J. Sel. Areas Commun. **33**(6), 1055–1067 (2015)
5. Li, J., Zhang, X., Wang, S., Wang, W.: Context-aware multi-rat connection with bi-level decision in 5g heterogeneous networks. In: 2017 IEEE/CIC International Conference on Communications in China (ICCC), pp. 1–6 (2017). https://doi.org/10.1109/ICCChina.2017.8330398
6. Wang, L., Kuo, G.S.G.: Mathematical modeling for network selection in heterogeneous wireless networks - a tutorial. IEEE Commun. Surv. Tutor. **15**(1), 271–292 (2013)
7. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)
8. Yan, M., Feng, G., Qin, S.: Multi-rat access based on multi-agent reinforcement learning. In: GLOBECOM 2017–2017 IEEE Global Communications Conference, pp. 1–6. IEEE (2017)
9. Chae, S.H., Hong, J.P., Choi, W.: Optimal access in ofdma multi-rat cellular networks with stochastic geometry: can a single rat be better? IEEE Trans. Wirel. Commun. **15**(7), 4778–4789 (2016)
10. Liu, Y.J., Cheng, S.M., Hsueh, Y.L.: enb selection for machine type communications using reinforcement learning based markov decision process. IEEE Trans. Veh. Technol. **66**(12), 11330–11338 (2017)