ARTIFICIAL INTELLIGENCE-BASED RESOURCE ALLOCATION IN ULTRADENSE NETWORKS

Applying Event-Triggered Q-Learning Algorithms

Haijun Zhang, Mengting Feng, Keping Long, George K. Karagiannidis, and Arumugam Nallanathan

Ultradense networks (UDNs) have emerged as a promising architecture that can support the extremely high demand for data traffic in the future. Through the dense deployment of massive small base stations (SBSs), the system can be well promoted in terms of network capacity and spectrum efficiency, but this deployment scheme will also bring huge challenges to wireless resource management. Artificial intelligence (AI) can be applied to UDN scenarios, enabling intelligent communication devices to learn and complete resource allocation. This article discusses resource allocation schemes based on AI algorithms in UDNs and proposes an event-triggered, reinforcement-learning-based subchannel and power allocation algorithm.

Digital Object Identifier 10.1109/MVT.2019.2938328
Date of current version: 16 October 2019

We consider nonorthogonal multiple access (NOMA) in UDNs, which allows a subchannel to be used by multiple users at the same time. Each user is regarded as an agent, and each agent obtains observational information from the environment during the learning process. We design event-triggered conditions based on current and previous moments of observational information, and the agent uses those conditions to decide whether to update policies and perform actions. Simulation results show the effectiveness of the proposed event-triggered, reinforcement-learning-based resource allocation algorithm.

Motivations and Background

The evolution of mobile communication technology has been witnessed all over the world, and the next-generation wireless network, unlike the previous one, will have stronger service capabilities. It has been observed that the communication network's data traffic will increase exponentially due to large-scale mobile users, so the next-generation wireless network needs a higher capacity and faster data transmission rate. One of the most effective solutions is to densely deploy cells to improve spatial multiplexing [1].

Recently, UDNs have attracted worldwide attention and become a promising architecture that supports massive mobile devices. In the UDN scenario, there is a large number of communication nodes, including low-power SBSs and wireless access points, that may be dozens of times more dense than current ones and can achieve great capacity and spectrum efficiency improvement in local hotspots [2]. In UDNs, the full-functional SBSs, including picocell and femtocell BSs, aim to enhance signals for hotspot areas. Relays are macro-extended access points that can quickly respond to burst-capacity improvement requirements [3]. UDNs also bring many challenges, such as severe interference caused by neighboring cells and performance degradation [4]. Therefore, it is essential to design an effective and efficient radio resource management mechanism for them.

NOMA can provide users with more access opportunities when there are limited frequency resources, allowing multiple users to share the same frequency [5]. In NOMA, different user signals on the same subchannel are transmitted nonorthogonally, and interference information is introduced actively. Successive interference cancelation (SIC) is adopted at the receiver to solve the problem of inter-user interference. Compared with traditional OMA schemes, NOMA can support more users by effectively utilizing nonorthogonal resources, which increases users' access opportunities.

However, the increasing number of users will make the decoding complexity of the receiver more serious, which will result in a reduction in NOMA efficiency. The densification of the network makes the BSs' coverage smaller, while the distance between the BS and the user is shorter and each BS needs to serve only a small number of users. Therefore, we consider adopting a NOMA scheme in SBSs. The cooperation between NOMA and UDNs further enhances network spectrum efficiency and also supports large-scale connection of users [6]. However, due to the characteristics of NOMA, radio resource management, especially in terms of subchannel and power allocation, becomes extremely challenging.

AI, a recently popular subject, enables computers to learn and perform complex tasks. Future intelligent mobile terminals are expected to achieve autonomous learning and decision making through AI to find the optimal resource allocation scheme [7]. At present, machine learning is widely used to solve the problem of AI and is the most important way to realize AI. Complex communication networks will generate large amounts of data, and machine-learning algorithms can extract valuable information from large data sets and make predictions, so they have excellent capabilities for finding new solutions [8]. Considering that the traditional resource allocation scheme takes a lot of time to calculate, we can use historical data generated by classical resource allocation algorithms as training samples and employ the *k*-nearest-neighbor algorithm, one of the simplest of all machine-learning algorithms, to locate the most similar sample sets from the historical database to find a matched resource allocation scheme [9]. In addition, a deep neural network (DNN) can be used to train existing data to find the relationship between inputs and outputs and learn the characteristics of optimal resource allocation for wireless resource management [10].

For the case without a training set, reinforcement learning is a good choice and is characterized by repeated trial and error. Agents regard system performance as a reward and constantly perform actions, such as resource allocation, to find an allocation scheme that maximizes the return [11]. A powerful resource optimization algorithm can quickly find global optimal solutions without increasing computational complexity. For complex networks, traditional optimization algorithms may bring huge computational complexity. For machine-learning algorithms, we need only build a learning model, and the machine can complete a lot of work through self-learning. Therefore, the application of AI in the field of wireless communication networks is an inevitable future trend.

System Architecture

In Figure 1 we describe the architecture of the UDN with NOMA. The UDN introduces multiple types of SBSs within the scope of the macro cell. A large number of low-power SBSs is characteristic of UDNs, in which the deployment density of SBSs is much higher than in current networks. SBSs have low power, small coverage, and a small physical volume, and they are flexible to deploy.

Different types of SBSs are often applied to different scenarios. For example, the femtocell BS is the initial design form of small cells. It is easy to manage, can be automatically configured and optimized, and is often used in indoor scenes, such as homes and offices, to enhance signals. Microcells and picocells are often used in hotspots, such as shopping streets and other crowded areas, where the number of users is extremely large. By deploying SBSs, high-speed access can be provided, and network capacity can be improved. Through SBSs, blind areas that are not covered by MBSs, such as underground parking lots and tunnels, can be covered. Similarly, SBSs can be deployed in areas with poor signal quality, such as cell edges, thereby extending the coverage of the MBSs and enhancing user experience and total system coverage.

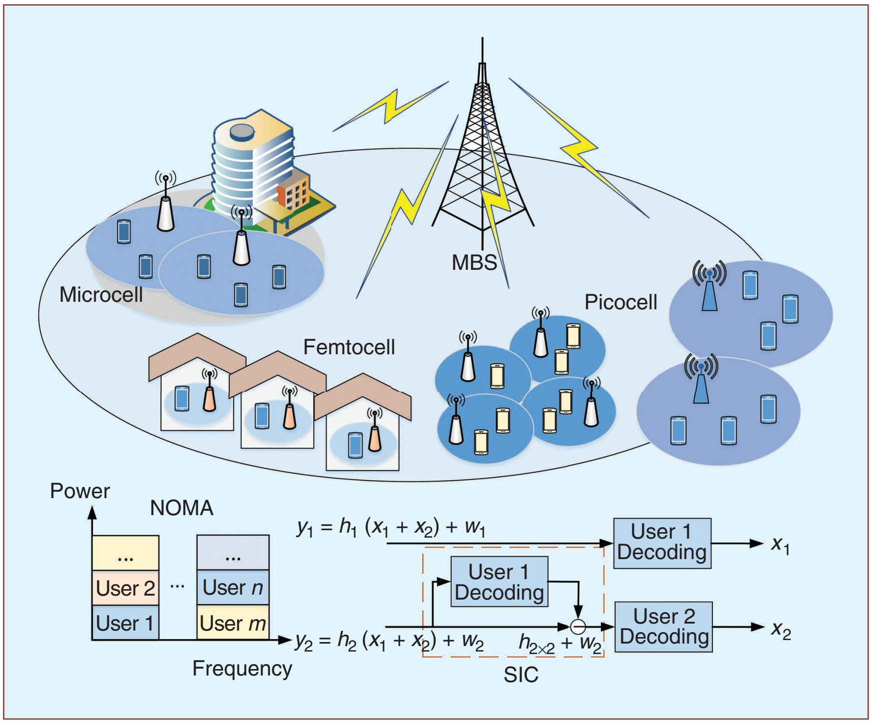

FIGURE 1 The architecture of a UDN with NOMA. MBS: macro base station.

We consider NOMA in a UDN; that is, multiple users in small cells can share one subchannel to expand the scale of access and improve spectrum efficiency. At the transmitter, nonorthogonal transmission between different users causes the problem of inter-user interference, which is also the reason for employing the SIC technique at the receiver. In the SIC process, the receiver ranks the users' power and gradually subtracts the interference from the users with the highest power, thereby eliminating the interference step by step [12].

For example, there are two user signals, x_1 and x_2, on one subchannel. The signal power of x_1 is greater than x_2. It is assumed that the signals received by the receiver are, respectively, $y_i = h_i(x_1 + x_2) + w_i$, where $i = 1, 2$, h_i is the channel coefficient and w_i is the Gaussian white noise. Since x_1 has the strongest signal power, it is decoded first. Then, x_1 is output and subtracted from the received signal; $h_2 x_2 + w_2$ is taken as the input, and x_2 is decoded. In fact, in the SIC process, only one signal can be detected at a time. If there are n users on one subchannel, the user with the lowest power needs to perform $n - 1$ times of SIC before decoding.

AI-Based Resource Allocation in UDNs

Due to the large number of wireless devices, resource allocation in the UDN may require more powerful computing capabilities than in the past, or better optimization algorithms must be found. Therefore, AI is a good choice; it provides the ability to parse abundant data. In this section, we discuss several resource allocation schemes based on AI in UDNs from three perspectives: supervised learning, unsupervised learning, and reinforcement learning.

The core of supervised learning is classification, and it is a good choice to achieve classification through neural networks. Therefore, for supervised learning, we discuss the approximate optimization scheme based on a DNN in a UDN. Artificial neural networks simulate the nervous system of the human brain and learn the complex relationships between inputs and outputs. There are hidden layers between the input and output of the neural network, and the DNN has multiple hidden layers, which is beneficial for data with complex relationships. In traditional optimization algorithms, a series of complex operations may be carried out to obtain results. In that respect, DNN-based resource management algorithms have great advantages.

As shown in Figure 2(a), the application of neural networks in resource allocation requires recording the input and output values of the optimized resource allocation algorithm (such as the water-filling algorithm) as training data sets for the neural network, which means that the existing optimization algorithm of the neural network is first learned when applying a neural network algorithm in UDN resource allocation; then, the neural network learns to optimize by approximating the existing algorithms [13]. The existing resource optimization algorithms can be well learned through a DNN algorithm. Therefore, if a DNN algorithm can approximate accurately, it can replace some complex algorithms.

The clustering algorithm is a typical unsupervised learning one, and K-means is a widely used clustering algorithm. It uses the characteristics of samples to classify those with many similarities into the same category. Figure 2(b) illustrates the steps of a resource management scheme based on a K-means clustering algorithm in a UDN. The clustering method can simplify the topology of the UDN before the UDN allocates resources to reduce the computational complexity of resource allocation. Due to the different coverage and transmission power of the varying types of SBSs in the UDN, the SBS in the UDN is divided into multiple clusters using the K-means algorithm. Users are grouped according to the interference among them in each SBS cluster, and every user is assigned to the group with the lower interference effect in each cluster to alleviate intra-cluster interference [14]. After clustering, other algorithms are used to allocate resources. By using clustering algorithms in UDNs, interference can be mitigated, and resource allocation complexity can be reduced.

Compared with those two schemes, reinforcement learning does not require a training set; it uses repeated trial and error. Q-learning is the most common algorithm in reinforcement learning. In the resource allocation scheme based on Q-learning in a UDN the system, performance can be regarded as the reward; presently occupied resources, such as the channel and power, can be viewed as the current state, and the allocating resources can be considered as the action. The action selection strategy is constantly updated to make the optimal decision and find the distribution scheme that can maximize the reward. Because we focus on resource allocation in UDNs and have no training set, we consider using a reinforcement learning algorithm in UDNs.

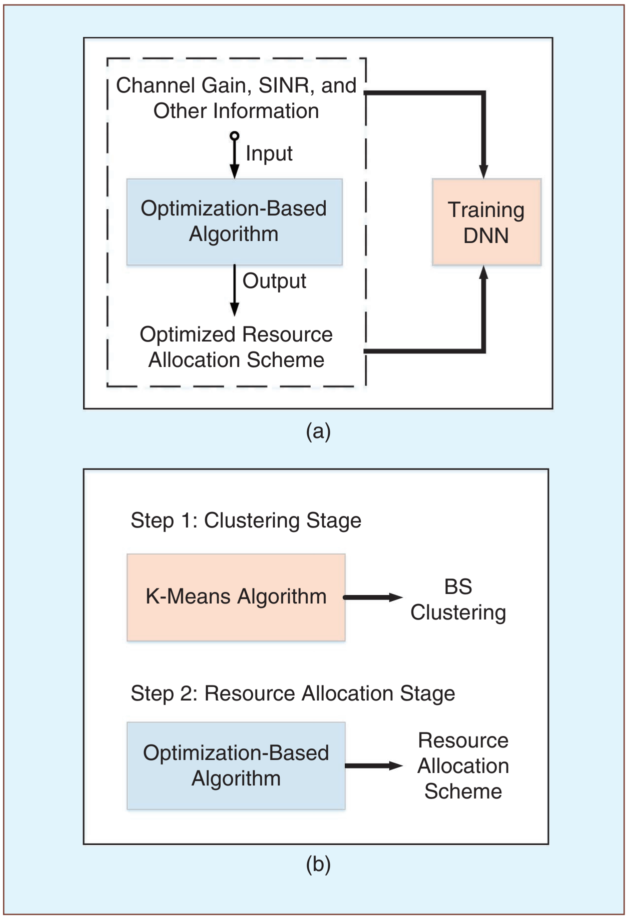

FIGURE 2 (a) The approximate optimization scheme based on a DNN. (b) The resource management scheme based on the K-means clustering algorithm. SINR: signal-to-interference-and-noise ratio.

Q-Learning–Based Resource Allocation in UDNs

We focus on subchannel and power allocation in UDNs for maximizing system energy efficiency. Actually, it is very difficult to solve the energy efficiency optimization problem directly. Moreover, as the number of users and BSs increases, the computational complexity rises dramatically. It should be noted that the user does not know the channel state information accurately and that the interference information between the BSs is also difficult to know. To solve those limitations, in this section, a reinforcement learning algorithm is used. We propose an event-triggered, multiagent Q-learning resource allocation algorithm. Users and BSs do not need to be informed of the channel status information and interference. The user can learn the optimal allocation strategy through historical energy efficiency, subchannels, and power values.

Q-*Learning*

Q-learning is a model-free reinforcement learning algorithm. Therefore, the process of interaction between agent and environment must be considered. The basic idea of Q-learning is to train the tuples composed of the state, action, reward, and next state. In the Q-learning algorithm, a Q-table is constructed between the state and action to store the Q-values, which are the reward for taking action at a certain time plus the maximum expected value for the next step. Then, the action that can obtain the greatest reward is selected according to the Q-value to get as much return as possible. At each time slot, the agent first observes the environment and then performs actions to interact with it. The environment changes after an action is executed, and the quality of those changes is expressed by the reward returned from the environment. Therefore, the purpose of Q-learning is to get as much reward as possible.

Suppose the state set is S and the action set is A. At each time slot, t, the agent adjusts the action, A_t, at the next moment according to the current state, S_t, and reward, R_t, so that the next state, S_{t+1}, can be determined. Considering that a variety of actions can be selected after a state and that each action gets different states and rewards after being executed, the next question is the choice of the action, that is, the formulation of the strategy, π. The actions under the optimal strategy can get the biggest reward. The future potential value of a state can be expressed by the value function, which is defined as the expectation of the cumulative discount reward, because, with the increase of time, the reward will be discounted. The value function under strategy π is the reward of the current state plus the discount of the state value function under the next strategy.

The value of an action in a certain state can be expressed by an action-value function, that is, $Q(s,a)$. The Q-function can be understood as the sum of the reward $r(s,a)$ obtained after the execution of the action and the discount state value function under the next optimal strategy. The value function of the state under the optimal strategy is equivalent to the maximum action-value function of all of the possible actions in the state. Therefore, as long as the maximum Q-value is found, the optimal strategy can be found. In Q-learning, the Q-value is updated recursively. The Q-value of the previous slot, the learning rate, and the maximum Q-value of the next state are taken into account in the updating rules.

Event-Triggered, Q*-Learning–Based Resource Allocation*

We introduced the idea of Q-learning into the research of subchannel and power allocation in UDNs, so we need to correlate the states, actions, rewards, and strategies in Q-learning with the actual process of energy efficiency optimization. The corresponding relationship is as follows. We regard users in small cells as agents.

- *State*: We define the current channel occupied by the user and the allocated power as the state of the agent.
- *Action*: The actions taken by each agent are divided into two aspects: subchannel allocation and power allocation. Because NOMA is considered in the UDN, the user can select all of the subchannels. We also discretize the power and divide it into different levels, and the user can select all of the levels of power.
- *Reward*: Users need to achieve quality of service constraints by adjusting their power. The energy efficiency of each user is regarded as a reward, but, at the same time, the user's signal-to-interference-and-noise ratio (SINR) needs to be greater than the SINR threshold. Therefore, when the SINR satisfies the condition, the reward is the value of the energy efficiency; otherwise, the reward is zero.
- *Action selection strategy*: In the strategy selection of the Q-learning algorithm, there is a balance between exploration and exploitation. In Q-learning, an ε-greedy algorithm and a Boltzmann distribution algorithm are often used as strategies. The ε is generally a small value, which is used as a probability value for selecting random actions. Random actions are used to explore the effects of unknown actions, which is conducive to updating Q-values and obtaining better strategies. The use of a greedy strategy to calculate an optimal action based on the current Q-value is the exploitation. It can be used to judge whether the algorithm is effective. The Boltzmann distribution algorithm can control the probability of the actions by adjusting the temperature parameters. The strategy can be understood as the probability of choosing an action. A larger temperature parameter means that all actions can be selected with equal probability. As the temperature parameter decreases, the action with the largest Q-value will be selected as the maximum probability. Therefore, in this article we choose the Boltzmann distribution algorithm as the strategy.

Because we treat users as agents, each user needs to conduct a policy search in each time slot. The agent's entire learning process is periodic. When the learning environment is relatively stable and if the number of users in the system is large, the periodic policy search will occupy a certain amount of computing resources. The event-triggered control method is an effective alternative to periodic control. Therefore, for the learning process of agents, we propose an event-triggered Q-learning algorithm that aims to reduce resource consumption in the learning process.

Before the agent makes a decision, we set some conditions in advance, that is, events. When the condition is established, the agent updates the policy and executes the action; otherwise, it executes the action at the last time slot. Agents need to observe the environmental state information before selecting actions. We regard the Q-value as the observed information of the current time slot. Then, we can use the change rate of the previously observed value and the current observed value as the event-trigger condition. If agents consider only the change rate of their own observations, it will not be conducive to learning the optimal strategy of a group. Therefore, we add the deviation of the current reward to the design of the event-triggering conditions. When calculating the energy efficiency of a certain user, other users on the same subchannel will interfere with it, and the reward will be affected. Thus, we divide the agents on the same channel into a group.' Then, the deviation of the agent's current reward is the difference between the average reward of the agent group and that of the agent. Therefore, in the event-triggered Q-learning algorithm, we take as events the rate of change of the Q-value in the current step and the previous step and also the deviation of the reward of the agent. When the change rate of the observed value and the deviation of the reward are greater than the set threshold, that is, the event is triggered, the agent updates the strategy and action; otherwise, it performs the action of the last moment.

Figure 3 illustrates the process of an event-triggered Q-learning algorithm. Each agent observes the environment information, state, and reward. Then, the agent needs to determine whether the event-triggered condition is satisfied before updating the strategy. If the condition is satisfied, the agent will find the optimal strategy and update the action based on current information. Otherwise, it will go directly to the next step, and the state will be the same as before. After interacting with the environment, the next state and reward can be obtained. Finally, the agent updates the Q-table. Each agent cycles the process until it converges.

Simulation Results

Having explained the proposed event-triggered Q-learning resource allocation algorithm process, we evaluate its performance. Because the aim of the algorithm is to maximize system energy efficiency, in the simulation we evaluate the algorithm only from the aspect of energy efficiency. SBSs are randomly distributed within the scope of the MBS, while small cells use NOMA to support access to numbers of users. We set the radius of the MBS and each SBS to 500 and 10 m, respectively. The bandwidth is limited to 1 MHz, and the carrier frequency is 2

GHz. The noise power spectral density is set to –174 decibels with reference to one milliwatt/Hz.

Figure 4 shows the energy efficiency curves of the event-triggered Q-learning algorithm and the traditional Q-learning algorithm in 500 time slots. Considering the large number of BSs deployed in the UDN, we set the density of the SBSs to 500 small cells/km^2. The Q-learning algorithm is a process of continuous trial and error, and the energy efficiency directly obtained is a curve of constant twists and turns. Therefore, we record the maximum energy efficiency; for each time slot, this recorded value is compared with the energy efficiency value obtained in the current time slot. If the recorded value is less than the energy efficiency value obtained in the current time slot, the recorded maximum energy efficiency is updated. In addition, we used the Monte Carlo method to smooth the resulting curve. From Figure 4, we find that the value of the energy efficiency can reach convergence with the increase of the time slots. Compared with the classical Q-learning algorithm, the proposed event-triggered Q-learning can achieve better results, which shows that the algorithm can find an improved strategy faster.

Figure 5 depicts the change of the energy efficiency when the number of users on the small cells increases from two to 12 under different optimization schemes. We compare the proposed algorithm with an equal power allocation scheme. As seen in the figure, the event-triggered Q-learning and classical Q-learning algorithms can be significantly better than existing ones. As the number of users grows, the energy efficiency of the Q-learning algorithm gradually decreases, but, when the number of users exceeds eight, energy efficiency remains stable.

Figure 6 illustrates the effect of different densities of small cells on energy efficiency under different optimization schemes. There are two definitions of UDNs. When the density of small cells is much larger than the density of users or when there are more than 1,000 small cells/km^2, the network is considered to be a UDN [15]. We simulated the change of the energy efficiency from 200 to 1,200 small cells/km^2. It can be seen that with increasing density the energy efficiency of the different schemes decreases. In addition, compared with OMA, NOMA may suffer some performance losses when the BSs' density is greater than a certain threshold in the UDN. In fact, in the case of high small-cell density, the interference between the BSs will be more serious. Moreover, more of the BSs' circuit power will be consumed, which leads to performance degradation of the system.

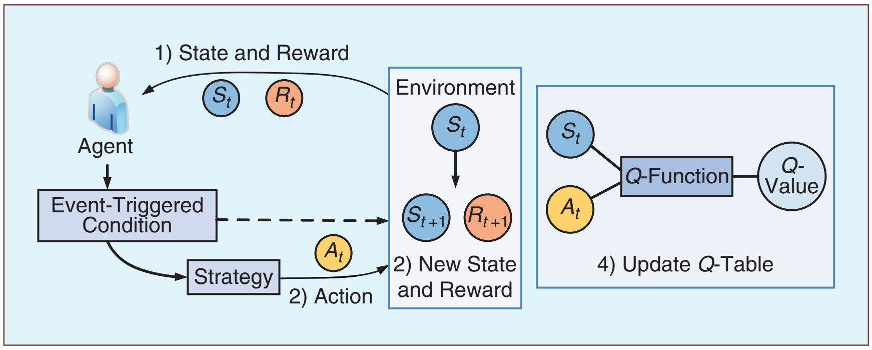

FIGURE 3 The event-triggered Q-learning algorithm process.

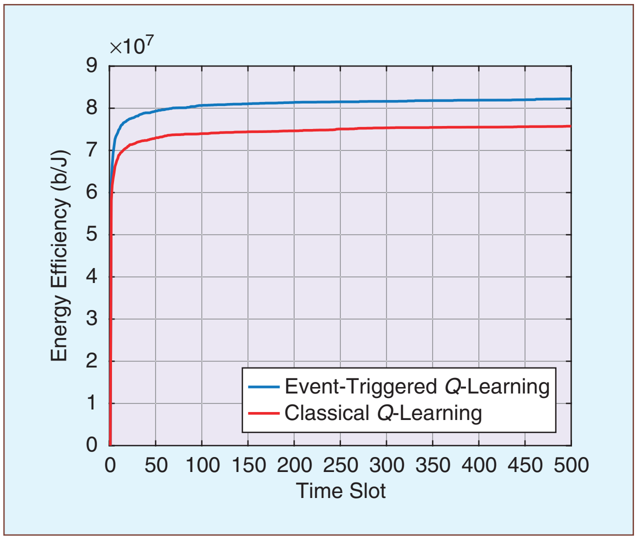

FIGURE 4 The energy efficiency versus the time slot.

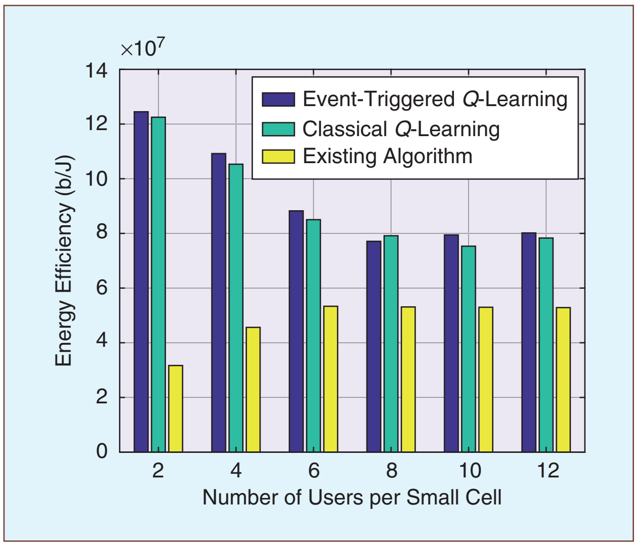

FIGURE 5 The energy efficiency versus the number of users per small cell, with different schemes.

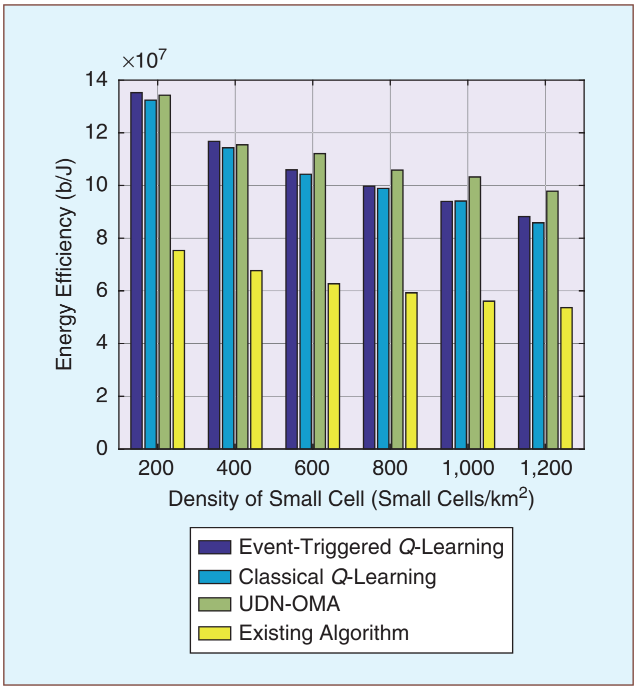

FIGURE 6 The energy efficiency versus the density of the small cells, with different schemes.

Conclusions

In this article, the idea of event-triggered Q-learning was introduced into the research of subchannel and power allocation in UDNs, and a resource allocation algorithm based on event-triggered Q-learning was proposed. Users in small cells are regarded as agents, and energy efficiency is viewed as the reward. Through interaction with the environment, users dynamically adjust the action of the next time slot, and the goal is to maximize the cumulative reward value.

In event-triggered Q-learning, users with learning ability do not need to know the exact channel state information and can choose subchannels and power autonomously by strategy. Event-triggered Q-learning focuses on the research of action and policy in the learning process. Before making a decision, the agent judges whether to update the policy and perform the action according to the event-triggered condition. The simulation results show the effectiveness of the proposed event-triggered, reinforcement-learning-based resource allocation algorithm. In the future, we will consider other techniques in event-triggered, Q-learning-based UDN resource allocation, such as millimeter-wave communications.

Future Works

In UDNs, although higher spectrum resource utilization and system capacity can be obtained by densely deploying small cells with low power consumption and short distances, interference and power consumption are also severe. Therefore, effective resource management schemes have always been the research hotspot for UDNs. However, there are still some problems to be solved in the resource allocation of event-triggered Q-learning in UDNs. The hybrid resource allocation mechanism of OMA and NOMA should be further considered to provide a more flexible matching mechanism for subchannels and users. When BS density increases gradually, the use of NOMA results in the performance degradation of the system.

In addition, there are many important factors that need to be considered and optimized together, for example, the user clustering mechanism. The increasing density of small cells in UDNs makes the distance between BSs closer. Therefore, the user's choice of access to the appropriate BS also plays an important role in the optimization of resources in UDNs. The application of machine learning to wireless resource allocation is another important aspect of our upcoming research. In the future, machine learning can be further applied to the field of communication, and algorithms that can quickly find an optimal resource allocation scheme without increasing computational complexity can be studied.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grants 61822104 and 61771044), Beijing Natural Science Foundation (grants L172025 and L172049), 111 Project (grant B170003), and the Fundamental Research Funds for the Central Universities (grant RC1631). The corresponding authors are Keping Long and Haijun Zhang.

Author Information

Haijun Zhang (haijunzhang@ieee.org) is a professor at the University of Science and Technology Beijing. His research interests include resource allocation, mobility management, and machine learning in wireless networks. He is a Senior Member of the IEEE.

Mengting Feng (G20178656@xs.ustb.edu.cn) is an M.S. degree student at the University of Science and Technology Beijing. Her research interests include 5G networks, resource allocation, power control, and energy efficiency in wireless communications.

Keping Long (longkeping@ustb.edu.cn) is a professor and dean at the School of Computer and Communication Engineering, University of Science and Technology, Beijing, China. His research interests include wireless communications,

optical fiber communication, optical modulation, and next generation networks. He is a Senior Member of the IEEE.

George K. Karagiannidis (geokarag@auth.gr) is a professor in the Electrical Computer Engineering Department and director of the Digital Telecommunications Systems and Networks Laboratory at Aristotle University of Thessaloniki, Greece. His research interests are in the broad area of digital communications systems and signal processing, with an emphasis on wireless communications, optical wireless communications, wireless power transfer and applications, communications for biomedical engineering, stochastic processes in biology, and wireless security. He is a Fellow of the IEEE.

Arumugam Nallanathan (a.nallanathan@qmul.ac.uk) is a professor of wireless communications in the School of Electronic Engineering and Computer Science at Queen Mary University of London. His research interests include radio access networks; resource allocation; multiple-input, multiple-output communication; interference suppression; telecommunication network reliability; and the Internet of Things. He is a Fellow of the IEEE.

References

[1] S. Chen, Z. Zeng, and C. Guo, "Exploiting polarization for system capacity maximization in ultra-dense small cell networks," *IEEE Access*, vol. 5, pp. 17,059–17,069, Aug. 2017. doi: 10.1109/ACCESS.2017.2745416.

[2] G. P. Koudouridis and P. Soldati, "Spectrum and network density management in 5G ultra-dense networks," *Wireless Commun.*, vol. 24, no. 5, pp. 30–37, Oct. 2017. doi: 10.1109/MWC.2017.1700087.

[3] W. Peng, M. Li, Y. Li, W. Gao, and T. Jiang, "Ultra-dense heterogeneous relay networks: A non-uniform traffic hotspot case," *IEEE Netw.*, vol. 31, no. 4, pp. 22–27, July 2017. doi: 10.1109/MNET.2017.1600295.

[4] H. Zhang, S. Huang, C. Jiang, K. Long, V. C. M. Leung, and H. V. Poor, "Energy efficient user association and power allocation in millimeter-wave-based ultra dense networks with energy harvesting base stations," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1936–1947, Sept. 2017. doi: 10.1109/JSAC.2017.2720898.

[5] Z. Qin, X. Yue, Y. Liu, Z. Ding, and A. Nallanathan, "User association and resource allocation in unified NOMA enabled heterogeneous ultra dense networks," *IEEE Commun. Mag.*, vol. 56, no. 6, pp. 86–92, June 2018. doi: 10.1109/MCOM.2018.1700497.

[6] Z. Zhang, G. Yang, Z. Ma, M. Xiao, Z. Ding, and P. Fan, "Heterogeneous ultradense networks with NOMA: System architecture, coordination framework, and performance evaluation," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 110–120, June 2018. doi: 10.1109/MVT.2018.2812280.

[7] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017. doi:10.1109/MWC.2016.1500356WC.

[8] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, "Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks," *IEEE Access*, vol. 6, pp. 32,328–32,338, May 2018. doi: 10.1109/ACCESS.2018.2837692.

[9] Y. Liu, C. He, X. Li, C. Zhang, and C. Tian, "Power allocation schemes based on machine learning for distributed antenna systems," *IEEE Access*, vol. 7, pp. 20,577–20,584, Jan. 2019. doi: 10.1109/ACCESS.2019.2896134.

[10] H. Ye, G. Y. Li, and B. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018. doi: 10.1109/LWC.2017.2757490.

[11] Y. Liu, S. Cheng, and Y. Hsueh, "eNB selection for machine type communications using reinforcement learning based Markov decision process," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11,330–11,338, Dec. 2017. doi: 10.1109/TVT.2017.2730230.

[12] H. Zhang, Y. Qiu, K. Long, G. K. Karagiannidis, X. Wang, and A. Nallanathan, "Resource allocation in NOMA-based fog radio access networks," *Wireless Commun.*, vol. 25, no. 3, pp. 110–115, June 2018. doi: 10.1109/MWC.2018.1700326.

[13] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, Learning to optimize: Training deep neural networks for wireless resource management. 2017. [Online]. Available: https://arxiv.org/abs/1705.09412

[14] L. Liang, W. Wang, Y. Jia, and S. Fu, "A cluster-based energy-efficient resource management scheme for ultra-dense networks," *IEEE Access*, vol. 4, pp. 6823–6832, Sept. 2016. doi: 10.1109/ACCESS.2016.2614517.

[15] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-dense networks: A survey," *IEEE Commun. Surv. Tut.*, vol. 18, no. 4, pp. 2522–2545, May 2016. doi: 10.1109/COMST.2016.2571730.

VT