Real-Time Resource Slicing for 5G RAN via Deep Reinforcement Learning

Ranran Xi *, Xin Chen*, Ying Chen*, Zhuo Li*
* *School of Computer Science, Beijing Information Science & Technology University*
Beijing, China
Email: xiranran2018@163.com, {chenxin, chenying, lizhuo}@bistu.edu.cn

***Abstract*—With the rapid growth of Internet of Things (IoT), network slicing is regarded as an important technology to support the multi-users' needs for 5G mobile network. Network slicing allows network operators to provide services to different users, which can improve the rational utilization of network and hardware resources. In order to ensure the quality of service and build low-cost network infrastructure services, it is a challenging problem to find an appropriate resource allocation mechanism. In this paper, we discuss resource allocation in 5G radio access network (RAN). Considering the real-time resource request of the slice user, we propose a semi-Markov decision system model, which enables the virtual network provider to effectively satisfy the different user demands in real time. Then, we propose a resource slicing algorithm based on deep reinforcement learning (RS-DRL), which aims to improve the long-term benefits of virtual network providers and the utilization of slicing resources. We evaluate the performance of the RS-DRL through evaluations and comparisons. The results show that the proposed RS-DRL algorithm can effectively improve the performance and achieve the long-term benefits quickly.**

***Index Terms*—5G RAN, Network slicing, Markov decision process(MDP), Resource allocation, Deep reinforcement learning**

I. Introduction

With the advent of 5G network, Internet of Things (IoT) such as mobile vehicular ad hoc network (VANET) [1], etc., are emerging constantly. Various high-quality services, such as 4K ultra-high-definition video, virtual reality, augmented reality and unmanned driving, need the support of future 5G network applications [2]. The rapid development of mobile computing paradigms and the improvement of users' quality of life have promoted the higher demand of customers for high-quality services [3] [4]. For mobile service providers, the ensuing multi-functional services also pose great challenges on them [5]. Especially with the proliferation of computation-intensive IoT applications, the service performance of end devices may be deteriorated [6]. Compared with 4G, 5G provides efficient and reliable data transmission rate, extremely low end-to-end delay, and large-scale machine equipment connection [7]. There will be many application scenarios in future 5G. Three typical application scenarios are: enhanced mobile broadband (eMBB) with new radio access technology; large-scale machine-type communication (mMTC) with high peak rate; high reliability and low latency communication (uRLLC) with end-to-end transmission delay. Different application scenarios have various performance requirements and quality of service (QoS) in terms of mobile security, delay, and reliability [8]. Existing network facilities are difficult to build an independent infrastructure for each application scenario.

Network slicing has been recently gaining momentum among growing community of researchers from both academia and industry, which can address the above problems [9]. The fundamental purpose of resource slicing is to maximize the benefits of mobile virtual network operator (MVNO) sharing physical RAN infrastructure. In network slicing architectural paradigm, it decouples the operation of the virtual network at the top of physical infrastructure through slicing isolation and resource allocation customized for tenants [10]. Network slicing improves flexibility and efficiency of network services.

Resource slicing has become a hot topic in current researches. Many works have conducted research in the wireless cellular technology based on time for user segmentation of resources, they considered the resource usage time [12]. Some literature thought throughput to be an important performance indicator for slice consideration [13], but did not consider other influencing factors. In other works, throughput and time were considered simultaneously [14], but these studies did not take into account the real-time requirements of slicing and network provider benefits. Therefore, how to allocate resources rationally and achieve maximum benefits are emerging challenges.

To tackle the aforementioned challenges, we establish a semi-Markov dynamic model in RAN. In order to solve the problem of super-large state space, we adopt a deep reinforcement learning method. The evaluation results show that our RS-DRL algorithm outperforms the Q-Learning algorithm. Our contributions are summarized as follows:

- We propose a model for allocating 5G RAN resources based on semi-Markov decision process. Considering bandwidth and energy consumption, MVNO makes real-time decisions on the arriving users for maximum benefit. The MVNO decides which slice to accommodate the user, which contains the resources that MVNO leases to it.
- Secondly, we present our resource slicing algorithm, which is based on deep reinforcement learning and allocates resources according to user demands to maximize

Corresponding author: Ying Chen, chenying@bistu.edu.cn.

This work is partly supported by the National Natural Science Foundation of China (Nos. 61872044, 61902029), the Key Research and Cultivation Projects at Beijing Information Science and Technology University (No.5211910958), the Supplementary and Supportive Project for Teachers at Beijing Information Science and Technology University (No.5111911128).

978-1-7281-2583-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICPADS47876.2019.00094

the long-term benefits of the system.

- Finally, we perform evaluations with the aim of not only the efficiency of the proposed solution in comparison with other traditional methods, but also to provide insightful analysis of the experimental results. Evaluation results show that the proposed RS-DRL algorithm can effectively improve system performance.

The rest of the paper is organized as follows. In section II, we discuss the related work in this field. We introduce our system model and present the problem in section III. Our RS-DRL algorithm is described in section IV. In section V, evaluation results are discussed. Finally, conclusions are given in section VI.

II. Related work

A number of research works have been conducted to solve the problem of network slicing resource allocation for the network providers [15]- [26]. Most of the resource slicing focus on the core network (CN), and the research on RAN slicing is limited [15]. In [16] and [17], Quek et al. considered the power minimization of resource allocation in downlink cloud wireless access networks, and proposed a framework for joint optimization of user tasks and power minimization. In [18], Gelenbe et al. studied how to provide resource services in different scenarios when wireless data traffic increases. Dighriri et al. mainly considered the priority of slices in resource allocation, but it did not take into account throughput [19]. These solutions may not be suitable for dynamic network slice resource allocation problems with wide demand.

Traditional network resource allocation methods mainly include game theory, auction mechanism and genetic algorithm. Liu H et al. applied game theory to resource allocation in mobile edge computing networks to reduce energy consumption of MEC networks and maximize resource efficiency [20]. Wang Qian et al. proposed a spectrum allocation method based on auction mechanism, in which multiple users participate in the whole auction game together to improve purchasing power and system performance [21]. Mojtaba et al. solved resource-constrained scheduling problem based on genetic algorithm [22].

However, with the development of the future network towards high density, large data, dynamics and multi-objective, traditional network resource allocation methods have some limitations. Nowadays, new artificial intelligence technologies, such as machine learning and deep learning, have been widely used in medical, health, education, transportation, smart home and other fields [23]. For example, deep reinforcement learning has been widely used in the development of networked vehicles. Ying He [24] et al. have proposed a comprehensive framework. By using deep reinforcement learning, the dynamic arrangement of network, cache and computing resources can be achieved, and the performance of the next generation of vehicular networks can be improved. Yonghua Wang [25] et al. summarized the research of cognitive radio resource allocation using model-free reinforcement learning method.

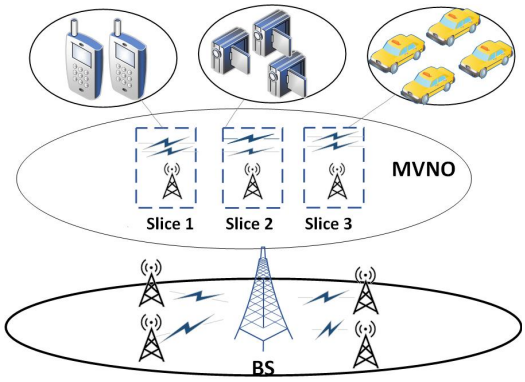

Fig. 1. Conceptual Illustration of 5G RAN Network Slicing

To solve the slice allocation problem effectively, we adopt the deep reinforcement learning method. We introduce the semi-Markov decision process (SMDP) framework [26], which allows network providers to effectively determine user demands in real time and allocates slice bandwidth resources to users. In this paper, we focus on the uncertainty and dynamics of user requests to allocate slice resources. The framework proposed in this paper can deal with large state space very well.

III. Sliced network model and problem formulation

In this part, we first introduce the research scenario under 5G, then propose the system model in RAN, and finally formulate the research problem. For easy reference, some mathematical notations are summarized in Table I.

A. Research Scenarios

Considering that most existing works of the resource allocation focuses on the CN, but in the field of RAN, there is less research on the resource slicing. So the resource slicing research scene is established: infrastructure providers have a series of base stations, whose bandwidth network resources are managed by MVNO. MVNO divides network resources into various slices according to tenants'needs, and then rent out the slices to gain profits. Slices resources can be fully utilized and released with the user tasks begin and end. As shown in Fig.1, there are different user tasks in each slice, and different user tasks have diverse requirements for bandwidth and energy consumption, which poses a difficult problem for us to allocate resources reasonably. To solve this problem, we focus on allocating resources dynamically, and then each slice allocates resources to its own users. From another point of view, we establish a slice selection mechanism. When users arrive at the system, the MVNO makes decisions about which slices users should access to maximize long-term benefits.

B. System Model

Suppose MVNO divides resources into three types of slices according to different service requirements (e.g., mobiles,

TABLE I
SYMBOLIC DESCRIPTION

Name	Description
B	the bandwidth
I	the number of slices
B_{s_i}	the bandwidth of i-th slice
B'_{s_i}	the total bandwidth that the slice s_i has allocated
P	the power
P_{s_i}	the power of i-th slice
t	decision period
S	the system state space
A	the system action space
L	the state transition probability
$B^-_{s_i}$	the bandwidth that the slice s_i can provide
r_{task}	minimum rate constraint
Γ_{s_i}	total number of users in slice s_i

videos, and vehicle communications) [27]: the first slice s_1 is with large bandwidth, which can provide faster service for users who need more traffic, but consumes the most power consumption; the second slice s_2 is with medium bandwidth and can provide general service, medium-sized tasks can be handled easily and energy consumption is moderate; the third slice s_3 is used for small bandwidth user tasks with minimal power consumption.

By definition,

$$B_{all} = B_{s_1} + B_{s_2} + B_{s_3}, \tag{1}$$

$$P_{all} = P_{s_1} + P_{s_2} + P_{s_3}, \tag{2}$$

where $B_{s_1} > B_{s_2} > B_{s_3} > 0$ and $P_{s_1} > P_{s_2} > P_{s_3} > 0$.

Consider that there are U users arriving in the system during the T time. Each user carries a task, represented by $U = \{u_1, ..., u_j\}$, where u_j represents the task j. We make the following assumptions to establish a queuing traffic model: the arrival interval between user tasks' data packets is independent, and the mean value is exponentially distributed $1/\iota_{u_j}$ per second, and ι_{u_j} is the user's packet arrival rate. In our resource allocation mechanism, when the j-th user chooses to access the slice, the rate of the user is related to the number of accessed users and the remaining bandwidth of the slice. According to the Shannon formula, the rate of user u_j accessing slice s_i can be calculated:

$$r_{u_j,s_i} = B^-_{s_i} \cdot log(1 + SNR_{u_j}), \forall u_j \in U, \tag{3}$$

where SNR_{u_j} is signal-to-noise ratio of user u_j . The SNR_{u_j} can be given as

$$SNR_{u_j} = \frac{h_{u_j} p_{u_j}}{\sigma^2_{u_j} + O_{u_j}}, \tag{4}$$

where h_{u_j} captures the channel gain from the base station to u_j, p_{u_j} is the base station transmission power, $\sigma^2_{u_j}$ denotes the background noise, and O_{u_j} denotes the other average interference sources seen by u_j.

Next, we assume that the slice s_i assigns resources to a user task each time with a weight of ω_i, given the average service rate of a slice,

$$\bar{r}_{s_i}(\omega_i) = \frac{\omega_i}{\Gamma_{s_i}} \sum r_{u_j,s_i}. \tag{5}$$

In our model, to calculate the resource utilization of each slice and find operator benefits, we focus on the widely accepted benefit function ($\alpha - fairness$ [28]),

$$\nu_{s_i}(\bar{r}_{s_i}) = \begin{cases} \frac{(\bar{r}_{s_i})^{1-\alpha}}{1-\alpha}, & \alpha \neq 1, \\ log(\bar{r}_{s_i}), & \alpha = 1. \end{cases} \tag{6}$$

where the α sets the concavity level of the benefit function to determine the underlying resource allocation criteria for the slice. In particular, $\alpha = 0$, $\alpha = 1$, $\alpha = 2$ and $\alpha \to \infty$ represents the maximum sum, the proportional fairness, the minimum potential delay fairness and the max-min fairness respectively.

For each slice, we define its average rate, which guarantees minimum service requirements. So the slice utility model is,

$$\Re_{s_i}(\omega_i) = \chi_u \nu(\bar{r}_{s_i}(\omega_i)), \tag{7}$$

where χ_u is user impact factor within the same slice (where $0 \leq \chi_u \leq 1$). It reflects the resource interference of multiple users in the same slice. For ease of calculation, it is set to a fraction, which is designed to reduce utility function values to simulate resource allocation. The above utility function must satisfy the minimum rate requirement for accessing user tasks to ensure quality of service.

Then, let κ_{u_j} represent the size of a user task, i.e. the processing data within the RAN during the execution of the task within the slice. The time cost in processing the task from the slice is given by

$$T_{u_j} = \frac{\kappa_{u_j}}{r_{u_j,s_i}}, \forall u_j \in U, \forall i \in I. \tag{8}$$

The power to process this task is p_{u_j}, then the energy consumed by the user u_j is

$$E_{u_j} = p_{u_j} \cdot T_{u_j} = \frac{p_{u_j} \kappa_{u_j}}{r_{u_j,s_i}}. \tag{9}$$

Each slice has its own power constraint as follows

$$\sum_{u_j \epsilon U, i=1}^{I} p_{u_j} \leq p_{all}. \tag{10}$$

To calculate the benefits of MVNO and maximize the use of slicing resources, we consider the usage of each slicing resource and the cost of consumption. Suppose the price per unit of bandwidth for different slices is λ_i, and the energy loss per unit is δ_i.

The revenue per slice of the operator is

$$B(\omega_i, \lambda_i) = \sum_{i=1}^{I} \Re_{s_i}(\omega_i) \cdot \lambda_i, \tag{11}$$

Also, we can get the energy consumption of each slice,

$$E_{s_i} = \sum E_{u_j} = \sum_{u_{j\epsilon U}, s_i} p_{u_j} \cdot T_{u_j}. \tag{12}$$

The operator's power consumption is

$$E(\delta_i) = \sum_{i=1}^{I} E_{s_i} \cdot \delta_i. \tag{13}$$

Therefore, the MVNO's total benefits model can be expressed as

$$\begin{aligned} \max \quad & \sum_{i=1}^{I} [B(\omega_i, \lambda_i) - E(\delta_i)] \\ s.t. \quad & \sum B'_{s_i} \leq B_{all}, \\ & \sum_{u_{j\epsilon U,} i=1}^{I} p_{u_j} \leq p_{all}, \\ & r_{u_j, s_i} \geq r_{task}, \\ & \omega \leq 1. \end{aligned} \tag{14}$$

C. Problem Formulation

To maximize the long-term benefits for the MVNO while accounting for the real-time arrivals of user requests, we apply the semi-Markov decision process in the model. It is very significant for MVNO to decide whether to accept user tasks that arrive randomly. The semi-Markov decision process model is usually constructed to solve this problem by researchers. We define the SMDP quintuple $< t, S, A, L, R >$, we only need to make decisions when the event is triggered without having to make decisions in every time slot, which greatly saves waiting time.

1) Decision Period: The MVNO needs to make a decision on how to assign slices to the user. Therefore, we define the inter-arrival time between two consecutive user requests as the decision period.

2) State Space: The system state s of the SMDP at the current decision periods captures the resource weights that the slice assigns to the user's task in the system. The state vector can be described as follows,

$$\mathbf{s} = [\omega_{1,1}, ..., \omega_{u_j,i}, ..., \omega_{U,I}]. \tag{15}$$

Here, $\omega_{u_j,i}$ represents the weight assigned to the user by slice i. We can see that the state space becomes larger as the number of users increases. The state space S of all possible states s is defined as:

$$S \triangleq \left\{ s = [\omega_{1,1}, ..., \omega_{u_j,i}, ..., \omega_{U,I}] : \sum_{I} \omega_{u_j,i} \leq 1 \right\}. \tag{16}$$

3) Action Space: At state s, if a user request arrives, the MVNO determines which slice the user accesses to maximize its long-term benefits. If the number of accommodated users of the slice reaches the upper limit, or the bandwidth is insufficient to provide a better service to the user task, the user u_j selects to access another slice in the next time slot. Let $\mathbf{a_s}$ denote the action to be taken at state s where a_{s_i} = 1 if user access to the i-th slice and a_{s_i} = 0 otherwise.

$$\mathbf{a_s} = [a_{s_1}, a_{s_2}, a_{s_3}]. \tag{17}$$

The state action space A_s can be defined by

$$A_s \triangleq \left\{ \mathbf{a_s} = a_{s_i} \epsilon \{0, 1\}; \sum_{i=1}^{I} a_{s_i} \leq 1 \right\}. \tag{18}$$

4) State Transition Probability: As mentioned earlier, we propose our RS-DRL algorithm, which can obtain the optimal strategy for MVNO without the need for information from the environment (in response to the uncertain and dynamic needs of sliced user requests). In order to lay the theoretical foundation and evaluate the performance of our proposed solution, we analyze the dynamics of SMDP, which is characterized by the state transition probability of the basic Markov chain. Then we can use the homogenization technique [29] to determine the probability of the event and derive the transfer probability L:

$$p_{s,s'}(a) = P\{s'|s\}, \forall s, s' \epsilon S, \tag{19}$$

$L = \left\{ p_{s,s'}(a) \right\}$ is the set of state transition probabilities, where $p_{s,s'}(a)$ is the state transition probability from state s to $s^{'}$ when taking action a in state s.

5) Reward Function: The immediate reward after performing action $\mathbf{a_s}$ is defined as follows,

$$r(s, a_s) = \begin{cases} r_c, & \text{if } a_{s_i} = 1, \\ 0, & \text{otherwise .} \end{cases} \tag{20}$$

At state s, if the user accesses the slice, i.e. $a_{s_i} = 1$, the system will move to next state s' and the MVNO receives an immediate reward r_c. If the user does not have access to any one slice or there is no user request arriving at the system, the immediate reward is equal to 0. The value of r_c represents the amount of money paid by the tenant based on resources and additional services required.

$$r_c = \sum_{i=1}^{I} [B(\omega_i, \lambda_i) - E(\delta_i)]. \tag{21}$$

IV. Proposed resource slicing based on deep reinforcement learning

Generally, many methods could be used to cope with the SMDP, but it is not suitable for system optimization with huge state space due to the curse of dimensionality problem. Therefore, we apply the deep reinforcement learning method to solve this problem. The curse of dimensionality can be overcome. And it can be used for online optimization of actual systems.

Algorithm 1 Resource Slicing Based on Deep Reinforcement Learning(RS-DRL)

Input:
List of actions to be taken by users;
Weight constraint value ω;
Minimum rate limit r_{task}.

Initialize the reply memory D and mini-batch size D';
Initialize the Q-network and randomly generate the weight Θ;
Initialize the target Q-network with the weight $\Theta^- = \Theta$;
Initialize the state s.
for each episode $k = 1, 2, ..., K_{max}$ **do**
  Reset evaluation parameters for the resource slicing model environment;
  Randomly generate an initial state $s^{(0)}$, 1 for each user agent $u_j \in U$;
  for each time slot $t = 1, 2, ..., T_{max}$ **do**
    for each user $u_j \in U$ **do**
      Choose which slice to access and decide on resource weight distribution by selecting an action $\mathbf{a_s}$ at the state s using $\epsilon-greedy$ from $Q(s, a_i; \Theta)$.
      if user accesses the specified slice **then**
        Execute action $\mathbf{a_s}$ independently at the user agent, and then calculate the immediate reward r_c based on (21).
      end if
      Observe the next state $s^{(t+1)}$ from the environment simulator, and update the network state at the next moment;
      Store experience $\left\{s^{(t)}, a_{s^{(t)}}, r, s^{(t+1)}\right\}$ in D.
      Pick a mini-batch of samples from D.
      if an episode terminates at s_{j+1} **then**
        $y_t^j = r_t^j$;
      else
        $y_t^j = r_t + \gamma Q(s^{(t+1)}, \underset{a^{(t+1)}}{max} Q(s^{(t+1)}, a_t; \Theta); \Theta^-)$.
      end if
      Perform gradient descent on $(y_t^j - Q(s, a_{s^{(t)}}, \Theta))^2$;
      Update the target Q network every C steps, $\Theta^- = \Theta$.
    end for
  end for
end for

In this section, we present that how to conduct RS-DRL algorithm, which maximizes long-term benefits of MVNO while ensuring that the demand of each user is satisfied. To better understand the advanced deep reinforcement learning method, we briefly introduce deep reinforcement learning in this chapter.

A. Deep Q-network Strategy

In DQN, testing and training operations are carried out at the same time, and the agent obtains rewards or punishments because of taking action by interacting with the environment to learn strategies. The goal of the agent is to maximize the cumulative reward by taking action. When using neural networks to map discrete states to functions of discrete actions, reinforcement learning can be promoted to deep reinforcement learning. The algorithm is able to evaluate the expected utility in available operations without prior knowledge of the system model and to adapt effectively when random conversion occurs. At the same time, expensive calculation requirements are overcome due to large state space and motion space.

Our RS-DRL tries to find a strategy π of the SMDP model, which maps a state $s \in S$ (i.e., resource utilization per slice) to an action $\pi(s)$, (i.e., $a_s \in A_s$, to choose which slice to access or not to access) to maximize the accumulative reward starting from the state $s^{(0)} = s$.

$$V^{\pi}(s) = E\left[\sum_{k=0}^{K} \gamma^k R(s, \pi(s)|s^{(0)}) = s\right], \forall s \in S, K \to \infty, \tag{22}$$

where the positive parameter γ is the discount factor that maps the future reward to the current state. Given the diminishing importance of future cost than the current one, we have $\gamma \leq 1$.

The optimal Q-functions for state-action pairs are denoted by:

$$Q^*(s, a_s) = r(s^t, a_{s^t}) + \gamma E[V^{\pi}(s^{t+1})], \forall s \in S. \tag{23}$$

Then, the optimal value function can be written as follows:

$$V^*(s) = max_{a_s}\left\{Q^*(s, a_s)\right\}. \tag{24}$$

The Q value is updated for each state-action pair according to the following rule:

$$\begin{aligned} &Q(s^t, a^t) = Q(s^t, a^t) \\ &+ \alpha[R(s^t, a^t) + \gamma \underset{a^{t+1}}{max} Q(s^{t+1}, a^{t+1}) - Q(s^t, a^t)], \end{aligned} \tag{25}$$

where α ($0 \leq \alpha \leq 1$) is the learning rate, usually is set close to zero. s^{t+1} is the observation obtained after the action a^t is performed, and a^{t+1} is the action function under the n iterations in the action set at the observed value s^{t+1}. To achieve an approximation of the action state value, i.e.,

$$Q_{t+1}(s, a) \approx Q_t(s, a). \tag{26}$$

Deep Q-network is trained by minimizing the loss function function $L(\Theta)$ toward the target value in each iteration, and the loss function can be written as

$$L(\Theta_t) = E\left[\left(y_t - Q(s, a; \Theta_t)^2\right)\right], \tag{27}$$

where $y_t = r + \gamma max_{a^{t+1}} Q\left(s^{t+1}, a^{t+1}; \Theta_t^-\right)$. The weights Θ_t^- are updated as $\Theta_t^- = \Theta_{t-T}$, the weights update every T time in the deep Q-networks instead of $\Theta_t^- = \Theta_{t-1}$.

B. Beyond Deep Q-Learning Strategy

The max operation in conventional Q-Learning and DQN uses the same value to select and evaluate an action, which tends to choose values that are overestimated. Therefore, when training DQN, we use dual Q-Learning to separate the selection of actions from the evaluation of Q values. This results in a significant reduction in overestimation and the training process run more reliably and quickly.

By reducing the correlation between training samples, experience replay is used to ensure that the optimal strategy cannot be trapped in the local minimum. But in the regular DQN, the selection action and the evaluation of the selected action all use the maximum Q value, which leads to an overly optimistic Q value estimate. Thus, the target value in double DQN is denoted to address the above problem [30]:

$$y_t^{DDQN} = r_t + \gamma Q\left(s^{t+1}, \underset{a^{t+1}}{argmax} Q\left(s^{t+1}, a^{t+1}; \Theta_t\right); \Theta_t^-\right). \quad (28)$$

C. Algorithm Description

The detailed procedure of the proposed RS-DRL is presented in Algorithm 1. In this section, we formulate the resource allocation optimization problem as a deep reinforcement learning process. Since we consider the realistic scenarios, each user selects access slices at a certain rate and power, where the weight and power per user are all dynamically changing. The MVNO faces with large amount of system states, and it has to make an decision on which slice will be accessed by user according to the systems current state. It is barely possible to solve this complicated task by using a traditional method.

More specifically, at the beginning of each episode, the network state is initialized. The MVNO has allocated resources and power to the three slices. Each user arrives at system with demands continuously. Then, the MVNO is responsible for collecting the states from each slice, and it assembles the whole information into a system state. Finally, the MVNO gets a feedback of the optimal policy π for arranging which resources for a certain user task.

In each step, the MVNO obtains the estimated $Q(s, a_t; \theta)$ for all actions. The policy is used to choose the action a_t. It is necessary for each user to send its request. The request contains the rate and power requirements required to complete the user's task. The MVNO selects which slice can accept the user request, depending on the available resources within the slice. If the MVNO accepts the request of the j-th user, the MVNO will send a feedback signal to indicate the available slice to the j-th user. Otherwise, MVNO will not reply to any information. Next, when each action gets the reward $r_t(s, a_t)$ and observes the next state $s^{(t+1)}$, $(s, a_t, r_t(s, a_t), s^{(t+1)})$ will be stored in replay memory D. Then, the agent update weights θ with random mini-batch samples from replay memory D .

V. Evaluation results

In this section, we evaluate the proposed RS-DRL to address dynamic resource allocation issues. First, we determine the value of the learning rate. Second, we compare and analyze the average rate of the Q-Learning (Q-L) and RS-DRL algorithms. Finally, the performance of using random resource allocation (RRA), Q-L and RS-DRL is compared. The evaluation results show that the performance of RS-DRL is greatly improved compared with the other two algorithms. When the number of users increases, the performance improvement is particularly manifest, which verifies the effectiveness and rationality of the algorithm.

TABLE II
PARAMETER CONFIGURATION

Parameters and units	**Values**
System bandwidth, B	100MHz
Number of base stations	4
Transmit power, P	30dBm
The background noise, σ^2	-95dBm
Coverage area	500m*500m
Packet arrival rates, ι_{u_j}	80-110 packets per second
Rate constraint, r_{task}	95 packets per second
Size of replay memory, D	500
Size of mini-batch, D'	30
ϵ-greedy	0.4
Learning rate, α	0.01
Activation Function	ReLU
Discount rete, γ	0.9

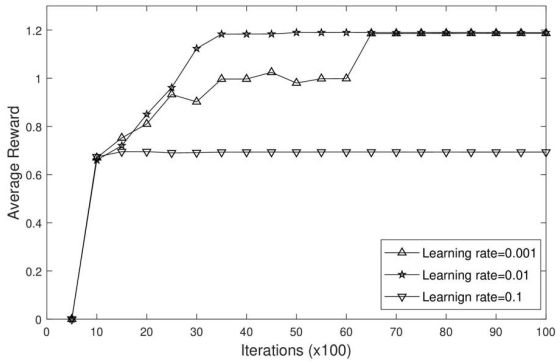

Fig. 2. Convergence of Different Learning Rates

A. Simulation Environment

To evaluate the performance of our proposed solution, we use MATLAB for numerical evaluation and analysis. The evaluation parameters are selected according to the 5G specifications and standards. We assume that the whole system covers a radius of $500m$. The system has one macro base station and four small base stations. Four small base stations are evenly distributed in this area. Without loss of generality, we consider three slices with different bandwidth and power requirements: slice s_1, slice s_2, and slice s_3. Slice s_1 is defined

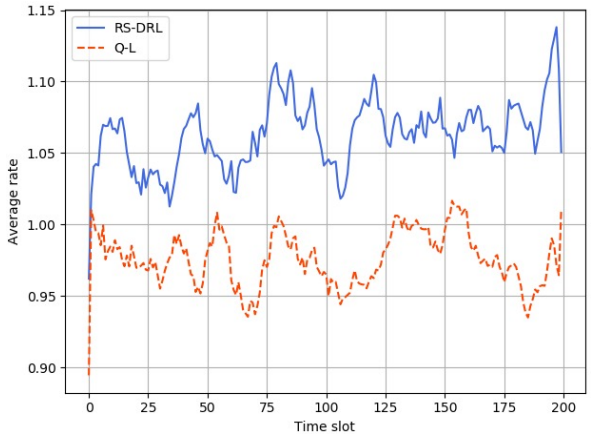

Fig. 3. Average Rate Comparison

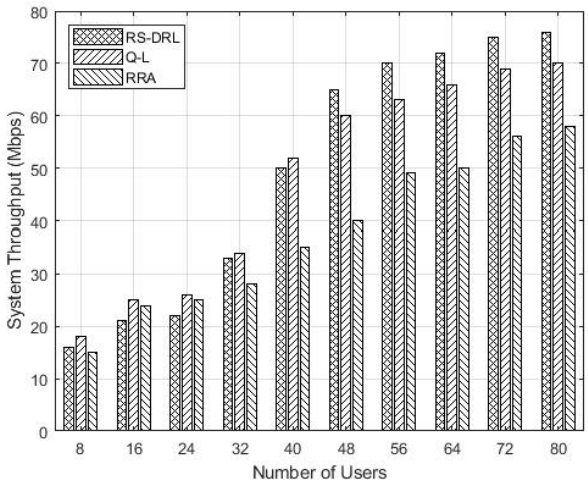

Fig. 4. Trend of System Throughput with Users

as a fast rate high power slice, slice s_2 is denoted as a medium demand slice, and slice s_3 is with the lowest rate and power slice. In the initial state, it is assumed that the arriving users follow the Poisson distribution, and the bandwidth resource size requested by each user is also random. MVNO distributes slices according to user needs to maximize long-term benefits.

To avoid local optimization of the goal, our RS-DRL algorithm uses the $\epsilon - greedy$ strategy. We select these parameters according to the common settings in [31]. For the sake of simplicity, most of the parameters of the systems and algorithms in the evaluation are listed in Table II.

B. Performance Analysis

The learning rate of RS-DRL affects the performance of the algorithm. Learning rate is the most critical parameter for adjusting the depth of training neural network. When the learning rate is set too low, the agent can only see the current reward, which may affect the long-term reward situation, and it takes a long time to converge to the optimal strategy. Conversely, if the learning rate is too high, the algorithm may not converge to the optimal strategy. This is a drawback of gradient descent used in deep learning networks. We perform $10,000$ iterations at three different learning rates (i.e. learning rate= 0.001, 0.01, 0.1, respectively) and obtain the results of Fig.2. As shown in the figure, at the 1000-th iteration, the RS-DRL with 0.1 starts to converge. According to (25), if the learning rate is high, the Q-value of the selected actions will become extremely large. Larger Q-values make it more likely that these actions will be selected again, so the chances of exploring other actions will be lower and the performance decreases rapidly. The RS-DRL with a learning rate of 0.01 starts to converge in about 4000 iterations. Compared with the learning rate of 0.001, it has more stable performance and faster convergence speed. It can get a higher reward than 0.1. With a learning rate of 0.01, the RS-DRL has the best performance in terms of average reward value and convergence rate compared to other learning rates. Therefore, we choose 0.01 as the final learning rate of our RS-DRL algorithm.

C. Comparison Algorithm

1) Convergence of Deep Reinforcement Learning: Here, we show the average speed comparison under the RS-DRL and Q-L methods. As shown in Fig.3, in 200 time slots, we have tripled the user's request arrival rate, $\iota_1 = 48$ requests/hour. When the learning rate of both is chosen to be 0.01, the average rate of the RS-DRL algorithm is significantly higher than that of Q-L. Specifically, the deep reinforcement learning algorithm can effectively deal with the problem of large state space and improve the convergence speed. The performance of RS-DRL is significantly higher than the performance of the Q-L algorithm.

2) System Benefits: In this part, we consider the system benefits and compare them with Q-L and RRA algorithms. Taking into account the amount of resources sliced and more user requests, we increase the number of base stations in the MVNO to 10, and then continuously increase the number of users requested to evaluate system benefits. Fig.4 shows the system throughput of Q-L, RS-DRL and RRA. It can be seen that when the number of user requests is increased to 40, the system throughput of RS-DRL is lower than that of Q-L algorithm. When the user continues to increase, the throughput of RS-DRL system is higher than Q-L. This is because the Q-L algorithm focuses on short-term rewards. The RRA algorithm is slightly higher than the RS-DRL except when it reaches 16 to 24 users, and then it is at the lowest throughput. This is because RRA has randomness. When the number of users is small, it takes up too much resources. If the remaining resources of a slice are insufficient or exceed the power constraint, the user's request is rejected.

Next, we consider the utility. We run the evaluation 10 times and gradually increase the number of users to determine the load limit of each algorithm in the dynamic environment scenario. Fig.5 is a comparison of the total utility of the three algorithms. The total utility increases as the number of users

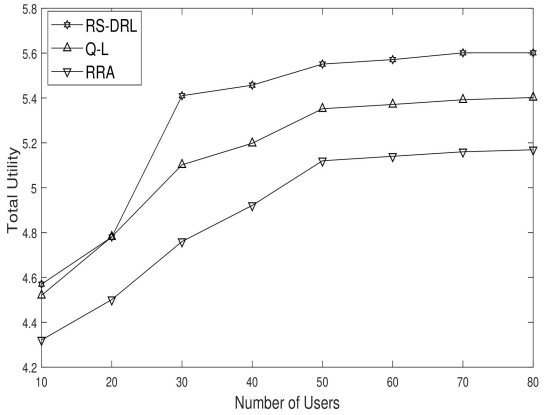

Fig. 5. System Utility Comparison

increases. When the number of users increases to a certain value, the total utility of the MVNO reaches a stable value. Because the overall slice resource is limited, the RS-DRL makes full use of resources by reasonably allocating users for the slice. RS-DRL has the advantage of mining the maximum action value of potential rewards. It can be seen from the figure that RS-DRL is superior to Q-L and RRA algorithms as a whole.

VI. Conclusion

In this paper, we study the resource allocation of slice for 5G RAN. We consider three types of slices, and the MVNO choose which slice to access based on the arriving user request to rationally allocate resources to maximize long-term benefits. To ensure the long-term benefits of MVNO, we describe the optimization problem as SMDP and propose the RS-DRL algorithm. Evaluation results show that the RS-DRL algorithm has the best performance. Compared with traditional reinforcement learning algorithms, RS-DRL can also greatly improve network performance.

References

[1] T. Qiu, X. Wang, C. Chen, M. Atiquzzaman and L. Liu, "TMED: A Spider-Web-Like Transmission Mechanism for Emergency Data in Vehicular Ad Hoc Networks, " in IEEE Transactions on Vehicular Technology, vol. 67, no. 9, pp. 8682-8694, Sept. 2018

[2] Li S, Li D X, Zhao S. 5G Internet of Things: A Survey[J]. Journal of Industrial Information Integration, 2018.

[3] T. Qiu, N. Chen, K. Li, M. Atiquzzaman and W. Zhao, "How Can Heterogeneous Internet of Things Build Our Future: A Survey, " in IEEE Communications Surveys and Tutorials, vol. 20, no. 3, pp. 2011-2027, 2018.

[4] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, X. Shen. TOFFEE: Task Offloading and Frequency Scaling for Energy Efficiency of Mobile Devices in Mobile Edge Computing. IEEE Transactions on Cloud Computing, accepted, to appear, DOI 10.1109/TCC.2019.2923692.

[5] S. Chen, F. Qin, B. Hu, X. Li and Z. Chen, "User-centric ultra-dense networks for 5G: challenges, methodologies, and directions, " in IEEE Wireless Communications, vol. 23, no. 2, pp. 78-85, April 2016.

[6] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, X. Shen. Energy Efficient Dynamic Offloading in Mobile Edge Computing for Internet of Things. IEEE Transactions on Cloud Computing, accepted,to appear, DOI 10.1109/TCC.2019.2898657.

[7] Chih-Lin I , Han S , Xu Z , et al. New Paradigm of 5G Wireless Internet[J]. IEEE Journal on Selected Areas in Communications, 2016.

[8] Y. Chen, J. Huang, C. Lin and J. Hu, "A Partial Selection Methodology for Efficient QoS-Aware Service Composition," in IEEE Transactions on Services Computing, vol. 8, no. 3, pp. 384-397, 1 May-June 2015.

[9] Massimo C , Toktam M . Softwarization and virtualization in 5G mobile networks: benefits, trends and challenges[J]. Computer Networks, 2018.

[10] Jiang M , Xenakis D , Costanzo S , et al. Radio Resource Sharing as a service in 5G: A software-defined networking approach[J]. Computer Communications, 2017.

[11] Zhang H , Liu N , Chu X , et al. Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges[J]. IEEE Communications Magazine, 2017.

[12] Richart, Matłas, Baliosian J , Serrat J , et al. Slicing in WiFi Networks Through Airtime-Based Resource Allocation[J]. Journal of Network and Systems Management, 2018.

[13] B. Matthiesen, O. Aydin and E. A. Jorswieck, "Throughput and Energy-Efficient Network Slicing, " WSA 2018; 22nd International ITG Workshop on Smart Antennas, Bochum, Germany, pp. 1-6 , 2018.

[14] A. E. Amri and A. Meddeb, "Resource Allocation Heuristics for Network Virtualization, " 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), Hammamet, pp. 55-62, 2017.

[15] Yonghua W , Zifeng Y , Pin W , et al. A survey of dynamic spectrum allocation based on reinforcement learning algorithms in cognitive radio networks[J]. Artificial Intelligence Review, 2018.

[16] Wenchao X , Jun Z , Quek T Q S , et al. Power Minimization Based Joint Task Scheduling and Resource Allocation in Downlink C-RAN[J]. IEEE Transactions on Wireless Communications, 2018.

[17] Shah S A W , Khattab T , Shakir M Z , et al. Association of Networked Flying Platforms with Small Cells for Network Centric 5G+ C-RAN[C]// IEEE International Symposium on Personal. IEEE, 2018.

[18] Wang L, Gelenbe E. Adaptive Dispatching of Tasks in the Cloud[J]. IEEE Transactions on Cloud Computing, 2018.

[19] Dighriri M , Ali S D A , Myoung Lee G , et al.2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA) - Resource Allocation Scheme in 5G Network Slices[J]. 2018.

[20] Liu H , Jia H , Chen J , et al. Computing Resource Allocation of Mobile Edge Computing Networks Based on Potential Game Theory[J]. 2019

[21] Qian W , Jing H , Yanjiao C , et al. PROST: Privacy-Preserving and Truthful Online Double Auction for Spectrum Allocation[J]. IEEE Transactions on Information Forensics and Security, 2018.

[22] Afzalirad M , Shafipour M . Design of an efficient genetic algorithm for resource-constrained unrelated parallel machine scheduling problem with machine eligibility restrictions[J]. Journal of Intelligent Manufacturing, 2015.

[23] Acemoglu D, Restrepo P. Artificial Intelligence, Automation and Work[J]. Social Science Electronic Publishing, 2018.

[24] He Y , Zhao N , Yin H . Integrated Networking, Caching, and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach[J]. IEEE Transactions on Vehicular Technology, 2018.

[25] Yonghua W , Zifeng Y , Pin W , et al. A survey of dynamic spectrum allocation based on reinforcement learning algorithms in cognitive radio networks[J]. Artificial Intelligence Review, 2018.

[26] Karbowski, Andrzej et al. Critical Infrastructure Risk Assessment Using Markov Chain Model. Journal of Telecommunications and Information Technology.2019.

[27] Third Generation Partnership Project (3GPP), "Technical Specification Group Services and System Aspects; Policy and charging control architecture," 3GPP TS 23.203, Jun. 2016.

[28] Celik, Abdulkadir Tsai, Ming-Cheng Radaydeh, R.M, et al. Distributed User Clustering and Resource Allocation for Imperfect NOMA in Heterogeneous Networks. IEEE Transactions on Communications.2019.

[29] Van Huynh N , Hoang D T , Nguyen D N , et al. Optimal and Fast Real-time Resource Slicing with Deep Dueling Neural Networks[J]. IEEE Journal on Selected Areas in Communications, 2019.

[30] Van Hasselt H , Guez A , Silver D . Deep Reinforcement Learning with Double Q-learning[J]. Computer Science, 2015.

[31] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji and M. Bennis, "Optimized Computation Offloading Performance in Virtual Edge Computing Systems Via Deep Reinforcement Learning," in IEEE Internet of Things Journal, 2019.