Reinforcement Learning Based Resource Allocation for Network Slicing in 5G C-RAN

Xiaofei Wang, Tiankui Zhang
Beijing Laboratory of Advanced Information Networks,
Beijing University of Posts and Telecommunications, Beijing, China
Email: {wangxiaofei, zhangtiankui}@bupt.edu.cn

***Abstract*—In network slicing enabled 5G cloud radio access networks (C-RAN), we study the network slice resource allocation to ensure the differentiated performance requirements of diversified services from mobile networks and improve the revenue of operators. Considering the characteristics of 5G C-RAN architecture, we propose a network slice resource allocation framework, which is composed of an upper layer, which performs the mapping of virtual protocol stack functions; and a lower layer, which manages radio remote unit (RRU) association, subchannel and power allocation. We model a utility maximization problem based on the proposed framework. Then we proposed a reinforcement learning based two-stage network slice resource allocation algorithm, which uses the multi-agent Q-learning process to reduce the complexity of the Q-value table. Simulation results demonstrate that the proposed algorithm can improve the whole network utility while ensuring the network performance of the virtual network operators compared with the baseline schemes.**

***Keywords*—5G, C-RAN, network slicing, resource allocation, Q-learning**

I. Introduction

Network virtualization has been introduced as a promising technology to meet the differentiated service requirements of different scenarios in 5G. It allows network sharing to be performed as a multi-tenant, so that multiple virtual networks can be created on the physical network hardware, each virtual network being called a network slice, and the owner of each network slice being called a tenant or virtual network operator (VNO) [1].

After the virtualization technology is introduced, the physical resources such as spectrum, antenna, power, and computing entity in the access networks are virtualized into slices, and shared among the VNOs in the way that the base station (BS) is actually owned [2]. In addition, the fact that the building base band unit (BBU) pool can be virtualized in the cloud data center facilitates the implementation of network slicing in C-RAN. While retaining the characteristics of centralization, collaboration and green energy saving, the C-RAN architecture reconstructs the BBU as the centralized unit (CU) and distributed unit (DU). Moreover, the CU/DU are implemented by high-performance general processors, which have the advantage of flexible resource orchestration [3]. At this point, each protocol function in C-RAN is virtualized into virtual network function (VNF) through software executed on the general processor, and the ordered VNFs in each slice constitutes function chains that share the network infrastructure.

Existing researchs on resource allocation for wireless access network slicing can be divided into two categories: radio resource allocation and VNF mapping. For radio resource allocation, [4] proposed a two-level spectrum allocation mechanism based on auction algorithm under the C-RAN architecture, including low-level auctions between users and VNOs and high-level auctions between VNOs and C-RAN operators to maximize the seller's income. [5] solved the slice bandwidth allocation problem based on deep reinforcement learning to maximize the weighted sum of spectrum efficiency and quality of experience (QoE). In [6], continuous convex approximation and complementary geometric programming algorithm were used to allocate subcarriers and powers to users in a multi-cell scenario to maximize the system throughput. For VNF mapping, [7] modularized the LTE protocol stack functions into different VNFs and solved the VNF mapping problem iteratively. In [8], the protocol stack function mapping was represented as 0-1 planning to balance network load by the optimal general server selection. [9] proposed a protocol stack mapping scheme in 5G CU/DU architecture to maximize the utility associated with the total delay and server-repair cost.

However, the above researchs consider the network slice resource allocation problem merely on the RRU side or BBU side. There is currently very little work to extend the network slicing problem to both the RRU side and BBU side. [10] proposed a two-level resource management framework, including user access control, user association, radio resource allocation and BBU capacity allocation, to maximize the multi-tenant system throughput. However, [10] ignored the advantages of flexible resource orchestration after BBU is reconstructed into CU/DU in 5G C-RAN.

The purpose of this study is to develop a two-level resource allocation framework for 5G C-RAN multi-tenant, in which the upper layer and lower layer correspond to the CU/DU side and RRU side respectively. For the lower-layer, the users in each slice receive a set of resources (i.e., associated RRU, subchannels, and transmission power levels) to obtain the service rate of each slice; for the upper-layer, the protocol stack functions in each slice complete the mapping on the virtual CU/DU nodes. After that, a network slice resource allocation scheme based on Q-learning algorithm is proposed to maximize the total utility of the whole networks.

This work is supported by the National Natural Science Foundation of China (No. 61971060 and No. 61502046).

978-1-7281-1973-1/19/$31.00 ©2019 IEEE

II. System Model

In this paper, we consider 5G C-RAN multi-tenant system. The system architecture is shown in Fig. 1. We assume a specific area compose of a set of RRUs, $\mathcal{J} = \{1, \ldots, J\}$. The total bandwidth of B Hz is divided into a set of subchannels, $\mathcal{C} = \{1, \ldots, C\}$. These subchannels are shared by all RRUs through orthogonal frequency-division multiple-access (OFDMA), where the downlink transmission is considered. Every RRU can provide services for a set of slices, $\mathcal{K} = \{1, \ldots, K\}$, where the slice k has a set of users $\mathcal{N}_k = \{1, \ldots, N_k\}$. The VNF function chain of the slice k is represented as $\{f_1^k, f_2^k, \ldots, f_M^k | f_m^k \in F\}$, where $\{f_x | x = 1, 2, \ldots, X\}$ represents all VNF types. The infrastructure provides services for VNOs through management and orchestration (MANO).

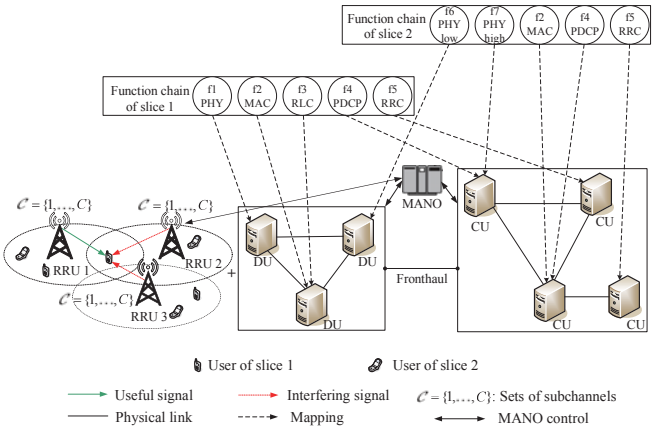

Fig. 1. Network virtualization architecture of 5G C-RAN

Multi-tenant network slices can be regarded as a resource allocation process, which involves: 1) RRU allocation, 2) subchannel allocation, 3) power allocation, and 4) VNF mapping. Let h_{j,c,n_k} and P_{j,c,n_k} be the channel power gain and allocated power, respectively, of the link from RRU j to user n_k of slice k on subchannel c. Due to OFDMA limitation, each user is assigned to one RRU. To avoid intra-cell interference, orthogonal subchannel assignment is assumed among users in a cell. The binary-valued user-association factor (UAF) $\varphi_{j,c,n_k} \in \{0, 1\}$ represents both subchannel allocation and RRU assignment indicator, i.e., $\varphi_{j,c,n_k} = 1$ when RRU j allocates subchannel c to user n_k of slice k, and $\varphi_{j,c,n_k} = 0$, otherwise. The rate of user n_k on subchannel c of RRU j can be expressed as

$$R_{j,c,n_k} = B_c \log_2 \left[1 + \frac{P_{j,c,n_k} h_{j,c,n_k}}{\sigma^2 + I_{j,c,n_k}} \right], \tag{1}$$

where

$$I_{j,c,n_k} = \sum_{\forall j' \in J, j' \neq j} \sum_{\forall k \in K} \sum_{\forall n'_k \in N_k, n'_k \neq n_k} P_{j',c,n'_k} h_{j,c,n'_k}, \tag{2}$$

is the interference from RRU j to user n_k on subchannel c, σ^2 is the Additive White Gaussian Noise (AWGN) power, B_c is the bandwidth of subchannel.

Therefore, the rate of network slice k can be expressed as

$$R^k = \sum_{j \in J} \sum_{n_k \in N_k} \sum_{c \in C} \varphi_{j,c,n_k} R_{j,c,n_k}. \tag{3}$$

When the VNF function chain of each slice performs data processing, the data processing rate of the first VNF is the service rate of network slice. But after each VNF process, the data flow rate in the function chain may change. We take this property into account and let β_m be the data rate expansion ratio of VNF m and R_m^k be the data processing rate of VNF m in the function chain of network slice k. Then the data processing rate of the VNF can be further expressed as

$$R_{m+1}^k = R_m^k \beta_m. \tag{4}$$

Fig. 2 shows an example of the VNF data processing rate in the function chain of network slice k on the basis of the data rate expansion ratio in this model.

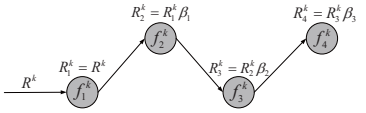

Fig. 2. An example of processing rate by data rate expansion ratio

Different from dedicated hardware, VNF can be deployed on any general server in the infrastructure, and multiple VNFs in the same function chain can also be deployed on the same server node. The VNF needs to occupy the resources of the general server, such as CPU, memory, disk, etc., and the resource demand is usually related to the amount of data that the VNF needs to process. In this paper, to simplify the complexity of the model, various resources on nodes are unified into computing resources, and define the amount of computing resources required by the VNF to be linear with the data rate that the VNF needs to process. Let α_m be the correlation coefficient between the VNF computing resource requirement and the data processing rate, then the computing resource requirement of VNF m in the function chain of network slice k is expressed as

$$v_m^k = \alpha_m R_m^k. \tag{5}$$

The physical networks are regarded as an undirected graph and defined as $G = (N, L)$, where $N = \{1, 2, \ldots, n\}$ represents the set of server nodes in DU and CU pool, $L = \{1, 2, \ldots, l\}$ represents the set of links between nodes. Specifically, the number of DU nodes is N_D, the other nodes are CUs, and the link between DU nodes and CU nodes is fronthaul link. Then the number of VNFs that the function chain of network slice k deploys on the DU nodes can be expressed as

$$W = \sum_{f_m^k \in k} \sum_{n \in N_D} \eta_{m,n}^k, \tag{6}$$

where $\eta_{m,n}^k \in \{0, 1\}$ represents the mapping relationship between VNFs and physical nodes, i.e., $\eta_{m,n}^k = 1$ when VNF f_m^k is deployed on node n, and $\eta_{m,n}^k = 0$, otherwise.

The end-to-end delay of the function chain of network slice k consists of processing delay and transmission delay. The processing delay can be expressed as

$$D_{proc}^k = \sum_{f_m^k \in k} d_{f_m^k}, \tag{7}$$

The transmission delay can be expressed as

$$D_{tran}^k = \sum_{f_m^k \in k} \sum_{n, n' \in N} \eta_{m,n}^k \eta_{m+1,n'}^k h_{n,n'} d_{n,n'}, \forall n \neq n', \tag{8}$$

where $h_{n,n'}$ represents the number of hops between node n and node n', $d_{n,n'}$ represents the transmission delay between node n and node n'.

Then the end-to-end delay of the function chain of network slice k can be expressed as

$$D^k = D^k_{tran} + D^k_{proc}. \tag{9}$$

III. Problem Formulation

The optimization goal of this model is to maximize the service utility of the whole network, where the revenue comes from the service rate, and the expenditure comes from the VNF deployment cost and end-to-end delay loss. The service rate revenue can be expressed as $u^k_R = R^k \delta^k_R$, where δ^k_R represents the unit price of service rate. The deployment cost can be expressed as $u^k_{CPU} = \sum_{f^k_m \in k} \sum_{n \in N} \eta^k_{m,n} v^k_m \delta^r_n$, where δ^r_n represents the unit price of computing resources on node n. The end-to-end delay loss can be expressed as $e^k_{QoS} = D^k \delta^k_D$, where δ^k_D represents the unit price of delay. So the total expenditure is $u^k_E = u^k_{CPU} + e^k_{QoS}$. The service utility of network slice k can be expressed as $u^k = u^k_R - u^k_E$.

Therefore, the optimization problem is

$$\max_{P,\varphi,\eta} \sum_{k \in K} u^k \tag{10a}$$

$$\text{s.t.} \sum_{j \in J} \sum_{n_k \in N_k} \sum_{c \in C} \varphi_{j,c,n_k} R_{j,c,n_k} \geq R^{rsv}_k, \forall k \in \mathcal{K}, \tag{10b}$$

$$\sum_{k \in K} \sum_{n_k \in N_k} \sum_{c \in C} P_{j,c,n_k} \leq P^{\max}_j, \forall j \in \mathcal{J}, \tag{10c}$$

$$\sum_{k \in K} \sum_{n_k \in N_k} \varphi_{j,c,n_k} \leq 1, \forall j \in J, \forall c \in \mathcal{C}, \tag{10d}$$

$$\left[\sum_{c \in C} \varphi_{j,c,n_k}\right] \left[\sum_{\forall j' \neq j} \sum_{c \in C} \varphi_{j',c,n_k}\right] = 0, \forall n_k \in \mathcal{N}_k, \tag{10e}$$

$$\sum_{n \in N} \eta^k_{m,n} = 1, \tag{10f}$$

$$\sum_{k \in K} \sum_{f^k_m \in k} \eta^k_{m,n} v^k_m \leq V^{\max}_n, \forall n \in N, \tag{10g}$$

$$\sum_{k \in K} R^k_{W+1} \leq B_{fh}, \tag{10h}$$

$$D^k \leq D^k_{\lim}. \tag{10i}$$

Constraint (10b) ensures that each slice is guaranteed a minimum data rate. Constraint (10c) indicates that the transmit power of each RRU does not exceed its maximum transmit power. Constraint (10d) indicates that one subchannel in each cell can only be provided to one user. Constraint (10e) implies that each user can be associated to only one RRU. Constraint (10f) indicates that one VNF in each slice can only be mapped to one node. Constraint (10g) indicates that the sum of computing resources occupied by the VNFs on any node is less than the total amount of computing resources of the node. Constraint (10h) indicates that the sum of bandwidths occupied by data flows is less than the total bandwidth of fronthaul link. Constraint (10i) indicates that the end-to-end delay of the function chain does not exceed its delay limit.

IV. Resource Allocation Based On Q-learning

This paper proposes a two-stage Q-learning (TSQL) algorithm, including function chain deployment based on Q-Learning and radio resource allocation based on Q-Learning, to avoid the problem of excessive Q space caused by solving all problems with one Q-learning. Firstly, based on the minimum data rate requirement of each slice, execute stage 1 of TSQL algorithm to obtain VNF mapping method η^* and update; then based on η^*, execute stage 2 of TSQL algorithm to obtain user association and power allocation method φ^*, P^*. Finally, the approximate optimal user association method φ^*, power allocation method P^* and VNF mapping method η^* are obtained.

A. Q-learning algorithm

The optimization goal of Q-learning is to maximize the cumulative rewards from a long-term perspective. The cumulative rewards, also known as the value function, can be expressed as $V^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r(s,a)\right]$, where s represents the state, a represents the action, π represents the policy, $r(s,a)$ is the reward function, which reflects the learning goal, and $\gamma \in [0,1)$ is the discount factor, indicating the attenuation degree of the reward. According to the dynamic programming equation, there is at least one optimal strategy π^* that makes the following equation

$$V^{\pi^*}(s) = \max_{a \in \mathcal{A}} E[r(s,a)] + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}(a) V^{\pi^*}(s'), \tag{11}$$

where s' is the next state when the environment state changes.

Since the state transition probability is difficult to calculate during the Q-learning process, Q-learning stores state and action information into a Q table $Q^\pi(s,a)$. The objective of the Q-learning is to find optimal policy π^*, and estimate the optimal state-action value function $Q^*(s,a) := Q^\pi(s,a), \forall s, a$, which can be shown that

$$\pi^*(s) = \arg\max_a Q^*(s,a), \forall s \in \mathcal{S}. \tag{12}$$

B. Function chain deployment based on Q-learning

We assume that the MANO considers the deployment of function chains one by one. With the above circumstances, the agent, state, action and reward function of the stage 1 of TSQL algorithm are defined as follows. Specifically, the stage 1 of TSQL algorithm is given in Algorithm 1.

Agent: VNF m in the function chain of network slice k.

State: The basic event of the state space $\mathcal{S}_{k,m}$ is the mapping state of each VNF m in the network slice k. Therefore, the state $s_{k,m}$ of a certain moment can be represented by the deployment relation matrix, which indicates the deployment relationship between the VNF and physical server node at the current moment.

Action: According to the state information at each moment, the current VNF needs to select a physical server node for deployment. In this paper, the number of physical server nodes

is N, and each action $a_{k,m}$ in the action set corresponds to a node, so there are N kinds of actions. The action set $\mathcal{A}_{k,m}$ is represented as $\{1, 2, \ldots, n\}$.

Reward function: When VNF in the slice selects actions from the action set, the reward function is also needed to judge the merits of the selected actions. We define the reward function as the respective utility of each slice. If the action selection does not satisfy the constraints (10b)-(10i), set the reward function to fixed -1.

$$r_{k,m} = \begin{cases} u^k, & \mathrm{C(10b)} - \mathrm{C(10i)} \\ -1, & \text{otherwise.} \end{cases} \tag{13}$$

C. Radio resource allocation based on Q-learning

Similarly, the stage 2 of TSQL algorithm is used to solve the user association and power allocation in the 5G C-RAN multi-tenant system. Its agent, state, action and reward function are defined as follows. Specifically, the stage 2 of TSQL algorithm is given in Algorithm 2.

Agent: RRU $j, \forall 1 \leq j \leq J$.

State: The algorithm takes the whole system as environment interacting with agents. The environment is set to a single-state environment because the environment state changes are not involved in the optimization problem. But the value of the reward function changes with the action selection, which causes the Q table to be updated.

Action: For RRU j, the action set $\mathcal{A}^j$ is defined as a series of vectors, and each vector represents the user and power selection of the RRU on all subchannels, which can be expressed as $a^j = \left[a^{j,1}, a^{j,2}, \ldots a^{j,c}, \ldots a^{j,C}\right]$, where $a^{j,c} = \{n_k, p_{j,c,n_k}\}, \forall k \in \mathcal{K}$, n_k represents the user associated with the subchannel c of RRU j, i.e., $\varphi_{j,c,n_k} = 1$. $p_{j,c,n_k} \in \{\rho_0, \rho_1, \rho_2\}$ represents the optional power levels on each subchannel.

Reward function: Each RRU j selects an action from the action set and performs an effect on the environment. The environment will inform the agent of the merits of its action selection through certain feedback. Considering that the optimization goal of this algorithm is to maximize the system utility, when the constraint conditions (10b)-(10i) are met, the reward function of each agent is defined as the system utility obtained after the RRU selects the respective service users and power; otherwise, it is defined as a negative feedback.

$$r_j = \begin{cases} \sum\limits_{k \in K} u^k, & \mathrm{C(10b)} - \mathrm{C(10i)} \\ -1, & \text{otherwise.} \end{cases} \tag{14}$$

D. Algorithm analysis

The above TSQL algorithm based on distributed Q-learning can decompose the large Q table into multiple small Q tables, which has greater advantages than centralized Q-learning. For stage 2, it can be decomposed into j parallel RRUs which execute action selection. In centralized Q-learning, the system combines the actions of all agents involved in Q-learning into a joint action, so that the total amount of joint actions is as high as $|\mathcal{A}|^j$. In addition, the system needs to collect real-time information of each agent through the central controller and maintain a Q table with its dimension up to $|\mathcal{S}| \cdot |\mathcal{A}|^j$. In order to fully cover the high-dimensional Q table, the agent needs to make extremely many action selection, and the algorithm complexity to realize the reasonable update of Q table is high, so the centralized Q-learning is not suitable for solving the optimization problem of this paper. In distributed Q-learning, each agent involved in Q-learning independently maintains its own Q table, so the overall dimension of the Q table is $j \cdot |\mathcal{S}| \cdot |\mathcal{A}|$, and the complexity of the algorithm is low. In addition, each agent does not need to interact with each other during learning process, which reduces the system overhead. By reasonably designing the reward function to be determined by the actions of all agents, distributed Q-learning can also obtain the overall optimal solution. In [11], the convergence of distributed Q-learning algorithm is proved.

In the above Q-learning process, Q value will be updated after the agent obtains the reward function each time, and gradually approaches the optimal Q value. The update rule of

Algorithm 1 The stage 1 of TSQL algorithm

Input: the minimum data rate of each slice R_k^{rsv}
Output: function chain deployment method η^*
for each function chain of network slice k **do**
  Check whether the physical resource can meet the total resource requirement of the function chain
  if meet the total resource requirement **then**
    for each VNF m in the function chain of network slice k **do**
      initialize the Q-table $Q_{k,m}(s, a) = 0$
    end for
    evaluate the starting state $s_{k,m} \in \mathcal{S}_{k,m}$
    loop
      for each VNF m in the function chain of network slice k **do**
        generate a random number $x_{k,m} \in [0, 1]$
        if $x_{k,m} < \varepsilon$ **then**
          select action randomly
        else
          select the action $a_{k,m} \in \mathcal{A}_{k,m}$ characterized by the maximum Q-value
        end if
        execute $a_{k,m}$
      end for
      for each VNF m in the function chain of network slice k **do**
        receive an immediate reward $r_{k,m}$
        observe the next state $s'_{k,m}$
        update the Q-table $Q_{k,m}(s, a)$ according to (15)
      end for
    end loop
  end if
end for

Algorithm 2 The stage 2 of TSQL algorithm

Input: function chain deployment method η^*
Output: user association φ^* and power allocation method P^*
for RRU j **do**
  initialize the Q-table $Q_j(s, a) = 0$
end for
loop
  for each RRU j **do**
    generate a random number $x_j \in [0, 1]$
    if $x_j < \varepsilon$ **then**
      select action randomly
    else
      select the action $a_j \in \mathcal{A}_j$ characterized by the maximum Q-value
    end if
    execute a_j
  end for
  for each RRU j **do**
    receive an immediate reward r_j
    update the Q-table $Q_j(s, a)$ according to (15)
  end for
end loop

Q value can be described as

$$Q(s_t, a_t) = (1-\alpha) Q(s_t, a_t) + \alpha \left[r(s_t, a_t) + \gamma \max_{a' \in \mathcal{A}} Q(s', a') \right] \tag{15}$$

where $\alpha \in (0, 1]$ represents the learning rate, which measures the convergence rate of Q-learning algorithm. When the value of α is small, the learning time is large; on the contrary, the algorithm may not converge. There are many ways to choose actions based on the current Q-value estimation, in this paper we use ε-greedy strategy [12].

V. Performance Evaluation

In the simulation, the RRUs are distributed uniformly in 2×2 km^2 area. The number of VNFs contained in each function chain is uniformly distributed from 2 to 5. The computing resource demand coefficients α_m are all 0.5, the data rate expansion ratios β_m are 0.2, 0.7, 0.9, 1.0 and 1.0, respectively. The VNF data processing delay is uniformly distributed from 0.2 to 0.8 ms and the link transmission delay is uniformly distributed from 0.1 to 0.5 ms. In this paper, we still focus on service rate, followed by delay loss. Therefore, the service rate unit price δ_R^k is 3.5, the mean unit price of the physical node computing resource δ_n^r is 2.5, the end-to-end delay loss unit price δ_D^k is 3. The remaining parameters are shown in Table I.

TABLE I: Simulation Parameters

Parameters	Value
Number of cells J	4
Number of subchannels C	2
Number of slices K	3
System bandwidth B	4 MHz
Pathloss from RRU to user	$37.6 \log 10(d[km]) + 128.1$ dB
Optional power levels ρ_0, ρ_1, ρ_2	0, 19, 39 dBm
Noise power	-174 dBm/Hz
Max RRU transmit power $P_j^{\max}$	43 dBm
Number of server nodes N	9
Number of DU nodes N_D	4
Number of VNF types	5
Minimum rate of slice R_k^{rsv}	10 Mbps, 5 Mbps, 51 kbps [5]
maximum tolerance delay D_{lim}^k	10 ms
Fronthaul resources B_{fh}	100 Mbps

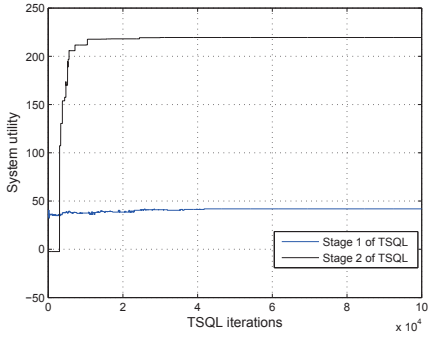

Fig. 3. System utility over TSQL iterations

The convergence behavior of TSQL algorithm can be observed in Fig. 3 when the number of total users is 6, the probability ε in ε-greedy strategy is 0.5, the learning rate α is 0.5, the discounted rate γ is 0.9. It can be seen from Fig. 3 that as the number of iterations increases, the system utility value gradually becomes stable and converges. When the stage 1 of TSQL converges, an approximate optimal VNF mapping scheme can be obtained; through one-step iteration, when the stage 2 of TSQL converges, an approximate optimal user association and power control scheme can be obtained, and the system utility value is further improved. Therefore, the effectiveness of the network slice resource allocation scheme proposed in this paper is proved.

We compare the proposed TSQL algorithm with the following schemes:

- minimum cost function chain deployment and maximum SINR radio resource allocation (MCMS): The VNF in the function chain of network slice preferentially selects the server node according to the minimum resource unit price; each RRU associates users in each slice by the nearest distance, the power allocated to each subchannel is the maximum value of optional power levels.
- function chain deployment based on Q-Learning and maximum SINR radio resource allocation (FQMS): User association and power control are performed as MCMS; the function chain of network slice is deployed as Section IV.
- minimum cost function chain deployment and radio resource allocation based on Q-Learning (MCRQ): The function chain of network slice is deployed as MCMS; user association and power control are performed as Section IV.

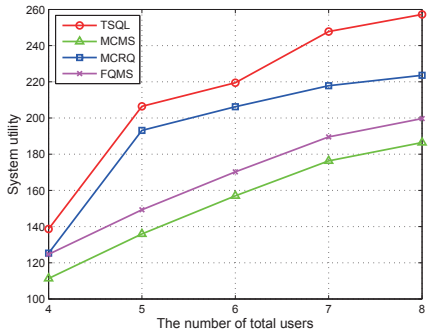

Fig. 4. System utility verus the number of total users

Fig. 4 illustrates the system utility performance with an increasing total number of users. The distribution of users of each slice is as even as possible, that is, when the total number of users is 4, the number of users of each slice is 2, 1, 1, respectively, when the total number of users is 6, it is 2, 2, 2. It can be seen from Fig. 4 that the proposed TSQL algorithm can achieve the best results with acceptable complexity. Although MCRQ algorithm selects the physical node to be deployed for VNF with the minimum resource price priority, it inevitably increases the delay loss, so this algorithm is inferior to TSQL algorithm. The reduction of total expenditure of FQMS algorithm is not enough to offset the reduction of total revenue. MCMS algorithm lags behind other algorithms in terms of revenue increase and expenditure reduction, with the worst performance.

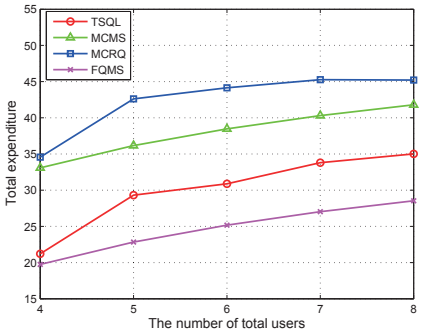

Fig. 5. Total expenditure verus the number of total users

Fig. 5 shows the total expenditure of the 5G C-RAN multi-tenant system as the number of total users increases. With the increase of total users, the total expenditure of the system also increases. According to Fig. 5, the total expenditure brought by MCRQ algorithm is highest, combined with Fig. 4 analysis, MCRQ algorithm increases system utility at a certain overhead. Compared with MCMS algorithm, TSQL algorithm can bring higher system utility at a lower expenditure. The total expenditure of FQMS algorithm is lower than that of MCMS algorithm, which proves the superiority of Q-learning algorithm in reducing system expenditure in this paper.

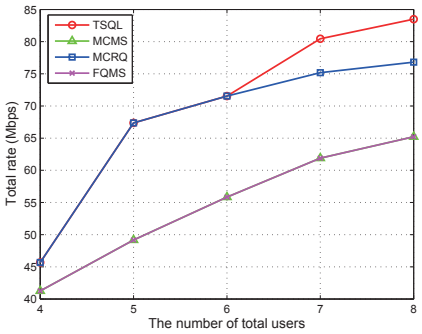

Fig. 6. Total rate verus the number of total users

From Fig. 6, it can be observed that TSQL algorithm proposed in this paper can guarantee the best system service rate. When the number of users is no more than 6, MCRQ algorithm can maintain a similar total rate with TSQL algorithm. However, as the number of users continues to increase, MCRQ algorithm tends to be more gradual due to the limitation of computing resources of physical nodes. The original intention of FQMS algorithm is to increase the service rate of a single user, but increase the interference between users, resulting in a decrease in total rate. MCMS algorithm adopts the same mechanism in user association and power control as FQMS algorithm, and the data rate does not exceed the physical resource limit, so the total rate is the same.

The total utility of the 5G C-RAN multi-tenant system at different service rate unit price is investigated in Fig. 7. As we expected, regardless of the service rate unit price is 5.5 or 3.5, the system utility of TSQL algorithm is always higher than that of MCMS algorithm. In addition, we can see from Fig. 7 that TSQL algorithm can increase the system utility, but there is also an upper limit due to the limited node resources.

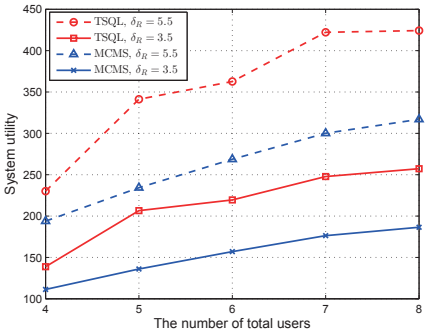

Fig. 7. System utility using different rate unit price

VI. Conclusions

In this paper, a new multi-tenant resource allocation scheme is proposed to solve the network slicing problem with differentiated requirements in 5G C-RAN scenario. The two-stage Q-learning algorithm proposed in this paper achieves the goal of maximizing the total utility of virtual network operators. Simulation results show that compared with the baseline schemes, the proposed algorithm can increase the utility of the whole system at a relatively low expenditure.

References

[1] M. Richart, J. Baliosian, J. Serrat, and J. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–476, Sep. 2016.

[2] S. O. Oladejo and O. E. Falowo, "5G network slicing: A multi-tenancy scenario," in *2017 Global Wireless Summit (GWS)*, Oct 2017, pp. 88–92.

[3] I. Chih-Lin, J. Huang, Y. Yuan, and S. Ma, "5G RAN architecture: C-RAN with NGFI," in *5G Mobile Communications*. Springer, 2017, pp. 431–455.

[4] M. Morcos, T. Chahed, L. Chen, J. Elias, and F. Martignon, "A two-level auction for C-RAN resource allocation," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2017, pp. 516–521.

[5] R. Li, Z. Zhao, Q. Sun, I. Chih-Lin, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74 429–74 441, 2018.

[6] S. Parsaeefard, R. Dawadi, M. Derakhshani, and T. Le-Ngoc, "Joint user-association and resource-allocation in virtualized wireless networks," *IEEE Access*, vol. 4, pp. 2738–2750, 2016.

[7] R. Wen, G. Feng, W. Tan, R. Ni, S. Qin, and G. Wang, "Protocol function block mapping of software defined protocol for 5G mobile networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1651–1665, 2017.

[8] R. Wen, G. Feng, W. Tan, R. Ni, W. Cao, and S. Qin, "Protocol stack mapping of software defined protocol for next generation mobile networks," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[9] Y. Yang, Q. Chen, G. Zhao, P. Zhao, and L. Tang, "The stochastic-learning-based deployment scheme for service function chain in access network," *IEEE Access*, vol. 6, pp. 52 406–52 420, 2018.

[10] Y. L. Lee, J. Loo, T. C. Chuah, and L.-C. Wang, "Dynamic network slicing for multitenant heterogeneous cloud radio access networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2146–2161, 2018.

[11] M. Lauer and M. Riedmiller, "An algorithm for distributed reinforcement learning in cooperative multi-agent systems," in *In Proceedings of the Seventeenth International Conference on Machine Learning*. Citeseer, 2000.

[12] S. Nie, Z. Fan, M. Zhao, X. Gu, and L. Zhang, "Q-learning based power control algorithm for D2D communication," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sep. 2016, pp. 1–6.