VNF Service Chain Deployment Algorithm in 5G Communication based on Reinforcement Learning

Hejun Xuan, Xuelin Zhao, Jianwei Fan, Yahui Xue, Fangfang Zhu and Yanling Li

Abstract—**With the rapidly deployment of 5G communication, network functional virtualization has became one of key technology. However, how to achieve virtual network function service chaining (VNF-SC) adaptively and cost-effectively in an inter datacenter elastic optical network (interDC-EON) has become an interesting and challenging problem. In addition, network function virtualization has became one of the key technology for 5G Commons. In this paper, we propose a resource scheduling framework based on artificial intelligence. In addition, an effective VNF-SC deployment algorithm based on deep reinforcement learning (RL) is designed. In this algorithm, VNF-SC deployed scheme is obtained by using deep Q-learning according to the state of the network and the reward of the action. To verify the efficient of the proposed algorithm, a large number of experiments have been conducted. Experimental results demonstrate that the RL achieves better performance than several benchmarks, in terms of balancing the tradeoff among the overall resource utilization, the vNF-SC request blocking probability.**

Index Terms—**Virtual Network Function (VNF); VNF-service chain (VNF-SC); Reinforcement learning (RL); 5G Communication**

I. Introduction

NETWORK function virtualization (NFV) becomes a hot research field as it can facilitate the flexibility of the network services[1], [2]. NFV decouples network functions from underlying hardware so these functions can run as software images on commodity hardware as well as custom-built hardware[3], [4], [5]. With network function virtualization technology, traditional hardware based network appliances are replaced by software-based virtual network functions (VNFs), and virtual network functions (VNFs) can be realized by using generic network resources[6]. VNFs can be deployed on high performance servers in datacenters (i.e., service chain), and the deployment of new network services can be easily realized by routing data traffic through a series of VNFs on datacenters[7].

In addition to the innovations on the datacenter, NFV also gains advantages from new optical networking architectures (e.g., elastic optical networks (EONs))[8], [9]. This is because VNF-SC can route high-throughput and gusty traffic across several datacenters has to use an optical network as its physical layer [10]. NFV MANO (NFV management and orchestration)[11] is responsible for management the infrastructure, resource and service. It can arrange different VNFs to realize management of the automatic deployment.

In the meantime, recent advances in reinforcement learning (RL) have demonstrated beyond human-level performance in handling large-scale online control tasks [12], [13]. By accumulating action experiences from repeated interactions with the target systems and by reinforcing actions leading to higher rewards, RL is able to learn successful policies progressively. The application of RL in the communication and networking domain has received intensive research interests during the past two years [14], [15]. In [16], the authors enhanced the general deep Q-learning framework in with novel exploration and experience replay techniques to solve the traffic engineering problem. The authors of [17] presented a RL-based framework for datacenter network management and demonstrated a RL agent which can learn the optimal topology configurations with respect to different application profiles.

In this paper, we focus on the VNF-SC deployment problem in inter-DC EONs. The main contributions of this paper are as follows: (1) We propose a resource scheduling framework based on artificial intelligence. (2) An effective VNF-SC deployment algorithm based on reinforcement learning (RL) is designed. In this algorithm, VNF-SC deployed scheme is obtained by using Q-learning according to the state of the network and the reward of the action.

II. Related Works

In recent years, there are more and more researches focusing on the VNF-SC deployment[18], [19]. To investigates the problem of how to optimize the provisioning of VNF-SCs in Inter-Dc EON, taking advantage of the broker-based hierarchical control paradigm for the orchestration of cross-stratum resources and propose to realize incentive-driven VNF-SC provisioning with a noncooperative mixed-strategy gaming approach is investigated[20]. A deep-learning (DL) model is designed to predict future VNF-SC requests, then lightpath establishment and VNF deployment are performed accordingly to pre-deploy resources for the predicted requests

Manuscript received July 27, 2020; revised September 10, 2020. This work is supported by National Natural Science Foundation of China (No. 61572391, 61572417), Science and Technology Department of Henan Province (No.182102210132, 182102210537), Innovation Team Support Plan of University Science and Technology of Henan Province (No.19IRTSTHN014), Nanhu Scholars Program for Young Scholars of XYNU, Youth Sustentation Fund of Xinyang Normal University (No.2019-QN-040), University Students Sustentation Fund of Xinyang Normal University (No.2019-DXS-035,2019-DXS-037)

Hejun Xuan, Xuelin Zhao, Yahui Xue and Fangfang Zhu are with School of Computer Science and Technology, Xinyang Normal University, Xinyang 464000, China; Email:xuanhejun0896@xynu.edu.cn; xlz_cit@xynu.edu.cn; 2825460732@qq.com; 1171606096@qq.com.

Jianwei Fan and Yanling Li are with School of Computer Science and Technology, Xinyang Normal University, Xinyang 464000, China; Henan Key Lab. of Analysis and Application of Education Big Data, Xinyang Normal University, Henan Xinyang, China, 464000. Email:fjwlijie@163.com; lyl75@163.com.

in [21]. Literature[22] addressesed the placement aspect of these service chains by finding the best locations and hosts for the VNFs and to steer traffic across these functions while respecting user requirements and maximizing provider revenue. We propose a novel eigendecomposition-based approach for the placement of virtual and physical network function chains in networks and cloud environments. A novel primal-dual decomposition using column generation that solves exactly a relaxed version of the problem and can serve as a benchmark approach is presented[23]. To minimize the rejection of VNF-SC bandwidth and reduce the energy consumed, a consolidation algorithm based on a migration policy of VNF-SC that considers the revenue loss due to QoS degradation that a user suffers due to information loss occurring during the migrations is proposed[24]. For general IP networks, a mixed integer linear programming (MILP) model to determine the upper bound on the reliability with max-min fairness is established[25]. In addition, an efficient heuristic is developed to address reliable multicast VN mapping with a low computational complexity. Literature [26] studied the SFC Embedding Problem (SFC-EP) with dynamic VNF placement in geo-distributed cloud system and formulated this problem as a Binary Integer Programming (BIP) model aiming to embed SFC requests with the minimum embedding cost. A framework employs a non-cooperative hierarchical game-theoretic mechanism, where the resource brokers and the VNF-SC users play the leader and the follower games, respectively is proposed[27].

III. Problem Description and Mathematical Modeling

A. Problem Description

In General, VNF deployment problem can be divided into two sub-problems, i.e., VNF placement and VNF-SC constructing. The mainly problem of VNF placement is placing the VNF to the suitable virtual machine in the data center. VNF-SC constructing determine the optimal traffic scheduling scheme for the VNF-SCs.

We use a directed graph $G = (V, E)$ to describe a inter-DC EONs, where V and E denote the node set and link set, respectively. v_i, which connected to the data center, denotes the i-th node in $V = \{v_1, v_2, \cdots, V_{N_V}\}$. N_V is the number of the nodes in the network. N_e and e_i represent the number of links and the i-th link in the set of link set $E = \{e_1, e_2, \cdots, e_{N_e}\}$. For each node $v_i (i = 1, 2, \cdots, N_V)$, there are N_{v_i} virtual machine in the node v_i. In addition, each virtual machine can service one VNF at the san time. Thus, one node can hold up N_{v_i} VNF. In this work, the shortest path is selected for each node pair v_i and $v_j (i \neq j)$, we can use $T(v_i, v_j)$ to denote the time delay in the shortest path between node v_i and v_j. $F = \{f_1, f_2, \cdots, f_{N_f}\}$ represent the set of VNF, and $f_i (i = 1, 2, \cdots, N_f)$ denotes the i-th VNF, such as fire wall, load balancer, etc. For each VNF-SC S_i, it can be described as $C_i = (c_{i,1}, c_{i,2}, c_{i,4}, \cdots, c_{i,j}, \cdots)$, where $c_{i,j}$ denotes the j-th network function of the VNF-SC C_i, and $|C_i|$ represents the length of the VNF-SC C_i. Each network function $c_{i,j}$ is realized by one VNF, denoted by $f(c_{i,j})$. $v(c_{i,j})$ is used to present the node of $c_{i,j}$ placed. The traffic between each two nodes has K candidate paths, denoted by $Q_{i,j} = \{Q^1_{i,j}, Q^2_{i,j}, \cdots, Q^k_{i,j}, \cdots, Q^K_{i,j}\}$, where $Q^k_{i,j} (k = 1, 2, \cdots, K)$ represents the k-th candidate path between node v_i and v_j.

B. Mathematical Modeling

In this investigation, we have two objectives, including average of transfer time delay and load of the service.

The first objective is minimizing the average of transfer time delay of all the VNF-SCs. We can express it as follows:

$$\overline{D} = \frac{1}{N_V} \sum_{i=1}^{N_V} \sum_{j=1}^{|C_i|} D(v(c_{i,j}), v(c_{i,j+1})) \tag{1}$$

where $D(v_i, v_j)$ denotes the time delay in the path occupied between the nodes v_i and v_j. If we use $D(s, d)$ to denote the maximum of the time delay bet each two node, thus we have

$$\frac{1}{N_V} \sum_{i=1}^{N_V} \sum_{j=1}^{|C_i|} D(v(c_{i,j}), v(c_{i,j+1})) \leq \frac{1}{N_V} \sum_{i=1}^{N_V} \sum_{j=1}^{|C_i|} D(s, d) \tag{2}$$

So, we can normalized the first objective as

$$F_1 = \frac{\sum\limits_{i=1}^{N_V} \sum\limits_{j=1}^{|C_i|} D(v(c_{i,j}), v(c_{i,j+1}))}{\sum\limits_{i=1}^{N_V} \sum\limits_{j=1}^{|C_i|} D(s, d)} \tag{3}$$

In addition, we have $0 \leq F_1 \leq 1$. The second objective is to reach the goal of load balance on all the node. We can expressed this objective as

$$\sigma^2 = \frac{1}{N_V} \sum_{i=1}^{N_V} \left(\widetilde{N_{v_i}} - \overline{N_{v_i}} \right)^2 \tag{4}$$

where $\widetilde{N_{v_i}}$ denotes the number of virtual machine that has been occupied in node V_i, $\overline{N_{v_i}}$ represents the average number of virtual machine that has been occupied in all nodes. $\overline{N_{v_i}}$ is calculated by

$$\overline{N_{v_i}} = \frac{1}{N_V} \sum_{i=1}^{N_V} \widetilde{N_{v_i}} = \frac{N^T_{vnf}}{N_V} \tag{5}$$

Thus, Eq.(4) can be re-written as

$$\begin{aligned} \sigma^2 =& \frac{1}{N_V} \sum_{i=1}^{N_V} \left(\widetilde{N_{v_i}} - \overline{N_{v_i}} \right)^2 \\ =& \frac{1}{N_V^3} \sum_{i=1}^{N_V} \left(\widetilde{N_{v_i}} - N^T_{vnf} \right)^2 \end{aligned} \tag{6}$$

Note that when all VNFs in all VNF-SCs are assigned to one node, σ^2 will arrive at the maximum value denoted by Γ^2. We can normalize σ^2 by σ^2/Γ^2, where Γ^2 can be easily calculated by

$$\begin{aligned} \Gamma^2 =& \frac{(N_V - 1) \frac{(N^T_{vnf})^2}{N_V^2} + \left(N^T_{vnf} - \frac{N^T_{vnf}}{N_V} \right)^2}{N_V} \\ =& \frac{\frac{(N^T_{vnf})^2}{N_V^2} \left((N_V - 1) + (N_V - 1)^2 \right)}{N_V} \\ =& \frac{(N^T_{vnf})^2 \left(N_V^2 - N_{DC} \right)}{N_V^3} = \frac{(N^T_{vnf})^2 (N_V - 1)}{N_V^2} \end{aligned} \tag{7}$$

So, the second objective function can be expressed as

$$\begin{aligned} F_2 = \frac{\sigma^2}{\Gamma^2} &= \frac{\frac{1}{N_V^3}\sum_{i=1}^{N_V}\left(N_V\widetilde{N_{v_i}} - N_{vnf}^T\right)^2}{\frac{(N_{vnf}^T)^2(N_V-1)}{N_V^2}} \\ &= \frac{\sum_{i=1}^{N_V}\left(N_V\widetilde{N_{v_i}} - N_{vnf}^T\right)^2}{N_V(N_V-1)(N_{vnf}^T)^2} \\ &= \frac{N_V\sum_{i=1}^{N_V}(\widetilde{N_{v_i}})^2 - 2N_{vnf}^T\sum_{i=1}^{N_V}\widetilde{N_{v_i}} + (N_{vnf}^T)^2}{(N_V-1)(N_{vnf}^T)^2} \\ &= \frac{N_V\sum_{i=1}^{N_V}(\widetilde{N_{v_i}})^2 - (N_{vnf}^T)^2}{(N_V-1)(N_{vnf}^T)^2} \end{aligned} \tag{8}$$

Thus, we have $0 \leq F_1 \leq 1$. Now we integrate the two objectives into one to be minimized as follows

$$\min f = \min\{\alpha F_1 + \beta F_2\} \tag{9}$$

where α and β are two weights to adjust the importance of the two objectives with $0 \leq \alpha, \beta \leq 1$, $\alpha + \beta = 1$. So, $0 \leq f \leq 1$.

The decision should be made under some conditions. These conditions constitute the constraints of the problem as follows:

(1) For each node, the number of virtual machine should not greater than its capacity of the node, that is

$$\widetilde{N_{v_i}} \leq N_{v_i} \tag{10}$$

IV. VNF-SC Deployment Algorithm based Reinforcement Learning

A. Reinforcement Learning

Reinforcement learning is one of the important methods of machine learning in the field of artificial intelligence. Reinforcement learning lies its interaction with the environment. This is the difference of reinforcement learning from supervised learning and unsupervised learning. Reinforcement learning is an interactive learning method, which emphasizes learning to obtain evaluative feedback signals in the interaction with the environment. In the absence of the expected output of input signals in various states, the goal of learning is to maximize future returns.Therefore, reinforcement learning has the advantages of self-learning and online learning, and has a wide range of applications in solving complex optimization decision problems with less prior information.

In the process of using reinforcement learning to solve practical problems, the most important thing is to transform a practical problem into a model of reinforcement learning and use relevant algorithms to get optimal strategy results.That is, the state set, action set and feedback function in the environment are defined according to the actual problem to be solved.In the service chain mapping problem, it is defined as follows:

State set. The key to selecting the physical server nodes for each VNF in the service chain for deployment is to select the appropriate placement nodes according to the real-time status of each node. First, the state of a single node is defined based on the number of vCPUs already used by the physical server node, and a threshold value is set for the vCPU usage of each node. If the threshold value is exceeded, the node state is 1; otherwise, the node state is 0. Then consider the state of the global node. If there are N_V nodes, each node has two states, namely 0 and 1, then there are 2^{N_V} states globally.If the binary representation state value is "0010", it means that the resource usage of one of the nodes exceeds the threshold, and the remaining nodes No more than the threshold.In addition to the state of the global node, since the business traffic needs to pass through the VNF in the service chain in order, the current deployment location of VNF has a certain relationship with the deployment location of the last VNF in the service chain, and the traffic will be forwarded between the two adjacent VNF node locations. Therefore, in this paper, the node location of the last VNF deployment is introduced into the state information. Thus, the number of states in the state set is $n = 2^{N_V}N_V$. The state set is represented as

$$S = \{s_1, s_2, \cdots, s_i, \cdots, s_n\}$$

Action set. According to the state information at each moment, it is necessary to select a physical server node for the currently deployed VNF to deploy. Assuming there are N_V nodes, each action in the action set corresponds to one node, and there are N_V kinds of actions. Action set is

$$A = \{a_1, a_2, \cdots, a_j, \cdots, a_{N_V}\}$$

Excitation function According to different situations, different feedback values are set. When the use of node resources selected by the action has exceeded the threshold value, the feedback value is fixed at -50. When the node resources selected by the action do not exceed the threshold value, the excitation function is

$$R = 0 - A[L_r(s_t) - L_r(s_{t+1})] - BD_r \tag{11}$$

where $L_r(s_t)$ represents the load balance of the system at state s_t, D_r denotes the minimum delay between the node selected for this action and the previous VNF deployment node.

In addition, due to the problems of reinforcement learning algorithm such as dimensional disaster, slow convergence speed, exploration and utilization balance, the algorithm in this paper will focus on the deployment of a service chain in a data center or enterprise network, temporarily ignoring the need to deploy a service chain in multiple domains at the same time.In other words, multiple SDN domains and NFV-O domains are temporarily ignored, and only the mapping of service chain on a certain infrastructure is considered. In addition, in the group chain stage of the service chain, flow table is basically issued by SDN controller to control the flow direction between different VNF, which is generally selected according to the shortest path strategy.

B. Proposed VNF-SC Deployment Algorithm

Reinforcement learning tasks are usually described by MDP (Markov Decision Process). As the state transition probability of the model is unknown in the Markov decision process of system modeling, q-learning algorithm is adopted

in this paper to carry out learning control in the model. Q-learning algorithm is a model-independent value iteration method, which does not need to know the state transition matrix in advance to solve the sequential optimization decision problem with delayed return.

In Q-Learning algorithm, the key is to define a state-action matrix $Q(s,a)$, and each value in the matrix represents the value of a 'state-action' pair. Each time an action is selected, it is selected according to the $Q(s,a)$ matrix. Assuming the current state is s_t, the selected action a needs to meet

$$Q(s,\tilde{a}) = \max_{\tilde{a}} Q(s,\tilde{a}) \tag{12}$$

In this paper, ε-greedy exploration is used to select actions. In this mechanism, an action is randomly selected with low probability, and the best action is selected according to the Q matrix with probability $1-\varepsilon$. This mechanism is mainly to prevent the algorithm from falling into the local optimal state and jump out of the local optimal state through the low-probability exploration mechanism.In algorithm design, ε is not fixed, but dynamically changing. $\varepsilon = 10/K$ is selected, where K is the number of learning cycles. The significance of doing so lies in the greater probability of using random method in the selection of early learning actions, which can be better explored. In the later stage of learning, the action selection is more inclined to be generated according to Q matrix calculation.

When the action is selected and executed, the system enters the next state s_{t+1}, and a feedback $r(s_t,a_t)$ of the system can also be obtained. According to Equation (13), the Q matrix is iteratively updated.

$$\begin{aligned} Q(s_t,a_t) =& Q(s_t,a_t) + \alpha_t[r(s_t,a_t) \\ &+ \gamma \max_{a_{t+1}} Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)] \end{aligned} \tag{13}$$

where (s_t,a_t) is the 'state-action' of Markov decision process at time t. s_{t+1} is the state of the time t. $r(s_t,a_t)$ is the return at time t. α_t is the learning factor at time t.

C. Algorithm Description

In Section 5.1, specific algorithm implementation description is given. Algorithm 1 focuses on describing the mapping algorithm process of the entire service chain, as shown below.

Algorithm 2 presents the placement process of VNF based on Q-learning, as shown below.

Algorithm 1: VNF-SC mapping algorithm based Q-learning

```
Receive a service chain placement request
Check whether the infrastructure compute storage
  network resources meet the total resource requirements
  of the service chain request;
if Meet the total resource demand then
    Deployment the head node of the service chain on
    the node with the least load;
    for The others VNF do
        Appropriate VNF placement nodes are selected
        according to q-Learning algorithm;
    end
else
    Denial of service chain placement requests;
end
```

Algorithm 2: VNF deployment algorithm based Q-learning

```
VNF placement request received
Generate a random number γ between (0,1);
if γ < ε then
    Random selection action to place VNF, that is,
    random selection node to place VNF;
else
    According to the Q matrix calculation select the
    optional action to place the VNF in the appropriate
    node;
    Calculate the feedback value r(a_t, s_t) after VNF
    placement;
    According to Equation (13), the Q matrix is updated
    for the next calculation of placement strategy;
end
```

V. Experimental Results and Analysis

A. Simulation Environment

In this paper, a test verification tool is written based on Java language to realize the simulation of the algorithm.The simulation implementation of the algorithm is divided into three modules. The first module is the construction of the underlying network environment, including the simulation of network topology nodes and physical server node resources. Then comes the generating module of the business. Each business contains a service chain. The business arrives in a Poisson flow, and the time completed in the system follows a negative exponential distribution. Finally, the service chain mapping algorithm module based on enhanced learning runs the Q-learning algorithm, makes decisions for the deployment of the service chain in the business, and then checks the next state and feedback of the system after each deployment to update the Q matrix.

Generation of VNF-SC: Assume that the number of VNF types in the network is 10. The length of the service chain is randomly generated from the uniform distribution of [1, 8]. The generation of each VNF in a service chain is randomly selected from the set, and the selected type of VNF is guaranteed not to be selected, so that the VNF type in a service chain is not the same. Service chain requests are generated according to the Poisson process, i.e., the generated time interval follows the exponential distribution $e^{-\lambda t}$, where $\lambda = 1$; Meanwhile, the duration of the service chain in the system follows exponential distribution $e^{-\mu t}$, where $\mu = 1$. The system manages the service chain life cycle according to the above probability distribution law.

Experimental topological generation: The network topology is generated by Brite topology generator. A total of N_V node network topologies are selected and generated. The connection relationship between nodes and the delay of links are generated by the topology generator. Assuming that each server node has 10 vCPU, this paper assumes that each

vCPU can host one VNF, so 10 VNFs can be placed on each server node.

VNF mapping algorithm based on Q-Learning: The key of Q-Learning algorithm lies in the Q matrix. Since there are N_V nodes in the experimental network topology, the number of states in the state set is $2^{N_V}N_V$, the number of actions in the action set is N_V, and the number of 'state-action' pairs in the Q matrix is $2^{N_V}N_V \times N_V$. Each request to the service chain is deployed and the Q matrix is updated. When the algorithm is evaluated, the convergent Q matrix is used to make decisions for the deployment of the service chain to test the performance of the algorithm.

B. Comparison Algorithm

In order to compare and verify the effectiveness of the proposed algorithm, the following classical service chain mapping algorithms are selected for comparative analysis.

RL(reinforcement learning): Reinforcement learning algorithm, i.e., proposed algorithm in this paper.

EBA(Eigendecomposition-based Approach)[28]: A novel eigendecomposition-based approach for the placement of virtual and physical network function chains in networks and cloud environments. A heuristic based on a custom greedy algorithm is also presented to compare performance and assess the capability of the eigendecomposition approach.

JoraNFV(Jointly Optimize the Resource Allocation in NFV)[29]: Considering network cost and service performance, a two-stage service chain mapping algorithm based on one-hop traffic scheduling and greedy algorithm for searching multiple paths is proposed.

CCMF(Closed-Loop with Critical Mapping Feedback)[3]: A closed-loop algorithm based on feedback from key subtopological mappings is used to jointly optimize VNF combination and service chain mapping to minimize overall bandwidth consumption.

C. Evaluation Indicators

Since the placement of the service chain involves such indexes as resource utilization rate, network delay, and system energy consumption, two important evaluation indexes are selected to measure the effectiveness of the proposed algorithm for the time being: 1) The average transmission delay of each service chain; 2) Load balancing of server nodes in the system.The algorithm in this paper is suitable for optimizing different targets and only needs to change the feedback value.

Average transmission delay of the service: If the form of a service chain is $\text{VNF}_1 \rightarrow \text{VNF}_2 \rightarrow \text{VNF}_3$, the transmission delay of the service is the link transmission delay from the node where VNF_1 is map to the node where VNF_2 map to and the node where VNF_3 is map to. If two VNF are deployed on the same node, the transmission delay between the two VNF is minimal and negligible, that is, $D(v(S_{i,j}), V(S_{i,j+1})) = 0$. If two VNFS are deployed on different nodes, $D(v(S_{i,j}), V(S_{i,j+1})) = 0$ is the minimum delay of the path between the two nodes.

Load balancing of the system: During the operation of the system, the load conditions on different physical server nodes are collected at different times, and then the variance of the load conditions on different server nodes is calculated to measure the load balance of the system. The smaller the variance is, the more balanced the system load is. The greater the variance is, the greater the system load fluctuation is.

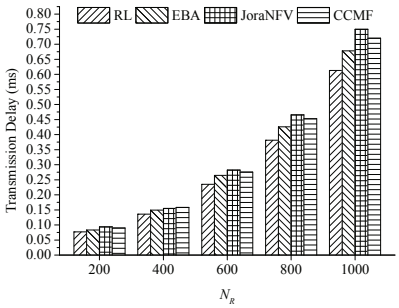

Fig. 1. Results obtained of the transmission delay when $N_V = 10$

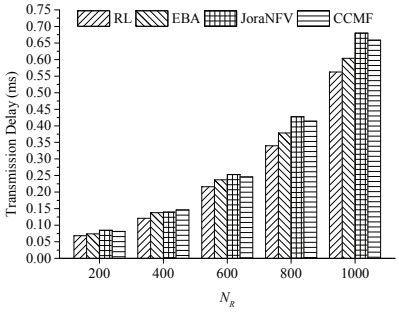

Fig. 2. Results obtained of the transmission delay when $N_V = 20$

D. Experimental Results

1) Result of the Average Transmission Delay: The service chain is deployed according to the convergent Q matrix and the algorithm performance is evaluated.In the experiment, N_R service chains were randomly generated and deployed in accordance with different algorithms. Then, the average link delay of the service chain bearing service was counted. The comparison of the average link delay index is shown in Fig.1 to Fig.3. In each figure, the number of VNF-SC ranges from 200 to 1000.

2) Result of Load Balancing: During the operation of the simulation platform, when the business arrives and leaves according to poisson flow, there will be some business chains being processed in the system. The deployment of these business chains and whether the load of the system is more balanced are also important factors for the system to be evaluated. During the operation of the system, the current load balancing status of the system is collected at certain

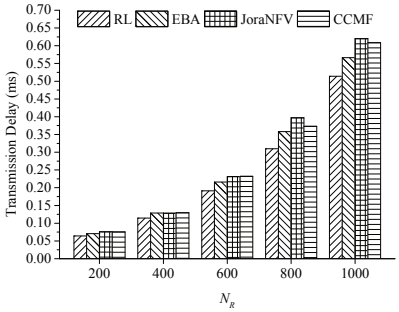

Fig. 3. Results obtained of the transmission delay when $N_V = 30$

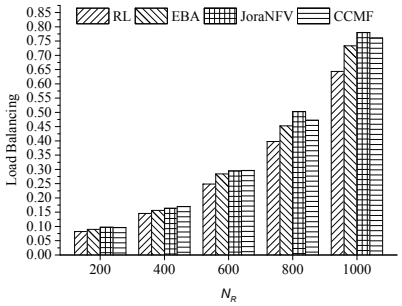

Fig. 4. Results obtained of the load balance when $N_V = 10$

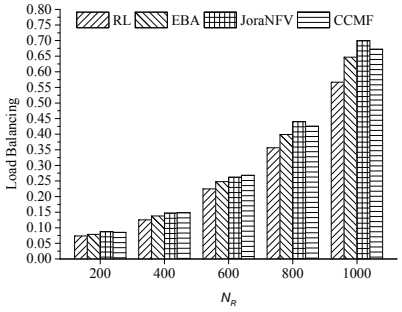

Fig. 5. Results obtained of the load balance when $N_V = 20$

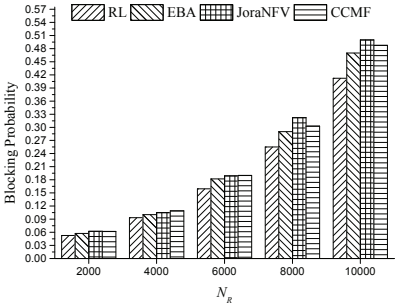

Fig. 7. Results obtained of the blocking probability when $N_V = 10$

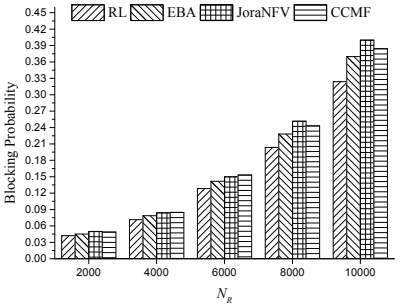

Fig. 8. Results obtained of the blocking probability when $N_V = 20$

intervals, that is, the monitoring node vCPU resource usage, and then the variance values of resource usage of different nodes are counted. A total of 30 groups of data are collected to calculate the variance, and then the average of the 30 times of variance is taken. In this paper, the mean value of variance is used to describe the equilibrium of the system, as shown in Fig.4 to Fig.6. Similar to the experiments in transmission delay, the number of VNF-SC ranges from 200 to 1000 in each figure.

3) Result of Blocking Probability: To evaluate the performance of the algorithm more comprehensively, another group experiment has been conducted. In this experiment, the the number of VNF-SC ranges from 2000 to 10000 in each figure. In this paper, the mean blocking probability is used to describe the performance of the system, as shown in Fig.7 to Fig.9.

E. Experimental Results Analysis

As shown in Fig.1 to Fig.3, RL has the lowest average service link delay, because it chooses to deploy two adjacent VNFS on the same service chain on the same node each time, or two adjacent VNFS on the two nodes with the lowest link delay, which can effectively reduce the service delay. The performance of CCMF algorithm on service average link delay is second only to RL, because this algorithm adds incentive on link delay optimization in the reward and punishment feedback of reinforcement learning. The smaller the link delay is, the greater the reward will be. Therefore, nodes with smaller delay will be selected to deploy VNF in strategy selection. JoraNFV has the worst performance, because JoraNFV selects the node with the lowest load to deploy VNF every time, leading to the possibility that VNF in the same service chain may be distributed on multiple server nodes, increasing the link delay.

As shown in Fig.4 to Fig.6, RL has small fluctuation and small load variance value of the system, indicating that the load of the system is balanced and performs best. This is because JoraNFV deploies each VNF in the service chain on the node with the smallest load every time. The performance of RL algorithm is better than JoraNFV, because the system load is considered in the algorithm reward and punishment feedback. Each time the strategy is selected, each VNF in the service chain will be deployed on the node with small load, so that the load of the system can be relatively balanced. The

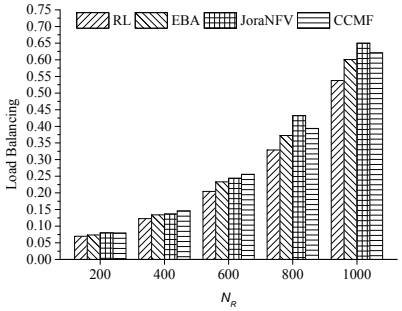

Fig. 6. Results obtained of the load balance when $N_V = 30$

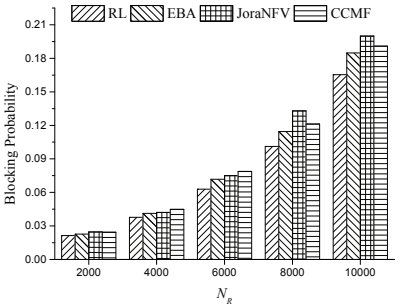

Fig. 9. Results obtained of the blocking probability when $N_V = 30$

EBA algorithm performs worst because, in order to reduce the link delay, EBA often deploys two adjacent VNFs on the same node, resulting in poor load balance performance of the system.

In this experiment, the strategy matrix is selected as the strategy representation of q-Learning algorithm. Due to the natural scalability of the strategy matrix, the algorithm cannot support large-scale network topology. Therefore, in view of the extensibility of the algorithm, this paper proposes the following improvement ideas as future research work. First, when the number of nodes is large, we can choose to initialize the strategy matrix according to experience, so that the strategy matrix is close to the optimal result, to accelerate the convergence of the strategy matrix. Secondly, deep neural network can be used to express status-action set. The industry has used deep neural network to represent status-action set in game scenes, proving that deep neural network can be applicable to complex scenes.

VI. Conclusion

Service chain mapping is the core of the NFV architecture, focusing on the dynamic provision and flexible orchestration of network service resources. In this paper, a service chain resource scheduling architecture based on reinforcement learning technology is designed, and a service chain mapping algorithm based on Q-learning reinforcement learning is proposed. Simulation results show that good optimization results are obtained in terms of average service transmission delay and server node load balance. In the following work, the service chain mapping mode driven by different business characteristics such as virtualized core network and virtual content distribution network will be studied.

References

[1] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications surveys & tutorials*, vol. 18, no. 1, pp. 236–262, 2015.

[2] Y. Sun, Y. Chen, Y. Pan, and L. Wu, "Android malware family classification based on deep learning of code images," *IAENG International Journal of Computer Science*, vol. 46, no. 4, pp. 524–533, 2019.

[3] Z. Ye, X. Cao, J. Wang, H. Yu, and C. Qiao, "Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization," *IEEE Network*, vol. 30, no. 3, pp. 81–87, 2016.

[4] H. Xuan, S. Wei, X. Zhao, Y. Zhou, X. Ma, D. Liu, and Y. Li, "Unavailable time aware scheduling of hybrid task on heterogeneous distributed system," *IAENG International Journal of Applied Mathematics*, vol. 50, no. 1, pp. 133–146, 2020.

[5] W. Fang, M. Zeng, X. Liu, W. Lu, and Z. Zhu, "Joint spectrum and IT resource allocation for efficient vNF service chaining in inter-datacenter elastic optical networks," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1539–1542, 2016.

[6] B. Yi, X. Wang, and M. Huang, "Optimised approach for VNF embedding in NFV," *IET Communications*, vol. 12, no. 20, pp. 2630–2638, 2018.

[7] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs, "Network function placement for NFV chaining in packet/optical datacenters," *Journal of Lightwave Technology*, vol. 33, no. 8, pp. 1565–1570, 2015.

[8] E. Vincenzo, M. Emanuele, and A. Mostafa, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2008–2025, 2017.

[9] G. H. Juliver and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 518–532, 2016.

[10] M. Zeng, J. Fang, and Z. Zhu, "Orchestrating tree-type VNF forwarding graphs in inter-DC elastic optical networks," *Journal of Lightwave Technology*, vol. 34, no. 14, pp. 3330–3341, 2016.

[11] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 236–262, 2016.

[12] B. Mehdi, M. P. Samir, B. Pol, H. Zhu, and P. Vincent, "Self-organization in small cell networks: A reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 12, no. 7, pp. 3202–3212, 2013.

[13] Y. Zhang, Z. Li, and L. Liu, "A global optimization algorithm for solving generalized linear fractional programming," *Engineering Letters*, vol. 28, no. 2, pp. 352–358, 2020.

[14] W. Li, G. Yin, and X. Chen, "Application of deep extreme learning machine in network intrusion detection systems," *IAENG International Journal of Computer Science*, vol. 47, no. 2, pp. 136–143, 2020.

[15] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[16] A. Marinescu, I. Dusparic, and S. Clarke, "Prediction-based multi-agent reinforcement learning in inherently non-stationary environments," *ACM Transactions on Autonomous & Adaptive Systems*, vol. 12, no. 2, pp. 1–23, 2017.

[17] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 44–55, 2018.

[18] N. Huin, A. Tomassilli, F. Giroire, and B. Jaumard, "Energy-efficient service function chain provisioning," *Journal of Optical Communications and Networking*, vol. 10, no. 3, pp. 114–124, 2018.

[19] M. Dieye, S. Ahvar, J. Sahoo, E. Ahvar, R. Glitho, H. Elbiaze, and N. Crespi, "CPVNF: Cost-efficient proactive VNF placement and chaining for value-added services in content delivery networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 774–786, 2018.

[20] X. Chen, Z. Zhu, J. Guo, S. Kang, R. Proietti, A. Castro, and S. Yoo, "Leveraging mixed-strategy gaming to realize incentive-driven VNF service chain provisioning in broker-based elastic optical inter-datacenter networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 2, pp. A232–A240, 2018.

[21] B. Li, W. Lu, S. Liu, and Z. Zhu, "Deep-learning-assisted network orchestration for on-demand and cost-effective vNF service chaining in inter-dc elastic optical networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. D29–D41, 2018.

[22] M. Mechtri, C. Ghribi, and D. Zeghlache, "A scalable algorithm for the placement of service function chains," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 533–546, 2016.

[23] H. A. Alameddine, S. Sebbah, and C. Assi, "On the interplay between network function mapping and scheduling in VNF-based networks: A column generation approach," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 860–874, 2017.

[24] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2008–2025, 2017.

[25] X. Gao, Z. Ye, J. Fan, W. Zhong, Y. Zhao, X. Cao, H. Yu, and C. Qiao, "Virtual network mapping for multicast services with max–min fairness of reliability," *Journal of Optical Communications and Networking*, vol. 7, no. 9, pp. 942–951, 2015.

[26] J. Pei, P. Hong, K. Xue, and D. Li, "Efficiently embedding service function chains with dynamic virtual network function placement in geo-distributed cloud system," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 10, pp. 2179–2192, 2018.

[27] X. Chen, Z. Zhu, R. Proietti, and S. B. Yoo, "On incentive-driven VNF service chaining in inter-datacenter elastic optical networks: A hierarchical game-theoretic mechanism," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 1–12, 2018.

[28] M. Mechtri, C. Ghribi, and D. Zeghlache, "A scalable algorithm for the placement of service function chains," *IEEE transactions on network and service management*, vol. 13, no. 3, pp. 533–546, 2016.

[29] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint optimization of service function chaining and resource allocation in network function virtualization," *IEEE Access*, vol. 4, no. 1, pp. 8084–8094, 2017.