Novel Virtual Network Function Service Chain Deployment Algorithm based on Q-learning

Hejun Xuan, Jun Lu, Na Li, and Leijie Wang

***Abstract*—As a new cloud service mode, mobile edge computing distributes the traditional centralized deployment and management of cloud resources to the wireless access network. Mobile services can be processed for the sake of obtaining a good service experience and reduce the network load of the backtrip network. In order to effectively improve the end-to-end service delay of the flow in multi-clusters coexisting mobile edge computing (MEC) network, an improved virtual network function (VNF) service chain deployment algorithm based on Q-learning was proposed. Based on the planning model, a spatial-temporal optimization model of VNF service chain deployment was established by markov decision process, and an improved BPQ-learning algorithm was designed to solve the model. The method considered both the virtual deployment of MEC service chain in space dimension and the life cycle management of VNF in time dimension. The optimization of VNF deployment is realized. The possible network congestion is avoided by deployment service nodes in advance. Experimental results show that the proposed strategy can provide end-to-end services with lower latency and better experience for delay-sensitive mobile services under different VNF service chain scale, service node scale, cluster number and logical connection relationship among the virtual network functions.**

***Index Terms*—mobile edge computing, virtual network function, service delay, Q-learning**

I. Introduction

Network virtualization (networkfunctionvirtualization, N-FV) is one of the key technologies to realize the 5g service architecture[1], [2]. NFV will all kinds of software network elements (virtualizednetworkfunction VNF) into a virtual network function, make the network get rid of the dependence on traditional hardware equipment, network equipment service updates[3], reduces the operating costs, improve the flexibility of business deployment. With the deepening of NFV application research, a series of key issues have been raised and become research hotspots [4].

In NFV-based networks, service functions are mainly carried by service function chains. SFC consists of a series of VNF instances that require a specific number of resources[5], [6]. The allocation of computing, storage, bandwidth and other resources of each VNF as well as the physical location of the deployment have an impact on end-to-end QoS performance[7]. Therefore, it is necessary to design an optimal deployment scheme to improve service performance. This kind of problem is called service function chain choreography. On the other hand, as NFV applications continue to expand, enterprises or individual users can outsource their IT service requests to operators to obtain more efficient and affordable network services. In general, carriers carry service requests from customers through data centers. Therefore, how to optimize the resource utilization of data centers and reduce the deployment cost on the premise of meeting the requirements is an urgent problem to be solved[8]. SFC layout requires to maximize deployment benefits under the premise of satisfying various constraints, and the processing rate of VNF is mostly treated as a constant in related studies. In practical application scenarios, there is a certain relationship between the amount of resources allocated by VNF and the performance[9]. Existing studies show that dynamic resource allocation technology can enhance the deployment flexibility and achieve further optimization of the choreography results.

Manuscript received April 7, 2022; revised February 20, 2023. This work was supported by National Natural Science Foundation of Henan Province (No. 232300420424), Science and Technology Department of Henan Province (No.222102210265), and by Foundation of Henan Educational Committee under Contract (No.22A520007).

Hejun Xuan is an Associate Professor at the School of Computer and Information Technology, Xinyang Normal University, Xinyang Henan, 46400, China. (email: xuanhejun0896@xynu.edu.cn)

Jun Lu is a Lecturer at the School of Computer and Information Technology, Xinyang Normal University, Xinyang Henan, 46400, China. (email: 690454390@qq.com)

Na Li is an Associate Professor at the Department of Information Engineering, Luo He Vocational Technology College, Luohe Henan, 462000, China. (email: nli_lh@126.com)

Leijie Wang is an Associate Professor at the Modern Education Technology Center, Luo He Vocational Technology College, Luohe Henan, 462000, China. (email: wljoffice@163.com)

II. Related Work

In recent years, the problem of service function chain arrangement has been widely concerned by the industry and has achieved rich results[10]. SFC choreography is a NP-Hard problem. When the number of service function chains is large or the scale of physical network is large, it is difficult to find the optimal solution in a limited time. Therefore, scholars introduce heuristic methods to realize the arrangement strategy that gives consideration to both computational speed and optimization effect. Ref.[11] modeled the SFC choreography problem in data centers as integer linear programming, considering the workload and basic resource consumption changing over time. Ref. [12] proposed NFV-RT algorithm to maximize the number of accepted requests under given end-to-end delay constraints in the data center. Qiao et al. [13] designed a heuristic algorithm combining genetic algorithm and fuzzy strategy to consider dynamic SFC choreography in data centers with the goal of minimizing end-to-end delay. Yu et al. [14] considered the SFC choreography scenario inside the data center, established an integer linear programming model with bandwidth resource optimization as the goal, and designed a two-stage algorithm to solve the problem. Thai et al. [15] proposed a two-stage algorithm to solve the SFC arrangement problem in data centers. In the first stage, greedy strategy was adopted to select VNF instances with minimum delay from the current node; in the second stage, solutions in the first stage were replaced by other available VNF instances. At the same time, the order of VNF in SFC is switched to try to search for a better arrangement scheme. In addition, some scholars have introduced reinforcement learning to solve the problem of SFC choreography, such as the service function chain mapping method based on reinforcement learning proposed by Liu et al. [15]. At present, most studies on SFC scheduling treat the processing rate of VNF as a constant. However, in practical application scenarios, the processing rate of VNF is affected by resource allocation, request arrival rate and other factors. Existing in the model by some scholars consider VNF dynamic characteristics, such as Alleg [16] combination of VNF elastic mechanism of resource allocation, to minimize the network time delay and the elastic resource allocation model is established with the target of energy consumption, The resource allocation of VNF is dynamically adjusted, the transmission performance of SFC is optimized, and the resource allocation and deployment scheme is obtained by solving integer linear programming[16]. The shortcoming of this work is that the request arrival rate, the relationship between processing delay and resource allocation are not considered comprehensively, and the algorithm based on integer programming has too much time cost. Pham et al. [17] designed a heuristic method to coordinate the orchestration of VNF and deploy it to the underlying network. Zhang et al. [18] designed two heuristic methods to solve the SFC deployment problem, one is the routing before deployment method, and the other is the node-first routing guided deployment method. Freitas et al.[19] modeled VNF deployment as a multi-objective problem that comprehensively considered deployment and energy consumption, and used two optimization algorithms, non-dominated sorting genetic algorithm and differential evolutionary algorithm, to solve the problem. In addition, the vigorous development of artificial intelligence has also attracted the attention of academia and industry, and researchers have conducted fruitful research on NFV deployment based on machine learning. For example, Soualah et al. [9] modeled the SFC choreography problem as a decision tree and designed an NFV deployment algorithm based on Monte Carlo tree search and reinforcement learning technology. Yuan Quan et al.[20] designed a virtual network functional resource demand prediction method based on multi-layer feedforward neural network by using deep learning method, and then realized the dynamic deployment of virtual network functions based on dynamic coding genetic algorithm according to the resource demand prediction results. Chen et al. [21] proposed an adaptive online scheduling method based on the perception of service quality and experience quality based on deep reinforcement learning to adapt to real-time network changes for NFV resource allocation.

In order to effectively improve the end-to-end service delay of the flow in multi-clusters coexisting mobile edge computing network, an improved virtual network function deployment method based on Q-learning algorithm was proposed. The method considered both the virtual mapping of MEC service chain in space dimension and the life cycle management of VNF in time dimension. The multi-objective optimization of VNF deployment is realized. The possible network congestion is avoided by mapping service nodes in advance.

III. problem Description and Mathematical Modeling

This article considers MEC network scenarios for cluster deployment. MEC clusters are deployed at the edge of mobile communication networks, connected to one or more eNode BS [5]. The MEC cluster is connected through a packet data network gateway (PDN-GW) and a cloud based 5G mobile core network. A MEC cluster can contain multiple virtual MEC nodes [22]. In addition, MEC clusters are interconnected and support collaboration between multiple MEC clusters. Unlike cloud based data center networks or mobile core networks, where IT resources are almost unlimited, the number of nodes and IT resources available in MEC clusters is limited. The MEC cluster first provides services for the requested services within the cluster. When resources cannot meet the demand, MEC clusters will utilize the available IT resources of other MEC clusters to build new VNF services. The resource statistics and allocation of the entire MEC cluster are implemented by the network controller located in the mobile core network. Cluster MEC deployment can span multiple eNode BS and network regions, providing end-to-end low latency services for time sensitive mobile services, which is particularly important for time sensitive applications such as smart cars and autonomous driving.

A clustered MEC network is defined as $G = G_1, G_2, ldots, G_g$, where g is the number of clusters and $G_N$is a n-th MEC cluster. Define an undirected graph $G_N = (V_n, E_n)$, where $V_N$and $E_N$represents MEC cluster G_n Edge nodes and network links within the cluster(u, w)represents the link between two edge nodes $u$and w. In addition, $u$and $w$can belong to the same or different cluster MECs$L_{u,w}$represents the available network bandwidth resources for link (u, w). The distance from edge node u to w is $D_{u,w}.N_V(vinV)$represents the number of universal service nodes (virtual machines) used to build VNFM_n is MEC cluster G_n common set of service nodes for edge node virtualization in n. The general business node $m(m \in M_n)$ in the MEC cluster Gn currently has available computing resources represented as W^n_m. The MEC network deployed in the cluster provides services for various mobile services$H$represents the MEC network receiving the $h$mobile VNF service chain within T, denoted as $H = d_1, d_2, ldots, d_h$. For Service Request $d_i(d_i \in H)$, where the entry and exit nodes are represented as I_i and E_i, respectively, from I_i to E_i The path of is represented as P_i, d_i. The data rate of d_i is expressed as R_i. According to different mobile application business requirements, multiple VNFS are logically connected in a certain order to form a serial or parallel SFC[23]. In MEC networks, adopting parallel connections can help improve the service efficiency of delay sensitive mobile services. Provide service for service request d_i of the SFC said for $S_i = \{S_{i,1}, S_{i,2}, \ldots, S_{i,K}\}$, its length is expressed as $|S_i|$, where $s_{i,j}$ represents a VNF on the SFC or the set of multiple VNF that provide services for d_i after parallel connection.

A. Service delay model

The latency generated when mobile services implement end-to-end services through cluster MEC networks includes: the queuing latency of business flow data packets waiting for processing in VNF, the latency of business flow data packets receiving and processing in VNF, and the latency generated when business flow data packets are transmitted between and within MEC clusters. Especially, the processing latency of VNF running and deploying on public business nodes may have varying degrees of impact, which must be considered when evaluating end-to-end services in clustered MEC networks. Firstly, the business flow from the entry node to the exit node needs to be processed by multiple VNFs within one SFC, and each VNF will generate processing delays when processing the business flow. Based on M/M/1/C type Jackson queuing network [24], this paper models the processing delay of traffic flow in MEC network, treats VNF in MEC network as a traffic node, and assumes that the process of traffic data grouping to VNF is a Poisson process. For VNF f_k, definition $lambda_{f,k}$and $u_{f,k}$represent their average arrival rate and average service rate, respectively. $\rho_{f,k} = \frac{\lambda_{f,k}}{u_{f,k}}$. In order to ensure the system stability of VNF service, $\rho_{f,k} < 1$. For a VNF in the MEC network, the input traffic is directly imported from the inlet node or the output of the VNF in the same or neighboring cluster. Therefore, P_{jw} is defined as the probability that the service flow is processed by the VNF_j and output to the VNF_w. λ_w^0 is defined as the traffic from the MEC inlet node to the VNF_w. The VNF_w input traffic rate can be expressed as

$$\lambda_w = \lambda_w^0 + \sum_{j=1}^{n_v} \lambda_j P_{jw} \tag{1}$$

Then the average of data groups in the cache queue of VNF f_k can be expressed as

$$M_{f,k} = \frac{\rho}{1-\rho} = \frac{\lambda_{f,k}}{u_{f,k} - \lambda_{f,k}}, i = 1, 2, \ldots, n_v \tag{2}$$

VNF f_k can be deployed on node W, so the service rate of f_k imported to physical node w can be expressed as $u_{f_k} = R(w)$, where $R(w)$ is determined by the transmission capacity of service flows. The processing delay of VNF f_k can be obtained $T_{f,k} = \frac{M_{f,k}}{\lambda_{f,k}}$. Since $\frac{M_{f,k}}{\lambda_{f,k}} = \frac{\frac{\lambda_{f,k}}{u_{f,k}-\lambda_{f,k}}}{\sum_{i\in H} \lambda_i Z^i_{f,k}}$, thus,

$$T_{f,k} = \frac{\frac{\lambda_w}{R(W)-\lambda_w}}{\sum\limits_{i\in H} R_i Z^i_{f,k} \varepsilon^{n,w}_{f,k}} \tag{3}$$

where λ_w is the rate of the w service flow. $Z^i_{f,k} = 1$ indicates that d_i needs f_k to provide services. $Z^i_{f,k} = 0$ indicates that d_i needs f_k to provide services. Then, the processing delay of mobile service request d_i can be expressed as

$$t_1^i = \sum_{i\in H} \sum_{f\in F} \sum_{k=1}^{|N_f|} Z^i_{f,k} T_{f,k} \tag{4}$$

Based on the modeling and analysis of M/M/1/C queuing networks, the average queuing delay of a single VNF can be expressed as

$$t_{qu}^{f,k} = \sum_{j=1}^{c} \frac{j q_j}{\mu} \tag{5}$$

where q_j represents the smoothing probability when j users arrive, and the queuing delay of business requests can be expressed as

$$t_2^i = \sum_{i\in H} \sum_{f\in F} \sum_{k=1}^{|N_f|} Z^i_{f,k} t_{qu}^{f,k} \tag{6}$$

Similarly, a matrix A composed of binary decision variables is defined. Any element in the matrix is $\varepsilon^{n,u}_{f,k}$ and $\varepsilon^{n,u}_{f,k} \in \{0,1\}$, $\{\varepsilon^{n,u}_{f,k} | f \in F, 0 < k \le |N_f|\}$. $\varepsilon^{n,u}_{f,k} = 0$ means that business flow f_k is processed by the universal service node u in MEC cluster n, and $\varepsilon^{n,u}_{f,k} = 0$ means that business flow f_k does not flow through the universal service node u in MEC cluster n. In addition, binary decision variables $\sigma^{u,w}_{i,f_k,f'_{k'}}$, $\sigma^{u,w}_{i,f_k,f'_{k'}} = 1$ denotes that $\beta^i_{f_k,f'_{k'}}$ mapped to the underlying network link $l_{u,w}$, $\sigma^{u,w}_{i,f_k,f'_{k'}} = 0$ denotes that $\beta^i_{f_k,f'_{k'}}$ not mapped to the underlying network link $l_{u,w}$. The propagation delay of a business request can be expressed as

$$t_3^i = \frac{\sum\limits_{i\in H} \sum\limits_{(u,w)\in E} D_{u,w} \sigma^{u,w}_{i,f_k,f'_{k'}}}{p} \tag{7}$$

where p represents the transmission rate of signals on physical links, $D_{u,w}$ represents the length of physical links, and E represents the overall set of network links in the entire MEC cluster.

In the case of cluster MECs receiving multiple service requests, this paper optimizes end-to-end service latency for a cluster MEC network with various resource constraints by deploying VNF and selecting the service flow path. The optimization model can be expressed as

$$\min \sum_{j=1}^{3} t_j^i \tag{8}$$

To reach to this objectives, some constraints should be satisfied:

(1) The link bandwidth resource limits between two common service nodes in the same MEC cluster, that is

$$\sum_{i\in H} \sum_{f\in F} \sum_{f'\in F} \sum_{k=1}^{|N_f|} \sum_{k'=1}^{N_{f'}} \beta^i_{f_k,f'_{k'}} \sigma^{u,w}_{i,f_k,f'_{k'}} \le l_{u,w} \forall (u,w) \in E \tag{9}$$

(2)Bandwidth resource limits of links between MEC clusters

$$\sum_{i\in H} \sum_{f\in F} \sum_{f'\in F} \sum_{k=1}^{|N_f|} \sum_{k'=1}^{N_{f'}} R_i \sigma^{u,w}_{i,f_k,f'_{k'}} \le l_{u,w}, \forall m \ne n \tag{10}$$

(3) The computing resources occupied by one or more VNFS deployed on node u in MEC cluster n cannot exceed the total computing resources.

$$\sum_{i\in H} \sum_{f\in F} \sum_{k=6}^{|N_f|} \omega^i_{f_k} \varepsilon^{n,u}_{f_k} \le W^n_u, \forall u \in V, 1 \le n \le g \tag{11}$$

(4) In the network, all active nodes except inlet nodes and egress nodes must satisfy traffic conservation.

$$\sum_{f\in F} \sum_{k=1}^{|N_f|} \sum_{f'\in F} \sum_{k'=1}^{|N_{f'}|} (\sigma^{p,u}_{i,f_k,f'_{k'}} - \sigma^{u,w}_{i,f_k,f'_{k'}}) = 0, \forall i \in H, p, w \tag{12}$$

(5) A service request enters the MEC network from only one entry node and leaves the clustered MEC network from only one exit node.

$$\sum_{f\in F} \sum_{k=1}^{|N_f|} \sum_{f'\in F} \sum_{k'=1}^{|N_{f'}|} (\sigma^{u,E_i}_{i,f_k,f'_{k'}} - \sigma^{E_i,u}_{i,f_k,f'_{k'}}) = 1, \forall u \ne E_i \tag{13}$$

$$\sum_{f\in F}\sum_{k=1}^{|N_f|}\sum_{f'\in F}\sum_{k'=1}^{|N_{f'}|}(\sigma_{i,f_k,f'_{k'}}^{I_i,u}-\sigma_{i,f_k,f'_{k'}}^{u,I_i})=1,\forall i\in H,u\neq I_i \tag{14}$$

(6) Each SFC service flow must pass through all its predefined VNFs. Delay constraints on business requests. Viable link and node mappings are required.

$$\varepsilon_{f_k}^{n,u}\varepsilon_{f_k}^{n',w}=\varepsilon_{f_k}^{n,w},1\leq n,n'\leq g,u,w\in V \tag{15}$$

$$\sum_{j=1}^{3}t_j^i\leq t^i,\forall 1\leq i\leq h \tag{16}$$

$$\sigma_{i,f_k,f'_{k'}}^{u,w}\in\{0,1\},\forall i\in H,(u,w)\in E \tag{17}$$

$$\varepsilon_{f_k}^{n,u}\in\{0,1\},1\leq n\leq g,\forall u\in V,f_k\in F \tag{18}$$

IV. Proposed Algorithm

A. VNF deployment based on Q-learning

The value iteration or strategy based on dynamic programming iterative algorithm is used for Markov decision process with known parameters. However, in the actual network environment, considering the delay parameters of dynamic and state-space scales, the traditional dynamic programming method cannot accurately solve the model in a short time. Reinforcement learning method becomes an effective means to solve the problem. Reinforcement learning combines the theory of dynamic programming with the mechanism of learning psychology to solve the sequential optimization decision problem with delay. Q learning algorithm model is an adaptive closed-loop control system. At the same time, the feedback loop will report the revenue function $r(s_t,a_t)$to the agent at time $t$and update the behavior value function $Q(s_t,a_t)$accordingly. The agent will update the behavior value function estimate table $Q(s,a)$according to $Q(s_t,a_t)$. Repeat this for the state s_{t+1}, and the loop continues until $Q(s,a)$reaches the optimal behavior value $Q^*(s_t,a_t)$. The agent selects the optimal strategy $Q^*(s_t,a_t)$according to the discount benefit and J of each strategy in π_{Q^*}.

$$\begin{aligned}Q(s_t,a_t)=&Q(s_t,a_t)+\alpha_t[r(s_t,a_t)+\\&\max Q(s_{t+1},a_{t+1})-Q(s_t,a_t)]\end{aligned} \tag{19}$$

$$Q^*(s_t,a_t)=E[r(s_t,a_t)+\max Q^*(s_{t+1},a_{t+1})] \tag{20}$$

The convergence of Q-learning algorithm has been proved, and the updating rule of behavior value function is shown in Eq.(19), where (s_t,a_t) is the state-behavior pair of markov decision process at time t, s_{t+1} is the state at time $t+1$, $r(s_t,a_t)$ represents the deployment revenue received by agent at time t, and $0<\alpha_t<1$ is the learning factor. The optimal behavior value function satisfies Bellman's optimal value equation as shown in Eq.(20), where $Q^*(s_t,a_t)$ represents the optimal behavior value at time t.

In the iteration process, assume a part of the network view at time t, where the internal number of the node represents the number of the underlying node, $d_{ij}(t)$represents the link $(i,j),i,j\in[1,n]$ at time t, and the number below the node represents the service processing delay after VNF is deployed to the node. The solid line with an arrow indicates the current policy π_Q. In this case, the service chain deployment method based on the q learning algorithm performs actions at the time $t$in the deployment state s_t, calculates the instantaneous value of the q learning benefit function $r(s_t,a_t)$according to the network deployment benefit and total delay cost, and updates the estimated value of $Q(s_t,a_t)$ dollars accordingly. The network view is updated at $t+1$, and the process is repeated until $Q*(s_t,a_t)$satisfies Bellman's equation shown in equation (20) and outputs the optimal policy. This method uses the idea of dynamic programming to solve the network delay in time dimension, and uses the sampling value of unit time delay to approximate the instantaneous delay, which improves the precision of delay optimization.

B. Behavior value function approximation based on BP neural network

In the real environment, due to the complexity of network topology, Q learning algorithm is usually "dimension disaster" problem dimension disaster "refers to when a markov decision process has great state space space $aS$and behavior, the size is $|S|*|A|$, Q learning algorithm estimates the value of the function of individual learning cycle table $Q(S,a)$ for (S,A). With the increase of learning cycle, resulting in the learning process cannot be completed. In order to solve the "dimensional disaster" problem, this section introduces BP neural network to realize the approximation of the behavior value function $Q(s,a)$.

The structure of BP neural network consists of input layer, hidden layer and output layer, each layer contains several neurons. The former and the latter layers are connected by weights. The learning process of BP network consists of two parts: forward transmission and reverse transmission. In the forward process, the state of each layer of neurons is only affected by the structure of one layer of neurons. If there is an error between the actual output and the expected output of the output layer, the reverse output process of the network can be adjusted by the gradient descent method to gradually approach the minimum output error. During the learning process, the two propagation processes are repeated until the error reaches the set range and the output function estimate is reached. At present, BP neural network, as a supervisory function estimation method, has been proved to be able to approximate the objective function of real number with arbitrary precision. According to the proposed reinforcement learning algorithm based on neural network and the proposed Q learning algorithm based on linear value function approximation, this section improves the Q learning system and designs the BP-Q learning system. At time t, agent performs action a_t a on the time-space optimization model of VNF deployment in state s_t s to obtain immediate revenue $r(s_t,a_t)$. The system input the current state of MDP to the neural network module-action pair (s_t,a_t). and immediate revenue $r(s_t,a_t)$. The neural network module approximates the behavior value function according to the input (s_t,a_t) and immediate revenue $r(s_t,a_t)$. Output the estimated value of the behavior value function $Q_{NN}(s_t,a_t)$ to the agent, the agent uses the estimated value to perform the behavior value function iteration, and transmits the learning result $Q(s_t, a_t)$ back to the neural network module to achieve weight vector adjustment.

According to the above procedure, the updating rule of the behavior value function in equation 19 can be modified to the form shown in equation 21. The improved algorithm is called BPQ learning algorithm. The improved system are not stored and updated size is $|S| * |A|$. A behavior value function estimation table $Q(S, A)$, and only need to store the weights of each neuron in the neural network. The storage size is only related to the topology of the neural network designed. Based on the improved BPQ learning system, the corresponding VNF automatic deployment algorithm is shown in algorithm1.

$$\begin{aligned} Q(s_t, a_t) =& Q_{NN}(s_t, a_t) + \alpha_t[r(s_t, a_t) + \\ & \max Q_{NN}(s_{t+1}, a_{t+1}) - Q_{NN}(s_t, a_t)] \end{aligned} \tag{21}$$

V. Experiments and Analysis

A. Experimental Setting

The underlying network topology is randomly generated by the GT-ITM tool, which contains 50 nodes and approximately 130 links. The computing power of the bottom node and bandwidth of the bottom link are evenly distributed [50, 100], and the cost per unit of computing resource c_S and bandwidth resource b_S is 1. The number of service types supported by the underlying network is set to 10. Each underlying node randomly provides one to five service types. The processing time of a service instance depends on the type of service and the processing power of the network node. The transmission delay of the underlying link is proportional to the Euclidean distance between the two endpoints of the link, which is set according to the principle above and ranges from 1 to 10 time units.

The arrival process of service requests follows the Poisson distribution. On average, four requests arrive within 100 time units, and the duration of each service request follows the exponential distribution of 1000 time units on average. The service chain consists of four services with random and non-repetitive types. The computing power required by each service is evenly distributed in [1,50], and the bandwidth required by the logical link is evenly distributed in [1,50]. The charge for each unit of computing power and unit of bandwidth is c_R and b_R. The maximum allowed end-to-end delay $D_{max}$is set to 100 time units. The time of each simulation experiment is about 50,000 time units, and the data is recorded every 4,000 time units starting from 2000 time units. Each group carried out 10 simulation experiments and averaged the experimental results. The population size is set to 100, the upper limit of iteration is set to 10000,

B. Experimental Results

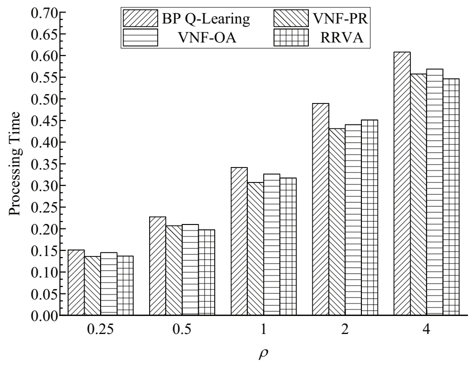
For the comprehensive evaluation model of the feasibility and effectiveness of the algorithm, this paper uses the processing time, acceptance rate, revenue and total latency of the underlying network of multiple VNF service chains as the calculation of performance evaluation indicators. Gao et al. [25] formally defines the VNF layout and Routing (VNF-PR) problem by presenting a generic linear programming formula

Algorithm 1: VNF deployment algorithm based on BP-Q learning

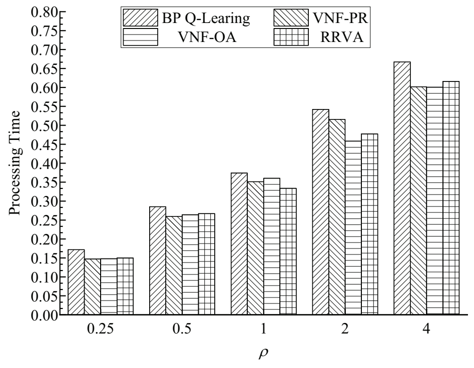
```
Input: service chain $l$; network $G_S$; parameters of the VNF
Output: Delay-revenue optimization strategy of service chain $l$
1  The state space $S$, behavior space $A$ and transition probability matrix $P$ of Markov decision process are initialized.
2  According to the request information of service chain $l$, the sequence constraint $\varphi_l$ of function chain $l$ is obtained by combining VNF.
3  Select strategy $\pi_Q$ according to sequence constraint $\varphi_l$ initialization;
4  Initialize q-learning learning factor $\alpha_0$, allowable error threshold $\varepsilon$, maximum learning period $M_T$;
5  The neural network weight vector $w$ and sample set $D_S$ were initialized, and the maximum learning step was $M_S$.
6  Establish an optimization model. According to the optimization model, the immediate revenue function is initialized $r(s_t, a_t)$;
7  Set $t = 0$, randomly initialize the initial state $s_0$;
8  while $Q_{NN}(s_t, a_t) - Q^*_{NN}(s_t, a_t) < \varepsilon$ or $t < M_T$ do
9      for $\forall a_t$ in $s_t$ do
10         $s_{tp} = 0$, and $Q_{NN}(s_t, a_t)$ is calculated;
11         Add sample $s_t, a_t, Q_{NN}(s_t, a_t)$ to the dataset $D_S$;
12         Perform the action at to get instant revenue $r(s_t, a_t)$, enter state $s_{t+1}$, $s_{tp} = s_{tp} + 1$;
13         if $rand() < 1 - \varepsilon$ then
14             $a^* = \arg maxQ_{NN}(s_{t+1}, a_{t+1})$;
15             $\delta = r(s_t, a_t) + Q_{NN}(s_{t+1}, a_{t+1}) - Q_{NN}(s_t, a_t)$;
16         end
17         Select the current action $a_t$ randomly according to $\pi_Q$;
18         $Q(s_t, a_t) = Q_{NN}(s_t, a_t) + \alpha_t\delta$, update dataset $D_S$;
19         if The storage space of the dataset $D_S$ is full. then
20             BP neural network starts training and updates weight vector $W$;
21         end
22         $t = t + 1$, update the learning factor $a_t$;
23     end
24 end
25 Calculate the benefits of service chain deployment scheme $P$ according to $Q^*(s, a)$;
```

that ADAPTS to the specific characteristics and constraints of NFV infrastructure and differs significantly from existing virtual network embedding formulas in prior art. We also design a mathematical heuristic that can be extended to multiple targets and large instances, expressed as VNF-PR. In order to reduce queue length by allowing a single virtual machine to install VNF based on local knowledge (offloading), achieve stable redeployment of VNF, adapt to network topology and time and space changes of service, an algorithm called VNF-OA is proposed in [26]. Yang et al. [27] investigated the problem of how to place VNF on edge and public clouds and route traffic between adjacent VNF pairs so that maximum link load ratio is minimized and delay per user is met. This problem exists for both fully ordered SFC and partially ordered SFCs. An efficient random rounding approximation algorithm to solve this problem is expressed as RRVA.

The number of data center nodes are fixed as $N_D = N_V/4$ and $N_D = 3N_V/4$. In each experiment, number of VNF-SCs are set as $N_R = \rho N_V(N_V - 1)$, and $\rho = 0.25$, 0.5, 1, 2 and 4, respectively. Fig.1 to Fig.2 show the processing time at different number of VNF service chain in NSFNET and ARPANET when $N_D = N_V/4$ and $N_D = 3N_V/4$.



(a) Obtained in NSFNET.



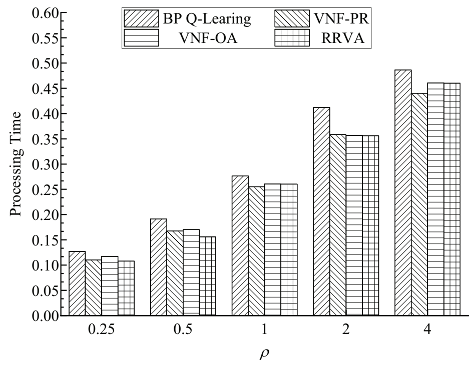
(b) Obtained in ARPANET.

Fig. 1. Processing time at different number of VNF service chain when $N_D = N_V/4$.

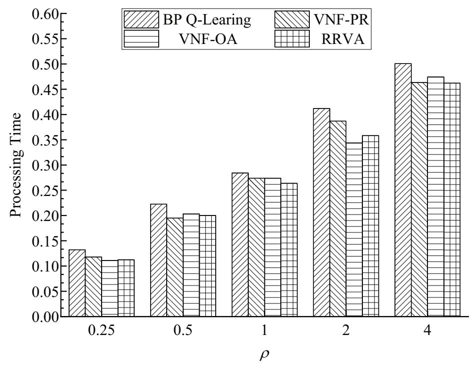
The ratio of VNF service chain acceptance obtained in NSFNET and ARPANET when $N_D = N_V/4$ and $N_D = 3N_V/4$ are shown in Figure 3 to Figure 4, respectively.

Similarly, Figure 5 and Figure 6 show the average revenue under different number of VNF service chain in NSFNET and ARPANET when $N_D = N_V/4$ and $N_D = 3N_V/4$.

Similarly, Figure 7 and Figure 8 show the average revenue under different number of VNF service chain in NSFNET

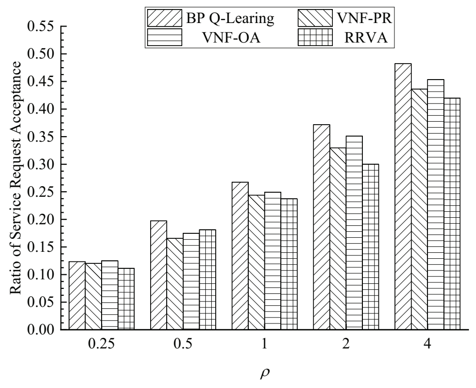


(a) Obtained in NSFNET.

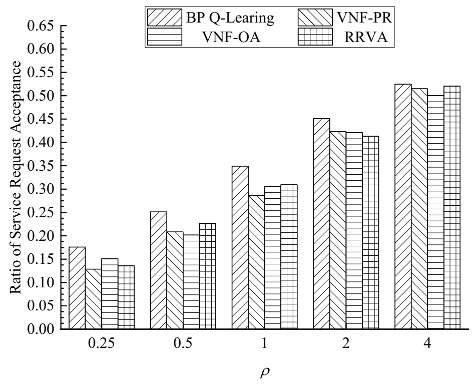


(b) Obtained in ARPANET.

Fig. 2. Processing time at different number of VNF service chain when $N_D = 3N_V/4$.

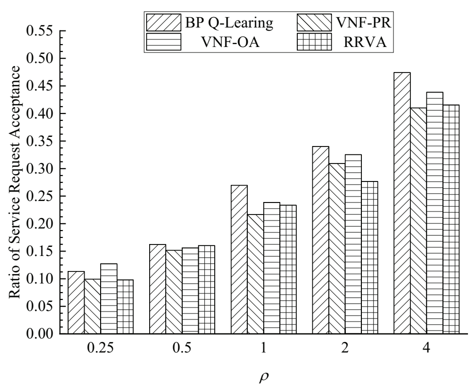


(a) Obtained in NSFNET.

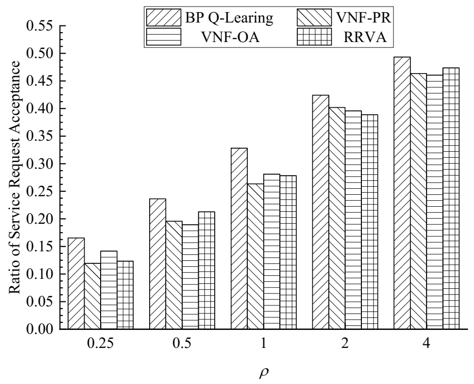


(b) Obtained in ARPANET.

Fig. 3. Ratio of VNF service chain acceptance at different number of VNF service chain when $N_D = N_V/4$.

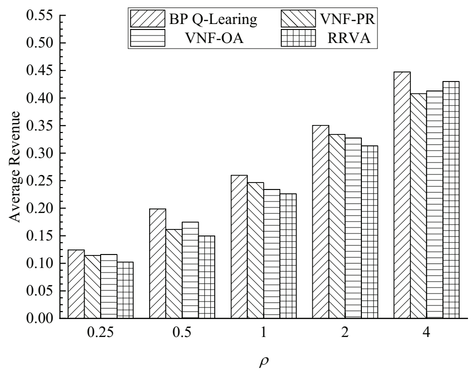


(a) Obtained in NSFNET.

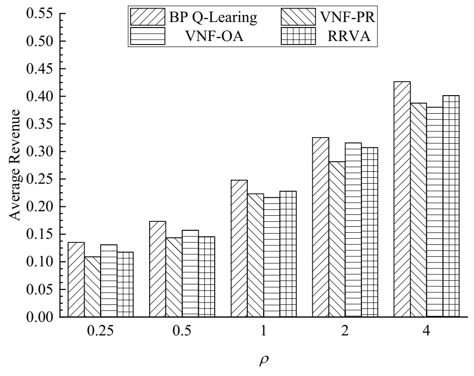


(b) Obtained in ARPANET.

Fig. 4. Ratio of VNF service chain acceptance at different number of VNF service chain when $N_D = 3N_V/4$.

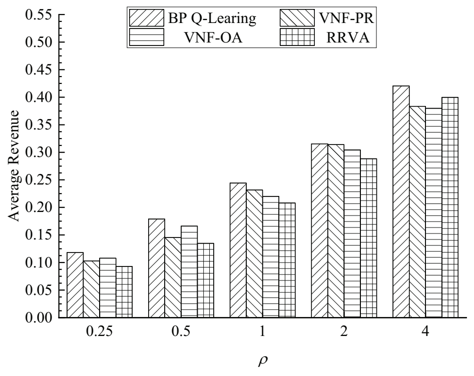


(a) Obtained in NSFNET.

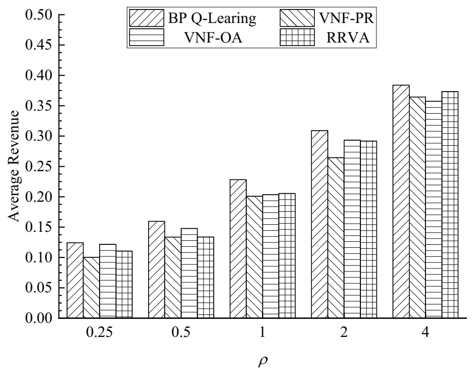


(b) Obtained in ARPANET.

Fig. 5. Average revenue under different number of VNF service chain when $N_D = N_V/4$.



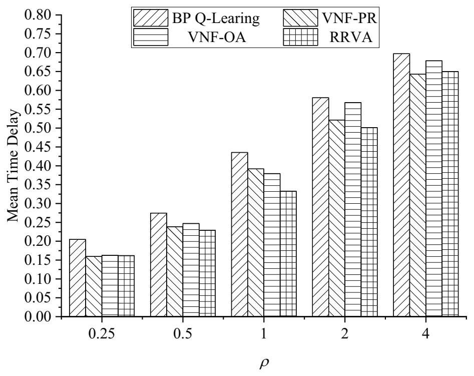
(a) Obtained in NSFNET.



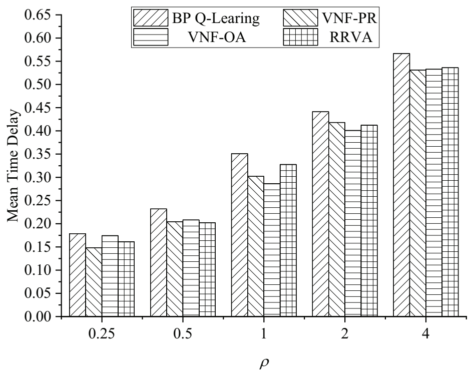
(b) Obtained in ARPANET.

Fig. 6. Average revenue under different number of VNF service chain when $N_D = 3N_V/4$.

and ARPANET when $N_D = N_V/4$ and $N_D = 3N_V/4$.

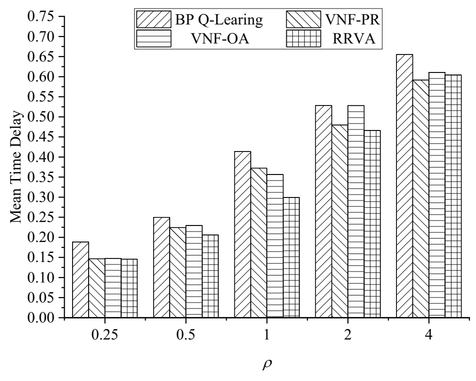


(a) Obtained in NSFNET.

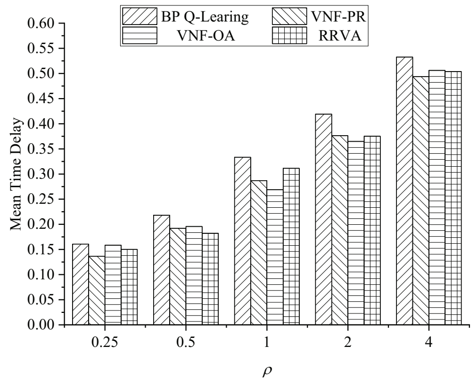


(b) Obtained in ARPANET.

Fig. 7. Mean time delay under different number of VNF service chain when $N_D = N_V/4$.



(a) Obtained in NSFNET.



(b) Obtained in ARPANET.

Fig. 8. Mean time delay under different number of VNF service chain when $N_D = 3N_V/4$.

C. Experimental results analysis

First, the VNF service chain processing time of each deployment method is analyzed. As shown in Figure 1 to Figure 2, the second algorithm takes the longest processing time because there are only two phase mapping algorithms. In this algorithm, stage 1 selects the selected node of stage 2 through the algorithm to constrain the minimum link of time delay between nodes of the VNF node deployment scheme. The VNF that completes deployment first must wait in the cache queue for the next VNF in the service chain to complete deployment before establishing a link with it. Since the first stage will traverse a large number of invalid solutions that do not satisfy the constraint, the VNF will wait in the queue for a long time, which reduces the efficiency of the algorithm. The method based on the third algorithm uses recursion to search for the shortest delay path without direction. Using the idea of dynamic programming, the service chain deployment problem is divided into multiple sub-chain deployment problems, and then each sub-chain is combined in turn. Based on the improved Q learning algorithm, the behavior value $Q(s,a)$is updated to continuously search the deployment strategy in the direction of discount benefit and maximization, avoiding blind traversal of the entire state space. However, in the initial learning stage, $Q(s,a)$contains less knowledge, and the efficiency of neural network training is low. The convergence rate of $Q(s,a)$is affected. Deployment methods based on quantum algorithms are developing towards higher fitness. Under the same VNF service chain, this method has the shortest VNF service chain processing time.

Figure 3 to Figure 4 shows the VNF service link acceptance rate for different number of VNF service chain strengths. The processing time of VNF service chain directly determines the resource usage cycle of the same number of VNF service links and affects the utilization of underlying network resources. There is a clear downward trend as the VNF service chain increases. This is because the approach does not take into account the impact of the underlying network node and link resource state on the deployment strategy. As a result, low-latency links are usually assigned a higher deployment priority, which results in frequent reuse of some links, local overloads of the network, and lower VNF service chain rates. Q-learning method realizes fine-grained management of VNF life cycle, and regards the creation and removal of each VNF as an independent random process within the VNF service chain time, which improves the utilization of physical resources in the time dimension. Therefore, Q-Learning method has the highest VNF service chain acceptance rate.

Figure 5 and Figure 6 show the long-term average revenue for each algorithm with different number of VNF service chain strengths. Q-Learning algorithm implements fine - grained management of VNF life cycle. Under the same VNF service chain strength, the Q-Learning algorithm based VNF deployment method can avoid idle VNF occupying physical resources, reduce the communication cost per unit time, and effectively improve the infrastructure benefit of the underlying network.

Figure 5 and Figure 6 show the long-term average revenue for each algorithm with different number of VNF service chain strengths. Q-Learning algorithm implements fine - grained management of VNF life cycle. Under the same VNF service chain strength, the Q-Learning algorithm based VNF deployment method can avoid idle VNF occupying physical resources, reduce the communication cost per unit time, and effectively improve the infrastructure benefit of the underlying network.

VI. CONCLUSIONS

This paper mainly studies the deployment of service chain VNF in virtualization environment. Aiming at the deficiency of traditional virtual network mapping algorithm and the high delay requirement of 5G service, a VNF deployment method based on Q learning algorithm is proposed, and its effectiveness is verified. In order to further improve the service quality of mobile services under the background of virtualization, VNF deployment under the condition of reliability will be studied to meet the carrier-level reliability requirements of mobile services.

REFERENCES

[1] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. D. Foy, and Y. Zhang, "Mobile edge cloud system: Architectures, challenges, and approaches," *IEEE Systems Journal*, pp. 1–14, 2017.

[2] A. A. Esswie and K. I. Pedersen, "Opportunistic spatial preemptive scheduling for urllc and embb coexistence in multi-user 5G networks," *IEEE Access*, vol. 6, pp. 38 451–38 463, 2018.

[3] B. Chatras and F. F. Ozog, "Network functions virtualization: The portability challenge," *IEEE Network the Magazine of Computer Communications*, vol. 30, no. 4, pp. 4–8, 2016.

[4] L. Lei, X. Xiong, H. Lu, and Z. Kan, "Collaborative edge caching through service function chaining: Architecture and challenges," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 94–102, 2018.

[5] H. Huang and S. Guo, “Proactive failure recovery for nfv in distributed edge computing,” *IEEE Communications Magazine*, vol. 57, no. 5, pp. 131–137, 2019.

[6] Y. Nam, S. Song, and J. M. Chung, “Clustered nfv service chaining optimization in mobile-edge clouds,” *IEEE Communications Letters*, vol. 21, no. 2, pp. 350–353, 2016.

[7] H. J. Xuan, S. W. Wei, X. L. Zhao, Y. Zhou, X. P. Ma, D. H. Liu, and Y. L. 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.

[8] S. Y. Li, K. B. Yuan, Y. Zhang, and W. Sun, “A novel bandwidth allocation scheme for elastic and inelastic services in peer-to-peer networks,” *IAENG International Journal of Computer Science*, vol. 46, no. 2, pp. 163–169, 2019.

[9] P. Mach and Z. Becvar, “Mobile edge computing: A survey on architecture and computation offloading,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[10] M. R. Sama, L. M. Contreras, J. Kaippallimalil, I. Akiyoshi, and N. Hui, “Software-defined control of the virtualized mobile packet core,” *IEEE Communications Magazine*, vol. 53, no. 2, pp. 107–115, 2015.

[11] T. Taleb, M. Corici, C. Parada, A. Jamakovic, S. Ruffino, G. Karagiannis, and T. Magedanz, “EASE: EPC as a service to ease mobile core network deployment over cloud,” *IEEE Network*, vol. 29, no. 2, pp. 78–88, 2015.

[12] S. Song, C. Lee, H. Cho, G. Lim, and J. M. Chung, “Clustered virtualized network functions resource allocation based on context-aware grouping in 5G edge networks,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 5, pp. 1072–1083, 2019.

[13] M. Peuster, S. Schneider, M. Zhao, G. Xilouris, and H. Karl, “Introducing automated verification and validation for virtualized network functions and services,” *IEEE Communications Magazine*, vol. 57, no. 5, pp. 96–102, 2019.

[14] F. Alvarez, D. Breitgand, D. Griffin, P. Andriani, S. Rizou, N. Zioulis, F. Moscatelli, J. Serrano, M. Keltsch, and P. Trakadas, “An edge-to-cloud virtualized multimedia service platform for 5G networks,” *IEEE Transactions on Broadcasting*, vol. 25, no. 2, pp. 369–380, 2019.

[15] A. D. Costa, L. Bondan, J. A. Wickboldt, C. B. Both, and L. Z. Granville, “Orchestra: A customizable split-aware nfvorchestrator for dynamic c-ran,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1014–1024, 2020.

[16] L. Y. Xu, A. Y. Deng, M. Z. Ge, J. Shi, and G. Y. Gao, “A MADM-based handover management in software-defined 5G network,” *Engineering Letters*, vol. 27, no. 4, pp. 842–849, 2019.

[17] Y. Wan, Y. Qu, L. Gao, and Y. Xiang, “Privacy-preserving blockchain-enabled federated learning for B5G-driven edge computing,” *Computer Networks*, vol. 204, p. 108671, 2022.

[18] K. Wang, S. P. Xu, C. M. Chen, S. H. Islam, and G. Aloi, “A trusted consensus scheme for collaborative learning in the edge AI computing domain,” *IEEE Network*, vol. 35, no. 1, pp. 204–210, 2021.

[19] H. J. Xuan, X. L. Zhao, J. W. Fan, Y. H. Xue, F. F. Zhu, and Y. L. Li, “VNF service chain deployment algorithm in 5G communication based on reinforcement learning,” *IAENG Internaitonal Journal of Computer Science*, vol. 48, no. 1, pp. 1–7, 2021.

[20] H. T. Zhang, G. F. Wu, and W. S. Bu, “Research of agent based control model for embedded networked system,” *Engineering Letters*, vol. 27, no. 2, pp. 278–286, 2019.

[21] H. J. Xuan, S. W. Wei, X. L. Zhao, Y. H. Xue, L. L. Qiao, and Y. L. Li, “tp-MA: Orchestrating three populations memetic algorithm for VNF deployment in 5G network,” *IAENG Internaitonal Journal of Computer Science*, vol. 47, no. 4, pp. 730–739, 2020.

[22] L. Magoula, S. Barmpounakis, S. I., and A. N., “A genetic algorithm approach for service function chain placement in 5G and beyond, virtualized edge networks,” *Computer Networks*, vol. 195, no. 9, p. 108157, 2021.

[23] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, “NFV and SDN - key technology enablers for 5G networks,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2468–2478, 2018.

[24] C. Song, M. Zhang, Y. Y. Zhan, D. S. Wang, and L. Y. Guan, “Hierarchical edge cloud enabling network slicing for 5G optical fronthaul,” *Journal of Optical Communications and Networking*, vol. 11, no. 4, pp. B60–B70, 2019.

[25] M. Gao, B. Addis, M. Bouet, and S. Secci, “Optimal orchestration of virtual network functions,” *Computer Networks*, vol. 142, no. 4, pp. 108–127, 2018.

[26] X. J. Chen, W. Ni, B. Iain, C. Xin, and W. S. G., “Automated function placement and online optimization of network functions virtualization,” *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1225–1237, 2018.

[27] S. Yang, F. Li, S. Trajanovski, X. Chen, and X. Fu, “Delay-aware virtual network function placement and routing in edge clouds,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 445–459, 2020.