Journal of Network Intelligence ©2023 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **8**, Number **1**, February **2023**

Dynamic Deployment Algorithm for Virtual Network Function Service Chain Based on DQN

Na Li*

Department of Information Engineering
Luo He Vocational Technology College
123 College Road, Henan Luohe, 462000, China
nli_lh@126.com

Leijie Wang

Modern Education Technology Center
Luo He Vocational Technology College
123 College Road, Henan Luohe, 462000, China
wljoffice@163.com

Yu Yan

Soft Engineering
Henan Normal University
46 East of Jianshe Road, Henan Xinxiang, 453000, China
yu1138960333@126.com

Jiading Wang

Department of Policy and Planning Sciences
University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki
wang.jiading.xg@alumni.tsukuba.ac.jp

*Corresponding author: Na Li

Received November 15, 2022, revised December 21, 2022, accepted January 27, 2023.

Abstract. *Network Function Virtualization (NFV) can provide the network resource according to the demands, and can improve the flexibility of the physical network. It has become the key technology of 5G/6G communication. However, virtual network function service chain (VNF service chain) deployment is the crucial problem of the NFV. To tackle the VNF service chain deployment problem, a multi-objective optimization model is proposed. In this model, deployment overhead and transmission delay are minimized, and load balance degree is maximized. For the sake of solving this model, a dynamic virtual network function deployment algorithm based on Deep Q network (DQN) is proposed. This method can effectively perception that the network resource state change dynamically. In addition, DQN learns the feature of VNF service chain deployment problem and determines the optimal deployment schemes for incoming VNF service chains. Finally, some simulation experiments are conducted to confirm the advantages of the proposed algorithm. The experimental results show that the proposed algorithm can effectively reduce deployment cost, transmission time delay, and increase the load balance degree than the compared algorithms.*

Keywords: Network Function Virtualization, Multi-objective, Transmission delay, Load balance

---

1. **Introduction.** In 5G network, network function virtualization is used to deploy the core network [1]. Network functional virtualization is divided into three stages: group chain, deployment and scheduling [2]. Among them, group chain is the construction process of service function chain, which solves the logical relationship between virtual network functions. Different group chain modes will affect the performance of network services [3]. The deployment process refers to the search for a physical link on the underlying physical network. The physical nodes in the link can bear the corresponding VNF function types and meet the demand for computing resources. The physical link should meet the demand for bandwidth resources [4]. Scheduling refers to the optimization decision of network services on VNF service chain mapping sequence [5]. A reasonable mapping sequence can reduce delay overhead and improve node resource utilization and bandwidth resource utilization [6]. Therefore, how to group, deploy and optimize the scheduling of service functions is a hot issue.

At present, a large number of studies have been carried out on the deployment mechanism of VNF. Among them, literature [7] ensures the quality of user service while reducing the cost of operators, but it adopts a static VNF deployment strategy. As the network environment is dynamically changing, long-term optimization needs to be considered. In order to minimize the end-to-end delay of VNF service chain [8], the end-to-end delay of VNF service chain can be reduced by reducing the transmission delay and processing delay of VNF service chain. However, the utilization rate of physical network resources is not paid attention to. In [9], when considering the server resource capacity and flow rate, VNF operation cost, VNF instance maintenance cost and VNF deployment cost are balanced, but the VNF service chain delay is not considered, thus ignoring the user quality of service. In [10, 11], the deep reinforcement learning network is used in the scenarios of access network, cross-domain mapping and core network respectively, and the mapping algorithm of service function chain is proposed. Literature [12] proposed a delay-and reliability-oriented service function chain deployment method, which solved the problem of service function chain construction through virtual network function aggregation and carried out mapping by evaluating the importance of physical nodes. Since the construction and mapping of service function chain are considered step by step, the obtained solution is suboptimal. Literature [13] proposes a resource-aware algorithm for collaborative construction and mapping of service function chains, which takes into account the underlying network state, but does not take into account the mapping sequence of service requests in the slice at the same time, which will increase the delay overhead. In [14], an optimized mapping strategy that prioritizes mapping service function chains with fewer requirements to reduce delay overhead and improve resource utilization is proposed. When determining the mapping priority, this method only considers the requirements of the service function chain construction stage, not the actual mapping stage, so it is easy to fall into local optimization.

Therefore, there are two main problems in the existing literature: on the one hand, considering service functional chain group and mapping step by step will lead to the reduction of solution space; On the other hand, calculating the mapping weight of service function chain based on the scheme of group chain stage rather than the scheme of mapping stage will lead to deviation of mapping order. In this paper, a multi-objective optimization model and DQN based dynamic deployment algorithm for VNF service chain is proposed. Firstly, a multi-objective optimization model, which minimizes deployment cost, transmission delay and maximizes the load balance, is established. Then, an improved algorithm based DQN is proposed to solve the model established.

2. **Related Works.** In recent years, more and more researches have been published [15, 16, 17]. Not only considers IDC-EON routing and VNF deployment issues, but also considers the number and location of data centers in the network [18]. Based on these issues, a bi-level planning model is proposed, and the BiHMA algorithm was designed to determine the optimal number and location of data centers as well as routing and VNF deployment. Literature [19] proposed the merged multicast service Chain model for optimizing the traditional multicast service chain. Based on this model, a heuristic CPT algorithm is designed to determine the location deployment of VNFs and routes on VNF service chains. A mixed strategy game method based on auxiliary graphs, which is convenient for users to choose a more suitable VNF service chains scheme, then proposes an operator game model, and develops an efficient heuristic algorithm to motivate operators compete with each other to provide service solutions [20]. In [21], the author abstracts the VNF service chains deployment problem as a grey system theory problem, proposed a configurable and flexible network service method, and demonstrates the feasibility of the method through an example. Literature [22] formulated the deployment problem of multicast VNF service chain as an integer linear programming, and designs a heuristic TSDV algorithm, which is divided into two stages: determining the location of VNFs in multicast VNF service chains and multicast routing. A hierarchical auxiliary graph based method for the deployment of VNF service chains on heterogeneous NFV platforms (virtual machines, docker containers, etc.) the total cost, while ensuring the user service quality requirements [23]. Based on deep learning models, an efficient algorithm was designed to predict future VNF service chains [24]. When the predicted VNF service chains comes, the service operator only needs to guide the data flow through the pre-deployed VNFs in turn, realizing the possibility of instant service. In [25], corresponding VNF service chains configuration strategy is proposed according to the delay of different links and the processing capacity of DC nodes. Based on the development of fixed and mobile convergence network, it maximizes the integration of existing VNF service chains in current VNF service chains. There are VNFs. Reference [26] proposed the global balancing factor and the local balancing factor, and designed a joint optimization selection algorithm to select the appropriate choice for user requests of VNFs. Reference [27] proposed a fine-grained scheduling scheme for VNF deployment, formulates the VNF adaptive resource allocation problem as a convex optimization problem through an integer linear programming method, and designs a new adaptive scaling joint optimization algorithm to improve VNF service chain. deployment efficiency. In [28], in order to avoid the unbalanced load of the network system, a higher service chain was established. In order to meet the flexible business needs of users, a NFV-based VNF service chain integration architecture and VNF service chain selection mechanism are proposed. Replicated VNF to solve the problem of balancing network load and reducing resource cost in VNF deployment was investigated [29]. The so-called replicated VNF is to segment the data stream in a standardized way, and the segmented data stream is directed to the server by one or more ordered VNFs. , and finally solved the problem effectively through the linear programming model. The VNFs deployment problem of NFV network mapping VNF service chain for the purpose of improving network performance and profit was studied [30]. It also considered the delay requirement of VNF service chain, and then described the problem as an integer in order to minimize resource consumption linear programming model, VNF service chain mapping and VNF deployment algorithm are proposed to map VNF service chain and optimize VNF deployment.

3. **Network and Problem Description.**

3.1. **Network Scenario.** In this paper, reasonable deployment of NFV in 5G core network can reduce deployment cost and improve resource utilization. NFV choreographer and control architecture. The management choreographer of the system is divided into three parts, in which the choreographer is responsible for VNF service chain, the VNF manager is responsible for VNF connections, and the infrastructure manager is responsible for global resource management [31].

3.2. **Problem Description.**

3.2.1. *Physical Network.* The physical network is represented by an undirected graph $G = (V, E)$, where V is a set of physical nodes and E is a set of physical links. Each physical node can carry a variety of specific types of VNF. For any physical node $v \in V$, C_a denotes the maximum number of virtual network function can carried in the node, $e_{m,n}$ is the physical link between v_m and v_n, and $B_{m,n}$ is the bandwidth resources of the physical link between v_m and v_n.

3.2.2. *VNF service chain.* A service request can consist of source node v_s, destination node v_d, initial traffic rate init, service work type set energy F, and service request delay threshold T. λ is the ratio of the input traffic to the output traffic of the service function, and μ is the number of processing units required to process 1Mb/s traffic. Literature [32] describes the dependency relationship between service functions and defines dependency matrix. In the process of service function grouping, each service function F can serve as the target of the next group of chain only if the current group chain scheme contains all the service functions that the service function depends on. Thus, a VNF service chains is a directed graph constructed based on the dependencies between service functions. Because the three processes of group chain, mapping and scheduling of service function chain are interrelated, if the cooperative group chain and mapping of VNF service chain cannot be carried out, the results will fall into local optimal. If scheduling is considered step by step with group chain and mapping scheme, the mapping weight will be biased. Therefore, the three processes should be considered together when deploying the service functional chain.

4. **Formula Description.**

4.1. **Service function chain mapping priority.** For VNF service chains with known construction and mapping schemes, the mapping priority calculation formula is defined as shown follows:

$$P = \frac{1}{\gamma_1 ct_V + \gamma_2 ct_B + \gamma_3 J} \tag{1}$$

where ct_V represents the resource cost, ct_Brepresents the bandwidth cost, J denotes the total number of hops deployed, and γ_1, γ_2 and γ_3 are three weight coefficients. In addition, $0 \le \gamma_1, \gamma_2, \gamma_3 \le 1$, and $\gamma_1 + \gamma_2 + \gamma_3 = 1$.

4.2. **Optimization Objectives.** (1) Deployment overhead: The first optimization objective is to minimize the deployment overhead of service requests by coordinating the grouping, mapping, and scheduling of service functional chains. The objective function is

$$ct = \alpha ct_V + \beta ct_B \tag{2}$$

where, α and β are normalized weight factors, and we have $0 \le \alpha, \beta \le 1$, and $\alpha + \beta = 1$. ct_V is calculated as shown in Eq.(3), ct_B is calculated as shown in Eq.(4),

$$ct_V = \sum_{v \in V} \sum_{i=1}^{F} p_{f,i} u_f (\tau_{init} \prod_{k=0}^{i-1} p_{g,k} \lambda_g) q_{f,v} h_{f,v} \tag{3}$$

$$ct_B = \sum_{v\in V}\sum_{i=0}^{F}(t_{init}\prod_{k=0}^{i-1}p_{g,k}\lambda_g)p_{f,i}l_{f,g,v,w} \tag{4}$$

To normalize the first optimization objective, two new variables are defined:

$$ct1_V = \sum_{v\in V}\sum_{i=1}^{F}p_{f,i}u_f(\tau_{init}\prod_{k=0}^{i-1}p_{g,k})q_{f,v}h_{f,v} \tag{5}$$

$$ct1_B = \sum_{v\in V}\sum_{i=0}^{F}(t_{init}\prod_{k=0}^{i-1}p_{g,k})p_{f,i}l_{f,g,v,w} \tag{6}$$

Thus,

$$ct1 = \alpha ct1_V + \beta ct1_B \tag{7}$$

Obviously, we have $ct < ct1$, thus, $0 \le f_1 = ct/ct1 \le 1$. The second objective is defined as:

$$\min f_1 = \min\left\{\frac{D(SP)}{D'(SP)}\right\} \tag{8}$$

(2) Transmission delay: The time it takes for the data flow to travel from the source node to the destination node is the end-to-end delay of the service path, including the execution delay of the service instance on the service path and the transmission delay of the communication link, defined as follows:

$$\begin{aligned} D(SP) = &\sum_{n^R\in N^R}\sum_{n^S\in M_S(S(n^R))} d_{S(n^R)} \\ &-\sum_{l^r\in L^R}\sum_{P^S\in M_L(l^R), l^s\in P^S} d(l^S) \end{aligned} \tag{9}$$

Another variable is defined:

$$D^{'}(SP) = \sum_{n^R\in N^R}\sum_{n^S} d_{S(n^R)} - \sum_{l^r\in L^R}\sum_{l^s\in P^S} d(l^S) \tag{10}$$

Obviously, we have $D(SP) < D^{'}(SP)$, thus, $0 \le f_2 = D(SP)/D'(SP) \le 1$. The second objective is defined as:

$$\min f_2 = \min\left\{\frac{D(SP)}{D'(SP)}\right\} \tag{11}$$

(3) Load balance degree: When building a business path, not only consider the functional requirements of the business, but also consider the load of the business bearing node and its adjacent links, and map the business and logical links to the underlying nodes and links with rich resources as much as possible. on the way. The load degree LD (Load degree) is used to measure the resource usage and load of the service path. The load intensity of the lowest node $n^R_{i_k}$ under the service path is defined as follows:

$$LD_{n^R_{i_k}} = \sum_{S(n^R)an^R_{i_k}\in M_{i_k}} \frac{\mu(S(n^R))}{C(n^R_{i_k})}, \forall n^R_{i_k} \in N^S P \tag{12}$$

The load intensity of the node comprehensively considers the available computing power of the node and the computing power demand of the business. It can be known from formula (3) that its value range is 0 1. The smaller the value, the more likely the service will be mapped to that node, and the better the load balancing of the underlying nodes.

Based on the idea of node load intensity, the load intensity of links in the service path is defined as follows:

$$LD_{l^R_{i_k}} = \sum_{P^S \in M_L(l^R), l^R_{i_k} \in P^S,} \frac{\mu(l^R)}{B(l^R_{i_k})}, \forall l^R_{i_k} \in L^S P \tag{13}$$

The load intensity of the service path SP is defined as:

$$LD(SP) = w_N \sum_{n^S_{i_k} \in N^{SP}} LD_{n^S_{i_k}} + w_L \sum_{l^S_{i_k} \in L^{SP}} LD_{l^S_{i_k}} \tag{14}$$

where w_N and w_N are two weight parameters used to adjust the load intensity of nodes and links, and $0 \le w_N, w_N \le 1$ and $w_N + w_N = 1$. Similarly, another variable is defined:

$$LD^{'}(SP) = \sum_{n^S_{i_k} \in N^{SP}} LD_{n^S_{i_k}} + \sum_{l^S_{i_k} \in L^{SP}} LD_{l^S_{i_k}} \tag{15}$$

Obviously, we have $LD(SP) < LD^{'}(SP)$, thus, $0 \le f_3 = LD(SP)/LD^{'}(SP) \le 1$. The third objective is defined as:

$$\min f_3 = \min \left\{ 1 - \frac{LD(SP)}{LD'(SP)} \right\} \tag{16}$$

4.3. **Constraints.** (1) A service functions in the service request can be deployed only on the same physical node, that is

$$\sum_{v \in V} q_{f,v} = 1, \forall f \in F \tag{17}$$

where $q_{f,v} = 1$ denotes the service functions in the service request f deployed on the physical node v, otherwise, $q_{f,v} = 0$.

(2) The outgoing traffic from node V can flow out of only one physical link,

$$\sum_{w \in V} l_{f,g,v,w} = 1, \forall f, g \in F, v \in V \tag{18}$$

(3) Only one network function can be available at any location in the service chain.

$$\sum_{v \in V} p_{f,i} = 1, \forall i \in \{1, 2, \ldots, F\} \tag{19}$$

Where, $p_{f,i}$ indicates that service function f is located at the i bit of the service function chain.

(4) There is only one network function of each type in the service chain.

$$\sum_{i=1}^{F} p_{f,i} = 1, \forall f \in F \tag{20}$$

(5) If service g depends on service f, traffic must pass through service f first.

$$i < j, p_{f,i} = 1 \wedge p_{g,j} = 1 \wedge D_f(g) = 1 \tag{21}$$

(6) The existing computing resources of the node are greater than the data traffic to be carried.

$$\sum_{i=1}^{F} p_{f,i} u_f q_{f,v} f_f \tau_{init} \prod_{k=0}^{i-1} (\sum_{g \in F} p_{g,k} \lambda_g) \le C_v, \forall v \in V, f, g \in F \tag{22}$$

(7) The remaining bandwidth of the link is greater than the required bandwidth.

$$\sum_{i=1}^{F} p_{f,g,v,w}\tau_{init}\prod_{k=0}^{i}(\sum_{g\in F} p_{x,k}\lambda_g) \le B_{v,w},\\ \forall v,w\in V, f,g,x\in F, p_w^k=1 \tag{23}$$

(8) Routing constraint: indicates that in the routing path from f to g, the number of outgoing links of intermediate nodes is equal to the number of incoming links except for the endpoint of the path.

$$\sum_{w\in V} l_{f,g,v,w} - \sum_{w\in V} l_{f,g,w,v} = \begin{cases} 1, & q_{f,v}=1\wedge q_{g,v}\neq 1\\ -1, & q_{f,v}\neq 1\wedge q_{g,v}=1\\ 0, & other\end{cases} \tag{24}$$

According to the objectives and the constraints, the three objectives optimization model can be established as follows:

$$\begin{cases} min\{f_1,f_2,f_3\}\\ s.t.\\ \sum_{v\in V} q_{f,v}=1,\forall f\in F\\ \sum_{w\in V} l_{f,g,v,w}=1,\forall f,g\in F, v\in V\\ \sum_{v\in V} p_{f,i}=1,\forall i\in\{1,2,\dots,F\}\\ \sum_{i=1}^{F} p_{f,i}=1,\forall f\in F\\ i<j, p_{f,i}=1\wedge p_{g,j}=1\wedge D_f(g)=1\\ \sum_{i=1}^{F} p_{f,i}u_f q_{f,v} f_f \tau_{init}\prod_{k=0}^{i-1}(\sum_{g\in F} p_{g,k}\lambda_g)\le C_v\\ \sum_{i=1}^{F} p_{f,g,v,w}\tau_{init}\prod_{k=0}^{i}(\sum_{g\in F} p_{x,k}\lambda_g)\le B_{v,w}\\ \sum_{w\in V}(l_{f,g,v,w}-l_{f,g,w,v}) = \begin{cases} 1, & q_{f,v}=1\wedge q_{g,v}\neq 1\\ -1, & q_{f,v}\neq 1\wedge q_{g,v}=1\\ 0, & other\end{cases}\end{cases} \tag{25}$$

5. **Proposed Algorithm.** In this section, we first introduces the service chain deployment architecture based on DQN, then introduces the online service chain deployment algorithm based on DQN, and finally describes the training process of DQN in detail. Markov decision process can continuously and automatically describe the change of network environment and the state transfer of network resources. Based on the above conditions, we need to find a suitable and efficient service chain deployment algorithm, which can automatically take appropriate actions in each state to obtain better returns. The infrastructure network environment is an NFV/SDN enabled network, which includes servers, SDN switches, and physical links. The management choreography agent based on DQN can obtain the status information from the underlying network environment and automatically select an action as the return value. After the action is performed, the NFV/SDN enabled network gives a reward feedback. Finally, the agent updates the relevant strategy according to the reward feedback and moves to the next state. Repeat the process until the reward converges. In particular, DQN based on multi-layer fully connected deep neural network can efficiently process complex network state input, and constantly optimize the output action selection through back propagation training to generate the optimal service chain deployment strategy $\pi^* = \pi|S\times A$

5.1. **VNF Service chain deployment algorithm based on DQN.** In order to deal with the dynamic changes of the network effectively, this paper uses Poisson process to represent the arrival and departure of service requests. When a service request arrives, the NFV/SDN network decides whether to accept the service request and updates the network status. System deploys the service chain of the service request in turn according to the arrival time. If the service chain deployment fails, the service request is rejected and the initial network status is returned. Possible causes of the service chain deployment failure are as follows: (1) VNFs deployment fails due to server resource shortage. (2) The bandwidth or delay constraints of service requests cannot be met. In order to reduce the value space of VNF deployment in the process of service chain deployment, this paper adopts a serialized mode for service chain deployment, that is, only one VNF is deployed in each MDP state transition. First, the network state is initialized, the deployment step is determined according to the length of the input service chain, the underlying network resource state is checked, the available server node set is generated as the action space, and an action at is selected as the return value according to the setting conditions. NFV/SDN enables the network to give reward feedback to the action at performed in the current network state. The agent updates the strategy according to the reward and moves to the next state. If the set of available server nodes is empty during deployment, System returns a message notifying the resource that the condition cannot be met, denies the service request, and returns the initialization status. The experience reuse pool is used for the training of neural network. In this paper, the size of the reuse pool is set as $P = 500$, and the training starts when the storage samples of the experience pool is full. If the network resources meet the deployment requirements of the entire service chain, check the end-to-end delay value of the service chain deployment policy in the input status. If the value meets the delay constraint of the service request, the policy is output to provide the service. Otherwise, system returns a message denial of service request that cannot meet the constraints.

5.2. **Online Training.** DQN includes two neural networks. The one is online training network, and the other is target network. In general, the online training network updates its parameters constantly, which is used to train the neural network parameters and calculate the estimated value. For the target network, it freezes the network parameters and updates the parameters at several intervals. Then, calculating realistic values according to the parameters updated. The structure of the target network is the same as that of the online training network. The different of this two network is that the parameters of the target network are updated after every L step. The estimated value remains stable for a period of time, which can reduce the correlation between the current estimated value and the target estimated value to a certain extent. It improves the stability of the training algorithm. In the process of network training, the loss function is generally optimized by differential evolution, and the loss function is described as:

$$L(\theta) = (r + \gamma max_a Q(s', a'|\theta^-) - Q(s, a|\theta))^2 \tag{26}$$

6. **Experiments and Analysis.**

6.1. **Experimental Setting.** The available resource capacity of the data center is randomly selected within the interval [5, 50](unit). The link bandwidth capacity is randomly selected with the following values: 100 Mbps, 150 Mbps,600 Mbps and 1 Gbps. The transmission delay is randomly generated within [500, 1000](ms). A service chain contains 5 9 VNFs, the number of resources requested by VNF follows the uniform distribution on [0.5, 2], the bandwidth of V_L request is randomly selected within 1 40 Mbps, and the maximum tolerated delay of service request is set at [5, 20] (ms). Pytorch 1.6 machine

Algorithm 1: VNF service chain deployment algorithm based on DQN

Input: VNF service chain; communication network; parameters of the VNF
Output: Optimal strategy of VNF service chain

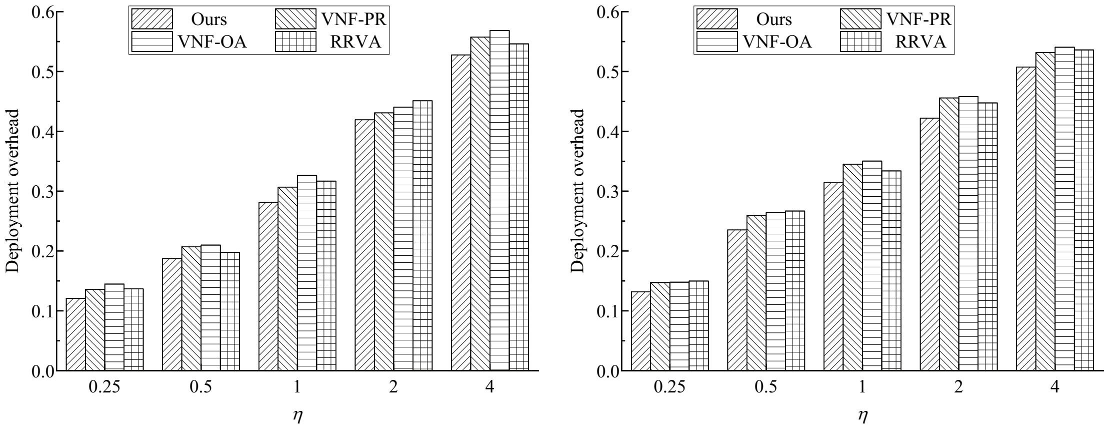
Initialize the capacity M of experience multiplexing pool D.
Initialize the Q value corresponding to the action randomly.
Initializes the selection policy π;
for *episode in range* **do**
  Initialize the state s;
  for *step in range* **do**
    Check the status of underlying network resources and generate a set Φ of server nodes that meet the conditions;
    if *random* $\leq \epsilon$ **then**
      Select an action a_t randomly;
    else
      Select $a_t = \max_a Q^*(s_t, a_t; \theta)$;
    end
    Perform the action in the service chain deployment simulator and observe the corresponding reward r_t and the next state s_{t+1};
    Store sample $e_t = (s_t, a_t, r_t, s_{t+1})$ in the experience reuse pool;
    Small batch samples (s_j, a_j, r_j, s_{j+1}) are randomly selected from the experience reuse pool; Let $y_j = \begin{cases} r_j, & r_j = end \\ r_j + \gamma \max_{a'} Q(s_{j+1}, a'; \theta), & r_j \neq end \end{cases}$
    Differential evolution is used to optimize function $(y_i - Q(s_{j+1}, a; \theta))^2$;
    Update the target network parameters θ^- at each L step;
  end
end
Calculate the revenue of VNF service chain deployment scheme π^* according to $Q^*(s_t, a; \theta)$;

learning library based on Python 3.8 was used to perform deep learning, and Network was used to simulate the underlying network of data center infrastructure. The following parameters were used to set the management arrangement agent, discount factor, Adma was used for parameter learning of the neural network, and the update cycle of the target network was adopted. The hidden layer of the neural network adopted a two-layer fully connected structure, the number of neurons was 100, and the Rectified Linear Unit (ReLU) was used as the activation function.

6.2. **Experimental Results.** To confirm the advantages and effectiveness of the proposed algorithm, proposed algorithm with three algorithms are compared, denoted as: VNF-PR [33], VNF-OA [34] and RRVA [35]. In experiments, number of VNF service chain are fixed as $N_R = \eta N_v(N_v - 1)$, and $\eta = 0.25$, 0.5, 1, 2 and 4, respectively. The deployment overhead of VNF service chain in NSFNET and CHNNET network topologies when $N_D = 0.25N_V$ and $N_D = 0.75N_V$ are shown in Figure 1 to Figure 2.

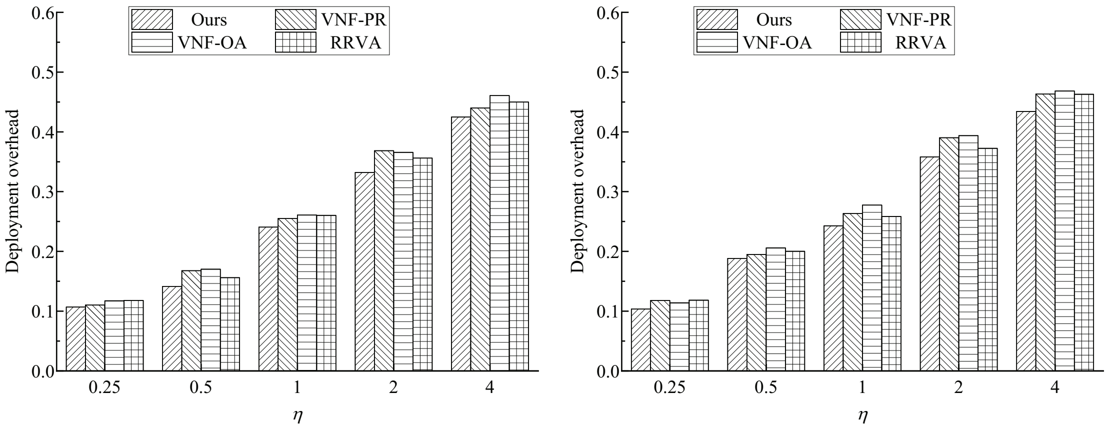
Figure 3 to Figure 4 show the transmission time delay obtained in NSFNET and CHNNET network topologies when $N_D = 0.25N_V$ and $N_D = 0.75N_V$, respectively.

Similarly, The load balance degree of VNF service chain in NSFNET and CHNNET network topologies obtained when $N_D = 0.25N_V$ and $N_D = 0.75N_V$ are shown in Figure 5 and Figure 6.



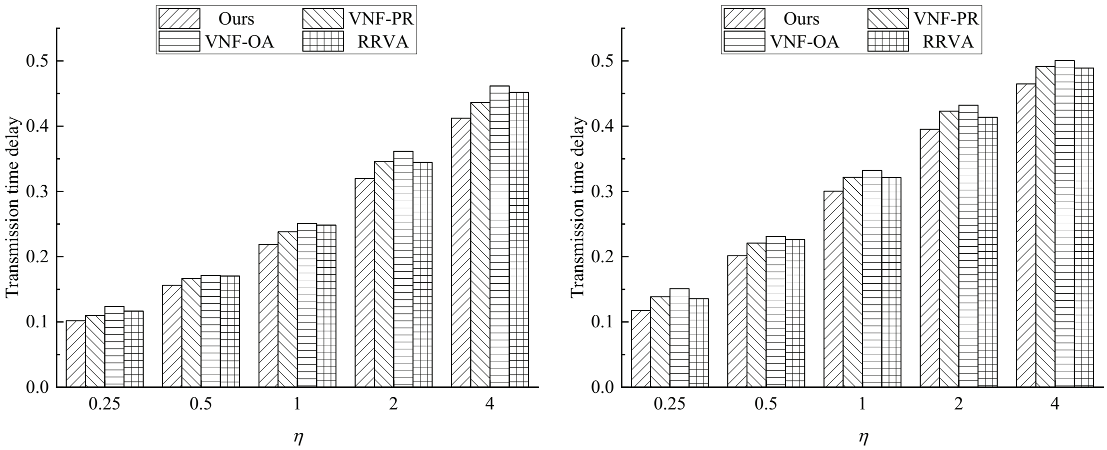
(a) Obtained in NSFNET. (b) Obtained in CHNNET.

FIGURE 1. Deployment overhead obtained when $N_D = 0.25N_V$.



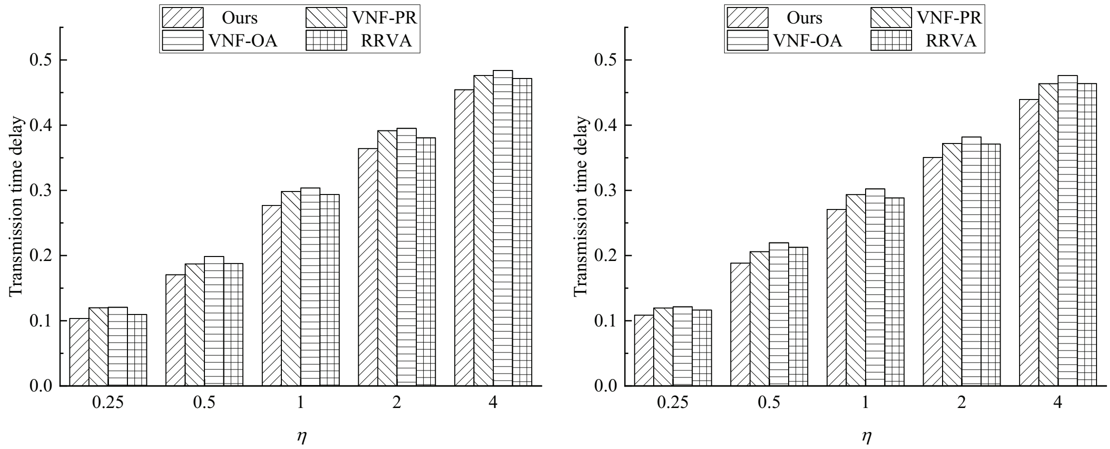
(a) Obtained in NSFNET. (b) Obtained in CHNNET.

FIGURE 2. Deployment overhead obtained when $N_D = 0.75N_V$.



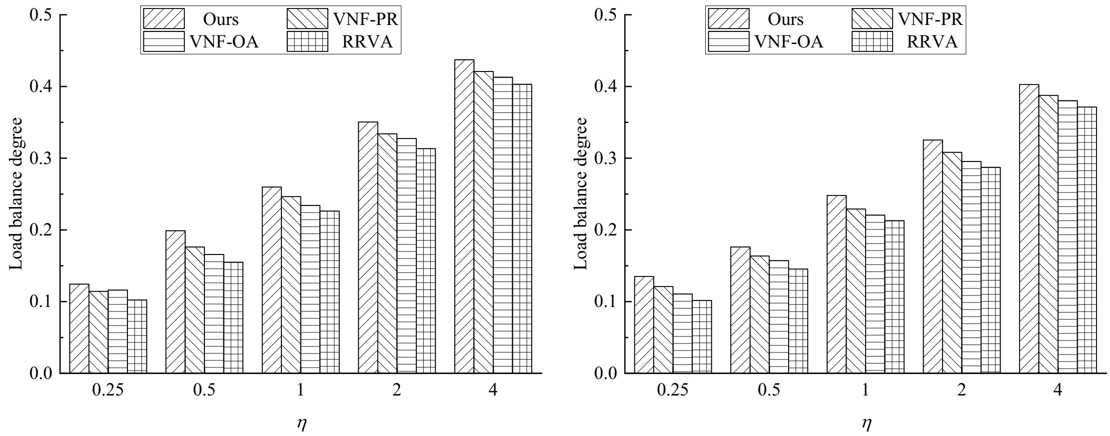
(a) Obtained in NSFNET. (b) Obtained in CHNNET.

FIGURE 3. Transmission time delay obtained when $N_D = 0.25N_V$.



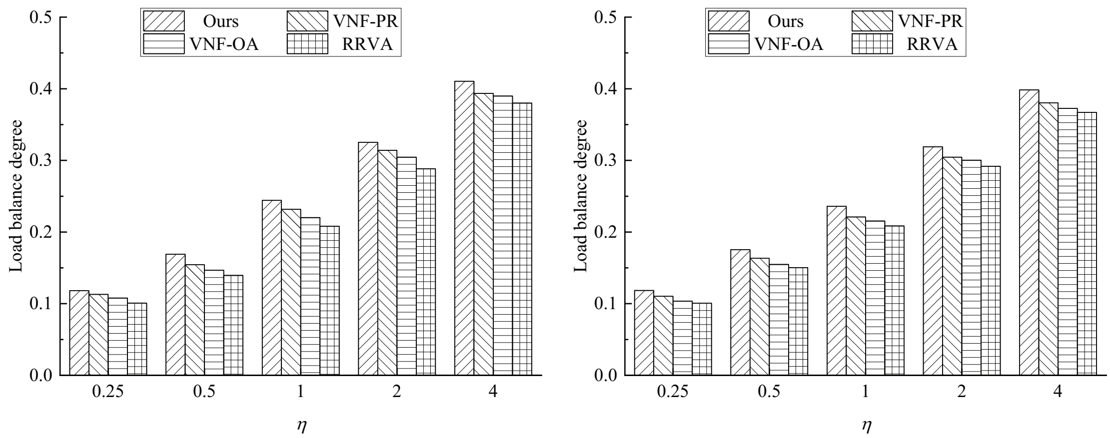
(a) Obtained in NSFNET. (b) Obtained in CHNNET.

FIGURE 4. Transmission time delay obtained when $N_D = 0.75 N_V$.



(a) Obtained in NSFNET. (b) Obtained in CHNNET.

FIGURE 5. Load balance degree obtained when $N_D = 0.25 N_V$.



(a) Obtained in NSFNET. (b) Obtained in CHNNET.

FIGURE 6. Load balance degree obtained when $N_D = 0.75 N_V$.

6.3. **Experimental results analysis.** Figure 3 to Figure 4 shows the experimental results of the deployment overhead in two different network topologies. From the results, we can see that proposed algorithm can obtain the lowest deployment overhead among the four algorithms. With the number of VNF service chain increase, the deployment overhead by proposed algorithm is more smaller than that obtained by compared algorithms. Proposed algorithm in the process of VNF service chain deployment serialization can better capture the dynamic change of network resource state, so it has advantages in the deployment overhead.

Figure 3 to Figure 4 show the experimental results of the transmission time delay in two different network topologies. Proposed algorithm updates the real-time link delay information in a serialized way during the deployment process, and updates the deployment strategy according to the input information, which makes the use of the whole network link more efficient. The results show that the proposed algorithm can effectively reduce the transmission time delay of the VNF service chain.

Figure 5 and Figure 6 show the experimental results of the load balance degree in two different network topologies. The Markov decision model adopted by proposed algorithm has better adaptability to the change of network state and therefore better deployment strategy. So, the proposed algorithm can obtain the highest load balance degree among the four algorithms.

7. **Conclusions.** This paper investigate the dynamic VNF service chain deployment problem. A multi-objective optimization model, which minimize deployment overhead and transmission delay, and maximizes load balance degree, is proposed. To tackle this model, an dynamic VNF service chain deployment algorithm based on DQN was proposed. Some simulation experiments are conducted to demonstrate the performance of the proposed algorithm. The experimental results show that the proposed algorithm can effectively reduce deployment cost and transmission delay, and increase the load balance than the compared algorithm.

Acknowledgment. This work was supported by Science and Technology Department of Henan Province (No.222102210265), and by Foundation of Henan Educational Committee under Contract (No.22A520007)

REFERENCES

[1] T.-Y. Wu, X.-L. Guo, Y.-C. Chen, S. Kumari, and C.-M. Chen, "Amassing the Security: An Enhanced Authentication Protocol for Drone Communications over 5G Networks," *Drones*, vol. 6, no. 1, pp. 10, 2022.

[2] C. Bu, X. Wang, H. Cheng, M. Huang, K. Li, and S.-K. Das, "Service Function Chain Embedding for NFV-Enabled IoT Based on Deep Reinforcement Learning," *IEEE Communications Magazine*, vol. 57, no. 11, pp. 102–108, 2019.

[3] M. Jalalitabar, E. Guler, D.-Y. Zheng, G.C. Luo, L. Tian, and X.-J. Cao, "Embedding Dependence-Aware Service Function Chains," *Journal of Optical Communications and Networking*, vol. 10, no. 8, pp. 64–74, 2018.
Journal article with article ID

[4] C. Bu, X. Wang, H. Cheng, M. Huang, K. Li, and S.-K. Das, "Enabling Adaptive Routing Service Customization via the integration of SDN and NFV," *Journal of Network and Computer Applications*, vol. 93, no. 9, pp. 123–136, 2017.

[5] L. Qu, C. Assi, and K. Shaban, "Delay-Aware Scheduling and Resource Optimization With Network Function Virtualization," *IEEE Transactions on Communications*, vol. 64, no. 9, pp. 3746–3758, 2016.

[6] C. Pham, N.-H. Tran, and C.-S. Hong, "Virtual Network Function Scheduling: A Matching Game Approach," *IEEE Communications Letters*, vol. 22, no. 1, pp. 22–1, 2017.

[7] M. Condoluci, and X. Xiao, "Soft Resource Reservation for Low-Delayed Teleoperation Over Mobile Networks," *IEEE Access*, vol. 15, pp.10445–10455, 2017.

[8] G. Xiong, Y.-X. Hu, L. Tian, J.-L. Lan, J.-F. Li, and Q. Zhou, "A virtual service placement approach based on improved quantum genetic algorithm," *Frontiers of Information Technology and Electronic Engineering*, vol. 17, no. 7, pp. 661–671, 2016.

[9] Z. Luo, and C. Wu, "An Online Algorithm for VNF Service Chain Scaling in Datacenters," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1061–1073, 2020.

[10] S. Troia, R. Alvizu, and G. Maier, "Reinforcement Learning for Service Function Chain Reconfiguration in NFV-SDN Metro-Core Optical Networks," *IEEE Access*, vol. 7, pp. 167944–167957, 2019.

[11] T. Subramanya, D. Harutyunyan, and R. Riggio, "Machine learning-driven service function chain placement and scaling in MEC-enabled 5G networks," *Computer networks*, vol. 166, no. 15, pp. 1–16, 2020.

[12] B. Martini, and F. Paganelli, "A Service-Oriented Approach for Dynamic Chaining of Virtual Network Functions over Multi-Provider Software-Defined Networks," *Future Internet*, vol. 8, no. 24, pp. 24–34, 2016.

[13] T. Truong-Huu, P.-M. Mohan, and M. Gurusamy, "Service Chain Embedding for Diversified 5G Slices with Virtual Network Function Sharing," *IEEE Communications Letters*, vol. 23, no. 5, pp. 826–829, 2019.

[14] Y. Wang, P. Lu, W. Lu, and Z.-Q. Zhu, "Cost-Efficient Virtual Network Function Graph (vNFG) Provisioning in Multidomain Elastic Optical Networks," *Journal of Lightwave Technology*, vol. 13, pp. 2712–2723, 2017.

[15] L. Yang, Y.-C. Chen, and T.-Y. Wu, "Provably secure client-server key management scheme in 5G networks," *Wireless Communications and Mobile Computing*, vol. 2021, 4083199, 2021.

[16] L.-L. Kang, R.-S. Chen, Y.-C. Chen, C.-C. Wang, X.-G. Li, and T.-Y. Wu, "Using Cache Optimization Method to Reduce Network Traffic in Communication Systems Based on Cloud Computing," *IEEE Access*, vol. 7, pp. 124397-124409, 2019.

[17] T.-Y. Wu, Z.-Y. Lee, M.-S. Obaidat, S. Kumari, and C.-M. Chen, "An Authenticated Key Exchange Protocol for Multi-server Architecture in 5G Networks," *IEEE Access*, vol. 8, pp. 28096-28018, 2020.

[18] H.-J. Xuan, S.-W. Wei, Y. Feng, D.-H. Liu, and Y.-L. Li, "Bi-level Programming Model and Algorithm for VNF Deployment with Data centers Placement," *IEEE Access*, vol. 7, pp. 185760–185772, 2019.

[19] N. Kiji, T. Sato, R. Shinkuma, and E. Oki, "Virtual network function placement and routing for multicast service chaining using merged paths," *Optical Switching and Networking*, vol. 36, 100554, 2020.

[20] W. Fang, M. Zeng, X. Liu, L. Wei, and Z.-Q. 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.

[21] X.-L. Chen, Z.-Q. Zhu, R. Proietti, and S.-J. Ben, "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.

[22] A.-M. Deif, and H.-A. Elmaraghy, "Modelling and analysis of dynamic capacity complexity in multistage production," *Production Planning and Control*, vol. 20, no. 8, pp. 737–749, 2009.

[23] D. Lu, N.-S. Fonseca, and Z.-Q. Zhu, "Application-driven Provisioning of Service Function Chains over Heterogeneous NFV Platforms," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3037–3048, 2020.

[24] B. Li, W. Lu, S. Liu, and Z.-Q. 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. 29–41, 2018.

[25] A. Hmaity, M. Savi, L. Askari, F. Musumeci, and A. Pattavina, "Latency-and Capacity-aware Placement of Chained Virtual Network Functions in FMC Metro Networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 35, pp. 100536, 2019.

[26] Y. Li, Y. Zhao,B. Li, X. Yu, H. Yang, X. Wang, and J. Zhang, "Joint balancing of IT and spectrum resources for selecting virtualized network function in inter-datacenter elastic optical networks," *Optics express*, vol. 27, no. 1, pp. 15116, 2019.

[27] J. Zu, G. Hu, Y. Wu, D. Shao, and J. Yan, "Resource Aware Chaining and Adaptive Capacity Scaling for Service Function Chains in Distributed Cloud Network," *IEEE Access*, vol. 7, pp. 157707-157723, 2019.

[28] Y.-W. Ma, J.-L. Chen, and J.-Y. Jhou, "Adaptive service function selection for Network Function Virtualization networking," *Future generation computer systems*, vol. 91, no. 2, pp.108–123, 2019.

[29] A. Laghrissi, and T. Taleb, "A Survey on the Placement of Virtual Resources and Virtual Network Functions," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp.1409–1434, 2018.

[30] Z. Yue, Y. Ou, H. Ali, K. Koteswararao, N. Reza, S. Dimitra, Y. Liu, and G. Lei, "Location-Aware Energy Efficient Virtual Network Embedding in Software-Defined Optical Data Center Networks," *Journal of Optical Communications and Networking*, vol. 10, no. 7, pp.B58, 2018.

[31] T. Li, H. Zhou, and H. Luo, "A new method for providing network services: Service function chain," *Optical Switching and Networking*, vol. 26, no. 11, pp.60–68, 2015.

[32] A.-H. Gandomi, X.-S. Yang, and A.-H. Alavi, "Erratum to: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 2, pp.245–245, 2013.

[33] 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.

[34] X. Chen, W. Ni, I.-B Collings, X. Wang, and S. Xu, Automated Function Placement and Online Optimization of Network Functions Virtualization, *IEEE Transactions on Communications*, vol.67, no.2, pp.1225–1237, 2019.

[35] 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.