Multi-objective Optimization Model and Improved Genetic Algorithm based on MOEA/D for VNF-SC Deployment

Na Li, Leijie Wang, Lidan Lin and Hejun Xuan

Abstract—**Network Function Virtualization (NFV) can provide the resource according to the request and can improve the flexibility of the network. It has become the key technology of next generation communication. Resource scheduling for virtual network function service chain (VNF-SC) mapping is the key issue of the NFV. A virtual network function service chain placement multi-objective optimization model and algorithm based on improved genetic algorithm is proposed. Firstly, a multi-objective optimization model, which minimizes deployment cost, transmission delay and maximizes the load balance, is established. On this basis, an improved genetic algorithm based on MOEA/D is proposed to solve the established multi-objective model. In this algorithm, the combination scheme and mapping scheme of the service request are coded by the two-level coding method in the mapping process, and then the improved sparrow search algorithm is used to obtain the service function chain deployment scheme of the request and calculate the mapping weight. In addition, an efficient individual generation strategy is proposed to generate some superior individual. Finally, some simulation experiments are conducted and the experimental results show that the algorithm can effectively reduce deployment cost and transmission delay than the compared algorithm.**

Index Terms—**Network Function Virtualization, Multi-objective, MOEA/D, Transmission delay**

I. Introduction

IN 5G network, network function virtualization is used to deploy the core network. Network functional virtualization is divided into three stages: group chain, deployment and scheduling[1], [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], [4]. 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 [5], [6]. Scheduling refers to the optimization decision of network services on VNF-SC mapping sequence[7], [8]. A reasonable mapping sequence can reduce delay overhead and improve node resource utilization and bandwidth resource utilization[9], [10]. Therefore, how to group, deploy and optimize the scheduling of service functions is a hot issue.

In the existing literatures, there are two main problems. First, consider the service function chain and step by step mapping space to reduce the solution space. Secondly, calculating the mapping weight of the service function chain based on the scheme of the chain stage rather than the mapping stage will lead to the deviation of the mapping. In this paper, a virtual network function service chain deployment multi-objective optimization model and algorithm based on improved genetic algorithm is proposed. Firstly, a multi-objective optimization model, which minimizes deployment cost, transmission delay and maximizes the load balance, is established. Then, an improved genetic algorithm based MOEA/D is proposed to solve the multi-objective model established. In the proposed algorithm, the composition scheme and the mapping scheme of service requests are coded together by a double-layer coding method in the mapping process, and then an improved sparrow search algorithm is used to obtain the service function chain deployment scheme of the request and calculate the mapping weight. In addition, an efficient individual generation strategy is proposed to generate some superior individual.

II. Related Work

At present, a large number of studies have been carried out on the deployment mechanism of VNF. Among them, literature [11] ensures the quality of user service while reducing the cost of operators, but it adopts a static VNF deployment strategy. Since the network environment is dynamically changing, long-term optimization needs to be considered. In order to minimize the end-to-end delay of VNF-SC[12], the end-to-end delay of VNF-SC can be reduced by reducing the transmission delay and processing delay of VNF-SC. However, the utilization rate of physical network resources is not paid attention to. In reference [13], when considering the server resource capacity and flow rate. The VNF operation cost, VNF instance maintenance cost and VNF deployment cost are balanced, but the VNF-SC delay is not considered, thus ignoring the user quality of service. In literature [14], [15], the deep reinforcement learning network is used in the scenarios of access network, cross-domain mapping and core

Manuscript received April 6, 2022; revised December 28, 2022. This work was supported by National Natural Science Foundation of China (No. 31872704), Science and Technology Department of Henan Province (No.222102210265), and by Foundation of Henan Educational Committee under Contract (No.22A520007).

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

Leijie Wang is a Lecturer at the Modern Education Technology Center, Luo He Vocational Technology College, Henan, Luohe, 462000, China. (e-mail: wljoffice@163.com).

Lidan Lin is a Lecturer at the Department of Information Engineering, Luo He Vocational Technology College, Henan, Luohe, 462000, China. (e-mail: Amy_lld@163.com).

Hejun Xuan is an Associate Professor at the Computer and Information Technology, Xinyang Normal University, Henan, Xinyang, 464000, China. (e-mail: xuanhejun0896@xynu.edu.cn).

network respectively, and the mapping algorithm of service function chain is proposed. Literature[16] 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 [17] 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. Literature [18] proposed an optimized mapping strategy that prioritizes mapping service function chains with fewer requirements to reduce delay overhead and improve resource utilization. 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. In [19], corresponding VNF-SC 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 (FMC) network, it maximizes the integration of existing VNF-SCs in current VNF-SCs. There are VNFs. Global balancing factor and the local balancing factor are proposed, a joint optimization selection algorithm is designed to select the appropriate choice for user requests[20]. Literature[21] proposeed 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-SCs. deployment efficiency. In literature [22], in order to avoid the unbalanced load of the network system, a higher VNF service chain is established. In order to meet the flexible business needs of users, a NFV-based VNF-SC integration architecture and VNF path selection mechanism are proposed. Literature [23] adopts replicated VNF to solve the problem of balancing network load and reducing resource cost in VNF deployment. 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. Reference [24] studied the VNFs deployment problem of NFV network mapping VNF-SCs request for the purpose of improving network performance and profit, and also considered the delay requirement of VNF-SCs request, and then described the problem as an integer in order to minimize resource consumption linear programming model, a new VNF-SC request mapping algorithm and VNF deployment algorithm are proposed to map VNF-SC requests and optimize VNF deployment.

III. Network and Problem Description

A. 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-SC requests, the VNF manager is responsible for VNF connections, and the infrastructure manager is responsible for global resource management[25].

B. Problem Description

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.

2) VNF-SC Request: 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 [26] describes the dependency relationship between service functions. 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, an VNF-SC 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-SC 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.

IV. Formula Description

A. Service function chain mapping priority

For VNF-SC 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, $0 \leq \gamma_1, \gamma_2, \gamma_3 \leq 1$, and $\gamma_1 + \gamma_2 + \gamma_3 = 1$.

B. 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 $0 \leq \alpha, \beta \leq 1$, 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, $0 \leq f_1 = ct/ct1 \leq 1$. The second objective is defined as:

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

(2) Transmission Delay: The end-to-end delay of the service path represents the time taken by the data flow from the source node to the destination node, and is composed of the execution delay of the service instance on the service path and the transmission delay of the communication link, as defined below:

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

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, $0 \leq f_2 = D(SP)/D'(SP) \leq 1$. The second objective is defined as:

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

(3) Load Degree: When constructing the service path, besides the functional requirements of the service, the load of the service bearing node and its adjacent links should be considered to map the service and logical links to the resource-rich underlying nodes and links as far as possible. Load degree (LD) measures the resource usage and load of a service path. Definition of the load intensity of the bottom node $n_{i_k}^R$ in the service path as follows:

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

The load intensity of a node takes into account the available computing capacity of the node and the computing capacity demand of the service. According to Formula (3), its value ranges from 0 to 1. The smaller the value is, the more likely it is to map services to this node, and the more beneficial it is to load balancing of the underlying node. Based on the idea of node load intensity, the load intensity of link S in the service path is defined as follows:

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

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

$$LD(SP) = w_N \sum_{n_{i_k}^S \in N^{SP}} LD_{n_{i_k}^S} + w_L \sum_{l_{i_k}^S \in L^{SP}} LD_{l_{i_k}^S} \tag{14}$$

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

$$LD^{'}(SP) = \sum_{n_{i_k}^S \in N^{SP}} LD_{n_{i_k}^S} + \sum_{l_{i_k}^S \in L^{SP}} LD_{l_{i_k}^S} \tag{15}$$

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

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

C. Constraints

(1) A service function 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.

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

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.

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

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

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

(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:

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

V. Proposed Algorithm

The mapping of service function chain is an NP-hard problem [27], which requires solving the optimal service function chain group chain, mapping and scheduling scheme, so the complexity of the problem is higher. As a population intelligent optimization algorithm, genetic algorithm has the advantages of strong optimization ability and fast convergence compared with traditional optimization algorithm, and is very suitable for optimization in dynamic and multi-objective scenarios [28]. However, genetic algorithm is often applied to solve continuous problems, but it is difficult to be directly applied to discrete problems[29]. Therefore, this paper proposes an improved genetic algorithm, which adds an optimization operator on the basis of the genetic algorithm to limit the position of the updated population, solves the problem of high randomness in the search process, and improves the convergence and stability of the algorithm. In order to coordinate the grouping and mapping process of service function chain, a double-layer coding method is introduced to combine the grouping and mapping scheme, so that VNF and physical nodes correspond to get the deployment scheme of service function chain. The improved genetic algorithm is used to optimize the deployment scheme to obtain the optimal deployment scheme, and the mapping weight of each VNF-SC is calculated. VNF-SC in the same slice are sorted first and then mapped according to the weight.

1	2	3	4	5
8	9	3	2	4

Fig. 1. Individual encoding and decoding

A. Encoding and Decoding

The construction scheme and mapping scheme of service function chain were coded, denoted as $p_{2 \times F}$. The upper F bit represents the construction scheme of the service function chain, and the lower F bit represents the location of the physical node mapped by the service function in the upper F bit of the service function chain. Fig.1 is an individual coding diagram, indicating that VNF1, VNF2, VNF3, VNF4 and VNF5 are mapped to nodes 8, 9, 3, 2 and 4 of the underlying network respectively. The K-Dijkstra algorithm is used to calculate all paths between any two physical nodes as a path set. In the decoding process, the shortest path satisfying the requirements of routing and forwarding relationship and bandwidth is found in the path set.

B. Fitness Value

The fitness of an individual is used to evaluate the quality of an individual in the population. In this paper, the deployment cost of service function chain is taken as the fitness value, so the smaller the fitness of an individual is, the better the individual is. The fitness expression of the i-th sparrow is shown in Formula (8), (11) and (16).

C. Crossover

Crossover operator is the most important operation in genetic algorithm to maintain the diversity of population. For different optimization problems, there are many different crossover operators. This paper designs the following crossover operators to generate new individuals.

$$x_{ij}^{k+1} = x_{bj}^{k+1} + \eta \times \|x_{ij}^k - x_{bj}^k\| \tag{26}$$

D. Mutation

Mutation operation is to change the gene value at a local with a small probability. It is also an operation method to generate new individuals. Mutation operators can improve the diversity of the population and improve the search ability of the algorithm.

$$x_{ij}^{k+1} = \begin{cases} x_{ij}^k + \beta \|x_{ij}^k - x_{bj}^k\|, f_i > f_b \\ x_{ij}^k + K \frac{\|x_{ij}^k - x_{bj}^k\|}{f_i - min\{f_1, f_2, f_3\}}, & else \end{cases} \tag{27}$$

E. MOEA/D

The decomposition-based multi-objective evolutionary algorithm (MOEA/D) transforms the multi-objective optimization problem into a series of single-objective optimization subproblems. Then, using the information of a certain number of neighboring problems, the evolutionary algorithm is used to optimize these sub-problems. Because a solution on Pareto's front surface corresponds to the optimal solution of every single objective optimization subproblem, a group of Pareto optimal solutions can be finally obtained. Due to the existence of decomposition operation, this method has a great advantage in maintaining the distribution of solutions, and can avoid falling into local optimum by analyzing the information of adjacent problems.

VI. Experiments and Analysis

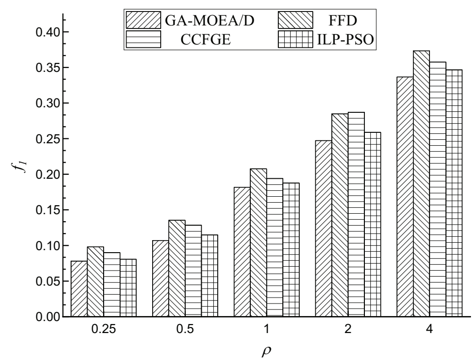
A. Experimental Setting

The underlying network topology is randomly generated by GT-ITM tool, which contains 50 nodes and about 130 links. The computing capacity of the bottom node and the bandwidth of the bottom link are evenly distributed [50,100], and the cost of unit computing resource c_S and bandwidth resource b_s are 1. The number of service types supported by the underlying network is set to 10. Each underlying node provides one to five service types randomly. According to literature [5,24], the processing time of a service instance depends on the type of the service and the processing capacity 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 above principle and ranges from 1 to 10 time units.

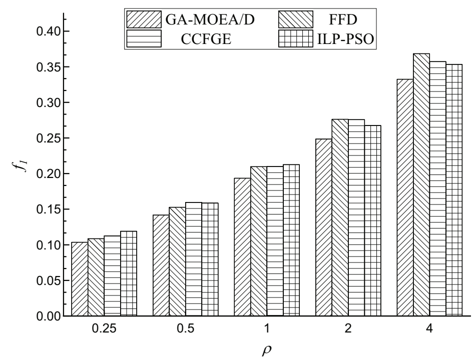
The arrival process of service requests obeys the Poisson distribution, with an average of four requests arriving within 100 time units, and the duration of each service request obeys an exponential distribution with an average of 1000 time units. The service chain is composed of four services, the service type is random and non-repetitive, the computing power required by each service is evenly distributed in [1, 50], and the bandwidth required by logical link is evenly distributed in [1, 50], and the charges c_R and b_R for unit computing power and unit bandwidth are both 1. The maximum allowable end-to-end delay D_{max} is set to 100 time units. The time of each simulation experiment was about 50000 time units, and the data were recorded every 4000 time units from the 2000 time unit. Each group was set up for 10 simulation experiments, and the experimental results were averaged. The population size was set as 100, the upper limit of iterations was set as 10000,

B. Experimental Results

To demonstrate the performance of the proposed algorithm, three compared algorithm are introduced. The first literature[30] proposed a multi-objective meta-heuristic solution (denoted as FFD), which uses the non-dominated sorting genetic algorithm for VNF-SC deployment problem. The purpose of this algorithm is to place VNFs based on different service chains onto physical hosts in such a way that. First, physical resource utilization is maximized. Second, the number of used physical hosts is minimized. Similarly, literature [31] investigated the tradeoff between end-to-end reliability and computational load per server via the joint design of VNF chain composition and forward graph embedding under the assumption of a bipartite forward graph that consists of a controller and regular VNFs, denoted as C-CFGE. Literature[32] proposed a formulation of this problem as an Integer Linear Program (ILP-PSO) that allows one to find the best feasible paths and virtual function placement for a set of services with respect to a total financial cost, while taking into account the order constraints for service function chains of each service and other constraints such as end-to-end latency, anti-affinity rules between network functions on the same physical node and resource limitations in terms of network and processing capacities. The number of data center nodes are fixed as $N_D = N_V/5$, $N_D = 2N_V/5$, $N_D = 3N_V/5$ and $N_D = 4N_V/5$. 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. Figure 2 to Figure 5 show the Profit obtained in CHNNET and ARPANET when $N_D = N_V/5$, $N_D = 2N_V/5$, $N_D = 3N_V/5$ and $N_D = 4N_V/5$.



(a) Deployment overhead obtained in CHNNET.



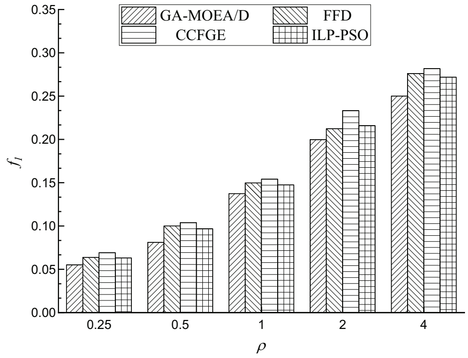
(b) Deployment overhead obtained in ARPANET.

Fig. 2. Deployment overhead obtained when $N_D = N_V/5$.

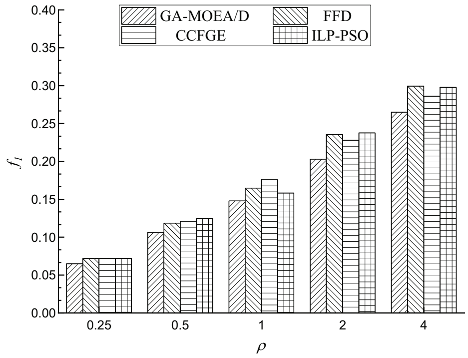
The Transmission Delay obtained in CHNNET and ARPANET when $N_D = N_V/5$, $N_D = 2N_V/5$, $N_D = 3N_V/5$ and $N_D = 4N_V/5$ are shown in Figure 6 to Figure 9, respectively.

Similarly, Figure 10 and Figure 13 show the Load Degree in CHNNET and ARPANET when $N_D = N_V/5$, $N_D = 2N_V/5$, $N_D = 3N_V/5$ and $N_D = 4N_V/5$.

To demonstrate the uniformity, convergence, diversity of the proposed algorithm, the following two metrics are used to evaluate the pareto solutions:

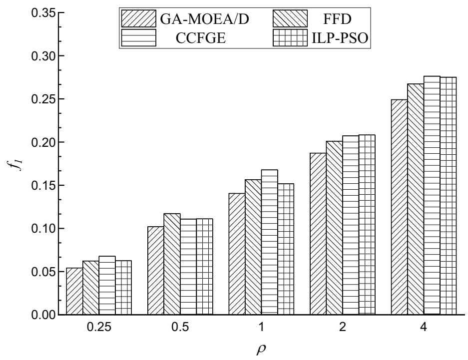


(a) Deployment overhead obtained in CHNNET.

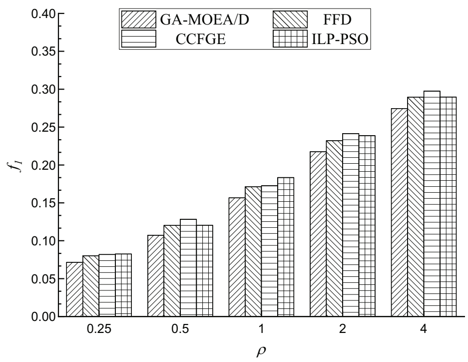


(b) Deployment overhead obtained in ARPANET.

Fig. 3. Deployment overhead obtained when $N_D = 2N_V/5$.

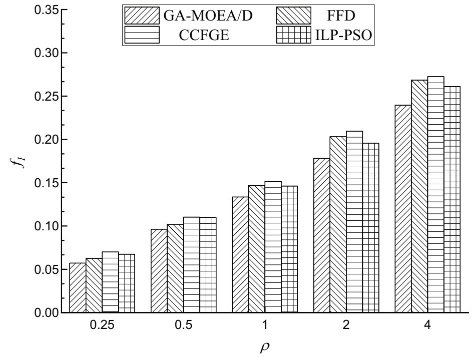


(a) Deployment overhead obtained in CHNNET.

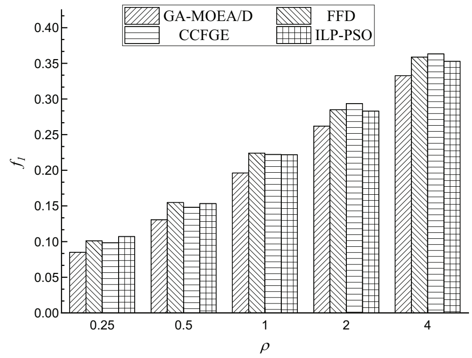


(b) Deployment overhead obtained in ARPANET.

Fig. 4. Deployment overhead obtained when $N_D = 3N_V/5$.

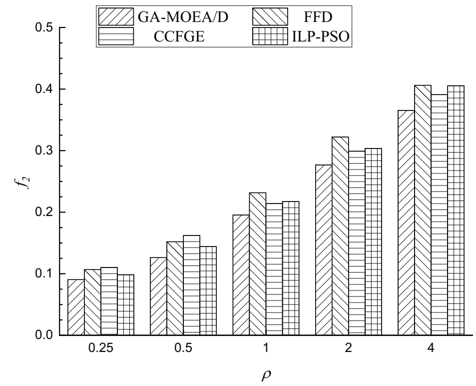


(a) Deployment overhead obtained in CHNNET.

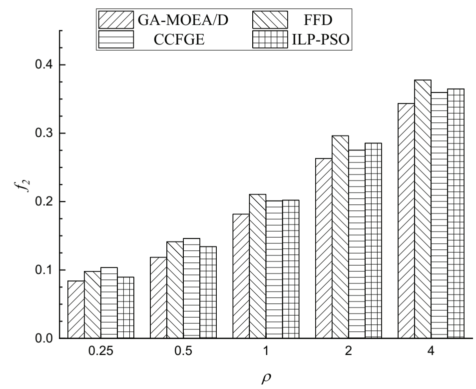


(b) Deployment overhead obtained in ARPANET.

Fig. 5. Deployment overhead obtained when $N_D = 4N_V/5$.

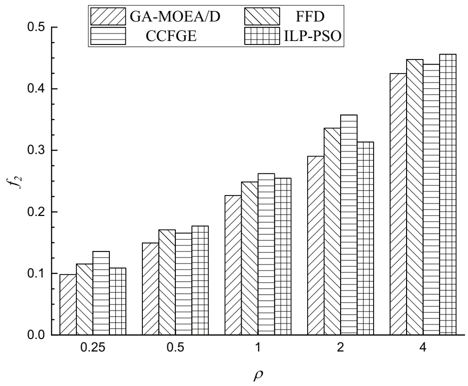


(a) Transmission delay obtained in CHNNET.

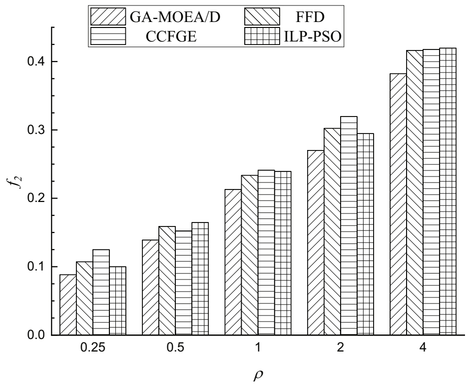


(b) Transmission delay obtained in ARPANET.

Fig. 6. Transmission delay obtained when $N_D = N_V/5$.

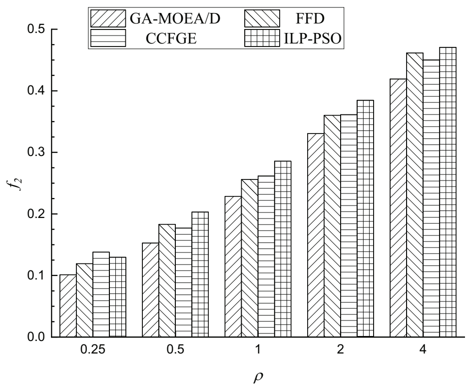


(a) Transmission delay obtained in CHNNET.

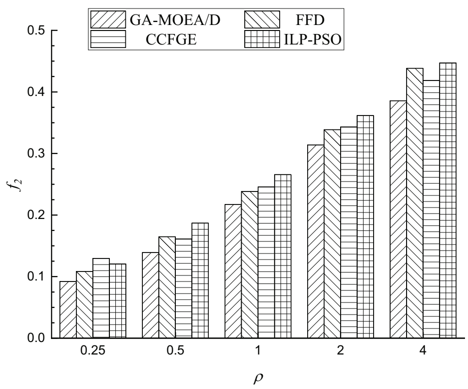


(b) Transmission delay obtained in ARPANET.

Fig. 7. Transmission delay obtained when $N_D = 2N_V/5$.

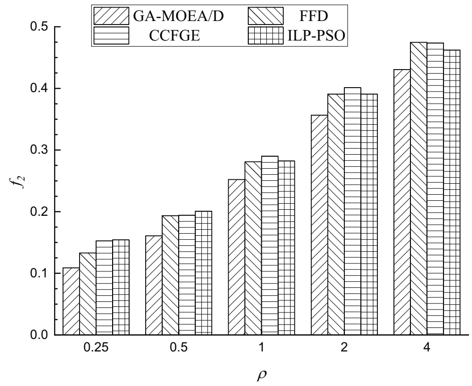


(a) Transmission delay obtained in CHNNET.

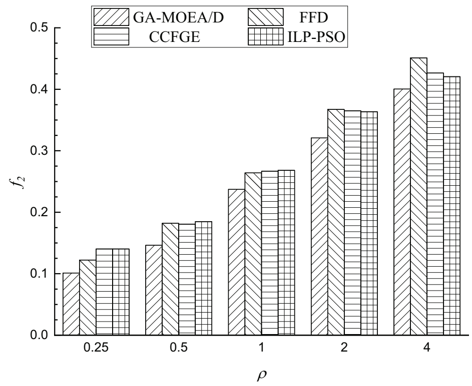


(b) Transmission delay obtained in ARPANET.

Fig. 8. Transmission delay obtained when $N_D = 3N_V/5$.

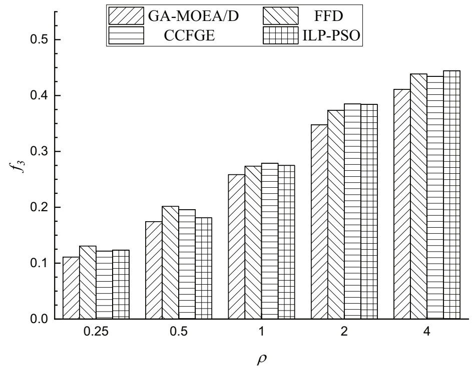


(a) Transmission delay obtained in CHNNET.

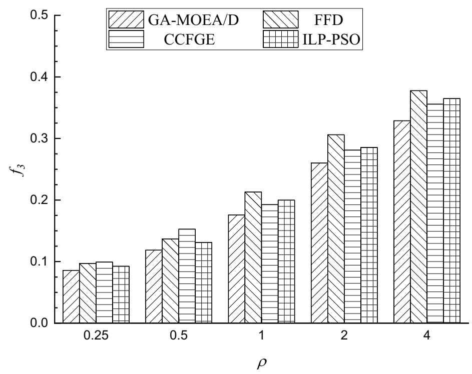


(b) Transmission delay obtained in ARPANET.

Fig. 9. Transmission delay obtained when $N_D = 4N_V/5$.

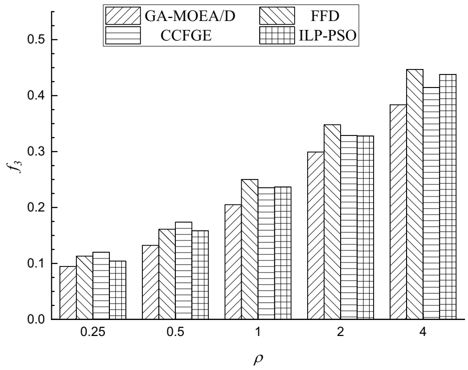


(a) Load degree obtained in CHNNET.

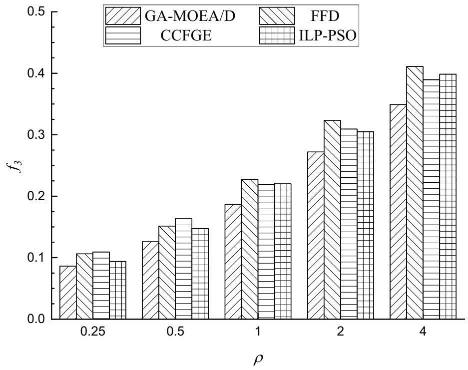


(b) Load degree obtained in ARPANET.

Fig. 10. Load degree obtained when $N_D = N_V/5$.

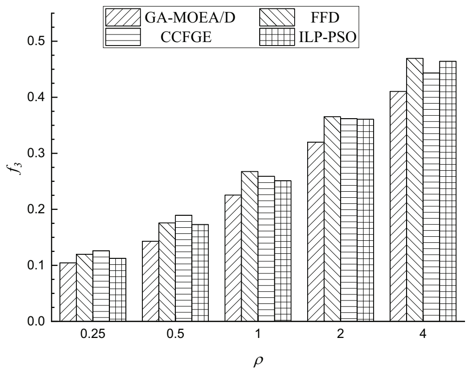


(a) Load degree obtained in CHNNET.

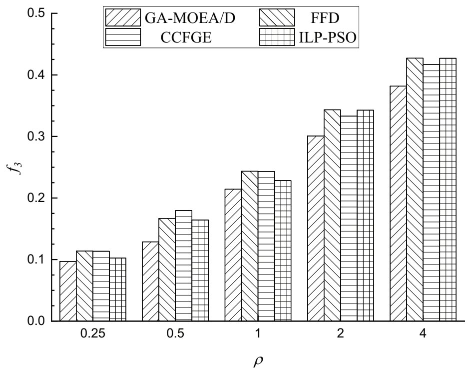


(b) Load degree obtained in ARPANET.

Fig. 11. Load degree obtained when $N_D = 2N_V/5$.

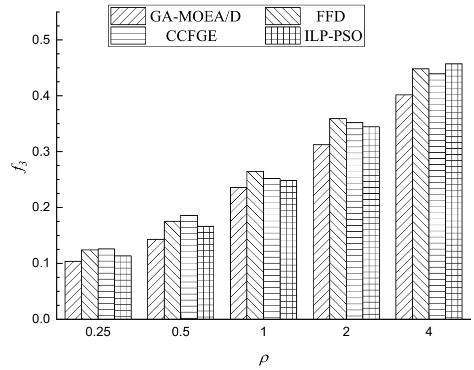


(a) Load degree obtained in CHNNET.

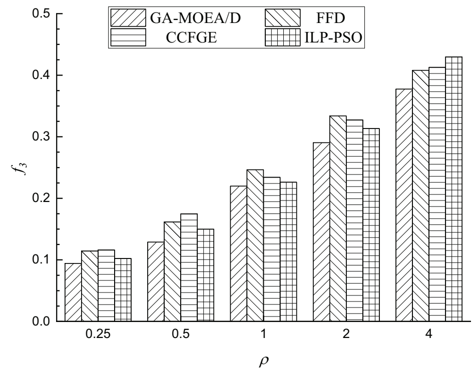


(b) Load degree obtained in ARPANET.

Fig. 12. Load degree obtained when $N_D = 3N_V/5$.



(a) Load degree obtained in CHNNET.



(b) Load degree obtained in ARPANET.

Fig. 13. Load degree obtained when $N_D = 4N_V/5$.

- *Spacing Index*(SI): defined by Eq.(28) below.

$$\begin{cases} SI(A) = \sqrt{\frac{1}{|PF^*|-1} \sum\limits_{z \in PF^*} \left(\bar{d} - d(z)\right)^2} \\ d(z) = \min\left\{\|z - z'\| \| z \neq z', z' \in PF^*\right\} \\ \bar{d} = \frac{1}{|PF^*|} \sum\limits_{z \in PF^*} d(z) \end{cases} \tag{28}$$

  Spacing Index is used to metric the uniformity of the pareto solution. The smaller of SI, the better of the solutions.
- *Hypervolume Index*(HI): which is used to test the uniformity, convergence and diversity of the solutions, and defined by the following Eq.(29).

$$HI(PF^*) = \bigcup_{z \in PF^*} vol(z) \tag{29}$$

  where $vol(z)$ is the hypervolume of area which is surrounded by z and the reference point $r = (r_1, r_2, \ldots, r_m)$. m is the dimensionality of the objective space.

C. Experimental Results Analysis

As the number of VNF-SC increases, both the deployment cost and the total system delay of the four algorithms will increase. Since the algorithm proposed in this paper optimizes the delay, its total system delay is the lowest. Considering the deployment cost of VNF and reasonable resource allocation, the deployment cost is the lowest. Comparison algorithm due to simultaneously optimize deployment of operational failure cost and time delay, so its total delay is higher than the paper algorithm, secondly, the comparison method is applicable to solve discrete searching space, because the search space is

TABLE I
STATISTICAL RESULTS (MEAN AND STANDARD DEVIATION) OF THE SI AND HI.

		NSFNET		CHNNET	
		SI	HI	SI	HI
0.2	0.25	1.2353(1.54E-02)	7.3561(2.64E-01)	2.2358(2.46E-02)	10.6547(3.48E-01)
	0.5	2.0981(2.35E-02)	8.2514(3.47E-01)	3.3214(3.65E-02)	11.4356(4.61E-01)
	1	2.8760(3.27E-02)	8.9947(4.62E-01)	4.5463(4.28E-02)	12.6533(5.39E-01)
	2	3.5651(4.64E-02)	9.3451(5.57E-01)	5.0467(5.62E-02)	15.5464(6.61E-01)
	4	4.3291(5.29E-02)	10.2367(6.13E-01)	5.9865(6.79E-02)	16.6574(7.49E-01)
0.4	0.25	2.0156(3.54E-02)	8.3456(3.82E-01)	2.8789(3.72E-02)	11.4575(4.79E-01)
	0.5	2.7688(4.62E-02)	9.2648(4.46E-01)	3.8325(4.68E-02)	12.7654(5.28E-01)
	1	3.4522(5.38E-02)	10.1498(5.71E-01)	4.8127(5.29E-02)	13.5473(6.84E-01)
	2	4.1871(6.36E-02)	10.9889(6.47E-01)	5.9783(6.36E-02)	14.9750(7.32E-01)
	4	4.9747(7.63E-02)	11.7689(7.38E-01)	6.5473(7.28E-02)	15.8796(8.48E-01)
0.6	0.25	2.5632(3.66E-02)	9.6785(4.43E-01)	3.2345(3.70E-02)	11.7685(5.26E-01)
	0.5	3.6732(4.21E-02)	10.4527(5.64E-01)	4.6534(4.57E-02)	12.5484(6.63E-01)
	1	4.2474(5.57E-02)	11.3457(6.79E-01)	5.4743(5.72E-02)	13.6975(7.57E-01)
	2	5.0352(6.58E-02)	12.6538(7.75E-01)	6.5436(6.39E-02)	14.5742(8.45E-01)
	4	5.9927(7.49E-02)	13.5481(8.50E-01)	7.8735(7.28E-02)	15.7668(9.31E-01)
0.8	0.25	3.5622(4.83E-02)	10.6538(4.34E-01)	4.2453(4.49E-02)	11.6732(6.56E-01)
	0.5	4.3463(5.91E-02)	11.4363(5.21E-01)	5.3557(5.62E-02)	12.9742(7.63E-01)
	1	5.1354(6.73E-02)	12.7636(6.54E-01)	6.2826(6.35E-02)	13.6741(8.47E-01)
	2	6.2512(7.82E-02)	13.6545(7.01E-01)	7.8643(7.19E-02)	14.8762(9.22E-01)
	4	7.0154(8.38E-02)	14.3621(7.72E-01)	8.5473(8.36E-02)	15.9715(10.36E-01)

continuous values, so the comparison algorithm in optimization of time delay and VNF deployment cost significantly inferior to algorithm and improved algorithm. Moreover, the proposed algorithm has lower deployment cost and total system delay than the comparison algorithm, because the text algorithm can find a better non-dominated solution and obtain a better objective function. Therefore, the utility of the proposed algorithm is the largest.

With the number of virtual network functional service chains increases, the maximum frequency slots number occupied in the network increases gradually. Because the comparison algorithm does not consider the dependency relationship between virtual network functions and adopts a fixed order to configure virtual network functions, it cannot solve the configuration problem of virtual network functions that consider the dependency relationship between virtual network services. The proposed algorithm considers the dependence between different virtual network functions and can search the optimal configuration scheme through multiple iterations. Therefore, the proposed algorithm can obtain better results than the comparison algorithm. The experimental results also show that the proposed algorithm can get lower cost than the comparison algorithm. When the number of virtual network functional service chains is $\rho = 0.25$, the proposed algorithm can get 2.9%-4.8% less cost comparison algorithm in the network. When the number of virtual network functional service chains is $\rho = 4$, the proposed algorithm can get 8.2%-10.7% less cost comparison algorithm in the network. It can be seen that for the same network topology and the same number of virtual network functional service chains, when the number of connected data centers in the network increases, the network cost decreases. When the number of data centers is small, these nodes in the network will become key nodes, and more virtual network function service chains will pass through this node, which also leads to more virtual network function service chains passing through links connected to this node, so that the cost occupied in the network will be large. On the contrary, when the number of data centers is small, the virtual network functional service chain will occupy different links more evenly, which will reduce the cost of network occupation.

As the number of virtual network function service chains increases, the network delay increases gradually. The experimental results also show that the designed algorithm can get smaller network delay than the comparison algorithm. When the number of virtual network functional service chains is $\rho = 0.25$, the network delay of the proposed algorithm can be 1.9%-3.2% smaller than that of the comparison algorithm, and when the number of virtual network functional service chains is $\rho = 4$, the network delay of the proposed algorithm can be 5.7%-9.4% smaller than that of the comparison algorithm.

As the number of virtual network function service chains increases, the number of virtual network functions configured on data centers increases. The experimental results also show that the designed algorithm can be more balanced than the comparison algorithm. When the number of virtual network functional service chains is $\rho = 0.25$, the proposed algorithm can obtain the network equilibrium degree is 4.2%-5.1% smaller than that of the comparison algorithm, and when the number of virtual network functional service chains is $\rho = 4$, the proposed algorithm can obtain the network equilibrium degree is 6.9%-8.1% smaller than that of the comparison algorithm.

VII. CONCLUSION

Network Function Virtualization (NFV) can provide the resource according to the request and can improve the flexibility of the network. It has become the key technology of next generation communication. Resource scheduling for virtual network function service chain (VNF-SC) mapping is the key issue of the NFV. A virtual network function service chain placement multi-objective optimization model and algorithm based on improved genetic algorithm is proposed. Firstly, a multi-objective optimization model, which minimizes deployment cost, transmission delay and maximizes the load balance, is established. Then, an improved genetic algorithm based MOEA/D is proposed to solve the multi-objective model established. In the proposed algorithm, The composition schemes and the mapping schemes of service requests are coded together by a double-layer coding method in the mapping process, and then an improved sparrow search algorithm is used to obtain the service function chain deployment scheme of the request and calculate the mapping weight. In addition, an efficient individual generation strategy is proposed to generate some superior individual. Finally, some simulation experiments are conducted and the experimental results show that the algorithm can effectively reduce deployment cost and transmission delay than the compared algorithm.

REFERENCES

[1] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Service function chain embedding for nfv-enabled iot based on deep reinforcement learning," *IEEE Communications Magazine*, vol. 57, no. 11, pp. 102–108, 2019.
[2] 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.
[3] M. Jalalitabar, E. Guler, D. Zheng, G. Luo, L. Tian, and X. Cao, "Embedding dependence-aware service function chains," *Journal of Optical Communications & Networking*, vol. 10, no. 8, pp. 64–74, 2018.
[4] 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.
[5] 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 & Computer Applications*, vol. 93, no. sep., pp. 123–136, 2017.
[6] 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.
[7] 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.
[8] 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.
[9] 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. 69–72, 2017.
[10] J. F. Y. X. F. Z. Y. L. Hejun Xuan, Xuelin Zhao, "Vnf service chain deployment algorithm in 5g communication based on reinforcement learning," *IAENG International Journal of Computer Science*, vol. 48, no. 1, pp. 1–7, 2021.
[11] M. Condoluci, Member, IEEE, X. Xiao, and S. Member, "Soft resource reservation for low-delayed teleoperation over mobile networks," *IEEE Access*, vol. 5, pp. 10 445–10 455, 2017.
[12] 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 & Electronic Engineering*, vol. 17, no. 7, pp. 661–671, 2016.
[13] Z. Luo and C. Wu, "An online algorithm for vnf service chain scaling in datacenters," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, p. 1061C1073, 2020.
[14] 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. 167 944–167 957, 2019.
[15] 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. Jan.15, pp. 1–16, 2020.
[16] 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. 2, pp. 24–24, 2016.
[17] 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.
[18] Y. Wang, P. Lu, W. Lu, and Z. Zhu, "Cost-efficient virtual network function graph (vnfg) provisioning in multidomain elastic optical networks," *Journal of Lightwave Technology*, no. 13, pp. 2712–2723, 2017.
[19] 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," *Optical Switching and Networking*, vol. 35, p. 100536, January 2020.
[20] 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. 11, p. 15116, 2019.
[21] 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 (july 2019)," *IEEE Access*, vol. 7, pp. 157 707–157 723, 2019.
[22] 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. FEB., pp. 108–123, 2019.
[23] A. Laghrissi and T. Taleb, "A survey on the placement of virtual resources and virtual network functions," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1409–1434, 2018.
[24] 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 & Networking*, vol. 10, no. 7, p. B58, 2018.
[25] T. Li, H. Zhou, and H. Luo, "A new method for providing network services: Service function chain," *Optical Switching & Networking*, vol. 26, no. nov., pp. 60–68, 2015.
[26] 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.
[27] 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.
[28] Y. You, J. Cao, and W. Zhou, "A survey of change detection methods based on remote sensing images for multi-source and multi-objective scenarios," *Remote Sensing*, vol. 12, no. 15, p. 2460, 2020.
[29] J. Krause, J. Cordeiro, R. S. Parpinelli, and H. S. Lopes, "A survey of swarm algorithms applied to discrete optimization problems," *Swarm Intelligence and Bio-Inspired Computation*, vol. 04, no. 9, pp. 169–191, 2013.
[30] J. Kang, O. Simeone, and J. Kang, "On the trade-off between computational load and reliability for network function virtualization," *IEEE Communications Letters*, vol. 21, no. 8, pp. 1767–1770, 2017.
[31] S. Tavakoli-Someh and M. H. Rezvani, "Multi-objective virtual network function placement using nsga-ii meta-heuristic approach," *Journal of Supercomputing*, vol. 75, p. 6451C6487, 2019.
[32] Z. Allybokus, N. Perrot, J. Leguay, L. Maggi, and E. Gourdin, "Virtual function placement for service chaining with partial orders and anti-affinity rules," *Networks*, vol. 71, no. 2, pp. 97–106, 2017.