Wireless Networks
https://doi.org/10.1007/s11276-021-02741-9

A survey of VNF forwarding graph embedding in B5G/6G networks

Biao Zhang[1] · Qilin Fan[1] · Xu Zhang[2] · Zhihan Fu[1] · Sen Wang[1] · Jian Li[3] · Qingyu Xiong[1]

Accepted: 28 July 2021

Abstract
With the development of heterogeneous network structure, dynamic user requests as well as complex service types and applications scenarios, current networks may not accommodate the increasingly stringent requirements. As a result, the research of the beyond fifth generation (B5G) or the sixth generation (6G) networks has been put on the agenda. In B5G/6G networks, achieving the automatic, flexible, and cost-effective orchestration and management of network resources is a significant but challenging issue. Network function virtualization (NFV), as a promising paradigm to address this issue, has received considerable attention from both industry and academia. NFV leverages the virtualization technology to decouple network functions from dedicated hardware appliances to software middleboxes or called virtual network functions (VNFs) that run on the commodity servers. The demand for a network service becomes a request for running a set of VNFs deployed on the substrate network. The requested network service is orchestrated in the form of a VNF-forwarding graph (VNF-FG). The problem of embedding the VNF-FG into the substrate network is known as VNF-FG embedding (VNF-FGE). The efficiency and the management cost of a network are highly dependent on the optimization of VNF-FGE. This paper mainly presents a survey on solving the VNF-FGE problem. To this end, we present a general formulation and several objectives of the VNF-FGE problem. In the meanwhile, we summarize its different application scenarios from four perspectives and divide the approaches into four main categories based on the optimization methods. The main challenges and potential future directions due to the appearance of B5G/6G are also discussed.

Keywords B5G/6G · NFV · VNF-FGE

1 Introduction

The last few years have witnessed the sky-rocketing growth of mobile data traffic. According to the report of International Telecommunication Union [1], the overall mobile traffic per month will reach to 5016 EB in 2030. The fifth generation (5G) network is gradually being deployed and would provide significant improvements over the existing fourth generation (4G) network. However, with the development of heterogeneous network structure, dynamic user requests as well as complex service types and application scenarios, current networks may not accommodate the increasingly stringent requirements. The researches on the beyond 5G (B5G) or the sixth generation (6G) networks have been stimulated. You *et al.* [2] gave a future vision on 6G networks. It is expected to provide wider coverage, enhanced spectral/energy/cost efficiency, better intelligence level and security, etc.

The traditional network functions (NFs) (e.g., firewall, load balancing, and intrusion detection system) produced by the network service suppliers generally run on the dedicated hardware appliances. It will lead to the following problems: (1) The development life cycle is long as the network-enabled applications are required to be designed with high quality, high stability, and strict protocol support; (2) The flexibility is poor as some specific network functions rely heavily on the specialized hardware appliances; (3) The mutability of the demand of users makes it hard for the dedicated hardware appliances to adapt to its frequent

✉ Qilin Fan
fanqilin@cqu.edu.cn

[1] School of Big Data & Software Engineering, Chongqing University, Chongqing 401331, China

[2] College of Engineering, Mathematics & Physical Sciences, University of Exeter, Exeter EX4 4QF, UK

[3] Department of Electrical and Computer Engineering, Binghamton University, State University of New York, Binghamton, NY 13902, USA

Published online: 06 August 2021

changes; (4) There is no unified standard for virtualization technology, which makes the network functional hardware appliances produced by different manufacturers differ from each other. Network function virtualization (NFV), as a key enabler for 5G networks, is also playing an important role in B5G/6G networks. It provides an inspiration for designing, deploying, and managing network functions through virtualization technology and can be recognized as a promising paradigm of next-generation network architecture due to its open, flexible standardization and low management cost. The core technology of NFV is to decouple network functions from dedicated hardware appliances to software middleboxes or called virtual network functions (VNFs) that run on the commodity servers. Figure 1 shows the traditional network functions and virtual network functions (VNFs), respectively. Generally, for a given user request, more than one NFs are demanded. These NFs can be implemented and run on virtual machines or containers on the server by virtualization technology and can be easily acquired for normal use without having to purchase and lease new hardware appliances.

Therefore, NFV facilitates a move towards flexible and scalable service provisions. NFV-based resource allocation is one of the main challenges for the deployment of NFV for service providers. According to the article [3], NFV-based resource allocation is carried out in three stages: (1) *VNF chain composition (VNF-CC)*. Service providers need to concatenate different VNFs efficiently in order to compose virtual network function forwarding graphs (VNF-FGs) with respect to the customized requirement of users; (2) *VNF forwarding graph embedding (VNF-FGE)*. Then the output of VNF-CC (i.e., VNF-FGs) is taken as the input of VNF-FGE process. This stage aims to assign VNFs into appropriate locations in the underlying substrate network (SN); and 3) *VNF scheduling (VNF-SCH)*. Based on previous stages, VNF-SCH is to schedule the VNFs to minimize the total execution time while guaranteeing the service performance and the corresponding constraints. This paper mainly gives a survey on VNF-FGE.

The requested service is orchestrated in the form of a VNF-FG defined by European Telecommunications Standards Institute (ETSI) [4], also known as service function chain (SFC) in Internet Engineering Task Force (IETF) [5]. SFC is similar to the VNF-FG in the IETF architecture by definition as it can be regarded as a simplified straight chain of VNF-FG. VNF-FGE is to embed VNF-FGs into SN. Under general circumstances, we transform the requirements of users for NFs into the form of VNF-FGs or take the resulting graph generated in the stage of VNF-CC as the beginning of VNF-FGE. VNF-FG is composed of the ordered set of VNFs with requirements of resources (e.g., memory, bandwidth) and quality of service (QoS) (e.g., delay, packet loss). When each link and each node of VNF-FGs are mapped to the SN satisfying the constraints, the embedding process of VNF-FGs is terminated.

There are several existing surveys discussing the problem of VNF-FGE or SFC deployment. In particular, Herrera *et al.* [3] mainly elaborated the three stages (including VNF-FGE) of resource allocation in NFV environment as mentioned above and analyzed the relevant pieces of works in these stages. Bhamare *et al.* [6] mainly focused on the work on the SFC deployment problem. They provided a

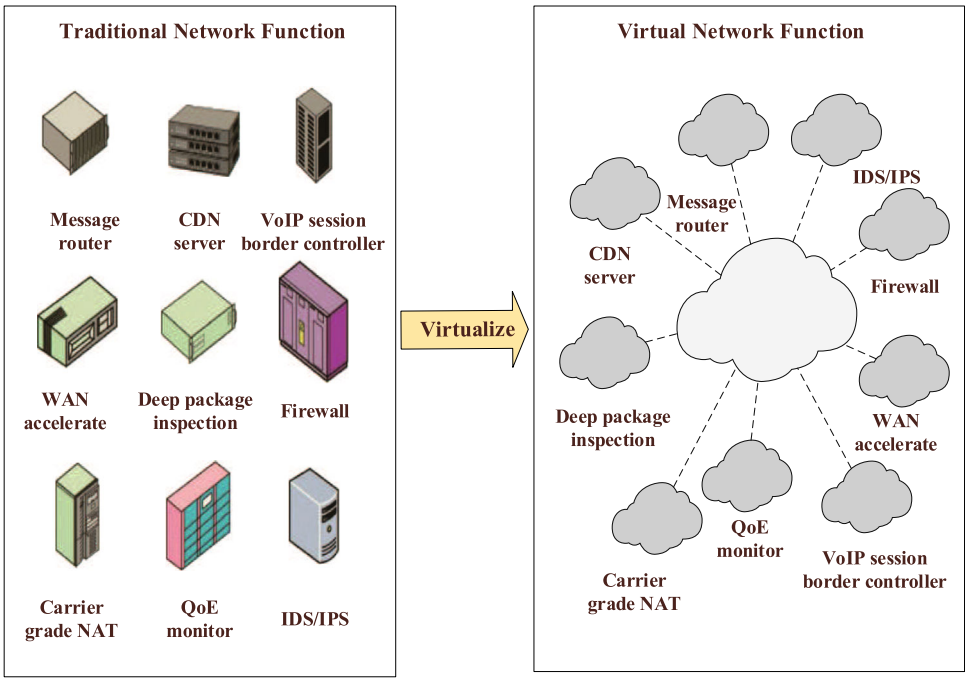


Fig. 1 Traditional network function and virtual network function

closer look at the current SFC architecture and gave a survey of the recent developments in SFC. Xie *et al.* [7] presented a survey of current researches in SFC deployment algorithms and several variants of SFC deployment problem. Mirjalily *et al.* [8] discussed the SFC from four stages and classified the existing approaches.

However, most of the existing surveys on VNF-FGE focused on the simplified VNF-FG (i.e., SFC) or discussed all the stages of resource allocation in the NFV environment. They are confronted with the following limitations: (1) There are no formulation models on VNF-FGE that have been discussed; (2) With the development of new network architectures (e.g., B5G/6G), some complex application scenarios are not included; (3) Some machine learning-based optimization strategies are missing in the surveys while the artificial intelligence is widely used in the network field in recent years [9]; (4) The most recent survey on VNF-FGE was published in 2018, which might result in an information lag.

According to the latest research advance, we give a comprehensive survey of the VNF-FGE problem. The main contributions of this survey are as follows:

- We mainly discuss the classification of the resources involving in the VNF-FGE problem. Moreover, we present a general formulation and several objectives of the VNF-FGE problem;
- We summarize different application scenarios of the VNF-FGE problem from four perspectives;
- We divide the approaches in existing works into four categories according to the distinct optimization methods;
- We conclude the emerging research directions on the VNF-FGE problem from several perspectives.

The remainder of this paper is organized as follows. We formulate the VNF-FGE problem in Sect. 2 and then present its different application scenarios in Section 3. We elaborate on the classification of optimization approaches of the VNF-FGE problem in Sect. 4. Bringing Section 3 and Section 4 together, we give a comprehensive taxonomy in Section 5. Section 6 discusses the promising future directions in the VNF-FGE field. Finally, we conclude this survey in Sect. 7. The list of abbreviations appeared in this paper is given in Table 1.

2 VNF-FGE Problem Formulation

In this section, we firstly introduce the background of VNF-FGE in Sect. 2.1. Then we give classifications for resources of VNF-FGE in Section 2.2. Finally, we present a typical mathematical model to describe the VNF-FGE in Section 2.3.

2.1 Background of VNF-FGE

NFV is a concept of network architecture that uses virtualization technology to divide the functions of the network node hierarchy into several functional blocks, which are implemented in software middleboxes and are no longer limited to hardware appliances. NFV is recognized as the primary means of next-generation network architecture due to its open, flexible standardization, and low management cost.

Under the organization of ETSI, some network operators have formally established an NFV working group, namely ETSI ISG NFV. This group is dedicated to the realization of network virtualization requirements definition and system architecture formulation and has proposed a general NFV network architecture [4] (shown in Fig. 2).

The NFV architecture mainly consists of three parts: NFV infrastructure (NFVI), VNFs, and NFV management and orchestration (NFV-MANO). NFVI includes the virtualization layer (hypervisor or container management system, such as docker and vswitch) and physical equipment, such as off-the-shelf servers, switches, storage devices, etc. NFVI is a general virtualization layer. All virtual resources should be managed in a unified shared resource pool, and certain VNFs running on it should not be restricted or treated specially. The VNFs that provide some kinds of network services are deployed in virtual machines, containers, or bare-metal physical machines using the infrastructures provided by NFVI. The NFV-MANO provides the overall management and orchestration of NFV, with an upward connection to the operations support system (OSS) or business support system (BSS) landscape. It is composed of NFV orchestrator (NFVO), VNF manager (VNFM), and virtualized infrastructure manager (VIM).

The management and orchestration of VNF is still a challenging task in B5G/6G networks. First, the heterogeneous network structure, dynamic user requests as well as complex service types and application scenarios in B5G/6G networks make the complexity of VNF-FGEs further increased. Second, due to the upgrading of wireless networks, the NFV architecture and software defined networks are also evolving [10]. The network is more likely to suffer from unexpected and unforeseen failures, which may be challenging to handle. Third, massive terminals and personalized services also bring about security issues in B5G/6G networks [11]. All these make the management and orchestration of VNF in NFVI particularly important but challenging. Therefore, the study of VNF-FGE, a type of management and orchestration of VNF, is also a promising research field.

Table 1 A list of abbreviations

4G	Fourth generation
5G	Fifth generation
5G	Fifth generation
6G	Sixth generation
B5G	Beyond fifth generation
BIP	Binary integer programming
BSS	Business support system
CPU	Central processing unit
DBN	Deep belief network
DDPG	Deep deterministic policy gradient
DQN	Deep Q network
DRL	Deep reinforcement learning
ETSI	European telecommunications standards institute
ETSI ISG NFV	The ETSI industry specification group for network functions virtualization
GCN	Graph convolutional neural network
IETF	Internet engineering task force
ILP	integer linear programming
InP	infrastructure provider
ISP	Internet service provider
LP	Linear programming
MILP	Mixed integer linear programming
ML	Mchine learning
MIQCP	Mixed integer quadratically constrained programming
NF	Network function
NFV	NF virtualization
NFVI	NFV infrastructure
NFV-MANO	NFV management and orchestration
NFVO	NFV orchestrator
NFV-RA	NFV resource allocation
OSS	Operations support system
QoS	Quality of service
RL	Reinforcement learning
SN	Substrate network
SFC	Service function chain
VIM	Virtualized infrastructure manager
VL	Virtual link
VNE	Virtual network embedding
VNF	Virtual network function
VNF-CC	VNF chain composition
VNF-FG	VNF forwarding graph
VNF-FGE	VNF-FG embedding
VNFM	VNF manager
VNF-SCH	VNF scheduling
VNO	Virtual network operator
VNS	Variable neighborhood search

The VNF-FGE is similar to the virtual network embedding (VNE) problem. VNE is a challenging resource allocation problem in the network virtualization. VNF-FGE can be regarded as a generalization of VNE. However, the differences between them are listed as follows:

- In VNF-FGE, the VNFs in VNF-FG may have different network functions, while virtual nodes in VNE perform

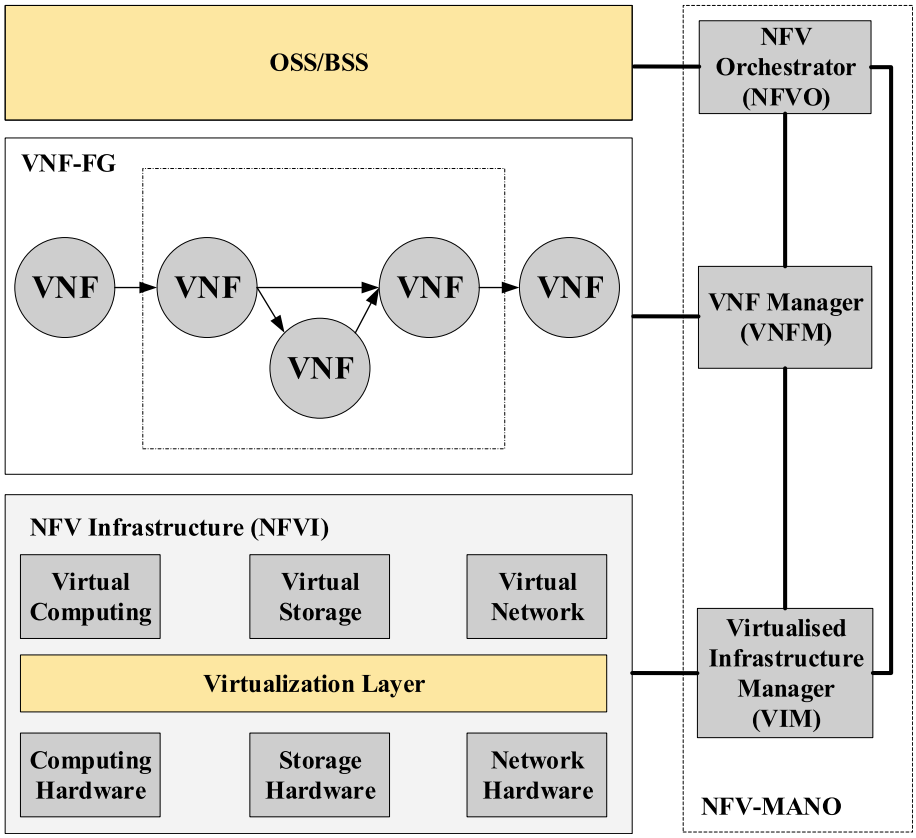


Fig. 2 NFV architecture

the same functions which can be deployed at any physical nodes.

- There may be a corresponding sequential relationship between some VNFs, namely a specific order has been predefined in VNF-FGE. Traffic needs to pass through the corresponding VNFs in this order. However, there is no order specifications between virtual nodes in VNE.
- Sometimes, several VNFs in a VNF-FG can be host by some virtual machines within the same server, while virtual nodes in VNE can only be placed on different physical nodes.

2.2 VNF-FGE resources

Multiple network service providers request resources in the form of VNF-FGs from the Internet service provider (ISP) which manages the physical network infrastructures. There are many kinds of resources requesting by VNF-FGs. We categorize them collectively according to their respective characteristics.

2.2.1 Node and link resources

As both SN and VNF-FGs are composed of nodes and links, we can divide the resources into node resources and link resources. Node resources are attributes that refer to servers and VNFs, such as CPU, memory, storage. The resources on the nodes may vary depending on the type of hosted functions. For instance, a memory server may have large storage resources, while a computationally intensive server may have abundant CPU resources. Link resources are attributes that refer to substrate links and virtual links (VLs), such as bandwidth, delay, and packet loss.

2.2.2 Primary and secondary resources

We divide the resources of nodes and links into primary and secondary resources regarding their mutability, accessibility, and interdependency with other resources. Primary resources are the intrinsic properties of the nodes or links themselves. These resources are independent of the state of other nodes or links but only on their utilization, such as central processing unit (CPU) and bandwidth. A VNF-FG can directly request these primary resources. Secondary resources are derived attributes that refer to nodes or links. These resources either depend on the states of their nodes or links or calculate from other resources, such as processing delay of a node, transmission delay of a link, and packet loss probability of them all. For example, the load of CPU of a node indicates the queue size at a router, which will influence the packet loss probability of

the node. The transmission and propagation delay of physical links with the processing and forward delay of physical nodes makes the transmission delay of a VL.

2.2.3 Consumable and Non-consumable resources

Consumable resources refer to the resources that will be consumed during the mapping process, such as the CPU of nodes and the bandwidth of links which will be consumed after the nodes and links have been embedded. Generally speaking, the bandwidth consumption of links are similar to the CPU consumption of nodes, which are fixed value. There are some VNF-FGE methods mentioning that the demanded bandwidth of a mapped VL might change over time. Some works considered it as a stochastic variable [12] while others appropriately shared it among different flows [13, 14]. Studying these issues requires separate analysis and is outside the scope of this work. In contrast with consumable resources, non-consumable resources (e.g., loss probability) are inherent properties in the SN, which are irrelevant to the substrate resources consumed by the VNF-FG requests during the mapping process.

2.3 Mathematical modeling

In general, the mapping process of VNF-FGs is to allocate enough resources on the SN to each VNF and link. In this case, the resources provided by the SN correspond to those required by VNF-FGs. For VNFs, we use many-to-one relationships to describe the mapping between VNFs and substrate nodes. More complex for VLs, the mapping between VLs and substrate links can be described as many-to-many relationships.

After finding the candidate substrate resources, we have to satisfy the resource demand represented by VNF-FG. For instance, a substrate link whose bandwidth is 100 Mbit/s cannot host a VL that requests 1000 Mbit/s bandwidth. Similarly, a substrate node can only host some VNF instances whose requested CPU resources are no more than the CPU resources provided by the substrate node. When it comes to the reliability issue, it may even need to reserve more resources. However, in the real-world environment, the substrate resources need to be used with careful calculation due to the strict budgeting. We need to optimize this process to embed the virtual resources into the substrate resources in a cost-efficient manner. Therefore, the problem of VNF-FGE can be described as Fig. 3. The general VNF-FGE mathematical modeling is shown as follows. The notations and descriptions of network model are provided in Table 2.

2.3.1 SN

We model an SN (also named physical network) as an edge-weighted undirected graph $G_s = (N_s, L_s)$, where N_s denotes the set of substrate nodes of G_s and L_s denotes the set of substrate links. We denote $u, v \in N_s$ as two substrate nodes, and $uv \in L_s$ as a substrate link. For substrate nodes, VNF instances which are running on the substrate nodes are also need discussing. We denote m as a VNF instance in the set of all VNF instances M, i.e., $m \in M$. The resources on the VNF instance are as same as node resources. In this paper, we only discuss three main resources on the substrate nodes, i.e., CPU, memory, and storage. They are denoted as $C_u^{cpu}, C_u^{mem}, C_u^{sto}$ of substrate node u, respectively. For substrate links, C_{uv}^{bw} is denoted as the bandwidth of uv. The available CPU, memory, storage resource ratios of node $u \in N_s$ and link $uv \in L_s$ are symbolized as a_u^{cpu}, a_u^{mem}, a_u^{sto}, and a_{uv}^{bw} respectively.

2.3.2 VNF-FG

We model each VNF-FG as an edge-weighted undirected graph $G_f = (N_f, L_f)$, where N_f denotes the set of VNFs and L_f denotes the set of VLs. We denote $\overline{u}, \overline{v} \in N_f$ as two VNFs, and $\overline{uv} \in L_f$ as a VL. For VNFs, they mainly request three network resources (e.g., CPU, memory and storage). The request for CPU, memory, and storage resources of VNF $\overline{u}$ in VNF-FG G_f are symbolized as $D_{f,\overline{u}}^{cpu}$, $D_{f,\overline{u}}^{mem}$, and $D_{f,\overline{u}}^{sto}$ respectively. We denote $D_{f,\overline{uv}}^{bw}$ as the required bandwidth of VL $\overline{uv}$ in VNF-FG G_f.

2.3.3 VNF-FGE

For the sake of description, we take a common application scenario (more details can be seen in Sect. 3) that is offline, single domain for example. In this case, the process of VNF-FGE is transformed into a mathematical model: given a set of VNF-FGs $G_{VNF-FG} = [G_1, G_2, \cdots, G_f, \cdots, G_F]$, and an SN G_s, we need to find appropriate nodes and links in SN to embed the VNF-FGs. In this mapping process, resource constraints need to be taken into account.

2.3.4 Resource constraints

To serve the VNF-FG G_f, its VNFs and its VLs all need to meet their resource constraints. For a VNF $\overline{u}$ in the VNF-FG G_f, it is successfully embedded when the corresponding substrate node has adequate resources. Therefore, we have:

$$\sum_{\overline{u} \in N_f} \Phi_{f,\overline{u}}^{u} D_{f,\overline{u}}^{cpu} \leq a_u^{cpu} C_u^{cpu}, \quad \forall u \in N_s, \tag{1}$$

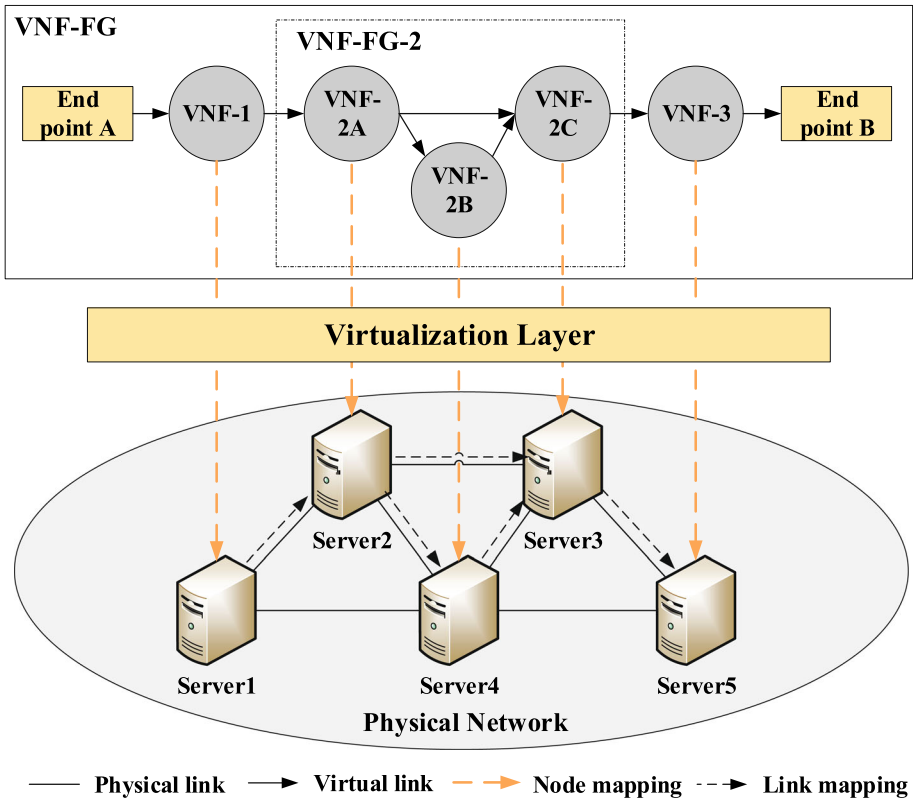


Fig. 3 An example of VNF-FG deployment

Table 2 Notations and descriptions of network model

Notation	Description
G_s	The substrate network.
N_s	The set of substrate nodes of G_s.
L_s	The set of substrate links of G_s.
G_f	The virtual network function forwarding graph.
N_f	The set of virtual network functions of G_f.
L_f	The set of virtual links of G_f.
u, v	Two substrate nodes.
$\bar{u}$, $\bar{v}$	Two virtual network functions.
uv	A substrate link.
$\bar{u}\bar{v}$	A virtual link.
$C_u^{cpu}, C_u^{mem}, C_u^{sto}$	The CPU, memory and storage resource of node u.
C_{uv}^{bw}	The bandwidth capacity of link uv.
a_u^{cpu}, a_u^{mem}, a_u^{sto}, a_{uv}^{bw}	The available resource ratios of node u and link uv.
$D_{f,\bar{u}}^{cpu}$, $D_{f,\bar{u}}^{mem}, D_{f,\bar{u}}^{sto}$	Three main resources requested by VNF $\bar{u}$ in VNF-FG G_f.
$D_{f,\bar{u}\bar{v}}^{bw}$	The required bandwidth of VL $\bar{u}\bar{v}$ in VNF-FG G_f.

$$\sum_{\bar{u}\in N_f} \Phi_{f,\bar{u}}^{u} D_{f,\bar{u}}^{mem} \le a_u^{mem} C_u^{mem}, \quad \forall u \in N_s, \tag{2}$$

$$\sum_{\bar{u}\in N_f} \Phi_{f,\bar{u}}^{u} D_{f,\bar{u}}^{sto} \le a_u^{sto} C_u^{sto}, \quad \forall u \in N_s, \tag{3}$$

where $\Phi_{f,\bar{u}}^{u}$ is a binary variable indicating $\Phi_{f,\bar{u}}^{\bar{u}} = 1$ if the VNF $\bar{u}$ in VNF-FG G_f is deployed at substrate node u, and 0 otherwise.

As for a VL $\bar{u}\bar{v}$ in the VNF-FG G_f, we should guarantee that the two VNF $\bar{u}$ and $\bar{v}$ which the VL connected are

embedded and the resources the VL requested are satisfied. For these constraints, we have:

$$\sum_{\bar{uv}\in L_f} \Phi_{f,\bar{uv}}^{uv} D_{f,\bar{uv}}^{bw} \le a_{uv}^{bw} C_{uv}^{bw}, \quad \forall uv \in L_s, \tag{4}$$

where $\Phi_{f,\bar{uv}}^{uv}$ is a binary variable indicating $\Phi_{f,\bar{uv}}^{uv} = 1$ if the VL $\bar{uv}$ in the VNF-FG G_f is deployed at the substrate link uv, and 0 otherwise. There are also some mapping constraints that we do not present, more detailed constraints can be seen in [15].

2.3.5 Structural constraints

In addition to the resource constraints, there are also some structural constraints that need to be satisfied. For each VNF, it should be deployed on one substrate node. Therefore, we have:

$$\sum_{\bar{u}\in N_f} \Phi_{f,\bar{u}}^{u} \le 1, \quad \forall \bar{u} \in N_f. \tag{5}$$

There are two constraints for the selected path. Firstly, let $I(u)$ and $O(u)$ denote the sets of incoming links and outgoing links of node u. A successive substrate path connecting two VNFs $\bar{u}$, $\bar{v}$ of virtual link $\bar{uv}$ should be satisfied:

$$\begin{aligned} &\sum_{uv\in O(u)} \Phi_{f,\bar{uv}}^{uv} - \sum_{vu\in I(u)} \Phi_{f,\bar{uv}}^{vu} = \Phi_{f,\bar{u}}^{u} \\ &- \Phi_{f,\bar{v}}^{u}, \quad \forall u \in N_s, \forall \bar{uv} \in L_f. \end{aligned} \tag{6}$$

Secondly, the path mentioned above should ensure avoiding the loop. Therefore, we have:

$$\sum_{uv\in O(u)} \Phi_{f,\bar{uv}}^{uv} - \sum_{vu\in I(u)} \Phi_{f,\bar{uv}}^{vu} \le 1, \quad \forall u \in N_s, \forall \bar{uv} \in L_f. \tag{7}$$

2.3.6 Objectives

The problem of VNF-FGE tends to be described as a multi-objective optimization problem. These objectives are always correlated and constrained. According to the characteristics of objectives of most existing works, we can divide the objectives into two categories. Table 3 lists the important notations and descriptions of objectives used in this paper.

The first category is to improve the overall QoS, including minimizing the end-to-end delay and balancing the workload.

(1) O_{11}: Minimize the end-to-end delay. As a most commonly discussed objective in QoS, delay is mainly defined in two ways. Agarwal *et al.* [16] and Wang *et al.* [17] both defined the delay DE_f^{total} as the delay of queuing, transmission and processing. The queuing delay refers to the delay of VNF-FG G_f in the service request queue. The processing delay contains the inherent processing delays of nodes and the dynamic processing delays of different VNFs (related to the traffic rate flowing through the VNF). Therefore, the total delay of VNF-FG G_f is defined as the sum of queuing delay, transmission delay, inherent delay and dynamic processing delay:

$$\begin{aligned} DE_f^{total} = d_f^{qd} &+ \sum_{uv\in L_s} \sum_{\bar{uv}\in L_f} \Phi_{f,\bar{uv}}^{uv} d_{uv}^{td} D_{f,\bar{uv}}^{bw} \\ &+ \sum_{u\in N_s} \sum_{\bar{u}\in N_f} \Phi_{f,\bar{u}}^{u} d_u^{id} \\ &+ \sum_{u\in N_s} \sum_{\bar{u}\in N_f} \Phi_{f,\bar{u}}^{u} DF_{K_{f,\bar{u}}}(R_{f,\bar{u}}), \end{aligned} \tag{8}$$

where d_f^{qd} denotes the queuing delay of the VNF-FG G_f, d_{uv}^{td} denotes the transmission delay of the substrate link $uv \in L_s$, d_u^{id} is the inherent delay of nodes $u \in N_s$, and $K_{f,\bar{u}}$ denotes the category (e.g., deep package inspection) of node $\bar{u}$ in VNF-FG G_f. $R_{f,\bar{u}}$ denote the traffic rate flowing through the VNF $\bar{u}$ in VNF-FG G_f. $DF_{K_{f,\bar{u}}}(R_{f,\bar{u}})$ is the dynamic processing delay function (related to $R_{f,\bar{u}}$) of a VNF whose category is $K_{f,\bar{u}}$.

There are also some studies (e.g., [15] and [18]) arguing that the delay on each entity which in a path can be distinguished separately. In this way, the total delay of VNF-FG G_f is defined as the sum of link delay, node delay, and instance delay:

$$\begin{aligned} DE_f^{total} = &\sum_{uv\in L_s} \sum_{\bar{uv}\in L_f} \Phi_{f,\bar{uv}}^{uv} d_{uv}^{ld} \\ &+ \sum_{u\in N_s} \sum_{\bar{u}\in N_f} \Phi_{f,\bar{u}}^{u} d_u^{nd} \\ &+ \sum_{m\in M} \sum_{\bar{u}\in N_f} \Phi_{f,\bar{u}}^{m} d_m^{fd}, \end{aligned} \tag{9}$$

where $\Phi_{f,\bar{u}}^{m}$ is a binary variable indicating $\Phi_{f,\bar{u}}^{m} = 1$ if the VNF $\bar{u}$ in VNF-FG G_f is deployed at the VNF instance m, and 0 otherwise. d_{uv}^{ld} denotes the delay of the link uv, d_u^{nd} denotes the delay of the node u, d_m^{fd} denotes the delay of the VNF instance m. Therefore, for a VNF-FGE problem, we can take the total delay of VNF-FG G_f into account to optimize the embedding process. In this way, the total delay of VNF-FG is mainly described as two formulations. This objective is to minimize the total delay of VNF-FG, which is shown as follows:

Table 3 Notations and descriptions of objectives

Notation	Description
DE_f^{total}	The total delay of the VNF-FG G_f.
d_f^{qd}	The queuing delay of the VNF-FG G_f.
d_{uv}^{td}	The transmission delay of the substrate link uv.
d_u^{id}	The inherent delay of nodes u.
$K_{f,\bar{u}}$	The category (e.g., deep package inspection) of node $\bar{u}$ in VNF-FG G_f.
$DF_{K_{f,\bar{u}}}(R_{f,\bar{u}})$	The dynamic processing delay function (related to $R_{f,\bar{u}}$) of a VNF whose category is $K_{f,\bar{u}}$.
d_{uv}^{ld}	The link delay of the link uv.
d_u^{nd}	The node delay of the node u.
d_m^{fd}	The VNF delay of the VNF instance m.
A^{cpu}	The average usage of CPU resources.
CO_f^{total}	The total cost of the VNF-FG G_f.
$c_u^{cpu}, c_u^{mem}, c_u^{sto}$	The unit costs of the resource of node u.
c_{uv}^{bw}	The unit cost of the resource of link uv.
c_u^{dc}	The cost of deploying the VNF on the node u.
CO_s^{total}	The total cost of the substrate network s.
c_m^{sc}	The setup cost of the VNF instance m.
c_m^{oc}	The operation cost of the VNF instance m.
c_m^{cc}	The unit communication cost of the VNF instance m.
R_m	The traffic rate flowing though the VNF instance m.
$E_{f,\bar{u}}^{u}$	The node incremental energy cost of a substrate node u when mapping a VNF $\bar{u}$ in VNF-FG G_f.
e_u^{be}	The baseline energy of node u without any CPU load.
e_u^{ce}	The unit energy cost of CPU utilization of node u.
$E_{f,\bar{u}\bar{v}}^{uv}$	The link incremental energy cost of a substrate link uv when mapping a VL $\bar{u}\bar{v}$ in VNF-FG G_f.
e_{uv}^{lc}	The unit energy cost of link uv.
$D_{f,\bar{u}\bar{v}}^{dis}$	The distance of uv for mapping VL $\bar{u}\bar{v}$ in VNF-FG G_f.
E^{sc}	The switching cost.
E_f^{total}	The total energy cost of the VNF-FG G_f.
$PR(t)$	The electricity price at time-slot t.
t_a, t_e	The arriving time and expiration time of the request.
Ψ_r	The acceptance ratio of the VNF-FG G_f.
m_u	The amount of VNF instances on substrate node u.

$$O_{11}: \min \quad DE_f^{total}. \tag{10}$$

(2) O_{12}: Balance the workload. Balancing the workload often refers to preventing a single server in ISPs from carrying too many VNFs to reduce QoS. For instance, given a mapping decision of an optimal algorithm, a server node in the mapping decision just meets its resource requirements while most of its adjacent nodes have a lot of resources left over. If the workload is not considered, the node will be put into use resulting in an inappropriate situation where the node is running at full load while most of its adjacent nodes are idle. Therefore, the relative factors can be weighed to prioritize the deployment of nodes with fewer used resources. The average usage of the resources is always applied to evaluate the workload. The resources tend to be CPU resources. Li *et al.* [19] defined system load balance as the variance of the physical node resource usage of all servers at a time. Therefore, we can minimize the variance of the CPU resources to balance the workload, which is given by: where A^{cpu}

$$O_{12}: \min \quad \sigma^2 = \sum_{u \in N_s} \frac{((1-a_u^{cpu})C_u^{cpu} - A^{cpu})^2}{|N_s|}, \tag{11}$$

$$A^{cpu} = \sum_{u \in N_s} \frac{(1-a_u^{cpu})C_u^{cpu}}{|N_s|}, \tag{12}$$

denotes the average usage of CPU resources of SN.

The second category is to increase the economic gain, including minimizing the cost, maximizing the acceptance ratio, and minimizing the number of VNF instances.

(1) O_{21}: Minimize the cost. As the most commonly discussed objective in economic interests, the cost is mainly defined in two ways. For the first one, they considered minimizing the cost of each VNF-FG to minimize the total cost. Wang *et al.* [17] defined the total cost CO_f^{total} of VNF-FG G_f as deployment cost and resource utilization cost. The deployment cost is the cost of deploying the VNFs on nodes. The resource utilization cost includes the CPU, memory, and storage cost of the nodes, and the bandwidth cost of the links. For the sake of simplicity, we ignore their proposed setting that node resources are related to the flow through the node and the category of the node. For more details, please refer to [17]. In this case, the total cost is the sum of resource utilization cost, including node and link resources, and the deployment cost:

$$\begin{aligned} CO_f^{total} = & \sum_{u \in N_s} \sum_{\bar{u} \in N_f} \Phi_{f,\bar{u}}^{u} (c_u^{cpu} D_{f,\bar{u}}^{cpu} \\ & + c_u^{mem} D_{f,\bar{u}}^{mem} + c_u^{sto} D_{f,\bar{u}}^{sto}) \\ & + \sum_{uv \in L_s} \sum_{\bar{u}\bar{v} \in L_f} \Phi_{f,\bar{u}\bar{v}}^{uv} c_{uv}^{bw} D_{f,\bar{u}\bar{v}}^{bw} \\ & + \sum_{u \in N_s} \sum_{\bar{u} \in N_f} \Phi_{\bar{u}}^{u} c_u^{dc}, \end{aligned} \tag{13}$$

where $c_u^{cpu}, c_u^{mem}, c_u^{sto}$ denote the unit costs of the resource of node u, and c_{uv}^{bw} denotes the unit cost of the resource of link uv. c_u^{dc} is the cost of deploying the VNF on the node u.

For the second one, there are some works to characterize cost in detail by dividing nodes (servers or VNF instances) into various states. In this case, minimizing the total costs is achieved by minimizing the cost of the whole SN. Gu *et al.* [20] argued that a VNF instance has the active state and inactive state. Therefore, they introduced a binary variable to represent such activation status. In order to unify the format, we denote the binary variable as following:

$$\phi_m(t) = \begin{cases} 1, & \text{if the VNF instance} m \text{is activated at time slot } t, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

They defined the cost CO_s^{total} as setup, operation and communication cost. The setup cost is the cost of transiting a VNF instance from the inactive state into the active state. The operation cost is the cost of running an active VNF instance, and the communication cost is the cost of transferring the traffic flow from the servers hosting its parent VNF instance. Therefore, the total cost is defined as the sum of setup cost, operation cost and communication cost:

$$\begin{aligned} CO_s^{total} = & \sum_{m \in M} max\{\phi_m(t) - \phi_m(t-1), 0\} c_m^{sc} \\ & + \sum_{m \in M} \phi_m(t) c_m^{oc} \\ & + \sum_{m \in M} \phi_m(t) c_m^{cc} R_m, \end{aligned} \tag{15}$$

where c_m^{sc} denotes the setup cost of the VNF instance m. c_m^{oc} is the operation cost of the VNF instance m and c_m^{cc} is the unit communication cost of the VNF instance m. The traffic rate flowing though the VNF instance m is denoted as R_m.

Node state is also vital in energy models while energy cost is rarely modeled in detail in the VNF-FGE problem. Here, we draw on a detailed energy modeling from a virtual network embedding (similar to VNF-FGE) problem. More details can be seen in [21] [22] and [23]. Su *et al.* [21] denoted the state of a node as an inactive state and an active state and gave an energy cost modeling including node incremental energy cost, link incremental energy cost, and switching cost. They achieved the goal of minimizing the energy cost by minimizing the incremental energy cost of mapping a request. Many works have presented that the full-system average energy cost of a node is approximately linear with CPU utilization [24] [25]. The energy cost of other resources (e.g., memory and storage) can be negligible [26]. Therefore, the node incremental energy cost $E_{f,\bar{u}}^{u}$ of a substrate node u when mapping a VNF $\bar{u}$ in VNF-FG G_f can be described as:

$$E_{f,\bar{u}}^{u} = \begin{cases} e_u^{be} + e_u^{ce} D_{f,\bar{u}}^{cpu}, & \text{if the node } u \text{ is inactivated,} \\ e_u^{ce} D_{f,\bar{u}}^{cpu}, & \text{otherwise,} \end{cases} \tag{16}$$

where e_u^{be} is the baseline energy of node u without any CPU load, and e_u^{ce} denotes the unit energy cost of

CPU utilization of node u.

As for the link incremental energy cost $E^{uv}_{f,\bar{u}\bar{v}}$ of a substrate link uv when mapping a VL $\bar{u}\bar{v}$ in VNF-FG G_f, it is set to be linear with the traffic rate and the distance of the link uv based on [27],

$$E^{uv}_{f,\bar{u}\bar{v}} = e^{lc}_{uv} D^{dis}_{f,\bar{u}\bar{v}} \frac{D^{bw}_{f,\bar{u}\bar{v}}}{C^{bw}_{uv}}, \tag{17}$$

where e^{lc}_{uv} is the unit energy cost of link uv, and $D^{dis}_{f,\bar{u}\bar{v}}$ denotes the distance of uv for mapping VL $\bar{u}\bar{v}$ in VNF-FG G_f. The switching cost is set to be a constant value E^{sc}, which is a one-time energy cost for transiting the node from the inactive state into the active state. Therefore, the total energy cost E^{total}_f can be described as the sum of node cost, link cost and the cost of node state change:

$$\begin{aligned} EN^{total}_f = & \sum_{u\in N_s}\sum_{\bar{u}\in N_f} \Phi^u_{\bar{u}} E^u_{f,\bar{u}} \int_{t_a}^{t_e} PR(t)\,dt \\ & + \sum_{uv\in L_s}\sum_{\bar{u}\bar{v}\in L_f} \Phi^{uv}_{\bar{u}\bar{v}} E^{uv}_{f,\bar{u}\bar{v}} \int_{t_a}^{t_e} PR(t)\,dt \\ & + \sum_{u\in N_s}\sum_{\bar{u}\in N_f} \Phi^u_{\bar{u}} (1-\phi_u) E^{sc} \int_{t_a}^{t_e} PR(t)\,dt, \end{aligned} \tag{18}$$

where $PR(t)$ denotes the electricity price at time-slot t. t_a and t_e denote the arriving time and expiration time of the request respectively. ϕ_u is a binary variable indicating $\phi_u = 1$ if the node is in an active state, and 0 otherwise. Therefore, O_{21} may be minimizing one of the three types of cost, i.e.,

$$O_{21}: \min \quad CO^{total}_f \ or \ CO^{total}_s \ or \ EN^{total}_f. \tag{19}$$

(2) O_{22}: Maximize the acceptance ratio. Many requests will arrive at the system with different resource constraints. When a system cannot serve the request, the request will be rejected. As for an SFC, it is usually regarded as a simplified straight chain of VNF-FG. Hence, the requesting bandwidth (always described as traffic rate in SFC) for the VLs in an SFC is all the same while the required bandwidth of VL in VNF-FG can be different. In this case, the acceptance ratio of SFC is always the accepted traffic rate or the ratio between the accepted traffic rate and all the traffic rate or the amount of SFCs arriving the system. As for VNF-FG, the acceptance ratio is always the amount of VNF-FGs that could be served. Therefore, maximizing the acceptance ratio can be described as following:

$$O_{22}: \max \quad \Psi_r = \sum_{G_f \in G_{VNF-FG}} \phi_f, \tag{20}$$

where ϕ_f is a binary variable indicating $\phi_f = 1$ if the VNF-FG G_f is accepted, and 0 otherwise.

(3) O_{23}: Minimize the number of VNF instances mapped on the infrastructure (i.e., M).

$$O_{23}: \min \quad M = \sum_{u\in N_s} m_u, \tag{21}$$

where m_u denotes the amount of VNF instances on substrate node u.

3 Application scenarios

In this section, a categorization of VNF-FGE is given from the perspective of application scenarios. Firstly, we briefly introduce online and offline scenarios in Sect. 3.1, then single-domain and multi-domain scenarios are presented in Sect. 3.2. In Sect. 3.3, centralized and distributed scenarios are discussed. Finally, we elaborate on the dynamic and static scenarios in Sect. 3.4.

3.1 Offline/Online

The VNF-FGE problem can be either offline or online. In the offline scenario, VNF-FGs are supposed to be known in advance [28–31]. For example, Cao et al. [30] proposed two frameworks based on multi-objective genetic algorithm and improved non-dominated sorting genetic algorithm. As the VNF-FG requests were assumed to be known, they encoded all the node mappings in binary and iteratively carried out selection, crossover, and mutation on the initial population to obtain the solution. Kuo et al. [31] sorted the demand of all VNF-FGs according to the stress testing, then each VNF-FG is deployed sequentially based on the relation between link and server usages.

In the online scenario, VNF-FG requests could arrive and depart in the system at any time. Particularly in the B5G/6G networks, the service requests of users are more dynamic. So the requirement for solving VNF-FGE in an online manner is more urgent, and many recent works have focused on this scenario [32–34]. For instance, Wang et al. [34] formulated an ILP model but cannot be applied to the online scenario of VNF-FGE directly. Therefore, they used a metaheuristic search method by sacrificing the precision to obtain a near-optimal solution online. In this case, the run time is crucial for algorithms. In the online scenario, the current VNF-FG requires to be appropriately embedded into the SN within a reasonable time tolerance before subsequent requests arrive.

3.2 Single-domain/Multi-domain

In general, VNF-FGs are requested to make use of the resources of SN provided by one ISP. In this case, the SN is provided and controlled by one operator. So the server specifications, the virtualization technology of the virtual machine, and the virtual network functions are of higher uniformity. It enables the embedding process of the VNF-FG to be more concise. In most existing works [35–37], the VNF-FGE problem is supposed to be single-domain. Jang et al. For instance, [35] formulated the problem of embedding multiple VNF-FGs in an NFV-enabled network. They jointly considered the objective of maximizing the acceptable flow rate and minimizing the energy cost. To this end, they transformed the multi-objective optimization problem into a single-objective mixed integer linear programming (MILP) problem and proposed a linear relaxation and rounding based algorithm to obtain an approximate optimal solution.

The SNs could also be comprised of multiple domains that are controlled by different ISPs. We call it the multi-domain scenario. Soares et al. [38] tried to balance the load among different domains within the constraint of the location and proposed a strategy based on integer linear programming to tackle this problem. However, different ISPs involved in VNF-FG would like to keep the information about their resources confidential. Quang et al. [32] proposed a distributed framework copying the local view of each domain orchestrator to other orchestrators resulting in a global view of the multiple domains to obtain the global optimal VNF-FG embedding. In this case, the information may not be confidential, not to mention the heavily increased complexity. Zhang et al. [39] avoided replicating global information for building a control channel to cooperation among the multiple domains.

For the sake of security and commercial sensitivity of each domain, Quang et al. [40] proposed a deep reinforcement learning framework under the multi-domain non-cooperative environment. As shown in Fig. 4, each domain has no topological or resource information of the other domains and cannot communicate with others. It proposed the prices of resources to the client who demands for the VNF-FG embedding and the client decides where to deploy so as to minimize its deployment cost.

3.3 Centralized/Distributed

When the application scenario falls into the centralized one, it indicates that there will be one entity to compute the VNF-FGE. This entity can be a dedicated machine or a general node server. For instance, Sun et al. [41] applied reinforcement learning to solve the VNF-FGE problem in dynamic networks. They proposed a reinforcement learning module to output alternative paths and a load balancing module to pick the optimal solution from them. This algorithm was always executed on a single computer and could get global information about the embedding process. However, for a single entity, it would suffer from a single point of failure. Moreover, when it comes to the scalability issue in heterogeneous and complex B5G/6G networks, a centralized entity may be overwhelmed by the massive number of VNF-FGs to handle. Conversely, under the distributed scenario, multiple entities are utilized to perform the VNF-FGE. These entities can be some distributed participating entities or maybe some dedicated distributed machines. Leivadeas *et al.* [32] split the various functions of an SFC into a set of partitions to solve the problem. Because of the multiple entities, this approach can share the workload and achieve higher scalability. However, multiple entities need to cost additional expenses to synchronize the information. So in this situation, a trade-off between the communication cost and the performance of the embedding needs to be considered.

3.4 Static/Dynamic

Most existing papers [30, 42, 43] are based on static VNF-FGs to solve the VNF-FGE problem. For example, Li et al. [43] proposed a merge-split viterbi algorithm to solve the static VNF-FGE problem. They found a basic global solution and continuously optimized this solution through the improvement procedures to approximate the optimum. The VNF-FGs stay static once they arrive at the system and do not readjust the mapping positions and requesting resources of the VNFs and the VLs in VNF-FGs after embedding. Without considering the possibility of recomposing and remapping of VNF-FGs, the VNF-FGE problem may obtain a feasible but non-adaptive strategy. In the B5G/6G environment, the wider network coverage and lower network latency render more and more users access their services anytime and anywhere. On the one hand, the VNF-FG requests from users may be more dynamic. In this case, the dynamic characteristic is mainly embodied in the change of accessing positions or the structure of VNF-FG. Particulary, the varying structure of VNF-FG can be attributed to the changing service type in which several VNFs need recomposition or the dynamic traffic requirement over time [44, 45]. For example, Pei et al. [18] considered the dynamic network load and proposed a VNF placement algorithm based on double deep Q network. On the other hand, the deployment of the previously embedded VNF-FGs may need to make some adjustments for some reasons (e.g., server failure or efficiency improvement). Therefore, the deployment of VNF-FG may need to be dynamic and adaptive in the above settings. As shown in

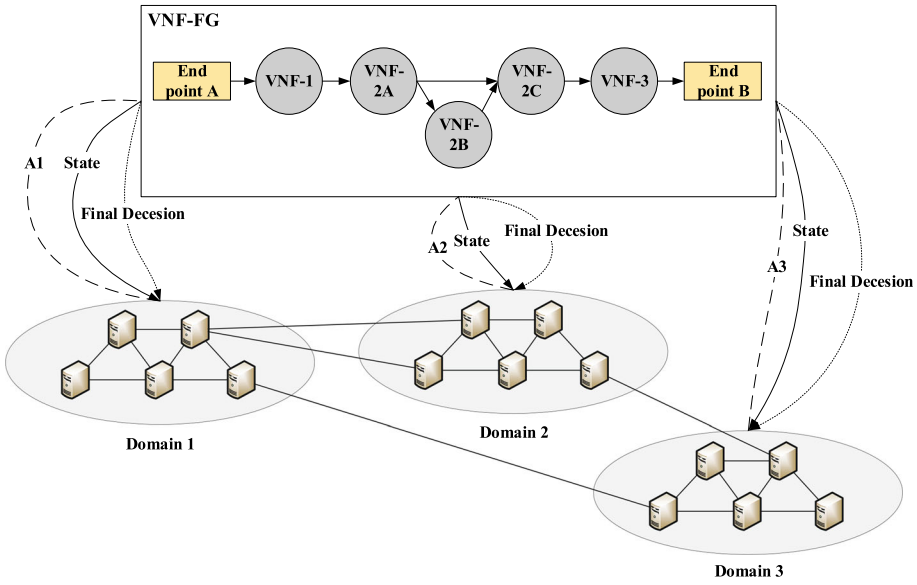


Fig. 4 Non-cooperative VNF-FG embedding framework under multiple domains

Fig. 5, a VNF-FG is simplified as a request of two VNFs connected by one VLs and each server is assumed to be used as an access point (i.e., wired or wireless) for the users. Fig. 5(a) shows a VNF-FG deployment decision at the previous service time, where User1 accesses the network from Server1. The service provider deploys a VNF-1 on Server1 and VNF-2 on Server5. It then finds that a new User2 joins at Server2 and requests a VNF-FG as VNF-1 to VNF-3, and User1 changes its access point to Server2. Hence, as shown in Fig. 5(b), the service provider decides to migrate the VNF-1 to Server2 where it can be efficiently shared by User1 and User2, and deploys a VNF-3 on Server3 for User2.

It indicates that solving the VNF-FGE problem dynamically is a noteworthy issue. Liu et al. [46] mainly focused on solving the first type of problem (i.e., the dynamic requests of users). They proposed a column generation based algorithm to solve the dynamic VNF-FGs deployment and readjustment problem to seek a trade-off between the resource consumption and operational overhead. Tajiki et al. [33] had considered both of these dynamics. They presented a resource allocation architecture that divided the mapping process into several processes to handle the dynamics separately.

4 VNF-FGE optimization approaches

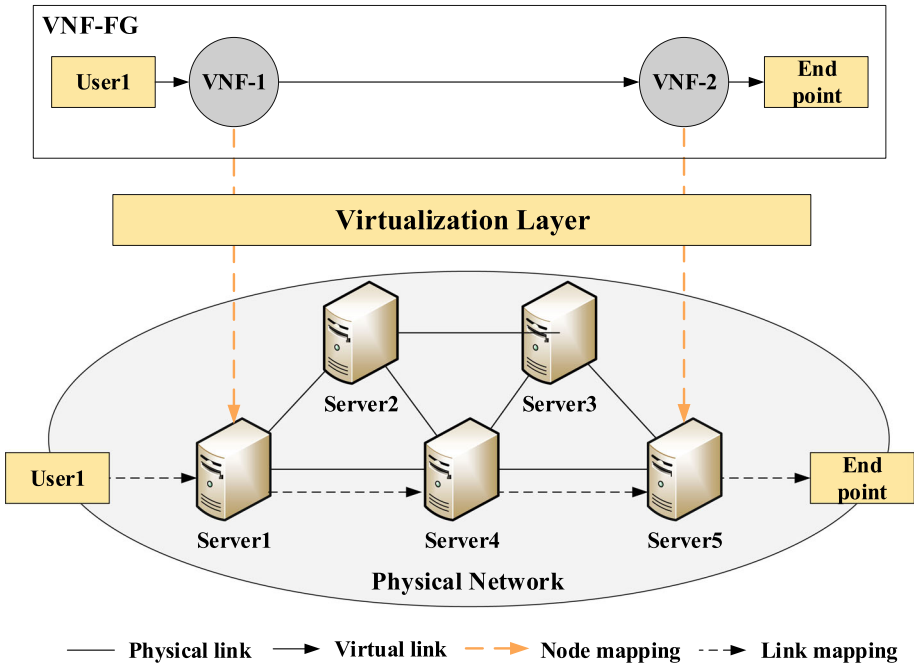
In this section, we focus on the optimization approaches of VNF-FGE in the existing works. Firstly, we introduce the exact approaches in Sect. 4.1, and then the heuristics-based approaches are presented in Sect. 4.2. Afterwards, we give the description of metaheuristics-based approaches in Sect. 4.3. In the end, due to the rapid development of machine learning (ML), it is expected that 6G networks will have much higher intelligence than their predecessors. So we elaborate on the ML-based approaches in Sect. 4.4.

4.1 Exact approaches

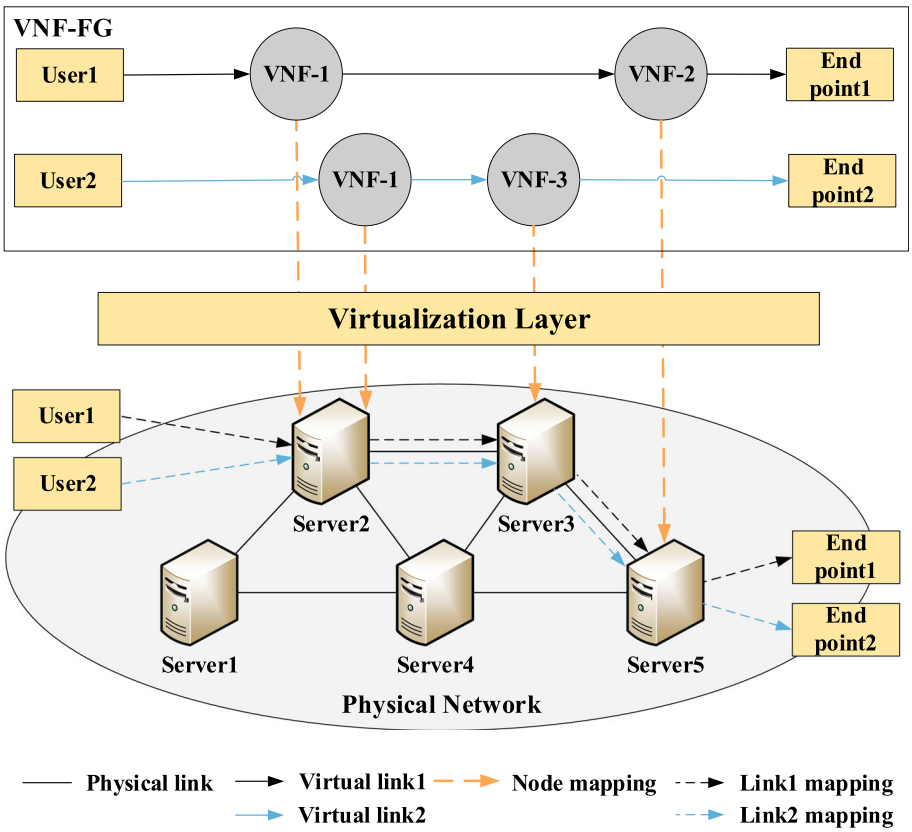
There are several works that utilize the exact approaches to obtain the optimal solution for small instances of the problem. We can divide these works into different categories according to their mathematical model. The commonly used mathematical models are integer linear programming (ILP) [28, 36, 47–52], binary integer programming (BIP) [53], mixed integer linear programming (MILP) [54–57] and mixed integer quadratically constrained programming (MIQCP) [58]. These approaches usually employ open source solvers (lpsolver) or commercial solvers (e.g., IBM ILOG CPLEX optimization) to calculate the optimal embedding result.

Soualah et. al. [36] formulated the VNF-FGE problem as an ILP model to maximize resource usage. Mijumbi et. al. [53] modeled the VNF-FGE as a BIP problem with the objective to minimize the cost. Marotta et. al. [54] modeled the VNF placement and routing problem as a MILP to minimize the power consumption of NFV infrastructure. Mehraghdam et. al. [58] modeled the VNF-FGE problem as a MIQCP problem with the objective of maximizing remaining data rate and minimizing total latency over all paths. More details of these exact approaches can be seen in Table 4.

Although the exact approaches can always obtain the optimal solution, they can only be solved in a small-scale network scenario. When it comes to a network with hundreds or even more nodes and links, these approaches are computationally prohibited. Bari et al. [47] applied CPLEX



(a) Deployment decision at the previous service time



(b) Deployment decision at the current service time

Fig. 5 Example on dynamic VNF-FG deployment and readjustment

to solve their ILP model in two types of networks (i.e., Internet2 research network (12 nodes, 15 links), and a university data center network (23 nodes, 42 links).). The average execution time of CPLEX in the data center is approximately 45 times the average execution time of CPLEX in Internet2.

4.2 Heuristics-based approaches

A variety of heuristic-based approaches are proposed in the existing works, which are usually designed based on the experience or whimsical ideas of the authors. Heuristic-based algorithms usually leverage some tricks to reduce the search space. These solutions are generally effective and could be implemented in real scenarios.

There are some studies [29, 43, 44, 47, 60] employing a multi-stage graph algorithm to solve the VNF-FGE problem. For example, Tastevin et al. [29] modeled the nodes of the service function chain and their relationships as a multi-level graph. They implemented the Viterbi algorithm on this multi-stage graph to obtain the optimal deployment of the VNF-FG. Li et al. [43] improved the Viterbi algorithm by merging and splitting the VNF instance which can take full advantage of the substrate resources. There are also some works [16, 31–33, 35, 61, 62] focusing on applying the relaxation method to solve the VNF-FGE problem. Agarwal et al. [16] converted a non-convex problem into a convex problem by replacing some variables to get some feasible solutions for finding an approximate optimal solution. Quang *et al.* [32] relaxed the complex model to get a feasible solution as the approximate optimal solution to the problem. Other works [63–65] proposed algorithms based on graph theory or dynamic programming to solve the problem. Ishigaki *et al.* [63] partitioned the correlation of VNFs and designed an algorithm based on dynamic programming to map VNFs into SN. More details of these heuristic-based approaches can be seen in Table 5.

4.3 Metaheuristics-based approaches

Metaheuristic algorithm is an improvement of the heuristic algorithm, which combines the randomized algorithm and local search algorithm. It mainly includes Tabu search algorithm, simulated annealing algorithm, genetic algorithm, particle swarm optimization algorithm, artificial fish swarms algorithm, etc.

Genetic algorithm is a widely adopted method to address the VNF-FGE problem [30, 42, 66, 67]. Kim et al. [66] firstly computed the shortest path and then mapped VNF-FG applying genetic algorithm with the objective of minimizing energy consumption. Khebbache *et al.* [67] proposed a non-dominated sorting genetic algorithm-II. It could achieve a Pareto optimal solution. These works demonstrated that the application of genetic algorithms could be alternatives to the combinatorial optimization

Table 4 Exact optimization-based approaches

Method	Reference	Contribution
LP	[59] Jang et al. (2016)	Determined the amount of flows assigned to each link and VNF instance to efficiently utilize the network resource.
ILP	[28] Li et al. (2015)	Considered hardware consumption.
	[36] Soualah et al. (2018)	Considered multiple criteria.
	[47] Bari et al. (2015)	Added a pseudo-switch to simplify the constraint.
	[48] Luizelli et al. (2015)	Pruned the search space to reduce complexity.
	[49] Moens et al. (2014)	Firstly formalized as an optimization problem.
	[50] Riggio et al. (2015)	Firstly formulated the VNF placement problem for radio access networks.
	[51] Sahhaf et al. (2015)	Applied service decompositions to refine abstract network functions.
	[52] Jahromi et al. (2018)	Considered the reuses and migrations.
BIP	[53] Mijumbi et al. (2016)	Traded solution simplicity and enhanced computation time for better resource management.
MILP	[54] Addis et al. (2015)	Present an analysis on egacy traffic engineering (TE) ISP goals and novel combined TE-NFV goals.
	[55] Marotta et al. (2016)	Considered uncertainty on the VNF resource demands.
	[56] Lin et al. (2016)	Firstly provided a techno-economic analysis on a consolidated design and provision scheme for VNF instance allocation and traffic routing in optical networks.
	[57] Ghaznavi et al. (2016)	Deployed the VNF-FG in a distributed and resource efficient manner.
MIQCP	[58] Mehraghdam et al. (2014)	Formalized with respect to data rate, number of used network nodes, and latency.

Table 5 Heuristics-based approaches

Method	Reference	Contribution
Multi-stage graph	[29] Tastevin Contribution (2017)	Aimed at being easily extensible.
	[43] Li Contribution (2018)	Did not take the iterative deployment strategy.
	[44] Pei Contribution (2019)	Dynamically adjusted the VNF instances.
	[60] Khebbache Contribution (2017)	Added a fictitious arc for VNF-FG to transform the problem.
Relaxation	[16] Agarwal Contribution (2019)	Decoupled the problem to reduce the complexity.
	[31] Kuo Contribution (2018)	Proposed a stress test to find a proper link and server relation.
	[32] Quang Contribution (2019)	Proposed a centralized and decentralized algorithm.
	[33] Tajiki Contribution (2019)	Defined various scenarios and different heuristic algorithms.
	[35] Jang Contribution (2017)	Applied a weakly pareto-optimal solutions to the problem.
	[61] Tajiki Contribution (2018)	Introduced an adaptive approach.
	[62] Jia Contribution (2018)	Exploited a software defined time evolving graph.
Dynamic programming	[63] Ishigaki Contribution (2019)	Introduced joint correlation-aware VNFs.
Minimum k-cut	[64] Yang Contribution (2016)	Defined a metric to quantize traffic intensity and dependency.
Graph partitioning	[65] Leivadeas Contribution (2017)	Proposed a graph partitioning algorithm.

problem which could provide various high-quality mapping solutions with low-complexity.

Other literature on metaheuristic-based algorithms is scattered. Wang et al. [34] sacrificed part of the accuracy to get an approximate optimal solution based on Tabu search. Li et al. [28] developed a simulated annealing algorithm to solve the problems. Luizelli et al. [68] combined variable neighborhood search (VNS) with mathematical programming to broaden the exploration of the space to obtain decent solutions. Table 6 shows more details of these metaheuristic-based approaches.

The generality of metaheuristic algorithms enables them to perform well when facing variants of the VNF-FGE problem. Unlike heuristic-based algorithms, even the constraints and objectives have changed, most of these metaheuristic-based algorithms could adjust to adapt to the variants of the problem.

4.4 Machine learning-based approaches

The application of ML in the VNF-FG placement problem is mainly based on reinforcement learning (RL) [69]. For instance, Kim et al. [70] balanced the workload on the node by rewarding the action of deploying VNFs on nodes with low resource used. Sun et al. [41] used an RL algorithm called Q-learning to record the rewards and strategies in a matrix. In RL, a learning agent interacts with its environment to get some information (i.e., variation in traffic type, network configuration). The agent performs actions, observes the rewards or penalties, and learns to perform the optimal action for each state [71]. Thus, a set of state-action-rewards tuples will be generated by RL for training.

Because of the difficulty in dealing with the large-scale action space of the VNF-FGE problem, a deep neural network is applied into the RL, forming a deep reinforcement learning (DRL) framework to assist in solving the VNF-FGE problem. A number of works [17, 18, 20, 37, 40, 72–74] utilized or improved the DRL algorithms such as deep deterministic policy gradient (DDPG) and deep Q network (DQN) algorithms to solve the VNF-FGE problem. Quang *et al.* [37] modelled the VNF-FGE problem as a Markov decision process [71] and formulated the descriptions of VNF-FGs as the states, the mapping of a VNF-FG and the SN as the actions, and the acceptance ratio as the reward. They improved the DDPG with a heuristic algorithm to explore the action space more efficiently and used multiple critic network method to accelerate the learning process. Pan et al. [73] first utilized a graph convolutional neural network (GCN) to approximate the value function in DRL framework. Gu et al. [74] proposed a model-assisted DRL framework for DDPG to solve the VNF-FGE problem. This framework can be built upon any DRL algorithms to guide more efficient training by auxiliary information. The evaluation result demonstrated its high performance in terms of convergence speed, performance, and efficiency. Pei et al. [18] formulated the problem of VNF-FGE as a BIP model with the objective of minimizing the cost, which took the VNF placement cost, VNFI running cost, and penalty of SFCs rejection into account. They proposed an algorithm based on double DQN to intelligently and efficiently solve the problem.

There are also some researches applying other ML methods to address the problem. Pei *et al.* [75] applied a deep belief network (DBN) to abstract and learn the

Table 6 Metaheuristics-based approaches

Method	Reference	Contribution
Genetic algorithm	[30] Cao et al. (2017)	Proposed a greedy non-dominated sorting genetic algorithm.
	[42] Ruiz et al. (2020)	Jointly solved the problems of VNF placement, VNF chaining and virtual topology design.
	[66] Kim et al. (2017)	Rearranged the locations or reconfigure the paths.
	[67] Khebbache et al. (2018)	Coped with problem for large instances.
Tabu Search	[34] Wang,et al. (2017)	Considered the total bandwidth consumption.
VNS	[68] Luizelli et al. (2017)	Combined mathematical programming and a meta-heuristic method.

features. Their subsequent work [15] improved the algorithm by designing two types of DBN networks. More details of the papers can be seen in Table 7.

5 Taxonomy of VNF-FGE approaches

In this section, combining the classifications in the previous three sections, we give a comprehensive taxonomy of VNF-FGE approaches in Table 8. The first column provides the optimization approaches and the second column presents the references of papers. The third column provides the information about the application scenario which is described as: [On|Off]/[S|M]/[C|D]/[St|Dy]. On and Off denote the online and offline respectively, S and M denote whether the problem is solved under the single-domain or the multi-domain environment. C and D denote whether the algorithm is centralized or distributed. St and Dy denote the static and dynamic respectively. The fourth column provides the optimization objectives mentioned in the Sect. 2.3.6. As B5G/6G networks pursue the low-latency service, we add the execution times of algorithms in the last column.

6 Emerging research challenges

In order to meet more complex and personalized requirements of users in B5G/6G networks, VNF-FGE problem is also confronted with some new challenges. In this section, we summarize some research directions that may be promising in the future. Firstly, we introduce the coordination of NFV-RA stages in Sect. 6.1. We offer several new objectives in Sect. 6.2. We present a further discussion of distributed VNF-FGE in Sect. 6.3. We give a new direction in prediction of VNF-FGs in Sect. 6.4. In the end, due to the rapid development of artificial intelligence, applying them into the network optimization is also a prospective research direction, which is elaborated in Sect. 6.5.

Table 7 Machine learning-based approaches

Method	Reference	Contribution
DBN	[15] Pei et al. (2020)	Designed two networks based on DBN to respectively solve VNF selection and chaining problem.
	[75] Pei et al. (2018)	Proposed a deep learning-based strategy based on DBN.
Q-learning	[41] Sun et al. (2018)	Proposed optimized Q-learning training algorithm.
	[70] Kim et al. (2017)	Constructed the entire problem as an RL model.
DRL-DDPG	[17] Wang et al. (2019)	Considered the resource consumption cost and service delay.
	[20] Gu et al. (2019)	Firstly applied DRL for joint optimization of VNF orchestration and flow scheduling.
	[37] Quang et al.et al. (2019)	Enhanced exploration of the DDPG.
	[40] Quang et al. (2019)	Considered VNF-FG deployment over non-cooperative multiple domains.
	[72] Quang et al. (2020)	Increased the number of critic networks in DDPG.
	[74] Gu et al. (2020)	Model-assisted DRL framework to accelerate the convergence.
DRL-DQN	[18] Pei et al. (2020)	Considered VNF placement cost, VNF running cost, and penalty of SFCs rejection.
	[73] Pan et al. (2020)	Employed GCN to approximate the value function.

Table 8 Taxonomy of VNF-FGE approaches

Optimi-zation	Reference	Category	Objective	Execution time
Exact	[59] Jang et al. (2016)	On/S/C/St	O_{21}	Not mentioned
	[49] Moens et al. (2014)	On/S/C/St	O_{21}	Affordable for small scale
	[52] Jahromi et al. (2018)	On/S/C/Dy	O_{21}	Affordable for small scale
	[55] Marotta et al. (2016)	On/S/C/St	O_{21}	Not mentioned
	[58] Mehraghdam et al. (2014)	On/S/C/St	O_{21} and O_{22}	Affordable for small scale
Heuristic	[16] Agarwal et al. (2019)	On/S/C/St	O_{11}	Not mentioned
	[31] Kuo et al. (2019)	Off/S/C/Dy	O_{22}	Affordable
	[35] Jang et al. (2017)	On/S/C/Dy	O_{21} and O_{22}	In polynomial time
	[43] Li et al. (2018)	Off/S/C/St	O_{21}	Affordable
	[44] Pei *et al.* (2019)	On/S/C/Dy	O_{11}*and*O_{21}	Affordable
	[60] Khebbache et al. (2017)	On/S/C/St	O_{21}	In polynomial time
	[63] Ishigaki et al. (2019)	On/S/C/St	O_{22}	Not mentioned
	[64] Yang et al. (2016)	On/S/C/St	O_{21}	Not mentioned
	[65] Leivadeas et al. (2017)	Off/S/D/St	O_{21}	Affordable
Exact and heuristic	[29] Tastevin et al. (2017)	Off/S/C/St	O_{21}	Affordable
	[32] Quang et al. (2019)	On/(S/M) /(C/D)/Dy	O_{21} and O_{22}	In polynomial time
	[33] Tajiki et al. (2019)	(On/Off)/S /C/(St/Dy)	O_{21}	Affordable
	[36] Soualah et al. (2018)	On/S/C/Dy	O_{21}	Unaffordable
	[43] Li et al. (2018)	Off/S/C/St	O_{21}	Affordable
	[47] Bari *et al.* (2015)	On/S/C/St	O_{21}	Affordable
	[48] Luizelli et al. (2015)	On/S/C/St	O_{23}	Affordable
	[50] Riggio et al. (2015)	Off/S/C/St	O_{21}	Affordable
	[51] Sahhaf et al. (2015)	On/S/C/St	O_{21}	Affordable
	[53] Mijumbi et al. (2016)	Off/S/C/St	O_{11} and O_{21}	Affordable
	[54] Addis et al. (2015)	Off/S/C/St	O_{21} and O_{23}	Affordable
	[56] Lin et al. (2016)	On/S/C/St	O_{21}	Affordable
	[57] Ghaznavi et al. (2016)	Off/S/C/St	O_{21}	Not mentioned
	et al.[61] Tajiki *et al.* (2018)	On/S/C/Dy	O_{21}	In polynomial time
	[62] Jia et al. (2018)	On/S/C/Dy	O_{21}	In polynomial time
Exact and metaheuristic	[28] Li et al. (2015)	Off/S/C/St	O_{21}	Affordable
Meta-heuristic	[30] Cao et al. (2017)	Off/S/C/St	O_{21}	Not mentioned
	[34] Wang,et al. (2017)	On/S/C/St	O_{21} and O_{22}	Not mentioned
	[42] Ruiz et al. (2020)	Off/S/C/St	O_{22}	Not mentioned
	[66] Kim et al. (2017)	On/S/C/Dy	O_{21}	Not mentioned
	[67] Khebbache et al. (2018)	On/S/C/St	O_{21}	Not mentioned
	[68] Luizelli et al. (2017)	On/S/C/St	O_{23}	In non-polynomial and polynomial time
ML	[15] Pei et al. (2020)	On/S/C/St	O_{11}	In short time
	[17] Wang et al. (2019)	On/S/C/St	O_{11} and O_{21}	Not mentioned
	[18] Pei et al. (2020)	On/S/C/Dy	O_{21}	In short time
	[20] Gu et al. (2019)	On/S/C/Dy	O_{21}	Affordable
	[37] Quang et al. (2019)	On/S/C/St	O_{22}	Affordable
	[40] Quang et al. (2019)	On/M/D/St	O_{21}	Not mentioned
	[41] Sun et al. (2018)	On/S/ (C/D)/Dy	O_{22}	In short time
	[70] Kim et al. (2017)	On/S/C/Dy	O_{12}	Not mentioned
	[72] Quang et al. (2020)	On/S/C/St	O_{22}	Affordable
	[73] Pan *et al.* (2020)	On/S/C/St	O_{21}	Not mentioned
	[74] Gu et al. (2020)	On/S/C/St	O_{21}	Affordable
	[75] Pei et al. (2018)	On/S/C/St	O_{11}	In short time

6.1 Coordination of NFV-RA stages

As described in Section 1, NFV-RA can be carried out in three stages: VNF-CC, VNF-FGE, and VNF-SCH. In this paper, we mainly focus on VNF-FGE. However, with the gradual deepening of VNF-FGE research, there is a certain relevance among these stages, and a better embedding performance can be obtained by considering them together. At present, there are some works [76–79] have been devoted to solving the VNF-FGE problem with the VNF-CC or VNF-SCH coordination. These works can achieve good results. However, few studies have combined all these stages, which might be an excellent potential for research in this area.

6.2 New objectives

6.2.1 Energy-aware embedding

In B5G/6G networks, global coverage will lead to the more widespread utilization of cloud servers and virtualized radio access networks, which highlights the energy consumption issue. The high energy consumption means that the infrastructure providers have to pay more for electricity and more taxes on the emitted carbon dioxide. Therefore, studying an energy-aware VNF-FGE is meaningful. Several works are trying to solve this problem from different perspectives. For example, Tajiki et al. [33] provided an energy consumption model considering three different modes of the server: off, on-idle and on-active. Eramo et al. [80] proposed a strategy to reduce the energy consumption by reconfiguring some VNF instances and turn off the unoccupied servers during idle hours. Therefore, energy-aware VNF-FGE is still a potential research direction in the future.

6.2.2 Security-aware embedding

Due to the introduction of virtualization layer into the network architecture, security issue arise. In B5G/6G networks, diverse service requirements (e.g., online payment, video conferencing, online gaming) are translated to fine-grained security levels. Security-aware embedding is bidirectional. For a virtual network operator (VNO) who rents underlying infrastructures from infrastructure providers (InPs) for VNF-FGs, the VNFs can be deployed on the substrate infrastructures owned by different InPs. VNO hope to choose the hardware facilities of InPs with high security. Similarly, an InP can also host network service requests from multiple VNOs. The InPs also intend to select VNOs with high security level to operate the network so that its infrastructure is not compromised. Therefore, there is a need to avoid the embedding that will increase the risk of security for each of stakeholders (i.e., VNO, InP). Very few works have focused on this direction. Li et al. [81] proposed an automatic algorithm based on Q-learning to choose appropriate security SFCs with various requirements when taking security performance, service quality, deployment cost, and network function diversity into consideration. So how to efficiently solve the VNF-FGE based on the security demands of VNOs and InPs' security policies without sacrificing the QoS and revenue is still an open issue.

6.2.3 Reliability-aware embedding

In B5G/6G networks, due to the growing complexity of NFV-MANO, the incidence of unexpected failures in the network increases. VNFs suffer from various types of failures in different aspects, such as malicious attacks, software, and hardware failures. Failure of VNFs can result in interruptions of network service. Service reliability refers to the probability that the service has not failed over a specific period [82]. Therefore, we should pay attention to the reliability problem in the VNF-FGE. The general way to improve the reliability of the network services is to provide redundant configuration and replace the failed function with the backup VNF. Khezri et al. [83] introduced an algorithm based on DQN for reliability-aware deployment of VNFs. They focused on the recovery order of the servers, considering the number of functions available during the recovery process. Alahmad et al. [84] modeled the reliability requirements of network services and proposed an optimized deployment strategy from reliability perspectives of network service. Although several studies have worked on this issue, reliability-aware embedding is still a crucial direction.

6.3 Distributed VNF-FGE

The problem of VNF-FGE can be tackled in either a centralized or distributed manner. In a centralized approach, there is only one entity to compute the optimal solution for embedding. Therefore, a centralized entity is too fragile that it will suffer from the single point failure and can not deal with the scalability issue in large networks. With the rise of massive terminals and mobile edge computing paradigm, distributed computing is also on the rise. The distributed approaches for solving the VNF-FGE problem can be divided into two branches: 1) There are multiple entities responsible for the embedding in a single domain. Leivadeas et al. [65] formulated the deployment of the VNFs of an SFC as a partitioning game, found the Nash equilibrium for the particular game, and worked out the relationship between the Nash equilibrium and the optimal

partitioning; 2) The second branch is utilizing multiple entities to implement the embedding in multiple domains. Hang et al. [39] adopted resource orchestrators to manage the physical nodes and links. Each resource orchestrator is associated with a network domain-orchestrators communicated by control channels to compute an SFC request. For the distributed approaches in VNF-FGE, several potential directions such as federated learning [85] and game-theoretical formulation occur when considering the privacy issue and selfish stakeholders.

6.4 The prediction of VNF-FGs

Many works have focused on the VNF-FGE from the theoretical perspective and evaluated the algorithms based on the synthetic data. The research on real-world data is rarely reported. Particularly, the prediction of VNF-FGs (e.g., topology, resource requirement, traffic profile) could help the InPs allocate resources in a coarse-grained manner in advance. Mijumbi et al. [86] proposed a prediction approach with the insight of obtaining the constraints of requests in advance. It will avoid system outages and QoS degradation effectively. Graphic neural network [87] was employed to predict the VNFs requirements. VNFs are formulated as two parametric functions which are implemented by feedforward neural networks. When the VNF-FG is large, this approach might need a large number of storage resources. So there is still some space to seek a trade-off between prediction accuracy and resources (i.e., computing, storage) consumption.

6.5 Artificial intelligence for optimization

The current research on artificial intelligence for solving problems in the network field is mainly based on DRL. These DRL algorithms include DQN, DDPG, etc. DRL means that agents learn to make decisions in a "trial and error" way. Rewards are obtained through interaction with the environment to guide behaviors. The goal is to maximize rewards for agents. This learning insight has shown the potential of solving the VNF-FGE problem. However, how to reduce the scale of action space and prevent lagging feedback of reward are still urgent issues.

In addition, other ML methods such as Hopfield networks [88], Boltzmann machines [15], pointer networks [89] and graph neural network [73] have also be applied to the VNF-FGE problem. These neural networks are capable of capturing the changes over time or the structural attributes of SN and VNF-FGs. However, figuring out the optimal parameter setting is a time-consuming task. Therefore, studying the method for automatically setting the optimal parameters for ML methods is also an interesting topic.

7 Conclusion

As the indispensable core technology in B5G/6G networks, NFV will absorb the attention of industry and academic increasingly. By decoupling network functions from dedicated hardware appliances, NFV could enhance the service flexibility and reduce the management cost. Resource allocation plays a crucial role in guaranteeing the performance of NFV-based service. As a pivotal process of resource allocation in NFV, VNF-FGE has become a noteworthy research direction. We have presented a comprehensive survey of VNF-FGE in this paper. A general formulation and several objectives have been firstly discussed. Then we have presented different application scenarios from four dimensions (online/offline, single-domain/multi-domain, centralized/distributed and static/dynamic). We also have divided the approaches into four main categories (exact, heuristic-based, metaheuristic-based and ML-based) according to distinct optimization methods. Finally, we have discussed the challenging and potential research directions of VNF-FGE in the B5G/6G networks.

Acknowledgements This work is supported in part by the Major Special Program for Technical Innovation & Application Development of Chongqing Science & Technology Commission (No. CSTC 2019jscx-zdztzxX0031), the National NSFC (No. 61902044, 62072060, 61902178), the National Key R & D Program of China (No. 2018YFF0214700, 2018YFB2100100), the Chongqing Research Program of Basic Research and Frontier Technology (No. cstc2019jcyj-msxmX0589, cstc2018jcyjAX0340), the Natural Science Foundation of Jiangsu (No. BK20190295), the Leading Technology of Jiangsu Basic Research Plan (No. BK20192003), and the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Grant Agreement No. 898588.

References

1. Union, I. (2015). *Imt traffic estimates for the years 2020 to 2030. Report ITU* (p. 2370)
2. You, X., Wang, C. X., Huang, J., Gao, X., Zhang, Z., Wang, M., Huang, Y., Zhang, C., Jiang, Y., Wang, J., et al. (2021). Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts. *Science China Information Sciences, 64*(1), 1–74.
3. Herrera, J. G., & Botero, J. F. (2016). Resource allocation in NFV: A comprehensive survey. *IEEE Transactions on Network and Service Management, 13*(3), 518–532.
4. Group, N., et al. (2013). Network functions virtualisation (NFV) architectural framework. *Tech. rep., Technical Report ETSI GS NFV,* 002.
5. Halpern, J. & Pignataro, C., et al. (2015). Service function chaining (SFC) architecture. In *RFC 7665*.
6. Bhamare, D., Jain, R., Samaka, M., & Erbad, A. (2016). A survey on service function chaining. *Journal of Network and Computer Applications, 75*, 138–155.

7. Xie, Y., Liu, Z., Wang, S., & Wang, Y. (2016). Service function chaining resource allocation: A survey. arXiv preprint arXiv:1608.00095.
8. Mirjalily, G., & Zhiquan, L. (2018). Optimal network function virtualization and service function chaining: A survey. *Chinese Journal of Electronics, 27*(4), 704–717.
9. Wang, X., Han, Y., Leung, V. C., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials, 22*(2), 869–904.
10. Wang, X., Li, X., Pack, S., Han, Z., & Leung, V. C. M. (2020). STCS: spatial-temporal collaborative sampling in flow-aware software defined networks. *IEEE Journal on Selected Areas in Communications, 38*(6), 999–1013.
11. Qiu, C., Yao, H., Wang, X., Zhang, N., Yu, F. R., & Niyato, D. (2020). AI-Chain: blockchain energized edge intelligence for beyond 5G networks. *IEEE Network, 34*(6), 62–69.
12. Sun, G., Yu, H., Li, L., Anand, V., Cai, Y., & Di, H. (2012). Exploring online virtual networks mapping with stochastic bandwidth demand in multi-datacenter. *Photonic Network Communications, 23*(2), 109–122.
13. Zhang, S., Qian, Z., Tang, B., Wu, J., & Lu, S. (2011). Opportunistic bandwidth sharing for virtual network mapping. In *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, pp. 1–5. IEEE.
14. Zhang, S., Qian, Z., Wu, J., & Lu, S. (2012). An opportunistic resource sharing and topology-aware mapping framework for virtual networks. In *2012 Proceedings IEEE INFOCOM*, pp. 2408–2416. IEEE.
15. Pei, J., Hong, P., Xue, K., Li, D., Wei, D. S., & Wu, F. (2020). Two-phase virtual network function selection and chaining algorithm based on deep learning in SDN/NFV-enabled networks. *IEEE Journal on Selected Areas in Communications, 38*(6), 1102–1117.
16. Agarwal, S., Malandrino, F., Chiasserini, C. F., & De, S. (2019). VNF placement and resource allocation for the support of vertical services in 5G networks. *IEEE/ACM Transactions on Networking, 27*(1), 433–446.
17. Wang, S., & Lv, T. (2019). Deep reinforcement learning for demand-aware joint VNF placement-and-routing. In *2019 IEEE Globecom Workshops*, pp. 1–6. IEEE.
18. Pei, J., Hong, P., Pan, M., Liu, J., & Zhou, J. (2020). Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks. *IEEE Journal on Selected Areas in Communications, 38*(2), 263–278.
19. Li, W., Wu, H., Jiang, C., Jia, P., Li, N. & Lin, P. (2020). Service chain mapping algorithm based on reinforcement learning. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 800–805. IEEE.
20. Gu, L., Zeng, D., Li, W., Guo, S., Zomaya, A. & Jin, H. (2019). Deep reinforcement learning based VNF management in geo-distributed edge computing. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 934–943. IEEE
21. Su, S., Zhang, Z., Liu, A. X., Cheng, X., Wang, Y., & Zhao, X. (2014). Energy-aware virtual network embedding. *IEEE/ACM Transactions on Networking, 22*(5), 1607–1620.
22. Su, S., Zhang, Z., Cheng, X., Wang, Y., Luo, & Y., Wang, J. (2012). Energy-aware virtual network embedding through consolidation. In *2012 Proceedings IEEE INFOCOM Workshops*, pp. 127–132. IEEE.
23. Zhang, Z., Su, S., Zhang, J., Shuang, K., & Xu, P. (2015). Energy aware virtual network embedding with dynamic demands: Online and offline. *Computer Networks, 93*, 448–459.
24. Rivoire, S., Ranganathan, P., & Kozyrakis, C. (2008). A comparison of high-level full-system power models. *HotPower, 8*(2), 32–39.
25. Fan, X., Weber, W. D., & Barroso, L. A. (2007). Power provisioning for a warehouse-sized computer. *ACM SIGARCH computer architecture news, 35*(2), 13–23.
26. Economou, D., Rivoire, S., Kozyrakis, C. & Ranganathan, P. (2006). Full-system power analysis and modeling for server environments. ACM SIGARCH computer architecture news pp. 70–77
27. Chiaraviglio, L., Mellia, M., & Neri, F. (2012). Minimizing ISP network energy cost: formulation and solutions. *IEEE/ACM Transactions on Networking, 20*(2), 463–476.
28. Li, X. & Qian, C. (2015). The virtual network function placement problem. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 69–70. IEEE.
29. Tastevin, N., Obadia, M., & Bouet, M. (2017). A graph approach to placement of service functions chains. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 134–141. IEEE*
30. Cao, J., Zhang, Y., An, W., Chen, X., Sun, J., & Han, Y. (2017). VNF-FG design and VNF placement for 5G mobile networks. *Science China Information Sciences, 60*(4), 040302.
31. Kuo, T. W., Liou, B. H., Lin, K. C. J., & Tsai, M. J. (2018). Deploying chains of virtual network functions: On the relation between link and server usage. *IEEE/ACM Transactions On Networking, 26*(4), 1562–1576.
32. Quang, P. T. A., Bradai, A., Singh, K. D., Picard, G., & Riggio, R. (2018). Single and multi-domain adaptive allocation algorithms for VNF forwarding graph embedding. *IEEE Transactions on Network and Service Management, 16*(1), 98–112.
33. Tajiki, M. M., Salsano, S., Chiaraviglio, L., Shojafar, M., & Akbari, B. (2018). Joint energy efficient and QoS-aware path allocation and VNF placement for service function chaining. *IEEE Transactions on Network and Service Management, 16*(1), 374–388.
34. Wang, W., Hong, P., Lee, D., Pei, J., & Bo, L. (2017). Virtual network forwarding graph embedding based on Tabu search. In *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6. IEEE.
35. Jang, I., Suh, D., Pack, S., & Dán, G. (2017). Joint optimization of service function placement and flow distribution for service function chaining. *IEEE Journal on Selected Areas in Communications, 35*(11), 2532–2541.
36. Soualah, O., Mechtri, M., Ghribi, C. & Zeghlache, D. (2018). A green VNF-FG embedding algorithm. In *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pp. 141–149. IEEE .
37. Quang, P. T. A., Hadjadj-Aoul, Y., & Outtagarts, A. (2019). A deep reinforcement learning approach for VNF forwarding graph embedding. *IEEE Transactions on Network and Service Management, 16*(4), 1318–1331.
38. Soares, J. & Sargento, S. (2015) Optimizing the embedding of virtualized cloud network infrastructures across multiple domains. In *2015 IEEE International Conference on Communications (ICC)*, pp. 442–447. IEEE
39. Zhang, Q., Wang, X., Kim, I., Palacharla, P., & Ikeuchi, T. (2016). Service function chaining in multi-domain networks. In *2016 Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1–3. IEEE.
40. Quang, P. T. A., Bradai, A., Singh, K. D. & Hadjadj-Aoul, Y. (2019). Multi-domain non-cooperative VNF-FG embedding: A deep reinforcement learning approach. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 886–891. IEEE.

41. Sun, J., Huang, G., Sun, G., Yu, H., Sangaiah, A. K., & Chang, V. (2018). A q-learning-based approach for deploying dynamic service function chains. *Symmetry, 10*(11), 646.
42. Ruiz, L., Barroso, R. J. D., De Miguel, I., Merayo, N., Aguado, J. C., De La Rosa, R., Fernández, P., Lorenzo, R. M., & Abril, E. J. (2020). Genetic algorithm for holistic VNF-mapping and virtual topology design. *IEEE Access, 8*, 55893–55904.
43. Li, H., Wang, L., Wen, X., Lu, Z., & Li, J. (2018). MSV: An algorithm for coordinated resource allocation in network function virtualization. *IEEE Access, 6*, 76876–76888.
44. Pei, J., Hong, P., Xue, K., & Li, D. (2018). Efficiently embedding service function chains with dynamic virtual network function placement in geo-distributed cloud system. *IEEE Transactions on Parallel and Distributed Systems, 30*(10), 2179–2192.
45. Clayman, S., Maini, E., Galis, A., Manzalini, A. & Mazzocca, N. (2014). The dynamic placement of virtual network functions. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pp. 1–9.
46. Liu, J., Lu, W., Zhou, F., Lu, P., & Zhu, Z. (2017). On dynamic service function chain deployment and readjustment. *IEEE Transactions on Network and Service Management, 14*(3), 543–553.
47. Bari, M. F., Chowdhury, S. R., Ahmed, R., & Boutaba, R. (2015) On orchestrating virtual network functions. In: 2015 11th International Conference on Network and Service Management (CNSM), pp. 50–56. IEEE.
48. Luizelli, M. C., Bays, L. R., Buriol, L. S., Barcellos, M. P., & Gaspary, L. P. (2015). Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions. In *2015 IFIP/IEEE International Symposium on Integrated Network Management*, pp. 98–106. IEEE.
49. Moens, H., & De Turck, F. (2014). VNF-P: A model for efficient placement of virtualized network functions. In *10th International Conference on Network and Service Management (CNSM) and Workshop*, pp. 418–423. IEEE.
50. Riggio, R., Bradai, A., Rasheed, T., Schulz-Zander, J., Kuklinski, S., & Ahmed, T. (2015). Virtual network functions orchestration in wireless networks. In *2015 11th International Conference on Network and Service Management (CNSM)*, pp. 108–116. IEEE.
51. Sahhaf, S., Tavernier, W., Rost, M., Schmid, S., Colle, D., Pickavet, M., & Demeester, P. (2015). Network service chaining with optimized network function embedding supporting service decompositions. *Computer Networks, 93*, 492–505.
52. Jahromi, N. T., Kianpisheh, S. & Glitho, R. H. (2018). Online VNF placement and chaining for value-added services in content delivery networks. In *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, pp. 19–24. IEEE.
53. Mijumbi, R., Serrat, J., Gorricho, J., Rubio-Loyola, J. & Davy, S. (2015). Server placement and assignment in virtualized radio access networks. In *2015 11th International Conference on Network and Service Management (CNSM)*, pp. 398–401. IEEE.
54. Addis, B., Belabed, D., Bouet, M. & Secci, S. (2015). Virtual network functions placement and routing optimization. In *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, pp. 171–177. IEEE.
55. Marotta, A., & Kassler, A. (2016). A power efficient and robust virtual network functions placement problem. In *2016 28th International Teletraffic Congress (ITC 28)*, vol. 1, pp. 331–339. IEEE.
56. Lin, T., Zhou, Z., Tornatore, M., & Mukherjee, B. (2016). Demand-aware network function placement. *Journal of Lightwave Technology, 34*(11), 2590–2600.
57. Ghaznavi, M., Shahriar, N., Ahmed, R., & Boutaba, R. (2016). Service function chaining simplified. arXiv preprint arXiv:1601.00751.
58. Mehraghdam, S., Keller, M., & Karl, H. (2014). Specifying and placing chains of virtual network functions. In *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pp. 7–13. IEEE.
59. Jang, I., Choo, S., Kim, M., Pack, S. & Shin, M. (2016). Optimal network resource utilization in service function chaining. In *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, pp. 11–14. IEEE.
60. Khebbache, S., Hadji, M. & Zeghlache, D. (2017). Scalable and cost-efficient algorithms for VNF chaining and placement problem. In *2017 20th conference on innovations in clouds, internet and networks (ICIN)*, pp. 92–99. IEEE .
61. Tajiki, M. M., Salsano, S., Shojafar, M., Chiaraviglio, L., & Akbari, B. (2018). Energy-efficient path allocation heuristic for service function chaining. In *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pp. 1–8. IEEE.
62. Jia, Z., Sheng, M., Li, J., Liu, R., Guo, K., Wang, Y., Chen, D. & Ding, R. (2018). Joint optimization of VNF deployment and routing in software defined satellite networks. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5. IEEE.
63. Li, B., Cheng, B., Wang, M., Liu, X., Yue, Y., & Chen, J. (2019). Joint correlation-aware VNF selection and placement in cloud data center networks. In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 171–176. IEEE.
64. Yang, K., Zhang, H. & Hong, P. (2016). Energy-aware service function placement for service function chaining in data centers. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6. IEEE.
65. Leivadeas, A., Kesidis, G., Falkner, M., & Lambadaris, I. (2017). A graph partitioning game theoretical approach for the VNF service chaining problem. *IEEE Transactions on Network and Service Management, 14*(4), 890–903.
66. Kim, S., Park, S., Kim, Y., Kim, S., & Lee, K. (2017). VNF-EQ: dynamic placement of virtual network functions for energy efficiency and QoS guarantee in NFV. *Cluster Computing, 20*(3), 2107–2117.
67. Khebbache, S., Hadji, M., & Zeghlache, D. (2018). A multi-objective non-dominated sorting genetic algorithm for VNF chains placement. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–4. IEEE.
68. Luizelli, M. C., da Costa Cordeiro, W. L., Buriol, L. S., & Gaspary, L. P. (2017). A fix-and-optimize approach for efficient and large scale virtual network function placement and chaining. *Computer Communications, 102*, 67–77.
69. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.
70. Kim, S. I., & Kim, H. S. (2017). A research on dynamic service function chaining based on reinforcement learning using resource usage. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 582–586. IEEE.
71. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
72. Quang, P. T. A., & Hadjadj-Aoul, Y. (2020). Outtagarts: Evolutionary actor-multi-critic model for VNF-FG embedding. In *IEEE Consumer Communications & Networking Conference (CCNC 2020)*, pp. 1–6. IEEE.
73. Pan, P., Fan, Q., Wang, S., Li, X., Li, J., & Shi, W. (2020). GCN-TD: A learning-based approach for service function chain deployment on the fly. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–6. IEEE
74. Gu, L., Zeng, D., Li, W., Guo, S., Zomaya, A. Y., & Jin, H. (2019). Intelligent VNF orchestration and flow scheduling via

model-assisted deep reinforcement learning. *IEEE Journal on Selected Areas in Communications, 38*(2), 279–291.
75. Pei, J., Hong, P. & Li, D. (2018). Virtual network function selection and chaining based on deep learning in SDN and NFV-enabled networks. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6. IEEE.
76. Spinnewyn, B., Isolani, P. H., Donato, C., Botero Botero, J. F., & Latré, S. (2018). Coordinated service composition and embedding of 5G location-constrained network functions. *IEEE Transactions on Network and Service Management, 15*(4), 1488–1502.
77. Wang, Z., Zhang, J., Huang, T., & Liu, Y. (2019). Service function chain composition, placement, and assignment in data centers. *IEEE Transactions on Network and Service Management, 16*(4), 1638–1650.
78. Alameddine, H.A., Qu, L., & Assi, C. (2017). Scheduling service function chains for ultra-low latency network services. In *2017 13th International Conference on Network and Service Management (CNSM)*, pp. 1–9. IEEE.
79. Huang, X., Bian, S., Gao, X., Wu, W., Shao, Z., & Yang, Y. (2019). Online VNF chaining and scheduling with prediction: optimality and trade-offs. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6. IEEE.
80. Eramo, V., Miucci, E., Ammar, M., & Lavacca, F. G. (2017). An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures. *IEEE/ACM Transactions on Networking, 25*(4), 2008–2025.
81. Li, G., Zhou, H., Feng, B., Li, G. & Yu, S. (2018). Automatic selection of security service function chaining using reinforcement learning. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6. IEEE.
82. Isg, N. (2016). Network functions virtualisation (nfv); reliability; report on models and features for end-to-end reliability. *ETSI Standard GS NFV-REL, 003*.
83. Khezri, H. R., Moghadam, P. A., Farshbafan, M. K., Shah-Mansouri, V., Kebriaei, H., & Niyato, D. (2019). Deep reinforcement learning for dynamic reliability aware NFV-based service provisioning. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6. IEEE.
84. Alahmad, Y., & Agarwal, A. (2019). VNF placement strategy for availability and reliability of network services in NFV. In *2019 Sixth International Conference on Software Defined Systems (SDS)*, pp. 284–289. IEEE.
85. Wang, X., Wang, C., Li, X., Leung, V. C. M., & Taleb, T. (2020). Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching. *IEEE Internet of Things Journal, 7*(10), 9441–9455.
86. Mijumbi, R., Hasija, S., Davy, S., Davy, A., Jennings, B., & Boutaba, R. (2017). Topology-aware prediction of virtual network function resource requirements. *IEEE Transactions on Network and Service Management, 14*(1), 106–120.
87. Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks, 20*(1), 61–80.
88. Blenk, A., Kalmbach, P., Zerwas, J., Jarschel, M., Schmid, S. & Kellerer, W. (2018). NeuroViNE: A neural preprocessor for your virtual network embedding algorithm. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 405–413. IEEE.
89. Solozabal, R., Ceberio, J., Sanchoyerto, A., Zabala, L., Blanco, B., & Liberal, F. (2020). Virtual network function placement optimization with deep reinforcement learning. *IEEE Journal on Selected Areas in Communications, 38*(2), 292–303.

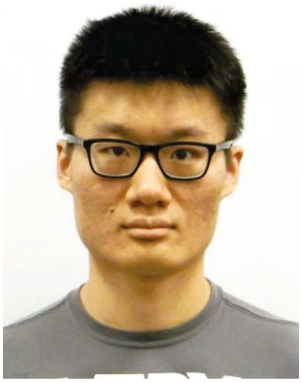
Biao Zhang received his B.E. degree in the School of Computer Science and Information Engineering, Hubei University, Wuhan, China, in 2018, and he is currently working towards his M.E. degree from the School of Big Data and Software Engineering, Chongqing University, Chongqing, China. His research interests include network optimization, information-centric networking, software defined networking and network functions virtualization.

Qilin Fan (M'19) is currently a Lecturer in the School of Big Data and Software Engineering, Chongqing University, Chongqing, China. She received the B.E. degree in the College of Software Engineering, Sichuan University, Chengdu, China, in 2011, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2017. Her research interests include network optimization, mobile edge computing and caching, network virtualization and machine learning.

Xu Zhang (M'19) received the B.Sc. degree in communication engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2012, and the Ph.D. degree from the Department of Computer Science and Technology at Tsinghua University, Beijing, China, in 2017. He is currently a Marie Sklodowska-Curie Individual Fellowship (Research Fellow) in the College of Engineering, Mathematics & Physical Sciences, University of Exeter, UK. His research interests include artificial intelligence, multimedia communication and network measurement.

Zhihan Fu received his B.E. degree in the School of Software Engineering, Nanchang hangkong University, Nanchang, China, in 2019, and he is currently working toward his M.E. degree from the School of Big Data and Software Engineering, Chongqing University, Chongqing, China. His research interests include network optimization, software defined networking, network virtualization and machine learning.

Sen Wang (M'15) is an Associate Professor in the School of Software Engineering, Chongqing University, Chongqing, China. He received B.S., M.S., and Ph.D. degree in computer science in University of Science and Technology of China (USTC), Chinese Academy of Sciences (CAS) and Tsinghua University, China, in 2005, 2008 and 2014, respectively. His research interests include in-network caching, information-centric networking, cloud computing, software defined networking and network functions virtualization.



Jian Li (S'16, M'18) is an Assistant Professor of Computer Engineering with the Department of Electrical and Computer Engineering at Binghamton University, State University of New York (SUNY). He was a postdoc with the College of Information and Computer Sciences, University of Massachusetts Amherst from January 2017 to August 2019. He received the Ph.D. degree in Computer Engineering from Texas A&M University in December 2016, and B.E. degree from Shanghai Jiao Tong University in June 2012. His current research interests lie in the areas of reinforcement learning, online learning, network optimization, online algorithms and their applications in large scale networked systems

Qingyu Xiong received the Ph.D. degree in Electrical and Electronic Systems Engineering from Kyushu University of Japan in 2002 C and the M.S. and B.S. degree from Chongqing University of China in 1991 and 1986, respectively. He is currently a professor in the School of Big Data & Software Engineering, Chongqing University of China. He is also the member of China Computer Federation. His research interests include artificial intelligent control, and software service.