Research Article Vol. 14, No. 10 / October 2022 / Journal of Optical Communications and Networking 757

JOURNAL OF
Optical Communications
and Networking

Delay-aware and resource-efficient service function chain mapping in inter-datacenter elastic optical networks

MIN ZHU,[1,2,*] JIAHUA GU,[1,2] TIANYU SHEN,[3] JIAO ZHANG,[1,2] AND PINGPING GU[4]

[1]National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China
[2]Purple Mountain Laboratories, Nanjing 211111, China
[3]School of Electronic Science and Engineering, Southeast University, Nanjing 210096, China
[4]Taicang T&W Electronics Company Ltd., Taicang 215400, China
*Corresponding author: minzhu@seu.edu.cn

Received 13 April 2022; revised 18 August 2022; accepted 21 August 2022; published 6 September 2022

Network function virtualization (NFV) is emerging as a promising paradigm for network architectures. By migrating network functions from dedicated hardware appliances to software instances running in a virtualized environment, NFV promises to offer a more flexible way to deploy and manage service function chains (SFCs). When deploying these SFCs to users, the network operators require not only the user's demands (e.g., end-to-end delay) to be satisfied, but require the cost of SFC mapping to be minimized (e.g., resource consumption). To fulfill these two goals, in this paper, we have investigated how to realize the delay-aware and resource-efficient SFC mapping in inter-datacenter elastic optical networks. We first formulate an integer linear programming (ILP) model to solve the problem exactly. The main optimization goal in the ILP model is to jointly minimize resource consumption and end-to-end delay to achieve optimal virtual network function placement. Then, a delay-aware and load-balancing mapping algorithm (DALB-MA) is proposed to obtain a near-optimal solution in a reasonable amount of time. Finally, we evaluate the proposed ILP model and heuristic algorithms via extensive simulations. The results indicate that the proposed ILP model and the DALB-MA outperform the benchmarks in terms of block rate, average cost, number of CPUs used, maximum frequency slot index, and delay margin gain.

https://doi.org/10.1364/JOCN.461263

1. INTRODUCTION

In traditional networks, network functions such as firewalls, deep package inspections (DPIs), and gateways are generally implemented by dedicated hardware appliances. These middle-boxes are generally expensive, vendor-specific and location-fixed, making it difficult to dynamically scale their capabilities and add new functions to existing hardware devices. To solve the dilemma faced by traditional networks, prevalent network function virtualization (NFV) has been proposed as a promising technology for those interested network service providers [1,2]. By decoupling software functions from hardware devices, NFV enables the replacement of the dedicated devices with software instances running on commodity servers. Specifically, NFV allows network service providers to deploy virtual network functions (VNFs) in a more efficient and agile manner using generic network resources (e.g., CPU cycles, bandwidth, and memory space), which can significantly reduce deployment time, capital expenditures (CAPEX), and operational expenses (OPEX) [3].

As one of the most important application scenarios of NFV, the VNF service function chain (SFC) steers network service flows through a series of VNFs in a predefined order to obtain a complete end-to-end service from source to destination [4]. As is shown in Fig. 1, an SFC request can be abstracted into a directed linear topology. The two ellipses at the two ends represent the service terminal and the user, respectively. Once the SFC request is given, the source (i.e., service terminal) and destination (i.e., user) nodes are determined in the network. The rectangles represent the VNFs such as DPIs, firewalls, and gateways in the network, which are interconnected by directed virtual links in a predefined order.

When deploying an SFC request, service providers need to find some appropriate datacenters (DCs) to instantiate VNFs requested by the SFC and provision light paths between the selected DCs to connect the service terminal, intermediate VNFs, and the user. Specifically, instantiating VNFs consumes IT resources (i.e., CPU and memory) of DCs and deploying a virtual light path occupies both fiber link resources (e.g., bandwidth) and node resources (e.g., optical transceiver).

1943-0620/22/100757-14 Journal © 2022 Optica Publishing Group

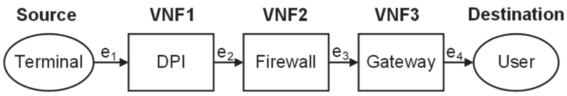

Fig. 1. Example of an SFC request.

To support high-bandwidth SFC requests, we consider inter-DC elastic optical networks (inter-DC EONs) with flexible grids as the NFV infrastructure (NFVI). With optical orthogonal frequency-division multiplexing (O-OFDM) technology, the EONs allow the optical spectra to be allocated at finer granularity (e.g., 12.5 GHz), which facilitates agile spectrum management and provides higher spectrum efficiency in the optical layer [5].

During the SFC mapping, to enhance the instance resource utilization, VNFs of the same type in different requests may be mapped to the same instance activated on the same DC, i.e., reuse of the VNF instance aimed for resource-efficient deployment. However, the VNF processing delay inevitably increases due to *instance competition* induced by instance reuse, i.e., when multiple SFC requests simultaneously access the same VNF instance, they have to wait in a queue before the requested VNF instance becomes available. In other words, the more times the VNF instances are reused, the fewer resources are required, but the longer the VNF queuing time would be, and thus the VNF processing delay increases, and vice versa. Hence, to ensure better user quality of experience (QoE), network service providers need to satisfy the delay requirement of the SFC requests, in addition to reducing resource consumption. Therefore, it is very desirable to pursue a delay-aware resource optimization scheme for SFC mapping that not only satisfies the end-to-end delay requirement of the SFC requests, but also reduces as much as possible the consumed computing resources of the DCs and the optical spectrum resource in fiber links to minimize the mapping costs.

Our main contributions in this paper are described as follows:

(1) To improve the DC resource utilization, a VNF reuse mechanism is adopted. Meanwhile, the reuse of VNF instances introduces higher VNF processing delay due to instance competition. How to achieve a trade-off between resource efficiency and processing delay is investigated in this paper.
(2) We formulate an integer linear programming (ILP) model for the delay-aware and resource-efficient SFC mapping problem in inter-DC EONs. The delay-related constraints in the ILP model include the VNF processing delay caused by the VNF reuse, which was not considered by most existing works.
(3) We propose a delay-aware load-balancing mapping heuristic algorithm (DALB-MA) that minimizes the required computing and network resources under the given request delay requirements. In the DALB-MA, we design a reuse control factor to balance between reusing the existing VNF instance for better resource efficiency and instantiating a new VNF instance to reduce VNF processing delay.
(4) We implement the proposed ILP model and heuristic algorithm via extensive simulations. The performances of the ILP model and the DALB-MA are compared and analyzed against the benchmark algorithms.

The rest of the paper is organized as follows. We provide a survey on the related works in Section 2. In Section 3, we introduce the network model and formulate the ILP model. A heuristic algorithm is proposed in Section 4, and the algorithm performance is evaluated by comparing it with the benchmark heuristics in Section 5. Finally, Section 6 summarizes the paper.

2. RELATED WORKS

A. SFC Mapping in IP-Based Networks

Recently, the SFC mapping problem in IP-based networks has been widely investigated. To minimize the overall cloud network resource cost, Feng *et al*. [6] proposed an offline approximation algorithm to find a low-cost solution to place VNFs and steer the corresponding traffic flow. In [7], an optimization model was presented to minimize the cost of IT and bandwidth resources by coordinating the deployment operations. Liu *et al*. [8] studied a dynamic SFC deployment issue, and a column generation (CG) model was designed to optimize the node and bandwidth resources while considering the trade-off between resource consumption and operational overhead. Kuo *et al*. [9] studied the joint problem of the VNF placement and path selection to improve network utilization. They first used the idea of stress testing to find an optimal link-server relation guide, and then proposed a chain deployment algorithm based on the guidance. The authors of [10] proposed a heuristic solution called merge-split Viterbi to solve the typical three-stage coordinated NFV resource allocation model. These works mainly optimized the resource allocation while ignoring the optimization of end-to-end delay of SFC requests.

To jointly minimize the delay and resource consumption, Qu *et al*. [11] considered the VNF transmission and processing delays and formulated the joint problem of VNF scheduling and traffic steering as a mixed integer linear program (MILP). Researchers [12] investigated the linear dependency between the number of resources allocated to a VNF and its processing delay and proposed a flexible resource allocation model (FRAM) to minimize the resource consumption while meeting the end-to-end delay requirements. Yang *et al*. [13] studied the delay-sensitive and available-aware VNF scheduling (DAVS) problem and proposed an efficient near-optimal heuristic that iteratively places VNFs per segment with a much shorter running time. In the literature [14–16], Sun *et al*. investigated the efficient SFC deployment problem; several heuristic deployment algorithms were proposed to minimize end-to-end delay and resource consumption. However, only link transmission delay and bandwidth resources were considered, and the bandwidth resources are viewed as a pool of virtualized resources without any resource allocation constraints (e.g., the spectrum continuity constraint, spectrum contiguity constraint, and spectrum nonoverlapping constraint [17]). In addition, the reuse of VNF instances is also overlooked in the above works.

B. SFC Mapping in Substrate EONs

Many researchers are working to reduce the overall resource cost of SFC implementation in substrate inter-DC EONs. Fang *et al*. [18] focused on the joint optimization of the spectrum and IT resources when SFCs were provisioned in inter-DC EONs. First, an ILP model was formulated to solve the problem exactly. Then, to reduce the complexity of the ILP model, a heuristic algorithm based on the longest common subsequence (LCS) was proposed, and the algorithm attempted to reuse as many VNF instances as possible to save IT and spectrum resources. Zeng *et al*. [19] investigated how to jointly optimize the VNF placement and the multicast routing for orchestrating NFV trees in inter-DC EONs. An ILP model and heuristic algorithms were proposed to solve the problem in both static and dynamic scenarios. In [20], Wang *et al*. studied the cost-efficient deployment of a VNF graph with more general topologies, where the total cost consisted only of used frequency slots (FSs) and deployed VNF instances. Li *et al*. [21] investigated a bandwidth prediction algorithm based on deep-learning techniques to orchestrate VNFs with a predeployment strategy, which allows for a shorter setup time without impacting the blocking probability. In [22], Li *et al.* proposed a deep reinforcement learning (DRL)-based adaptive service framework that achieves better trade-offs among overall resource utilization, VNF-SFC request-blocking probability, and the number of network reconfigurations in an inter-DC EON. Xuan *et al*. [23] studied the network planning problem in inter-DC EONs. An efficient bilevel hybrid memetic algorithm was proposed to determine the optimal routing and VNF deployment scheme along with the optimal number and location of DCs. In another study [17], Yu *et al*. investigated SFC orchestration in inter-DC EONs with the aim of maximizing Internet service provider (ISP) profits by balancing the acceptance ratio and deployment cost. The above works focused primarily on the optimization of IT and bandwidth resources without considering the modulation constraints and the SFC delay requirements. However, these constraints should be considered based on QoS requirements. Peng *et al*. [24] proposed a protection cover list-based VNF protection (PCL-VP) algorithm to solve the VNF failure protection cover (VFPC) problem. In [25], the authors presented a self-learning system based on reinforcement learning to optimize the resource allocation of SFCs in NFV-SDN-enabled metro-core optical networks. Both the studies above take end-to-end service delay into consideration. Note that only the signal propagation time is included, while the VNF processing time was omitted. In [26], the authors presented an orchestration system to select and allocate network and cloud resources for network services (VNF-forwarding graph) in distributed DCs connected through a packet over a flexi-grid optical network. To further reduce the blocking probability, they proposed a retrial mechanism in [27] for provisioning VNF-SFC requests in a flexible optical network. Although these two studies considered the VNF processing time when calculating the end-to-end service delay, the VNF reuse mechanism, which has the potential to further reserve cloud resources, was not considered. In [28], Chen *et al*. investigated mixed-strategy game-theoretic approaches to optimize the network-wide profits. In another study [29], researchers proposed to realize an incentive-driven VNF-SC provisioning in an inter-DC EON with a noncooperative mixed-strategy gaming approach. In [28,29], the signal propagation time, VNF processing time, and queuing time were taken into account. However, the authors did not consider the distance-dependent modulation formats and their effect on the spectrum assignment, while in our work, the routing, modulation, and spectrum assignment (RMSA) has been taken into consideration.

Motivated by the above observations, in our original Optical Fiber Communication Conference (OFC) paper in 2021 [30], we investigated the SFC mapping problem in inter-DC EONs with the goal of minimizing the overall resource cost (e.g., IT resources and bandwidth resources) and end-to-end delay of SFC requests while meeting the spectrum allocation constraints in EONs. However, due to the page limit, only a brief review was provided. This paper is an extension of our previous OFC paper [30], with more detailed explanations and descriptions of the network model, the ILP model, and the proposed heuristic algorithm (i.e., the DALB-MA). We also conduct more simulations to show how the key parameters of the DALB-MA (e.g., the safety level) would affect the outcome.

3. PROBLEM FORMULATION

In this section, we first show the principle of SFC mapping in inter-DC EONs with an illustrative example. Then, we describe the network model in detail and formulate an ILP model to solve the aforementioned delay-aware and resource-efficient SFC mapping problem exactly.

A. SFC Mapping in Inter-DC EON

In the inter-DC EON, a number of geographically distributed nodes are interconnected by the EON, some of which are locally attached with a DC that provides the services for different types of VNFs. Figure 2(a) shows an example of SFC mapping in an inter-DC EON, where two SFC requests R_1 and R_2 need to be deployed. R_1 requests for an SFC consist of VNF1 and VNF2 to steer traffic from Node1 to Node6. SFC request R_2 consists of VNF1; its traffic flow originates from Node1 and terminates at Node4. Figure 2(a) shows a deployment scheme where R_1 takes the path $1 \rightarrow 3 \rightarrow 5 \rightarrow 6$ and deploys VNF1 on DC2 (attached to Node3), and VNF2 on DC3 (attached to Node5). The corresponding spectrum allocations are given next to the light paths. Note that an optical transponder is required for realizing optical/electronic/optical (O/E/O) conversion at each intermediate DC site where the user's traffic is processed by a VNF. Therefore, the spectrum allocation on each link along path $1 \rightarrow 3 \rightarrow 5 \rightarrow 6$ can be done separately due to the spectrum conversion capabilities at DC2 (Node3) and DC3 (Node5). For the second SFC request R_2, it goes through $1 \rightarrow 3 \rightarrow 5 \rightarrow 4$ and reuses VNF1, which has been deployed on DC2. Because VNF1 is processed at DC2 (Node3), the spectrum allocation on links $1 \rightarrow 3$ and $3 \rightarrow 5 \rightarrow 4$ can be done separately, while that on path segments $3 \rightarrow 5$ and $5 \rightarrow 4$ should follow the spectrum continuity and contiguity constraints for transparent transmission, which is also widely adopted by most EONs (e.g., in [18,25,26]). In the alternative deployment scheme

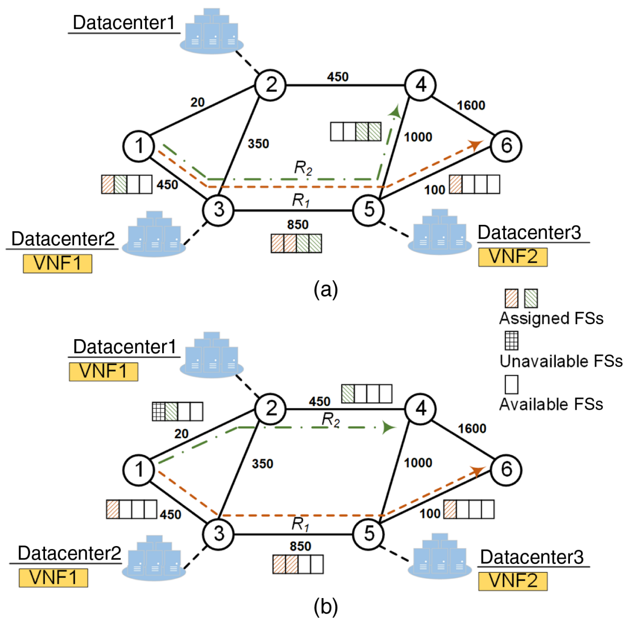

Fig. 2. Example of SFC mapping in EONs. (a) Deployment scheme 1, (b) deployment scheme 2.

presented in Fig. 2(b), R_1 still takes the path $1 \rightarrow 3 \rightarrow 5 \rightarrow 6$ while R_2 deploys VNF1 on DC1 (Node2) and takes the path $1 \rightarrow 2 \rightarrow 4$, which results in quite different DC and spectrum usages. Additionally, reusing VNFs in deployment scheme 1 inevitably increases the end-to-end delay due to the VNF instance reuse. In this paper, our objective is to minimize the DC and spectrum resource consumption simultaneously without adding end-to-end delay.

It is worth noting that we did not consider the deployment of multiple VNF instances of the same type within the same DC site, although the deployment of multiple VNF instances could reduce latency. The corresponding explanations are as follows. As we know, within the same DC site, increasing the number of VNF reuse times or/and deploying multiple VNF instances of the same type (e.g., type c), could potentially create a hotspot in the network. Specifically, if the number of VNF reuse times is not restricted, more SFC requests containing type c VNFs would traverse the node v. Hence, the DC node v may become a hotspot, and thus the processing time of VNFs would grow indefinitely. In the other case, when multiple VNF instances of the same type are deployed within the DC node v, it may become a hotspot. Accordingly, these SFC requests might be rerouted to a longer path to traverse the node when the hotspot is not on the shortest path of the request, hence consuming more spectrum resources in the network. Therefore, in this paper, it is necessary to avoid deploying multiple VNF instances of the same type in a DC node and also set limits on the number of times a VNF instance can be reused in a DC node.

As is well known, a traffic grooming policy could help reduce network resource consumption, which is often used in traditional wavelength division multiplexing (WDM) networks [31]. However, in our paper, a fine-grained EON has been adopted as a substrate network, where each FS occupies 12.5 GHz bandwidth. The fine-grained spectrum provision has already efficiently improved the spectrum utilization compared with the traditional WDM network. Further, when used in the EONs, how the traffic grooming might affect the bandwidth requirements of transponders and network costs are out of the scope of this paper and will be studied systematically in our future works.

B. Network Model

To model the SFC mapping problem, an EON is presented as a directed graph $G(V, E)$, where V and E represent the sets of optical nodes and fiber links, respectively. A part of nodes $v \in V$ is equipped with a local DC. An SFC request is denoted as $R_i(s_i, d_i, C_i, b_i, D_i^{\text{thres}})$, where s_i and d_i are the source and destination nodes, respectively. $C_i = \langle c_{i,1}, c_{i,2}, \ldots, c_{i,J_i} \rangle$ denotes the VNF sequence in the SFC request, where $c_{i,j}$ is the type of the jth VNF and J_i is the total number of requested VNFs. b_i denotes the bandwidth requirement. For simplicity, we assume the bandwidth requirement of the SFC request does not change after steering through a VNF. The number of required FSs can be obtained by $\lceil b_i/(ML_m \times B_{FS}) \rceil$, where B_{FS} is the bandwidth for each FS, i.e., 12.5 GHz, and ML_m is the level of modulation format $m \in [1, 2, 3, 4]$, corresponding to binary phase-shift keying (BPSK), quaternary phase-shift keying (QPSK), 8-quadrature amplitude modulation (8-QAM), and 16-quadrature amplitude modulation, respectively. D_i^{thres} is the end-to-end delay threshold specified by the SFC request R_i, which is mainly determined by its length (number of requested VNFs) and the types of VNF that compose the SFC. The end-to-end delay of an SFC consists of two parts: the sum of the propagation delay of physical links and the sum of the processing delay of VNF instances. The propagation delay is given as

$$D_{i,j}^{\text{link}} = l_{i,j} / v, \tag{1}$$

where $l_{i,j}$ is the physical length of the path connecting two adjacent nodes of the SFC and v is the propagation speed of signals in the physical link medium. For the configuration of processing delay, we adopt the delay model presented in [28,29]. By assuming the user traffic as the input queue and the processing CPU(s) of each VNF instance as the single server, the processing of user traffic in each VNF can be modeled as an M/M/1 queue. Hence, the average processing delay for each VNF of type c in DC node v can be calculated by

$$D_{v,c}^{\text{proc}} = \frac{1}{\varsigma_c - t_{v,c} \cdot \varphi_c}, \tag{2}$$

where ς_c is the capacity limit of a type c VNF instance, and $t_{v,c} \cdot \varphi_c$ is the total required processing capacity for a type c VNF in DC node v. $t_{v,c}$ is the number of times the VNF c is reused in node v, and φ_c is the processing resource requirement of the type c VNF.

Let us recall the SFC mapping example in Fig. 2. Assume the two SFC requests are R_1 (1,6, $\langle VNF1, VNF2 \rangle$, 50 Gbps, 400 ms) and R_2 (1,4, $\langle VNF1 \rangle$ 30 Gbps, 200 ms). Additionally, we assume there are five more SFC requests already reusing the VNF1 instance in DC2 (not drawn in Fig. 2 for simplicity). The deployment scheme shown in Fig. 2(a) reuses VNF1, which leads to the increased processing delay D^{proc} of VNF1 for R_1 and R_2. To reduce the D^{proc}, another scheme is presented in Fig. 2(b), where two VNF1 instances are deployed in two DCs, and thus more DC computing resources are required. In Fig. 2, the numbers on the line denote the link length in kilometers. Assume that the transmission distances (TDs) of BPSK, QPSK, 8-QAM, and 16-QAM are 4000, 2000, 1000, and 500 km, respectively. The capacity limits of VNF1 and VNF2 instances are 20 GOPS and 40 GOPS, respectively, and the processing resource requirement of VNF1 and VNF2 are 1 GOPS and 2 GOPS, respectively. In deployment scheme 1, two VNFs are instantiated, with the maximum frequency slot index (MFSI) being 4. The processing delay of VNF1 in DC2 is 76.9 ms, which is calculated by Eq. (2), while the processing delay of VNF2 in DC3 is 26 ms. The distance between the source and destination of the two requests is 1400 and 2300 km, respectively. Finally, the end-to-end delay of R_1 is $(76.9 + 26.3) + 7 = 110.2$ ms and the end-to-end delay of R_2 is $76.9 + 11.5 = 88.4$ ms. In deployment scheme 2, two VNFs are instantiated, with the MFSI being 2. The end-to-end delay of R_1 and R_2 is $(71.4 + 26.3) + 7 = 104.7$ ms and $52.6 + 2.4 = 55$ ms, respectively. From the two examples in Fig. 2, we can observe that the reuse of VNF instances clearly impacts system performance. Specifically, reusing VNF instances reduces DC resource consumption while causing larger MFSI and longer end-to-end delay, and vice versa. Therefore, effectively managing the reuse of the VNF instance has become the key to minimizing the consumed resources without adding end-to-end delay.

C. ILP Formulation

To deploy the SFC requests in inter-DC EONs, we need to accomplish two tasks. One is to instantiate VNFs in the DCs, and the other is to set up light paths by allocating modulation format and FSs on fiber links to satisfy the bandwidth requirements of the SFC requests. During this process, the allocated computing resources in the DCs and spectrum resources on the links should not exceed their capacities, and the light paths should also satisfy the spectrum contiguous and continuity constraints. For simplicity, in this paper, we assume the SFC requests are compute-intensive tasks with high CPU demands; hence the VNF's computing demand can be characterized by GOPS. Meanwhile, other resources in the DCs such as storage and memory are assumed to be sufficient and not a bottleneck. It should be noted that the O/E/O conversions would occur in the DC site for data processing. Hence, in the corresponding switching node, there exists a data flow add/drop procedure. Thus, the end-to-end light paths for SFC requests are not required to stay in the same spectrum portion when connecting the distributed VNFs deployed across the DC sites. For each node pair in inter-DC EONs, we precalculate the shortest paths and obtain all the possible RMSA solutions on each path between two DC nodes, which are used as the ILP's input. In the following, we formulate an ILP model to solve the problem of delay-aware and resource-efficient SFC mapping in inter-DC EONs exactly. (As in other works [18–20], we consider K shortest paths instead of enumerating all possible candidate paths for every u–v pair because the gain from using those excessive long routing paths is very limited.)

Notations:

- $G(V, E)$: the substrate inter-DC EON.
- $\{R_i(s_i, d_i, C_i, b_i, D_i^{\text{thres}})\}$: set of SFC requests.
- $V_{i,j}$: set of feasible nodes for deploying VNF $c_{i,j}$.
- B_{FS}: bandwidth (gigahertz) of each FS.
- N: set of all available CPU cores on each DC node.
- C: set of all the possible VNF types. To instantiate a type c VNF in a DC would consume r_c CPU cores.
- CP: computing capacity (GOPS) of each CPU core.
- ς_c: capacity limit of a type c VNF, $\varsigma_c = CP{\cdot}r_c$
- φ_c: processing resource requirement of a type c VNF.
- F: number of FSs on each fiber link $e \in E$.
- M: set of modulation formats sorted by their spectrum efficiencies ML_m. Each mode $m \in M$ has a maximum TD TD_m.
- A: a very large number.
- $P_{u,v}$: set of K shortest routing paths from u to v, where $u, v \in V$ and $|P_{u,v}| = K$. The length of the path is denoted as $d_{u,v}$.
- $G_{i,j,m}$: set of available FS blocks that each contain $\lceil b_i/(ML_m \times B_{FS}) \rceil$ FSs on the path p connecting $v_j \to v_{j+1}$, where $p \in P_{v_j,v_{j+1}}$, $v_j \in V_{i,j}$, and $v_{j+1} \in V_{i,j+1}$. For generalization, we denote $v_0 = s_i$ and $v_{J_i+1} = d_i$.
- $W_{i,m,v_j,v_{j+1}}$: set of RMSA solutions for $v_j \to v_{j+1}$. Each element w is a tuple $\langle p, g \rangle$, i.e., a path $p \in P_{v_j,v_{j+1}}$ and an available FS block $g \in G_{i,j,m}$ on it, where $v_j \in V_{i,j}$, $v_{j+1} \in V_{i,j+1}$, and $j \in [0, J_i]$.
- W_i: set of RMSA solutions for connecting the source, the intermediate requested VNFs, and the destination of R_i, i.e., $W_i = \bigcup_{j\in[0,J_i]} W_{i,m,v_j,v_{j+1}}$.

Variables:

- x_{i,v_j}: Boolean variable that equals 1 if VNF $c_{i,j}$ chooses node $v_j \in V_{i,j}$, and 0 otherwise.
- $h_{v,c}$: Boolean variable that equals 1 if a type $c \in C$ VNF is deployed on node $v \in V$, and 0 otherwise.
- n_v: integer variable that indicates the number of used CPU cores on node $v \in V$.
- $D_{v,c}^{\text{proc}}$: the processing delay of VNF $c \in C$ on node $v \in V$.
- D_i^{map}: the total end-to-end mapping delay of request R_i.
- $t_{v,c}$: the number of times the VNF $c \in C$ is reused on node $v \in V$.
- $y_{i,j,w}$: Boolean variable that equals 1 if R_i chooses RMSA solution $w \in W_{i,m,v_j,v_{j+1}}$ to connect $v_j \to v_{j+1}$, and 0 otherwise.
- $z_{e,f}$: Boolean variable that equals 1 if the fth FS on link $e \in E$ is used, and 0 otherwise.
- $f^{\max}$: integer variable that indicates the MFSI.
- $L_{i,v,c}$: binary variable that equals 1 if $t_{v,c} = i$, $\forall v \in V$, $\forall c \in C$.

Objective:

$$\text{Minimize} \left(\frac{1}{|N| \cdot |V|} \sum_{v\in V} n_v + \frac{f^{\max}}{F} + \sum_i \frac{D_i^{\text{map}}}{D_i^{\text{thres}}} \right). \quad \textbf{(3)}$$

The ILP objective is to simultaneously minimize the consumed computing and spectrum resources and the end-to-end mapping delay. Here, the first term represents the normalized value of the total number of CPU cores used in the network, which we try to minimize to improve the efficiency of the IT resource utilization in the DCs. The second term reflects the normalized MFSI. A smaller MFSI indicates that the spectrum utilization of EON is more load-balanced. The last term reflects the normalized value of end-to-end mapping delay, and reducing it can result in a better user experience.

Constraints:

(1) VNF placement constraints:

$$\sum_{v_j \in V_{i,j}} x_{i,v_j} = 1, \forall i, \forall j \in [1, J_i], \tag{4}$$

$$h_{v_j,c_{i,j}} \geq x_{i,v_j}, \forall i, \forall j \in [1, J_i]. \tag{5}$$

Equation (4) ensures that each VNF in request R_i chooses one and only one DC for deployment, and Eq. (5) determines whether a type of $c_{i,j}$ VNF is deployed on DC v_j.

$$n_v = \sum_{c \in C} h_{v,c} \cdot r_c, \forall v \in V. \tag{6}$$

Equation (6) sums up the number of CPU cores used on each DC.

(2) RMSA related constraints

$$\sum_{w \in W_{i,m,v_j,v_{j+1}}} y_{i,j,w} = \begin{cases} x_{i,v_j} \cdot x_{i,v_{j+1}}, & v_j \neq v_{j+1} \\ 0, & v_j = v_{j+1} \end{cases}, \tag{7}$$
$$\forall i, \forall j \in [0, J_i], \forall v_j \in V_{i,j}, \forall v_{j+1} \in V_{i,j+1}.$$

Equation (7) ensures that each SFC request gets the proper RMSA solution to connect the source, intermediate VNFs, and destination. Note that one and only one RMSA solution should be selected to connect two adjacent VNFs in the request if they are on different substrate nodes; otherwise, no RMSA scheme should be used. Since Eq. (7) is a nonlinear constraint when $v_j \neq v_{j+1}$, it can be further linearized as

$$\begin{cases} x_{i,v_j} + x_{i,v_{j+1}} - 1 \leq \sum_{w \in W_{i,m,v_j,v_{j+1}}} y_{i,j,w} \leq \frac{1}{2}(x_{i,v_j} + x_{i,v_{j+1}}), \; v_j \neq v_{j+1} \\ \sum_{w \in W_{i,m,v_j,v_{j+1}}} y_{i,j,w} = 0, \; v_j = v_{j+1} \end{cases}, \tag{8}$$
$$\forall i, \forall j \in [0, J_i], \forall v_j \in V_{i,j}, \forall v_{j+1} \in V_{i,j+1}.$$

$$\sum_{w \in W_{i,m,v_j,v_{j+1}}} y_{i,j,w} \cdot TD_m \geq \sum_{w \in W_{i,m,v_j,v_{j+1}}} y_{i,j,w} \cdot d_{v_j,v_{j+1}},$$
$$\forall i, \forall j \in [0, J_i]. \tag{9}$$

Equation (9) ensures the proper modulation format $m \in M$ is chosen based on its distance limitation TD_m.

$$\sum_i \sum_{\substack{w \in W_i \\ w = <p, g> \\ e \in p, f \in g}} y_{i,j,w} = z_{e,f}, \forall e \in E, \forall f \in F, \tag{10}$$

$$f^{\max} \geq f \cdot z_{e,f}, \forall e \in E, \forall f \in F. \tag{11}$$

Equation (10) ensures the spectrum nonoverlapping constraint. By traversing all the links e in the network (E) and all the FSs f on each link $e \in E$, Eq. (11) can obtain the MFSI of the network.

(3) Delay-related constraints

$$t_{v,c} = \sum_{\substack{c = c_{i,j} \\ v = v_j}} x_{i,v_j}, \forall v \in V, \forall c \in C, \tag{12}$$

$$D_{v,c}^{\text{proc}} = \frac{1}{\varsigma_c - t_{v,c} \cdot \varphi_c}, \forall v \in V, \forall c \in C. \tag{13}$$

Equation (12) counts the value of the reuse times of each type c VNF on each DC. Then, Eq. (13) calculates the processing delay of each VNF instance on each DC based on the reuse time. Since Eq. (13) is a nonlinear constraint, we could linearize it by introducing an auxiliary variable $O_{v,c} = D_{v,c}^{\text{proc}} \cdot t_{v,c}$.

$$A \cdot \left(h_{v,c} - 1\right) \leq t_{v,c} - \sum_i i \cdot L_{i,v,c} \leq -A \cdot \left(h_{v,c} - 1\right),$$
$$\forall v \in V, \forall c \in C. \tag{14}$$

$$D_{v,c}^{\text{proc}} = \sum_i \frac{1}{i} \cdot O_{v,c} \cdot L_{i,v,c}, \forall v \in V, \forall c \in C, \tag{15}$$

$$\sum_i L_{i,v,c} = 1, \forall v \in V, \forall c \in C, \tag{16}$$

$$A \cdot \left(h_{v,c} - 1\right) \leq D_{v,c}^{\text{proc}} \cdot \varsigma_c - O_{v,c} \cdot \varphi_c - 1 \leq -A \cdot \left(h_{v,c} - 1\right),$$
$$\forall v \in V, \forall c \in C. \tag{17}$$

Equations (14)–(17) list the constraints for the variable $O_{v,c}$. Note that Eq. (15) is also a nonlinear constraint, which is further linearized by introducing an auxiliary variable $Q_{i,v,c} = O_{v,c} \cdot L_{i,v,c}$.

$$Q_{i,v,c} \leq O_{v,c}, \forall i, \forall v \in V, \forall c \in C, \tag{18}$$

$$Q_{i,v,c} \geq O_{v,c} + A \cdot \left(L_{i,v,c} - 1\right), \forall i, \forall v \in V, \forall c \in C, \tag{19}$$

$$Q_{i,v,c} \leq A \cdot L_{i,v,c}, \forall i, \forall v \in V, \forall c \in C, \tag{20}$$

$$A \cdot \left(h_{v,c} - 1\right) \leq D_{v,c}^{\text{proc}} - \sum_i \frac{1}{i} \cdot O_{i,v,c} \leq -A \cdot \left(h_{v,c} - 1\right),$$
$$\forall v \in V, \forall c \in C. \tag{21}$$

Equations (18)–(21) list the constraints for the variable $Q_{i,v,c}$.

$$D^{\text{node}}_{i,j} = \sum_{c=c_{i,j}} x_{i,v_j} \cdot D^{\text{proc}}_{v_j,c}, \forall i, \forall j \in [1, J_i]. \tag{22}$$

Equation (22) calculates the processing delay of each VNF node in request R_i. It is worth noting that Eq. (22) is also a nonlinear constraint. Next, we linearize it by introducing an auxiliary notation $a_{i,v_j,c}$, where $a_{i,v_j,c} = x_{i,v_j} \cdot D^{\text{proc}}_{v_j,c}$:

$$a_{i,v_j,c} \geq 0, \forall i, \forall j \in [1, J_i], \forall c \in C, \tag{23}$$

$$a_{i,v_j,c} \leq D^{\text{proc}}_{v_j,c}, \forall i, \forall j \in [1, J_i], \forall c \in C, \tag{24}$$

$$a_{i,v_j,c} \geq D^{\text{proc}}_{v_j,c} + A \cdot (x_{i,v_j} - 1), \forall i, \forall j \in [1, J_i], \forall c \in C, \tag{25}$$

$$a_{i,v_j,c} \leq A \cdot x_{i,v_j}, \forall i, \forall j \in [1, J_i], \forall c \in C. \tag{26}$$

Equations (23)–(26) list the constraints for variable $a_{i,v_j,c}$. Then, we convert the constraint Eq. (22) into constraint Eq. (27):

$$D^{\text{node}}_{i,j} = \sum_{c=c_{i,j}} a_{i,v_j,c}, \forall i, \forall j \in [1, J_i], \forall c \in C. \tag{27}$$

The propagation delay of the link between every two adjacent nodes is presented in Eq. (28):

$$D^{\text{link}}_{i,j} = \frac{\sum\limits_{w \in W_{i,m,v_j,v_{j+1}}} y_{i,j,w} \cdot d_{v_j,v_{j+1}}}{V_0 / R}, \forall i, \forall j \in [0, J_i], \tag{28}$$

where V_0 denotes the propagation speed of signals in the vacuum and R denotes the refractive index of optical fiber.

$$D^{\text{map}}_{i} = \sum_{j \in [1, J_i]} D^{\text{node}}_{i,j} + \sum_{j \in [0, J_i]} D^{\text{link}}_{i,j}, \forall i. \tag{29}$$

Equation (29) calculates the total end-to-end mapping delay of request R_i consisting of the total processing delay of the nodes and the total propagation delay of the links.

$$D^{\text{thres}}_{i} = \sum_{j \in [1, J_i]} \frac{1}{\varsigma_{c_{i,j}} - \alpha \cdot \varphi_{c_{i,j}}} + \beta, \forall i, \tag{30}$$

$$D^{\text{map}}_{i} \leq D^{\text{thres}}_{i}, \forall i. \tag{31}$$

The end-to-end delay threshold for each request is calculated according to Eq. (30); the former item represents the threshold for the node processing delay and the latter (β) represents the threshold for the propagation delay in units of microseconds. More specifically, according to Eq. (13), α represents the upper bound of the number of times a type $c_{i,j}$ VNF is reused. The end-to-end delay of each request should not exceed the sum of these two items. Finally, Eq. (31) ensures that the delay threshold is satisfied.

4. HEURISTIC ALGORITHMS

Many studies have shown that the SFC mapping problem is an NP-hard problem, which means that the computational complexity grows exponentially as the network size expands [7–12,18–23]. Hence, to cope with the high time complexity issue of the ILP model, particularly in large-scale scenarios, we introduce the DALB-MA to effectively allocate network resources to SFC requests while satisfying their delay threshold constraints. The general idea behind the DALB-MA is that it actively reuses VNFs while trying to keep the reuse time below a certain safety level, thus achieving a higher resource usage and lower latency simultaneously. Specifically, under light load conditions, the algorithm will energetically reuse the VNFs that have already been instantiated in the network. Because there are only a small number of requests and the DCs deployed with VNFs are light-loaded, reusing VNFs will improve IT resource utilization without causing significant VNF processing delay. When the traffic load increases and VNF reuse time exceeds the specified safety level, the algorithm will attempt to create a new VNF instance to avoid high processing latency caused by excessive VNF reuse on a single DC.

By considering the objective function and the constraints, the proposed DALB-MA is divided into three phases. The first phase, known as node mapping, involves selecting appropriate DCs to deploy or reuse the required VNFs sequentially. The second phase, known as link mapping, is to assign modulation format and spectrum resources to establish the light paths for connecting the two adjacent nodes in the SFC request. The last phase is to determine whether serving the request would significantly worsen the processing delay of the VNF instance on the DC, such that some of the served SFC requests fail to meet their end-to-end delay threshold. The main procedure of the DALB-MA is shown in Algorithm 1. For each SFC request R_i, we first find the K shortest paths between the source and destination node, and then we apply the three phases on the shortest path to map the request to the substrate network. If any of the three phases fails, the DALB-MA will try the next shortest path and so on and so forth. If none of the K paths can be used to map the request, i.e., either the resource requirement or delay threshold is not met, R_i will be blocked. Next, we elaborate on the node mapping scheme, link mapping scheme, and delay check algorithm, respectively.

Algorithm 1. DALB-MA

```
for each SFC request R_i(s_i, d_i, C_i, b_i, D_i^thres) do
  get K shortest paths connecting s_i and d_i
  for k = 1 to K do
    execute Node Mapping Scheme;
    if Node Mapping succeeds then
      execute Link Mapping Scheme;
      if Link Mapping succeeds then
        execute Delay Check Algorithm;
        if Delay Check passes then
          break;
        end
      end
    end
  end
end
```

A. Node Mapping Scheme

Algorithm 2 describes the node mapping scheme in detail. For each requested VNF node in SFC request R_i, Line 2 obtains a feasible DC node set $V_{i,j}$ for the jth VNF $c_{i,j}$. If $|V_{i,j}| > 0$, Lines 4–7 obtain the reuse times $t_{v,c_{i,j}}$ of VNF $c_{i,j}$ on each DC node $v \in V_{i,j}$ and store them in the set $T_{i,j}$. It should be noted that if no VNF $c_{i,j}$ is instantiated on DC v, $t_{v,c_{i,j}} = 0$. Lines 8–9 get the minimum value in $T_{i,j}$ as well as the DC node v' that corresponds to the minimum value. Then, Line 10 determines the minimum value and the size of $T_{i,j}$. If $MIN \neq 0$ or there is only one element remaining in $T_{i,j}$, Algorithm 2 proceeds to Line 17 to deploy or reuse the VNF $c_{i,j}$ on node v'. Otherwise, Algorithm 2 executes the loop body of Lines 10–16. The main idea of this part is that we want to get the minimum value except zero (*MEZ*) of $T_{i,j}$ and compare it with the given safety level. If *MEZ* ≥ *Safety level*, we select the DC that corresponds to the *MEZ* value and reuse the VNF $c_{i,j}$ that is already deployed on it. If *MEZ* ≥ *Safety level*, it means that the reuse time of VNF $c_{i,j}$ on the DC corresponding to the *MEZ* value has reached the specified safety level, which may cause an extremely high processing delay. To avoid this, we consider instantiating a new VNF $c_{i,j}$ on another DC. So far, the node mapping is successful. However, if $|V_{i,j}| = 0$, which means that there is no available DC node on the current path connecting s_i and d_i of request R_i, the node mapping will fail. In this case, Lines 19–20 will release the resource occupied by R_i and return to Line 3 of Algorithm 1 to consider the next candidate path and repeat Algorithm 2.

Next, we will elaborate on the role of the safety level in the algorithm. According to Line 10 and Line 13 of Algorithm 2, only if certain conditions are met can the safety level affect the selection of a DC for deploying VNF $c_{i,j}$. Figure 3 shows several illustrative examples of DC node selection. We assume that VNF $c_{i,j}$ is currently deployed with a safety level of 5 and there are three available DCs on the path. The following three cases are discussed:

(1) MIN $\neq$ 0

As is shown in Fig. 3(a), all DCs are deployed with VNF $c_{i,j}$, and the initial reuse time set is $T_{i,j} = <7, 6, 8>$, with the minimum value of 6 on DC#2. In this case, Algorithm 2 will skip the loop in Lines 10–16 and perform Line 17 directly, i.e., reuse the deployed VNF $c_{i,j}$ on DC#2. Note that since we did not execute the loop body, the selection of DC will not be affected by the safety level. In this case, we simply choose the DC with the minimum reuse time to achieve the load balancing.

(2) MIN = 0 and MEZ < Safety Level

Figure 3(b) illustrates the second example of node mapping, in which DC#1 and DC#3 are deployed with VNF $c_{i,j}$, with the reuse times of 2 and 6, respectively, and DC#2 has no VNF $c_{i,j}$ instantiated, so the initial reuse time set is $T_{i,j} = <2, 0, 6>$. The minimum value is 0 and *MEZ* = 2. Since *MEZ* < *Safety Level*, the algorithm will choose DC#1 to reuse the deployed VNF $c_{i,j}$.

(3) MIN = 0 and MEZ < Safety Level

The last example is illustrated in Fig. 3(c), where DC#1 and DC#3 are deployed with VNF $c_{i,j}$, with the reuse times of 5 and 6, respectively, and DC#2 has no VNF $c_{i,j}$ deployed, so the initial reuse time set is $T_{i,j} = <5, 0, 6>$. The minimum value is 0 and *MEZ* = 5. Since *MEZ* has reached the safety level, we should instantiate a new VNF $c_{i,j}$ on DC#2 to avoid the excessive reuse of VNF $c_{i,j}$ on DC#1 and DC#3.

As can be seen from the above discussion, the safety level of VNF reuse does not always play a role. It can affect the selection of DCs under certain conditions and can only reduce the number of VNF reuse to a certain extent, but it does not limit the maximum VNF reuse time in the network. The role of the safety level is more like a "warning value." If the reuse time of VNFs in DCs reaches this warning value, we should be cautious about further reusing it because the processing delay of the instance is already high, and further reusing it would risk blocking the requests. If the same type of VNF has not been instantiated in other available DCs, we should consider setting up a new VNF instance in them, exchanging resources for time efficiency.

Algorithm 2. Node Mapping Scheme

1 **for** $j \in [1, J_i]$ **do**
2   get feasible DC node set $V_{i,j}$;
3   **if** $|V_{i,j}| > 0$ **then**
4   **for** node $v \in V_{i,j}$ **do**
5   obtain the reuse times $t_{v,c_{i,j}}$ of VNF $c_{i,j}$ on DC node v;
6   put value $t_{v,c_{i,j}}$ into set $T_{i,j}$;
7   **end**
8   get $MIN = \min(T_{i,j})$;
9   find node $v' = \arg\min_{v \in V_{i,j}} (T_{i,j})$;
10   **while** $|V_{i,j}| > 1$ and $MIN = 0$ **do**
11   remove MIN from $T_{i,j}$;
12   get $MIN = \min(T_{i,j})$;
13   **if** $MIN \neq 0$ and $MIN <$ ***Safety Level*** **then**
14   find node $v' = \arg\min_{v \in V_{i,j}} (T_{i,j})$;
15   **end**
16   **end**
17   **Node Mapping succeeds** and deploy/reuse VNF $c_{i,j}$ on node v';
18   **else**
19   **Node Mapping fails** and release the resource occupied by R_i;
20   **break;**
21   **end**
22 **end**

B. Link Mapping Scheme

After all the VNF nodes in request R_i have been successfully mapped to the network, Algorithm 3 uses the link mapping scheme to properly connect the source, intermediate VNFs, and the destination of R_i. For each node pair (v_j, v_{j+1}), Line 2 gets the K' shortest paths connecting $v_j \rightarrow v_{j+1}$. For each shortest path p_k, Lines 4–7 find the optimal RMSA solution on the path and acquire the MFSI of the network using the above RMSA solution. Then, Lines 9–10 select the RMSA solution with the smallest MFSI and assign the spectrum resources on the corresponding path. Note that Lines 4–7 only

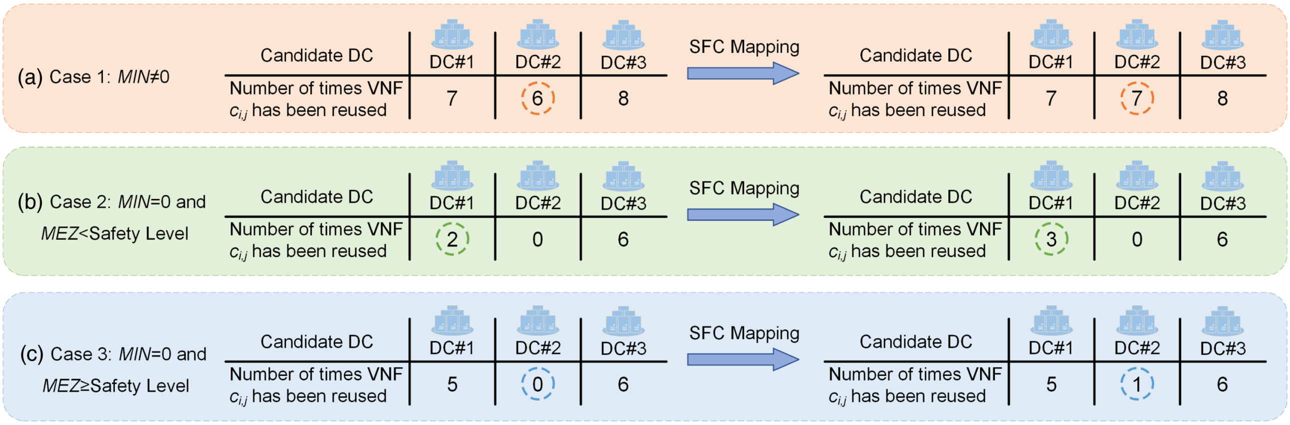

Fig. 3. Example of node mapping. (a) $MIN \neq 0$, (b) $MIN = 0$ and $MEZ <$ Safety Level, (c) $MIN = 0$ and $MEZ <$ Safety Level.

use the hypothetical deployment scheme on each path and do not occupy any spectrum resource on the substrate links. Only after Line 9 has decided on the final RMSA scheme will the spectrum resources connecting $v_j \rightarrow v_{j+1}$ be allocated. If no FS block is available on any of the K' shortest paths connecting $v_j \rightarrow v_{j+1}$, the link mapping fails, and we should release all the resources occupied by request R_i before repeating the node mapping scheme on the next shortest path from s_i to d_i.

C. Delay Check Algorithm

After mapping the request R_i to the substrate network, we need to ensure that all the deployed requests in the network continue to meet their delay threshold. Lines 2–6 of Algorithm 4 calculate the end-to-end delay D_t^{map} of each request R_t that has been deployed in the network and compare it to its delay threshold D_t^{thres}. If the end-to-end delay of any deployed request exceeds its delay threshold, the delay check of R_i fails. In this case, we should release all the resources occupied by R_i before returning to Line 3 of Algorithm 1 to consider the next shortest path from s_i to d_i and repeat Algorithm 2 on it.

Algorithm 3. Link Mapping Scheme

1 **for** $j = 0$ to J_i **do**
2   Obtain K' shortest paths connecting $v_j \rightarrow v_{j+1}$;
3   **for** $k = 1$ to K' **do**
4   choose the appropriate modulation format according to the length of path p_k;
5   calculate the number of required FSs;
6   find an RMSA solution with the **First-Fit Principle** according to the spectrum usage on path p_k;
7   acquire the MFSI with the above MFSI solution;
8   **end**
9   select the RMSA solution with the smallest MFSI;
10   assign spectrum resources with the corresponding RMSA solution;
11 **end**
12 **Link Mapping succeeds.**

D. Complexity Analysis

The time complexities of the algorithms are analyzed as follows:

In the node mapping scheme, each request has $O(|J_i|)$ VNF nodes for mapping, and each VNF node has $O(|V|)$ DCs for selection, so the time complexity of the node mapping scheme is $O(|J_i| \cdot |V|)$.

In the link mapping scheme, each request has $O(|J_i|)$ virtual links for mapping. For each link, we consider $O(|K'|)$ candidate paths to build the light paths. The time complexity of the node mapping scheme is $O(|J_i| \cdot |K'|)$.

Algorithm 4 calculates the end-to-end delay of all the requests that have already been deployed in the network. We assume that there is a total of $O(|R|)$ requests, and each request has $O(|J_i|)$ VNF nodes, so the time complexity of this part is $O(|J_i| \cdot |R|)$.

So far, the overall time complexity of the node mapping scheme, the link mapping scheme, and the delay check algorithm is $O(|J_i| \cdot (|V| + |K'| + |R|))$.

Finally, since the DALB-MA considers $O(|K|)$ shortest paths when deploying each SFC request, the overall time complexity of the DALB-MA is $O(|K| \cdot |J_i| \cdot (|V| + |K'| + |R|))$.

5. SIMULATION RESULTS

In this section, we perform simulations to evaluate the performance of the proposed ILP and heuristic algorithms. In the simulations, a small-scale six-node network [32] and a large-scale 28-node network topology U.S. backbone [33]

Algorithm 4. Delay Check Algorithm

1 **for** all the requests $R_t \in \Re$ that are already deployed **do**
2   calculate the end-to-end delay D_t^{map} of request R_t;
3   **if** $D_t^{\text{map}} > D_t^{\text{thres}}$ **then**
4   **Delay Check failed** and release the resource occupied by R_i;
5   **return;**
6   **end**
7 **end**
8 **Delay Check passed.**

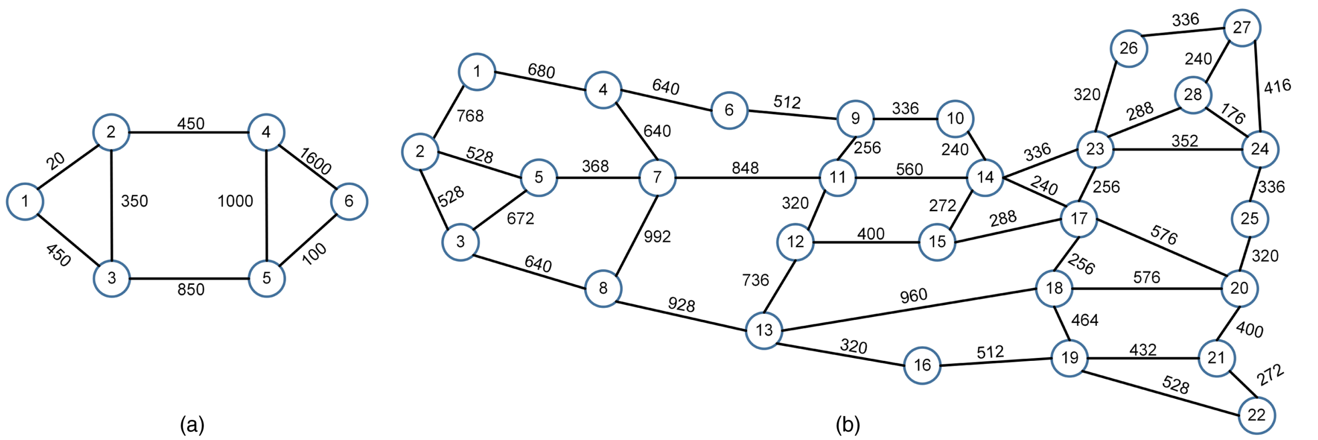

Fig. 4. Topologies used in simulations. (a) Six-node topology, (b) 28-node U.S. backbone topology.

Table 1. Simulation Parameters

	Number of DCs	Types of VNFs ($\lvert C_i \rvert$)	Capacity Limit of Each VNF Instance (ς_c)	Processing Resource Requirement of Each VNF (φ_c)	Length of SFC Request ($\lvert C_i \rvert$)	Bandwidth Requirement (b_i)	Available CPU Cores in Each DC ($\lvert N \rvert$)	Available FSs on Each Fiber Link ($\lvert F \rvert$)
Six-node network	6	3	[20,60] GOPS	[1,3] GOPS	[1,2]	[10,100] Gbps	6	15
28-node network	20	8	[20,160] GOPS	[1,8] GOPS	[1,3]	[20,200] Gbps	36	300

are used, as shown in Fig. 4. In the figure, the numbers on the line represent the link length in kilometers. Four modulation formats are considered in our simulations: BPSK, QPSK, 8-QAM, and 16-QAM. The TD of the four modulation formats is determined based on the experimental results reported in [34,35]. Since we are addressing the static network planning in this paper, we assume the static SFC requests all arrive at once. And once successfully provisioned, they do not expire in the network. Each SFC request is modeled as $R_i(s_i, d_i, C_i, b_i, D_i^{\text{thres}})$, as introduced in Section 3.B. The source node s_i and the destination node d_i are chosen randomly from the network. The length of the VNF sequence ($|C_i|$), the types of VNFs in C_i (i.e., $|C|$), the capacity limit of each VNF instance (ς_c), the processing resource requirement of each VNF (φ_c), and the bandwidth requirement (b_i) are given in Table 1. The end-to-end delay threshold of each request (D_i^{thres}) is given by Eq. (30), where α and β are set to be 15 and 10, respectively. The computing capacity of each CPU core is set to be 20 GOPS. Other detailed simulation parameters are also summarized in Table 1.

For comparison, two benchmark heuristics are considered, i.e., the modified shortest-path and batch algorithm (MSBA) and the shortest-path random algorithm (SRA) [21]. For each request, the MSBA first selects the DC with the minimum overall reuse times on the shortest path from s_i to d_i. In this way, it can distribute VNF instances to different DCs as much as possible to reduce processing delay. Note that if there is no available DC on the shortest path, the MSBA will try the second shortest path and so on and so forth. The VNFs are then instantiated/reused accordingly. Finally, Algorithm 3 is used to connect the VNFs and Algorithm 4 is used to check the delay time. The SRA is similar to the MSBA, except that it randomly selects the DCs on the shortest path to instantiate/reuse VNFs.

A. Performance Evaluation Parameters

1. Average Cost

$$AC = \frac{1}{|N| \cdot |V|} \cdot \sum_{v \in V} n_v + \frac{f^{\max}}{F} + \sum_i \frac{D_i^{\text{map}}}{D_i^{\text{thres}}}. \tag{32}$$

The average cost (AC) is defined in Eq. (32), which is the same as the ILP objective in Section 3. As we can see from the previous discussion, we want to jointly reduce the resource consumption and end-to-end delay of requests in the network, so the smaller the AC, the better the performance of the algorithm.

2. Number of CPUs Used

$$\text{NCPU} = \sum_{v \in V} n_v. \tag{33}$$

The total number of CPUs (NCPU) used is defined in Eq. (33), where n_v represents the number of used CPUs on each node $v \in V$.

3. MFSI

$$\text{MFSI} = \max_{e \in E} f_e^{\max}. \tag{34}$$

The MFSI of the network can be defined as Eq. (34), where $f_e^{\max}$ is the maximum frequency index on link e. By traversing all the links in the network (E), the MFSI is acquired.

4. Delay Margin Gain

$$\mathrm{DMG} = \sum_i \frac{D_i^{\text{thres}} - D_i^{\text{map}}}{D_i^{\text{thres}}}. \tag{35}$$

The delay margin gain (DMG) is defined in Eq. (35), where D_i^{thres} and D_i^{map} represent the delay threshold and the actual end-to-end delay after the request is mapped to the network, respectively. We want to increase the margin between the delay and the threshold as much as possible because a larger margin usually means a relatively smaller processing delay of the VNF instance. A larger DMG indicates that the VNF instances can be reused by more requests, resulting in better utilization of IT resources.

B. Performance Evaluation under a Small-Scale Network Scenario

Figure 5(a) depicts the results of the AC, which is calculated by Eq. (32). We find that the ILP achieves the lowest costs, followed by the DALB-MA. The benchmark heuristics, the MSBA and SRA, have higher costs. This is because the ILP model is the mathematical model that can obtain the optimal solution for SFC deployment in a small-scale network topology. According to the optimization objective Eq. (3), the ILP model jointly considers the reduction of IT resources in DCs, spectrum resources on the link, and end-to-end delay of requests to obtain the lowest cost. Meanwhile, the DALB-MA considers the impact of VNF reuse on VNF processing delay when selecting DCs for VNF deployment, resulting in the DALB-MA solution being the closest to the optimal ILP solution. However, because the DALB-MA only restricts the reuse times under certain conditions and does not consider delay as an optimization target, the cost of the DALB-MA has not reached the optimal solution. The MSBA considers the relationship between VNF reuse and processing delay when selecting DCs on the shortest path. It consolidates all the VNF nodes of request R_i onto a single DC with the fewest overall reuse times. This can help to reduce the processing delay, but it will consume more IT resources because more VNFs will be instantiated in the network. The SRA randomly selects the DCs on the shortest path from source to destination, optimizing only the spectrum resources without considering IT resources or end-to-end delay, resulting in poor SRA performance.

C. Performance Evaluation under a Large-Scale Network Scenario

Figure 5(b) shows the simulation results of an SFC request block rate using the U.S. backbone network topology. The results indicate that the DALB-MA offers the lowest request block rate. This is because the DALB-MA adopts a load-balancing strategy in heavy load cases. It will select the DC with the fewest VNF reuse times to reduce the processing delay and, as a result, the block rate of SFC requests. The block rate of the MSBA is similar to the DALB-MA, and much lower than the SRA. This is because the MSBA selects the DCs with the minimum overall VNF reuse times, so it will try to instantiate new VNFs as much as possible, hence balancing the traffic load as well.

Figure 5(c) depicts the results of the heuristic algorithms for deploying SFC requests in the U.S. backbone network in terms of the AC. The results are comparable to those obtained in the small-scale network, and the cost from the DALB-MA is 7%–22% and 10%–16% less than that of the MSBA and SRA, respectively.

Figure 6(a) shows the results of the CPU cores used in the network. It is clear that the DALB-MA has remarkable advantages in this regard, and they are more pronounced under light load conditions. The reason for this is that under light load conditions, VNF reuse times are relatively small that have not reached the safety level. Therefore, the DALB-MA will choose to reuse the existing VNF instances as much as possible to reduce CPU usage and improve IT resource utilization. When the number of requests increases, the DALB-MA will try to deploy a new VNF instance in an empty DC only if the reuse times of the corresponding VNF in other DCs reach the safety level. Hence, the DALB-MA consumes the fewest CPU cores in the network. In contrast, the MSBA has the worst performance in terms of the number of CPU cores used. This is because the MSBA selects the DC with the smallest reuse times to deploy the SFC requests all the time, so it will try to instantiate as many new VNFs as possible, resulting in more CPU cores being used.

Figure 6(b) gives out the results on the MFSI. In general, three heuristic algorithms produce similar results in this regard because they use the same link mapping scheme to construct light paths connecting the source and destination of the SFC requests.

Figure 6(c) shows the results on the DMG calculated by Eq. (35). Under light load, the DALB-MA obtains the smallest DMG value, while as traffic increases, the DMG performance of the DALB-MA gradually catches up and eventually achieves the suboptimal result. As discussed, because the reuse times of VNF instances have not reached the safety level in this case, the DALB-MA will try to reuse the instantiated VNFs as much as possible under light load, resulting in a higher processing delay and a lower DMG value. When the number of SFC requests increases, the VNF reuse time reaches the safety level, and the DALB-MA will try to instantiate new VNFs in empty DCs to deploy the VNFs in a more load-balanced manner, which leads to a lower processing delay and a higher DMG compared with the SRA.

D. Impact of the Safety Level on the DALB-MA under a Large-Scale Network Scenario

As introduced, to achieve a better trade-off between the VNF reuse and VNF processing delay, the safety level is adopted in the DALB-MA to flexibly control the timing of instantiating new VNF instances in the network. It is clear that the assigned safety level would have a significant impact on the performance of the DALB-MA. Figure 7(a) shows how the number of CPU

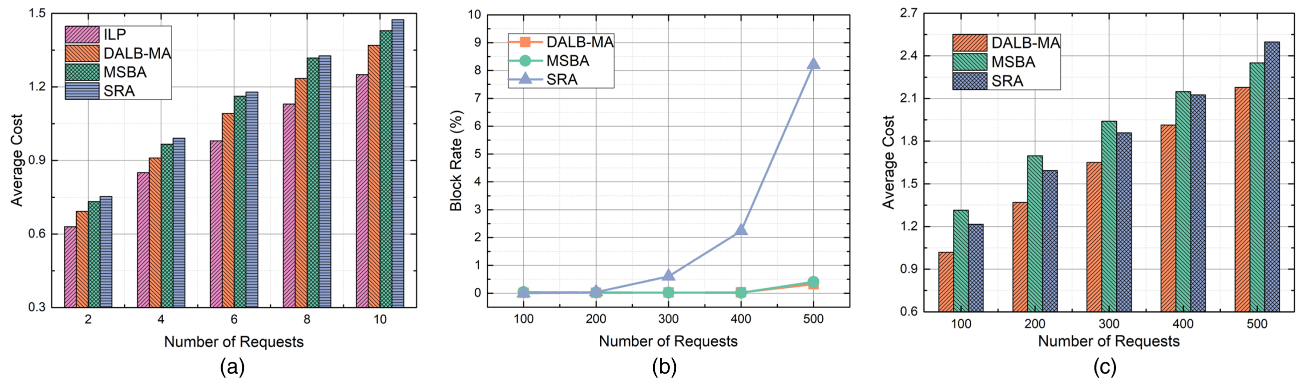

Fig. 5. (a) Simulation results of the AC in a small-scale six-node network, (b) simulation results of the block rate in a large-scale 28-node U.S. backbone topology, (c) simulation results of the AC in a large-scale 28-node U.S. backbone topology.

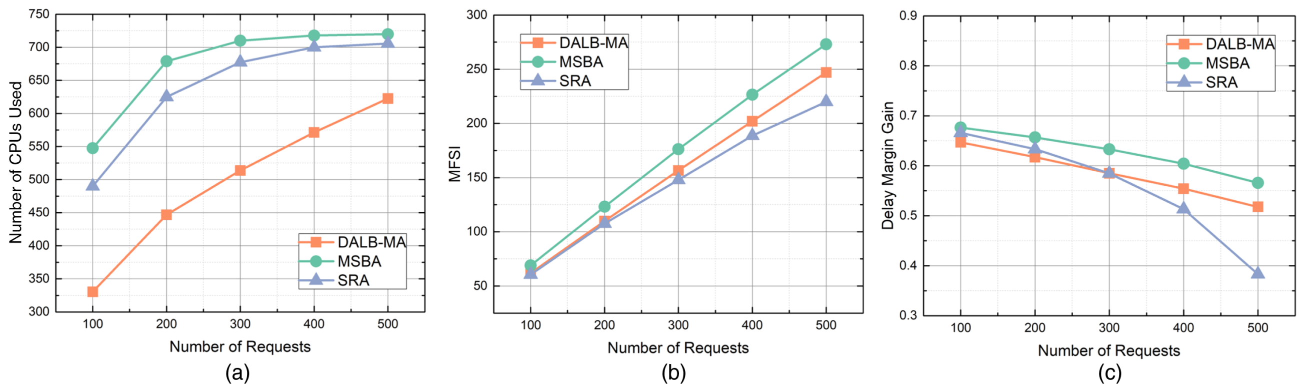

Fig. 6. Simulation results of the heuristic algorithms in a large-scale 28-node U.S. backbone topology. (a) Number of CPUs used, (b) MFSI, (c) DMG.

cores used changes with the safety level under various traffic loads. Under light load conditions, the curve's trend is seen to first decrease and then flatten. Taking the case of 100 requests, for example, we observe that when the safety level takes the value between 1 and 6, the number of CPUs used drops sharply as the safety level rises because the higher the safety level, the more reuse times are permitted, resulting in fewer VNF instances being deployed in the network. Meanwhile, since the overall demand for VNFs is relatively low under light load conditions, there exists an upper bound for the reuse time of VNFs. In our simulations, we found that after deploying 100 requests to the substrate network, the reuse times of most VNF instances in DCs are below 6. Hence, if a safety level greater than 6 is assigned, the reuse of VNF instances will not be affected by it. In other words, the safety level does not affect the number of CPUs used, and the curve becomes flat. When traffic load increases, the curve's trend first flattens and then decreases. In the case of 500 requests, the number of CPU cores used remains stable when the safety level is set between 1 and 3. The reason is that, since the overall demand for VNFs is relatively high under heavy load conditions, VNF instances are more likely to be reused by multiple SFC requests. In our simulations, we found that, after deploying all 500 requests to the substrate network, the reuse times of most VNFs in DCs are greater than 3. Hence, when a safety level below 3 is assigned, whether or not to instantiate new VNFs will not be affected by it. In other words, the safety level has no effect on the number of CPUs used, and the curve is flat. When the safety level is increased further, i.e., when the value exceeds 3, the reuse of VNFs is more likely to occur, and thus fewer VNF instances are deployed in the network and the number of CPUs used decreases.

Figure 7(b) shows the impact of the safety level on the MFSI of the network. Since the safety level is a parameter that mainly affects the selection of DCs for VNF deployment, it has little effect on the MFSI.

Figure 7(c) depicts the variation curve of the DMG versus the safety level under various traffic loads. It is clear that the DMG trend is similar to that of used CPU cores. This is because the DMG and the number of CPUs used are both inversely proportional to the VNF reuse times, which are directly related to the safety level.

Figure 8 shows the effect of the safety level on the AC. According to Eq. (32), the AC is determined by the three indices discussed in Fig. 7, so the trend of the variation curve mainly depends on them. The results indicate that, after reaching a certain safety level, the AC tends to be stable. Combined with Fig. 7(a) and Fig. 7(b), we can conclude that for different cases, the safety level should be set accordingly. For example, if the delay requirement of SFC requests is more stringent, a lower safety level should be assigned to instantiate more VNFs so that the processing delay can be minimized. Instead, if the

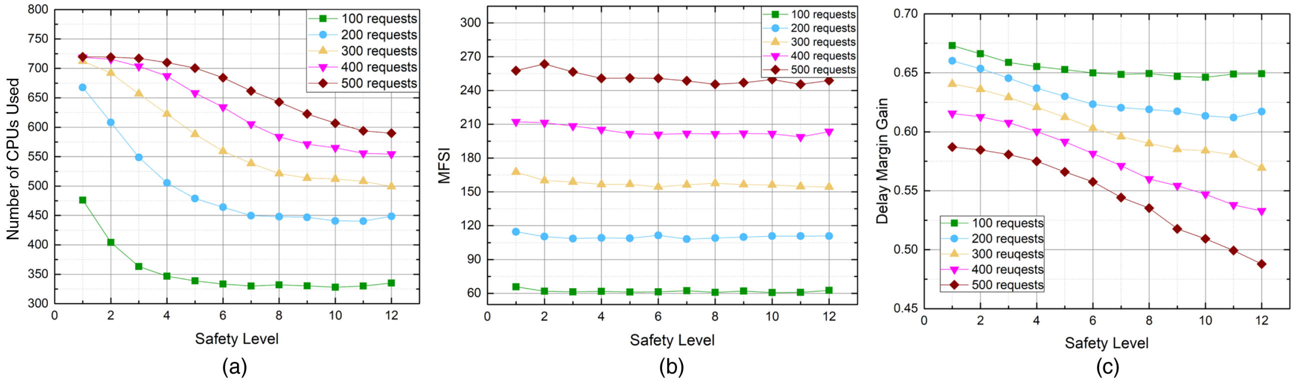

Fig. 7. Simulation results of the impact of the safety level on the DALB-MA in a large-scale 28-node U.S. backbone topology. (a) Number of CPUs used, (b) MFSI, (c) DMG.

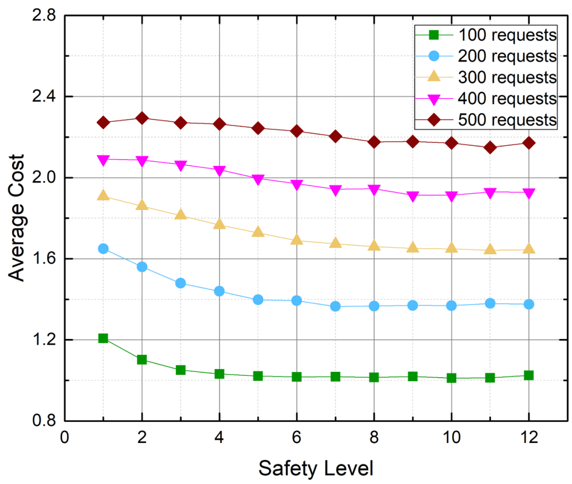

Fig. 8. Simulation results of the impact of the safety level on the AC of the DALB-MA in a large-scale 28-node U.S. backbone topology.

resource efficiency is more important, a higher safety level should be assigned for VNF instance reuse so that more IT resources can be saved.

6. CONCLUSION

In this paper, we investigated the resource-efficient SFC mapping problem in inter-DC EONs with the consideration of the end-to-end delay threshold. An ILP model and a delay-aware and resource-efficient SFC deployment heuristic were proposed, in which a load balancing policy was used to reduce the required computing resources through VNF instance reuse. Meanwhile, a given safety level of reuse times was introduced to reduce the VNF processing delay. Simulation results show that our proposed ILP model and DALB-MA outperformed the benchmarks. Furthermore, the different settings of the safety level would impact the performance of the DALB-MA in many aspects. It should be determined according to the actual need and the traffic load conditions.

Funding. Jiangsu Provincial Key Research and Development Program (BE2020012); Transformation Program of Scientific and Technological Achievements of Jiangsu Province (BA2019026).

REFERENCES

1. R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, “Network function virtualization: state-of-the-art and research challenges,” IEEE Commun. Surv. Tutorials **18**, 236–262 (2016).
2. A. M. Medhat, T. Taleb, A. Elmangoush, G. A. Carella, S. Covaci, and T. Magedanz, “Service function chaining in next generation networks: state of the art and research challenges,” IEEE Commun. Mag. **55**(2), 216–223 (2016).
3. T. Shen, M. Zhu, J. Gu, X. Ren, B. Chen, and C. Shi, “When virtual network functions are deployed in network resource virtualized elastic optical networks,” Proc. SPIE **11435**, 114350J (2020).
4. L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, “Joint optimization of service function chaining and resource allocation in network function virtualization,” IEEE Access **4**, 8084–8094 (2016).
5. M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, “Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies,” IEEE Commun. Mag. **47**(11), 66–73 (2009).
6. H. Feng, J. Llorca, A. M. Tulino, D. Raz, and A. F. Molisch, “Approximation algorithms for the NFV service distribution problem,” in *IEEE INFOCOM—IEEE Conference on Computer Communications* (2017).
7. M. Ghaznavi, N. Shahriar, S. Kamali, R. Ahmed, and R. Boutaba, “Distributed service function chaining,” IEEE J. Sel. Areas Commun. **35**, 2479–2489 (2017).
8. J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, “On dynamic service function chain deployment and readjustment,” IEEE Trans. Netw. Service Manage. **14**, 543–553 (2017).
9. T. Kuo, B. Liou, K. C. Lin, and M. Tsai, “Deploying chains of virtual network functions: on the relation between link and server usage,” IEEE/ACM Trans. Netw. **26**, 1562–1576 (2018).
10. H. Li, L. Wang, X. Wen, Z. Lu, and J. Li, “MSV: an algorithm for coordinated resource allocation in network function virtualization,” IEEE Access **6**, 76876–76888 (2018).
11. L. Qu, C. Assi, and K. Shaban, “Delay-aware scheduling and resource optimization with network function virtualization,” IEEE Trans. Commun. **64**, 3746–3758 (2016).
12. A. Alleg, T. Ahmed, M. Mosbah, R. Riggio, and R. Boutaba, “Delay-aware VNF placement and chaining based on a flexible resource allocation approach,” in *13th International Conference on Network and Service Management (CNSM)* (2017).
13. S. Yang, F. Li, R. Yahyapour, and X. Fu, “Delay-sensitive and availability-aware virtual network function scheduling for NFV,” IEEE Trans. Services Comput. **15**, 188–201 (2022).
14. G. Sun, G. Zhu, D. Liao, H. Yu, X. Du, and M. Guizani, “Cost-efficient service function chain orchestration for low-latency applications in NFV networks,” IEEE Syst. J. **13**, 3877–3888 (2019).
15. G. Sun, Z. Xu, H. Yu, X. Chen, V. Chang, and A. V. Vasilakos, “Low-latency and resource-efficient service function chaining

orchestration in network function virtualization," IEEE Internet Things J. **7**, 5760–5772 (2020).
16. G. Sun, R. Zhou, J. Sun, H. Yu, and A. V. Vasilakos, "Energy-efficient provisioning for service function chains to support delay-sensitive applications in network function virtualization," IEEE Internet Things J. **7**, 6116–6131 (2020).
17. H. Yu, Z. Chen, G. Sun, X. Du, and M. Guizani, "Profit maximization of online service function chain orchestration in an inter-datacenter elastic optical network," IEEE Trans. Netw. Service Manage. **18**, 973–985 (2021).
18. W. Fang, M. Zeng, X. Liu, W. Lu, and Z. Zhu, "Joint spectrum and IT resource allocation for efficient VNF service chaining in inter-datacenter elastic optical networks," IEEE Commun. Lett. **20**, 1539–1542 (2016).
19. M. Zeng, W. Fang, and Z. Zhu, "Orchestrating tree-type VNF forwarding graphs in inter-DC elastic optical networks," J. Lightwave Technol. **34**, 3330–3341 (2016).
20. Y. Wang, P. Lu, W. Lu, and Z. Zhu, "Cost-efficient virtual network function graph (vNFG) provisioning in multidomain elastic optical networks," J. Lightwave Technol. **35**, 2712–2723 (2017).
21. B. Li, W. Lu, S. Liu, and Z. Zhu, "Deep-learning-assisted network orchestration for on-demand and cost-effective VNF service chaining in inter-DC elastic optical networks," J. Opt. Commun. Netw. **10**, D29–D41 (2018).
22. B. Li, W. Lu, and Z. Zhu, "Deep-NFVOrch: leveraging deep reinforcement learning to achieve adaptive vNF service chaining in DCI-EONs," J. Opt. Commun. Netw. **12**, A18–A27 (2020).
23. H. Xuan, S. Wei, Y. Feng, D. Liu, and Y. Li, "Bi-level programming model and algorithm for VNF deployment with data centers placement," IEEE Access **7**, 185760 (2019).
24. C. Peng, D. Zheng, and X. Cao, "Latency-aware VNF protection for network function virtualization in elastic optical networks," in *IEEE Global Communications Conference (GLOBECOM)* (2021).
25. S. Troia, R. Alvizu, and G. Maier, "Reinforcement learning for service function chain reconfiguration in NFV-SDN metro-core optical networks," IEEE Access **7**, 167944 (2019).
26. S. Fichera, R. Martínez, B. Martini, M. Gharbaoui, R. Casellas, R. Vilalta, R. Muñoz, and P. Castoldi, "Latency-aware resource orchestration in SDN-based packet over optical flexi-grid transport networks," J. Opt. Commun. Netw. **11**, B83–B96 (2019).
27. M. Gharbaoui, B. Martini, G. Cecchetti, and P. Castoldi, "Resource orchestration strategies with retrials for latency-sensitive network slicing over distributed telco clouds," IEEE Access **9**, 132801 (2021).
28. X. Chen, Z. Zhu, R. Proietti, and S. J. B. Yoo, "On incentive-driven VNF service chaining in inter-datacenter elastic optical networks: a hierarchical game-theoretic mechanism," IEEE Trans. Netw. Service Manage. **16**, 1–12 (2019).
29. X. Chen, Z. Zhu, J. Guo, S. Kang, R. Proietti, A. Castro, and S. J. B. Yoo, "Leveraging mixed-strategy gaming to realize incentive-driven VNF service chain provisioning in broker-based elastic optical inter-datacenter networks," J. Opt. Commun. Netw. **10**, A232–A240 (2018).
30. J. Gu, M. Zhu, T. Shen, C. Shi, and X. Ren, "Delay-aware and resource-efficient VNF-service chain deployment in inter-datacenter elastic optical networks," in *Optical Fiber Communication Conference (OFC)* (2021), paper W1G.4.
31. F. Musumeci, O. Ayoub, M. Magoni, and M. Tornatore, "Latency-aware CU placement/handover in dynamic WDM access-aggregation networks," J. Opt. Commun. Netw. **11**, B71–B82 (2019).
32. R. Lin, S. Luo, J. Zhou, S. Wang, A. Cai, W.-D. Zhong, and M. Zukerman, "Virtual network embedding with adaptive modulation in flexi-grid networks," J. Lightwave Technol. **36**, 3551–3563 (2018).
33. L. Gong, X. Zhou, X. Liu, W. Zhao, W. Lu, and Z. Zhu, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," J. Opt. Commun. Netw. **5**, 836–847 (2013).
34. A. Bocoi, M. Schuster, F. Rambach, M. Kiese, C. Bunge, and B. Spinnler, "Reach-dependent capacity in optical networks enabled by OFDM," in *Optical Fiber Communication Conference (OFC)* (2009), paper OMQ4.
35. M. Schuster, A. Bocoi, M. Kiese, F. Rambach, C. Bunge, and B. Spinnler, "Advantage of reach-dependent capacity in optical networks enabled by OFDM," in *ITG Symposium on Photonic Networks* (2009).