Shortest Path Tour Problem Based Integer Linear Programming for Service Chaining in NFV Networks

Masahiro Sasabe
Graduate School of Science and Technology
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan
m-sasabe@ieee.org

Takanori Hara
Graduate School of Science and Technology
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan
hara.takanori.hm8@is.naist.jp

***Abstract*—Network functions virtualization (NFV) is a new paradigm to achieve flexible and agile network services by decoupling network functions from proprietary hardware and running them on generic hardware as virtual network functions (VNFs). In the NFV network, a certain network service can be modeled as a sequence of VNFs, called a service chain. Given a connection request (origin node, destination node, and service chain requirement, which is a sequence of functions), the service chaining problem aims to find an appropriate service path, which starts from the origin and ends with the destination while executing the VNFs at the intermediate nodes in the required order. Some existing work noticed that the service chaining problem was similar to the shortest path tour problem (SPTP). To the best of our knowledge, this is the first work that exactly formulates the service chaining problem as the SPTP-based integer linear programming (ILP). Through numerical results, we show the SPTP-based ILP formulation can support about two times larger scale systems than the existing ILP formulation.**

***Index Terms*—Network functions virtualization (NFV), service chaining, integer linear programming (ILP), shortest path Tour problem (SPTP).**

I. Introduction

Network function virtualization (NFV) can realize flexible and agile network services by decoupling network functions (e.g., routing, firewall, deep packet inspection, and load balancing) from dedicated hardware (e.g., appliances and middleboxes) and running them on generic hardware as virtual network functions (NFVs) [1], [2]. In addition to the flexibility and agility, the NFV can also reduce both capital expenditures (CAPEX) and operating expenditures (OPEX) because it can be built on the generic hardware and liberate operators from learning knowledge for the conventional dedicated hardware.

The resource allocation is one of the challenging issues in the NFV networks. We can view a certain network service as a sequence of VNFs, called a service chain. Given a connection request with origin node, destination node, and service chain requirement, which is a sequence of functions, the service chaining problem aims to find an appropriate service path, which starts from the origin node and ends with the destination node while executing the corresponding VNFs at the intermediate nodes in the required order.

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI (C) under Grant 19K11942, Japan.

Some existing work noticed the similarity between the shortest path tour problem (SPTP) [3] and the service chaining problem [4], [5]. The SPTP is a variant of the shortest path problem (SPP) and tries to find a shortest path from an origin node to a destination node such that the path must visit a sequence of disjoint node subsets $\mathcal{T}_1, \ldots, \mathcal{T}_K$ in this order. For example, $\mathcal{T}_k$ $(k = 1, \ldots, K)$ can be regarded as sightseeing spots in a certain tour. Bhat and Rouskas first showed the possibility of modeling the service chaining problem as the SPTP [4]. Gao and Rouskas proposed the SPTP-based heuristic algorithm for the online service chaining [5].

Recently, Andrade and Saraiva proposed the integer linear programming (ILP) formulation for the constraint SPTP [6]. The constrained SPTP is a special case of the SPTP such that the path does not include repeated edges. In this paper, combining this approach and a novel network model called augmented network, we will show that the service chaining problem can exactly be modeled as the SPTP-based ILP. Although there are various types of ILP formulations for the service chaining [7]–[12], to the best of our knowledge, this is the first work that exactly formulates the service chaining problem as the SPTP-based ILP. Through numerical results, we show the proposed ILP formulation can support about two times larger scale systems than the existing ILP formulation.

The rest of the manuscript is organized as follows. Section II gives the related work. After introducing the system model in Section III, we develop the SPTP-based ILP for the service chaining in NFV networks. We demonstrate that the SPTP-based ILP is more scalabe than the existing ILP formulation in Section V. Finally, Section VI gives the conclusions and future work.

II. Related Work

A. NFV and Its Resource Allocation Problems

The resource allocation is one of the most challenging problems in NFV networks because it requires to deal with various types of requirements (e.g., service (VNFs) chaining,

VNF placement, scheduling, and demand provisioning) as well as their complex dependence. The recent survey on NFV networks can be found in [1], [2].

Service chaining is one of the major NFV resource allocation problems, which aims to find an appropriate service path under the constraints of function placement and service chain requirement (e.g., processing/bandwidth demand and a sequence of functions). It is a kind of combinatorial optimization problems and there are multiple studeis on the mathematical formulation of the service chaining problem [7]–[12]. Nguyen et al. formulated the service chaining problem as ILP over an expanded network to minimize the blocking probability of service chain requests [7]. Huin et al. developed ILP over a layered graph to minimize the total bandwidth usage [8]. Gupta et al. also formulated ILP to minimize the total bandwidth usage [9]. Savi et al. modeled ILP to minimize the number of active nodes by considering the processing-related costs, i.e., context switching costs and upscaling costs, in a VNF consolidation scenario where multiple VNFs were served at the same hardware [10]. Nejad et al. [11] aimed to maximize the total revenue. Hyodo et al. formulate ILP to minimize the placement cost and bandwidth usage under the relaxation of VNF order and non-loop constraints [12].

The path-selection based service chaining aims to find an appropriate service path from given path candidates. D'Oro et al. applied the non-cooperative game theory to achieve service chaining a distributed manner [13]. They also proposed a two-stage Stackelberg game composed of leading servers and following users to achieve the distributed resource allocation and orchestration [14].

Some researchers noticed that the service chaining is similar to the conventional SPTP [4], [5]. Bhat and Rouskas showed that service chaining could be modeled as SPTP [4]. They proposed a heuristic algorithm to solve the SPTP and compared it with existing algorithms. Gao et al. proposed an online path-selection algorithm to minimize the maximum network congestion [5]. To the best of our knowledge, this is the first work that exactly formulates the SPTP-based ILP for the service chaining.

B. Shortest Path Tour Problem

The SPTP is a variant of the SPP where some intermediate nodes (locations) should to be visited in a predefined order [3].

Festa et al. achieved polynomial-time reduction of the SPTP to a classical SPP over a modified digraph and proposed heuristics to solve the SPTP [15]. Ferone et al. studied constrained SPTP, a variant of the SPTP, where the path did not include repeated links, proved that it belonged to the complexity class of NP-complete, and proposed a branch-and-bound based heuristic [16]. Ferone et al. further proposed a mathematical model and new branch-and-bound heuristics for the CSPTP [17]. Recently, Andrade and Saraiva developed the ILP formulation for the constrained SPTP [6]. In this paper, we formulate the SPTP-based ILP for the service chaining, with the help of the existing approach [6] and the development of a novel network model, called augmented network. (We will give the detail of the augmented network in Section III-B.)

III. System Model

A. Service Chain Request

We consider an NFV network where connection requests randomly arrive as in [7]. The orchestrator waits for C ($C = 1, \ldots, C_{\max}$) requests and solves the service chaining problem for the collected requests $\mathcal{C}$. The way of processing the request(s) can be categorized into three types, i.e., online, batch, and offline, depending on the size of C, i.e., C. In case of the online processing ($C = 1$), the orchestrator serves the connection request just after its arrival. If the orchestrator knows all the requests ($C = C_{\max}$) in advance, it solves the service chaining problem for them at once, and thus this results in the offline processing. In other cases, i.e., $C = 2, \ldots, C_{\max} - 1$, the orchestrator adopts the batch processing with size of C. In what follows, we focus on the online processing type ($C = 1$) as in [5], [7]. Note that the proposed approach in this paper can also be extended to support other processing types.

A connection c has a service chain requirement that consists of origin node o_c, destination node d_c, sequence of K_c functions $\mathcal{R}_c = (f_{c,1}, \ldots, f_{c,K_c})$, bandwidth capacity per physical link b_c, processing capacity for packet transfer per physical node p_c^{node}, and processing capacity for execution of function $f_{c,k}$, $p_{c,f_{c,k}}^{\mathrm{func}}$. o_c (resp. d_c) is a physical node that connection c starts from (resp. ends with). $\mathcal{R}_c$ is a sequence of K_c functions $(f_{c,1}, \ldots, f_{c,K_c})$ with which connection c will be served in this order. In general, connection c may require a more complex processing order rather than the sequential order, e.g., split and merge. In this paper, we mainly focus on the sequential type of requirement, which results in the simple yet novel ILP formulation of the service chaining problem. As for the bandwidth/processing capacity requirement, we use the same assumptions in [7]. We consider b_c is a fixed value, e.g., constant bit rate (CBR). Whenever connection c passes through a physical node, it needs a processing capacity p_c^{node} to process the packets through the node. In addition, connection c requires processing capacity $p_{c,f_{c,k}}^{\mathrm{func}}$ for executing kth function $f_{c,k}$ in $\mathcal{R}_c$. An example of the service chain requirement for connection c, i.e., $(o_c, d_c) = (v_1, v_5)$, $\mathcal{R}_c = (f_{c,1}, f_{c,2}, f_{c,3}) = (f_2, f_3, f_1)$, b_c, p_c^{node}, and $\{p_{c,f}^{\mathrm{func}}\}_{f \in \mathcal{R}_c}$, is shown in the top layer of Fig. 1.

Given a connection request c with origin node o_c, destination node d_c, and service chain requirement $\mathcal{R}_c$, which is a sequence of functions, we try to find an appropriate service path $\mathcal{S}_c$, which starts from o_c and ends with d_c while executing functions in $\mathcal{R}_c$.

B. Augmented Network

We consider a physical network $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ (resp. $\mathcal{E}$) is the set of physical nodes (resp. links), as shown in the middle layer of Fig. 1. G is a directed graph where an arrow from node $i \in \mathcal{V}$ to $j \in \mathcal{V}$ expresses the physical link (i, j). For example, the bidirectional arrow between nodes v_1 and v_2

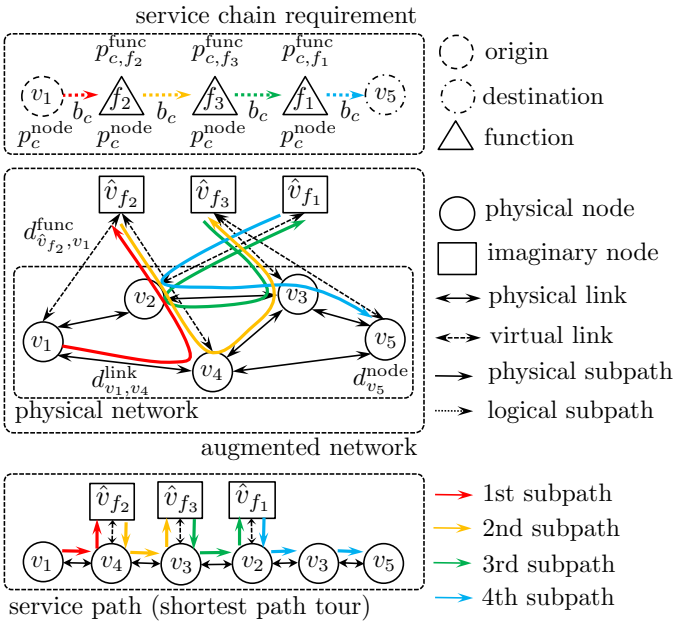

Fig. 1: Overview of service chaining: relationship between service chain requirement, augmented network, and service path.

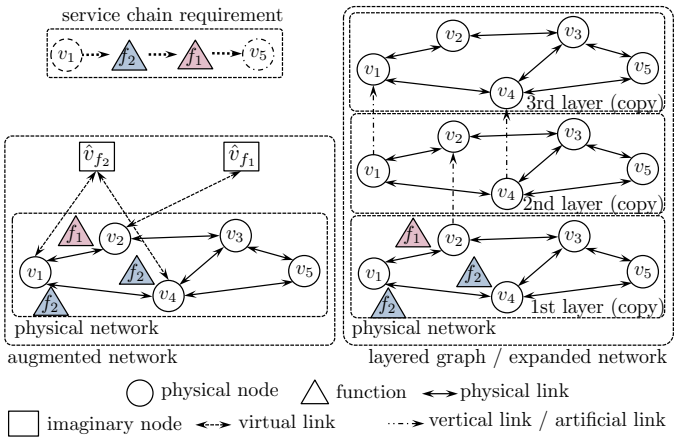

Fig. 2: Structure comparison among network models.

TABLE I: Scale comparison among network models.

network model	# of nodes	# of links
layered graph / expanded network	$(K_c+1)V$	$(K_c+1)E+K_cN$
augmented network	$V+K_c$	$E+2K_cN$

indicates that there are two physical links between them, i.e., (v_1, v_2) and (v_2, v_1), in Fig. 1. Each physical link $(i,j) \in \mathcal{E}$ (resp. each physical node $i \in \mathcal{V}$) has residual bandwidth $B_{i,j}$ (resp. residual processing capacity P_i) at arrival of c. The NFV network supports a set of F functions, $\mathcal{F} = \{f_1, \ldots, f_F\}$, in the whole network while each physical node $i \in \mathcal{V}$ can support part of functions, $\mathcal{F}_i \subseteq \mathcal{F}$. Each function f is supported by N_f $(N_f > 0)$ physical nodes. In actual systems, N_f should be carefully determined according to the demand for network services. This results in the problems of demand provisioning and function placement, which will be considered in the future work. In what follows, we assume that N_f is identical, i.e., $N_f = N$ $(N > 0)$.

A connection c requests a service chain requirement $\mathcal{R}_c$, which includes a sequence of K_c functions, $\mathcal{R}_c = (f_{c,1}, \ldots, f_{c,K_c})$, from origin node o_c to destination node d_c. For example, in Fig. 1, $\mathcal{R}_c = (f_2, f_3, f_1)$ and $(o_c, d_c) = (v_1, v_5)$. Service chaining for connection c is finding an appropriate service path $\mathcal{S}_c$, which starts from o_c and ends with d_c while executing functions of $\mathcal{R}_c$ in this order and passing through the corresponding physical links. Inspired by the approach in [6], we can regard service chaining as SPTP. (The detail will be given in Section IV.) The SPTP tries to find a shortest path from an origin node to a destination node with the constraint that the path should visit at least one node from K $(K > 0)$ given disjoint node subsets $\mathcal{T}_1, \ldots, \mathcal{T}_K$ [15]. In our case, $\mathcal{T}_k = \{\hat{v}_{f_{c,k}}\}$ $(k \in \mathcal{K}_c)$, where $\hat{v}_{f_{c,k}}$ is an imaginary node supporting kth function $f_{c,k}$ in $\mathcal{R}_c$. For example, in Fig. 1, $\mathcal{T}_1 = \{\hat{v}_{f_2}\}, \mathcal{T}_2 = \{\hat{v}_{f_3}\}$, and $\mathcal{T}_3 = \{\hat{v}_{f_1}\}$. In our case, the disjoint node subsets can be regarded as a set of imaginary nodes, $\widehat{\mathcal{V}} = \{\hat{v}_{c,1}, \ldots, \hat{v}_{c,K_c}\}$, where kth imaginary node $\hat{v}_{c,k}$ supports kth function $f_{c,k}$ in $\mathcal{R}_c$. Note that imaginary nodes are not actual virtual nodes serving the corresponding functions but they play a key role in formulating service chaining as the SPTP-based ILP. (The detail of formulation will be given in Section IV.) We further introduce two sets of virtual links, $\widehat{\mathcal{E}}^{\text{in}}$ and $\widehat{\mathcal{E}}^{\text{out}}$. $\widehat{\mathcal{E}}^{\text{in}}$ is a set of links incoming to imaginary nodes, $\widehat{\mathcal{E}}^{\text{in}} = \{(v, \hat{v}_f) \mid v \in \mathcal{V}, \hat{v}_f \in \widehat{\mathcal{V}}, f \in \mathcal{F}_v\}$. On the other hand, $\widehat{\mathcal{E}}^{\text{out}}$ is a set of links outgoing from imaginary nodes, $\widehat{\mathcal{E}}^{\text{out}} = \{(\hat{v}_f, v) \mid \hat{v} \in \widehat{\mathcal{V}}, v \in \mathcal{V}, f \in \mathcal{F}_v\}$. For example, in Fig. 1, the virtual link $(\hat{v}_{f_2}, v_1)$ indicates that physical node v_1 can support function f_2. We call the finally obtained network the *augmented network* $G^+ = (\mathcal{V}^+, \mathcal{E}^+)$ where $\mathcal{V}^+ = \mathcal{V} \cup \widehat{\mathcal{V}}$ and $\mathcal{E}^+ = \mathcal{E} \cup \widehat{\mathcal{E}}^{\text{in}} \cup \widehat{\mathcal{E}}^{\text{out}}$. An example of the augmented network of G is shown in the middle layer of Fig. 1. For simplicity in description, the neighbors of node i in G^+ is defined as $\mathcal{V}_i^+ \subset \mathcal{V}^+$.

One of the main contributions of this paper is the proposal of the augmented network, which is much smaller than the existing network models: layered graph [8] and expanded network [7], and contributes to the simple yet exact SPTP-based ILP formulation. Fig. 2 illustrates the structure comparison among the three network models. The layered graph, which is given in the right of Fig. 2, consists $K_c + 1$ layers of the original physical network and multiple vertical links connecting two successive layers. A vertical link exists between node $i \in \mathcal{V}$ at kth layer and that at $(k+1)$th layer $(k = 1, \ldots, K_c)$ only when node i supports kth function of the chain request c, $f_{c,k}$. The expanded network is an extension of the layered graph. In the expanded network, an vertical (artificial) link only exists between nodes when the corresponding link and both the end nodes have enough residual bandwidth and processing capacity supporting the connection c.

Table I presents the scale comparison among the network

models. Note that the number of links in case of the expanded network is the upper limit where all the nodes and links are available for the connection request. We can confirm that the augmented network is much smaller than the layered graph and expanded network.

C. Service Path

According to the SPTP, the service path $\mathcal{S}_c$ for $\mathcal{R}_c = (f_{c,1}, \ldots, f_{c,K_c})$ with origin o_c and destination d_c can be expressed by a sequence of K_c+1 subpaths, $(\mathcal{S}_{c,1}, \ldots, \mathcal{S}_{c,K_c+1})$, where kth subpath $\mathcal{S}_{c,k}$ has origin node $a_{c,k}$ and destination node $b_{c,k}$, which are given as follows:

$$(a_{c,k}, b_{c,k}) = \begin{cases} (o_c, \hat{v}_{f_{c,1}}), & k = 1, \\ (\hat{v}_{f_{c,k-1}}, \hat{v}_{f_{c,k}}), & k = 2, \ldots, K_c, \\ (\hat{v}_{f_{c,K_c}}, d_c), & k = K_c + 1. \end{cases}$$

$\mathcal{S}_{c,k}$ starts from its origin node $a_{c,k}$ and ends with its destination node $b_{c,k}$ through appropriate physical and virtual links in G^+. (Finding an optimal service path $\mathcal{S}_c^*$ is our goal and will be discussed in Section IV.) Note that there is no loop in each subpath but loop(s) may occur in the whole path, that is, some links may be used multiple times. For example, the bottom layer of Fig. 1 shows an example of service path $\mathcal{S}_c = (\mathcal{S}_{c,1}, \ldots, \mathcal{S}_{c,4})$ where $\mathcal{S}_{c,1} = ((v_1, v_4), (v_4, \hat{v}_{f_2}))$ (red arrow), $\mathcal{S}_{c,2} = ((\hat{v}_{f_2}, v_4), (v_4, v_3), (v_3, \hat{v}_{f_3}))$ (orange arrow), $\mathcal{S}_{c,3} = ((\hat{v}_{f_3}, v_3), (v_3, v_2), (v_2, \hat{v}_{f_1}))$ (green arrow), and $\mathcal{S}_{c,4} = ((\hat{v}_{f_1}, v_2), (v_2, v_3), (v_3, v_5))$ (blue arrow). We can also confirm service path $\mathcal{S}_c$ in the augmented network, as shown in the middle layer of Fig.1. In this case, each subpath (i.e., $\mathcal{S}_{c,1}, \mathcal{S}_{c,2}, \mathcal{S}_{c,3}$, and $\mathcal{S}_{c,4}$) does not have any loop while the whole service path $\mathcal{S}_c$ has a loop. (Physical link (v_2, v_3) is used twice.)

We consider that the optimality of service path $\mathcal{S}_c$ is evaluated by total delay consisting of processing delay at nodes and propagation delay at physical links included in $\mathcal{S}_c$. (The detail will be given in Section IV.) Each physical link $(i, j) \in \mathcal{E}$ has propagation delay $d_{i,j}^{\text{link}}$ ($d_{i,j}^{\text{link}} > 0$). As mentioned above, connection c requires packet processing at each physical node that it uses, and thus the corresponding processing delay at physical node $i \in \mathcal{V}$ is given by d_i^{node} ($d_i^{\text{node}} > 0$). In $\mathcal{S}_c$, each function $f \in \mathcal{R}_c$ is executed at physical node $v \in \mathcal{V}$ with processing delay of $d_{\hat{v}_f,v}^{\text{func}}$ ($d_{\hat{v}_f,v}^{\text{func}} > 0$).

Table II summarizes the notations used in this paper.

TABLE II: Notations in the model.

Notation	Definition
G	Physical network $G = (\mathcal{V}, \mathcal{E})$
$\mathcal{V}$	Set of physical nodes, $V = \lvert\mathcal{V}\rvert$
$\mathcal{E}$	Set of physical links, $E = \lvert\mathcal{E}\rvert$
$\mathcal{F}$	Set of functions, $\mathcal{F} = \{f_1, \ldots, f_F\}$, $F = \lvert\mathcal{F}\rvert$
$\mathcal{F}_i$	Set of functions contained in physical node $i \in \mathcal{V}$
c	Connection with origin o_c and destination d_c
$\mathcal{R}_c$	Sequence of functions $(f_{c,1}, \ldots, f_{c,K_c})$ required by c
$\mathcal{K}_c, \mathcal{K}_c^+$	$\mathcal{K}_c = \{1, \ldots, K_c\}, \mathcal{K}_c^+ = \{1, \ldots, K_c + 1\}$
$\widehat{\mathcal{V}}$	Set of imaginary nodes, $\widehat{\mathcal{V}} = \{\hat{v}_f\}_{f \in \mathcal{F}}$, where imaginary node $\hat{v}_f$ supports function f
$\widehat{\mathcal{E}}^{\text{in}}$	Set of links incoming to imaginary nodes, $\widehat{\mathcal{E}}^{\text{in}} = \{(v, \hat{v}) \mid v \in \mathcal{V}, \hat{v}_f \in \widehat{\mathcal{V}}, f \in \mathcal{F}_v\}$
$\widehat{\mathcal{E}}^{\text{out}}$	Set of links outgoing from imaginary nodes, $\widehat{\mathcal{E}}^{\text{out}} = \{(\hat{v}, v) \mid \hat{v}_f \in \widehat{\mathcal{V}}, v \in \mathcal{V}, f \in \mathcal{F}_v\}$
G^+	Augmented network of G, $G^+ = (\mathcal{V}^+, \mathcal{E}^+)$
$\mathcal{V}^+$	Set of nodes in G^+, $\mathcal{V}^+ = \mathcal{V} \cup \widehat{\mathcal{V}}$, $V^+ = \lvert\mathcal{V}^+\rvert$
$\mathcal{E}^+$	Set of links in G^+, $\mathcal{E}^+ = \mathcal{E} \cup \widehat{\mathcal{E}}^{\text{in}} \cup \widehat{\mathcal{E}}^{\text{out}}$
$\mathcal{S}_c$	Service path for c, $(\mathcal{S}_{c,1}, \ldots, \mathcal{S}_{c,K_c+1})$
$\mathcal{S}_{c,k}$	kth sub service path with origin $a_{c,k}$ and destination $b_{c,k}$
b_c	Bandwidth requirement of c
p_c^{node}	Processing requirement of c for traversing a node
$p_{c,f_{c,k}}^{\text{func}}$	Processing requirement of $f_{c,k} \in \mathcal{R}_c$ at a node
$B_{i,j}$	Residual bandwidth of link (i, j) at arrival of c
P_i	Residual processing capacity of node i at arrival of c
$d_{i,j}^{\text{link}}$	Propagation delay of link $(i, j) \in \mathcal{E}$
d_i^{node}	Traversal delay of node $i \in \mathcal{V}$
$d_{\hat{v}_f,v}^{\text{func}}$	Processing delay of function f of node $v \in \mathcal{V}$
$x_{i,j}^{c,k}$	Binary decision variables: 1: if link (i, j) is included in $\mathcal{S}_{c,k}$, 0: otherwise

IV. Modeling Service Chaining as Shortest Path Tour Problem Based Integer Linear Programming

Inspired by the SPTP, we formulate the service chaining as the following ILP P, which has the binary decision variables $x_{i,j}^{c,k}$ ($c \in \mathcal{C}, (i, j) \in \mathcal{E}^+, k \in \mathcal{K}_c^+$):

$$x_{i,j}^{c,k} = \begin{cases} 1, & \text{if physical/virtual link } (i, j) \text{ is used in } k\text{th} \\ & \text{subpath of service path for connection } c, \\ 0, & \text{otherwise.} \end{cases}$$

$$\min \quad \sum_{(i,j) \in \mathcal{E}^+} d_{i,j} \sum_{k \in \mathcal{K}_c^+} x_{i,j}^{c,k} \tag{1}$$

$$\text{s.t.} \quad x_{i,j}^{c,k} = \{0, 1\}, \quad (i, j) \in \mathcal{E}^+, k \in \mathcal{K}_c^+, \tag{2}$$

$$\sum_{(a_{c,k},j) \in \mathcal{E}^+} x_{a_{c,k},j}^{c,k} = 1, \quad k \in \mathcal{K}_c^+, \tag{3}$$

$$\sum_{(j,b_{c,k}) \in \mathcal{E}^+} x_{j,b_{c,k}}^{c,k} = 1, \quad k \in \mathcal{K}_c^+, \tag{4}$$

$$\sum_{(i,j) \in \mathcal{E}^+} x_{i,j}^{c,k} = \sum_{(j,i) \in \mathcal{E}^+} x_{j,i}^{c,k}, \quad \forall i \in \mathcal{V} \setminus \{a_{c,k}\}, \forall j \in \mathcal{V} \setminus \{b_{c,k}\}, k \in \mathcal{K}_c^+, \tag{5}$$

$$x_{i,\hat{v}_{f_{c,k}}}^{c,k} = x_{\hat{v}_{f_{c,k}},i}^{c,k+1}, \quad \forall (i, \hat{v}_{f_{c,k}}) \in \widehat{\mathcal{E}}^{\text{in}}, \forall (\hat{v}_{f_{c,k}}, i) \in \widehat{\mathcal{E}}^{\text{out}}, k \in \mathcal{K}_c, \tag{6}$$

$$x_{i,\hat{v}_{f_{c,m}}}^{c,k} = 0, \quad \forall (i, \hat{v}_{f_{c,m}}) \in \widehat{\mathcal{E}}^{\text{in}}, k \in \mathcal{K}_c^+, m \neq k, \tag{7}$$

$$b_c \sum_{k \in \mathcal{K}_c^+} x_{i,j}^{c,k} \leq B_{i,j}, \quad \forall (i, j) \in \mathcal{E}, \tag{8}$$

$$p_c^{\text{node}} \sum_{(v,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}_c^+} x_{v,j}^{c,k} + \sum_{(\hat{v}_f,v) \in \widehat{\mathcal{E}}^{\text{out}}} p_{c,f}^{\text{func}} \sum_{k \in \mathcal{K}_c^+} x_{\hat{v}_f,v}^{c,k} \leq P_v, \quad \forall v \in \mathcal{V}. \tag{9}$$

Equation (1) is the objective function where $d_{i,j}$ is given as

follows:

$$d_{i,j} = \begin{cases} d_i^{\text{node}} + d_{i,j}^{\text{link}}, & \text{if } (i,j) \in \mathcal{E}, \\ d_{i,j}^{\text{func}}, & \text{if } (i,j) \in \widehat{\mathcal{E}}^{\text{out}}, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

We first observe that the objective function (1) is the same as that of the SPTP. From the viewpoint of service chaining, equation (10) indicates that passing each physical link $(i,j) \in \mathcal{E}$ suffers both forwarding delay d_i^{node} and propagation delay $d_{i,j}^{\text{link}}$. The service path also suffers the execution delay of each function $i \in \mathcal{R}_c$ at physical node j. As a result, the objective function (1) can be rewritten by

$$\sum_{(i,j)\in\mathcal{E}} (d_i^{\text{node}} + d_{i,j}^{\text{link}}) \sum_{k\in\mathcal{K}_c^+} x_{i,j}^{c,k} + \sum_{(\hat{v}_f,v)\in\widehat{\mathcal{E}}^{\text{out}}} d_{\hat{v}_f,v}^{\text{func}} \sum_{k\in\mathcal{K}_c^+} x_{\hat{v}_f,v}^{c,k},$$

where the first (resp. second) term corresponds to the physical forwarding and propagation delay (resp. function execution delay). For example, in the bottom of Fig. 1, the horizontal (resp. vertical) arrows correspond to the physical (resp. virtual) links included in service path $\mathcal{S}_c$. Note that the objective function can be replaced with conventional ones, e.g., minimization of the total bandwidth usage.

Equations (2)–(9) give the constraints. Constraint (2) represents the decision variables. Constraints (3)–(5) present the flow rules in each subpath $\mathcal{S}_{c,k}$ $(k \in \mathcal{K}_c^+)$. Constraint (3) (resp. Constraint (4)) indicates that the origin (resp. destination) node $a_{c,k}$ (resp. $b_{c,k}$) of the subpath k has incoming (resp. outgoing) flow. Equation (5) is the flow conservation constraint in the subpath k $(k \in \mathcal{K}_c^+)$. For example, focusing on the 1st subpath in Fig. 1, we observe that the flow occurs at physical node v_1, i.e., $\sum_{(v_1,j)\in\mathcal{E}^+} x_{v_1,j}^{c,1} = 1$, goes through any physical node $v \in \mathcal{V}$, i.e,. $\sum_{(v,j)\in\mathcal{E}^+} x_{v,j}^{c,1} = \sum_{(j,v)\in\mathcal{E}^+} x_{j,v}^{c,1}$, and finally ends with imaginary node $\hat{v}_{f_2}$, i.e., $\sum_{(j,\hat{v}_{f_2})\in\mathcal{E}^+} x_{j,\hat{v}_{f_2}}^{c,1} = 1$.

Constraint (6) guarantees the connectivity between two successive subpaths $\mathcal{S}_{c,k}$ and $\mathcal{S}_{c,k+1}$ $(k \in \mathcal{K}_c^+)$. More specifically, $(k+1)$th subpath should start from the same physical node as the last physical node of kth subpath. For example, focusing on the 1st and 2nd subpaths in Fig. 1, we observe that $x_{i,v_{f_{c,1}}}^{c,1} = x_{v_{f_{c,1}},i}^{c,2}$ $(\forall (i,v_{f_{c,1}}) \in \widehat{\mathcal{E}}^{\text{in}}, \forall (v_{f_{c,1}},i) \in \widehat{\mathcal{E}}^{\text{out}})$ where $f_{c,1} = f_2$. Constraint (7) prohibits imaginary node $\hat{v}_{f_{c,m}}$ from being used in mth subpath $(m \neq k)$. For example, focusing on the 1st subpath in Fig. 1, we observe that $x_{i,\hat{v}_{f_{c,m}}}^{c,1} = 0$ $(m \neq 1, \forall (i,\hat{v}_{f_{c,m}}) \in \widehat{\mathcal{E}}^{\text{in}})$.

Equation (8) gives the physical link capacity constraint where the bandwidth consumption of physical link $(i,j) \in \mathcal{E}$ should be equal or less than the residual bandwidth capacity $B_{i,j}$. Similarly, equation (9) shows the processing capacity constraint where the total processing load of physical node $v \in \mathcal{V}$ should be equal or less than the processing capacity P_v. The processing load consists of the traversal cost, $p_c^{\text{node}} \sum_{(v,j)\in\mathcal{E}} \sum_{k\in\mathcal{K}_c^+} x_{v,j}^{c,k}$, and the processing cost, $\sum_{(\hat{v}_f,v)\in\widehat{\mathcal{E}}^{\text{out}}} p_{c,f}^{\text{func}} \sum_{k\in\mathcal{K}_c^+} x_{\hat{v}_f,v}^{c,k}$.

V. Numerical Results

In this section, we compare the proposed SPTP-based ILP formulation and the existing ILP formulation [7] in terms of the computational complexity. We solve the both ILP formulations by using the existing solver CPLEX 12.8 [18] running on the server with Intel Xeon E7-8895v3 (18 cores and 2.60 GHz) and 2 TB memory.

A. Evaluation Scenario

To evaluate the fundamental characteristics of the proposed scheme, we first use the physical network consisting of 200 physical nodes where each physical node $i \in \mathcal{V}$ has the same processing capacity of $P_i = 1.71$. We then generate physical links between two arbitrary physical nodes at the probability of $\pi = 0.032$ by using the existing scheme [19]. All physical links between nodes i and j $(i,j \in \mathcal{V})$ have the same link capacity of $B_{i,j} = 1.14$. We set the physical link delay between nodes i and j, $d_{i,j}^{\text{link}}$, to be 10 [ms]. Each physical node i has the same traversal and processing delay, i.e., $d_i^{\text{node}} = 1$ [ms] and $d_i^{\text{func}} = 50$ [ms], respectively.

We set the total number of functions, F, to be 20 and each function can be supported by five physical nodes $(N = 5)$. In this paper, we focus on the online scenario $(C = 1)$ where the orchestrator immediately performs the service chaining per connection request, as in [7]. For each connection c, the origin node o_c and destination node d_c are randomly chosen from the physical nodes. Each connection c requires K_c functions, each of which is randomly chosen from the F functions. We set the bandwidth requirement, processing requirement for traversing a node, and processing requirement of $f_{c,k} \in \mathcal{R}_c$ at a node as follows: $b_c = 0.1, p_c^{\text{node}} = 0.05$, and $p_{c,f_{c,k}}^{\text{func}} = 0.1$.

We evaluate the computational complexity in terms of the execution time and CPLEX deterministic time. The execution time is the actual time required to solve the problem. Note that CPLEX supports the parallel optimization and we set the number of threads to be 32. Since the execution time depends on the hardware spec of the server, we also use the CPLEX deterministic time to evaluate the substantial complexity of the problem. CPLEX provides us with a choice between determinism and opportunism algorithms [20]. The determinism algorithm gives us a solution path in a deterministic manner while the opportunism one takes advantage of opportunities to improve performance. We adopt the determinism algorithm.

To show the scalability of the SPTP-based ILP in terms of the computation complexity, we compare it with the existing ILP for the service chaining in [7]. Note that the authors proposed not only the ILP formulation but also several heuristic algorithms to overcome the computation complexity, in [7]. In addition, the objective function is the minimization of the link and node utilization to alleviate the blocking probability of connection requests. In what follows, focusing on the computation complexity of the ILP formulation itself, we slightly replace the objective function of the existing ILP with that of the SPTP-based ILP, i.e., the minimization of the total delay given by Equation (1).

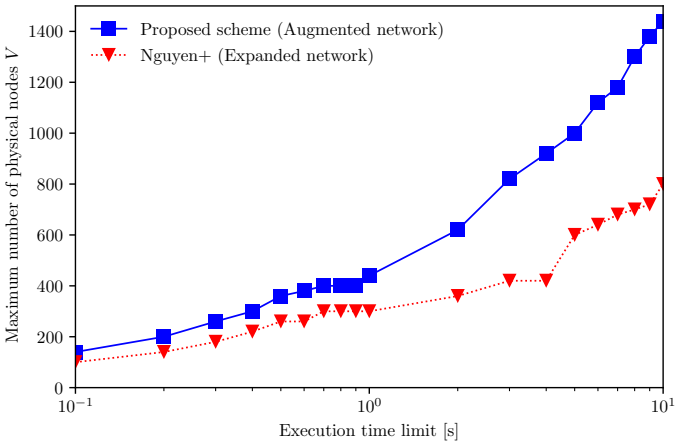

Fig. 3: The maximum number of physical nodes to be solved within the execution time limit.

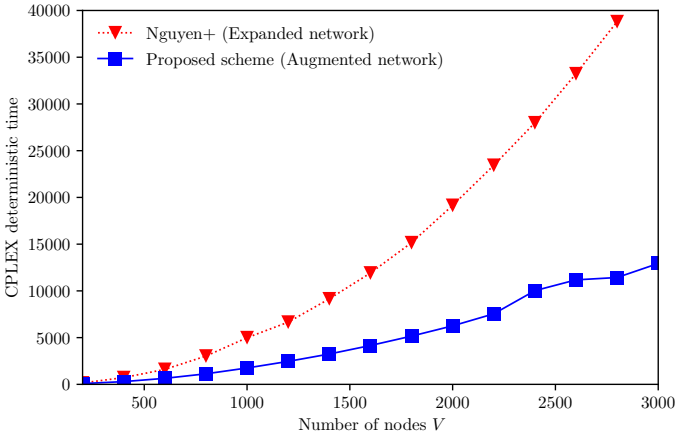

Fig. 4: Impact of the number of nodes on deterministic time.

In what follows, we show the average of 100 independent numerical experiments.

B. Impact of The Number of Nodes on Computational Complexity

Fig. 3 illustrates the maximum number of physical nodes that the SPTP-based ILP and existing ILP can solve within the execution time limit. We observe that the SPTP-based ILP can solve 1.33–2.19 times larger scale systems than the existing ILP under the same execution time limit. In particular, the SPTP-based ILP can support 440 (resp. 1440) nodes within 1 (resp. 10) [s].

Fig. 4 presents how the deterministic time increases with the number of nodes, V. Note that we limit the deterministic time to 40,000 [ticks]. We observe that the deterministic time of both ILP exponentially grows with increase of V, but the increasing rate of the SPTP-based ILP is much smaller than that of the existing ILP. As a result, the SPTP-based ILP can solve the service chaining even in case of $V = 3000$ while the existing ILP reaches the limitation of the deterministic time. As mentioned in Section III-B, the augmented network is more compact than the expanded network, which contributes to the scalability.

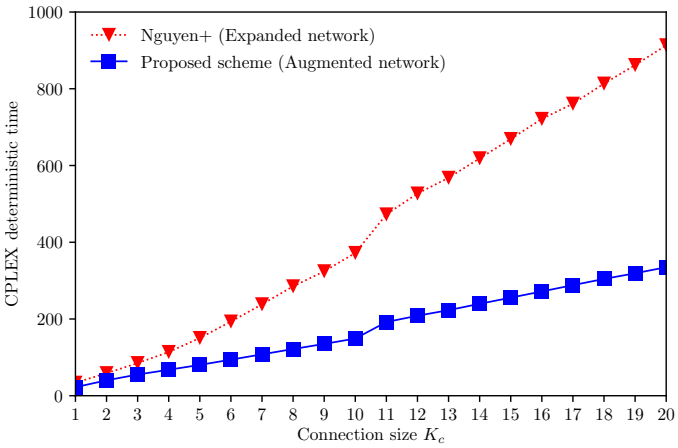

Fig. 5: Impact of connection size K_c on deterministic time.

C. Impact of Connection Size on Computational Complexity

Next, Fig. 5 shows the impact of the connection size K_c on the deterministic time when $V = 200$. We first observe that the deterministic time of both ILP shows almost linear growth with increase of K_c. We also confirm that the deterministic time of the SPTP-based ILP is always smaller than that of the existing ILP. Specifically, the SPTP-based ILP can reduce the deterministic time by 34.4% ($K_c = 1$) and 63.3% ($K_c = F$) compared with the existing ILP.

D. Impact of Objective Function on Computation Complexity

Finally, we examine how the difference of the objective function affects the computation complexity. Figs. 6, 7, and 8 are the results in case of the minimization of node and link utilization, each of which corresponds to Figs. 3, 4, and 5, respectively. Note that the SPTP-based ILP can also support the minimization of node and link utilization by replacing $d_{i,j}$ as follows:

$$d_{i,j} = \begin{cases} \frac{b_c}{B_{i,j}}, & \text{if } (i,j) \in \mathcal{E}, \\ \frac{p^{\text{func}}_{c,f(i)}}{P_i}, & \text{if } (i,j) \in \widehat{\mathcal{E}}^{\text{out}}, \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

where $f(i)$ represents the function supported by the imaginary node i. We observe that the results in case of the minimization of node and link utilization are similar to those in case of the minimization of total delay.

VI. Conclusions

In this paper, we have developed a simple yet novel ILP formulation for service chaining in NFV networks, which is an extension of the ILP formulation for the SPTP with the help of the augmented network model. Through numerical experiments, we have showed that the SPTP-based ILP can support about two times larger scale systems than the existing ILP.

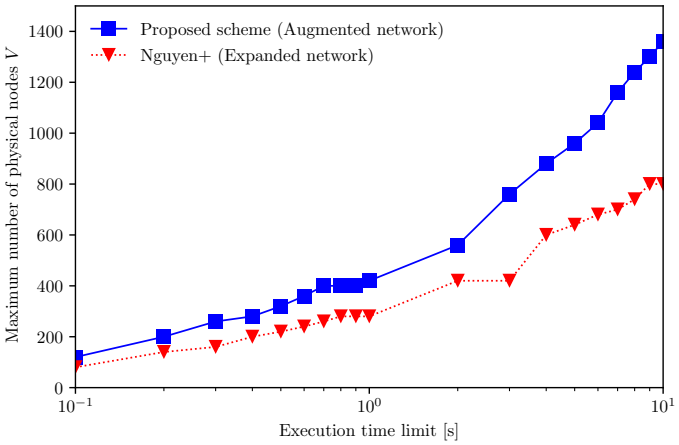

Fig. 6: The maximum number of physical nodes to be solved within the execution time limit (blocking probability minimization case).

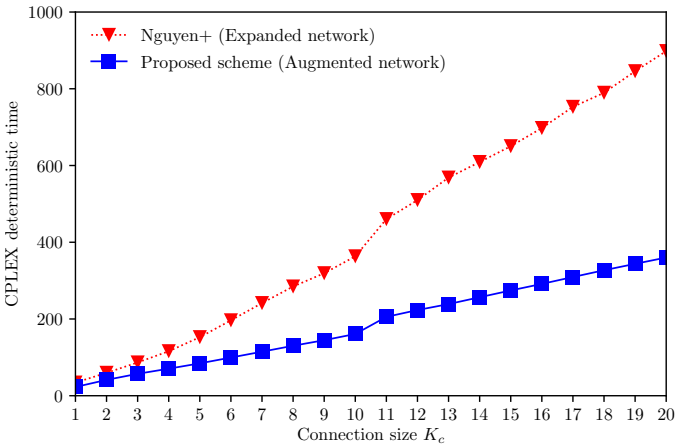

Fig. 8: Impact of connection size K_c on deterministic time (blocking probability minimization case).

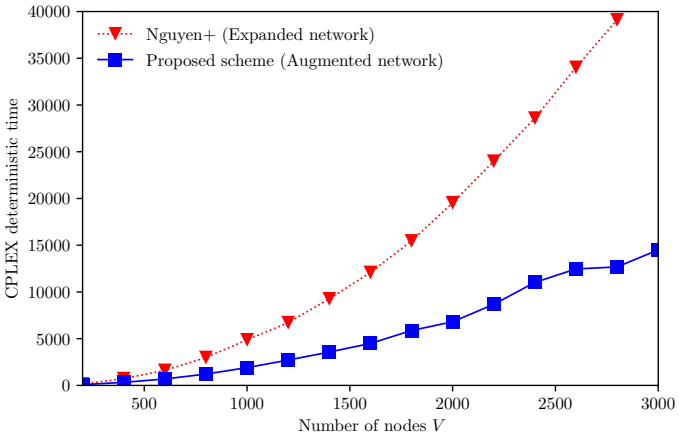

Fig. 7: Impact of the number of nodes on deterministic time (blocking probability minimization case).

As future work, we will extend the SPTP-based ILP formulation for both service chaining and function placement. In addition, we also plan to apply heuristic algorithms for the original SPTP to tackle the computation complexity problems.

REFERENCES

[1] J. G. Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[2] B. Yi, X. Wang, K. Li, S. k. Das, and M. Huang, "A Comprehensive Survey of Network Function Virtualization," *Computer Networks*, vol. 133, pp. 212–262, Mar. 2018.

[3] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed., 2000.

[4] S. Bhat and G. N. Rouskas, "Service-Concatenation Routing with Applications to Network Functions Virtualization," in *Proc. of 26th International Conference on Computer Communication and Networks (ICCCN)*, Jul. 2017, pp. 1–9.

[5] L. Gao and G. N. Rouskas, "On Congestion Minimization for Service Chain Routing Problems," in *Proc. of IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.

[6] R. C. de Andrade and R. D. Saraiva, "An Integer Linear Programming Model for the Constrained Shortest Path Tour Problem," *Electronic Notes in Discrete Mathematics*, vol. 69, pp. 141–148, Aug. 2018.

[7] T. Nguyen, A. Girard, C. Rosenberg, and S. Fdida, "Routing via Functions in Virtual Networks: The Curse of Choices," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1192–1205, Jun. 2019.

[8] N. Huin, B. Jaumard, and F. Giroire, "Optimal Network Service Chain Provisioning," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1320–1333, Jun. 2018.

[9] A. Gupta, B. Jaumard, M. Tornatore, and B. Mukherjee, "A Scalable Approach for Service Chain Mapping With Multiple SC Instances in a Wide-Area Network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 529–541, Mar. 2018.

[10] M. Savi, M. Tornatore, and G. Verticale, "Impact of Processing Costs on Service Chain Placement in Network Functions Virtualization," in *Proc. of IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, Nov. 2015, pp. 191–197.

[11] M. A. T. Nejad, S. Parsaeefard, M. A. Maddah-Ali, T. Mahmoodi, and B. H. Khalaj, "vSPACE: VNF Simultaneous Placement, Admission Control and Embedding," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 542–557, Mar. 2018.

[12] N. Hyodo, T. Sato, R. Shinkuma, and E. Oki, "Virtual Network Function Placement for Service Chaining by Relaxing Visit Order and Non-Loop Constraints," *IEEE Access*, pp. 1–12, Aug. 2019.

[13] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "Exploiting Congestion Games to Achieve Distributed Service Chaining in NFV Networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 407–420, Feb. 2017.

[14] ——, "A Game Theoretic Approach for Distributed Resource Allocation and Orchestration of Softwarized Networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 3, pp. 721–735, Mar. 2017.

[15] P. Festa, F. Guerriero, D. Laganà, and R. Musmanno, "Solving the Shortest Path Tour Problem," *European Journal of Operational Research*, vol. 230, no. 3, pp. 464–474, Nov. 2013.

[16] D. Ferone, P. Festa, F. Guerriero, and D. Laganà, "The Constrained Shortest Path Tour Problem," *Computers & Operations Research*, vol. 74, pp. 64–77, Oct. 2016.

[17] D. Ferone, P. Festa, and F. Guerriero, "An Efficient Exact Approach for the Constrained Shortest Path Tour Problem," *Optimization Methods and Software*, Jan. 2019.

[18] ILOG, "IBM ILOG CPLEX optimizer," https://www.ibm.com/products/ilog-cplex-optimization-studio, 2019, Accessed 15 Dec. 2019.

[19] V. Batagelj and U. Brandes, "Efficient Generation of Large Random Networks," *Physical Review E*, vol. 71, no. 3, p. 036113, Mar. 2005.

[20] IBM, "Deterministic time," https://www.ibm.com/support/knowledgecenter/en/SSSA5P_12.7.1/ilog.odms.studio.help/CPLEX/ReleaseNotes/topics/releasenotes125/newDetTime.html, 2019, Accessed 15 Dec. 2019.