Online Joint VNF Chain Composition and Embedding for 5G Networks

Junling Li*, Weisen Shi*, Qiang Ye*, Weihua Zhuang*, Xuemin (Sherman) Shen*, and Xu Li†
*Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada, N2L 3G1
†Huawei Technologies Canada Inc., Ottawa, ON, Canada, K2K 3J1
Email: {j742li, w46shi, q6ye, wzhuang, sshen}@uwaterloo.ca, Xu.LiCA@huawei.com

***Abstract*—Network function virtualization (NFV) is one of the enabling technologies for fifth generation (5G) networks. How to allocate physical resources to customized network services both fairly and efficiently remains a challenging research issue in NFV. This paper proposes a two-stage approach to jointly optimize the chaining and embedding of virtual network functions (VNFs), to obtain feasible composition and embedding results with low complexity, while the average embedding cost is minimized and the total revenue is increased. In the first stage, the VNF chaining order is optimized based on the location and functionality of substrate nodes, and the ratio of outgoing data rate over incoming data rate for each required VNF. In the second stage, we allocate the physical resources based on the preliminary VNF ordering under the resource capacity constraints. A node splitting mechanism is also employed to improve the resource allocation fairness and increase the service acceptance ratio for the substrate network. Simulation results are presented to validate the feasibility and effectiveness of the proposed approach.**

***Index Terms*—5G, network function virtualization, chaining of virtual network functions, resource allocation.**

I. Introduction

In 5G core networks, customized services will be provided by letting traffic go through different sequences of network functions (NFs). Various NFs, such as firewall, deep packet inspection, data monitoring, encryption and decryption, require specific hardware configurations, and are currently implemented on physical middleboxes or network servers. Due to the explosive development of network elements for Internet of Things (IoT), the number of services will be prominently diversified in future 5G networks [1]-[3]. However, high capital and operational expenditures are required for operators to deploy, maintain, and upgrade physical middleboxes.

Network function virtualization (NFV) is a promising technology to provide cost-effective service customization. In NFV, different NFs are decoupled from dedicated network servers, and virtualized as software instances running on high-capacity commodity servers placed in various network locations. In such a way, network services can be fulfilled by orchestrating different virtual network functions (VNFs) to form VNF forwarding graphs (VNF-FGs) or service function chains (SFCs), which are then embedded onto an NFV-enabled physical network infrastructure [4]. This process is referred to as *VNF chain composition and embedding*. Accordingly, physical resources in the network (computing resources on serves and transmission bandwidth resources on links) are virtualized to allow efficient resource sharing among different services. With NFV, the customized services can be accommodated in a cost-effective, agile, and flexible manner [5].

One of the challenging research issues in NFV is how to efficiently and fairly allocate the physical resources in the substrate network to accommodate more service requests, which is referred as the NFV resource allocation (NFV-RA) problem. The NFV-RA problem can be solved in three stages [6]: 1) VNF chain composition (VNF-CC), which deals with how to compose the order of VNFs; 2) VNF-FG embedding (VNF-FGE), which tackles how to embed the composed VNF-FGs onto the substrate network; 3) VNF scheduling (VNF-SCH), which determines the schedule for processing the VNFs on each substrate node. In this paper, we focus on the first two stages of NFV-RA and leave VNF-SCH for our future research.

Many existing research works study the NFV-RA problem, most of which focus on the second stage [6]-[7]. A few studies jointly address the first and second stages [8]-[10]. In [8], an uncoordinated approach is proposed to solve the first two stages of the NFV-RA problem. Specifically, a heuristic is employed for composing the VNF-FG, then the VNF-FG embedding problem is formulated as a constrained mixed integer problem. The proposed approach is suitable for small-scale networks due to the computational complexity. In addition, as the two stages are solved independently, the composition results are not optimized. A coordinated approach is proposed in [9], where VNF-FGs are composed and embedded simultaneously. The result from VNF-CC is fed into the VNF-FGE optimization problem, increasing the successful embedding probability.

In this paper, we propose a heuristic approach to solve a joint VNF-CC and VNF-FGE problem. The objective is to find a feasible VNF chain composition and embedding with low computational complexity, while minimizing the embedding cost and increasing the number of accommodated services simultaneously. The proposed heuristic consists of two stages: In the first stage, an order of VNFs to compose the VNF-FG is determined for minimizing the embedding cost. After this stage, the order of VNFs and the capacity constraints of the initial VNF-FG are determined. In the second stage, the heuristic algorithm tries to allocate resources to the virtual nodes and links in the VNF-FG. If certain nodes are overloaded, a node splitting mechanism is employed to modify the VNF-FG determined in the first stage. The proposed approach improves the resource utilization of the network, with an increased service acceptance ratio.

The remainder of this paper is organized as follows. The system model is presented in Section II. In Section III, we give the problem description. In Section IV, the heuristic approach is proposed to solve the joint VNF chain composition and embedding problem. Simulation results are presented in Section V to validate the feasibility and advantages of the proposed approach. Section VI concludes this work.

II. System Model

The system model includes a substrate network, VNF requests, and VNF-FGs.

Substrate network - The substrate network is represented as a directed graph $G = (N, L)$ with set N of physical nodes and set L of physical transmission links. All physical nodes have both routing ability and processing capacities for operating multiple VNFs of different types. Let $\mathcal{F}$ be the set of available VNFs in the substrate network. We define $c(n)$ as the available CPU processing capacity of substrate node n, $\forall n \in N$. Let $\mathcal{H}(n)$ be a set of VNFs of different types that substrate node n can host. All physical links are directed edges and associated with bandwidth resources. Denote $b(l)$ as the bandwidth capacity of link l, $\forall l \in L$. Let $\mathcal{P}$ represent the set of all the substrate paths (each path contains a set of links between two substrate nodes) in G. Let $b(P)$ denote the available bandwidth capacity of a substrate path $P \in \mathcal{P}$.

VNF request (VNFR) - Suppose that there are J service requests. For the j-th VNF request V^j, $j = 1, \ldots, J$, the following information is required [9]: 1) The set of VNFs with dependency $F_j = \{f_1^j, f_2^j, \ldots, f_{|F_j|}^j\}$, where $|F_j|$ is the number of VNFs in V^j; 2) Source and destination substrate nodes for V^j, represented by n_s and n_d (we focus on unicast communications), respectively; 3) Initial data rate d_r entering the chained functions; 4) The ratio of outgoing data rate over incoming data rate $r(f_i^j)$ of each VNF f_i^j, $i = 1, \ldots, |F_j|$; 5) The processing resource demand per bandwidth unit $p(f_i^j)$ of each VNF f_i^j.

VNF-FG - Given the information of a VNFR, a VNF-FG can be composed in different orders. We denote a VNF-FG by a directed graph $G' = (N', L')$. Denote the VNF to which the virtual node $n' \in N'$ corresponds by $f(n')$. Let $c(n')$ denote the CPU processing resource requirement on $n' \in N'$ and $b(l')$ represent the bandwidth requirement on $l' \in L'$.

Fig. 1(a) shows an example of a VNFR, where the number of VNFs is 5 and the initial date rate entering the chained functions is 1 Gbps. The dependency between consecutive VNFs is denoted by a dashed arrow. For example, in Fig. 1(a), the dashed arrow from f_4 to f_2 indicates that f_4 must be processed after f_2. Two possible VNF-FG compositions based on the VNFR information are shown in Fig. 1(b). The numbers on the links represent the bandwidth resource requirements (in Gbps) while the numbers in the rectangles indicate the CPU processing resource requirements.

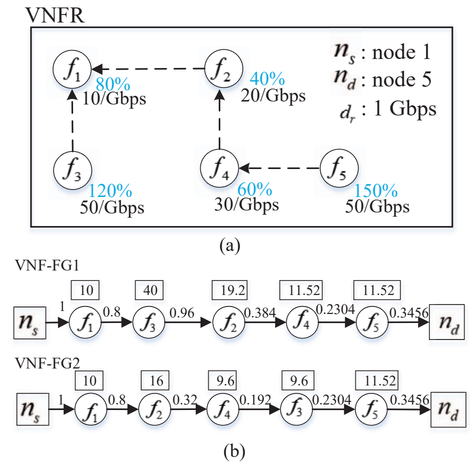


Fig. 1. (a) Example of a VNFR. (b) Two possible VNF-CC results for the VNFR in (a).

III. Problem Description

Allocating physical resources to a service request consists of composing a VNF-FG for a VNFR and embedding the resultant VNF-FG onto the substrate network. By jointly considering VNF-CC and VNF-FGE, the embedding performance can be maximized. The objective of the VNF chain composition and embedding problem is to minimize the average embedding cost or to maximize the total network revenue.

A. Objectives

Let $C(G')$ and $R(G')$ represent the cost and the revenue of embedding a VNF-FG, respectively. They are defined as follows [11]

$$C(G') = \alpha \sum_{l' \in L'} \sum_{l \in L} b_l(l') + \beta \sum_{n' \in N'} c(n') \tag{1}$$

$$R(G') = \gamma \sum_{l' \in L'} b(l') + \delta \sum_{n' \in N'} c(n') \tag{2}$$

where $b_l(l')$ represents the total amount of bandwidth resources allocated to virtual link l' on the substrate link l, α and β are weighting coefficients for $C(G')$, while γ and δ are weighting coefficients for $R(G')$ for bandwidth and computing resources respectively. The average embedding cost $\bar{C}(t)$ and total revenue $\mathcal{R}(t)$ at time t are given by

$$\bar{C}(t) = \frac{\sum_{G' \in \mathcal{G}'(t)} C(G')}{|\mathcal{G}'(t)|} \tag{3}$$

and

$$\mathcal{R}(t) = \sum_{G' \in \mathcal{G}'(t)} R(G') \tag{4}$$

where $\mathcal{G}'(t)$ is the set of VNF-FGs that have been embedded successfully before t. To allow efficient resource sharing among customized services, $\bar{C}(t)$ should be minimized, while the total revenue $\mathcal{R}(t)$ should be maximized.

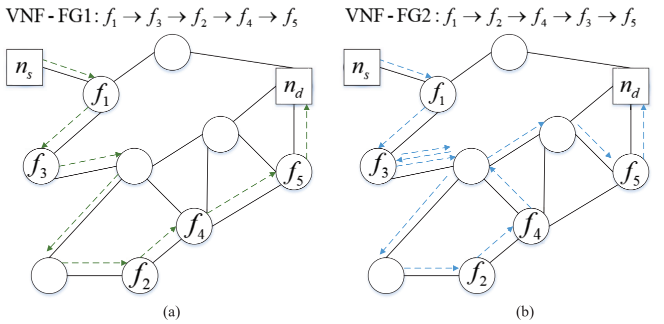


Fig. 2. Illustration of the impact of substrate node locations on the actual embedding cost.

B. VNF Chain Composition

The VNF chain composition problem is described as follows: Given the information of a VNFR, we compose the VNF-FG respecting the dependencies between VNFs, while ensuring that the actual cost $C(G')$ of the VNF-FG is highly likely to be the lowest in the embedding process. For simplicity consider VNF-FGs in a linear topology, i.e., two endpoints are connected via a set of ordered VNFs. In this case, the composition problem reduces to finding the order of VNFs in the VNFR to compose the VNF-FG. For j-th VNFR, we define a binary variable x_{ijp} to indicate the order of VNFs as

$$x_{ijp} = \begin{cases} 1, \text{if } f_i^j \text{ is the } p\text{-th VNF in the chain} \\ 0, \text{otherwise} \end{cases} \quad (5)$$

where $i = 1, \ldots, |F_j|$, and $p = 1, \ldots, |F_j|$. To achieve a feasible VNF ordering, the following two constraints should be satisfied

$$\sum_{i=1}^{|F_j|} x_{ijp} = 1, \forall p = 1, \ldots, |F_j| \quad (6)$$

$$\sum_{p=1}^{|F_j|} x_{ijp} = 1, \forall i = 1, \ldots, |F_j|. \quad (7)$$

To ensure the dependency between two VNFs, we have

$$p \mid_{x_{ujp}=1} > p \mid_{x_{vjp}=1} \quad (8)$$

for function f_u^j depending on f_v^j, where $u, v = 1, \ldots, |F_j|$.

C. VNF Forward Graph Embedding

The VNF-FGE problem is further divided into two subproblems, i.e., node embedding problem and link embedding problem. For the node embedding, one tries to find a mapping relation, $\mathcal{M}^N : N' \to N$, from a virtual node to a substrate node, such that for all $n' \in N'$, we have $\mathcal{M}^N(n') \in N$, subject to

$$f(n') \in \mathcal{H}(\mathcal{M}^N(n')) \quad (9)$$

and

$$c(\mathcal{M}^N(n')) \geq c(n') \quad (10)$$

where constraint (9) ensures that the VNF corresponding to n' can be hosted on the substrate node to which n' is embedded. Constraint (10) guarantees that the CPU processing resource requirement of n' can be satisfied. For the link embedding, one aims to find a mapping $\mathcal{M}^L$ from the virtual links to substrate paths, such that for all $l' \in L'$, we have $\mathcal{M}^L(l') \in \mathcal{P}$, subject to

$$b(\mathcal{M}^L(l')) \geq b(l') \quad (11)$$

where constraint (11) entails that the bandwidth resource requirement of l' can be satisfied. In order to maximize the resource utilization and achieve the largest total revenue, the VNF-FG should be embedded in a way that the actual cost $C(G')$ is minimized.

IV. A Heuristic Approach for Joint VNF-CC and VNF-FGE

A. Problem Analysis

The joint VNF-CC and VNF-FGE problem is in fact a combinatorial optimization problem, which is known to be NP-hard. For a large-scale network, its computational complexity is extremely high. To this end, we propose a heuristic approach to obtain feasible solutions efficiently. The heuristic algorithm is designed based on the following two intuitions: 1) The chosen VNF-CC with the substrate network condition (e.g., node locations and VNF hosting abilities) should minimize the average embedding cost $\bar{C}(t)$. Specifically, suppose that VNF-FG1 and VNF-FG2 are embedded onto the same substrate network, as shown in Fig. 2. The link embedding cost of VNF-FG1 is lower than that of VNF-FG2, since the embedded VNF-FG2 consumes more bandwidth resources on substrate links. Therefore, the substrate network condition should be taken into consideration when determining the order of VNFs in the composition stage; 2) The average embedding cost $\bar{C}(t)$ can be minimized by chaining the VNFs in a certain way, because different VNF orderings result in different link and node resource requirements for the VNF-FG. As illustrated in Fig. 1, the amount of physical resources required by VNF-FG1 is larger than that required by VNF-FG2. Therefore, VNF-FG2 is more likely to achieve a lower embedding cost in the embedding stage. This implies that the ratio of outgoing data rate over incoming data rate of each VNF should also be considered in the composition stage. Noticed that, by minimizing $\bar{C}(t)$, the number of VNFRs accommodated by the substrate network is also expected to increase, leading to a larger total revenue $\mathcal{R}(t)$ of the substrate network.

Based on the processing analysis, we present a two-stage heuristic algorithm to jointly solve the VNF-CC and VNF-FGE problems. In the first stage, we compose the VNF-FG for minimizing the embedding cost; In the second stage, we embed the composed VNF-FG onto the substrate network while satisfying the physical resource requirements.

B. VNF Chaining

Upon a VNFR arrival, the chaining order of VNFs for the VNF-FG is determined to minimize the embedding cost. To this end, the heuristic algorithm is designed with two considerations: 1) the locations of substrate nodes that have

Algorithm Part 1: VNF chaining for V^j

1 $\mathcal{O} \leftarrow \phi$ $n_c \leftarrow n_s$;
2 $F_r \leftarrow F_j$;
3 $success \leftarrow true$;
4 **for** $k = 0, \dots, |F_j|$ **do**
5 $\quad \mathcal{L} \leftarrow 1$;
6 $\quad$ **while** $\mathcal{L} \leq \mathcal{L}_{max}$ **do**
7 $\quad\quad \mathcal{V} \leftarrow findFeasibleNodes(n_c, F_r, \mathcal{L})$;
8 $\quad\quad$ **if** *$\mathcal{V}[i]$ is empty, $\forall i = 1, \dots, |F_j|$* **then**
9 $\quad\quad\quad \mathcal{L} \leftarrow \mathcal{L} + 1$;
10 $\quad\quad$ **end**
11 $\quad\quad$ **else**
12 $\quad\quad\quad$ sort f_i^j in ascending order according to $r(f_i^j)$, $\forall i \in \{i|\mathcal{V}[i] \neq \phi\}$;
13 $\quad\quad\quad$ select f_i^j with the smallest $r(f_i^j)$, denote it as $f_{i^*}^j$;
14 $\quad\quad\quad$ $\mathcal{O}$.push_back($f_{i^*}^j$);
15 $\quad\quad\quad$ remove $f_{i^*}^j$ from F_r;
16 $\quad\quad\quad$ **if** *the size of $\mathcal{V}[i^*]$ is larger than 1* **then**
17 $\quad\quad\quad\quad$ sort the substrate nodes in $\mathcal{V}[i^*]$ in a descending order according to their available CPU processing resources;
18 $\quad\quad\quad$ **end**
19 $\quad\quad\quad n_c \leftarrow \mathcal{V}[i^*][0]$;
20 $\quad\quad$ **end**
21 $\quad$ **end**
22 $\quad$ **if** *the size of $\mathcal{O}$ is not $k+1$* **then**
23 $\quad\quad success \leftarrow false$;
24 $\quad\quad$ break;
25 $\quad$ **end**
26 **end**

Algorithm Part 2: VNF-FG embedding with node splitting

1 $nodesuccess \leftarrow true$;
2 $linksuccess \leftarrow true$;
3 **for** $n' \in N'$ **do**
4 $\quad$ calculate $c(n')$;
5 $\quad$ **if** *$c(n')$ cannot be satisfied* **then**
6 $\quad\quad$ try to do node splitting for n';
7 $\quad\quad$ **if** *$c(n')$ is still not satisfied* **then**
8 $\quad\quad\quad nodesuccess \leftarrow false$;
9 $\quad\quad\quad$ break;
10 $\quad\quad$ **end**
11 $\quad$ **end**
12 $\quad$ allocate CPU resource to n';
13 **end**
14 **if** $nodesuccess = true$ **then**
15 $\quad$ **for** $l' \in L'$ **do**
16 $\quad\quad$ calculate $b(l')$;
17 $\quad\quad$ use k-shortest path algorithm embed the link;
18 $\quad\quad$ **if** *$b(l')$ can be satisfied on all the sub paths* **then**
19 $\quad\quad\quad$ allocate bandwidth resource to l';
20 $\quad\quad$ **end**
21 $\quad\quad$ **else**
22 $\quad\quad\quad linksuccess \leftarrow false$;
23 $\quad\quad$ **end**
24 $\quad$ **end**
25 **end**

the ability to host each VNF; and 2) the ratio of outgoing data rate over incoming data rate for each VNF $r(f_i^j)$.

The pseudo-code of determining the VNF ordering is presented in Algorithm Part 1. Let $\mathcal{O}$ be a list that contains the ordered VNFs in the chain. Starting from the source node of the VNFR, n_s, the algorithm determines the VNF ordering by pushing all the VNFs to $\mathcal{O}$ with $|F_j|$ trials. In each trial, a searching procedure is performed starting from the current substrate node n_c hosting the preceding VNF (line 8). Our goal is to find a set of substrate nodes that have the ability to host the remaining VNFs in the VNFR, F_r, within a specified range $\mathcal{L}$. If no feasible substrate nodes are found for all the remaining VNFs, increase $\mathcal{L}$ by 1 (lines 9-10). To limit the substrate link resource usage, the maximum path length between two consecutive VNFs is set as $\mathcal{L}_{max}$. Otherwise, we sort all the VNFs having feasible substrate nodes to be embedded in an ascending order according to their ratios of outgoing data rate over incoming data rate (line 12). The VNF with the smallest ratio is chosen and pushed into $\mathcal{O}$ (lines 13-14). Then, we remove this VNF from the set of remaining VNFs for the VNFR (line 15). If there are more than one feasible nodes for the selected VNF, we sort all the feasible nodes in a descending order according to their available processing resources (lines 16-18). Then, the substrate node with the largest amount of available processing resources is taken as the next substrate node (line 19). If no VNF is pushed into $\mathcal{O}$ in any trial, the VNF ordering stage is a failure and the VNFR is rejected.

C. VNF-FG Embedding with Node Splitting

After the VNF ordering is determined, the VNF-FG is composed. In this stage, the VNF-FG is embedded onto the substrate network. Algorithm Part 2 presents the details of how to embed the VNF-FG with node splitting mechanism. For each virtual node, its processing resource requirement is calculated (line 4). Then, we check if the requirement can be satisfied on the embedded substrate node. If the processing resource on the substrate node is not sufficient, the virtual node n' is split into two virtual nodes (lines 5-6). If the processing resource requirement is still not satisfied, the node embedding fails and the VNF-FG request (along with the corresponding VNFR) is rejected (lines 7-10). Otherwise, the processing resource is allocated (line 12). The link embedding starts once a successful node embedding is achieved. The k-shortest path algorithm is employed for link embedding (line 17) [12]. If a substrate path with enough bandwidth resource exists, the virtual link is embedded to that path and the corresponding

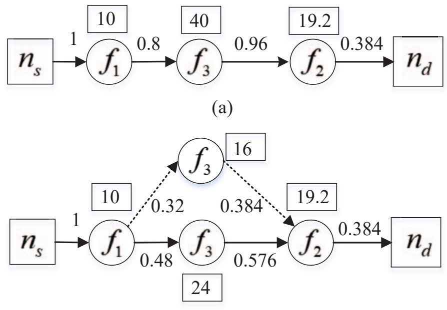


Fig. 3. Illustration of the node splitting mechanism. (a) VNF-FG before node splitting and (b) VNF-FG after node splitting.

bandwidth resource is allocated (lines 18-19). Otherwise, the link embedding fails and the VNF-FG is rejected (line 21).

The node splitting mechanism is explained as follows. Fig. 3(a) shows a VNF-FG consisting of three VNFs. Suppose that the processing resource requirement for f_3 exceeds the resource capacity on the substrate node to which f_3 is embedded. Starting from the substrate node to which the preceding VNF (which is f_1 in this case) is embedded, the node splitting mechanism finds another substrate node having the ability to host f_3 within the specified maximum range $\mathcal{L}_{max}$. If a feasible substrate node is found, the processing resource requirement, as well as the incoming and outgoing link bandwidth resource requirements are split. The splitting ratio is determined based on the available processing resource capacities of the two substrate nodes (the original one to which f_3 is embedded and the new one just found). By using the node splitting mechanism, the node successful embedding probability can be increased.

V. Performance Evaluation

In this section, we evaluate the effectiveness of the proposed heuristic algorithm. To obtain the best performance, we demonstrate an appropriate value of $\mathcal{L}_{max}$. Then, we compare the heuristic algorithm with other benchmarks algorithms. Simulations are carried out to determine the $\mathcal{L}_{max}$ value for the heuristic algorithm. The substrate network is randomly generated with 20 nodes using the GT-ITM tool [13] in a (25 × 25) grid. Each pair of substrate nodes is randomly connected with probability 0.5. We assume that the VNFRs arrive in a Poisson process with arrival rate of 4 VNFRs per 100 time units. Each VNFR has an exponentially distributed lifetime with an average of 1000 time units. The total simulation time is 50000 time units. The dependencies between the VNFs in each VNFR are randomly generated. Other setting for the parameters is given by Table I.

TABLE I

Parameter Setting in Simulation

Parameter	Value	Parameter	Value
$\lvert N \rvert$	25	$\lvert F_j \rvert, \forall j=1,...,\lvert R \rvert$	5
$\lvert L \rvert$	24	d_r	[25, 75]
$c(n), \forall n \in N$	[50, 100]	$r(f_i^j), \forall f_i^j \in F_j$	[0.5, 1.5]
$b(l), \forall l \in L$	[200, 400]	$p(f_i^j), \forall f_i^j \in F_j$	0.5
$\lvert \mathcal{F} \rvert$	10	$\lvert \mathcal{H}(n) \rvert, \forall n \in N$	5

$\lvert \bullet \rvert$ denotes the number of elements of a finite set

A. Value of Parameter $\mathcal{L}_{max}$

As described in Section IV-B, parameter $\mathcal{L}_{max}$ is used to limit the distance (and the link bandwidth resource usage) between any two substrate nodes chosen to host two consecutive VNFs in the chain. A smaller $\mathcal{L}_{max}$ value leads to a higher failure probability of VNF chain composition, but increases the efficiency of resource utilization. We compare the acceptance ratio for different values of $\mathcal{L}_{max}$, as shown in Fig. 4. The acceptance ratios of $\mathcal{L}_{max} = 3$ and $\mathcal{L}_{max} = 4$ are close, and are approximately 20% and 60% higher than that of $\mathcal{L}_{max} = 2$ and $\mathcal{L}_{max} = 1$, respectively. This is because a smaller $\mathcal{L}_{max}$ means a more strict distance limitation between the two substrate nodes to host two consecutive VNFs. When $\mathcal{L}_{max}$ is chosen to be small, the probability of finding feasible substrate nodes for the VNF candidates decreases. It can be observed that the acceptance ratio converges at $\mathcal{L}_{max} = 3$. Therefore, we set $\mathcal{L}_{max} = 3$ for the proposed heuristic in the following simulation scenarios.

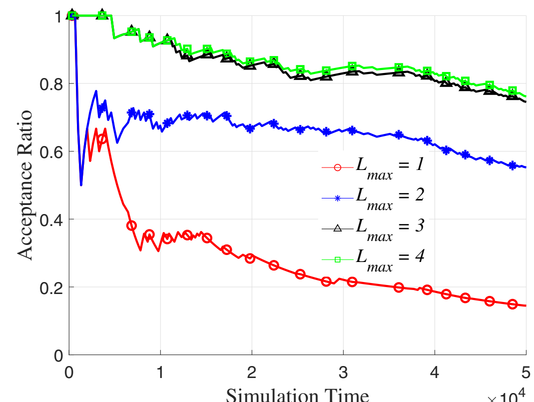
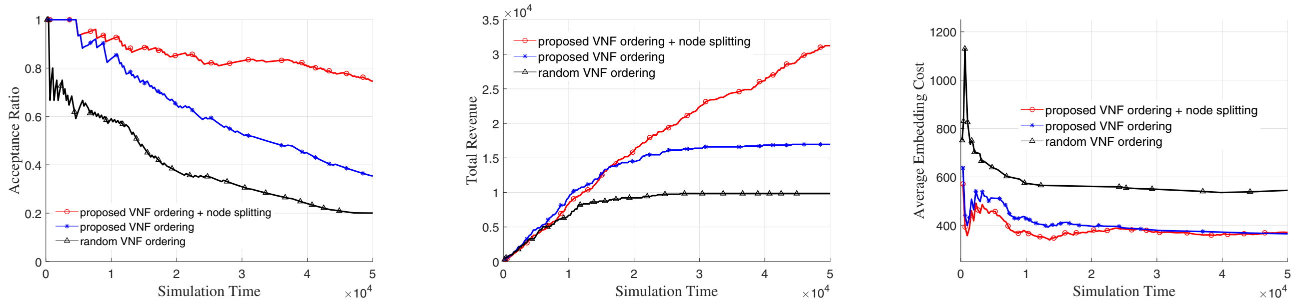


Fig. 4. The acceptance ratio along with time for different $\mathcal{L}_{max}$.

B. Performance Comparisons

We compare our proposed algorithm with two basic benchmark algorithms: 1) For VNF-CC, VNFs are chained in the same manner as the proposed heuristic algorithm and, for VNF-FGE, the node splitting mechanism is not employed; 2) For VNF-CC, VNFs are chained in a random order and, for VNF-FGE, the VNFs are embedded randomly on the substrate nodes under the condition that the node should have the ability to host the VNF embedded on it. We compare the three algorithms in terms of acceptance ratio, total revenue, and average embedding cost.

1) Acceptance ratio: Fig. 5(a) shows a comparison of the acceptance ratio for the three algorithms. It is observed that the proposed algorithm achieves the highest acceptance ratio. Chaining and embedding the VNFs randomly performs the



(a) (b) (c)

Fig. 5. Comparison of three algorithms in terms of different metrics (a) acceptance ratio, (b) total revenue, (c) average embedding cost.

worst, because the physical resource requirements for nodes and links are not optimized. The physical resources of the substrate network are depleted much faster than the other two algorithms. Also, the node splitting mechanism is useful to increase the number of accepted VNFRs in the substrate network, due to more balanced node resource utilization.

2) Total revenue: The comparison of the total revenue for the three algorithms is shown in Fig. 5(b) (with $\gamma = \delta = 1$). It can be seen that the total revenues remain close at the beginning. The difference becomes more and more significant after 15000 time units. The total revenue of the proposed algorithm is the highest among all the three algorithms. This is because in the beginning phase, the substrate network has sufficient physical resources to accommodate all the arriving VNFRs. Even though the VNFs are not chained in a proper order or the node splitting mechanism is not employed, the number of accepted VNFRs remain close for all the three algorithms. As the resources become depleted, with the proper VNF ordering and the node splitting mechanism, the proposed algorithm is able to minimize the embedding cost and achieve a balanced resource usage among all the substrate nodes, which leads to a higher VNFR acceptance ratio and a larger total revenue.

3) Average embedding cost: Fig. 5(c) shows the average embedding cost for the three algorithms (with $\alpha = \beta = 1$). It can be seen that, chaining and embedding the VNFs in a random way leads to a much larger average embedding cost than chaining VNFs in a proper order and then embedding them accordingly. Because the physical resource requirements are not optimized. The resource requirements for both nodes and links are higher than that of the proposed VNF ordering. Note that, the node splitting mechanism does not lead to a lower embedding cost. The reason is that, when we split a virtual node, the summation of CPU processing resource requirements of the split virtual nodes is the same as that of the original virtual node, which is also the case for the incoming and outgoing links of the split virtual nodes. Therefore, the embedding cost remains the same.

VI. Conclusion and Future Work

In this paper, we propose a two-stage heuristic approach that jointly optimizes the chaining and resource allocation of VNFs. By jointly considering the substrate node locations and the ratios of outgoing data rate over incoming data rate for VNFs, the physical resource requirements for VNF-FGs are satisfied efficiently. A node splitting mechanism is adopted to increase the VNFR acceptance ratio of the substrate network. Simulation results demonstrate that the proposed approach achieves a promising acceptance ratio for the dedicated scenario. For future works, we will develop algorithms to find the optimal solution based on rigorous mathematical formulation. Moreover, dynamic chaining and embedding will also be considered to allow reconfiguration of the already embedded VNF-FGs to accommodate more arriving VNFRs. We will also compare the performance of the proposed method with that of conventional offline methods.

References

[1] J. Li, N. Zhang, Q. Ye, W. Shi, W. Zhuang, and X. Shen, “Joint resource allocation and online virtual network embedding for 5G networks,” in *Proc. IEEE GLOBECOM’ 17*, pp. 1-6, Dec. 2017.

[2] Q. Ye, J. Li, K. Qu, W. Zhuang, X. Shen, and X. Li, “End-to-end quality of service in 5G networks - Examining the effectiveness of a network slicing framework,” *IEEE Veh. Technol. Mag.*, to appear, doi: 10.1109/MVT.2018.2809473.

[3] H. Zhang, W. Quan, H.-C. Chao, et al., “Smart Identifier Network: A Collaborative Architecture for the Future Internet,” *IEEE Netw.*, vol. 30, no. 3, pp. 46-51, May 2016.

[4] P. Quinn and T. Nadeau, “Service Function Chaining Problem Statement,” *draft-ietf-sfc-problem-statement-07 (work in progress)* (2014).

[5] N. Zhang, P. Yang, S. Zhang, D. Chen, W. Zhuang, B. Liang, and X. Shen, “Software defined networking enabled wireless network virtualization: Challenges and solutions,” *IEEE Netw.*, vol. 31, no. 5, pp. 42-49, 2017.

[6] J. G. Herrera, and J. F. Botero. “Resource allocation in NFV: A comprehensive survey,” *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 518-532, 2016.

[7] Y. Xie, Z. Liu, S. Wang, and Y. Wang. “Service function chaining resource allocation: A survey,” *arXiv preprint arXiv:1608.00095*, Jul. 2016.

[8] S. Mehraghdam, M. Keller, and H. Karl. “Specifying and placing chains of virtual network functions.” in *Proc. IEEE Cloud Networking (CloudNet)*, pp. 7-13, 2014.

[9] M. T. Beck, and J. F. Botero. “Coordinated allocation of service function chains.” in *Proc. IEEE GLOBECOM’15*, pp. 1-6, 2015.

[10] L. Wang, *et al.*, “Joint optimization of service function chaining and resource allocation in network function virtualization, *IEEE Access*, vol. 4, pp. 8084-8094, 2016.

[11] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, “Virtual network embedding with coordinated node and link mapping,” in *Proc. IEEE INFOCOM’09*, pp. 783-791, 2009.

[12] D. Eppstein, “Finding the *k* shortest paths,” *SIAM Journal on Computing*, vol. 28, pp. 652-673, Feb. 1999.

[13] E. Zegura, K. Calvert, and S. Bhattacharjee, “How to model an internetwork,” in *Proc. IEEE INFOCOM’96*, pp. 594-602, 1996.