IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 13, 2019, accepted October 7, 2019, date of publication October 11, 2019, date of current version October 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2946648

An Approach for Service Function Chain Reconfiguration in Network Function Virtualization Architectures

YICEN LIU[1], HAO LU[2], XI LI[1], AND DONGHAO ZHAO[1]
[1]Shijiazhuang Campus, Army Engineering University, Shijiazhuang 050003, China
[2]Beijing Aerospace Control Center, Beijing 100094, China

Corresponding author: Hao Lu (luhaoak47@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51377170 and Grant 61271152, and in part by the National Youth Science Fund of China under Grant 61602505.

ABSTRACT Network function virtualization (NFV) technology continues gaining attention as a paradigm shift, and telecommunication services can be flexibly deployed and managed, resulting in lower CAPEX and OPEX for the Internet service providers (ISPs). Any telecommunication service is represented by a so-called service function chain (SFC), which can be achieved by a set of virtual network functions (VNFs) to be executed on virtual machines (VMs) in a strictly order according to SLAs. The NFV-enabled SFCs applied in the future 5G networks gradually emerges a challenging problem, particularly more and more users are trying to access their telecommunication services whenever and wherever, SFCs are needed to be dynamically and adaptively reconfigured, thus ensuring lower resource consumption and higher revenues for ISPs. This problem is called service function chain dynamic reconfiguration problem (SFC-DRP), and an integer linear programming (ILP) formulation with implementation in CPLEX is provided to exactly solve this problem. To reduce the algorithm time complexity, a novel heuristic solution is provided that can balance the tradeoff between service provider's revenue and reconfiguration cost. The obtained simulation results show that our proposal can approximate the performance of the ILP within a polynomial time, and outperform the existing benchmark in terms of reconfiguration overhead and the revenue from service provisioning respectively.

INDEX TERMS Network function virtualization (NFV), service function chain (SFC), reconfiguration overhead, revenue, heuristic solution, 5G networks.

I. INTRODUCTION

Driven by the diversity of heterogeneous network services, 5G network framework design has been transiting from the monolithic pattern to the softwaried paradigm, which is supported by the Network Function Virtualization (NFV) and Software-defined Network (SDN) technologies [1]–[3]. These technologies present themselves as revolution pivotal network architectural design concepts that leverage virtualization and cloud infrastructure elasticity for the purpose of supporting this quantum leap of the existing packet core which, in turn, leads to the remarkable improvement of provisioned service [4]. Especially, the telecommunication service is represented by service function chain (SFC), which is a set of VNFs that requires the traffic can be steered to traverse through predefined ordered network functions [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Matti Hämäläinen.

In this paper, we consider that any virtual network function (VNF) can be run on a VNF Instances (VNFIs) in virtualized resources (i.e., Virtual Machine). Without unnecessary function, NFV-enabled SFC can be flexibly deployed in the substrate network and the corresponding virtualized resources (e.g., CPU, storage) can be dynamically requested, so as to improve the substrate network utilization. In addition, SFC can be agilely and automatically placed in the same substrate network whenever there is an incoming request, which can significantly reduce the network operation overhead. Due to these advantages mentioned above, SFC technology has been deemed as a vital technology that can be used in the future 5G networks, where have the largest demands for telecommunication services [2], [3].

NFV-related SFC technology splits the role of network operator into multiple separated users. In order to support this technology, the standardization of SFC architecture have been proposed by IETF and ETSI [6], [7], which can efficiently allocate virtualized resources for VNFs so as to achieve VNFI placement and service function path (SFP) routing. The SFC architecture can be classified into three layers including substrate layer, control layer and upper layer. At the substrate layer, Infrastructure Providers (InPs) focus on optimizing the placement of their own network services. At the control layer, Internet Service Providers (ISPs) manage virtualized resources and orchestrate VNFs in a strictly order as SFCs according to incoming service requirements (e.g., user ratio, latency). At the upper layer, ISPs specify their service requirements and lease customized SFCs. Besides, they can also reconfigure their SFCs according to traffic variations in timely manner. Therefore, SFC greatly simplifies the telecommunication services deployment, operation, and maintenance, while simultaneously improving flexibility and efficiency, and reducing operational overhead for provisioning services [8].

To fully explore the advantages of SFC, more challenges are needed to work on [9], [10]. Especially, the grand objective of SFC technology is to support a typical generic service with vastly heterogenous requirement in 5G massive machine-type communication (mMTC) scenario. For example, with the rapidly growth of mobile devices in the future 5G mMTC scenario, more and more users are trying to access their telecommunication services whenever and wherever. SFCs are needed to be dynamically and adaptively reconfigured when reaching the reconfiguration condition, thus ensuring lower resource consumption and higher revenues for ISPs. For SFCs, the optimal resource allocation of VNFIs will change when dynamic service request arrives. However, static SFC resource schemes can easily make the paths among the users and their VNFIs sub-optimal, which would not only cause unnecessary resource consumption and operational cost. Therefore, determining the migration of VNFIs and re-routing SFP that optimize the resource consumption and reconfiguration overhead is a challenging problem, this problem is called *Service Function Chain Dynamic Reconfiguration Problem* (SFC-DRP).

This paper studies how to optimize the SFC reconfiguration in respond to the varying traffic demand and resource availability, with balancing the tradeoff optimization objective that is to maximize the service provider's revenue, which is total revenue from the served requests minus the total reconfiguration cost (i.e., VNFIs migration cost and bandwidth cost). Here, we assume that the lower/upper thresholds were fixed based on service level agreements (SLAs). However, frequent reconfigurations incur certain cost and might cause service interruption. The reconfiguration condition triggers if and only if the real-time utilization ratio reaches the lower/upper threshold and lasts for a duration. To solve SFC-DRP with these considerations, we first formulate an integer linear programming (ILP) to balance the tradeoff between service provider's revenue and operational overhead under the constraints (the specific constraints are detailed in the Section IV). Then, the ILP model has limited applicability due to computational complexity, we design a novel heuristic algorithm for the optimization model to approximate the performance of the ILP. In summary, our main contributions are as follows.

- We study the SFC-DRP in a practical network scenario, and aim to balance the service provider's revenue and operational overhead. We also formulate an ILP model to exactly solve the optimization.
- We propose a novel heuristic-based approach to efficiently reconfigure SFC in a dynamic network, so as to accelerate the problem-solving. This proposed approach is a local search algorithm based on tabu search (TS), which starts by creating an optimal initial solution via Fuzzy C-Means (FCM), and is iteratively improved by searching for better solutions in its neighborhood.
- Furthermore, our proposed solution has been evaluated in terms of reconfiguration cost, service provider's revenue, network resource utilization and reconfiguration time. The significance of this new approach is that it can find a near-optimal solution compared with currently mainstreaming algorithms.

To the best of our knowledge, this is first attempt to systematically formulate the SFC-DRP, and to propose and evaluate algorithms for the same purpose. Given the key role that NFV can play in the future 5G vertical industries, we hope our proposals will be important as bench marks for other researchers in this regard.

The remainder of the paper is organized as follows: Section II introduces the related works. Section III presents network model and formulates SFC-DRP. Next, we propose a novel heuristic solution to address the SFC-DRP within a polynomial time. Finally, we give the simulation results in Section VI and conclude the paper in VII.

II. OVERVIEW OF RELATED WORKS

Since the emergence of SFC oriented to future 5G networks, it has aroused great interest from industry and academia. There are numbers of comprehensive survey regrading SFC system and architecture aspects [11], [12]. Meanwhile, the standardization activities about SFC technology are making progress recently [13], and drafts on the use cases of SFC can be found in [6] and [7]. The studies in [11] and [14] have investigated NFV-enabled SFC from the perspective of system implementation. In the field of SFC system implementation, Martins *et al.* [11] pioneered the accomplishment of SFC system implementation using ClickOS. In the same filed, Mamatas *et al.* [14] proposed a so-called information exchange management as a service facility as an extension to ETSI's NFV management and orchestration framework, namely, the virtual infrastructure information service (VIIS), which exhibits the dynamic characteristics of such paradigms and can support information flow establishment, operation and optimization.

One of the main challenges of SFC is the resource allocation [9] (i.e., VNFs-Chains composition, VNF-Forward Graph Embedding, and VNFs Scheduling) of demanded network services in NFV-enabled network infrastructures.

In order to efficiently concatenate the different VNFs, Mehraghdam *et al.* [15] formulated a context-free language for formalizing the service requests. Besides, the authors also proposed a greedy heuristic algorithm that aims to minimize the total network data rate by sorting the VNFs in ascending order based on their ratio of outgoing to incoming date rate. Beck and Botero [16] proposed a scalable recursive heuristic algorithm that composes VNFs in the service chain. To achieve an efficient service chain composition with regard to network operator's multiple objectives, we have proposed a breadth first search algorithm [40], which achieves the efficient composition of service chain to deploy customized and dynamic NFV-enabled security services. Various studies focused on the perspective of VNF-FGE, Bari *et al.* [17] proposed a dynamic programming based heuristic called Viterbi to solve the SFC deployment problem. Ghaznavi *et al.* [18] demonstrated a mixed integer programming problem trying to minimizing the host and bandwidth deployment cost, and proposed a scalable heuristic to solve large instances of the problem in the data center networks (DCNs). In order to joint optimize resource consumption in terms of flow routing and VNF placement for SFC, we have proposed a greedy heuristic solution, and solved the VNF-FGE problem within a polynomial time [19]. In the field of VNFs Scheduling, Riera *et al.* [20] analyzed a model for the resource allocation with the aim to optimize the execution time of the deployed network services. Yuan *et al.* [21] improved upon the flexible job-shop model by introducing the process if bandwidth allocation, which minimizes the execution time of all VNFs in SFCs. Nevertheless, none of the aforementioned studies considered to support dynamic user demands in SFC resource allocation. In other words, the resource allocation in these studies are indeed a response to maintain isolation, performed at large time scales without consideration of reconfiguration overhead.

There are some pioneering studies with regard to network resource configuration [22], [23]. However, their studies mostly focus on the virtual/overlay networks that differ from SFC in service logic, end-to-end latency, and so on. In the field of SFC reconfiguration, Wen *et al.* [24] pioneered the Virtual Network Function Consolidation (VNFC) problem, which is followed by an integer linear programming formulation. They also design a greedy based heuristic, GNFC, to maximize the decrement of VNF number by reconfiguring VNFIs in SFCs. However, they did not consider reconfiguration costs. Ghaznavi *et al.* [25] proposed a VNF consolidation and migration algorithm, SLFL, to minimize operational costs in providing VNF as a service. SLFL selects the migrated VNFI candidate based on the current instants. The most related to our works are [26]–[29], all of which focus on the SFC readjustment/migration problem and formulating this problem as a mixed integer programming. Eramo *et al.* [26] proposed Viterbi based migration policies that establish when and where cold migrations of VNFIs should be accomplished, thus minimizing the total energy, which is characterized by the sum of the consolidation and migration energies occurring when the VNFIs are moved from the initial locations. Eramo *et al.* [27] demonstrated a heuristic migration algorithm that is based on MASRN, RLACM and VMMPC, this method used back-to-back results in VNFI placement, service chain routing, and VNFI migration in response to changing loads. Nevertheless, this work did not consider the bandwidth consumption of SFCs. Liu *et al.* [28] studied how to optimize SFC deployment and readjustment in dynamic situation, and try to joint optimize the deployment of new users' SFCs and readjustment of in-service users' SFCs by using a so-called column generation (CG) algorithm. However, this method was hard to applicable to the future 5G networks due to time complexity. Wang *et al.* [29] focused on the network slice reconfiguration problem, and proposed an HSR framework that can avoid mismatch of slice's resources and demands.

In summary, existing works on SFC technology are mainly focused on the resource allocation (i.e., VNFs-Chains composition, VNF-Forward Graph Embedding, and VNFs Scheduling), with the aim of maximizing resource utilization or minimizing operation overhead, which varies from different demands of scenarios. Besides, these works regarding resource allocation belong to the static scheme, which can not perform at large time scales and fail to efficiently utilize the substrate resource in a dynamic network environment. Therefore, our proposal enhances the state-of-the-art in that it complements SFC reconfiguration that has not been fully explored yet, with a way to accomplish optimal SFC reconfiguration oriented to future 5G networks. For the most related work to ours [26]–[29], few literatures have well balanced the tradeoff between service provider's revenue and reconfiguration cost (i.e., VNFI migration cost and bandwidth cost) when designing their algorithms, which is the focus in this paper. Here, we denote the optimal SFC reconfiguration problem, focusing on determining the reconfiguration of VNFIs at the optimal locations and how to find optimal re-routing path, with balancing the tradeoff optimization objective that is to maximize the service provider's revenue, which is total revenue from the served requests minus the total reconfiguration cost satisfying the constraints.

III. NETWORK MODEL AND PROBLEM DESCRIPTION

In this section we introduce the network model for our system (Section II-A, Section II-B) and formally define the optimal service function chain dynamic reconfiguration problem (Section II-C).

A. SUBSTRATE NETWORK

The physical network can be defined as an undirected graph $G^s = (N^s, E^s)$, where N^s represents a set of physical nodes where the VNFIs can be migrated on, E^s represents a set of physical links connecting $n^s \in N^s$. Each physical node $n^s \in N^s$ is associated with a set of x types of resources $S_n^x = \{s_n^i | i \in [1, x]\}$, where s_n^i represents the capacity of resources of type i, i.e., CPU, memory, or bandwidth. $\chi_{n^s} \in (0, 1)$ represents the resource utilization of server node $n^s \in N^s$. Given the set of resources available at a physical node $n^s \in N^s$, the substrate node set N^s can be classified into forwarding node set N_f^s with packets forwarding capacity and VNF instance N_f^v with processing capacity. Each VNF requires a set of resources to perform one single network function denoted $f_n \in F$, where F represents the set of all VNFs.

B. SERVICE FUNCTION CHAIN REQUEST

A software-driven 5G slice is used to serve a collection of data flows, which are based on a subset of substrate network nodes and links. The data flows in network slices can be processed by the service chains of network slices. It is assumed that there is a set of m SFC request represented by $\Gamma = (t_1, t_2, \ldots, t_m)$ and the SFC can be mathematically described as $t_i = (N^s, L^s, w^s, c^s, v^s)$, where N^s and L^s represent the subset of nodes and links built on the network slice s, w^s, c^s, and v^s represents the capacity of forwarding nodes, links, and VNFIs. Assume that there exist $|F^s|$ data flows in network slice s, which can be expressed by $F^s = \{f_i \ | f_1, f_2, \ldots\}$. A data flow built on the network slice can be defined as $f = \{s_f, \ d_f, \ r_f\}$, where s_f, d_f relatively represent the fixed 5G slicing ingress and egress of data flow f_i, and r_f represents the data flow rate.

C. SERVICE FUNCTION CHAIN DYNAMIC RECONFIGURATION PROBLEM (SFC-DRP)

In the future 5G networks, more and more users are trying to access their customized services wherever and whenever, Internet Service Providers (ISPs) expect SFCs to be dynamic and adaptive. Thus, consolidation techniques can be applied by migrating VNFIs and SFC requires to be reconfigured according to the varying traffic demands and resource availability, so as to improve the network resource utilization, save the resource consumption and decrease the number of SLA violations. It should be noted that an SFC can be considered to be available at a given time if all associated VNFIs are not below their lower thresholds, or not exceeding their thresholds, namely, all substrate nodes have heterogenous threshold, and an SFC may be blocked due to the lack of resources. When the SFC reconfiguration reaches trigger condition and lasts for a duration, the control layer dynamically achieves SFC reconfiguration.

SFC reconfiguration involves VNFIs migration, flow re-routing, etc., which consumes additional resources and might easily lead to service interruption, thus in turn degrading the user's experience quality. The SFC reconfiguration strategy can be defined that selects VNFIs to be removed from the original VNF-capable server node, reconfigure it onto a new VNF-capable server node and flows re-routing, considering a right compromise between operation cost reduction and service provider's due to SFC reconfiguration. The objective of this paper is to determine an optimal SFC reconfiguration strategy that can well balance the service provider's revenue and operational overhead under the constraints. The major constraints are those on IT resources (i.e., CPU, storage) and bandwidth resources.

We consider a 5G based cloud-radio access network (C-RAN) in which ISPs aim to enable reconfiguration of SFCs by the advantages of NFV technology. We assume that there exists some Infrastructure Network Providers (InPs) with servers that ISPs can use them to select VNFIs to be removed from the original VNF-capable server node and reconfigure it onto a new VNF-capable server node. Each server has a limited amount of IT resources, and we assume that each server has different thresholds. Figure 1 shows an intuitive example on the dynamic SFC reconfiguration. The control layer achieves a reconfiguration of the SFC allocated by the flow re-routing and VNFIs migration in the periods when the incoming SFC request can not enough satisfy the substrate node thresholds.

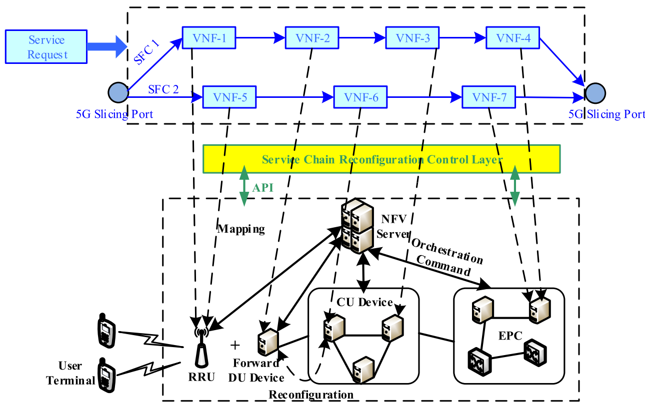


FIGURE 1. SFC Reconfiguration in 5G Based C-RAN.

Example: For simplicity, we omit that each substrate node has different thresholds. Assume that SFC 1 was deployed at the previous service time, where *User*1 accesses the network from the remote radio unit (RRU) and takes an SFC as VNF 1→VNF 4. Hence, to set up this SFC 1 for *User*1, ISPs can deploy VNF 1 on RRU, VNF 2 on distributed unit (DU) device, VNF 3 on centralized unit (CU) device, and VNF 4 on evolved packet core (EPC). Then, at the current service time, ISPs find that a new user joins at RRU to request an SFC as VNF 5→VNF 7, ISPs initially deploy VNF 5 on RRU, VNF 6 on DU device, and VNF 7 on EPC. The in-service SFC 1 served for *User*1 stayed unchanged, but the amount of requested resource exceeds the threshold of DU device. Hence, as shown in Figure 1, ISPs decide to migrate the VNF 6 to CU device where it can be efficiently shared by *User*1 and *User*2, and deploy VNF 6 on CU device for *User*2.

Remark: Some implementation issues should be justified. Firstly, the calculation of the SFC reconfiguration is operated at the controller of the substrate networks. The online network state information (e.g., network topology, service function description, network resource including bandwidth and network-wide traffic load) can be available in the controller via SDN techniques, the optimal reconfiguration strategy should be synchronized across an SFC, and the inconsistency of configurations need probably handling. Finally, the signaling cost is relatively low since we have limited the number of reconfigurations to VNFIs allocation.

IV. INTERGER LINEAR PROGRAMMING (ILP) FORMULATION

SFC-DRP is a considerably harder problem to solve than traditional Virtual Machine (VM) migration problems. There is no node ordering requirement in VM reconfiguration, while in SFC-DRP we need to preserve the ordering of VNFIs. In this section, we formulate an integer linear programming (ILP) model to exactly solve the optimal SFC-DRP.

A. SFC RECONFIGURATION OBJECTIVES

In the process of SFC dynamic reconfiguration, we mainly consider two trade-off objectives such as maximum revenue and minimizing reconfiguration cost, and both objectives are optimized from the view of ISPs. The symbols and corresponding definitions used in this paper are listed in Table 3.

1) COST

We can define the SFC reconfiguration cost as the total amount of substrate network resource that are utilized by a given SFC reconfiguration. In particular, the two main components of the SFC reconfiguration cost function are the VNFI migration cost and the bandwidth cost.

As defined in [26], the migration energy occurring during the mapping state transition is due to the moving of memories of VNFIs when these ones are migrated. In addition, the amount of packet data caused by the migration of VNFIs migrating is approximately proportional to the migration energy consumption when the VNFI mapping state transfers to another new state. Similarly, the amount of VNFI migration cost can be mainly determined by the moving of memories of VNFIs occupied on the network substrate link. In the reconfiguration process of the SFC request, we focus on how to effectively remap the VNFIs and virtual links between them. It is worth noting that we assume that the cost of achieving the remapping of VNFIs is a linear function of the binary decision variable, $x^u_{i,\ r}$. This represents the binary variable assuming the value 1 if and only if the VNFI $r \in N^v$ can be embedded in the server node $u \in N^s$; otherwise it is zero.

Let $C^t_{M,\ i}(u,\ r)$ donate the summation of VNFIs migration cost when the VNFI $r \in N^v$ is migrating to new VNF-capable server node u in the t^{th} slot, which is expressed as

$$C^t_{M,\ i}(u,\ r) = \sum_{r\in N^v}\sum_{(u,v)\in E^s}\frac{M^t(u)}{b^t(u,\ v)} \times x^u_{i,r} \tag{1}$$

where $M^t(u)$ represents the amount of memory on the VNF-capable server node $u \in N^s$ in the t^{th} slot, $b^t(u,\ v)$ represents the amount of remaining available bandwidth on the substrate link $(u, v) \in E^s$ in the t^{th} slot.

The second main component of the SFC reconfiguration cost function is the cost of re-flowing service path which is donated by C^t_B. It is worth noting that the bandwidth costs involved when VNFIs are migrating during the traffic state transitions. We can write the $C^t_{B,\ i}$ as the summation of bandwidth cost for migration of each VNFI when the VNFI r is migrating to new VNF-capable server node $u \in N^s$ in the t^{th} slot. The bandwidth cost component is a nonlinear function of the binary decision variable, $\alpha^{v_{i,r}v_{i,r+1}}_{u,v}$. This binary variable assumes the value 1 if and only if a virtual link $(v_{i,r},\ v_{i,r+1})$ is provisioned on link $(u,\ v)$; otherwise its value is zero. We represent it mathematically in (2).

$$C^t_B(u, r) = \sum_{(u,v)\in E^s}\ \sum_{(v_{i,r}v_{i,r+1})\in E^v} \alpha^{v_{i,r}v_{i,r+1}}_{u,v} \times H^{r,r+1}_i \times hop^t(u, v) \tag{2}$$

where $H^{r,r+1}_i$ represents the bandwidth usage of virtual link $(r,\ r+1)$, $hop^t(u,\ v)$ represents The shortest hop from substrate node $u \in N^s$ to substrate node $v \in N^s$.

The total SFC reconfiguration cost when the VNFI $r \in N^v$ is migrating to the new VNF-capable server node $u \in N^s$ can be mathematically formulated as follows:

$$\begin{aligned} C^t_{total}(u, r) &= \alpha \cdot C^t_M(u, r) + \beta \cdot C^t_B(u, r) \\ &= \alpha \cdot \sum_{r\in N^v}\sum_{(u,v)\in E^s}\frac{M^t(u)}{b^t(u,\ v)} \times x^u_{i,r} \\ &\quad + \beta \cdot \sum_{(u,v)\in E^s}\ \sum_{(v_{i,r}v_{i,r+1})\in E^v} \alpha^{v_{i,r}v_{i,r+1}}_{u,v} \\ &\quad \times H^{r,r+1}_i \times hop^t(u, v) \end{aligned} \tag{3}$$

where α and β represent the impact factors of the VNFI migration cost and the bandwidth cost, respectively.

2) REVENUE

As defined in [30], revenue is a measure of how efficiently substrate resources are utilized. SFC reconfiguration revenue $R^t_{total}(u,\ \ r)$ can be defined as the income from the total amount of substrate network resources that are utilized by a given SFC reconfiguration (i.e., VNFIs migration, flow re-routing). The reconfiguration revenue $R^t_{total}(u,\ \ r)$ includes the resource demands for the migration of VNFIs $r \in N^v$ on the VNF-capable server node $u \in N^s$ where it is remapped, as well as their processing times. This can be mathematically expressed as follows:

$$R^t_{total}(u,\ \ r) = \sum_{r\in N^v}\delta_r + \sum_{r\in N^v}\sum_{u\in N^s} x^u_{i,\ r} \times \rho^u_r \tag{4}$$

where δ_r represents the amount of resource that the VNFI $r \in N^v$ utilizes from the VNF-capable server node $u \in N^s$ onto which it is remapped, ρ^u_r represents the processing time for the VNFI $r \in N^v$ on the VNF-capable server node $u \in N^s$.

B. ILP FORMULATION

In this part, we formulate the optimal SFC-DRP when the objective function to be maximized. The objective function is the revenue $R^t_{total}(u,\ \ r)$, minus the SFC reconfiguration cost $C^t_{total}(u,\ r)$ (note that the first component represents the VNFI migration cost and the second component represents bandwidth cost).

The solution to the optimization problem is characterized by the following variables:

- $x^u_{i,r}$, binary variable assuming the value 1 if and only if the VNFI $r \in N^v$ can be embedded in the server node $u \in N^s$; otherwise its value is zero;
- $\alpha^{v_{i,r}v_{i,r+1}}_{u,v}$, binary variable assuming the value 1 if and only if a virtual link $(v_{i,r}, v_{i,r+1})$ is provisioned on link (u, v); otherwise its value is zero;

Two other variables are introduced to give a linear formulation of the problem:

- $\beta^r_{u,v}$, binary variable assuming the value 1 if and only if the VNFI $r \in N^v$ is migrating from initial location u to v;
- υ^r_u, binary variable assuming the value 1 if and only if the server $u \in N^s$ is switched on when a VNFI $r \in N^v$ works.

The introduced variables have to satisfy the following constraints of the optimization problem. We formulate the objective of ISPs throughout the time as an optimization problem. We assume that the purpose of ISPs is maximizing the objective function regarding the requirements of incoming reconfiguration services and InP's resource constraint. Thus, the optimization problem can be written as:

$$\underset{(u,r)}{Maximize} \quad R^t_{total}(u, r) - C^t_{total}(u, r) \tag{5}$$

$$\text{s.t.} \sum_{r \in N^v} x^u_{i,r} = 1, \quad \forall i \in [1..N-1, N], \; u \in N^s \tag{6}$$

$$\sum_{(u,v) \in E^s} \alpha^{v_{i,r}v_{i,r+1}}_{u,v} = 1 \quad \forall i \in [1..N-1, N], \quad (v_{i,r}, \; v_{i,r+1}) \in E^v \tag{7}$$

$$\alpha^{v_{i,r}v_{i,r+1}}_{u,v} \leq x^{n^v_a}_{i,n^s_a}, \quad \forall i \in [1..N-1, N], \quad n^s_a \in N^s, \; n^v_a \in N^v \tag{8}$$

$$\alpha^{v_{i,r}v_{i,r+1}}_{u,v} \leq x^{n^v_b}_{i,n^s_b}, \quad \forall i \in [1..N-1, N], \quad n^s_b \in N^s, n^v_b \in N^v \tag{9}$$

$$\sum_{SFC_i \in Q} \sum_{r=1}^{M} x^u_{i,r} \cdot c^r_i \leq R(u), \quad \forall u \in N^s, \; r \in N^v \tag{10}$$

$$\sum_{SFC_i \in Q} \sum_{r=1}^{M} \alpha^{v_{i,r}v_{i,r+1}}_{u,v} \cdot H^{r,r+1}_i \leq b(u, v), \quad \forall i \in [1 \ldots \text{N}-1, \; N], \quad (u, v) \in E^s, \; r \in N^v \tag{11}$$

$$\upsilon^f_u \leq \sum_{r \in N^v} x^u_{i,r}, \quad \forall i \in [1..N-1, N], \; u \in N^s \tag{12}$$

$$\upsilon^f_u \geq \frac{1}{|N^v|} \sum_{r \in N^v} x^u_{i,r}, \quad \forall i \in [1..N-1, N], \; u \in N^s \tag{13}$$

$$\beta^r_{n^s_a, \, n^s_b} \geq x^{n^s_a}_{i,r} - x^{n^s_b}_{i,r}, \quad \forall i \in [1..N-1, N], \quad n^s_a, n^s_b \in N^s, \; r \in N^v \tag{14}$$

$$\sum_{r \in N^v} \beta^r_{u,v} \leq NUM_{\text{SFC}_i}, \quad \forall u, \; v \in N^s, \; \text{SFC}_i \in Q \tag{15}$$

$$\sum_{SFC_i \in Q} \sum_{(u,v) \in E^s} \sum_{r=1}^{M} \alpha^{v_{i,r}v_{i,r+1}}_{u,v} \cdot d(u, v) + \sum_{SFC_i \in Q} \sum_{u \in N^s} \sum_{r=2}^{M} x^u_{i,r} \cdot dp^u_{i,r} \leq D^G_i \tag{16}$$

where the objective function is the revenue, minus the reconfiguration cost (Note that the cost includes the VNFI migration cost and the bandwidth cost). The intent of the normalization introduced in the above expression is to minimize the reconfiguration cost while maximizing the revenue. Constraint (6) ensures that each VNFI can be exactly re-hosted on only one VNF-capable server node $u \in N^s$. Constraint (7) represents the routing of one virtual link of the directed graph $G^v = (N^v, E^v)$ in only one new service function path re-hosting the SFC link. Constraints (8)-(9) guarantee that when the virtual link $(v_{i,r}, v_{i,r+1})$ is routed on the physical network path (u, v) re-hosting the SFC link, n^s_a and n^s_b must be the VNF-capable server node that the virtual node n^v_a and n^v_b are assigned to. Constraint (10) ensures that the total amount of the CPU resource occupied by the VNFIs assigned to the new substrate VNF-capable server node $u \in N^s$ should be less than or equal to the amount of available CPU capacity. Constraint (11) guarantees that the total amount of the bandwidth resource carried by the new re-hosting substrate link $(u, v) \in E^s$ should be less than or equal to its bandwidth capacity. Constraints (12)-(13) represent that any VNF-capable server is switched on when at least one VNFI that satisfies the threshold constraint hosted on the VNF-capable server. Constraint (14) represents that a VNFI is migrating when it is not hosted by the same VNF-capable server. Equation (15) makes sure that there are limited number of reconfigurations to VNFIs in the network to maintain service continuity. In order to avoid exceeding the latency requirement of an SFC, the latency constraint is formulated in Constraint (16).

As reported in [27], SFC-DRP can reduce to the Multi-dimensional Bin Packing Problem [31], Bakiras [32] and Mills *et al.* [33] have shown to be strongly NP-Hard. Therefore, the optimization problem in (5)-(16) is intractable for large networks with various services. Heuristic based techniques can be helpful to solve such intractable problem [34]–[36]. Due to high complexity of the SFC-DRP, we propose the local search algorithm based on the improved tabu search (TS) described in Section V. The goal of the heuristic technique is to find an optimal reconfiguration policy P^G which determines what reconfiguration action to take in each network state. In the following, we introduce a novel algorithm called *Tabu Search Reconfiguration based Fuzzy C-Means*(*TSRFCM*) for SFC-DRP.

V. HEURISTIC SOLUTION

Based on the complexity of the above-presented ILP, finding the optimal solution belongs to NP-Hard. To address the intractable problem, the novel heuristic algorithm is proposed to further accelerate the problem-solving, which produces solution that are very close to the optimal solution within a polynomial time. Compared to the related works [26]–[29], the advantage of our novel reconfiguration strategy focuses on the trade-off between revenue and cost. In addition, the reconfiguration cost on VNFIs migration and flow bandwidth cost, which has not been fully considered in currently mainstreaming literatures.

A. INTRODUCTION OF THE ORIGINAL TS ALGORITHM

The tabu search (TS) is a heuristic algorithm, and firstly proposed by Glover [37]. TS employs local search scheme to solve the mathematical optimization problem, and it operates by imitating the memory behavior of human beings. The main steps of the original TS consist of five components, namely, the initial solution Z, the neighborhood solution $N(Z)$, tabu list T, aspiration criterion and stopping condition. This original TS starts from an initial solution Z, a local search scheme operates in the neighborhood $N(Z)$, record the currently optimal solution Z^*, and migrate the state Z to the new state Z^*. Then the TS algorithm can not stop iterating until meeting the stopping condition. In each iteration, each "best so far" solution in each previous iteration is recorded in the tabu list T so as to prevent the circulation of searching the same local optimal solutions.

B. FORMULATION OF THE TSRFCM ALGORITHM

In SFC-DRP, four key problems (i.e., trigger condition, overload/underload VNFIs selection, target VNF-capable server node selection, and service function path re-routing) need to be considered [38]. The first main problem is to how to determine the trigger condition. Here, we consider two trigger conditions as follows. One condition is that when the substrate network utilization is less than lower threshold $\check{\xi}$, the control layer accomplishes dynamically VNFIs consolidation, and migrates all the VNFIs to the new VNF-capable server nodes. At this moment, the lower-threshold server goes into dormancy state, and reduce entire energy consumption. The other condition is that when the resource ratio ξ is higher than the upper threshold $\hat{\xi}$, the control layer migrates some VNFIs to the new VNF-capable server nodes that satisfy the resource constraints until the resource utilization of these overloaded server nodes gradually reduce in a reasonable range. In addition, frequent reconfigurations incur certain cost and might cause service interruption. The reconfiguration condition triggers if and only if the real-time utilization ratio reaches the lower/upper threshold (namely, $\check{\xi} \leq \xi_i \leq \hat{\xi}$) and lasts for a duration time τ. The second problem is overload/underload VNFIs selection, this can be solved by the calculation of real-time utilization ratio ξ. The real-time resource usage information can be available by SDN techniques, thus determining the overload/underload VNFIs that are to be migrated. The last two problems can be solved via TSRFCM (Our proposed TSRFCM algorithm is discussed in more detailed below).

Our proposal is an application of TS. However, original TS is strongly dependent on the initial solution [37]. A random initial solution will easily reduce the convergence speed. To address this narrow, we extend the original TS algorithm with Fuzzy C-Means (FCM) algorithm that is used to obtain the optimal SFC reconfiguration strategy P^G. The proposed TSRFCM based SFC reconfiguration algorithm can be decomposed into these steps: Firstly, an initial solution $Z = (U^*, V^*)$ can be determined via FCM, where U^* represents fuzzy membership matrix, and $V^* = (N_m^v, N_m^s)$ represents the initial targeted physical node (PN) in the tabu list T. Secondly, the tabu search runs in the neighborhood $N(Z)$, the candidate PNs are judged via tabu scheme and tabu list can be constantly updated. Then, when the algorithm reaches maximum number of iterations Max_GEN, the "best so far" solution in T can be regarded as the optimal targeted PN. Eventually, the re-routing of service function path can be determined via K-Dijkstra.

The main process of the TSRFCM algorithm is discussed in detail as follows:

1) FUZZY C-MEANS INITIAL SOLUTION

Once the real-time utilization ratio ξ reaches trigger condition, the control layer is achieving VNFIs migration and flows re-routing. An initial solution $Z = (U^*, V^*)$ is determined via the FCM algorithm, which prioritizes the VNFIs migrating onto the PNs with maximized objective. This FCM based initial procedure mainly runs in two steps. First, for each VNFI i to be migrated, a candidate PN j is selected from set $n^s \subseteq N^s$ via FCM while simultaneously meeting the restrictions $\check{\xi} \leq \xi_i \leq \hat{\xi}$, namely, the utilization ratio ξ_i of the incoming request is more than the a lower threshold $\check{\xi}$ or less than the upper threshold $\hat{\xi}$. Then, starting with the first VNF along SFC, the selected overload/underload VNFIs $F_{mig} \subseteq N^v$ are migrated onto the new PNs, considering all the constrains as described in Section IV-B. Thus, $V^* = (n_m^v, n_m^s)$ is set as initial solution of tabu search algorithm, where $n_m^v \in N^v$ represents the selected VNFIs to be migrated, and $n_m^s \in N^s$ represents the targeted PN set. The current solution Z can be seen as the initial solution Z_0 of the TS algorithm.

2) NEIGHBORHOOD SOLUTION

Neighborhood function is employed that finds another solution Z' that is better than the "best so far" solution Z. Here, $N(Z)$ is defined that involves all possible solutions in neighborhood of Z', which results from the VNF to be migrated from one PN to a new PN, which satisfy the restrictions $\check{\xi} \leq \xi_i \leq \hat{\xi}$. To avoid searching all the solutions in the solution space, we restrict the neighborhood to be based on the changes in the migration of the VNFIs with maximized optimization objective value. The available PN j with the maximized optimization objective is selected as a candidate for VNFI migration. If the VNFI to be migrated is successfully mapped to the candidate PN j, then store this VNFI in set $N_m^s \subseteq N^s$; otherwise, trace back to the suboptimal solution in set $N_{sub}^s \subseteq N^s$. After the VNFIs migration, all the proceeding VNFs are evaluated, thus ensuring that the total optimization objective is maximum.

3) TABU LIST

The tabu list T is used to avoid searching the optimal solutions in a loop way, and select a better solution than the "best so far" solution. Assume that a VNFI i has been migrated from PN j_1 to PN j_2, tabu can be defined that forbids the VNFI i moving back to j_1 during the next m-1 iterations (the given maximum number of iterations is m). This ensures that the VNFI i have m-1chances to search a better migration solution before the VNFI i can be moved back to PN j_1. The tabu list consists of two main metrics including tabu target j_1 and tabu length λ. Tabu scheme can also select a solution with a higher reconfiguration cost than the "best so far" solution. Therefore, it is assumed that the defined tabu list in this paper can be defined as two-tuple $T\{i, j_1\}$, which is recorded in short-term memory.

4) ASPIRATION CRITERION

We adopt the aspiration criterion that avoid all candidate solution being in the tabu list when TS runs. We take the aspiration criterion into consideration that allows for a tabu move when it results in a solution with a higher objective than that of the "best so far" solution Z^*, this is because that the solution resulting from moving VNFI i to PN j_1 has not been previously visited. Furthermore, when all available moves are all regarded as tabu due to causing a lower objective than that of the "best so far" solution Z^*, a "least tabu" move is determined that is the move with highest objective of all the tabu moves, then the obtained solution is the initial solution for the next iteration round.

5) STOPPING CONDITION

Stopping condition is employed to ensure the time complexity of TS algorithm and terminate the TS algorithm. Here, two stopping criterions are defined that determines when TS algorithm stops. The specific stopping criterions are as follows: (i) the maximum tabu visit frequency of a solution is set, that is, there is no feasible solution in the neighborhood of solution Z for all VNFIs; (ii) the maximum iterative step *Max_GEN* is set, namely, the solution Z is without an improvement after *Max_GEN* continues iterations.

Based on the above description, we input all algorithm parameters, namely, physical network graph G^s, VNFI graph G^v, network resources (i.e. bandwidth and network-wide traffic workload), overloaded substrate nodes $S \in N_m^s$ and the set of VNFList on $S_m \in N_m^s$, and call the TSRFCM algorithm to obtain a reconfiguration policy P^G which determines what action to take in each network state. The pseudocode of the proposed TSRFCM algorithm is shown in **algorithm 1**.

Algorithm 1 TSRFCM Algorithm for SFC-DRP

Initialize migration VNFIs $N_m^v \subseteq N^v$, targeted PNs $N_m^s \in N^s$, $c_{min} \to \infty$
procedure INITIAL SOLUTION
  for each incoming requested VNF $n_i^v \in N^v$ **do**
    for each PN $n_j^s \in N^s$ **do**
    if the real-time utilization ratio ξ ranges in $\check{\xi} \le \xi_i \le \hat{\xi}$, and lasts for a duration τ **then**
      Determine the VNFIs $n_m^v \in N^v$ to be migrated, and add an available PN $n_j^s \in N^s$ to subset N_m^s, N_m^s contains the PNs that satisfy the resource constraints;
      else
      continue
    end for
  end for
  Search for the available PN set of the substrate node n^s that satisfies the restrictions via the FCM algorithm, calculate the maximized objective function o_i, and determine the targeted PN set $N_m^s \in N^s$;
  if $o_i > o_{max}$ & $n_j^s \notin \complement_{N^s} N_m^s$ **then**
    $o_{max} \leftarrow o_i$;
    $N_m^s \leftarrow n_j^s$;
    $N_m^v \leftarrow n_i^v$;
  end if
  while $n_j^s \in \complement_{N^s} N_m^s$ **do**
    Trace back to the suboptimal solution, and store it targeted PN set $n_m^s \in N^s$;
  end while
end procedure
procedure TABU SEARCH BASED VNFIS MIGRATION
Determine initial solution $Z_0 = (N_m^v,\ N_m^s)$;
if Z_0 notPossible **then**
  SFC reconfiguration failed;
  return
end if
Initialize reconfiguration solution $Z = Z_0$, optimal solution $Z^* = Z_0$, aspiration value $A(Z^*) = c(Z^*)$;
Determine VNFI i to move and its neighborhood $N(Z)$;
while Stopping condition notMeet **do**
  for candidate set $C \in N(Z)$ **do**
    if C notViolate T_l **then**
      Reconfiguration solution set $X = X \bigcup C$;
    end if
  end for
  $C = BestSoFar\,(X)$;
  $Z = C$;
  Update tabu list T;
  if $o(X) > o(X^*)$ **then**
    The features are added to the tabu list addNewTabu;
    $Z^* = X$;
  end if
end while
end procedure
procedure FLOW RE-ROUTING

Algorithm 1 *(Continued.)* TSRFCM Algorithm for SFC-DRP

```
    for virtual link candidate d ∈ E^v
        Determine the shortest path p with end substrate network node where to route the virtual link d;
        if virtual link successfully mapping p ← d then
            α_{u,v}^{v_{i,r} v_{i,r+1}} = 1;
        else
            Trace back to the suboptimal solution in E^v;
        end if
    end for
end procedure
```

Upon the SFC reconfiguration controller, the decision module runs the TSRFCM algorithm and achieves the results, i.e., the migration VNFI node $VNF_m \in N_m^s$, the targeted PN $S_d \in N_m^s$, the maximum objective $Obj = R_{total}^t(u, r) - C_{total}^t(u, r)$.

C. COMPLEXITY ANALYSIS

Next, we report the complexity analysis of our proposal. Our proposed TSRFCM algorithm aims to work around the complexity of the formulated ILP. Let the number of the VNFI nodes, the virtual links and the substrate nodes be $|N|$, $|E|$, and $|M|$. For each network service, the main processes of the proposed TSRFCM algorithm are: *i*) creating initial solution; *ii*) the generating neighborhood solution; *iii*) determining tabu list; *iv*) removing tabu list. The first process is based on the FCM algorithm, it is well known that this process follows the order of $O(|M| \cdot |N| \cdot |t|)$, where $|t|$ represents the number of iterations of FCM. In the second part of TS, the generation of neighborhood solution has a complexity of $O(C_{|N|\cdot|E|}^2)$. The rest process of the time complexity is mainly determined by the length of tabu list, which is of the order of $O(|N| \cdot |E| \cdot \lambda)$, where λ represents the length of tabu list. In the flow rerouting process of the proposed TSRFCM algorithm, the K-Dijkstra algorithm has a complexity of $O(K + |N| \log |N| + |M|)$. To determine the migration of VNFIs and re-routing service function paths, the TS algorithm also requires a finite number of iterations, namely, *Max_GEN* (e.g., an upper bound of maximum number of iterations). In conclusion, the maximum run time is $O(|M| \cdot |N| \cdot |t| + Max\_GEN \times (|N|^2 \cdot |E|^2 + |N| \cdot |E| + \lambda) + K + |N| \log |N| + |M|)$ for the process of one SFC reconfiguration. Thus, TSRFCM belongs to polynomial time algorithm.

VI. NUMERICAL RESULTS

The performance evaluation focused on the following aspects: (i) compare the results of the proposed TSRFCM based SFC reconfiguration algorithm to the results of the ILP based solution via the CPLEX tool on three scales of ISP simulation networks; (ii) analyze the performance of the proposed TSRFCM based SFC reconfiguration algorithm on two different substrate network topologies that are generated by GT-ITM software. Compare the performance of the proposed solution to that of the RLACM, GA, No-reconfiguration-aware based solution that have already been studied for solving SFC-DRP. The proposed TSRFCM is evaluated in terms of the reconfiguration blocking ratio, reconfiguration cost and revenue and reconfiguration time. Besides, we treat the cost of VNFI migration and bandwidth resources equally in the optimization, and thus the simulations use $\alpha = \beta = 1$ in the objective. Moreover, all the experiments were executed on a personal computer with a 3.6 GHz dual core Intel® CoreTM i7 processor and 8 GB RAM. All statistics in the simulation are the average results.

A. ILP VS. TSRFCM

To evaluate how close our provided solutions to the exact one, we use MATLAB 2013a with the CPLEX toolbox to solve the ILP solver corresponding to TSRFCM. Three scales ISP simulation networks are considered, namely: (i) an ISP simulation network consisting of 10 nodes and hosting 5 network services, (ii) an ISP simulation network consisting of 20 nodes and hosting 10 network services, and (iii) an ISP simulation network consisting of 40 nodes and hosting 20 network services. The detailed results of this comparison are presented in Table 1.

TABLE 1. ILP vs TSRFCM.

Substrate topology	Solution	Objective	Run time
Small scale (10 nodes, 5 services)	ILP	5525	5.82s
	TSRFCM	**5450**	**0.93s**
Medium scale (20 nodes, 10 services)	ILP	12890	623.5s
	TSRFCM	**12635**	**2.6s**
Large scale (40 nodes, 20 services)	ILP	N/A	∞
	TSRFCM	**25760**	**8.2s**

As indicated in the table, the CALEX solution obtained an objective value when the proposed TSRFCM solution could not. This was because the typical CPLEX tool used to solve the ILP provided an exact solution, albeit at the expense of execution time. As the number of physical nodes increased, the ILP solution fails to obtain the objective value. Therefore, the CALEX solution had limited applicability due to its computational complexity, especially in the cases with a medium or large ISP network. In contrast, the proposed TSRFCM solution iteratively approached a near-optimal solution within a polynomial time and therefore satisfied the execution time requirement in the large ISP network. In conclusion, our proposed TSRFCM solution can provide solutions that are very close to the optimal one, and its execution time is several orders of magnitude faster than the exact solution.

B. FURTHER DISCUSSION

In this section, we conduct numerical simulations to evaluate the effectiveness of the proposed TSRFCM algorithm. In addition, our proposal is compared to the currently mainstreaming SFC reconfiguration algorithms. The notations and descriptions of the different compared algorithms are listed in Table 2.

TABLE 2. Comparison of algorithms.

Algorithm	Description
RLACM [27]	Revenue loss aware choosing mapping based on policy-iteration strategy
GA [39]	Solving SFC-DRP based on Genetic Algorithm (GA)
Baseline [40]	Solving SFC Resource Allocation (SFC-RA) without considering reconfiguration cost during optimization
TSRFCM	Solving SFC-DRP based on Tabu Search Reconfiguration based on Fuzzy C-Means

1) EXPERIMENTAL SETUP

In the following, setup parameters were introduced as follows. For the request dataset, the online service request arrives following a Possion process, and the service time follows an exponential distribution. Each service chain request consists of different types of functions, the number of which was randomly distributed in range [2], [5]. And each VNF-capable node can hold 5 types of services, and the evaluation is carried out where commercial appliances are used with the values of 120 ms, 160 ms, 160 ms, 120 ms, and 80 ms for the packet processing times of FW, IDS, NAT, DPI and Proxy respectively. The latency on the link was assumed to be in range of (0,100] ms, which follows the distribution of the stationary random process. According to [41], the lower/upper thresholds for the VNF-capable nodes were fixed to 0.2, 0.8 respectively. In the proposed TSRFCM algorithm, the size of candidate solution $N(Z)$ was fixed to 15, the length of tabu list λ was fixed to 6, the maximum number of algorithm iterations Max_GEN was fixed to 50, and the duration τ of reconfiguration trigger was fixed to 2 time units. For the topology dataset, we have used GT-ITM software [42] to generate random graphs for testing. GT-ITM is a well-known data analysis and visualization tools for graphs. A sample with 100 substrate nodes (ten of the total nodes in the network are VNF-capable server nodes) is shown in Figure 2. The amount of IT resources and bandwidth are also provided by GT-ITM.

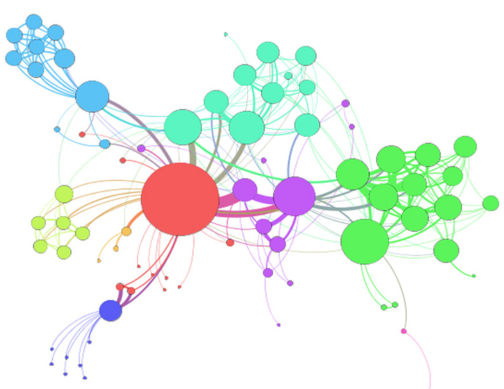


FIGURE 2. A sample 100 node topology generated using GT-ITM.

2) PERFORMANCE METRICS

We use the following metrics to evaluate the performance of algorithm. In the scenario of two different network sizes (i.e., 100-node topology, 200-node topology), we measure the SFC reconfiguration cost, revenue, resource utilization, numbers of SFC mapping changes, reconfiguration time and algorithm run time. The detailed explanations of these metrics are as follows:

- **The SFC Reconfiguration Cost and Revenue**: The total cost and revenue of using substrate network resources for SFC remapping and reconfiguring all SFC requests. These two metrics can be calculated by (3) and (4), respectively.
- **Resource Utilization**: This metric is the total load accumulation on the service function path, and the resource utilization can be quantified as a function of the state difference of an SFC after reconfiguration. The whole network resource utilization χ_i of each SFP can be mathematically computed as follows:

$$\chi_i = \omega_1 \cdot \frac{\sum\limits_{(u,v)\in E^s} \sum\limits_{SFC_i\in Q} \sum\limits_{r=1}^{M} \alpha_{u,v}^{v_{i,r}v_{i,r+1}} \cdot H_i^{r,r+1}}{\sum\limits_{(u,v)\in E^s} b(u,v)} + \omega_2 \cdot \frac{\sum\limits_{u\in N^s} \sum\limits_{SFC_i\in Q} \sum\limits_{r=1}^{M} x_{i,r}^{u} \cdot c_i^r}{\sum\limits_{u\in N^s} R(u)} \quad (17)$$

where ω_1 and ω_2 represent the impact factors of the VNF-capable server node load and the substrate link load, respectively.

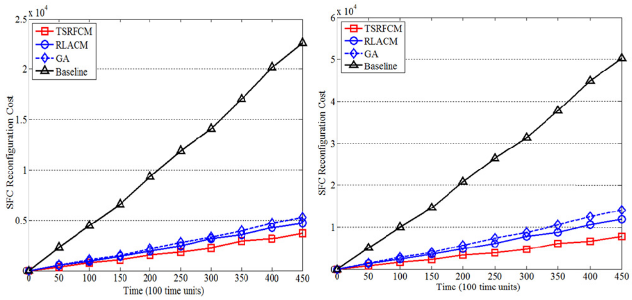
- **Number of SFC Mapping Changes**: The SFC reconfiguration can be performed when the VNFIs are reaching the reconfiguration condition. Define NUM_t to be the number of VNFI $r \in N^v$ migrating from initial location u to v in the time slot t when the requested resources reach the reconfiguration thresholds of the VNF-capable server node $u \in N^s$. Now, we can the number of SFC mapping changes NUM_t as follows:

$$NUM_t = \sum_{r\in N^v} \beta_{u,v}^{r}, \quad \forall u,\ v \in N^s \quad (18)$$

- **Average SFC Reconfiguration Time**: The average reconfiguration time of SFC request is the reconfiguration time of VNFIs and flows that lastly complete reconfiguration. The average reconfiguration time can be calculated similar with literature [43]. Therefore, the average reconfiguration time of each SFC request can be expressed as follows:

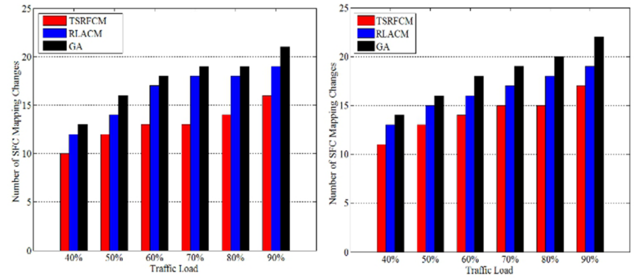
$$T_{rec}^{average} = \frac{\sum\limits_{SFC_i} T_{i,\,rec}}{|SFC_i|} = \frac{\sum\limits_{SFC_i} \sum_{i=1}^{n_i+1} \frac{V_i \cdot (1-r_i^{n_i+1})}{B_i \cdot (1-r_i)}}{|SFC_i|} \quad (19)$$

where n_i represents the actual number of iterations, and the reconfiguration was stopped at the $(n_i+1)^{th}$ procedure. V_i is the size of original memory of the i^{th} VNFI, and r_i is the ratio of the dirtying rate to the transmission rate B_i.



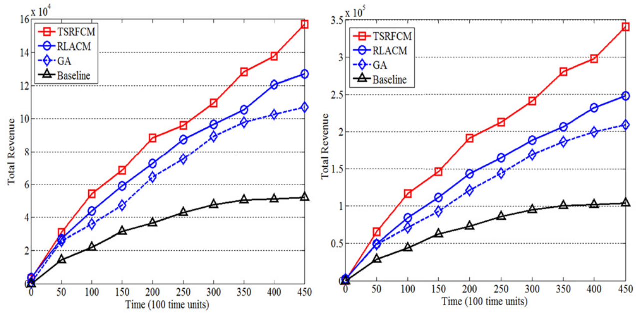
(a) 100-node topology (b) 200-node topology

FIGURE 3. Effect on SFC reconfiguration cost.



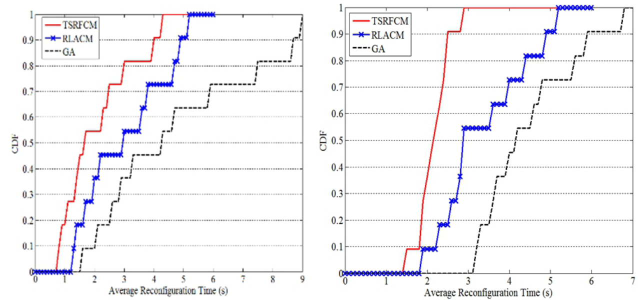
(a) 100-node topology (b) 200-node topology

FIGURE 6. Effect on number of SFC mapping changes.



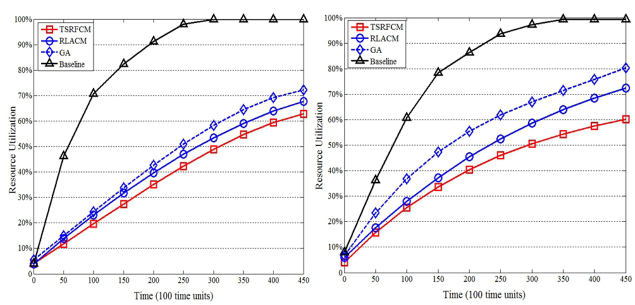
(a) 100-node topology (b) 200-node topology

FIGURE 4. Effect on total revenue.



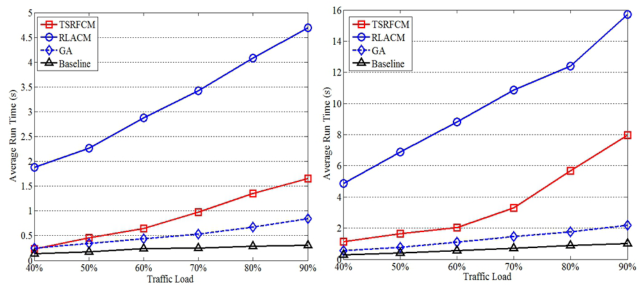
(a) 100-node topology (b) 200-node topology

FIGURE 7. Effect on CDF of reconfiguration time.



(a) 100-node topology (b) 200-node topology

FIGURE 5. Effect on resource utilization.



(a) 100-node topology (b) 200-node topology

FIGURE 8. Algorithm run time under different traffic load.

- **Algorithm Run Time**: The run time of algorithm is a measure how efficiency these reconfiguration algorithms operate. Note that the computation time could be reduced if we used a more powerful computing platform other than a personal computer.

3) EXPERIMENTAL RESULTS

The results of these simulations are shown in Figs.3-8. In the following, we present these results.

Metric 1 (SFC Reconfiguration Cost and Revenue): Figure shows the SFC reconfiguration costs and revenues as defined in Section IV. This objective function is a measure of how close these reconfiguration algorithms reaches the optimal solution. The values shown are cumulative, implying that after every successful remapping. Figure 3 shows the accumulated reconfiguration cost of the observed SFC under two different network sizes (100-node, 200-node). The SFC reconfiguration cost of RLACM, GA and TSRFCM is much reduced, compared to *baseline* scheme. We observe better performance by the proposed TSRFCM even at larger topologies and service arrivals as well. For example, for 100-node topology at 20000 time units the costs observed using the TSRFCM, RLACM, GA solution are approximately 2230, 2360, 2410; while using the *baseline* solution, the costs are approximately 9880, an increase of almost 75.6%. This is because TSRFCM is iteratively improved by searching for better solutions in its neighborhood, which might result in lower SFC reconfiguration cost. The fact that *baseline* performs worst because as a non-reconfiguration algorithm, it performs SFC resource allocation without considering reconfiguration cost during optimization, which would cause more unnecessary resource consumption. Eventually, we observe that the most related works, i.e., RLACM, performs better than GA that finds the optimal reconfiguration policy based on the iterative method. This is because a genetic algorithm, GA can be easily trapped by local optima when the problem's scale becomes larger and thus it cannot follow the increasing trend of the RLACM algorithm does, while RLACM is based on policy-iteration method at the cost of increasing time complexity. Looking at the cumulative revenue profiles in Figure 4 shows a profile similar to that of the SFC reconfiguration cost in Figure 3, where the obtained objective value is higher, the revenue is higher. An algorithm that can iteratively obtain a higher objective value is likely to have a high revenue in the long run, which would lead to better profitability for infrastructure providers.

Metric 2 (Resource Utilization): Since SFC resource allocation is performed at a large time scale, the resource of SFC might not be fully utilized due to the mismatch of service demand and resources. Figure 5 shows the resource utilization comparison with the service arrivals under two different network sizes (100-node topology, 200-node topology). The resource utilization ratio drastically changes over periods, which is caused by dynamic users' demands. Higher reconfiguration cost might result in higher resource utilization. As noted from the Figure 5 (a) and (b), the resource utilization of the *baseline* scheme is higher, while that of our proposed TSRFCM is much lowest. This indicates that the *baseline* scheme can not well adapt to the varying traffic demand and resource availability. The reason why RLACM performs only slightly better than GA is because RLACM has a lower cost during optimization, which gives RLACM an advantage in utilizing the substrate resources while balancing the load. This can be explained as that RLACM can reserve partial resources for future traffic, while GA is biased towards favoring allocating more resources to flows with lower operation overheads (e.g., with shorter paths), which could possibly lead to bottleneck resources and poor load-balancing. Finally, TSRFCM has a considerably lower value of resource utilization compared to other algorithms. This is because this algorithm specifically searches for better solutions with the objective of minimizing the bandwidth consumption and VNFI migration cost at the same time.

Metric 3 (Number of SFC Mapping Changes): Figure 6 shows the number of SFC mapping changes for 40% and 90% traffic loads under two different network sizes (100-node, 200-node). The results indicate that TSRFCM can cause the minimum number of SFC mapping changes. In contrast, GA is less effective and the other related work, namely, RLACM, lies in the middle of the compared schemes. Large number of mapping changes incur certain reconfiguration cost and might cause service interrupt, higher SFC reconfiguration costs might result in the larger number of mapping changes. The reason why TSRFCM can perform well could be attributed to the iterative strategy, which has a chance to improve the solution, and thus reducing the resource utilization and the number of SFC mapping changes. The reason why RLACM performs only slightly better than GA could be due to the fact that RLACM uses the exact based method, which confirms our analysis that RLACM can reserve partial resources for future traffic and obtain lower resource consumption for the service provider at the cost of time complexity when the traffic load gradually becomes larger.

Metric 4 (Reconfiguration Time): Figure 7 shows the cumulative distribution function (CDF) for all the flows when using the 100-node topology and 200-node topology. This cumulative distribution function can clearly show the reconfiguration time distribution of all SFCs over the entire time period. The higher the curvature of a CDF curve, the more concentrated the delay value distribution. As noted form figure, we can observe that the average reconfiguration time increases with service request arrivals. Moreover, TSRFCM performs significantly better than the compared algorithms, while GA has a highest reconfiguration time, and RLACM lies in the middle of the compared algorithms. The average reconfiguration time of the SFC request of the TSRFCM algorithm is lower than that of the tested algorithms due to ours uses local search scheme to migrate the SFC migration request and also creates an initial solution via FCM that can reduce the reconfiguration time of the SFC migration request. Next, the reason why RLACM performs only slightly worse than TSRFCM could be because RLACM is the application of policy-iteration strategy, which requires the policy evaluation after each iteration that could result in more reconfiguration time. Eventually, as the number of user requests increases, the advantages of both TSRFCM and RLACM over GA in reconfiguration time becomes more and more significant. This is because as a heuristic algorithm, GA can be easily trapped by local optima when the problem's scale becomes larger and thus it needs more time to converge.

Metric 5 (Algorithm Run Time): Figure 8 shows the run time for 40% and 90% traffic loads under two different network sizes (100-node topology, 200-node topology). Towards the end of the simulation (approximately 90% traffic load), the run time in 100-node network topology of our proposed TSRFCM algorithm is 1.6 s, the run time of the GA algorithm is 0.9 s, and the run time of the RLACM algorithm is 4.6 s, while the run time of the *baseline* scheme is 0.3 s. The results show that the run time of the *baseline* scheme is at least 3 times longer than that of these SFC reconfiguration schemes. This is because the *baseline* scheme did not consider the reconfiguration when designing our SFC-RA algorithm, and hence there is no more operation time it takes to transit to the new state. The reason why the GA algorithm performs best among the SFC reconfiguration schemes is because GA searches for an available state based on the iterative method, which might quickly convergence to the local optima and drop out of loop, and hence this method can cause less computation time. GA performs $O(|M| \cdot |N|)$ computations (for details, refer to Lu *et al.* [44]). Next, we found that RLACM performs much worse than TSRFCM is because RLACM operates based on the policy-iteration method that performs $O(|M|^2 |N|^2)$ computations (for details, refer to Eramo *et al.* [27]). Combined with the time complexity analysis presented in Section V, the trend of the corresponding

TABLE 3. List of symbol and corresponding definition.

Symbol	Definition
Q	The set of service chain requests
N	The number of service chain requests
M	The number of VNFs
N^s	The set of physical nodes
E^s	The set of physical links
N^v	The set of VNFs
E^v	The set of logical links between VNFs
$H_i^{r,r+1}$	The bandwidth usage of virtual link $(r, r+1)$
S_i	The occupied bandwidth by service chain i
c_i^r	The CPU usage of the VNFI node r
S_i	The occupied bandwidth resource along the service chain
$M^t(u)$	The storage resources for the server node $u \in N^s$
$C^t_{M,i}(u,r)$	The summation of migration cost of VNFIs migrating from substrate node $v \in N^s$ to substrate node $u \in N^s$
$C^t_B(u,r)$	The summation of bandwidth cost of VNFIs migrating from substrate node $v \in N^s$ to substrate node $u \in N^s$
$hop^t(u, v)$	The shortest hop from substrate node $u \in N^s$ to substrate node $v \in N^s$
$R(u)$	The remaining CPU resources for the server node $u \in N^s$
$b(u, v)$	The remaining bandwidth resource for the link (u, v)
NUM_{SFCi}	The maximum number of reconfigurations to VNFIs in the network to maintain service continuity.
D_i^G	The maximum latency of an SFC $SFC_i \in Q$ based on SLA
$x_{i,r}^u$	A binary variable that equals 1 if and only if the VNFI $r \in N^v$ can be embedded in the server node $u \in N^s$
$\alpha_{u,v}^{v_{i,r},v_{i,r+1}}$	A binary variable that equals 1 if and only if a virtual link $(v_{i,r}, v_{i,r+1})$ is provisioned on link (u, v)
$\beta_{u,v}^r$	A binary variable that equals 1 if and only if the VNFI $r \in N^v$ is migrating from initial location u to v
υ_u^r	A binary variable that equals 1 if and only if the server node $u \in N^s$ is switched on when a VNFI $r \in N^v$ works

curves in Figure is basically consistent with the theoretical analysis.

VII. CONCLUSION

To solve the SFC-DRP in dynamic NFV-enabled scenarios, an optimal model with applicable constraints is formulated. This problem has been solved via the integer linear programming (ILP) model. However, this ILP model has limited applicability due to computational complexity. To reduce the algorithm time complexity, we designed an approximation algorithm based on it and proposed an effective heuristic to further accelerate the problem-solving. Furthermore, simulation results showed that our proposed TSRFCM algorithm significantly outperformed the most related algorithm in terms of the service provider's reconfiguration cost, revenue, resource utilization and reconfiguration time. In the future, we will focus on the SFC reconfiguration algorithm in combination with machine-learning based methods. Since our proposed TSRFCM depends on online information of resource utilization and traffic demand, we will investigate Deep Learning (e.g., GNN) to predict the trends of link/node load and traffic variations, so as to improve the SFC reconfiguration algorithm's efficacy.

APPENDIX

The symbols and corresponding definitions used in this paper are listed in Table 3.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for the instructive comments.

REFERENCES

[1] D. Bhamare, R. Jain, A. Erbad, and M. Samaka, "Optimal virtual network function placement in multi-cloud service function chaining architecture," *Comput. Commun.*, vol. 102, pp. 1–16, Apr. 2017.

[2] L. Gupta, R. Jain, D. Bhamare, and A. Erbad, "The P-ART framework for placement of virtual network services in a multi-cloud environment," *Comput. Commun.*, vol. 139, pp. 103–122, May 2019. [Online]. Available: https://doi.org/10.1016/j.comcom

[3] M. A. Sharkh, M. Jammal, A. Shami, and A. Ouda, "Resource allocation in a network-based cloud computing environment: Design challenges," *IEEE Commun. Mag.*, vol. 51, no. 11, pp. 46–52, Nov. 2013.

[4] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Hong Kong, Apr./May 2015, pp. 1346–1354.

[5] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, "Network slicing for 5G: Challenges and opportunities," *IEEE Internet Comput.*, vol. 21, no. 5, pp. 20–27, Sep. 2017.

[6] *Network Functions Virtualization (NFV): Use Cases*. Accessed: Oct. 2013. [Online]. Available: http://www.etsi.org/deliver/etsigs/nfv/001/099/001/01.01.01 60/gsnfv001v010101p.pdf

[7] *Problem Statement for Service Function Chaining*, document RFC 7498, Apr. 2015. [Online]. Available: https://www.rfc-editor.org/rfc/pdfrfc/rfc7498.txt.pdf

[8] D. Bhamare, A. Erbad, M. Samaka, M. Zolanvari, and R. Jain, "Efficient virtual network function placement strategies for cloud radio access networks," *Comput. Commun.*, vol. 127, pp. 50–60, Sep. 2018.

[9] 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, Sep. 2016.

[10] Z. Liu, S. Wang, Y. Xie, and Y. Wang, "Service function chaining resource allocation: A survey," 2016, *arXiv:1608.00095*. [Online]. Available: https://arxiv.org/abs/1608.00095

[11] J. Martins, M. Ahmed, V. Olteanu, M. Honda, R. Bifulco, F. Huici, and C. Raiciu, "ClickOS and the art of network function virtualization," in *Proc. 11th USENIX Conf. Netw. Syst. Design Implement.*, Apr. 2014, pp. 459–473.

[12] L. Mamatas, S. Clayman, and A. Galis, "Information exchange management as a service for network function virtualization environments," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 564–577, Sep. 2016.

[13] M. Chiosi. (2012). *Network Functions Virtualisation*. [Online]. Available: https://portal.etsi.org/nfv/nfv white paper.pdf

[14] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2014, pp. 7–13.

[15] M. T. Beck and J. F. Botero, "Coordinated allocation of service function chains," in *Proc. IEEE Global Commun. Conf.*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[16] M. F. Bari, S. R. Chowdhury, R. Boutaba, and R. Ahmed, "On orchestrating virtual network functions," in *Proc. 11th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2015, pp. 50–56.

[17] M. Ghaznavi, A. Khan, N. Shahriar, K. Alsubhi, R. Ahmed, and R. Boutaba, "Elastic virtual network function placement," in *Proc. 4th IEEE Conf. Cloud Netw. (CloudNet)*, Oct. 2015, pp. 1–6.

[18] Y. Liu, Y. Lu, L. Chen, X. Li, W. Qiao, and X. Chen, "A dynamic placement mechanism of service function chaining based on software-defined networking," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 10, pp. 4640–4661, 2018.

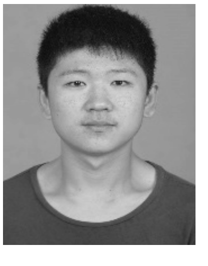
[19] J. F. Riera, X. Hesselbach, J.-F. Botero, M. Szostak, and M. Zotkiewicz, "Modelling the NFV forwarding graph for an optimal network service deployment," in *Proc. IEEE ICTON*, Jul. 2015, pp. 1–4.

[20] Q. Yuan, H. Tang, X. Wang, Y. Zhao, and W. You, "Virtual network function scheduling via multilayer encoding genetic algorithm with distributed bandwidth allocation," *Sci. China Inf. Sci.*, vol. 61, no. 9, pp. 93–111, Sep. 2018.
[21] J. Fan and M. H. Ammar, "Dynamic topology configuration in service overlay networks: A study of reconfiguration policies," in *Proc. 25TH IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Barcelona, Spain, Apr. 2006, pp. 1–12.
[22] I. Fajjari, N. Aitsaadi, H. Zimmermann, and G. Pujolle, "VNR algorithm: A greedy approach for virtual networks reconfigurations," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Houston, TX, USA, Dec. 2011, pp. 1–6.
[23] T. Wen, H. Yu, L. Liu, and G. Sun, "Network function consolidation in service function chaining orchestration," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.
[24] M. Ghaznavi, A. Khan, N. Shahriar, K. Alsubhi, R. Ahmed, and R. Boutaba, "Elastic virtual network function placement," in *Proc. 4th IEEE Conf. Cloud Netw. (CloudNet)*, Oct. 2015, pp. 1–6.
[25] V. Eramo, M. Ammar, and F. G. Lavacca, "Migration energy aware reconfigurations of virtual network function instances in NFV architectures," *IEEE Access*, vol. 5, pp. 4927–4938, 2017.
[26] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2008–2025, Aug. 2017.
[27] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 3, pp. 543–553, Sep. 2017.
[28] G. Wang, G. Feng, S. Qin, R. Wen, W. Tan, and T. Q. S. Quek, "Reconfiguration in network slicing—Optimizing the profit and performance," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 519–605, Jun. 2019.
[29] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and S. Davy, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, Apr. 2015, pp. 1–9.
[30] C. C. Chiou, "Transshipment problems in supply chainsystems: Review and extensions," in *Supply Chain*. Rijeka, Croatia: InTech, 2008.
[31] S. Bakiras, "Approximate server selection algorithms in content distribution networks," in *Proc. IEEE Int. Conf. Commun.*, May 2005, pp. 1490–1494.
[32] K. Mills, J. Filliben, and C. Dabrowski, "Comparing VM-placement algorithms for on-demand clouds," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2011, pp. 91–98.
[33] H. Goudarzi and M. Pedram, "Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems," in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2011, pp. 324–331.
[34] Y. O. Yazir, C. Matthews, S. Neville, A. Guitouni, S. Ganti, Y. Coady, and R. Farahbod, "Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2010, pp. 91–98.
[35] Y. Awano and S.-I. Kuribayashi, "A joint multiple resource allocation method for cloud computing environments with different QoS to users at multiple locations," in *Proc. IEEE Pacific Rim Conf. Commun., Comput. Signal Process. (PACRIM)*, Aug. 2013, pp. 1–5.
[36] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, 1986.
[37] X. Zhu and Y. Wang, "Research on virtual network function (VNF) migration," *J. Nanjing Univ. Posts Telecommun. (Natural Sci. Ed.)*, vol. 38, no. 1, pp. 45–53, 2018.
[38] Z. Chen, G. Feng, Y. Zhou, and B. Liu, "Delay optimization oriented service function chain migration and re-deployment in operator network," *Chin. J. Electron.*, vol. 46, no. 9, pp. 2229–2237, Sep. 2018.
[39] Y. Liu, Y. Lu, X. Chen, and W. X. Qiao, "A dynamic composition mechanism of security service chaining oriented to SDN/NFV-enabled networks," *IEEE Access*, vol. 6, pp. 53918–53929, 2018.
[40] *Google Apps Service Level Agreement*. [Online]. Available: http://www.google.com/apps/intl/en/terms/sla.html
[41] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an Internetwork," in *Proc. IEEE (INFOCOM)*, Mar. 1996, pp. 594–596.
[42] F. Callegati and W. Cerroni, "Live migration of virtualized edge networks: Analytical modeling and performance evaluation," in *Proc. IEEE SDN Future Netw. Services (SDN4FNS)*, Nov. 2013, pp. 1–6.
[43] Y. Lu, Y. Liu, X. Chen, and X. Li, "Research on placement algorithm of service function chaining oriented to software defined networking," *J. Electron. Inf. Technol.*, vol. 41, no. 1, pp. 74–82, 2019.

YICEN LIU received the B.S. and M.S. degrees in communication and information system from Army Engineering University, Shijiazhuang, China, where he is currently pursuing the Ph.D. degree in communication and information system. His current research interests include SDN/NFV/AI technology and network slicing in 5G communication networks. He has been invited to become a Reviewer of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC), IEEE ACCESS, *ETRI Journal*, and *KSII Transactions on Internet and Information Systems*.

HAO LU received the Ph.D. degree in instrument science and technology from Army Engineering University, Shijiazhuang, China, in 2017. He is currently an Engineer with the Beijing Aerospace Control Center, Beijing, China. His current research interest includes software defined space-terrestrial integrated networks.

XI LI received the Ph.D. degree in software engineering from Army Engineering University, Shijiazhuang, China, in 2017, where he is currently a Lecturer with the Information and Communication Engineering Department. His current research interest includes the optimization of network function virtualization technology in 5G.



DONGHAO ZHAO received the M.S. degree in communication and information system from Army Engineering University, Shijiazhuang, China, where he is currently a Lecturer with the Information and Communication Engineering Department. His current research interest includes SDN/NFV/AI, service function chain resource allocation (SFC-RA) in 5G with AI.