Service Function Chaining Embedding in Hybrid Optical-Electronic Networks

Lin, Rongping; Ding, Bencheng; Luo, Shan; Zukerman, Moshe

Published in:
Journal of Lightwave Technology

Published: 01/08/2022

Document Version:
Post-print, also known as Accepted Author Manuscript, Peer-reviewed or Author Final version

Published version (DOI):
10.1109/JLT.2022.3176473

Publication details:
Lin, R., Ding, B., Luo, S., & Zukerman, M. (2022). Service Function Chaining Embedding in Hybrid Optical-Electronic Networks. *Journal of Lightwave Technology*, *40*(15), 4922-4933. https://doi.org/10.1109/JLT.2022.3176473

Service Function Chaining Embedding in Hybrid Optical-Electronic Networks

Rongping Lin, Bencheng Ding, Shan Luo, and Moshe Zukerman, *Life Fellow*, *IEEE*

***Abstract*—Network function virtualization (NFV) provides cost-effective network service by means of service function chain (SFC) for users. Meanwhile, billions of terminals are connected to the Internet (Internet of Things), and the vast variability of bandwidth requirements of SFC requests from heterogeneous terminals poses challenges to the network management, then neither only electronic networks nor only optical networks can provide highly efficient transmissions for network services. In this paper, a hybrid optical-electronic network paradigm is considered to provide connection services for SFC requests to benefit from the large bandwidth of optical networks and the flexibility of electronic networks. A profit-based utility function is established, and a mathematical formulation of the SFC embedding problem with the log-sum-exp approximation is provided. An approach based on Markov chain modeling is used to adaptively embed SFC requests with the aim to maximize the utility function. Simulation results demonstrate the quality and adaptivity of the proposed approach in terms of the utility function.**

***Index Terms*—Service function chaining, hybrid optical-electronic network, Markov chain**

I. Introduction

The application of network function virtualization (NFV) enables network services to be provided by means of service function chaining (SFC) for users. SFC realizes network services by processing data through virtual network functions (VNFs) in a predetermined order [1]. In NFV networks, the SFC requests are embedded into the underlying network where VNFs are embedded in servers at network nodes, and the data transmissions between VNFs are supported by the underlying network. Meanwhile, the embedding of SFC requests affects the resource allocations of the network and leads to different network states, which decide the efficiency of the network [2]. As a significant number of terminals are connected to the Internet, these terminals are heterogeneous and host various applications [3]. With the significant increase of applications, various types of network services can be efficiently provided by the NFV technology that may instantiate new VNFs for specific applications. However, the underlying network faces challenges to provide flexible and efficient transmission paths among VNFs.

This work was supported by a grant from the National Natural Science Foundation of China (NSFC) (61871097), and by a grant from the City University of Hong Kong, Hong Kong SAR, China (9610544).

R. Lin and B. Ding are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), China (e-mail: linrp@uestc.edu.cn, BenChengDing@163.com).

S. Luo is with the School of Aeronautics and Astronautics, UESTC, China (e-mail: luoshan@uestc.edu.cn).

M. Zukerman is with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China (e-mail: m.zu@cityu.edu.hk).

Specifically, optical networks provide large bandwidths but have efficiency problems in providing fine-granularity connections. Electronic networks provide fine-granularity connections but they face bandwidth capacity challenges. Therefore, neither optical networks on their own, nor only electronic networks can efficiently support data transmissions for various network services.

Most of the existing work on SFC embedding only considered either optical networks or electronic networks as the underlying physical networks for SFC embedding, which need to be extended to consider hybrid optical-electronic networks (both fiber links and electronic links coexist in a network) for better system performance. Meanwhile, for edge computing networks that have computational abilities at network edges [4], some edge devices, like base stations, are connected to each other with optical links, which makes the network a hybrid optical-electronic network. Then, this study on the SFC embedding in hybrid optical-electronic networks can also facilitate the deployment of the NFV in such edge computing networks. The key novelty of this work is to consider the hybrid optical-electronic network in the context of SFC embedding.

Considering the large bandwidth capacity of optical networks and the flexibility of electronic networks, in this paper, we investigate the SFC embedding problem in hybrid optical-electronic networks that provide an efficient transmission platform for network services. A profit-based utility function is established on the scenarios of SFC requests that have different bandwidth requirements. Because the SFC embedding problem is a combinatorial optimization problem and has been proved to be NP-hard [5], approximation methods are applied to solve the problem. In the paper, the log-sum-exp approximation method is used to approximate the SFC embedding optimization problem with a maximization profit-based utility function. Then, a time-reversible Markov chain model is considered for the network system and an algorithm is also provided based on the Markov chain model to solve the approximate problem. It is noted that instead of propose an algoirthm that efficiently embeds SFC requests in a network as the existing work in SFC embedding, this paper will provide a method that can actively vary the network state (i.e. resource provision) to the SFC requests for optimal network performance. Under the Markov chain modeling, a network state is defined as a state of the Markov chain, and the change of network state is caused by VNF on/off switching in the network and/or optical paths set-up/take-down in the optical layer of the network. Then, the network converges to the Markov chain state that corresponds to the optimal utility function value, which achieves that optical and electronic paths adaptively support the data transmission among VNFs in hybrid optical-electronic networks.

The remainder of this paper is organized as follows. In Section II, we briefly review the related work. In Section III, the mathematical model of the SFC embedding problem in hybrid optical-electronic networks is provided by applying the $log\text{-}sum\text{-}exp$ approximation. Simulation results are provided in Section IV. Finally, the paper is concluded in Section V.

II. Related Work

The SFC embedding is a challenging problem in the implementation of NFV networks. In particular, the key challenge is how to embed the VNFs of the SFC to the underlying physical network, which allocates the required resources in a cost-effective way and provides paths between adjacent VNFs. Publications on the SFC embedding problem in electronic networks include the following. Chen et al. [6] investigated the SFC embedding problem in edge computing networks where user terminals are connected to edge devices and served by VNFs provided by the edge devices. In that work, the maximum energy consumption was minimized considering and without considering the energy constraints of physical devices. Zheng et al. [7] studied the SFC embedding problem with different contents in forward and backward traffic from customer to edge server/cloud and from edge server/cloud to the customer. respectively, in edge computing networks, and an algorithm was proposed to embed two different SFCs for the forward and backward traffic, called a hybrid SFC. Bhamare et al. [8] investigated the SFC embedding problem in multi-domain networks, and the data traffic crossing domains were reduced to achieve the minimal embedding cost while satisfying the end-to-end delay requirements. Jia et al. [9] investigated the SFC embedding problem to minimize the embedding cost in the scenario where multiple data centers were considered, and an online algorithm was provided to divide the original problem into multiple sub-problems. Guo et al. [10] investigated the SFC orchestration problem in edge computing networks considering the trust issue in private and public networks resource sharing, and the adaptation issue in resource allocation. Blockchain and deep reinforcement learning methods were applied to construct a trusted resource sharing and dynamic hierarchical SFC orchestration algorithm. Fan et al. [11] investigated an online SFC embedding problem with service reliability, which can minimize resource usages while satisfying reliability requirements. Thanh et al. [12] investigated the resource and energy efficiencies in the SFC embedding problem in edge computing networks, and an SFC embedding strategy in the edge-cloud environment was proposed. Also, a traffic monitoring IP camera system with a service function chain was deployed in a testbed. However, all these publications are limited to electronic networks, and if the bandwidth requirements of the SFC requests are large, electronic networks may not be able to meet the demand.

There are also several publications on the SFC embedding problem in the context of optical networks. Fang et al. [13] investigated the SFC embedding problem in flexi-grid optical networks, and a heuristic algorithm was proposed to share VNFs with other SFC requests, which also achieved the load balance and optimized bandwidth resource usages. Lin et al. [14] investigated the VNF placement problem in optical networks. An integer linear programming formulation was provided for the problem and a game-theory model was proposed to realize efficient resource allocation in VNF embedding and routing to achieve high utilization of network resources. Zeng et al. [15] investigated the multicast NFV embedding problem to minimize the embedding cost in flexi-grid optical networks, and an algorithm was provided to solve VNF placement, multicast routing, and grid assignment simultaneously. Bi et al. [16] studied the SFC embedding problem in multi-layer networks where the bottom layer was an optical layer that provides optical paths for the upper layer. In particular, they considered a multi-objective optimization for the SFC embedding that was solved by deep reinforcement learning. Troia et al. [17] also investigated the SFC embedding problem in multi-layer networks where the bottom layer was a flexi-grid optical layer. To decrease the blocking probability, reinforcement learning was applied in [17] to perform reconfiguration and dynamic resources allocation based on the current state of the network and historical traffic load. A key difference between publications [16], [17] and the present paper is that in these two publications only the optical layer provides bandwidth resources for connections, while in the present paper both optical and electronic links coexist in the network and provide bandwidth resources for connections with different granularities. All these publications are based on optical networks, and if bandwidth requirements of SFC requests are small, optical networks cannot efficiently provide flexible and adequate connections for requests. The related publications presented above focus on the SFC embedding with homogeneous bandwidth requirements, and the underlayer physical networks can provide efficient transmission for the upper layer connections. However, if connection requirements of SFC requests have large variances (from kbps to Gbps), neither only electronic network nor only optical network can provide efficient transmissions (the former cannot provide large bandwidth and efficient switching for large bandwidth requirements, and the latter cannot provide flexible and fine granularity connections for small bandwidth requirements).

In this paper, considering heterogeneities of bandwidth requirements of SFC requests, we investigate the SFC embedding problem in hybrid optical-electronic networks. Because of the complexity of the problem, the $log\text{-}sum\text{-}exp$ approximation is applied to solve this problem, and an algorithm based on Markov chain modeling is proposed to realize the adaptive support of SFC requests with optical and electronic paths.

III. Problem Formulation

In this section, we first illustrate the SFC embedding problem in the context of a hybrid optical-electronic network and then propose a utility (objective) function for the problem. The problem is a combinatorial optimization problem, in which the solution consists of VNF placement and traffic routing of the SFC embedding. All notations used in this paper are summarized in Table I.

TABLE I
SUMMARY OF USED NOTATIONS

Notation	Description
N	Set of network nodes
L	Set of links that contains optical and electronic links
$G(N,L)$	The hybrid optical-electronic network
L_o	Set of optical links, $L_o \subseteq L$
L_e	Set of electronic links, $L_e \subseteq L$
W	Set of wavelengths per optical link
R	Set of SFC requests
L_r	Set of virtual links in SFC r, $r \in R$
F_r	Set of VNFs in SFC r
$d_{r,f}$	Computational requirement of VNF f in SFC r
$d_{r,ij}$	Bandwidth requirement of virtual link ij in SFC r
F_s	Set of VNFs at network node s
C_s	Computational capacity of network node s
C_f	Computational resource for instantiating a VNF f
$M_{r,f}^s$	VNF f of SFC r embeds in node s or not
$M_{r,ij}^{mn}$	Link ij of SFC r embeds in electronic link mn or not
$M_{r,ij}^{mn,w}$	Link ij embeds from m to n with wavelength w or not
u_f	Revenue per computation unit of VNF f
P_h^s	Energy cost per computation utilization unit of node s
P_b^s	Energy cost of idle power of node s
P_b^e	Energy cost of idle power of link e
P_h^e	Energy cost per bandwidth utilization unit of link e
P^o	Energy cost of a wavelength in an optical link

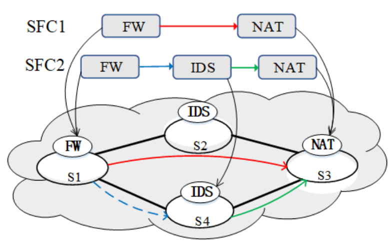

Fig. 1. SFC embedding in a hybrid optical-electronic network.

According to the hybrid optical-electronic network paradigm, the SFC traffic is processed by a VNF and then transmitted to the next VNF for other processing in an SFC. Both electronic and optical paths provide transmission service for the SFC traffic. For example, in Fig. 1, there are two SFCs embedded in the hybrid optical-electronic network. The firewall (FW) of SFC1 is embedded in network node S1 and the network address translation (NAT) is embedded in network node S3, and the FW, intrusion detection system (IDS), and NAT of SFC2 are embedded in network nodes S1, S4, and S3, respectively. Meanwhile, the virtual links between the VNFs are embedded in electronic and optical paths in the network. Specifically, because the data transmission between FW and NAT in SFC1 requires a large bandwidth, an optical path that has the route S1→S2→S3 is applied to embed the virtual link. On the other hand, two virtual links of SFC2 are embedded in electronic paths S1→S4 and S4→S3, which provide a fine granularity for the data transmissions. Then, the bandwidths or wavelengths along the paths are allocated for the virtual links. In this example, the path that embeds the virtual link of SFC1 is an optical path that traverses two optical links, optical link from S1 to S2, and optical link from S2 to S3, and this path provides the all-optical end-to-end data transmission in the optical domain. It is noted that optical paths can provide very high bandwidths and efficient data transmission for SFC traffic that is bandwidth-hungry. However, an optical path must be set up first before being used, and only a limited number of optical paths can be set up in the network due to the lack of resources. While on the other hand, the electronic paths have a limited bandwidth capacity and fine granularity for data transmissions, and the electronic path can be used hop by hop from the source to the end if there are enough available bandwidths.

A. Utility Function

In this problem, the utility function (U) representing the profit of the network is maximized. We consider the revenue (U_n) of the network to accommodate SFC requests, and the costs of energy consumptions from nodes (mostly from servers at nodes), electronic and optical links (Z_n, Z_e and Z_o). Then, the utility function of the problem is

$$U = U_n - Z_n - Z_e - Z_o. \tag{1}$$

From the perspective of a user, to obtain a network service, the user is concerned about the computation required for the network service, then in the U_n calculation, we assume that the revenue of the network is proportional to the computational requirements of the VNFs (charge more if more computational resources are used). Then the revenue of the network U_n is

$$U_n = \sum_{s\in N}\sum_{r\in R}\sum_{f\in F_r} u_f M_{r,f}^s d_{r,f}, \tag{2}$$

where N is the set of network nodes, R is the set of SFCs, F_r is the set of VNFs in SFC r, u_f is the revenue per computation unit, $M_{r,f}^s$ indicates the VNF f of the SFC request r is embedded in node s or not, and $d_{r,f}$ is the computational requirement of VNF f in the SFC r.

There are energy consumptions of network nodes and links, which are related to the resource utilizations. To calculate energy cost of network nodes Z_n, the utilization of computational resource on node s is

$$\rho_s = \frac{\sum_{r\in R}\sum_{f\in F_r} M_{r,f}^s d_{r,f}}{C_s}, \tag{3}$$

where C_s is the computational capacity of node s, Then, the energy cost of network nodes is as follows,

$$Z_n = \sum_{s\in N}\rho_s P_h^s + \sum_{s\in N} P_b^s Min\left(\sum_{r\in R}\sum_{f\in F_r} M_{r,f}^s, 1\right), \tag{4}$$

where P_h^s is the energy cost per computation utilization unit, P_b^s is the energy cost of idle power that is the basic power if the network node (server at the node) is turned on. If $Min\left(\sum_{r\in R}\sum_{f\in F_r} M_{r,f}^s, 1\right)$ is equal to 1, it means there are VNFs have been embedded into node s, which requires the turning-on state of the node and causes the idle power of the node. Similarly, the utilization of electronic link ρ_{mn} is

$$\rho_{mn} = \frac{\sum_{r\in R}\sum_{ij\in L_r} M_{r,ij}^{mn} d_{r,ij}}{C_{mn}}, \tag{5}$$

where L_r is the set of virtual links of SFC r, $M_{r,ij}^{mn}$ indicates virtual link ij of SFC r is embedded in electronic link mn or not, and $d_{r,ij}$ is the bandwidth requirement of virtual link ij. Then the energy cost of electronic links Z_e is

$$Z_e = \sum_{mn\in L_e} \rho_{mn} P_h^e + \sum_{mn\in L_e} P_b^e Min\bigg(\sum_{r\in R}\sum_{ij\in L_r} M_{r,ij}^{mn}, 1\bigg), \tag{6}$$

where L_e is the set of electronic links, P_h^e is energy cost per bandwidth utilization unit, P_b^e is the energy cost of idle power that is the basic power if the electronic link is turned on. When the value of $Min\left(\sum_{r\in R}\sum_{ij\in L_r} M_{r,ij}^{mn}, 1\right)$ is equal to 1, it means there are virtual links have been embedded into electronic link mn, which requires the turning-on state of the electronic link and causes the idle power of the link.

In the hybrid network, an optical link contains multiple wavelengths and each wavelength has a large bandwidth capacity, then a wavelength can be shared by multiple low-rate transmission connections. Meanwhile, an all-optical path consists of multiple optical links in sequential order, which use the same wavelength to satisfy the wavelength continuity constraint, and this path provides an all-optical transmission from source to end. Then the transmission of each optical link with a wavelength consumes energy, and the total energy cost of optical links Z_o is

$$Z_o = \sum_{mn\in L_o}\sum_{w\in W} Min\bigg(\sum_{r\in R}\sum_{ij\in L_r} M_{r,ij}^{mn,w}, 1\bigg) P^o, \tag{7}$$

where L_o is the set of optical links, W is the set of wavelengths in an optical link, $M_{r,ij}^{mn,w}$ indicates virtual link ij of SFC r is embedded in optical link mn with wavelength w or not, and P^o is the energy cost of a wavelength in an optical link.

In this paper, given a set of SFC requests, the utility function value is decided by a combination of: 1) the network resource provisioning (servers or VNFs, electronic links, and optical paths) that incurs energy cost; and 2) the SFC embedding algorithm that decides the revenue by accommodating the SFC requests and the resource allocation. We formulate the optimization problem of maximizing the utility function and provides the optimal solution that includes the network resource provisioning and SFC embedding for the problem. Specifically, the optimization problem is to maximize the utility function, with constraints that defines the feasible region. Without loss of generality, the feasible region is the set of all the feasible configurations CF, where a configuration $cf \in CF$ comprises two parts: the resource provisioning of the network, i.e. servers or VNFs, electronic links, and optical paths; and the feasible SFC embedding solution based on this resource provisioning. Then, different resource provisioning, e.g. open/close a VNF or set-up/take-down an optical path, and/or different SFC embedding make different network configurations lead to different utility function values. Note that given the same resource provisioning, if different SFC embedding algorithms are applied, the embedding solutions and resource usages, and even acceptance ratios may be different. We use a new notation U_{cf} to replace U, which denotes the utility function value for configuration cf as follows

$$\begin{aligned} \max \quad & U_{cf} = U_n - Z_n - Z_e - Z_o \\ s.t. \quad & cf \in CF. \end{aligned} \tag{8}$$

This optimization problem is to find the optimal configuration among the set of feasible configurations CF, which also means to optimize the resource provisioning and SFC embedding in the hybrid optical-electronic network while maximizing the utility function. This optimization problem is more challenging than the general SFC embedding problem because this problem considers resource provisioning and SFC embedding, while in the general SFC embedding problem, the network resources are already known and it does not consider optimizing the resource provisioning. As the simpler SFC embedding problem is known to be NP-hard, then the new problem we consider must also be NP-hard and is intractable when the problem size is large. To address this scalability problem, we will provide an approximation algorithm to efficiently solve this problem.

B. *Log-sum-exp Approximation for the Problem*

In this paper, we apply the $log\text{-}sum\text{-}exp$ method to approximately solve the optimization problem (8). The $log\text{-}sum\text{-}exp$ function is applied to approximate the utility function, and the network system is modeled as a Markov chain. Then, near optimal performance can be achieved [18]. First, we know that our optimization problem (8) has the same optimal value as the following problem

$$\begin{aligned} \max \quad & \sum_{cf\in CF} p_{cf} U_{cf} \\ s.t. \quad & \sum_{cf\in CF} p_{cf} = 1, \end{aligned} \tag{9}$$

where p_{cf}, $0 \le p_{cf} \le 1$, is the percentage of time that network configuration cf is selected. Next, following [18], the problem (9) can be further approximated by the following optimization problem

$$\begin{aligned} \max \quad & \sum_{cf\in CF} p_{cf} U_{cf} - \frac{1}{\beta} \sum_{cf\in CF} p_{cf} \log p_{cf}, \\ s.t. \quad & \sum_{cf\in CF} p_{cf} = 1, \end{aligned} \tag{10}$$

where β is a positive constant. The gap between these two optimization problems (9) and (10) is bounded by $\frac{1}{\beta} log|CF|$, where $|CF|$ is the size of the set CF, and the optimal solution of problem (10) is the same as that of problem (9) when $\beta \to \infty$ [18]. By solving the Karush-Kuhn-Tucker (KKT) conditions of problem (10), the optimal solution p_{cf}^* is given as

$$p_{cf}^* = \frac{exp(\beta U_{cf})}{\sum_{cf'\in CF} exp(\beta U_{cf'})}, \tag{11}$$

which is the optimal solution of problem (10), and an approximate solution of problem (9). Then, with the $log\text{-}sum\text{-}exp$ method, the difference between the final solution and the original optimal solution can be controlled by the value of β.

C. Time-reversible Markov Chain Model

There are two challenges analyzing the SFC embedding system of problem (10). The first is the enormous number of network configurations, and the second is complicated relationships among network configurations. In this paper, a time-reversible Markov chain model is applied to approximate the relationships among network configurations that address the second challenge. Then, based on the Markov chain model, an approach can be built to operate as the system and derive the optimal utility function value, where only a limited number of network configurations are explicitly traversed, and this addresses the first challenge. In the time-reversible continuous Markov chain, the state space is the set of network configurations CF, and the stationary distribution of state cf is p^*_{cf} as (11). Each network configuration (state) is now represented by a state on the Markov chain, and later we use them interchangeably. This Markov chain approximately models the original system and significantly simplifies the operation of the system. Then, the network configuration will be operated following the time proportion p^*_{cf}, $cf \in CF$, for a high utility function value.

To design the configuration space CF, the network resource provisioning contains VNF provision, optical path provision and electronic link provision. Let

$$Z = \left\{ z_s^f | z_s^f \in \{0,1\}, s \in N, f \in F_s \right\}$$

denote the VNF provision for a given VNFs state space at network nodes, where z_s^f indicates VNF f at node s is instantiated or not, and

$$Y = \left\{ y_{pq}^w | y_{pq}^w \in \{0,1\}, p \in N, q \in N, w \in W \right\}$$

denotes the optical path provision for a given optical path state space in the network, where y_{pq}^w indicates whether the optical path from node p to node q with wavelength w is setup or not. We note that all the electronic links of the network are ready to be used by default, then the network resource provisioning does not explicitly indicate the electronic link provision with states of electronic links in the network configuration. Then, we define $CF = \{Z, Y, Alg\}$ as the configuration space, which contains the network resource provisioning and the SFC embedding algorithm Alg that embeds SFC requests into the physical network. Usually, in a specific system, an SFC embedding algorithm Alg is provided for the system operation and usually does not change, then the configuration space is decided by the network resource provisioning. On the other hand, the SFC embedding algorithm decides the accommodation of SFC requests and resource allocations, which affects the utility function value, then a specific configuration $cf \in CF$ generates a utility function value U_{cf}.

With the Markov chain assumption, we design state transitions representing changes of network resource provisioning $\{Z, Y\}$ that contains instantiating/closing a VNF or setting up/closing an optical path. $q_{cf,cf'}$ is defined as the transition rate from state cf to $cf^{'}$ for two adjacent states, which satisfies the local balance equation of the Markov chain as follows

$$p^*_{cf} q_{cf,cf'} = p^*_{cf'} q_{cf',cf} \quad \forall cf, cf^{'} \in CF. \tag{12}$$

We also assume that two states cf and $cf^{'}$ that satisfy the condition $|cf \cup cf^{'}| - |cf \cap cf^{'}| = 2$ have nonzero transition rate, which means two states have the same resource provisioning except only one resource element (a VNF or an optical path) is different. Otherwise, transition rate is zero. With this assumption, the state transition is only caused by a VNF instantiating/closing or an optical path setting up/closing. This assumption significantly reduce the complexity of the Markov chain and system operation because the number of adjacent states of a state is significantly reduced. Following [18], the transition rates are designed as follows

$$\begin{aligned} q_{cf,cf'} &= exp(\frac{1}{2}\beta(U_{cf'} - U_{cf}) - \tau), \\ q_{cf',cf} &= exp(\frac{1}{2}\beta(U_{cf} - U_{cf'}) - \tau), \end{aligned} \tag{13}$$

where τ is a non-negative constant that guarantees that the transition rate never has infeasible value. With this transition rate setting, the transition rates to enter a network state that has a large utility function value have high values.

According (11) and (13), the time percentage of network configuration and transition rate between network configurations are decided by the utility values of network configurations. As discussed above, a network configuration contains network resource provisioning and SFC embedding. In this Markov chain design, a specific SFC embedding algorithm and the corresponding network resource provisioning are considered together in network configurations. In this way, the *method* of designing the Markov chain is the same for any SFC embedding algorithm. However, different SFC embedding algorithms result in different Markov chains because different utility function values may be obtained from the same network resource provisioning. Finally, the optimal utility value is independent of the particular SFC embedding algorithm used. This is because any SFC embedding algorithm has a proper network resource provisioning by the Markov chain for large utility values. Then, the network system can obtain the optimal performance independent of SFC embedding algorithms.

IV. MARKOV CHAIN BASED APPROACH

The system has been approximated as the Markov chain to simplify the relationships among network configurations. However, the system complexity due to the enormous number of network configurations hinders the direct calculation of the optimal utility function value from (10). In this section, we present an approach that is based on the proposed Markov chain model of the SFC embedding system. This approach works as the Markov chain, and the network configurations are operated following the relationships shown in (11) to (13).

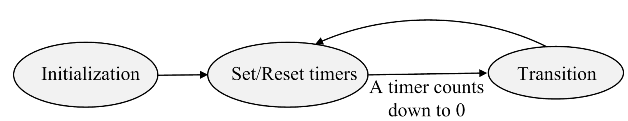

Fig. 2. State machine of the approach.

The state machine of the approach is shown in Fig. 2, where the change of network configuration happens in Transition, and the specific network configuration transited to is controlled by the timers in the old network configuration. With the assumption that two adjacent states in the chain have the same resource provisioning except only one resource element (a VNF or an optical path) is different, the state of each resource element in the existing network configuration may be changed and the system transits to the new corresponding network configuration. Since the number of resource elements in a network configuration may be large, we provide a timer for each network node instead of a timer for each VNF in the node, and the choice of VNF is decided randomly according to a given probability function (discuss next). Similarly, only one timer is used for all the optical paths instead of a timer for each optical path, and the choice of each one of the optical paths is decided based on a probability function (discussed next).

See Algorithm 1 for the details of the approach. In Algorithm 1, to initialize the network system, a set of network nodes is randomly selected and a set of VNFs on those nodes is randomly instantiated, which provides the initial network resource provisioning. Then, SFCs are embedded with the initial network resource provisioning, and this generates an initial network configuration cf. Next, one timer is started for each network node, and only one timer is started for all the optical paths. When any timer counts down to zero, the network configuration is transited to a new configuration that has a new resource provisioning and SFC embedding. After the transition, all times are reset, and the network system continues. Specifically, in Algorithm 1, the While loop contains all the operations in the Set/Reset timers and the Transitions states. When any timer hits zero, the algorithm goes to the Transition as shown in Fig. 2. In Transition processing, the state of the VNF or the state of the optical path is changed. Specifically, the state is changed means instantiating the VNF if there was no such VNF running in the node or closing this VNF otherwise, or setting up the optical path if there was no such optical path in the network or closing it otherwise. After that, SFC requests are re-embedded with the new network resource provisioning. New timers are reset based on the new network resource provisioning.

Algorithm 1: Markov chain based approach

```
Randomly select nodes from N to form a set N_1 ⊆ N;
for s ∈ N_1 do
    Randomly select a VNF in F_s;
    Instantiate this VNF in node s;
Run Algorithm 2 to embed SFC requests;
Run Algorithm 3 to set timers for nodes;
Run Algorithm 4 to set a timer for optical paths;
while System is running do
    if The timer of node s hits zero then
        Select VNF f ∈ F_s in node s with probability
          p_{s,f} and change the state of the VNF;
    if The timer of optical paths reaches zero then
        Select an optical path from m ∈ N to n ∈ N
          and wavelength w ∈ W with probability
          p_{mn,w} and change the state of the optical
          path;
    Run Algorithm 2 to re-embed SFC requests;
    Run Algorithm 3 to set timers for nodes;
    Run Algorithm 4 to set a timer for optical paths;
```

A. SFC Embedding

Given the network provisioning, a set of SFC requests is embedded and the resources are allocated, then the corresponding utility function value is obtained. Algorithm 2 shows the details of the embedding procedure for the SFC requests. Considering the objective function is to maximize the utility value, the given SFC requests are sorted in the descending order of the total computation requirement of each SFC request. Then, the sorted requests are embedded one by one following the order, and the SFC requests with high computation requirements are embedded first, which contributes more to the utility value.

To embed an SFC request, the virtual links of the SFC are embedded one by one from the beginning of the SFC. Before embedding virtual link ij, two physical node sets N_i and N_j that can embed VNFs i and j are built, respectively, which sequentially contains three types of nodes: the embedding physical node decided by the adjacent virtual link already, the physical nodes that have the existing VNFs with enough resources, and the physical nodes can instantiate the VNF. The physical nodes in the sets are checked sequentially from the beginning to find paths for VNF and virtual link embedding. This node set design and sequential checking increase sharing of the existing VNFs. Meanwhile, the optical path between two physical nodes is checked first. If there is no such optical path that can embed the virtual link, a hybrid path with electronic links and optical paths (seems as direct links between two nodes) is derived by the shortest path algorithm (Dijkstra) to embed the virtual link. Any virtual link embedding failure leads to the blocking of the SFC request, and the revenue of this SFC request will be excluded from the total utility value.

B. Timer Design

To decide the lengths of timers at the current network configuration, we consider all the possible network configurations $cf^{'}$ that transit from the current configuration cf.

Algorithm 3 shows the details of timer settings for network nodes. With the condition of (12), for a node $s \in N$, by assuming to close an existing VNF or initiate a new VNF at node s while other VNFs and optical paths remain, and re-embedding the SFC requests, the network configuration cf is transited to a new network configuration $cf^{'}$. In the inner for loop of Algorithm 3, considering all VNFs at node s, all new network configurations that have a VNF state change at node s (denoted as CF_s) can be investigated, and the utility function values of the new network configurations are obtained. Then, with (13), the transition rates are obtained.

The length of the timer for node s is decided by the utility function values of the new network configurations CF_s that

Algorithm 2: Embed SFC requests

```
Sort SFC requests R in the descending order of
  ∑_{f∈F_r} d_{r,f}, r ∈ R;
for r ∈ R do
    result_r = ∅;
    for virtual link ij ∈ L_r do
        N_i = ∅; N_j = ∅; Flag=0;
        If VNF i has been embedded into node t,
          N_i = {t};
        N_i = {N_i, nodes with existing VNF i that
          have enough resource d_{r,i}};
        N_i = {N_i, nodes can instantiate VNF i and
          have enough resource C_i};
        If VNF j has been embedded into node t,
          N_j = {t};
        N_j = {N_j, nodes with existing VNF j that
          have enough resource d_{r,j}};
        N_j = {N_j, nodes can instantiate VNF j and
          have enough resource C_j};
        for i' ∈ N_i do
            for j' ∈ N_j do
                if an optical path P from i' to j' that
                  has enough resource d_{r,ij} exists then
                    result_r = result_r ∪ P;
                    Flag=1; Break;
                if a hybrid path P from i' to j' that has
                  enough resource d_{r,ij} is found then
                    result_r = result_r ∪ P;
                    Flag=1; Break;
            If Flag==1, Break;
        If Flag==0, Break;
        Update network resources according to P;
```

transit from the existing configuration cf. A node with high utility function values of configurations CF_s has a short timer length; otherwise, the timer is long. The timer length of node s is set to a random variable that is exponentially distributed with the mean of T_s as follows

$$T_s = \frac{1}{\sum_{cf' \in CF_s} q_{cf,cf'}}. \tag{14}$$

From this definition, T_s is small when the new network configurations are with high utility function values (large transition rate), and the timer is short, which likely makes the system transit to the new network configurations with high utility values. When a timer of a physical node counts down to zero, an exact VNF must be decided to change its state for the new network resource provisioning. Because there is only one different element between two adjacent network configurations, the new network configuration can be used to select the VNF that has the relevant state change. When the timer of node $s \in N$ counts down to zero, a VNF f at node s will be selected with probability $p_{s,f}$ that is defined as the ratio of the transition rate to the total transition rate of CF_s, which is as follows

$$p_{s,f} = \frac{q_{cf,cf'}}{\sum_{cf'' \in CF_s} q_{cf,cf''}}, \tag{15}$$

where $cf^{'}$ is the new network configuration because of the state change of VNF f at node s.

Algorithm 3: Timers at nodes

```
for s ∈ N do
    t_s = 0;
    for f ∈ F_s do
        if VNF f is running at node s then
            Assume to close a VNF f;
            Run Algorithm 2 to re-embed SFC requests;
            Calculate U_{cf'} and q_{cf,cf'};
            t_s = t_s + q_{cf,cf'};
        else if There is enough resource to instantiate
          a VNF f then
            Assume to instantiate a VNF f at node s;
            Run Algorithm 2 to re-embed SFC requests;
            Calculate U_{cf'} and q_{cf,cf'};
            t_s = t_s + q_{cf,cf'};
    Generate a random value following the exponential
      distribution with the mean of 1/t_s, and set the
      timer length of node s with this generated value;
```

Algorithm 4: Timer for optical paths

```
t = 0;
for m ∈ N do
    for n ∈ N, m ≠ n do
        for w ∈ W do
            if There is an optical path from m to n
              using wavelength w then
                Assume to close the optical path;
                Run Algorithm 2 to re-embed SFC
                  requests;
                Calculate U_{cf'} and q_{cf,cf'};
                t = t + q_{cf,cf'};
            else if An optical path from m to n using
              wavelength w can be set up then
                Assume to set up the optical path;
                Run Algorithm 2 to re-embed SFC
                  requests;
                Calculate U_{cf'} and q_{cf,cf'};
                t = t + q_{cf,cf'};
Generate a random value following the exponential
  distribution with the mean of 1/t, and set the timer
  length with this generated value;
```

Similarly, Algorithm 4 shows details of the timer setting for optical paths. All new network configurations due to optical path changes are checked in three for loops of Algorithm 4. Two nodes $m \in N$ and $n \in N$, and a wavelength $w \in W$

are selected, and check if an optical path from m and n can be set up or not. If yes, a new network configuration with a new optical path is obtained. In addition, a new network configuration can be obtained if an existing optical path is closed. Next, Algorithm 4 will do the same calculations as Algorithm 3 does, and the timer length of optical paths is set to a random variable that is exponentially distributed with the mean of T_o as follows

$$T_o = \frac{1}{\sum_{cf' \in CF_o} q_{cf,cf'}}, \tag{16}$$

where CF_o denotes all new network configurations that have an optical path state changes from the current network configuration. When the timer of optical paths counts down to zero, the optical path from node $m \in N$ to node $n \in N$ with wavelength w will be selected with probability $p_{mn,w}$ as follows

$$p_{mn,w} = \frac{q_{cf,cf'}}{\sum_{cf'' \in CF_o} q_{cf,cf''}}, \tag{17}$$

where $cf^{'}$ is the new network configuration because of the state change of optical path from m to n with wavelength w. With the probability settings of $p_{s,f}$ and $p_{mn,w}$, new network configurations that have high utility values are likely transited to, and the system will remain in these configurations with high probabilities.

Accordingly, given a set of SFC requests, our Markov chain based method is designed to derive the optimal network configuration that maximizes the utility function value. Then this method can be directly applied in static traffic scenarios with a given set of SFC requests, and can provide the optimal network resource provisioning and SFC embedding solution. In this case the method runs as an offline SFC embedding method. However, this method has the potential to also be used as an online method for SFC embedding that can dynamically adjust the network configuration according to the online state of the network. If there are changes of the network associated for example with link/node failures, network upgrades, or additional SFC requests, this Markov chain based method can be applied to adaptively adjust the transition rates of network states so that the utility function values of network states are also adaptively changed, then the network transits to new network states where the utility function value is maximized.

C. Performance Analysis

This section provides the performance analysis of the proposed approach, which also guarantees the performance of the algorithms in the approach.

Lemma 1. *A time-reversible Markov chain with the stationary distribution given as* (11) *is realized by Algorithm 1.*

Proof: For the timer length of node $s \in N$, because of the exponential distribution with the mean of T_s, the transition rate is $\frac{1}{T_s}$. In addition, when the timer of node s counts down to zero, the probability of an exact VNF selection is set to $p_{s,f}$ in (15). Then, we can get the transition rate from the current network configuration cf to an exact new network configuration $cf^{'}$ as follows

$$\begin{aligned} q_{cf,cf'} &= \frac{1}{T_s} \cdot p_{s,f} \\ &= \sum_{cf'' \in CF_s} exp(\frac{1}{2}\beta(U_{cf''} - U_{cf}) - \tau) \cdot \\ &\qquad \frac{exp(\frac{1}{2}\beta(U_{cf'} - U_{cf}) - \tau)}{\sum_{cf'' \in CF_s} exp(\frac{1}{2}\beta(U_{cf''} - U_{cf}) - \tau)} \\ &= exp(\frac{1}{2}\beta(U_{cf'} - U_{cf}) - \tau), \end{aligned} \tag{18}$$

which is the same as the transition rate designed as in (13). Similarly, the transition rate from the current network configuration cf to an exact new network configuration $cf^{'}$ because of the state changes of an optical path is

$$\begin{aligned} q_{cf,cf'} &= \frac{1}{T_o} \cdot p_{mn,w} \\ &= \sum_{cf'' \in CF_o} exp(\frac{1}{2}\beta(U_{cf''} - U_{cf}) - \tau) \cdot \\ &\qquad \frac{exp(\frac{1}{2}\beta(U_{cf'} - U_{cf}) - \tau)}{\sum_{cf'' \in CF_o} exp(\frac{1}{2}\beta(U_{cf''} - U_{cf}) - \tau)} \\ &= exp(\frac{1}{2}\beta(U_{cf'} - U_{cf}) - \tau), \end{aligned} \tag{19}$$

which is the same as the transition rate designed as in (13).

Therefore, the transition rate of Algorithm 1 satisfies the equilibrium equation of the Markov chain shown as follow

$$\begin{aligned} & p^*_{cf} q_{cf,cf'} \\ &= \frac{exp(\beta U_{cf})}{\sum_{cf' \in CF} exp(\beta U_{cf'})} \cdot exp(\frac{1}{2}\beta(U_{cf'} - U_{cf}) - \tau) \\ &= \frac{1}{\sum_{cf' \in CF} exp(\beta U_{cf'})} \cdot exp(\frac{1}{2}\beta(U_{cf'} + U_{cf}) - \tau) \\ &= \frac{exp(\beta U_{cf'})}{\sum_{cf' \in CF} exp(\beta U_{cf'})} \cdot exp(\frac{1}{2}\beta(U_{cf'} - U_{cf}) - \tau) \\ &= p^*_{cf'} q_{cf',cf}. \end{aligned} \tag{20}$$

According to theorem 1.3 and 1.4 in [19], the Markov chain realized by the Algorithm 1 is time-reversible with the stationary distribution given in (11). ■

With Lemma 1, the network system operates with the optimal performance and obtains the largest utility value. This optimal performance is independent of the SFC embedding algorithm, which means with any embedding algorithm, the optimal performance could be obtained. This is because the network resource provisioning is adaptive to any embedding algorithm (embedding solutions accordingly) during state transitions of the Markov chain, and matches embedding solutions with proper network resources to the largest utility value.

Lemma 2. *The gap of the Algorithm 1 is bounded by* $\frac{1}{\beta}log|CF|$*, where* $|CF|$ *is the size of the set* CF*.*

Proof: From [18], the optimization problem (10) has the approximation gap to the original problem which is bounded by $\frac{1}{\beta}log|CF|$. Meanwhile, follow Lemma 1, Algorithm 1 performs optimally with the optimal state probability, then the gap of the Algorithm 1 is bounded by $\frac{1}{\beta}log|CF|$, where $|CF|$ is the size of the set CF. ■

Lemma 3. *The mixing time (convergence time) of the designed Markov chain is bounded as follows*

$$t_{mix} \geq \frac{exp(\frac{1}{2}\beta(U_{max} - U_{min}) - \tau)}{2||CF_o| + \sum_{s \in N} |CF_s||} \ln \frac{1}{\frac{2}{\beta}log|CF|} \quad (21)$$

and

$$t_{mix} \leq (|CF_o| + \sum_{s \in N} |CF_s|)|CF|^2 exp(\frac{3}{2}\beta(U_{max} - U_{min}) + \tau) \cdot (\ln \frac{1}{\frac{2}{\beta}log|CF|} + \frac{1}{2}\ln|CF| + \frac{1}{2}\beta(U_{max} - U_{min})), \quad (22)$$

where U_{max} and U_{min} are the maximal and minimal of utility values, respectively.

The proof of Lemma 3 is omitted here due to space limitation and it can be referred to [18].

V. Simulation Results

In this section, we present simulation results to compare the performance of the algorithm with the USNET network that has 24 nodes and 43 links [20]. The parameter settings in the various cases are listed in Table II.

TABLE II
PARAMETER SETTINGS

Parameter	Value
Types of VNFs	5
Optical wavelength bandwidth	2000
Electronic link bandwidth	400
C_f	$100, 105, 110, 115, 120$
C_s	600
$\|F_r\|$	$unif[2, 5]$
$d_{r,f}$	$unif[10, 20]$
$d_{r,ij}$	$unif[10, 40]$
P_h^s	1
P_b^s	50
P_h^e	1
P_b^e	10
P_o	10
u_f	$8, 9, 10, 11, 12$
$\|W\|$	2
β	$1, 3, 5$
τ	0.001

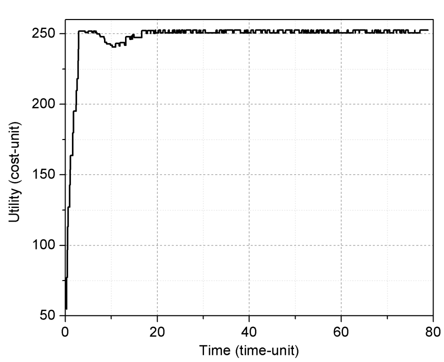

Fig. 3. Utility function value in an experiment.

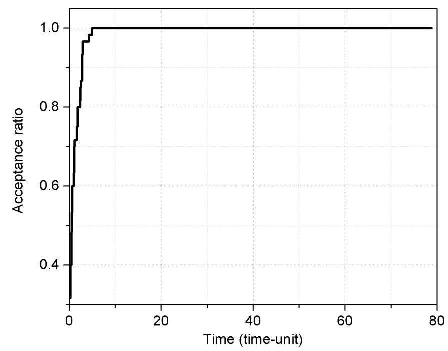

Fig. 4. Acceptance ratio in an experiment.

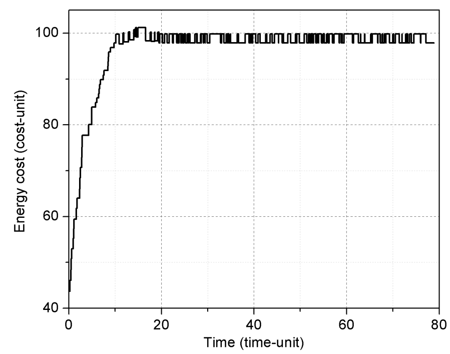

Fig. 5. Energy cost in an experiment.

Figs. 3 to 5 show the utility function value, SFC request acceptance ratio, and energy cost along with the algorithm execution with the unified time (time-unit) for an experiment in the USNET network. Fig. 3 shows the variation in the utility function value during the execution of the algorithm, and in the experiment, there are 60 SFC requests to be embedded in the network. The figure shows the changes in the utility value, and we can see that the utility value is stable at about 250 which is the maximal value in this scenario. In Fig. 3, it can be observed that before reaching stability, the utility value increases quickly. This is because the network transits to better network configurations (with large utility values) following the state traversing of the Markov chain. However, the utility value converges as the algorithm runs, and the network tries to remain in the network configuration that has the highest utility value finally. It is noted that after reaching stability, the network still changes its network configuration with possibilities, which leads to the variations of the utility value, but the network will immediately back to the optimal network configuration with a high state transition rate as shown in the figure. The sojourn time of each network configuration in the time-reversible Markov chain is decided by its utility function value. Specifically, the sojourn time tends to be long when the utility value at that network configuration is large, otherwise, the sojourn time tends to be short. Then, in practice, the network may turn to the network configuration with a low utility value, and it will return to the optimal network configuration in a short time and stay in the optimal network configuration for most of the time.

Fig. 4 shows the SFC request acceptance ratio during the execution of the algorithm. In the figure, the acceptance ratio increases and becomes stable as the algorithm executes. This is because better network configurations that can accommodate more SFC requests are applied, and the utility value in Fig. 3 verifies this explanation. When the network is stable, there is no variation of the acceptance ratio value. This indicates that even the network configuration changes with possibilities (shown in Fig. 3), the SFC requests still can be accommodated by these network configurations that are adjacent to the optimal network configuration denoted as states in the Markov chain. Fig. 5 shows the total energy cost during the execution of the algorithm. Still, the energy cost value becomes stable after tens of state transitions. After reaching stability, the energy cost value has minor variations because of the changes of the network configurations with possibilities. These energy cost changes are incurred by the working state changes of VNFs and electronic/optical links in different network configurations.

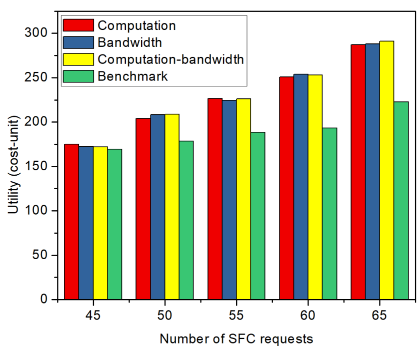

Fig. 6. Utility function values of different SFC embedding algorithms.

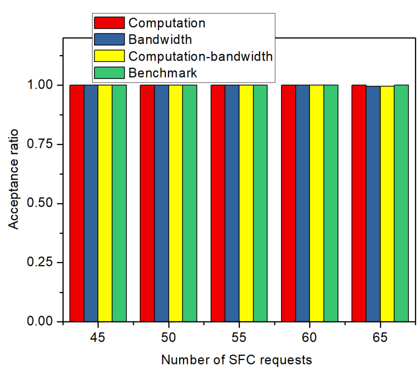

Fig. 7. Acceptance ratios of different SFC embedding algorithms.

To verify the network performance is independent of the SFC embedding algorithm with the Markov chain design, three SFC embedding algorithms are applied to compare the network performances (utility value and acceptance ratio). First, the SFC embedding algorithm shown in Algorithm 2 and named as **Computation** is to arrange the SFC requests by total VNF computational requirements in ascending order, then embed the ordered SFC requests one by one from the beginning for a high utility value. For an SFC request, the algorithm embeds the SFC request segment by segment, where a segment contains two adjacent VNFs and the virtual link between them. We consider a simple embedding method for a segment, which is to search network nodes for VNF sequentially and use the shortest path between two selected network nodes to embed the virtual link. If any resource allocation fails in the operations, this SFC request is blocked. Then, the second SFC embedding algorithm named as **Bandwidth** is the same as the Computation algorithm except that the total bandwidth requirement of virtual links is used in the ordering instead of the total computational requirement. At last, the third embedding algorithm named as **Computation-bandwidth** is still the same as the Computation algorithm except that the ratio of total bandwidth requirement to the total computational requirement is used in the ordering. In addition, we also demonstrate the performance of the three algorithms as compared to the [16]-based benchmark method. To implement the [16]-based benchmark method, the open source code of [21] available in <https://github.com/kevin031060/RL_TSP_4static> is used as it is closely related to [16]. Notice that both [21] and [16] are based on actor-critic architecture and the pointer network mechanism to solve the optimization problem. Therefore, we implement the [16]-based benchmark method based on the open source code of [21] and we also make publicly available in <https://github.com/ding0git/SFC-embedding> the resulting source code that we implemented. Our source code involves certain modifications that are needed on the method of [16] to make it applicable to our problem in this paper. First, the single objective (utility cost) is used instead of the bio-objective (latency and resource utilization). Second, the latency constraints of [16] are not considered in our problem.

In Fig. 6, the utility function values of the four algorithms (i.e. our three algorithms and the [16]-based benchmark method) increase as the number of SFC requests increases. This is because more SFC requests are accommodated and the revenue increases. In the figure, the utility value of the benchmark method is lower than that of our three methods. This implies that our methods are more economically effective than the [16]-based benchmark method in the context of the SFC embedding problem with our utility consideration. Also, in the benchmark method, the model parameters are trained beforehand. Therefore, if the network has a small change, e.g., adding or removing a node or a link, the model parameters require retraining, while our three algorithms are adaptive to network changes. The figure also shows that the three algorithms have almost the same utility values in scenarios with different numbers of SFC requests. This verifies that the network system obtains the optimal performance independent of the SFC embedding algorithm. Similarly, Fig. 7 shows that our three algorithms and the benchmark method achieve almost the same acceptance ratio which is nearly equal to 1. We notice that its value does not change as the number of SFC requests increases. This is because with the increase of the number of requests, more resources are turned on to accommodate these requests, so the acceptance ratio remains close to 1. More simulations with other SFC embedding algorithms have also been performed, the same conclusions are obtained, and these results are not shown here for brevity.

We also show the algorithm performance with different β values, which affect the accuracy of the approximation

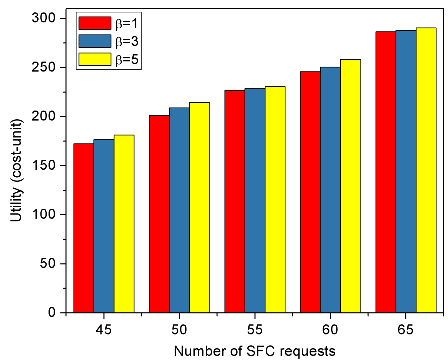

Fig. 8. Utility function values with different β.

problem (10) to the original problem according to Lemma 2. Fig. 8 shows the utility values of the algorithm with different β given different numbers of SFC requests. The figure shows that the larger β is, the larger utility value is achieved, which also leads to a higher bound of convergence time from Lemma 3. However, the balance between utility value and the convergence time is not investigated in this work.

VI. Conclusion

We have investigated the SFC embedding problem with the maximization of the utility value in hybrid optical-electronic networks. An approach based on Markov chain modeling has been proposed with the log-sum-exp approximation to find the near-optimal solution of the SFC embedding problem, and the performance is independent of the SFC embedding algorithm. In the approach, network configurations containing network resource provisioning and the SFC embedding algorithm are considered to obtain different utility values, where states of VNFs and optical paths are explicitly considered for the state transitions and network configurations with high utility values are likely to be applied by the system. Simulation results have demonstrated that the algorithm of the approach realizes the adaptive support for SFC embedding in hybrid optical-electronic networks with near-optimal utility values.

References

[1] P. Quinn and T. D. Nadeau, "Problem statement for service function chaining," Internet Engineering Task Force, RFC 7498, Apr. 2015.

[2] D. Zheng, H. Gu, W. Wei, C. Peng, and X. Cao, "Network service chaining and embedding with provable bounds," *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7140–7151, 2021.

[3] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded VNF chains in 5G core networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 692–704, 2019.

[4] R. Lin, Z. Zhou, S. Luo, Y. Xiao, X. Wang, S. Wang, and M. Zukerman, "Distributed optimization for computation offloading in edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 8179–8194, 2020.

[5] A. Fischer, J. F. Botero, M. T. Beck, H. Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, 2013.

[6] X. Chen, "Energy efficient NFV resource allocation in edge computing environment," in *Proc. International Conference on Computing, Networking and Communications*, 2020, pp. 477–481.

[7] D. Zheng, C. Peng, X. Liao, and X. Cao, "Toward optimal hybrid service function chain embedding in multiaccess edge computing," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6035–6045, 2020.

[8] D. Bhamare, R. Jain, M. Samaka, A. Erbad, L. Gupta, and H. A. Chan, "Optimal virtual network function placement and resource allocation in multi-cloud service function chaining architecture," *Computer Communications*, vol. 102, pp. 1–16, 2017.

[9] Y. Jia, C. Wu, Z. Li, F. Le, and A. Liu, "Online scaling of NFV service chains across geo-distributed datacenters," *IEEE/ACM Trans. Netw.*, vol. 26, no. 2, pp. 699–710, 2018.

[10] S. Guo, Y. Dai, S. Xu, X. Qiu, and F. Qi, "Trusted cloud-edge network resource management: DRL-driven service function chain orchestration for IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6010–6022, 2020.

[11] J. Fan, Z. Ye, C. Guan, X. Gao, K. Ren, and C. Qiao, "GREP: Guaranteeing reliability with enhanced protection in NFV," in *Proc. ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, 2015, pp. 13–18.

[12] N. H. Thanh, N. T. Kien, N. Van Hoa, T. T. Huong, F. Wamser, and T. Hossfeld, "Energy-aware service function chain embedding in edge-cloud environments for IoT applications," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13 465–13 486, 2021.

[13] W. Fang, M. Zeng, X. Liu, W. Lu, and Z. Zhu, "Joint spectrum and IT resource allocation for efficient VNF service chaining in inter-datacenter elastic optical networks," *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1539–1542, 2016.

[14] T. Lin, Z. Zhou, M. Tornatore, and B. Mukherjee, "Demand-aware network function placement," *J. Lightw. Technol.*, vol. 34, no. 11, pp. 2590–2600, 2016.

[15] M. Zeng, W. Fang, and Z. Zhu, "Orchestrating tree-type VNF forwarding graphs in inter-DC elastic optical networks," *J. Lightw. Technol.*, vol. 34, no. 14, pp. 3330–3341, 2016.

[16] Y. Bi, C. C. Meixner, M. Bunyakitanon, X. Vasilakos, R. Nejabati, and D. Simeonidou, "Multi-objective deep reinforcement learning assisted service function chains placement," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4134–4150, 2021.

[17] S. Troia, R. Alvizu, and G. Maier, "Reinforcement learning for service function chain reconfiguration in NFV-SDN metro-core optical networks," *IEEE Access*, vol. 7, pp. 167 944–167 957, 2019.

[18] M. Chen, S. C. Liew, Z. Shao, and C. Kai, "Markov approximation for combinatorial network optimization," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6301–6327, 2013.

[19] F. P. Kelly, *Reversibility and Stochastic Networks*. Cambridge University Press, 2011.

[20] R. Lin, S. Luo, J. Zhou, S. Wang, B. Chen, X. Zhang, A. Cai, W.-D. Zhong, and M. Zukerman, "Column generation algorithms for virtual network embedding in flexi-grid optical networks," *Opt. Express*, vol. 26, no. 8, pp. 10 898–10 913, 2018.

[21] K. Li, T. Zhang, and R. Wang, "Deep reinforcement learning for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3103–3114, 2021.