Robust Service Provisioning With Service Function Chain Requirements in Mobile Edge Computing

Jing Li, Weifa Liang, *Senior Member, IEEE*, and Yu Ma

***Abstract*—With the advent of Network Function Virtualization (NFV) technology, more and more mobile users make use of virtual network services in Mobile Edge Computing (MEC) networks. Each service request not only requests for a service but also a Service Function Chain (SFC) associated with the request. How to effectively allocate resources in MEC to meet the resource demands of user service requests to maximize the expected profit of the network service provider poses a great challenge. Most existing studies considered resource allocation and scheduling in MEC for user request admissions, under the assumption that the amounts of different resources demanded by each request are given *a prior* and do not change during the execution of the request. In practice, the resource demands during the implementation of a request are dynamically evolving. This uncertainty of resource demands at different execution stages of the request does impact the service quality and the profit of network service providers. Thus, providing robust services to users against their resource demand uncertainties is a critical issue. In this paper, we study the robust service function chain placement (RSFCP) problem under the uncertainty assumption of both computing resource and data rates demanded by request executions, through the placement of SFCs. We first formulate the RSFCP problem as a Quadratic Integer Programming (QIP) and show that the problem is NP-hard. We then develop a near-optimal approximation algorithm for it, by adopting the Markov approximation technique. We also analyze the proposed approximation algorithm with the optimality gap, and the bounds on the convergence time and perturbation caused by resource demand uncertainties. We finally evaluate the performance of the proposed algorithm through analytical and empirical analyses. Experimental results demonstrate that the proposed algorithm is promising, compared with existing baseline algorithms.**

***Index Terms*—Mobile edge computing networks, network function virtualization (NFV), uncertain resource demands and measurement, service function chain placement, Markov approximation technique, resource allocation and optimization, QoS-aware request admissions, performance analysis.**

I. Introduction

With the advance of mobile communication technology and mobile device fabrications, more and more mobile devices including mobile phones, tablets, and various sensors are deployed for business, entertainment, social networking, smart cities, and the Internet of Things (IoT) [16], [17]. The demanded computing and storage resources for various applications by mobile users grow exponentially [18]. Mobile Edge Computing (MEC) facilitates the offloading of computing-intensive tasks from mobile devices with low latency, by utilizing cloudlets co-located with Access Points (APs) in the proximity of mobile users [9].

Traditional network service providers provide services to their users with Network Functions (NFs) that are implemented on dedicated middleboxes [13]. However, the deployment of various middleboxes for different NFs usually incurs high instantiation expenses, complex management, and weak extensibility [37]. Network Function Virtualization (NFV) as a promising technology recently is introduced to implement Virtualized Network Functions (VNFs) as software in Virtual Machines [28], which enables network service providers to dynamically scale up or scale down their service provisioning to meet the Quality of Services (QoS) of users [19]. In reality, most mobile users not only request network services but also have specified service requirements such as Service Function Chain (SFC) and latency requirements [27], where an SFC is a sequence of VNFs that directs the data traffic flow of the request to pass through the listed VNFs in their specified order through automatic interconnection and resource allocation [9]. For each mobile user request, a certain SFC is requested to be coordinated and embedded into some physical nodes (cloudlets) in an MEC network in consideration of resource and latency constraints, this is the service function chain placement (SFCP) problem [37].

Service robustness is crucial to limit the performance degradation of network services with unpredictable changes, due to dynamic resource demands [1]. Most previous studies considered the SFCP problem under the assumption of accurate amounts of resources demanded for its VNF instance placements of each request [2], [12], [21], [30], [34], [37]–[39]. In reality, there is an uncertainty of resource demands during various stages of a request implementation [32]. In addition, there exists potential blockage during the transmission of 5G mmWave signals [11]. And the resource demands are time-varying to guarantee high-quality task offloading [40]. In this paper, we assume that the demanded computing resource and data rate of a request are different at its different execution stages, and we refer to this problem as the robust service function chain placement (RSFCP) problem, which poses the following three challenges. (1) *Optimization of SFC placements*: The requested VNF instances are first instantiated

Manuscript received December 9, 2020; revised February 16, 2021; accepted February 24, 2021. Date of publication March 5, 2021; date of current version June 10, 2021. The work of Jing Li, Weifa Liang, and Yu Ma was supported by Australian Research Council under its Discovery Project Scheme with Grant No. DP200101985. The associate editor coordinating the review of this paper and approving it for publication was B. Martini. *(Corresponding author: Weifa Liang.)*

The authors are with the Research School of Computer Science, Australian National University, Canberra, ACT 2601, Australia (e-mail: jing.li5@anu.edu.au; wliang@cs.anu.edu.au; yu.ma@anu.edu.au).

Digital Object Identifier 10.1109/TNSM.2021.3062650

into cloudlets under the cloudlet capacity constraint. Then, VNF instances will be chained in the specified order in the SFC, subject to both the link capacity constraint and the latency requirement of the request. (2) *Uncertain resource demands*: During the request execution period, the request may demand more resources than the ones initially allocated to it, then the scheduling will be infeasible, and an admitted request may never fully executed. Consequently, the network service provider will earn much less profits due to underestimating the resource demands of each admitted request. (3) *Inaccurate measurements*: As the RSFCP problem depends heavily on real-time measurements of system features in the MEC network (e.g., dynamic resource consumptions of different components such as cloudlets and links), it is difficult to obtain accurate measurements on these features in order to achieve global optimality. The actual profit collected could be perturbed from the initially expected one.

The novelty of this paper lies in the formulation of a novel RSFCP problem in MEC environments, under the assumption of both demanded resource uncertainty and measurement inaccuracy, and a near-optimal approximation algorithm with a good performance guarantee for the problem is devised.

The main contributions of this paper are as follows. We first formulate a novel RSFCP problem with the aim to maximize the expected profit collected by the network service provider of an MEC network, under the uncertainty assumption of both computing resource and data rate demands in the implementation of a user request. We show that the problem is NP-hard and provide a Quadratic Integer Programming (QIP) formulation for it. We then develop an efficient approximation algorithm for it. Specifically, we start with a special case of the problem where the measurement of the expected demanded resources for each request admission is accurate, under which we propose a near-optimal approximation algorithm by adopting the Markov approximation technique, which can achieve a provable optimality gap. We then extend the proposed approach to the problem of concern, for which we show that the proposed algorithm is still applicable, and the solution delivered has a moderate optimality gap with bounded perturbation errors on the profit measurement. We finally evaluate the performance of the proposed algorithm through experimental simulations. Experimental results demonstrate that the proposed algorithm is promising, and outperforms the baseline algorithms.

The remainder of this paper is organized as follows. Section II reviews related works. Section III introduces the system model and the problem definition. Section IV formulates a QIP solution to the problem and shows the NP-hardness of the problem. Section V proposes a near-optimal approximation algorithm for the problem. Section VI evaluates the performance of the proposed algorithm by experimental simulations, and Section VII concludes the paper.

II. Related Work

The extensive effort on the SFCP problem and its variants have been conducted in the past several years under both datacenter networks and MEC environments. Particularly, most studies of SFC request admissions were based on the fixed resource allocation mode, i.e., all resource demands are fixed prior to the execution of a request. For example, Beck and Botero [2] proposed a heuristic to coordinate the composition of SFCs and their embedding into a substrate network with the aim to minimize bandwidth utilization. Jalalitabar *et al.* [12] studied how to efficiently accommodate SFC requests while taking into account the function dependence in SFCs, the computing demand, and the bandwidth demand by SFC requests. They devised a heuristic algorithm for the construction and mapping of an SFC, by incorporating Dependence Sorting and Independent Grouping. Liu *et al.* [21] dealt with a profit maximization problem by jointly deploying the SFCs of incoming user requests and readjusting the SFC placement of accepted user requests. They developed a column generation based algorithm, considering the resource capacity constraint and the operational overhead constraint. Sun *et al.* [30] devised two heuristic algorithms for the SFC orchestration problem considering the distinct domains provisioned by different network providers. They proposed two SFC partitioning methods, and a bidding mechanism-based sub-SFC mapping solution. Tomassilli *et al.* [34] studied the problem of deploying reliable SFCs over a virtualized network function architecture. They then applied the column generation technique to deal with the decomposition model derived from the Integer Linear Programming (ILP) formulation. Zhu *et al.* [37] took both link bandwidth and node computing resource capacities into consideration. They devised an efficient online heuristic algorithm to improve the resource utilization rate by reducing resource fragmentation in physical networks. Zheng *et al.* [38] aimed to minimize the service delay considering the composition and embedding of hybrid SFCs. They proposed an Eulerian Circuit based approximation algorithm for the problem under a special case where each substrate node is assumed to provide only one unique VNF. They also applied the betweenness centrality technique to devise a heuristic algorithm for the problem under the general case. Zhang *et al.* [39] studied an SFC placement problem with the aim to minimize the total energy consumption in a telecom network. They developed efficient algorithms based on the Markov approximation technique for the problem under offline and online scenarios, respectively.

On the other hand, there are also several studies focusing on request admissions through dynamic resource allocation in MEC, under uncertain resource demands of request implementations. For example, Ali *et al.* [1] proposed a novel metric to measure the robustness of resource allocation in distributed systems against various perturbations applied to system parameters. A procedure was also described by them to guide the efficient resource allocation based on the metric. Chen *et al.* [4] devised a QoS-guaranteed SFC outsourcing algorithm based on the Hidden Markov Model to plan the outsourcing of SFCs, by predicting state sequences with the highest probability. Esposito *et al.* [7] designed an SFC instantiation prototype to deal with the random failures of processes or communication links, based on a fully distributed asynchronous consensus mechanism. Eshraghi and Liang [8] studied a task offloading problem with unknown processing

time for a three-tier cloud computing network with multi-processor access points. They developed an efficient algorithm for the problem by constructing a series of locally tight approximate geometric programming problems, which are iteratively updated. Nguyen *et al.* [24] investigated the deadline-aware SFC orchestration problem under the co-located and geo-distributed schemes, respectively, and developed approximation algorithms for SFC placement and routing with the partial knowledge of traffic demands. Psychas and Ghaderi [26] considered a job scheduling problem with random resource requirements to maximize the throughput. They proposed two obvious scheduling algorithms for the problem based on a 'Best-Fit' packing method and a 'universal partitioning' method, respectively. Wang *et al.* [35] studied a parallelized SFC placement problem with the aim to ensure efficient data transmission while reducing the end-to-end delay of an SFC. They proposed resource-efficient VNF placement methods to enhance the resiliency of a parallelized SFC via multi-flow backups. Zhang *et al.* [40] investigated a task offloading problem in a 5G small cell network considering the uncertainty of both the resource consumption and the reward of a task. They developed a multi-armed bandit based online learning algorithm, and its regret and violations are proved to be bounded sub-linearly.

Unlike the aforementioned studies that assumed the resource demands of each SFC request are given in advance [2], [12], [21], [30], [34], [37]–[39], this assumption may not be realistic as the accurate resource demands of a request implementation are not known until its completion [32]. There are a few other studies that considered the uncertainty of the other metrics such as the execution time, throughput, etc., [1], [4], [7], [8], [24], [26], [35]. The study of [40] includes the uncertainty of the resource consumption in task offloading, however, the efficient SFC placement is not taken into consideration. In this paper we study the RSFCP problem under the assumption that both computing resource and the data rate demand of each request dynamically change during its implementation. We aim to maximize the expected profit collected by the network service provider through admitting as many requests as possible. To the best of our knowledge, we are the first to study this RSFCP problem under the uncertainty of both computing resource and data rate demands of all admitted requests, by devising the very first near-optimal approximation algorithm with a provable performance gap to the problem.

III. Preliminaries

In this section, we first introduce the system model and notations. We then define the RSFCP problem precisely.

A. System Model

We consider an MEC network that consists of Access Points (APs), cloudlets, and links connecting the APs. Each cloudlet in which VNFs are deployed, is co-located with an AP, while an AP may not necessarily be co-located with any cloudlet. The MEC network can be represented by an undirected graph $G = (V, E)$, where V is the set of network nodes and E is the set of optical links. The set V is the union of the set V_C in which each network node consists of both an AP and a cloudlet, and the set V_A in which each network node consists of an AP only. For the sake of convenience, in the rest of the discussion, we only focus on computing resource consumption in each cloudlet. Each cloudlet $v \in V_C$ has computing capacity c_v. For a node $v \in V_A$ without any computing resource, its capacity is $c_v = 0$. Figure 1 illustrates an MEC network that consists of six APs and three cloudlets co-located with APs.

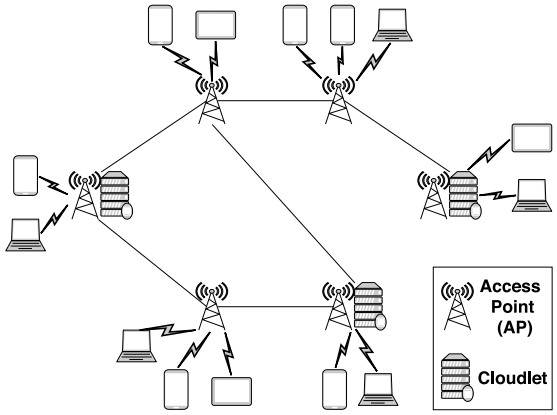


Fig. 1. Illustrative example of an MEC network consisting of six APs and three cloudlets co-located with APs.

Denote by $N^+(v)$ the set that contains all neighbors of node v in G and itself, i.e., $N^+(v) = \{u \mid (u, v) \in E\} \cup \{v\}$. Each link $(v, u) \in E$ between nodes v and u has bandwidth capacity $c(v, u)$. Denote by $l(v, u)$ the latency per unit data traffic along link $(v, u) \in E$.

Denoted by F the set of VNFs in the MEC and any SFC consists of some VNFs from F in a certain order. The VNFs are instantiated in virtual machines on cloudlets, and denote by c_f the required amount of computing resource to deploy a specific VNF of type f for all $f \in F$. Denote by l_f^v the processing time per unit data traffic of VNF f on node v.

B. Uncertainties of Demanded Computing Resource and Data Rate

Consider a set R of user requests, we assume that the demanded data rate B_i of a request $r_i \in R$ is not precisely measured, and the amount of computing resource c_f of a VNF f is various at different stages of a request implementation. In a realistic scenario, the network service provider is informed of the exact requested SFC instead of the exact resource requirements of a request (e.g., the demanded data rate B_i and the amount of computing resource c_f for a VNF of f are not known until the request is processed to its completion [32]). Although the same types of VNFs are established, they always have similar but different computing resource requirements [26]. The network system can perform some prediction mechanisms to obtain necessary information about resource demands of admitted requests and to estimate the amounts of computing resource and data rate demands by analyzing past traces [8].

We first assume that both the lower bounds and the upper bounds on the expected demanded computing resource and the data rate of each request are given in advance, which can be

obtained through analyzing the past traces [8]. Recall that B_i is the data rate of request r_i. Denote by B_i^L and B_i^U the lower and upper bounds on B_i with expectation $\overline{B_i}$, which are given in advance. Similarly, denote by c_f^L and c_f^U the lower and upper bounds on the computing resource demand c_f of a VNF instance of function f with expectation $\overline{c_f}$. However, it is difficult to accurately estimate the expected amount of demanded resources with past traces, due to the uncertainty of actual resource demands in the execution of requests. Therefore, the expected amounts of resources may experience perturbations and the analysis of such perturbations will be given later. We also introduce an adjustable control parameter of cost variation caused by the uncertain resource demands in our model to achieve a desired stable system performance, which will be shown later.

C. User Requests With Both SFC and Latency Requirements

Each request $r_i \in R$ is represented by a tuple $\langle s_i, d_i, B_i, \mathcal{L}_i, sfc_i, pay_i \rangle$, where s_i and d_i are the source and destination nodes of the data traffic of the request, respectively, B_i is the demanded data rate, $\mathcal{L}_i$ is the latency requirement, sfc_i is the service function chain, and pay_i is the payment of the request. For request r_i, its latency consists of the processing times of r_i on cloudlets and the communication latency along links in the routing path of r_i [23].

Although the data traffic of request r_i has to start from node s_i and end at node d_i, the deployed VNF instances in its SFC may be in neither of them. Meanwhile, a specific SFC may go through a node $v \in V$ without any VNF instance deployed on it. Also, the routing path of a request may do retracing multiple times. Therefore, it is complicated to characterize the traffic flow under the context of SFCs, and we herein introduce a set of dummy VNFs, denoted by $\mathbb{D}$, in a routing path construction of data traffic to ensure that when a routing path for request r_i passes through a node $v \in V$, at least one VNF or one dummy VNF is deployed on it. Therefore, the deployment of the dummy VNF helps the Quadratic Integer Programming (QIP) formulation of the problem, which will be formally defined later. And the solution by the QIP formulation serves as the exact solution to the problem. Especially, dummy VNFs g_0 and g_1 are always appended at the start and end of sfc_i. Dummy VNF g_0 is placed on node s_i while dummy VNF g_1 is place on node d_i. Therefore, it is guaranteed that the data flow for request r_i always starts from node s_i and ends at node d_i. It is worth mentioning that dummy VNFs consume neither computing resource nor processing time (i.e., for a dummy VNF $g \in \mathbb{D}$, $c_g = 0$ and $l_g^v = 0$, where c_g and l_g^v are the computing resource cost and processing time per unit data traffic of dummy VNF g on node v). Thus, for any network node $v \in V_A$ that consists of only a single AP, only dummy VNFs can be implemented on it.

Denote by S_i the extended SFC of sfc_i including both g_0 and g_1, and the length of S_i is $(|sfc_i|+2)$. Let $\Gamma = \mathbb{D} \backslash \{g_0, g_1\}$.

D. Problem Definition

Given an MEC network $G = (V, E)$, a set F of VNFs, a set R of user requests, the admission of each request $r_i \in R$ suffers uncertainties of the amounts of demanded computing resource c_f of VNF f in its SFC and demanded data rate B_i, *the Robust Service Function Chain Placement (RSFCP) problem* is to maximize the expected profit collected of the network service provider, by admitting as many requests in R as possible while meeting the latency requirement of each admitted request, subject to both computing and bandwidth resource capacities in G, where the expected profit will be defined in Section IV.

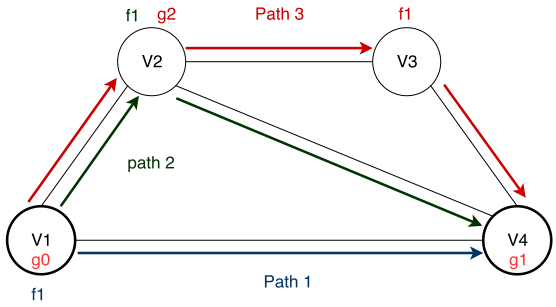


Fig. 2. Example of a routing path of $g_0 \rightarrow f_1 \rightarrow g_1$.

IV. QIP Formulation

In this section, we first formulate the RSFCP problem as a Quadratic Integer Programming (QIP), and we then show that the problem is NP-hard.

A. Traffic Flow of Requests

We deal with VNF placements of SFCs of multiple requests. For a single request $r_i \in R$, we introduce a binary decision variable x_i indicating whether request $r_i \in R$ is admitted ($x_i = 1$) or rejected ($x_i = 0$) by the system.

Recall that S_i is the extended SFC of sfc_i including both g_0 and g_1. To satisfy the network function dependence in S_i of r_i, we divide the SFC into a set of two-VNF sub-chains and then consider these sub-chains separately pair by pair. The core idea behind is to find a routing path between every two adjacent VNFs in S_i first, $\forall r_i \in R$. Then, the selected routing paths between the VNF instance pairs are concatenated to form an ordered chain. Denote by h_i^k the kth VNF of S_i, e.g., $h_i^1 = g_0$.

We introduce a binary variable $\rho_{i,k}^{v,a,u,b}$ denoting that when constructing the routing path between the kth two-VNF sub-chain $h_i^k \rightarrow h_i^{k+1}$, whether the data traffic of request r_i traverses from VNF a on node v to VNF b on node u. Thus, the routing subpaths for r_i are expressed by the values of $\{\rho_{i,k}^{v,a,u,b} \mid \forall r_i \in R, \forall k \in \{1, \ldots, |S_i| - 1\}, \forall v \in V, \forall u \in N^+(v), \forall a \in \{h_i^k\} \cup \Gamma, \forall b \in \{h_i^{k+1}\} \cup \Gamma, a \neq b\}$. For example, as shown in Figure 2, for the requested SFC, $g_0 \rightarrow f_1 \rightarrow g_1$, dummy VNFs g_0 and g_1 are first deployed on source node v_1 and destination node v_4, respectively. The SFC $g_0 \rightarrow f_1 \rightarrow g_1$ is then considered as two two-VNF sub-chains: $g_0 \rightarrow f_1$ and $f_1 \rightarrow g_1$. Once the routing paths for two-VNF sub-chains $g_0 \rightarrow f_1$ and $f_1 \rightarrow g_1$ are decided, they are concatenated together to map the SFC $g_0 \rightarrow f_1 \rightarrow g_1$ into the network. Meanwhile, several dummy VNFs in Γ may have been inserted to construct the routing path to ensure that when a routing path passes through a node $v \in V$, at least one VNF

$f \in S_i \cup \Gamma$ is deployed on it. Figure 2 shows the three routing paths.

As multiple VNFs of an SFC could be mapped to a single node, the data traffic of a request can traverse between the VNF instances in the same node multiple times. As shown in Figure 2, in *path 1*, both VNF g_0 and f_1 are placed on node v_1, which is represented by setting $\rho_{i,1}^{v_1,g_0,v_1,f_1} = 1$. Also, in Figure 2, inserting one dummy VNF g_2 on node v_2 is enough in describing *path 3*, it is not allowed to place multiple dummy VNFs on the same node for the same *k*, i.e., when considering a two-VNF sub-chain, we have

$$\sum_{a,b\in\mathbb{D},a\neq b} \rho_{i,k}^{v,a,v,b} = 0,\ \forall r_i \in R,\ \forall v \in V, \quad \forall k \in \{1,\ldots,|S_i|-1\}. \tag{1}$$

When we insert dummy VNFs, the routing path of the *k*th two-VNF sub-chain in S_i goes through at most all nodes. Thus, we let $|\Gamma| = |V|$. Dummy VNF g_0 is placed on node s_i when constructing the routing path for the first two-VNF sub-chain. While dummy VNF g_1 is placed on node d_i when constructing the routing path for the last two-VNF sub-chain. We then have

$$\sum_{u\in N^+(s_i),b\in\left\{h_i^2\right\}\cup\Gamma} \rho_{i,1}^{s_i,g_0,u,b} = 1, \tag{2}$$

$$\sum_{v\in N^+(d_i),a\in\left\{h_i^{|S_i|-1}\right\}\cup\Gamma} \rho_{i,|S_i|-1}^{v,a,d_i,g_1} = 1. \tag{3}$$

The construction of a routing path for the *k*th two-VNF sub-chain in S_i, i.e., $h_i^k \to h_i^{k+1}, \forall r_i \in R, \forall k \in \{1,\ldots,|S_i|-1\}$ is as follows.

The source and destination VNFs of the *k*th routing path are h_i^k and h_i^{k+1}. Then, the outgoing flow from h_i^k and incoming flow to h_i^{k+1} must be 1, which is represented by the following equations.

$$\sum_{v\in V,u\in N^+(v),\ b\in\{h_i^{k+1}\}\cup\Gamma} \rho_{i,k}^{v,h_i^k,u,b} = 1, \quad \forall r_i \in R, \quad \forall k \in \{1,\ldots,|S_i|-1\}. \tag{4}$$

$$\sum_{v\in V,u\in N^+(v),\ a\in\{h_i^k\}\cup\Gamma} \rho_{i,k}^{v,a,u,h_i^{k+1}} = 1, \quad \forall r_i \in R, \quad \forall k \in \{1,\ldots,|S_i|-1\}, \tag{5}$$

And for any inserted dummy VNF $g \in \Gamma$ in the *k*th two-VNF sub-chain, the incoming flow at VNF g equals the outgoing flow from it, and each dummy VNF $g \in \Gamma$ appears in the path at most once, i.e.,

$$\sum_{u\in N^+(v),\ b\in\{h_i^{k+1}\}\cup\Gamma,a\neq b} \rho_{i,k}^{v,a,u,b} - \sum_{u\in N(v),\ b\in\{h_i^k\}\cup\Gamma,a\neq b} \rho_{i,k}^{u,b,v,a} = 0, \quad \forall r_i \in R,\ \ \forall k \in \{1,\ldots,|S_i|-1\},\ \ \forall v \in V, \forall a \in \Gamma. \tag{6}$$

$$\sum_{v\in V,u\in N^+(v),\ b\in\{h_i^{k+1}\}\cup\Gamma,a\neq b} \rho_{i,k}^{v,a,u,b} \leq 1, \quad \forall r_i \in R, \quad \forall k \in \{1,\ldots,|S_i|-1\}, \quad \forall a \in \Gamma. \tag{7}$$

Let $y_{i,k}^v$ be a binary variable indicating whether VNF $h_i^k \in S_i$ is placed in cloudlet node *v*. As the first and last VNFs in S_i are dummy VNFs, we only consider the other VNFs in S_i,

$$y_{i,k}^v = \sum_{u\in N^+(v),\ b\in\{h_i^{k+1}\}\cup\Gamma} \rho_{i,k}^{v,h_i^k,u,b},\ \forall r_i \in R, \quad \forall v \in V, k \in [2,|S_i|-1]. \tag{8}$$

Recall that the upper bound of computing resource consumption of VNF h_i^k is $c_{h_i^k}^U$. To ensure that no resource capacity is violated, the computing capacity constraint on each node *v* is expressed as follows.

$$\sum_{r_i\in R,k\in\{2,\ldots,|S_i|-1\}} x_i \cdot y_{i,k}^v \cdot c_{h_i^k}^U \leq C_v, \quad \forall v \in V. \tag{9}$$

The routing path for request r_i may pass through a link multiple times. Let $z_i^{v,u}$ be an integer variable indicating the number of times link (*v*, *u*) is contained in the routing path for request r_i. Then, we have

$$z_i^{v,u} = \sum_{k\in[1,|S_i|-1],\ \forall a\in\left\{h_i^k\right\}\cup\Gamma,\ \forall b\in\left\{h_i^{k+1}\right\}\cup\Gamma} \left(\rho_{i,k}^{v,a,u,b} + \rho_{i,k}^{u,a,v,b}\right), \quad \forall r_i \in R, \forall e(v,u) \in E. \tag{10}$$

Recall that the upper bound of demanded data rate of request r_i is B_i^U. To ensure that no resource capacity is violated, the link capacity constraint on each link (*v*, *u*) for r_i can be expressed as follows.

$$\sum_{r_i\in R} x_i \cdot z_i^{v,u} \cdot B_i^U \leq c(v,u), \quad \forall e(v,u) \in E. \tag{11}$$

Note that the latency of request r_i consists of the processing times on nodes and the communication latency along links in the routing path [23]. To admit request r_i, we need to ensure that its latency requirement must be met. With the upper bound of demanded data rate B_i^U of request r_i, the latency constraint of request r_i can be expressed as follows.

$$\sum_{v\in V,\ k\in\{1,\ldots,|S_i|-1\}} x_i \cdot y_{i,k}^v \cdot B_i^U \cdot l_{h_i^k}^v + \sum_{e(v,u)\in E} x_i \cdot z_i^{v,u} \cdot B_i^U \cdot l(v,u) \leq \mathcal{L}_i, \quad \forall r_i \in R. \tag{12}$$

B. Admission Cost of Requests

Recall that the problem objective is to maximize the expected profit collected by the network service provider, which is equivalent to minimize the accumulative expected admission cost of all admitted requests, where the expected admission cost of a request is the sum of the expected resource

usage cost on both computing and bandwidth resource consumptions for implementing the request, which is defined as follows.

Considering the computing resource consumption for placing VNF instances for SFCs of requests in nodes, the computing resource usage cost at each node v, denoted by $\mathcal{E}_v$, is

$$\mathcal{E}_v = \varphi_v \cdot \sum_{r_i \in R, k \in \{2,\ldots,|S_i|-1\}} x_i \cdot y_{i,k}^v \cdot c_{h_i^k}, \tag{13}$$

where φ_v is the cost of a unit computing resource on node v with $\varphi_v > 0$. However, $\mathcal{E}_v$ is uncertain due to the uncertainty of computing resource consumption. The expected computing resource usage cost $\mathbb{E}(\mathcal{E}_v)$ of $\mathcal{E}_v$ at node v then is

$$\mathbb{E}(\mathcal{E}_v) = \varphi_v \cdot \sum_{r_i \in R, k \in \{2,\ldots,|S_i|-1\}} x_i \cdot y_{i,k}^v \cdot \overline{c_{h_i^k}}. \tag{14}$$

Considering the demanded data rate in links for the data traffic of request r_i, the bandwidth resource usage cost on link (v, u), denoted by $\mathcal{E}_{v,u}$, is

$$\mathcal{E}_{v,u} = \phi_{(v,u)} \cdot \sum_{r_i \in R} x_i \cdot z_i^{v,u} \cdot B_i, \tag{15}$$

where $\phi_{(v,u)} > 0$ is the cost of a unit bandwidth resource on link (v, u). However, the value of $\mathcal{E}_{v,u}$ is also uncertain due to the uncertainty of demanded data rates of requests. The expected bandwidth resource usage cost $\mathbb{E}(\mathcal{E}_{v,u})$ of $\mathcal{E}_{v,u}$ on link (u, v) then is

$$\mathbb{E}\big(\mathcal{E}_{v,u}\big) = \phi_{(v,u)} \cdot \sum_{r_i \in R} x_i \cdot z_i^{v,u} \cdot \overline{B_i}, \tag{16}$$

The total admission cost $\mathcal{E}$ of all admitted requests in R thus is the sum of the resource usage costs on both computing and bandwidth resource consumptions, which is defined as follows.

$$\mathcal{E} = \sum_{v \in V} \mathcal{E}_v + \sum_{e(v,u) \in E} \mathcal{E}_{v,u}. \tag{17}$$

The expectation $\mathbb{E}(\mathcal{E})$ of $\mathcal{E}$ is as follows.

$$\mathbb{E}(\mathcal{E}) = \sum_{v \in V} \mathbb{E}(\mathcal{E}_v) + \sum_{e(v,u) \in E} \mathbb{E}\big(\mathcal{E}_{v,u}\big). \tag{18}$$

The total admission cost has the uncertainty due to unknown demanded computing resource and data rates for the request implementation. In spite of this uncertainty, statistical information provided by experimental studies can be exploited to limit the risk of cost fluctuation. Reducing cost fluctuation is necessary to maintain a desired system performance for different realizations of uncertainty [8].

To model the cost fluctuation, we consider the effect of $\Delta B_i = B_i^U - B_i^L$ and $\Delta c_f = c_f^U - c_f^L$ on the cost variation of $\mathcal{E}$, denoted by $\Delta\mathcal{E}$. Since $\mathcal{E}$ is a linear function of both B_i and c_f, the relationship among $\Delta\mathcal{E}$, ΔB_i and Δc_f is given as follows.

$$\Delta\mathcal{E} = \sum_{v \in V} \varphi_v \cdot \sum_{r_i \in R, k \in \{2,\ldots,|S_i|-1\}} x_i \cdot y_{i,k}^v \cdot \Delta c_{h_i^k} + \sum_{e(v,u) \in E} \phi_{(v,u)} \cdot \sum_{r_i \in R} x_i \cdot z_i^{v,u} \cdot \Delta B_i. \tag{19}$$

C. QIP Formulation

The RSFCP problem can be formulated as a Quadratic Integer Programming (QIP) for a set R of requests with the optimization objective to

$$maximize \sum_{r_i \in R} x_i \cdot pay_i - \mathbb{E}(\mathcal{E}) - \zeta \cdot \Delta\mathcal{E}, \tag{20}$$

subject to:

$$\begin{aligned} &(1),\ (2),\ (3),\ (4),\ (5),\ (6),\ (7),\ (8),\ (9),\ (10),\\ &x_i, \rho_{i,k}^{v,a,u,b} \in \{0,1\}, \quad \forall r_i \in R,\ \forall k \in \{1,\ldots,|S_i|-1\},\\ &\forall v,u \in V, \quad \forall a,b \in S_i \cap \Gamma, \end{aligned} \tag{21}$$

where pay_i is the payment by request r_i if it is admitted and all its constraints are met, and ζ is the adjustable control parameter of cost variation compared with the expected total admission cost to stabilize the total admission cost [8].

D. NP-Hardness of the Problem

Theorem 1: The RSFCP problem in an MEC $G(V, E)$ is NP-hard.

Proof: We prove the NP-hardness of the RSFCP problem through reducing from the well-known knapsack problem.

The knapsack problem is NP-hard [25] and is defined as follows. Given a set of items $\mathcal{N}$, each item $i \in \mathcal{N}$ has a weight ω_i and a profit $p_i > 0$, $\forall i \in \mathcal{N}$. There is a bin with a capacity of $\mathcal{W}$. If item i can be placed in the bin without capacity violation, the associated profit p_i will be collected. The problem is to maximize the profit collected by packing as many items in $\mathcal{N}$ as possible, subject to the bin capacity.

Given an instance of the knapsack problem, we generate an instance of the RSFCP problem as follows.

We consider a special case of the RSFCP problem and assume that there is a single cloudlet in the MEC network with a capacity of $\mathcal{W}$, we also ignore the bandwidth and latency constraints, and we assume that there exists no resource demand uncertainty.

We then generate a set of requests R and further assume that the processing time of a VNF is negligible. For each item $i \in \mathcal{N}$, there is a corresponding request, which has the amount ω_i of computing resource consumed by the requested SFC. The expected profit p_i of request r_i is calculated by its revenue and admission cost. If a request $r_i \in R$ is admitted, the requested SFC is placed in the cloudlet and the profit p_i is collected. The RSFCP problem is to maximize the expected profit collected by the network service provider by admitting as many requests as possible, subject to computing resource capacity.

It can be seen that a solution to the RSFCP problem returns a solution to the knapsack problem. And it takes polynomial time to reduce an instance of the knapsack problem to an instance of the RSFCP problem. Due to the NP-hardness of the knapsack problem [25], the RSFCP problem is NP-hard, too. ■

For the sake of convenience, we list all symbols adopted in this paper in Table I.

TABLE I
TABLE OF SYMBOLS

Notations	Descriptions
$G = (V, E)$	an MEC network with a set V of network nodes and a set of E links
V_C and V_A	the set of nodes in which each node consists of both an AP and a cloudlet, and the set of nodes in which each node consists of an AP only
v and c_v	a node $v \in V$ and the computing capacity of node v
(v, u), $c(v, u)$ and $l(v, u)$	a link connecting node v and node u, the bandwidth capacity of $e(v, u)$, and the latency per unit data traffic along link (v, u)
$N^+(v)$	the set of nodes that contains all neighbors of node v in G and itself
F, f, c_f and l_f^v	a set of VNFs offered in the MEC network, a VNF $f \in F$, the computing resource consumption of f, and the processing time per unit data traffic of VNF f on node v
R	a set of all user requests
r_i=$\langle s_i, d_i, B_i, \mathcal{L}_i, sfc_i, pay_i\rangle$	a user request, where s_i and d_i are the source and destination nodes of the data traffic of the request, respectively, B_i is the demanded data rate, $\mathcal{L}_i$ is the latency requirement, sfc_i is the service function chain, and pay_i is the payment of the request.
g_0 and g_1	dummy VNFs g_0 and g_1 are always appended in the start and end of each requested SFC
g, $\mathbb{D}$, and Γ	a dummy VNF, the set of all dummy VNFs, and $\Gamma = \mathbb{D} \setminus \{g_0, g_1\}$
S_i and h_i^k	the extended SFC of sfc_i including both g_0 and g_1, and the kth VNF of S_i
B_i^L, B_i^U and $\overline{B_i}$	the lower bound, upper bound, and expectation of demanded data rate of request r_i
ΔB_i	$\Delta B_i = B_i^U - B_i^L$ the variation of demanded data rate of request r_i
c_f^L, c_f^U and $\overline{c_f}$	the lower bound, upper bound, and expectation of computing resource consumption of VNF f
Δc_f	$\Delta c_f = c_f^U - c_f^L$ the variation of computing resource consumption of VNF f
x_i	a binary decision variable indicating whether request r_i is admitted or rejected
$\rho_{i,k}^{v,a,u,b}$	a binary decision variable denoting that when constructing the routing path between the kth two-VNF sub-chain $h_i^k \rightarrow h_i^{k+1}$, whether the data traffic of request r_i traverses from VNF a on node v to VNF b on node u.
$y_{i,k}^v$	a binary variable indicating whether VNF $h_i^k \in S_i$ is placed in cloudlet node v
$z_i^{v,u}$	the number of times link $e(v, u)$ is contained in the routing path for request r_i
φ_v, $\mathcal{E}_v$ and $\mathbb{E}(\mathcal{E}_v)$	the cost of a unit computing resource on node v, the computing resource usage cost at node v, and the expected computing resource usage cost at node v
$\phi_{(v,u)}$, $\mathcal{E}_{v,u}$ and $\mathbb{E}(\mathcal{E}_{v,u})$	the cost of a unit bandwidth resource on link $e(v, u)$ the bandwidth resource usage cost on link $e(v, u)$, and the expected bandwidth resource usage cost on link $e(v, u)$
$\mathcal{E}$, $\mathbb{E}(\mathcal{E})$ and $\Delta\mathcal{E}$	The total admission cost of all admitted requests, the expected total admission cost of all admitted requests, and the total cost variation of all admitted requests
ζ	the adjustable control parameter of cost variation compared with the expected total admission cost to stabilize the total admission cost
η_i	a scheduling for request r_i that consists of determining the values of x_i and $\rho_{i,k}^{v,a,u,b}$ for r_i
w and W	a state (a feasible solution to the RSFCP problem), and the set of all states
Λ_w, τ and λ_w	Λ_w is the expected profit collected from all admitted requests under the state w, and τ is a large positive constant to guarantee $\lambda_w = \Lambda_w + \tau \geq 0$
p_w	a real value within the range of $[0, 1]$, which is the fraction of horizontal time (assuming the entire time horizon is 1) that the system stays in state w
p_w^*	the stationary distribution of the designed Markov chain model
$\overline{p}_w$	the stationary distribution of the Markov chain model with measurement perturbations
$\tilde{p}_w$	the time fraction distribution in the optimal solution
β and γ	both β and γ are positive constants
ψ_i	a random exponentially timer created by the thread of with the mean value γ
$q_{w,w'}$	A non-negative transition rate between two states w and w' with $w \neq w'$ $\forall w, w' \in W$.
θ_w	the perturbation error bound under state w
$\alpha(w, j)$	the probability of the perturbed λ_w taking the value $\lambda_w + (j/n_w) \cdot \theta_w$, where $j \in \{-n_w, \ldots, n_w\}$, n_w is a positive constant and $\sum_{j\in\{-n_w,\ldots,n_w\}} \alpha(w, j) = 1$.
Λ_{max}, Λ_{avg} and $\overline{\Lambda}_{avg}$	the value of the optimal solution, the expected profit with the designed Markov chain model, and the expected profit with the perturbed Markov chain model

V. MARKOV BASED APPROXIMATION ALGORITHM

As the RSFCP problem is NP-hard, in this section we devise a near-optimal approximation algorithm for the problem, by adopting the Markov approximation technique [5]. Specifically, we first introduce the log-sum-exp approximation concept, and then construct a time-reversible Markov chain with designed transition rates satisfying the detailed balance equation [10].

A. Log-Sum-Exp Approximation

Recall that there are two binary decision variables in the RSFCP problem, x_i (i.e., acceptance decision) and $\rho_{i,k}^{v,a,u,b}$

(i.e., routing path decision). We then define a scheduling η_i for request r_i that consists of determining the values of x_i and $\rho_{i,k}^{v,a,u,b}, \forall r_i \in R, \forall k \in \{1,\ldots,|S_i|-1\}, \forall v \in V, \forall u \in N^+(v), \forall a \in \{h_i^k\} \cup \Gamma, \forall b \in \{h_i^{k+1}\} \cup \Gamma, a \neq b$.

To this end, we combine all $\eta_i, \forall r_i \in R$ to form a state, denoted by *w*, following resource capacity constraints in the MEC network and the latency constraints of requests in *R*.

Denote by *W* the set of all states, i.e., the set of all feasible solutions to the RSFCP problem. Let Λ_w be the value achieved by the optimization objective (20) under a state *w* (i.e., the expected profit collected from all admitted requests under the state *w*). Because Λ_w could be negative (i.e., deficit may exist), we let $\lambda_w = \Lambda_w + \tau$, $\forall w \in W$ where τ is a large positive constant to guarantee $\lambda_w \geq 0$. The problem then is reformulated as *the Maximum Weighted Combinatorial Optimization* (MWCO) problem as follows.

$$Maximize\ \{\lambda_w | w \in W\}. \tag{22}$$

It can be seen that the optimization objective (22) has the same optimal value as the following problem.

$$Maximize \sum_{w \in W} p_w \cdot \lambda_w, \tag{23}$$

subject to

$$\sum_{w \in W} p_w = 1, \tag{24}$$

where p_w is a real value within the range of [0, 1], which is the fraction of horizontal time (assuming the entire time horizon is 1) that the system stays in state *w*.

Proposition 1 [5]: Given a positive constant $\beta > 0$ and a set of non-negative real variables $\{\gamma_1, \gamma_2, \ldots, \gamma_n\}$, we have

$$\max_{i \in [1,n]} \gamma_i \leq \frac{1}{\beta} \ln\Big(\sum_{i \in [1,n]} exp(\beta \cdot \gamma_i)\Big) \leq \max_{i \in [1,n]} \gamma_i + \frac{\ln n}{\beta}. \tag{25}$$

Following Proposition 1, when β approaches infinity, $\frac{\ln n}{\beta} = 0$, we then have

$$\max_{i \in [1,n]} \gamma_i = \lim_{\beta \to \infty} \frac{1}{\beta} \ln\left(\sum_{i \in [1,n]} exp(\beta \cdot \gamma_i)\right). \tag{26}$$

As λ_w is non-negative for any $w \in W$, we have

$$\max_{w \in W} \lambda_w \approx \frac{1}{\beta} \ln\left(\sum_{w \in W} exp(\beta \cdot \lambda_w)\right), \quad \text{for a very large } \beta. \tag{27}$$

Definition 1 [3]: Let *a* be a real vector and *b* be the dual of *a*, for $\delta : \mathbb{R}^n \mapsto \mathbb{R}$, then the conjugate function of δ, $\delta^* : \mathbb{R}^n \mapsto \mathbb{R}$ is defined in terms of the supremum as

$$\delta^*(b) = \sup\Big(b^T a - \delta(a)\Big). \tag{28}$$

Proposition 2 [3]: The conjugate function of the conjugate function of a convex function is the function itself.

Lemma 1: When β approaches infinity, the objective function (22) has the same value as the value of the following objective function.

$$Maximize \sum_{w \in W} p_w \cdot \lambda_w - \frac{1}{\beta} \cdot \sum_{w \in W} p_w \cdot \ln p_w, \tag{29}$$

$$\text{subject to} \sum_{w \in W} p_w = 1. \tag{30}$$

Proof: Denote by

$$\delta(\boldsymbol{\lambda}) = \frac{1}{\beta} \ln\left(\sum_{w \in W} exp(\beta \cdot \lambda_w)\right), \tag{31}$$

where $\boldsymbol{\lambda} = [\lambda_w \mid w \in W]$ is a vector of λ_w under each state, i.e., a vector of the expected profit collected, added by τ at each state.

Following Definition 1, the conjugate function of $\delta(\boldsymbol{\lambda})$ is

$$\delta^*(\boldsymbol{P}) = \begin{cases} \frac{1}{\beta} \cdot \sum_{w \in W} p_w \cdot \ln(p_w), & if\ \forall w \in W, p_w \geq 0, \\ & and\ \sum_{w \in W} p_w = 1. \\ \infty & otherwise, \end{cases} \tag{32}$$

where $\boldsymbol{P} = [p_w | w \in W]$ is a vector of the time fraction assigned to each state.

As log-sum-exp functions are both closed and convex, following Proposition 2, the conjugate of $\delta^*(\boldsymbol{P})$ is $\delta(\boldsymbol{\lambda})$.

Following Definition 1, we have

$$\delta(\boldsymbol{\lambda}) = \sup \sum_{w \in W} p_w \cdot \lambda_w - \frac{1}{\beta} \cdot \sum_{w \in W} p_w \cdot \ln p_w. \tag{33}$$

Combining (27), (31), and (33), the objective function (22) has the same value as the value of the objective function (29) when β approaches infinity. Lemma 1 then follows. ■

Following Lemma 1, the approximate value of the objective function of (22) can be obtained, by solving (29).

Let μ be the Lagrangian multiplier associated with Equation (30). Let p_w^* be the optimal solution to (29). It can be seen that the Karush-Kuhn-Tucker (KKT) conditions for (29) are met. We then have

$$\begin{cases} \lambda_w - \frac{1}{\beta} \cdot \ln(p_w^*) - \frac{1}{\beta} + \mu = 0, \ \forall w \in W, \\ \sum_{w \in W} p_w^* = 1, \\ \mu \geq 0. \end{cases} \tag{34}$$

The value of p_w^* thus is

$$p_w^* = \frac{exp(\beta \cdot \lambda_w)}{\sum_{w' \in W} exp(\beta \cdot \lambda_{w'})}, \ \forall w \in W. \tag{35}$$

B. Markov Chain Model Design

Let p_w^* be the stationary distribution of the designed Markov chain model that is trained to demonstrate the convergence to the specific stationary distribution set in advance, i.e., p_w^*. The authors in [5] showed that there exists at least one time-reversible Markov chain model in which the stationary distribution is p_w^*. Let $q_{w,w'}$ be a non-negative transition rate between two states *w* and w' with $w \neq w'$ $\forall w, w' \in W$. To

construct a time-reversible Markov chain [5], the following two conditions must be met: (1) each state could be transited to any other state; and (2) the detailed balance equation between states w and w' should be satisfied.

$$p^*_w \cdot q_{w,w'} = p^*_{w'} \cdot q_{w',w}, \quad \forall w, w' \in W, w \neq w'. \quad (36)$$

We first initialize the transition rates among all states as 0s. Considering (35) and (36), we then design the transition rate $q_{w,w'}$ $\forall w, w' \in W, w \neq w'$ as follows.

$$\begin{cases} q_{w,w'} = \dfrac{exp(\beta \cdot \lambda_{w'})}{|R| \cdot \gamma \cdot \max\{exp(\beta \cdot \lambda_w), exp(\beta \cdot \lambda_{w'})\}}, \\ q_{w',w} = \dfrac{exp(\beta \cdot \lambda_w)}{|R| \cdot \gamma \cdot \max\{exp(\beta \cdot \lambda_w), exp(\beta \cdot \lambda_{w'})\}}, \end{cases} \quad (37)$$

where R is the set of user requests and γ is a positive constant which is set as the mean time to transit from current state to another state.

C. Markov Based Approximation Algorithm

The implementation details of the Markov based approximation algorithm are given in Algorithm 1. In particular, the system controller creates a thread for each request in parallel, i.e., each thread deals with one request with the aim of the expected profit optimization. Initially, the dedicated thread for each request r_i randomly makes the acceptance decision (i.e., x_i) and the routing path decision (i.e., $\rho^{v,a,u,b}_{i,k}$) to choose a feasible scheduling for itself. The system controller finally combines the scheduling of each request to form the current state w, and calculates the expected profit Λ_w in state w. Recall that $\lambda_w = \Lambda_w + \tau$, where τ is a large positive constant to guarantee $\lambda_w \geq 0$.

For each request $r_i \in R$, the associated thread randomly looks for a new feasible scheduling for itself based on current residual resources in the network to form the next state w' in a parallel manner, where w' is the next state from the current state w through one transition. λ_w and $\lambda_{w'}$ under w and w' then are calculated. Then a random exponentially timer ψ_i for request $r_i \in R$ is triggered with a mean γ, where γ is a positive constant. And the timer ψ_i starts to count down. Let $p_{w,w'}$ be the probability that the system transits from state w to state w'. When the timer ψ_i for request $r_i \in R$ expires, the thread of r_i transits to state w' with probability of $p_{w,w'} = \frac{exp(\beta \cdot \lambda_{w'})}{\max\{exp(\beta \cdot \lambda_w), exp(\beta \cdot \lambda_{w'})\}}$, and broadcasts a Reset signal to the rest threads. For the rest threads, upon receiving the Reset signal, their timers are terminated. Then each thread of requests looks for a new feasible scheduling for itself in parallel based on residual resources in the network and the above steps are repeated until it converges.

D. Algorithm Analysis

In the following, we analyze the performance of the proposed algorithm. We also study the impact of perturbations on the performance of the proposed algorithm.

Lemma 2: Given an MEC network and a set of requests R, there is an Markov based approximation algorithm, Algorithm 1, which constructs a time-reversible Markov chain, and the stationary distribution is Equation (35).

Algorithm 1 Markov Based Approximation Algorithm

Input: An MEC network $G = (V, E)$ and a set of requests R.

Output: A scheduling of requests in R, s.t., the collected profit collected is maximized.

1: **procedure** INITIALIZATION
2: **for** $r_i \in R$ **do**
3: Initialize a thread for r_i;
4: The thread computes a feasible scheduling for r_i;
5: **end for**
6: The scheduling for each request r_i forms the current state w, and the expected profit Λ_w collected from state w is calculated.
7: Then let $\lambda_w = \Lambda_w + \tau$, where τ is a large constant to guarantee $\lambda_w \geq 0$;
8: Execute **TRANSIT**(r_i), $\forall r_i \in R$.
9: **end procedure**
10: **procedure** SET TIMER(r_i)
11: The thread of r_i creates a random exponentially timer ψ_i with the mean value γ;
12: The timer ψ_i starts to count down.
13: **end procedure**
14: **procedure** TRANSIT(r_i)
15: **while** It has not converged **do**
16: The thread of r_i randomly chooses a new feasible scheduling to form next state w';
17: Execute **SET TIMER**(r_i).
18: **if** ψ_i expires **then**
19: the thread of r_i transits to state w' with probability of $p_{w,w'} = \frac{exp(\beta \cdot \lambda_{w'})}{\max\{exp(\beta \cdot \lambda_w), exp(\beta \cdot \lambda_{w'})\}}$;
20: Broadcast **RESET** signals to other threads.
21: Execute **TRANSIT**(r_i).
22: **else if** ψ_i does not expire and the thread of r_i receives a **RESET** signal **then**
23: Terminate current timer.
24: Execute **TRANSIT**(r_i).
25: **end if**
26: **end while**
27: **end procedure**

Proof: As mentioned, we design a Markov chain model and guarantee that in the constructed Markov chain model, each state can be transited to any another state. We then show that the stationary distribution of the designed Markov chain model is Eq. (35). From Algorithm 1, it can be seen that we randomly select a state $w' \in W$ as the next state. And we generate a random exponentially timer ψ_i with mean value equal to γ for each request r_i. Furthermore, when the timer expires, the thread of r_i transits from current state w to next state w' with the probability of $p_{w,w'} = \frac{exp(\beta \cdot \lambda_{w'})}{\max\{exp(\beta \cdot \lambda_w), exp(\beta \cdot \lambda_{w'})\}}$. Thus, we are able to calculate the transition rate as follows,

$$\begin{aligned} q_{w,w'} &= \frac{p_{w,w'}}{|R| \cdot \gamma} \\ &= \frac{exp(\beta \cdot \lambda_{w'})}{|R| \cdot \gamma \cdot \max\{exp(\beta \cdot \lambda_w), exp(\beta \cdot \lambda_{w'})\}}, \end{aligned} \quad (38)$$

which is mentioned in Eq. (37).

Combining (35) and (37), it can be seen that $p^*_w \cdot q_{w,w'} = p^*_{w'} \cdot q_{w',w}, \forall w, w' \in W, w \neq w'$. The detailed balance equation of the designed Markov chain model is satisfied, and the designed Markov chain model is time-reversible, and the stationary distribution of which is Equation (35) [14]. Hence, the theorem follows. ■

Ideally, if the exact value of λ_w of each state $w, \forall w \in W$ can be obtained (i.e., the expected profit collected from each state can be accurately measured), then the designed Markov chain model will always converge to the preset stationary distribution p^*_w. However, the value of λ_w is very likely to be perturbed, and the designed Markov chain model might not be able to achieve the global optimality, instead a sub-optimal stationary distribution will be obtained [29].

In this paper, we assume the measurement perturbations are caused by the uncertainty of resource demands in the execution of requests, and the expected amounts of resources may experience perturbations as mentioned in Section III-B. In the sequel, it is necessary to analyze the impact of measurement perturbations on the solution and to mitigate the impact of such perturbations on the performance of the proposed approximation algorithm.

We quantify the perturbation error under a state $w \in W$ as a value ranged from $-\theta_w$ to θ_w, by introducing θ_w as a perturbation error bound, and λ_w is drawn from $2 \cdot n_w + 1$ discrete values: $\lambda_w - \theta_w, \ldots, \lambda_w - (1/n_w)\theta_w, \lambda_w, \lambda_w + (1/n_w)\theta_w, \ldots, \lambda_w + \theta_w$, where n_w is a positive constant.

Denote by $\alpha(w, j)$ the probability of the perturbed λ_w taking the value $\lambda_w + (j/n_w) \cdot \theta_w$, $\forall w \in W$, $\forall j = \{-n_w, \ldots, n_w\}$, and $\sum_{j \in \{-n_w, \ldots, n_w\}} \alpha(w, j) = 1$.

Lemma 3: Given an MEC network *G*(*V*, *E*), with measurement perturbations, the stationary distribution of the Markov chain model, denoted by $\overline{p}_w$, is

$$\overline{p}_w = \frac{\kappa_w \cdot exp(\beta \cdot \lambda_w)}{\sum_{w' \in W} \kappa_{w'} \cdot exp(\beta \cdot \lambda_{w'})}, \quad \forall w \in W, \tag{39}$$

where $\kappa_w = \sum_{j \in \{-n_w, \ldots, n_w\}} \alpha(w, j) \cdot exp(\beta \cdot ((j \cdot \theta_w)/n_w))$.

The proof can be found in Appendix-A.

Theorem 2: Given an MEC network *G*(*V*, *E*) and a set of requests *R*, without analysis perturbation, there is an Markov based approximation algorithm, Algorithm 1, its optimality gap is as follows.

$$0 \leq \Lambda_{max} - \Lambda_{avg} \leq \frac{\ln |W|}{\beta}, \tag{40}$$

where the optimality gap of an algorithm is the absolute difference between the solution obtained by the algorithm and the optimal solution of the problem. $\Lambda_{max} = \max_{w \in W}\{\Lambda_w\}$ is the value of the optimal solution, Λ_{avg} is the expected profit with the designed Markov chain model and *W* is the collection of all states.

The proof can be found in Appendix-B.

Theorem 3: Given an MEC network and a set of requests *R*, with analysis perturbation, there is an Markov based approximation algorithm, Algorithm 1, its optimality gap is given as follows.

$$0 \leq \Lambda_{max} - \overline{\Lambda}_{avg} \leq \frac{\ln |W|}{\beta} + \theta_{max}, \tag{41}$$

where $\overline{\Lambda}_{avg} = \sum_{w \in W} \overline{p}_w \cdot \Lambda_w$ is the expected profit with the perturbed Markov chain model, and the time fraction distribution $\overline{p}_w$, $\forall w \in W$, is the optimal solution under the perturbation case, and $\theta_{max} = \max_{w \in W}\{\theta_w\}$.

The proof can be found in Appendix-C.

In the context of Markov Chain, the convergence time is examined by the mixing time [5].

Definition 2 [5]: Let $H_t(w)$ be the probability distribution of all states in *W* at time t with the initial state *w*, $\boldsymbol{p}^*$ be the stationary distribution of the Markov Chain, and $\epsilon > 0$ be a constant and represent the gap between the converged solution and the optimal one, then the mixing time of the constructed Markov Chain is defined as

$$t_{mix}(\epsilon) \coloneqq inf\left\{ t \geq 0 : \max_{w \in W} \| H_t(w) - \boldsymbol{p}^* \|_{TV} \leq \epsilon \right\}, \tag{42}$$

where term $||H_t(w) - \boldsymbol{p}^*||_{TV}$ is the total variance distance between $H_t(w)$ and $\boldsymbol{p}^*$.

Denote by $\Lambda_{min} = \min_{w \in W}\{\Lambda_w\}$ and recall that $\Lambda_{max} = \max_{w \in W}\{\Lambda_w\}$.

Theorem 4: Given an MEC network *G*(*V*, *E*) and a set *R* of requests, there is an Markov based approximation algorithm, Algorithm 1, its convergence time is bounded as follows.

$$t_{mix} \geq \frac{|R| \cdot \gamma \cdot exp(\beta \cdot (\Lambda_{min} - \Lambda_{max}))}{2 \cdot |W|} \cdot \ln \frac{1}{2 \cdot \epsilon},$$

and

$$t_{mix} \leq 2 \cdot |W|^3 \cdot |R| \cdot \gamma \cdot exp(5 \cdot \beta \cdot (\Lambda_{max} - \Lambda_{min})) \times \left(\ln \frac{1}{2 \cdot \epsilon} + \frac{1}{2} \cdot (\ln |W| + \beta \cdot (\Lambda_{max} - \Lambda_{min})) \right).$$

The proof can be found in Appendix-D.

VI. Performance Evaluation

In this section, the performance of the proposed algorithm is evaluated by experimental simulations. The impact of parameters on the performance of the proposed algorithm is investigated, too.

A. Environment Settings

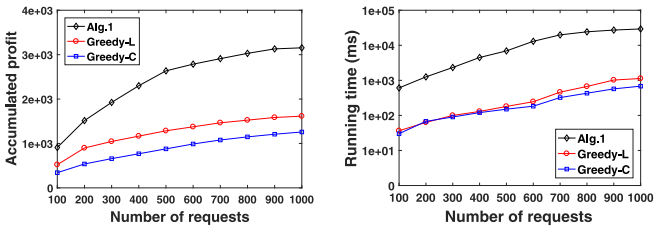
We generate topologies of MEC networks through a tool GT-ITM [33]. We consider an MEC network with 100 APs, and 10 percent of the APs are randomly selected to be co-located with cloudlets. The capacities of cloudlets are randomly drawn between 30,000MHz and 60,000MHz [9]. We further assume that there are 20 types of VNFs, and the expected computing resource demanded by each type of VNFs is ranged from 40MHz to 600MHz [2]. The length of a requested SFC is randomly chosen from 2 to 10 [28] and each VNF instance is randomly drawn from the provisioned VNFs. The bandwidth capacity of a link is randomly drawn from 2,000Mbps to 20,000Mbps [10]. The transmission latency of a link is randomly drawn from 2ms to 7ms [23]. The expected demanded data rate of each request varies from 1 to 10 packets per millisecond, and the size of each packet is 64KB [22]. The upper bound of the demanded computing resource and

data rate are randomly set as 105%, 110% or 115% of the expected one, while the lower bound of those are randomly set as 95%, 90% or 85% of the expected one. Considering the perturbation, the actual consumed computing resource and data rate are randomly drawn between the upper bounds and lower bounds. The actual demanded resource is utilized to calculate the actual accumulated profit. The processing rate of each VNF instance varies from 5 to 20 packets per millisecond [22]. The latency requirement of each request is set from 20ms to 100ms randomly [23]. The computing resource cost on each cloudlet is randomly drawn from a range between \$0.05 to \$0.2 per MHz, while the bandwidth cost on each link varies from \$0.002 to \$0.005 per Mbps [36]. Each value in figures is the mean of the results of 20 topologies randomly generalized by GT-ITM [33] with the same size. The payment of each request is randomly drawn from \$10 to \$30. The parameter β in the designed Markov Chain is set as 1 and the control parameter ζ associated with the cost variation is set as 0.1 [8]. The running time of each algorithm is based on a desktop with a 3.60 GHz Intel 8-Core i7-7700 CPU and 16 GB RAM. Unless specified, the above parameters are adopted by default.

We here introduce two benchmarks against the proposed algorithm. The first is a greedy algorithm considering the least latency on links for each request, referred to as `Greedy-L`. We consider the set R of requests randomly in sequence. For request $r_i \in R$, we first eliminate the nodes and links with insufficient resource for its admission through constructing an auxiliary graph. `Greedy-L` finds a feasible path with sufficient residual resource and the least latency on links from the source node to the destination in the auxiliary graph. Then, we find a feasible placement of the requested SFC on the chosen routing path, following the computing resource constraint and latency constraint. A request is rejected if `Greedy-L` cannot find a feasible placement for its SFC. The other benchmark is a greedy algorithm considering the least cost on the links for each request, referred to as `Greedy-C`. Similarly, for each request r_i, `Greedy-C` first finds a feasible routing path with the least cost on links from the source node to the destination node in the auxiliary graph with sufficient resource for its admission. Then, we find a feasible placement of the requested SFC on the chosen routing path, following the computing resource constraint and latency constraint. A request is rejected if `Greedy-C` cannot find a feasible placement for its SFC. The average result delivered by each algorithm is calculated based on 20 topologies of the same size. According to [31], we assume that the Markov based approximation algorithm converges if the collected profit does not change more than 0.1% of the value obtained at the current state.

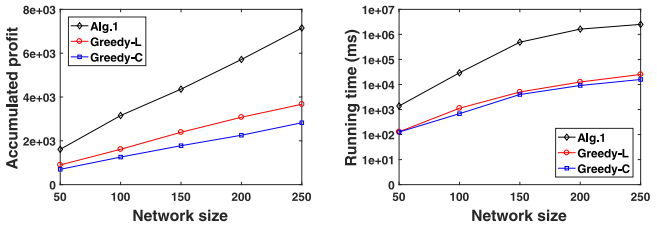
B. Performance Evaluation of Different Algorithms

We first studied the performance of different algorithms, by varying the number of requests from 100 to 1,000 with the network size of 100. Figure 3(a) depicts the accumulated profit with varying number of requests while keeping other settings unchanged. In addition, the associated running time is shown in Figure 3(b). It can be seen from Figure 3(a) that Algorithm 1 outperforms the benchmarks `Greedy-L` and `Greedy-C` in all cases. When the number of requests is 100, the profits collected by `Greedy-L` and `Greedy-C` are 57.4% and 37.6% of that by Algorithm 1, respectively. This is because Algorithm 1 has an advantage in lowering the admission cost when the network has enough resource to admit all requests. When the number of requests is 1,000, the profits collected by `Greedy-L` and `Greedy-C` are 51.2% and 39.8% of that by Algorithm 1, respectively. This is because Algorithm 1 delivers a more reasonable scheduling of resource to admit more requests with the limited resource in a network.



(a) The accumulated profit (b) The running time

Fig. 3. Performance of different algorithms by varying the number of requests from 100 to 1,000 with the network size of 100.



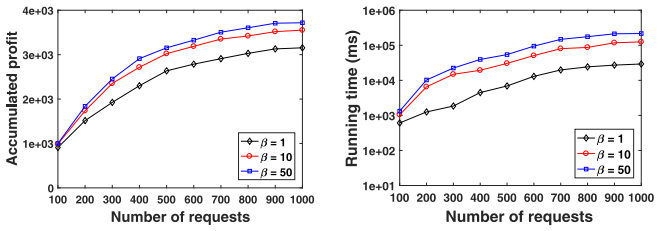
(a) The accumulated profit (b) The running time

Fig. 4. Impact of network size on different algorithms by varying the number of nodes from 50 to 250 with 1,000 requests.

C. Impact of Different Parameters on the Performance of the Proposed Algorithm

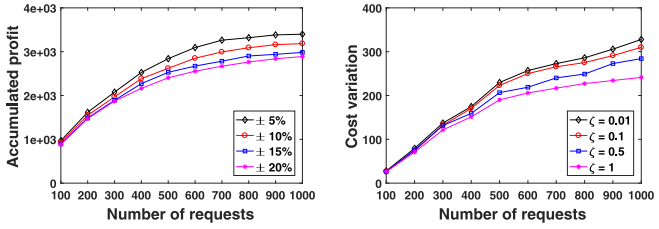
We then investigated the impacts of important parameters on the performance of the proposed algorithm, such as the network size, the parameter β, the uncertainty of demanded resource and the control parameter ζ. Recall that β is an important parameter in designing a Markov chain. And the control parameter ζ is related to the cost variation.

We started by investigating the impact of network size on the performance of the proposed algorithm against the benchmarks `Greedy-L` and `Greedy-C`, by varying the network size from 50 to 250 with 1,000 requests. Recall that the number of cloudlets is set as 10% of the network size. Figure 4(a) depicts the accumulated profit with varying numbers of network size, while Figure 4(b) depicts the related running time. It can be seen from Figure 4(a) that when the network size is 250, the profit collected by `Greedy-L` is 52.3% of that by Algorithm 1, while the profit collected by `Greedy-C` is 38.5% of that by Algorithm 1. This can be justified that when the network size is large, compared with the greedy algorithms, Algorithm 1 achieves better utilization



(a) The accumulated profit (b) The running time

Fig. 5. Impact of parameter β on the proposed algorithm by varying the number of requests from 100 to 1,000 with the network size of 100.



(a) Impact of demanded resource uncertainlity (b) Impact of control parameter ζ.

Fig. 6. Impact of uncertainty of demanded resource and control parameter ζ on the proposed algorithm by varying numbers of requests from 100 to 1,000 with the network size of 100.

of computing resource and bandwidth resource to avoid the overloading on links and cloudlets.

We then studied the impact of parameter β on the performance of the proposed algorithm, by varying the number of requests from 100 to 1,000 with the network size of 100. Figure 5(a) demonstrates the impact of parameter β on the collected profit of Algorithm 1 while Figure 5(b) demonstrates the convergence time with different β. With 1,000 requests, when parameter $\beta = 50$, the proposed algorithm achieves the best performance, which is 17.8% higher than that by the proposed algorithm with parameter $\beta = 1$. However, in this case, it takes the longest convergence time. The rationale behind this is that when the value of parameter β is small, the optimality gap is enlarged by Theorem 2. However, following the convergence time analysis by Theorem 4, it consumes much less time to achieve convergence. It implies that Algorithm 1 demonstrates good flexibility for us to set parameter β with a reasonable value to achieve a good trade-off between the performance and convergence time.

We thirdly evaluated the impact of the uncertainty of demanded resource on the performance of the proposed algorithm, by varying the number of requests from 100 to 1,000 with the network size of 100. Figure 6(a) depicts the accumulated profit obtained with different uncertainties of demanded resource. E.g., the uncertainty of demanded resource is $\pm 5\%$ when the upper bounds of the demanded computing resource and data rate are set as 105% of the expected value, while the lower bounds of the demanded computing resource and data rate are set as 95% of the expected value. The actual consumed computing resource and data rate are randomly drawn between the upper bounds and lower bounds. It can be seen from Figure 6(a) that the proposed algorithm performs better with lower uncertainty of demanded resource. With 1,000 requests, when the uncertainty is $\pm 20\%$, Algorithm 1 achieves 84.9% of the accumulated profit by itself when the uncertainty is $\pm 5\%$. The reason is that higher uncertainty of demanded resource not only enlarges the cost variation, but also perturbs the stationary distribution of the designed Markov chain by Lemma 3.

We finally studied the impact of the control parameter ζ on the performance of the proposed algorithm, by varying the number of requests from 100 to 1,000 with the network size of 100. Figure 6(b) depicts the cost variation with varying control parameter ζ. Recall that the control parameter ζ is used to stabilize the total admission cost as a coefficient of cost variation. With 1,000 requests, when control parameter $\zeta = 1$, Algorithm 1 achieves 73.7% of the cost variation by itself when control parameter $\zeta = 0.01$. It can be seen from Figure 6(b) that a larger ζ leads to a smaller cost variation, and the total admission cost becomes more stable with a larger ζ. The reason is that we assign a higher weight to cost variation compared with the accumulated cost.

VII. Conclusion

In this paper, we studied user service request admissions with both SFC and latency requirements in an MEC network. We first formulated a novel RSFCP problem with the aim to maximize the expected profit of the network service provider through admitting as many user requests as possible. We then formulated a QIP exact solution to the problem when its size is small or moderate. Furthermore, we developed a Markov based approximation algorithm, which can deliver a near-optimal solution with a bounded moderate gap for the problem without measurement perturbation. We also extended the proposed approach to the measurement perturbation case, for which we showed that the proposed approximation algorithm is still applicable, and the solution delivered has a near-optimal gap with a guaranteed error bound. We finally evaluated the performance of the proposed algorithm through experimental simulations with practical settings. Experimental results demonstrated that the proposed algorithm is promising, and outperforms the mentioned benchmarks. Several potential topics based on this study can be further explored in the future. For example, the problem can be extended to an online setting where requests arrive one by one without the knowledge of future arrival information. Furthermore, the mobility of mobile users can also be taken into account when dealing with robust service provisioning.

Appendix A
Proof of Lemma 3

The core idea of the proof is to first deliver the modified transition rate with perturbations by treating each state w as $2 \cdot n_w + 1$ states. With (35) and the detailed balance equations, the stationary distribution of the Markov chain model with perturbations can be obtained then.

In the case of perturbation, each state w are then treated as $2 \cdot n_w + 1$ states (i.e., w_j, $\forall j = \{-n_w, \ldots, n_w\}$) with perturbation error bound θ_w. Denote by $\lambda_{w,j}$, the expected

profit collected in state w_j added by τ, we have

$$\lambda_{w,j} = \lambda_w + (j/n_w)\cdot\theta_w, \quad \forall w \in W, \\ \forall j \in \{-n_w, \ldots, n_w\}. \tag{43}$$

The modified transition rate $\overline{q}_{w_j,w'_{j'}}$ with perturbations is

$$\overline{q}_{w_j,w'_{j'}} = \frac{\alpha(w',j')\cdot exp\big(\beta\cdot\lambda_{w',j'}\big)}{|R|\cdot\gamma\cdot\max\big\{exp\big(\beta\cdot\lambda_{w',j'}\big), exp\big(\beta\cdot\lambda_{w,j}\big)\big\}}, \tag{44}$$

With regard to the detailed balance equations $p_{w_j}\cdot\overline{q}_{w_j,w'_{j'}} = p_{w'_{j'}}\cdot\overline{q}_{w'_{j'},w_j}$, we have

$$\frac{p_{w_0}}{\alpha(w,0)\cdot exp(\beta\cdot\lambda_{w_0})} = \frac{p_{w'_{j'}}}{\alpha(w',j')\cdot exp\Big(\beta\cdot\lambda_{w'_{j'}}\Big)}, \tag{45}$$

where w_0 is state w with no perturbation and $\alpha(w,0)$ is the probability that no perturbation exists for state w.

From (35) and (45), we have

$$p_{w_j} = \frac{\alpha(w,j)\cdot exp\big(\beta\cdot\lambda_{w_j}\big)}{\sum_{w'\in W}\sum_{j'\in\{-n_w,\ldots,n_w\}}\cdot\alpha(w',j')\cdot exp\Big(\beta\cdot\lambda_{w'_{j'}}\Big)}, \\ \forall w \in W, \quad \forall j \in \{-n_w, \ldots, n_w\}. \tag{46}$$

Denote by $\kappa_w = \sum_{j\in\{-n_w,\ldots,n_w\}}\alpha(w,j)\cdot exp(\beta\cdot((j\cdot\theta_w)/n_w))$, we have

$$\begin{aligned}\overline{p}_w &= \sum_{j\in\{-n_w,\ldots,n_w\}}\cdot p_{w_j} \\ &= \frac{\sum_{j\in\{-n_w,\ldots,n_w\}}\alpha(w,j)\cdot exp\big(\beta\cdot\lambda_{w_j}\big)}{\sum_{w'\in W}\sum_{j'\in\{-n_w,\ldots,n_w\}}\cdot\alpha(w',j')\cdot exp\Big(\beta\cdot\lambda_{w'_{j'}}\Big)} \\ &= \frac{\kappa_w\cdot exp(\beta\cdot\lambda_w)}{\sum_{w'\in W}\kappa_{w'}\cdot exp(\beta\cdot\lambda_{w'})}, \text{by (43).}\end{aligned} \tag{47}$$

Thus, Lemma 3 follows. ∎

APPENDIX B
PROOF OF THEOREM 2

We show the optimality gap as follows. Let w_{max} be the state to obtain the maximum profit. Let the time fraction distribution $\tilde{p}_w, \forall w \in W$ be the optimal solution. We then have

$$\tilde{p}_w = \begin{cases} 1, & if\ w = w_{max}, \\ 0, & otherwise. \end{cases} \tag{48}$$

As p^*_w is the desired stationary distribution calculated in (35), following Lemma 1, we have

$$\begin{aligned}&\sum_{w\in W} p^*_w\cdot\lambda_w - \frac{1}{\beta}\cdot\sum_{w\in W} p^*_w\cdot\ln p^*_w \\ &\quad\geq \sum_{w\in W}\tilde{p}_w\cdot\lambda_w - \frac{1}{\beta}\cdot\sum_{w\in W}\tilde{p}_w\cdot\ln\tilde{p}_w = \lambda_{max},\end{aligned} \tag{49}$$

where $\lambda_{max} = \max_{w\in W}\{\lambda_w\}$.

Apply Jensen's inequality [3], we have

$$\begin{aligned}\sum_{w\in W} p^*_w\cdot\ln p^*_w &= -\sum_{w\in W} p^*_w\cdot\ln\frac{1}{p^*_w} \\ &\geq -\ln\left(\sum_{w\in W} p^*_w\cdot\frac{1}{p^*_w}\right) = -\ln|W|.\end{aligned} \tag{50}$$

Combining equations (49) and (50), we have

$$\begin{aligned}\lambda_{avg} &= \sum_{w\in W} p^*_w\cdot\lambda_w \leq \sum_{w\in W} p^*_w\cdot\lambda_{max} = \lambda_{max} \\ &\leq \lambda_{avg} - \frac{1}{\beta}\cdot\sum_{w\in W} p^*_w\cdot\ln p^*_w \leq \lambda_{avg} + \frac{1}{\beta}\cdot\ln|W|,\end{aligned} \tag{51}$$

where λ_{avg} is the expected value with the designed Markov chain model.

Thus, we have

$$0 \leq \lambda_{max} - \lambda_{avg} \leq \frac{\ln|W|}{\beta}. \tag{52}$$

Because $\lambda_w = \Lambda_w + \tau$, $\forall w \in W$, we have,

$$0 \leq \Lambda_{max} - \Lambda_{avg} \leq \frac{\ln|W|}{\beta}. \tag{53}$$

The theorem then follows. ∎

APPENDIX C
PROOF OF THEOREM 3

Recall that $\kappa_w = \sum_{j\in\{-n_w,\ldots,n_w\}}\alpha(w,j)\cdot exp(\beta\cdot((j\cdot\theta_w)/n_w))$, $\forall w \in W$, we have

$$exp(-\beta\cdot\theta_w) \leq \kappa_w \leq exp(\beta\cdot\theta_w). \tag{54}$$

$$-\theta_w \leq \frac{\ln\kappa_w}{\beta} \leq \theta_w. \tag{55}$$

From Lemma 3, we have,

$$\overline{p}_w = \frac{exp\left(\beta\cdot\left(\lambda_w + \frac{\ln\kappa_w}{\beta}\right)\right)}{\sum_{w'\in W} exp\left(\beta\cdot\left(\lambda_{w'} + \frac{\ln\kappa_{w'}}{\beta}\right)\right)}, \quad \forall w \in W. \tag{56}$$

As a result, for $\lambda_{w''} = \lambda_w + \frac{\ln\kappa_w}{\beta}$, the stationary distribution with perturbation $\overline{p}_w, \forall w \in W$, works as an optimal solution in this case.

From Theorem 2, we have,

$$\max_{w''\in W}\lambda_{w''} - \sum_{w''\in W}\overline{p}_{w''}\cdot\lambda_{w''} \leq \frac{\ln|W|}{\beta}. \tag{57}$$

Let $\lambda_{w''} = \lambda_w + \frac{\ln \kappa_w}{\beta}$, we have,

$$\max_{w\in W}\left(\lambda_w + \frac{\ln \kappa_w}{\beta}\right) - \sum_{w\in W}\left(\overline{p}_w \cdot \lambda_w + \frac{\ln \kappa_w}{\beta}\right) \leq \frac{\ln |W|}{\beta}. \quad (58)$$

Then,

$$\begin{aligned}\lambda_{max} &= \max_{w\in W}\lambda_w \leq \max_{w\in W}\left(\lambda_w + \frac{\ln \kappa_w}{\beta}\right)\\ &\leq \sum_{w\in W}\left(\overline{p}_w \cdot \lambda_w + \frac{\ln \kappa_w}{\beta}\right) + \frac{\ln |W|}{\beta}, \quad \text{by (58)},\\ &\leq \sum_{w\in W}\overline{p}_w \cdot \lambda_w + \theta_w + \frac{\ln |W|}{\beta}, \quad \text{by (55)}. \quad (59)\end{aligned}$$

Also,

$$\begin{aligned}\overline{\lambda}_{avg} &= \sum_{w\in W}\overline{p}_w \cdot \lambda_w \leq \sum_{w\in W}\overline{p}_w \cdot \lambda_{max} = \lambda_{max}\\ &\leq \overline{\lambda}_{avg} + \theta_w + \frac{\ln |W|}{\beta}, \quad \text{by (59)},\end{aligned}$$

where $\overline{\lambda}_{avg}$ is the expected value with the perturbed Markov chain model.

We thus have

$$0 \leq \lambda_{max} - \overline{\lambda}_{avg} \leq \frac{\ln |W|}{\beta} + \theta_w \leq \frac{\ln |W|}{\beta} + \theta_{max}.$$

Because $\lambda_w = \Lambda_w + \tau$, $\forall w \in W$, we have,

$$0 \leq \Lambda_{max} - \overline{\Lambda}_{avg} \leq \frac{\ln |W|}{\beta} + \theta_{max}.$$

Hence, the theorem follows. ∎

Appendix D
Proof of Theorem 4

By stationary distribution (35), the minimum probability among the stationary distribution, denoted by p_{min}, is:

$$\begin{aligned}p_{min} &\coloneqq \min_{w\in W} p^*_w \geq \frac{exp(\beta \cdot \Lambda_{min})}{|W| \cdot exp(\beta \cdot \Lambda_{max})}, \quad \text{by (35)},\\ &= \frac{1}{|W|} \cdot exp(\beta(\Lambda_{min} - \Lambda_{max})). \quad (60)\end{aligned}$$

We then adopt the uniformization technique [5]. Denote by $\mathcal{Q} = \{q_{w,w'}\}$ the transition matrix of the constructed Markov Chain. Then a Markov Chain $Z(m)$ is constructed with a transition matrix $P = I + \frac{\mathcal{Q}}{\sigma}$, where I denotes a unit matrix and σ denotes the uniform rate parameter. We then assume that with the Markov Chain $Z(n)$, the system transits its state following the Poisson process $N(t)$ with rate σ [10]. Denote by $Z(N(t))$ the state of the system at time t.

From the transition rate (37), we have, $\forall w, w' \in W$,

$$q_{w,w'} \leq \frac{1}{|R| \cdot \gamma} \cdot exp(\beta \cdot (\Lambda_{max} - \Lambda_{min})). \quad (61)$$

And we have

$$\sum_{w\neq w'} q_{w,w'} \leq \frac{|W|}{|R| \cdot \gamma} exp(\beta \cdot (\Lambda_{max} - \Lambda_{min})). \quad (62)$$

Then we have σ as:

$$\sigma = \frac{|W|}{|R| \cdot \gamma} exp(\beta \cdot (\Lambda_{max} - \Lambda_{min})). \quad (63)$$

According to the uniformization theorem [15], the Markov Chain and its counterpart $Z(N(t))$ with discrete-time manner share the same probability distribution. Denote by $\rho 2$ the second eigenvalue of transition matrix $\boldsymbol{P}$ for $Z(n)$. We then adopt the spectral gap inequality [15], we have,

$$\begin{aligned}\frac{exp(-\sigma \cdot (1-\rho 2) \cdot t)}{2} &\leq \max_{w\in W}\|H_t(w) - \boldsymbol{p}^*\|_{TV}\\ &\leq \frac{exp(-\sigma \cdot (1-\rho 2) \cdot t)}{2 \cdot (p_{min})^{\frac{1}{2}}}. \quad (64)\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{1}{\sigma \cdot (1-\rho 2)} \cdot \ln \frac{1}{2\cdot\epsilon} &\leq t_{mix}(\epsilon)\\ &\leq \frac{1}{\sigma \cdot (1-\rho 2)}\\ &\quad \times \left(\ln \frac{1}{2\cdot\epsilon} + \frac{1}{2} \cdot \ln \frac{1}{p_{min}}\right). \quad (65)\end{aligned}$$

With Cheeger's inequality [6], we bound $\rho 2$ as follows:

$$1 - 2 \cdot \Phi \leq \rho 2 \leq 1 - \frac{1}{2} \cdot \Phi^2, \quad (66)$$

where Φ is the "conductance" of $\boldsymbol{P}$ and is defined as follows,

$$\Phi \coloneqq \min_{N\subset W, \pi_N \in (0,0.5]} \frac{\mathbb{F}(N, N^c)}{\pi_N}, \quad (67)$$

where $\pi_N = \sum_{w\in N} p^*_w$ and $\mathbb{F}(N, N^c) = \sum_{w\in N, w'\in N^c} p^*_w \cdot P(w, w')$. With (65) and (66), we have,

$$\begin{aligned}\frac{1}{2\cdot\sigma\cdot\Phi} \cdot \ln \frac{1}{2\cdot\epsilon} &\leq t_{mix}(\epsilon) \leq \frac{2}{\sigma \cdot \Phi^2}\\ &\quad \times \left(\ln \frac{1}{2\cdot\epsilon} + \frac{1}{2} \cdot \ln \frac{1}{p_{min}}\right). \quad (68)\end{aligned}$$

Then, $\forall N' \subset W, \pi_{N'} \in (0, 0.5]$, we have

$$\begin{aligned}\Phi &\coloneqq \min_{N\subset W, \pi_N \in (0,0.5]} \frac{\mathbb{F}(N, N^c)}{\pi_N}\\ &\leq \frac{1}{\pi_{N'}} \cdot \sum_{w\in N, w'\in N'^c} p^*_w \cdot P\big(w, w'\big)\\ &= \frac{1}{\pi_{N'}} \cdot \sum_{w\in N'} \cdot p^*_w \cdot \sum_{w'\in N'^c} P\big(w, w'\big)\\ &\leq \frac{1}{\pi_{N'}} \cdot \sum_{w\in N'} \cdot p^*_w = 1 \quad (69)\end{aligned}$$

From (63), (68) and (69), we have the lower bound of $t_{mix}(\epsilon)$ as follows,

$$\begin{aligned}t_{mix}(\epsilon) &\geq \frac{1}{2\cdot\sigma} \cdot \ln \frac{1}{2\cdot\epsilon}\\ &= \frac{|R| \cdot \gamma \cdot exp(\beta \cdot (\Lambda_{min} - \Lambda_{max}))}{2 \cdot |W|} \cdot \ln \frac{1}{2\cdot\epsilon}. \quad (70)\end{aligned}$$

From the transition rate (37), we have, $\forall w, w' \in W$,

$$q_{w,w'} \geq \frac{1}{|R| \cdot \gamma} \cdot exp(\beta \cdot (\Lambda_{min} - \Lambda_{max})). \quad (71)$$

From (67), we have,

$$\begin{aligned}\Phi &\geq \min_{N\subset W,\pi_N\in(0,0.5]} \mathbb{F}(N,N^c) \geq \min_{w\neq w',p(w,w')>0} \mathbb{F}\big(w,w'\big)\\ &= \min_{w\neq w',p(w,w')>0} p_w^* \cdot P\big(w,w'\big) = \min_{w\neq w',p(w,w')>0} p_w^*\\ &\quad\times \frac{q_{w,w'}}{\sigma}\\ &\geq \frac{p_{min}}{\sigma}\cdot\frac{1}{|R|\cdot\gamma}\cdot exp(\beta\cdot(\Lambda_{min}-\Lambda_{max})), \text{by (71)}\ .\end{aligned} \tag{72}$$

From (68), we have the upper bound of $t_{mix}(\epsilon)$ as follows,

$$\begin{aligned}t_{mix}(\epsilon) &\leq \frac{2}{\sigma\cdot\Phi^2}\cdot\left(\ln\frac{1}{2\cdot\epsilon}+\frac{1}{2}\cdot\ln\frac{1}{p_{min}}\right)\\ &\leq \frac{2\cdot|R|^2\cdot\gamma^2\cdot\sigma\cdot exp(2\cdot\beta\cdot(\Lambda_{max}-\Lambda_{min}))}{p_{min}^2}\\ &\quad\times\left(\ln\frac{1}{2\cdot\epsilon}+\frac{1}{2}\cdot\ln\frac{1}{p_{min}}\right), \text{by (72)}\\ &= \frac{2\cdot|W|\cdot|R|\cdot\gamma\cdot exp(3*\beta\cdot(\Lambda_{max}-\Lambda_{min}))}{p_{min}^2}\\ &\quad\times\left(\ln\frac{1}{2\cdot\epsilon}+\frac{1}{2}\cdot\ln\frac{1}{p_{min}}\right), \text{by (63)}\\ &\leq 2\cdot|W|^3\cdot|R|\cdot\gamma\cdot exp(5\cdot\beta\cdot(\Lambda_{max}-\Lambda_{min}))\\ &\quad\times\left(\ln\frac{1}{2\cdot\epsilon}+\frac{1}{2}\cdot(\ln|W|+\beta\cdot(\Lambda_{max}-\Lambda_{min}))\right),\\ &\quad \text{by (60)}.\end{aligned}$$

Hence, the theorem follows. ∎

ACKNOWLEDGEMENT

The authors appreciate the three anonymous referees and the Associate Editor for their constructive comments and invaluable suggestions, which help them improve the quality and presentation of the paper greatly.

REFERENCES

[1] S. Ali, A. A. Maciejewski, H. J. Siegel, and J.-K. Kim, "Measuring the robustness of a resource allocation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 7, pp. 630–641, Jul. 2004.

[2] M. T. Beck and J. F. Botero, "Coordinated allocation of service function chains," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2016, pp. 1–6.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[4] H. Chen *et al.*, "Towards optimal outsourcing of service function chain across multiple clouds," in *Proc. IEEE Int. Commun. Conf. (ICC)*, 2016, pp. 1–7.

[5] M. Chen, S. C. Liew, Z. Shao, and C. Ka, "Markov approximation for combinatorial network optimization," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2010, pp. 1783–1791.

[6] P. Diaconis and D. Stroock, "Geometric bounds for Eigenvalues of Markov chains," *Ann. Appl. Probab.*, vol. 1, no. 1, pp. 36–61, 1991.

[7] F. Esposito *et al.*, "Necklace: An architecture for distributed and robust service function chains with guarantees," *IEEE Trans. Netw. Service Manag.*, early access, Nov. 10, 2020, doi: 10.1109/TNSM.2020.3036926.

[8] N. Eshraghi and B. Liang, "Joint offloading decision and resource allocation with uncertain task computing requirement," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Paris, France, 2019, pp. 1414–1422.

[9] J. Fan, C. Guan, Y. Zhao, and C. Qiao, "Availability-aware mapping of service function chains," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2017, pp. 1–9.

[10] H. Huang, S. Guo, W. Liang, K. Li, B. Ye, and W. Zhuang, "Near-optimal routing protection for in-band software-defined heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 11, pp. 2918–2934, Nov. 2016.

[11] L.-T. Hwang and T.-S. J. Horng, *3D IC and RF SiPs: Advanced Stacking and Planar Solutions for 5G Mobility*. Hoboken, NJ, USA: Wiley, 2018.

[12] M. Jalalitabar, E. Guler, G. Luo, L. Tian, and X. Cao, "Dependence-aware service function chain design and mapping," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, 2017, pp. 1–6.

[13] M. Jia, W. Liang, M. Huang, Z. Xu, and Y. Ma, "Routing cost minimization and throughput maximization of NFV-enabled unicasting in software-defined networks," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 2, pp. 732–745, Jun. 2018.

[14] F. P. Kelly, *Reversibility and Stochastic Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[15] D. A. Levin and Y. Peres, *Markov Chains and Mixing Times*, vol. 107. Providence, RI, USA: Amer. Math. Soc., 2017.

[16] J. Li, W. Liang, M. Huang, and X. Jia, "Providing reliability-aware virtualized network function services for mobile edge computing," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Dallas, TX, USA, 2019, pp. 732–741.

[17] J. Li, W. Liang, M. Huang, and X. Jia, "Reliability-aware network service provisioning in mobile edge-cloud networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 7, pp. 1545–1558, Jul. 2020.

[18] J. Li, W. Liang, W. Xu, Z. Xu, and J. Zhao, "Maximizing the quality of user experience of using services in edge computing for delay-sensitive IoT applications," in *Proc. 23rd Int. ACM Conf. Model. Anal. Simulat. Wireless Mobile Syst. (MSWiM)*, 2020, pp. 113–121.

[19] W. Liang, Y. Ma, W. Xu, X. Jia, and S. C.-K. Chau, "Reliability augmentation of requests with service function chain requirements in mobile edge-cloud networks," in *Proc. 49th Int. Conf. Parallel Process. (ICPP)*, 2020, pp. 1–11.

[20] S. Lin, W. Liang, and J. Li, "Reliability-aware service function chain provisioning in mobile edge-cloud networks," in *Proc. 29th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Honolulu, HI, USA, 2020, pp. 1–9.

[21] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 3, pp. 543–553, Sep. 2017.

[22] Y. Ma, W. Liang, J. Wu, and Z. Xu, "Throughput maximization of NFV-enabled multicasting in mobile edge cloud networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 2, pp. 393–407, Feb. 2020.

[23] Y. Ma, W. Liang, Z. Xu, and S. Guo, "Profit maximization for admitting requests with network function services in distributed clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 5, pp. 1143–1157, May 2019.

[24] M. Nguyen, M. Dolati, and M. Ghaderi, "Deadline-aware SFC orchestration under demand uncertainty," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2275–2290, Dec. 2020.

[25] D. Pisinger, *Algorithms for Knapsack Problems*, Ph.D. thesis, Univ. Copenhagen, Denmark, 1995. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.9780

[26] K. Psychas and J. Ghaderi, "Scheduling jobs with random resource requirements in computing clusters," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Paris, France, 2019, pp. 2269–2277.

[27] H. Ren *et al.*, "Efficient algorithms for delay-aware NFV-enabled multicasting in mobile edge clouds with resource sharing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 9, pp. 2050–2066, Sep. 2020.

[28] G. Sallam, G. R. Gupta, B. Li, and B. Ji, "Shortest path and maximum flow problems under service function chaining constraints," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Honolulu, HI, USA, 2018, pp. 2132–2140.

[29] Z. Shao, X. Jin, W. Jiang, M. Chen, and M. Chiang, "Intra-data-center traffic engineering with ensemble routing," in *Proc. IEEE INFOCOM*, Turin, Italy, 2013, pp. 2148–2156.

[30] G. Sun, Y. Li, D. Liao, and V. Chang, "Service function chain orchestration across multiple domains: A full mesh aggregation approach," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 3, pp. 1175–1191, Sep. 2018.

[31] Z. Shao, H. Zhang, M. Chen, and K. Ramchandran, "Reverse-engineering BitTorrent: A Markov approximation perspective," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012, pp. 2996–3000.

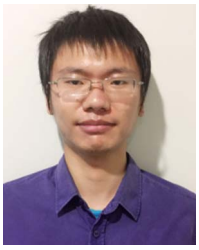
[32] D. B. Shmoys, J. Wein, and D. P. Williamson, "Scheduling parallel machines on-line," *SIAM J. Comput.*, vol. 24, no. 6, pp. 1313–1331, 1995.

[33] M. Thomas, E. Edwards, and S. Bhattacharjee. (2020). *GT-ITM*. [Online]. Available: http://www.cc.gatech.edu/projects/gtitm/

[34] A. Tomassilli, N. Huin, F. Giroire, and B. Jaumard, "Resource requirements for reliable service function chaining," in *Proc. Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, 2018, pp. 1–7.
[35] M. Wang, B. Cheng, S. Wang, and J. Chen, "Availability-and traffic-aware placement of parallelized SFC in data center networks," *IEEE Trans. Netw. Service Manag.*, early access, Jan. 18, 2021, doi: 10.1109/TNSM.2021.3051903.
[36] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis, "Efficient NFV-enabled multicasting in SDNs," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2052–2070, Mar. 2019.
[37] Z. Zhu, H. Lu, J. Li, and X. Jiang, "Service function chain mapping with resource fragmentation avoidance," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, 2017, pp. 1–6.
[38] D. Zheng, C. Peng, X. Liao, L. Tian, G. Luo, and X. Cao, "Towards latency optimization in hybrid service function chain composition and embedding," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Toronto, ON, Canada, 2020, pp. 1539–1548.
[39] X. Zhang, Z. Xu, L. Fan, S. Yu, and Y. Qu, "Near-optimal energy-efficient algorithm for virtual network function placement," *IEEE Trans. Cloud Comput.*, early access, Oct. 15, 2019, doi: 10.1109/TCC.2019.2947554.
[40] X. Zhang, R. Zhou, Z. Zhou, J. C. S. C. S. Lui, and Z. Li, "An online learning-based task offloading framework for 5G small cell networks," in *Proc. 49th Int. Conf. Parallel Process. (ICPP)*, 2020, pp. 1–11.

Jing Li received the B.Sc. degree (First Class Hons.) in computer science from the Australian National University in 2018, where he is currently pursuing the Ph.D. degree with the Research School of Computer Science. His research interests include mobile edge computing, network function virtualization, and combinatorial optimization.

Weifa Liang (Senior Member, IEEE) received the B.Sc. degree in computer science from Wuhan University, China, in 1984, the M.E. degree in computer science from the University of Science and Technology of China in 1989, and the Ph.D. degree from the Australian National University in 1998, where he is currently a Full Professor with the Research School of Computer Science. His research interests include design and analysis of energy efficient routing protocols for wireless ad hoc and sensor networks, Internet of Things, mobile edge computing, network function virtualization, and software-defined networking, design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory. He currently serves as an Associate Editor on the Editorial Board of IEEE TRANSACTIONS ON COMMUNICATIONS.



Yu Ma received the B.Sc. degree (First Class Hons.) in computer science from Australian National University in 2014, where he is currently pursuing the Ph.D. degree with the Research School of Computer Science. His research interests include software defined networking, Internet of Things, and social networking.