Digital Twin-Assisted, SFC-Enabled Service Provisioning in Mobile Edge Computing

Jing Li, Song Guo, *Fellow, IEEE*, Weifa Liang, *Senior Member, IEEE*, Quan Chen, *Member, IEEE*, Zichuan Xu, *Member, IEEE*, Wenzheng Xu, *Member, IEEE*, and Albert Y. Zomaya, *Fellow, IEEE*

Abstract—Mobile Edge Computing (MEC) has been identified as a desirable computing paradigm that provides efficient and effective services for various applications, while meeting stringent service delay requirements. Orthogonal to the MEC computing paradigm, Network Function Virtualization (NFV) technology is another enabling technology that provides the network resource management with great flexibility and scalability, where the instances of Virtual Network Functions (VNFs) are deployed in edge servers as Service Function Chains (SFCs) for SFC-enabled services. Although reliable service provisioning in MEC environments is fundamentally important, the deployed VNF instances usually are not reliable, which can be affected by their software implementation, their execution duration, the workload among edge servers, and so on. Empowered by digital twin techniques, the states of VNF instances can be maintained by their digital twins in a real-time manner and their reliability can be accurately predicted through their digital twins. In this paper, we study digital twin-assisted, SFC-enabled reliable service provisioning in MEC networks by exploiting the dynamics of VNF instance reliability. We concentrate on two novel optimization problems of reliable service provisioning: the service cost minimization problem, and the dynamic service admission maximization problem. We first show their NP-hardness. We then formulate an Integer Linear Program (ILP) solution, and devise an approximation algorithm with a constant approximation ratio for the service cost minimization problem. We third provide an ILP solution to the offline version of the dynamic service admission maximization problem. Built upon this offline ILP solution, we also develop an online algorithm with a provable competitive ratio for the problem, by adopting the primal-dual dynamic updating technique. We finally evaluate the performance of the proposed algorithms via simulations. Simulation results demonstrate that the proposed algorithms outperform their comparison benchmarks, and improve the performance of their comparison counterparts by no less than 10.2%.

Index Terms—Mobile edge computing, service function chain, digital twin, reliable service provisioning, approximation algorithm, online algorithm, resource allocation and optimization

1 Introduction

Facilitated by the advance of telecommunication technologies, the Mobile Edge Computing (MEC) paradigm is envisioned as a complement of cloud computing. It provides various delay-sensitive services to end users at the edge of core networks with small-scale cloud computing capabilities, thereby curtailing the service delay and alleviating the core network utilization [16], [27]. Network Function Virtualization (NFV) as an appealing solution in MEC now becomes prevalent in managing network resources with flexibility and scalability, through placing Virtual Network Functions (VNFs) over edge servers (cloudlets) [30], and chaining the deployed VNFs as Service Function Chains (SFCs) for various applications [34].

Provisioning reliable SFC-enabled services in MEC is important and challenging for 5 G and beyond 5 G networks [7], as the failure of any constituent VNF instance will render an SFC-enabled service invalid [34]. To improve the reliability of SFC-enabled services, the adoption of service redundancy has become a mainstream method, where multiple backup VNF instances are mapped to the primary VNF instance of each service for swift recovery, and these backups usually remain in a standby mode until the primary one failure [1]. Most existing studies assumed that the reliability of each VNF is fixed when it is deployed in MEC networks [1], [5], [7], [12], [13], [18], [19], [20], [32], [34]. However, the reliability of any VNF instance is not always

- *Jing Li and Song Guo are with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, China. E-mail: jing.li5@anu.edu.au, song.guo@polyu.edu.hk.*
- *Weifa Liang is with the Department of Computer Science, City University of Hong Kong, Hong Kong, China. E-mail: weifa.liang@cityu.edu.hk.*
- *Quan Chen is with the School of Computers, Guangdong University of Technology, Guangzhou 510006, China, and also with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, China. E-mail: quan.c@gdut.edu.cn.*
- *Zichuan Xu is with the School of Software, Dalian University of Technology, Dalian 116024, China. E-mail: z.xu@dlut.edu.cn.*
- *Wenzheng Xu is with the College of Computer Science, Sichuan University, Chengdu 610017, China. E-mail: wenzheng.xu@scu.edu.cn.*
- *Albert Y. Zomaya is with the School of Computer Science, University of Sydney, Camperdown, NSW 2006, Australia. E-mail: albert.zomaya@sydney.edu.au.*

Manuscript received 18 August 2022; revised 26 October 2022; accepted 1 December 2022. Date of publication 7 December 2022; date of current version 5 December 2023.
The work by Jing Li and Song Guo was supported in part by Hong Kong RGC Research Impact Fund (RIF) under Grant R5060-19, in part by General Research Fund (GRF) under Grants 152221/19E and 15220320/20E, in part by the National Natural Science Foundation of China under Grant 61872310, and in part by Shenzhen Science and Technology Innovation Commission under Grant R2020A045. The work by Weifa Liang was supported by the City University of Hong Kong under Grant 9380137/CS. The work of Zichuan Xu was supported by the National Natural Science Foundation of China under Grant 61802048 and the "Xinghai Scholar Program" in Dalian University of Technology, China. The work by Wenzheng Xu was supported by NSFC under Grant 61602330.
(Corresponding author: Weifa Liang.)
Digital Object Identifier no. 10.1109/TMC.2022.3227248

fixed, which is determined by its deployment environment such as the workload of its host edge server, and its implementation software [7], [27]. Furthermore, the reliability of a VNF instance for different user requests may also vary, because it may be affected by many factors such as the execution duration of such a request, the issuing time of the request (the peak versus off-peak period), user locations and mobility.

Recently, the emerging digital twin technique offers a promising way to capture the system dynamics and furnish future insights by integrating data analytics and machine learning methods [8]. The digital twin applications have received increasing attention for achieving sustainability with accurate failure prediction in a plethora of areas, such as industrial maintenance [24], aeronautics and astronautics [29], and reliable service provisioning in MEC. Each VNF instance is mirrored by a digital twin in an MEC network, and the digital twin works as the virtual representation of the VNF instance [4]. Digital twins can grasp the states of VNF instances running in an MEC network in realtime, by continuously monitoring and creating vivid virtual simulation scenarios [8]. Through simulating the lifecycle of each VNF instance by its digital twin, it can predict the potential failures of the VNF instance, thereby enabling to accurately predict the reliability of the VNF instance [4].

In this paper, we focus on digital twin-assisted, SFC-enabled service provisioning in MEC environments, where the digital twin technique is adopted for the prediction of the reliability of VNF instances in a real-time manner. Specifically, in this paper we need to address the following questions. First, for each service request, how many backup VNF instances do we need to minimize the computing resource consumption of the service request to meet its reliability requirement? Second, considering limited computing resource in MEC, how to deploy different VNF instances to cloudlets so that the total admission cost of all user requests is minimized? Finally, how to develop a smart request admission strategy so that as many requests as possible can be admitted while their service reliability is still met?

The novelties of this work lie in the exploration of dynamic reliability issues on VNF instance placement for different service requests in MEC networks by leveraging digital twin techniques to predict the reliability of VNF instances. Two novel optimization problems of digital twin-assisted, SFC-enabled service provisioning in an MEC network are formulated under the settings of static and dynamic service request arrivals, and efficient approximation and online algorithms with guaranteed performance for the problems are then developed, respectively.

The main contributions of this paper are given as follows. We study the reliability dynamics in VNF instance deployments for user requests, by leveraging digital twin techniques. We study two novel SFC-enabled service provisioning problems through reliability-aware VNF placement: the service cost minimization problem, and the dynamic service admission maximization problem, respectively. We show the NP-hardness of the two defined problems. We first formulate an Integer Linear Program (ILP) solution to the service cost minimization problem when the problem size is small. Otherwise, we devise an approximation algorithm with a provable approximation ratio for the problem based on the ILP solution. We then propose an ILP solution for the offline version of the dynamic service admission maximization problem. We also develop an online algorithm with a competitive ratio for the dynamic service admission maximization problem by adopting the primal-dual dynamic updating technique. We finally evaluate the performance of the proposed algorithms for digital twin-assisted, SFC-enabled service provisioning in MEC via simulations. The simulation results show that the proposed algorithms outperform their corresponding benchmarks, improving the algorithm performance by no less than 10.2% in comparison with those of their benchmarks.

The rest of this paper is arranged as follows. The related work on digital twin-assisted, SFC-enabled service provisioning in MEC is surveyed in Section 2. Section 3 includes the problem definitions and the NP-hardness proofs of both defined problems. Section 4 develops an approximation algorithm for the service cost minimization problem, and Section 5 proposes an online algorithm for the dynamic service admission maximization problem. Section 6 evaluates the performance of the proposed algorithm. The conclusion of the paper is provided in Section 7.

2 Related Work

With the advance of Network Function Virtualization (NFV) and Mobile Edge Computing (MEC) techniques, SFC-enabled reliable service provisioning in MEC networks has attracted lots of attention in past years [5], [6], [12], [13], [22]. For instance, Huang et al. [5] studied how to provide reliable VNF services in MEC, and devised performance-guaranteed approximation algorithms under different network resource assumptions. Ishigaki et al. [6] developed a recovery technique by adopting a Deep Reinforcement Learning (DRL) method to handle random failures in providing VNF services. Li et al. [12], [13] proposed online algorithms for reliable VNF service provisioning in MEC networks with the aim to maximize the accumulative revenue of the service provider. Li et al. [17] also paid attention to Internet of Things (IoT) service provisioning with multiple data sources of an IoT application, and proposed efficient approximation and online algorithms for minimizing the total service cost or maximizing the number of user requests admitted. Although the investigation in [17] dealt with both static and dynamic service requests, neither SFC nor reliability requirements were mentioned for such service request admissions. Martín-Peréz et al. [22] addressed the efficient VNF placement and data routing in MEC to minimize the total cost while meeting all key performance indicators: service latency, geographical availability, and reliability requirements, and developed efficient heuristics for problems.

Several recent studies investigated reliable SFC-enabled service provisioning in MEC networks [1], [18], [19], [20], [30]. For instance, Alleg et al. [1] designed a redundancy mechanism, where each VNF instance is split into multiple thinner active instances, and multiple backups are deployed to improve the reliability of the SFC. They also provided a Mixed Integer Linear Program (MILP) formulation based on the redundancy mechanism to minimize the total service cost. Liang et al. [18], [19] considered reliable service augmentation problems by proposing efficient approximation and heuristic algorithms to maximize the total SFC-enabled service reliability of all service requests. Lin et al. [20] designed an efficient approximation algorithm to provide reliable SFC services in MEC to maximize SFC-enabled request admissions, while meeting the reliability demands of requests. Wang et al. [30] investigated uncertain demand levels of VNFs and workload balancing among edge servers, and devised an online learning algorithm to maximize the average SFC backup hit.

It is noticed that there are only a handful of studies considering the dynamics of VNF instance reliability of SFC-enabled service provisioning in MEC [8], [14], [15], [27], [32]. For instance, Karimzadeh-Farshbafan et al. [8] formulated a Markov Decision Process (MDP) model to capture the dynamic arrivals and departures of SFC service requests with reliability requirements, and developed a heuristic algorithm to alleviate the total request implementation cost while maximizing the number of requests admitted. Li et al. [15] addressed the robust SFC placement problem, incorporating the computing resource and data rate uncertainty assumptions, and developed an approximate solution, based on the Markov approximation. Shang et al. [27] designed a self-adapting scheme, by jointly deploying static and dynamic VNF backups in edge and cloud servers to deal with dynamic failures of VNFs for minimizing the backup deployment cost. Yang et al. [32] devised a Reinforcement Learning (RL) method to deal with random arrivals of SFC-enabled requests with the objective of maximizing the number of requests admitted while meeting both reliability and delay requirements of admitted requests. Recently, digital twin techniques have been successfully adopted to monitor the system dynamics [4], [10], [24], [28], [29]. For instance, the authors in [10] leveraged digital twins to explore the dynamics of mobile users, resource devices and UAVs, while the authors in [28] captured the dynamic states of both edge servers and the entire MEC network via the assistance of their digital twins.

In contrast to the aforementioned works, in this paper we investigate digital twin-assisted, reliable SFC service provisioning in MEC networks under both static and dynamic service request arrivals. We consider the dynamic reliability of VNF instances by leveraging digital twin techniques to provide accurate reliability prediction of VNFs for each user request. This journal version is an extension of a recent conference paper [11]. In addition to dealing with the dynamic service admission maximization problem [11], we also consider the service cost minimization problem for a set of user requests and develop a constant approximation algorithm for the problem in this extended version.

3 Preliminaries

In this section, we first formulate a system model and introduce notions and notations. We then define two optimization problems precisely. We finally show the NP-hardness of the two defined problems.

3.1 System Model

Consider an MEC network $G = (V \cup \{v_0\}, E)$ with a set V of Access Points (APs), the remote cloud v_0, and a set E of links connecting APs. Each AP is assumed to be connected to its co-located cloudlet by an optical fiber cable. For better readability, we adopt notation $v \in V$ to denote an AP in the MEC network or its co-located cloudlet. Let cap_v be the available amount of computing resource in cloudlet $v \in V$. The unit cost of its computing resource for cloudlet $v \in V$ is denoted by $cost_v$, while the remote cloud v_0 is assumed to possess unlimited computing and storage resources.

We assume the corresponding digital twins of all VNF instances are stored at a remote cloud [21], which are used for dynamically predicting the reliability of VNFs in the admissions of user service requests. Denote by F the set of types of VNFs offered within the MEC network G and $c(f)$ the demanded amount of computing resource for deploying a VNF instance of $f \in F$.

3.2 User Service Requests With Reliability Requirements

Let U be the set of requests, and each request $u \in U$ can be expressed by a tuple $\langle SC_u, R_u \rangle$, where SC_u is the SFC requirement and R_u is its reliability requirement, assuming that $f_{u,i} \in \mathcal{F}$ is the ith VNF in SFC SC_u. We assume that the instance for each VNF in the SFC for each request is referred to as the *primary VNF instance* of its corresponding function. We also assume that each primary VNF instance may have up to K backups across cloudlets [5], where K (≥ 1) is a given integer. Then we further assume the reliability requirement of each request $u \in U$ can be satisfied by deploying up to K backups for each primary VNF instance.

We adopt digital twin techniques to dynamically provide accurate and personalized reliability prediction for VNF instances of each request. Denote by $r_{u,i}$ the predicted reliability of VNF $f_{u,i}$ during the execution of request $u \in U$.

Suppose the primary VNF instance of VNF $f_{u,i}$ is deployed in a cloudlet with the reliability of $r_{u,i}$, and it has $(L-1)$ backups deployed in cloudlets, respectively. Then the reliability $R(f_{u,i})$ of VNF $f_{u,i}$ is obtained as follows.

$$R(f_{u,i}) = 1 - (1 - r_{u,i})^L. \quad (1)$$

The reliability $R(SC_u)$ of SFC SC_u of user request u can be calculated as follows.

$$R(SC_u) = \prod_{f_{u,i} \in SC_u} R(f_{u,i}). \quad (2)$$

To meet the reliability requirement R_u of request $u \in U$, we have

$$R(SC_u) \geq R_u. \quad (3)$$

3.3 Problem Definitions

We first consider a snapshot of arrived requests in the MEC network, where there is a set U of requests with SFC and reliability requirements. We then deploy VNF instances in cloudlets for each request to meet its reliability requirement, by predicting the dynamic reliability of VNF instances via their digital twins. We further assume that cloudlets in the MEC network possess sufficient computing resource for

TABLE 1
Table of Symbols

Notations	Descriptions
$G = (V \cup \{v_0\}, E)$	An MEC network as an undirected graph $G = (V \cup \{v_0\}, E)$, where V is a set of Access Points (APs), v_0 is the remote cloud, and E is a set of links connecting APs
v	An AP in the MEC network or its co-located cloudlet with $v \in V$
cap_v and $cost_v$	The available amount and the unit cost of computing resource in cloudlet $v \in V$
F	The set of types of VNFs offered within the MEC network G
$c(f)$	The demanded amount of computing resource for deploying a VNF instance of $f \in F$
U and u	A set of user service requests and a user service request
SC_u and R_u	The SFC requirement and the reliability requirement of request $u \in U$
$f_{u,i}$ and $r_{u,i}$	The ith VNF in SFC SC_u and the predicted reliability of VNF $f_{u,i}$ during the execution of request u by digital twin
K	Each primary VNF instance may have up to K backups across cloudlets, with K (≥ 1) a given integer
$R(f_{u,i})$ and $R(SC_u)$	The reliability of of VNF $f_{u,i}$ with $(L-1)$ backups, and the achieved reliability of SFC SC_u of user request u
$\mathcal{R}(f_{u,i}, k)$	The reliability $\mathcal{R}(f_{u,i}, k)$ of VNF $f_{u,i}$ with k (≥ 0) backups
$H(u,i,k)$ and $H'(u)$	The utility gain $H(u,i,k)$ for the kth backup of VNF $f_{u,i}$ defined in Eq. (7), and $H'(u)$ is defined in Eq. (9)
$x_{u,i,k,v}$	Binary decision variable, indicating whether the kth backup of VNF $f_{u,i}$ is placed in cloudlet v, and we assume the primary VNF instance as a backup with $k = 0$
$y_{u,b}$	Binary decision variable, indicating whether backup $b \in B_u$ of request $u \in U$ is deployed in a cloudlet
z_u	Binary decision variable, indicating whether user request u is admitted

VNF instance deployment of all requests in U. An optimization problem is formulated to minimize the total service cost of requests for VNF instance deployment among cloudlets.

Definition 1. *Given an MEC network $G = (V \cup \{v_0\}, E)$ and the set U of user requests with each having SFC and reliability requirements, digital twins run in the remote cloud to provide reliability prediction of VNF instances for each request in real-time.* The service cost minimization problem *of request admissions in G is to minimize the total service cost of deploying VNF instances to cloudlets for all user requests, while meeting their SFC and reliability requirements, subject to the computing capacity on each cloudlet in G.*

We then assume that a sequence of incoming requests arrives one by one without the knowledge of future arrivals, and we aim to maximize the number of requests admitted. With dynamically predicting the reliability of each VNF instance by its digital twin, we need to determine whether to admit or reject an incoming request immediately. A request with a higher reliability requirement is very likely to consume more computing resource than a request with a lower reliability requirement in order to meet its reliability requirement. Also, the dynamic reliability issue of VNF instances affects the computing resource consumption for each user service request to meet the reliability requirement of the request. We thus deal with dynamic service request admissions as follows.

Definition 2. *Given an MEC network $G = (V \cup \{v_0\}, E)$ and a sequence U of user requests arriving one by one without the knowledge of future arrivals, digital twins of VNF instances run in the remote cloud that can provide its reliability prediction in real-time, assuming each incoming request is either admitted or rejected immediately.* The dynamic service admission maximization problem *is to maximize the number of requests admitted while meeting the SFC and reliability requirements of each admitted request, subject to the computing capacity on each cloudlet in G.*

For the sake of convenience, the symbols used in this paper are summarized in Table 1.

3.4 NP-Hardness Proofs of the Defined Problems

Theorem 1. *The service cost minimization problem in an MEC network $G = (V \cup \{v_0\}, E)$ is NP-hard.*

Proof. We show the NP-hardness of the service cost minimization problem via polynomial reduction from an NP-hard problem - the minimum-cost Generalized Assignment Problem (GAP) [25]. Given M bins and N items, each bin m is associated with a capacity $C(m)$, and packing item n into bin m leads to a cost $cost(n, m)$ with a size $size(n)$. The minimum-cost GAP is to minimize the total cost of packing all items into bins, subject to the capacities on bins.

We consider a special service cost minimization problem where the SFC of each request $u \in U$ consists of a single VNF f_u only. We calculate the number of VNF instances needed to meet the reliability requirement of request u. Let F' be the set of VNF instances for all requests in U. Let each cloudlet $v \in V$ represent a bin associated with a capacity cap_v, $\forall v \in V$, and each VNF instance $f \in F'$ represent an item associated with a size $c(f)$, i.e., the computing resource consumption of f. The cost of placing item f into bin v is $c(f) \cdot cost_v$, i.e., the cost of deploying a VNF instance of f in cloudlet v. This special problem is to minimize the total service cost of deploying VNF instances in F' for all requests to cloudlets. It can be seen that this special problem is equivalent to the minimum-cost GAP, and the latter is NP-hard [25]. Therefore, the service cost minimization problem is NP-hard too, and this theorem follows. □

Theorem 2. *The dynamic service admission maximization problem in an MEC network $G = (V \cup \{v_0\}, E)$ is NP-hard.*

Proof. Consider a special case of the dynamic service admission maximization problem, where all requests are given in advance, and assume when an item f is packed into a bin v, a profit of 1 will be collected. It can be seen that this special problem is an instance of a maximum-profit GAP, which is NP-hard [25]. Therefore, the theorem follows. □

4 Approximation Algorithm for the Service Cost Minimization Problem

In this section, We first provide an Integer Linear Program (ILP) formulation for the service cost minimization problem. We devise an approximation algorithm with a provable approximation ratio for the problem. We finally analyze the approximation ratio and time complexity of the proposed approximation algorithm.

4.1 ILP Formulation

With regard to the reliability requirement R_u of user request $u \in U$, by Eq. (2) and Ineq. (3), we have

$$\prod\nolimits_{f_{u,i} \in SC_u} R(f_{u,i}) \geq R_u, \tag{4}$$

which is equivalent to

$$\sum\nolimits_{f_{u,i} \in SC_u} \log_2 R(f_{u,i}) \geq \log_2 R_u. \tag{5}$$

Recall that we assume each primary VNF instance can have up to K backups, where K (≥ 1) is a fixed positive integer. Let $x_{u,i,k,v}$ be a binary variable with $u \in U$, $f_{u,i} \in SC_u, k \in [0, K]$ and $v \in V$, where $x_{u,i,k,v} = 1$ means placing the kth backup of VNF $f_{u,i}$ in cloudlet v, and we assume the primary VNF instance as a backup with $k = 0$; otherwise, $x_{u,i,v,k} = 0$.

From Eq. (1), the reliability $\mathcal{R}(f_{u,i}, k)$ of VNF $f_{u,i}$ with k (≥ 0) backups is calculated as follows.

$$\mathcal{R}(f_{u,i}, k) = 1 - (1 - r_{u,i})^{k+1}. \tag{6}$$

We now define the *utility gain* $H(u, i, k)$ for the kth backup of VNF $f_{u,i}$ as follows.

$$H(u,i,k) = \begin{cases} \log_2 \mathcal{R}(f_{u,i}, k) - \log_2 \mathcal{R}(f_{u,i}, k-1) & k \geq 1 \\ \log_2 r_{u,i} & k = 0 \end{cases} \tag{7}$$

It can be seen that $\sum_{k'=1}^{k} H(u, i, k') + \log_2 r_{u,i} = \log_2 \mathcal{R}(f_{u,i}, k)$. By Ineq. (5), we have the following reliability constraint.

$$\sum_{f_{u,i} \in SC_u} \sum_{k'=1}^{k} H(u, i, k') \geq \log_2 R_u - \log_2 \prod_{f_{u,i} \in SC_u} r_{u,i}, \tag{8}$$

where $\prod_{f_{u,i} \in SC_u} r_{u,i}$ is the reliability of SC_u for request u with only the primary VNF instances.

For notation simplicity, we define $H'(u)$ as follows.

$$H'(u) = \log_2 R_u - \log_2 \prod_{f_{u,i} \in SC_u} r_{u,i}. \tag{9}$$

We then have the following reliability constraint.

$$\sum_{f_{u,i} \in SC_u} \sum_{k'=1}^{k} H(u, i, k') \geq H'(u). \tag{10}$$

Denote by **P1** the service cost minimization problem. Its ILP formulation is given as follows.

$$\textbf{P1}: \text{Minimize} \sum_{u \in U} \sum_{f_{u,i} \in SC_u} \sum_{k=0}^{K} \sum_{v \in V} c(f_{u,i}) \cdot cost_v \cdot x_{u,i,k,v}, \tag{11}$$

subject to: Eq. (7), Eq. (9),

$$\sum_{f_{u,i} \in SC_u} \sum_{k=1}^{K} \sum_{v \in V} H(u,i,k) \cdot x_{u,i,k,v} \geq H'(u), \quad \forall u \in U \tag{12}$$

$$\sum_{u \in U} \sum_{f_{u,i} \in SC_u} \sum_{k=0}^{K} c(f_{u,i}) \cdot x_{u,i,k,v} \leq cap_v, \quad \forall v \in V \tag{13}$$

$$\sum_{v \in V} x_{u,i,0,v} = 1, \quad \forall u \in U, \ \forall f_{u,i} \in SC_u \tag{14}$$

$$\sum_{v \in V} x_{u,i,k,v} \leq 1, \quad \forall u \in U, \forall f_{u,i} \in SC_u, \forall k \in [1, K] \tag{15}$$

$$x_{u,i,k,v} \in \{0, 1\}, \forall u \in U, \forall f_{u,i} \in SC_u, \forall k \in [0, K], \forall v \in V, \tag{16}$$

where Constraint (12) is the reliability constraint of each request $u \in U$ shown in Ineq. (10), and $H(u, i, k)$ and $H'(u)$ are defined in Eqs. (7) and (9), respectively. Constraint (13) means the capacity of no cloudlet is violated. Constraint (14) shows that the primary VNF instances must be deployed on exactly one cloudlet $v \in V$. Constraint (15) shows that each backup can be deployed on at most one cloudlet. Constraint (16) shows that the binary decision variable $x_{u,i,k,v}$ indicates whether the kth backup of VNF $f_{u,i}$ is placed in cloudlet v, and the 0th backup is the primary VNF instance.

4.2 Approximation Algorithm

The general strategy for the service cost minimization problem is to reduce the problem (denoted by **P1**) to another problem (denoted by **P2**), and an approximate solution to problem **P1** in turn returns from an approximate solution to problem **P2**. Considering a set U of admitted requests, problem **P2** is to minimize the total computing resource consumption of deploying VNF instances in cloudlets for admitted requests, while meeting their reliability requirements. Note that the computing capacity constraint (13) on problem **P1** is ignored for the time being.

Recall that each primary VNF instance can be equipped with up to K backups with $K \geq 1$. Let B_u be the set of all potential backups for request $u \in U$ with $|B_u| = K \cdot |SC_u|$, where SC_u is the SFC of u. Denote by $c(b)$ the computing resource consumption of backup $b \in B_u$. Recall that $H(u, i, k)$ by Eq. (7) is the utility gain for the kth backup of VNF $f_{u,i}$. Let $H(b)$ be the utility gain of backup $b \in B_u$. Let $y_{u,b} = 1$ be a binary decision variable indicating that backup $b \in B_u$ of request $u \in U$ is deployed in a cloudlet; otherwise, $y_{u,b} = 0$. Note that the primary VNF instances must be deployed in cloudlets, and $\sum_{u \in U} \sum_{f_{u,i} \in SC_u} c(f_{u,i})$ is the total computing resource consumption by deploying all primary VNF instances for requests.

Problem **P2** is formulated as follows.

$$\textbf{P2}: \text{Minimize} \sum_{u \in U} \sum_{f_{u,i} \in SC_u} c(f_{u,i}) + \sum_{u \in U} \sum_{b \in B_u} c(b) \cdot y_{u,b}, \tag{17}$$

subject to: Eq. (7), Eq. (9),

$$\sum_{b \in B_u} H(b) \cdot y_{u,b} \geq H'(u), \quad \forall u \in U \tag{18}$$

$$y_{u,b} \in \{0, 1\}, \quad \forall u \in U, \ \forall b \in B_u, \tag{19}$$

where the objective function (17) means to minimize the total computing resource consumption of both primary and backup VNF instances for requests. Constraint (18) is the reliability constraint of requests based on Constraint (12).

Let $\gamma_b = \frac{c(b)}{H(b)}$ be the ratio of a backup b, where $c(b)$ is its computing resource consumption and $H(b)$ is the utility gain of deploying backup b by Eq. (7). For each request u, we first sort its potential backups B_u in non-decreasing order of the ratio γ_b. For notation simplicity, we assume that $B_u = \{b_1, b_2, \ldots, b_{|B_u|}\}$ with $\gamma_{b_1} \leq \gamma_{b_2} \leq \gamma_{b_3} \leq \ldots \leq \gamma_{b_{|B_u|}}$.

We propose an approximate solution to problem **P2** by building a VNF instance set $\mathbb{F}$ for all requests in U as follows. Initially, set $\mathbb{F}$ is empty. For each request $u \in U$, we first add its primary VNF instances into set $\mathbb{F}$, i.e., $\mathbb{F} = \mathbb{F} \cup SC_u$. If its reliability requirement is met, i.e., $H'(u) \leq 0$, no backup is needed; otherwise ($H'(u) > 0$), we deploy backups B_u one by one in non-decreasing order of the ratio γ_b until meeting its reliability constraint (18). Especially, we first deploy backup b_1 and let $\mathbb{F} = \mathbb{F} \cup \{b_1\}$. If its reliability requirement is met (i.e., $H(b_1) \geq H'(u)$), this is done. Otherwise ($H(b_1) < H'(u)$), we deploy backups b_1 and b_2 by letting $\mathbb{F} = \mathbb{F} \cup \{b_1, b_2\}$, and examine whether request u has its SFC reliability satisfied or not. The procedure continues until we deploy a backup b_j with $1 \leq j \leq |B_u|$ and $\sum_{1 \leq j' \leq j} H(b_{j'}) \geq H'(u)$, i.e., the reliability requirement of u is met. Note that we assume the reliability requirement of each request $u \in U$ can be satisfied with no more than K backups deployed for each primary VNF instance.

Algorithm 1. An Approximation Algorithm for Problem **P2**.

Input: The MEC network $G = (V \cup \{v_0\}, E)$, the set U of requests with reliability requirements, and digital twins running in the remote cloud to provide reliability prediction of VNF instances for requests in real-time.

Output: Minimize the total computing resource consumption of deploying VNF instances for admitted requests, while meeting their reliability requirement.

```
F ← ∅; /* the solution */
for each user request u ∈ U do
  F ← F ∪ SC_u;
  if H'(u) > 0 then
    Sort potential backups in B_u in non-decreasing order
    of γ_{b_j} (= c(b_j)/H(b_j)) with 1 ≤ j ≤ |B_u|;
    for each backup b_j ∈ B_u in the sorted order do
      F ← F ∪ {b_j};
      if Σ_{1≤j'≤j} H(b_{j'}) ≥ H'(u) then
        break;
      end if;
    end for;
  end if;
end for ;
return The VNF instance set F for Problem P2.
```

The detailed approximation algorithm for problem **P2** is given in `Algorithm` 1.

We claim that compared with the optimal solution to problem **P1**, it consumes at most c_{max} more computing resource for each request $u \in U$ to meet its reliability requirement, by deploying VNF instances in $\mathbb{F}$, where $c_{max} = \max_{f \in F}\{c(f)\}$ indicates the maximum amount of demanded computing resource of any VNF. This claim will be shown later in Lemma 4. Note that the computing resource consumption for each request u is at least $\sum_{f_{u,i} \in SC_u} c(f_{u,i})$ in problem **P1**, because of deploying his primary VNF instances. Recall that we assume that the cloudlets V possess sufficient computing resource to accommodate all requests U while meeting their reliability requirements. Then we further assume it exists sufficient computing resource among cloudlets V for placing VNF instances from $\mathbb{F}$.

We now tackle problem **P1** by adopting a greedy strategy. We deploy VNF instances from $\mathbb{F}$ to cloudlets in V for user requests in U, where the VNF instances in $\mathbb{F}$ are sorted in non-increasing order of the cost of their computing resource consumption. For notation simplicity, let $f_1, f_2, \ldots, f_{|\mathbb{F}|}$ be the sorted VNF instances with $c(f_1) \geq c(f_2) \geq \ldots \geq c(f_{|\mathbb{F}|})$. Meanwhile, let $v_1, v_2, \ldots, v_{|V|}$ be the list of sorted cloudlets in non-decreasing order by the unit cost of computing resource, i.e., $cost_{v_1} \leq cost_{v_2} \leq \ldots \leq cost_{v_{|V|}}$. For each VNF instance $f \in \mathbb{F}$ in the sorted order, we consider deploying f into a cloudlet $v \in V$ by the sorted order, with sufficient residual computing resource for f. The procedure continues until all VNF instances are deployed. Because cloudlets are assumed to possess sufficient computing resource for deploying all VNF instances in $\mathbb{F}$, there will be no resource violations on cloudlets.

The detailed approximation algorithm for the service cost minimization problem (problem **P1**) is given in `Algorithm` 2.

Algorithm 2. An Approximation Algorithm for the Service Cost Minimization Problem (Problem **P1**).

Input: The MEC network $G = (V \cup \{v_0\}, E)$, the set U of requests with reliability requirements, and digital twins running in the remote cloud to provide reliability prediction of VNF instances for requests in real-time.

Output: Minimize the total cost of deploying VNF instances in cloudlets V for each admitted requests, while meeting their reliability requirements.

```
Construct an instance of problem P2, and obtain VNF
instance set F for problem P2 by Algorithm 1;
Sort the VNF instances in F in non-increasing order based
on their computing resource demands;
Sort cloudlets in V with non-decreasing order of their unit
cost of computing resource;
for each sorted VNF instance f ∈ F do
  for each cloudlet v ∈ V in its sorted order do
    if cloudlet v possesses sufficient computing resource
    for VNF instance f then
      Deploy VNF instance f to cloudlet v and update the
      residual computing resource of cloudlet v
    end if;
  end for;
end for;
return The deployment of the VNF instance set F.
```

4.3 Algorithm Analysis

Lemma 1. *For any VNF $f_{u,i}$, (1) $H(u,i,k) > 0$ with $1 \leq k \leq K$; and (2) $H(u,i,k) > H(u,i,k')$ with $1 \leq k < k' \leq K$, where $H(u,i,k)$ is defined in Eq. (7).*

Proof. (1) By Eqs. (6) and (7), for $1 \leq k \leq K$,

$$H(u,i,k) = \log_2 \mathcal{R}(f_{u,i},k) - \log_2 \mathcal{R}(f_{u,i},k-1) = \log_2 \frac{1-(1-r_{u,i})^{k+1}}{1-(1-r_{u,i})^{k}}. \quad (20)$$

Since $0 < r_{u,i} < 1$, then $H(u,i,k) > 0$ with $1 \leq k \leq K$. (2) By Eqs. (7) and (20), for $1 \leq k < k' \leq K$, then

$$H(u,i,k) - H(u,i,k') = \log_2 \frac{1-(1-r_{u,i})^{k+1}}{1-(1-r_{u,i})^{k}} - \log_2 \frac{1-(1-r_{u,i})^{k'+1}}{1-(1-r_{u,i})^{k'}}. \quad (21)$$

Let $\rho = 1 - r_{u,i}$, then $0 < \rho < 1$ since $0 < r_{u,i} < 1$. As $\log_2(x)$ is a monotonically increasing function with $x > 0$, we compare the values of $\frac{1-\rho^{k+1}}{1-\rho^{k}}$ and $\frac{1-\rho^{k'+1}}{1-\rho^{k'}}$, and we have

$$\frac{1-\rho^{k+1}}{1-\rho^{k}} - \frac{1-\rho^{k'+1}}{1-\rho^{k'}} = \frac{(1-\rho)\cdot(\rho^{k}-\rho^{k'})}{(1-\rho^{k})\cdot(1-\rho^{k'})} > 0, \quad (22)$$

because $0 < \rho < 1$ and $k < k'$.

We then have $\frac{1-\rho^{k+1}}{1-\rho^{k}} > \frac{1-\rho^{k'+1}}{1-\rho^{k'}}$, which implies that $H(u,i,k) > H(u,i,k')$. □

Lemma 2. *There is an optimal solution for the service cost minimization problem by the ILP formulation (11).*

Proof. The ILP solution may deliver a solution, where $\sum_{v \in V} x_{u,i,k,v} = 0$ and $\sum_{v \in V} x_{u,i,k,'v} = 1$ with $1 \leq k < k' \leq K$, i.e., given a VNF $f_{u,i}$, its k'th backup has already been deployed in a cloudlet while its kth backup has not been deployed in any cloudlet yet.

Because the amount of demanded computing resource by a backup for VNF $f_{u,i}$ is $c(f_{u,i})$, we can always replace its k'th backup with its kth backup, without changing the objective value or violating the computing capacity constraint (13). Following Lemma 1, $H(u,i,k) > H(u,i,k') > 0$. Therefore, the reliability constraint (12) is not violated because such a replacement leads to a larger value in the left-hand side of Constraint (12). □

Lemma 2 demonstrate that the ILP solution (11) is an optimal solution to the service cost minimization problem (problem **P1**), which will be treated as a benchmark in the evaluation of the algorithm performance.

Lemma 3. *Given an MEC network $G = (V \cup \{v_0\}, E)$ and a set U of admitted requests with each having SFC and reliability requirements without the computing capacity constraints on cloudlets in V, there is an approximation algorithm,* `Algorithm` *1, with the approximation ratio of $(1 + \frac{c_{max}}{|SC|_{min} \cdot c_{min}})$ for problem* ***P2****, which takes $O(|U| \cdot |SC|_{max} \cdot K \cdot \log_2(|SC|_{max} \cdot K))$ time, where c_{max} and c_{min} are the maximum and minimum computing resource consumptions of any VNF instance, $|SC|_{max}$ and $|SC|_{min}$ are the maximum and minimum lengths of any SFC, and K is the maximum number of backups for each primary VNF instance.*

Proof. Because we ignore the computing capacity constraints on cloudlets in problem **P2**, the backup deployment of request u does not impact the backup deployment of another request u'. We consider the Linear Program (LP) relaxation of problem **P2**, by relaxing the binary decision variable $y_{u,b} \in \{0,1\}$ to real variable $\widetilde{y}_{u,b}$ with $0 \leq \widetilde{y}_{u,b} \leq 1$, and $\widetilde{\mathbf{P2}}$ is denoted as the relaxation vision of **P2**. Because problems **P2** and $\widetilde{\mathbf{P2}}$ aim to minimize the total computing resource consumptions, it can be seen $\widetilde{OPT_2} \leq OPT_2$, where OPT_2 and $\widetilde{OPT_2}$ are the values of the optimal solutions for problems **P2** and $\widetilde{\mathbf{P2}}$, respectively.

Because we allow a fractional backup deployment for problem $\widetilde{\mathbf{P2}}$, for each request $u \in U$, it is equivalent to minimizing the ratio of the total computing resource consumption of request u to the utility requirement $H'(u)$ defined in Eq. (9). Recall that $\gamma_b = \frac{c(b)}{H(b)}$ with $b \in B_u$, where $c(b)$ is the demanded computing resource by backup b, and $H(b)$ is the utility gain by deploying backup b by Eq. (7), i.e., γ_b is the computing resource consumption for a unit utility gain of deploying backup b. Assuming that $B_u = \{b_1, b_2, \ldots, b_{|B_u|}\}$ is sorted in a non-decreasing order of their ratio γ_b with $\gamma_{b_1} \leq \gamma_{b_2} \leq \ldots \leq \gamma_{b_{|B_u|}}$.

We observe that the optimal solution to problem $\widetilde{\mathbf{P2}}$ is to deploy backups in B_u in sorted order fully or fractionally until meeting the reliability requirement of request u as follows. For each request $u \in U$, we deploy its primary VNF instances initially. If its reliability requirement can be met, i.e., $H'(u) \leq 0$, no backup is needed; otherwise ($H'(u) > 0$), we consider deploying backup b_1. If $H(b_1) \geq H'(u)$, we fractionally deploy b_1 by setting $\widetilde{y}_{u,b_1} = \frac{H'(u)}{H(b_1)}$, and done. Otherwise ($H(b_1) < H'(u)$), we fully deploy b_1 by setting $\widetilde{y}_{u,b_1} = 1$, and consider deploying the next backup b_2. Generally speaking, when deploying backup b_j with $1 \leq j \leq |B_u|$, if $\sum_{1 \leq j' \leq j} H(b_{j'}) < H'(u)$, we deploy b_j by setting $\widetilde{y}_{u,b_j} = 1$. Otherwise ($\sum_{1 \leq j' \leq j} H(b_{j'}) \geq H'(u)$), we fractionally deploy b_j by setting $\widetilde{y}_{u,b_j} = \frac{H'(u) - \sum_{1 \leq j' \leq j-1} H(b_{j'})}{H(b_j)}$ and set $\widetilde{y}_{u,b_{j'}} = 0$, $\forall j' > j$, i.e., its reliability requirement has been satisfied by deploying backups $\{b_1, \ldots, b_j\}$. We can see the optimal solution to problem $\widetilde{\mathbf{P2}}$ is $\{\widetilde{y}_{u,b} \mid \forall u \in U, \forall b \in B_u\}$, i.e., $\widetilde{OPT_2}$.

Let $\mathcal{A}_2$ be the solution for problem **P2** by `Algorithm` 1. `Algorithm` 1 deploys a backup for each $\widetilde{y}_{u,b} > 0$ in problem **P2**, while it exists only at most one backup $b \in B_u$ for each request $u \in U$ with $0 < \widetilde{y}_{u,b} < 1$ in problem $\widetilde{\mathbf{P2}}$. Therefore, each request $u \in U$ in the solution $\mathcal{A}_2$ consumes the amount of computing resource no more c_{max} than the amount of computing resource in the solution $\widetilde{OPT_2}$.

Because problem $\widetilde{\mathbf{P2}}$ is the relaxed vision of problem **P2**, then the value of $\mathcal{A}_2$ is no greater than $OPT_2 + c_{max} \cdot |U|$. We have $\mathcal{A}_2 \leq OPT_2 + |U| \cdot c_{max}$. We then have

$$\frac{\mathcal{A}_2}{OPT_2} \leq 1 + \frac{|U| \cdot c_{max}}{OPT_2}. \quad (23)$$

Because the primary VNF instances for each request $u \in U$ must be deployed already, we then have

$$OPT_2 \geq \sum_{u \in U} \sum_{f_{u,i} \in SC_u} c(f_{u,i}) \geq |U| \cdot |SC|_{min} \cdot c_{min}. \quad (24)$$

Thus, we have

$$\frac{\mathcal{A}_2}{OPT_2} \leq 1 + \frac{c_{max}}{|SC|_{min} \cdot c_{min}}. \quad (25)$$

The time complexity for Algorithm 1 is dominated by sorting potential backups for all requests $u \in U$, which is $O(|U| \cdot |SC|_{max} \cdot K \cdot \log_2(|SC|_{max} \cdot K))$. □

Lemma 4. *Compared with the optimal solution to the service cost minimization problem (problem **P1**), deploying VNF instances in $\mathbb{F}$ consumes at most c_{max} more computing resource for each request $u \in U$ to meet its reliability requirement, where $c_{max} = \max_{f \in \mathbb{F}}\{c(f)\}$ is the maximum computing resource consumption of any VNF instance.*

Proof. Let $\mathcal{A}_2$ be the value of the solution (or the solution itself if no confusion arises) for problem **P2** by Algorithm 1, i.e., the total demanded computing resource by VNF instances from $\mathbb{F}$. Let OPT_2 be the optimal solution to problem **P2**. We have shown that each request $u \in U$ in solution $\mathcal{A}_2$ consumes at most c_{max} more computing resource than it in $\widetilde{OPT}_2$ by Lemma 3. Also, we ignore computing capacity constraints on cloudlets in problem **P2**, and the amount of demanded computing resource by each request in the optimal solution to problem **P2** is no greater than that in the optimal solution to problem **P1**. □

Theorem 3. *Given an MEC network $G = (V \cup \{v_0\}, E)$, the set U of admitted requests with each having SFC and reliability requirements, there is an approximation algorithm, Algorithm 2, with an approximation ratio $\frac{cost_{max}}{cost_{min}} \cdot (1 + \frac{c_{max}}{|SC|_{min} \cdot c_{min}})$ for the service cost minimization problem. The algorithm takes $O(\mathcal{N} \cdot \log_2 \mathcal{N} + |V| \cdot \log_2 |V| + \mathcal{N} \cdot |V|)$ time, where $\mathcal{N} = |U| \cdot |SC|_{max} \cdot K$, $cost_{max}$ and $cost_{min}$ are the maximum and minimum unit computing resource costs on any cloudlet, respectively, c_{max} and c_{min} are the maximum and minimum computing resource consumption of any VNF instance, respectively, $|SC|_{max}$ and $|SC|_{min}$ are the maximum and minimum length of any SFC, respectively, K is the maximum number of backups deployed for a primary VNF instance, and V is the cloudlet set.*

Proof. Recall that problem **P1** (the service cost minimization problem) aims to minimize total cost, while problem **P2** aims to minimize total computing resource consumption. Denote by $\mathcal{A}_1$ the value of the approximate solution for problem **P1** by Algorithm 2. Let $\mathcal{A}_2$ be the value of the solution for problem **P2** by Algorithm 1. Because Algorithm 2 deploys the VNF instances from set $\mathbb{F}$ obtained by Algorithm 1 for problem **P1**, the solution delivered by Algorithm 2 for problem **P1** also leads to the total computing resource consumption $\mathcal{A}_2$. We then have

$$\mathcal{A}_1 \leq cost_{max} \cdot \mathcal{A}_2. \tag{26}$$

Let OPT_1 and OPT_2 be the values of the optimal solutions to problems **P1** and **P2**, respectively. Due to ignoring the computing capacity on each cloudlet in problem **P2**, we then have

$$OPT_1 \geq cost_{min} \cdot OPT_2. \tag{27}$$

By Eqs. (25), (26) and (27), we have

$$\frac{\mathcal{A}_1}{OPT_1} \leq \frac{cost_{max}}{cost_{min}} \cdot \frac{\mathcal{A}_2}{OPT_2} \leq \frac{cost_{max}}{cost_{min}} \cdot \left(1 + \frac{c_{max}}{|SC|_{min} \cdot c_{min}}\right). \tag{28}$$

The time complexity of Algorithm 2 is analyzed as follows. It takes $O(|U| \cdot |SC|_{max} \cdot K \cdot \log_2(|SC|_{max} \cdot K)))$ time to invoke Algorithm 1 to obtain the set $\mathbb{F}$ of VNF instances, by Lemma 3. Algorithm 2 then sorts the VNF instances in $\mathbb{F}$ and cloudlets in V respectively, which takes $O(|U| \cdot |SC|_{max} \cdot K \cdot \log_2(|U| \cdot |SC|_{max} \cdot K)) + |V| \cdot \log_2 |V|)$ time. The iterations for deploying VNF instances in $\mathbb{F}$ to cloudlets in V takes $O(|U| \cdot |SC|_{max} \cdot K \cdot |V|)$ time. Let $\mathcal{N} = |U| \cdot |SC|_{max} \cdot K$, then Algorithm 2 takes $O(\mathcal{N} \cdot \log_2 \mathcal{N} + |V| \cdot \log_2 |V| + \mathcal{N} \cdot |V|)$ time. □

5 Online Algorithm for the Dynamic Service Admission Maximization Problem

In this section, we investigate the dynamic service admission maximization problem without the knowledge of future request arrivals. We first formulate an Integer Linear Program (ILP) solution for the offline version of the problem. We then devise an online algorithm for the problem by the primal-dual dynamic updating technique [2]. We finally analyze the competitive ratio of the proposed online algorithm and its time complexity.

5.1 ILP Formulation

We assume the reliability requirement of request $u \in U$ can be satisfied by deploying at most K backups for each primary VNF instance, i.e., $\sum_{f_{u,i} \in SC_u} \sum_{k=1}^{K} H(u,i,k) \geq H'(u)$. Recall that $x_{u,i,k,v}$ serves as a binary decision variable indicating whether we deploy the kth backup of VNF $f_{u,i}$ in cloudlet v, and the 0th backup is the primary VNF instance. Let z_u be a binary decision variable for each $u \in U$, where $z_u = 1$ means that user request u is admitted; otherwise, user request u is rejected. Denote by **P3** the dynamic service admission maximization problem, and we give the ILP for the offline version of problem **P3**.

$$\mathbf{P3:} \quad \text{Maximize} \quad \sum\nolimits_{u \in U} z_u, \tag{29}$$

subject to: Eq. (7), Eq. (9),

$$\sum_{f_{u,i} \in SC_u} \sum_{k=1}^{K} \sum_{v \in V} H(u,i,k) \cdot x_{u,i,k,v} \geq H'(u) \cdot z_u, \ \forall u \in U \tag{30}$$

$$\sum_{u \in U} \sum_{f_{u,i} \in SC_u} \sum_{k=0}^{K} c(f_{u,i}) \cdot x_{u,i,k,v} \leq cap_v, \quad \forall v \in V \tag{31}$$

$$\sum_{v \in V} x_{u,i,0,v} = z_u, \quad \forall u \in U, \forall f_{u,i} \in SC_u \tag{32}$$

$$\sum_{v \in V} x_{u,i,k,v} \leq z_u, \ \forall u \in U, \forall f_{u,i} \in SC_u, \forall k \in [1,K] \tag{33}$$

$$x_{u,i,k,v} \in \{0,1\}, \forall u \in U, \forall f_{u,i} \in SC_u, \forall k \in [0,K], \forall v \in V \tag{34}$$

$$z_u \in \{0,1\}, \quad \forall u \in U, \tag{35}$$

where Constraint (31) means that the computing capacities of cloudlets are not violated. If request $u \in U$ is admitted ($z_u = 1$), Constraint (30) is its reliability constraint shown in Ineq. (10), Constraint (32) means each primary VNF instance is deployed on exactly one cloudlet, and Constraint (33) means each backup is deployed on at most one cloudlet. Otherwise (request u is rejected and $z_u = 0$), then $x_{u,i,k,v} = 0$, $\forall f_{u,i} \in SC_u$, $\forall k \in [0,K]$, and $\forall v \in V$, by Constraints (30), (32) and (33), i.e., each rejected request has no VNF instance deployed. Constraint (34) shows variable $x_{u,i,k,v}$ indicates whether the kth backup for VNF $f_{u,i}$ is deployed in cloudlet v. Constraint (35) shows variable z_u indicates whether to admit request u or not.

5.2 Online Algorithm

Denote by problem **P3** the dynamic service admission maximization problem. Let problem **P4** be the relaxed Linear Program (LP) of the offline version for problem **P3**. Let problem **P5** be the dual of problem **P4**. We can obtain an online solution to problem **P3**, based on a feasible online solution to problem **P5**. Especially, problem **P4** is

$$\mathbf{P4:}\quad \text{Maximize} \quad \sum_{u\in U} z_u, \tag{36}$$

subject to:

$$\text{Eq. (7) and (9), Constraint (30), (31), (32) and (33),}$$

$$z_u \le 1, \quad \forall u \in U \tag{37}$$

$$x_{u,i,k,v} \ge 0, \quad \forall u\in U,\ \forall f_{u,i}\in SC_u,\ \forall k\in[0,K],\ \forall v\in V \tag{38}$$

$$z_u \ge 0, \quad \forall u \in U, \tag{39}$$

where Constraints (37) and (39) show that the binary decision variable z_u is relaxed to a real number between 0 and 1. By Constraints (32), (33), and (37), we have $x_{u,i,k,v} \le 1$. Then the binary decision variable $x_{u,i,k,v}$ is also relaxed to a real number between 0 and 1 by Constraint (38).

Notice that problem **P5** is the dual of problem **P4**, which is expressed as follows.

$$\mathbf{P5:}\quad \text{Minimize} \quad \sum_{v\in V} cap_v \cdot \beta_v + \sum_{u\in U} \mu_u, \tag{40}$$

subject to: Eq. (7), Eq. (9),

$$H'(u)\cdot\alpha_u - \sum_{f_{u,i}\in SC_u}\sigma_{u,i} - \sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K}\lambda_{u,i,k} + \mu_u - 1 \ge 0, \quad \forall u\in U \tag{41}$$

$$c(f_{u,i})\cdot\beta_v + \sigma_{u,i} \ge 0, \quad \forall u\in U,\ \forall f_{u,i}\in SC_u,\ \forall v\in V \tag{42}$$

$$-H(u,i,k)\cdot\alpha_u + \beta_v\cdot c(f_{u,i}) + \lambda_{u,i,k} \ge 0, \quad \forall u\in U,\ \forall f_{u,i}\in SC_u, \forall k\in[1,K], \forall v\in V \tag{43}$$

$$\alpha_u \ge 0, \beta_v \ge 0, \lambda_{u,i,k} \ge 0, \mu_u \ge 0, \quad \forall u\in U,\ \forall f_{u,i}\in SC_u, \forall k\in[1,K], \forall v\in V, \tag{44}$$

where $\alpha_u, \beta_v, \sigma_{u,i}, \lambda_{u,i,k}, \mu_u$ are dual variables for Constraints (30), (31), (32), (33) and (37), respectively. Especially, $\sigma_{u,i}$ is unconstrained, while $\alpha_u, \beta_v, \lambda_{u,i,k}, \mu_u$ are non-negative, as shown in Constraint (44).

By Constraint (42), we have

$$\sum_{f_{u,i}\in SC_u}\sigma_{u,i} \ge -\frac{1}{|V|}\cdot\sum_{f_{u,i}\in SC_u} c(f_{u,i})\cdot\sum_{v\in V}\beta_v. \tag{45}$$

Combining Constraints (41) and (45), we have

$$\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K}\lambda_{u,i,k} \le H'(u)\cdot\alpha_u + \frac{1}{|V|}\cdot\sum_{f_{u,i}\in SC_u} c(f_{u,i})\cdot\sum_{v\in V}\beta_v + \mu_u - 1. \tag{46}$$

Because $\lambda_{u,i,k} \ge 0$, we have

$$H'(u)\cdot\alpha_u + \frac{1}{|V|}\cdot\sum_{f_{u,i}\in SC_u} c(f_{u,i})\cdot\sum_{v\in V}\beta_v + \mu_u - 1 \ge 0. \tag{47}$$

We assume $H'(u) > 0$, i.e., request u cannot have its SFC reliability requirement met with only the primary VNF instances. We have

$$\alpha_u \ge \frac{1 - \frac{1}{|V|}\cdot\sum_{f_{u,i}\in SC_u} c(f_{u,i})\cdot\sum_{v\in V}\beta_v - \mu_u}{H'(u)}. \tag{48}$$

From Constraint (43), we have

$$\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K}\lambda_{u,i,k} \ge \sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K} H(u,i,k)\cdot\alpha_u - \frac{1}{|V|}\cdot\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K} c(f_{u,i})\cdot\sum_{v\in V}\beta_v. \tag{49}$$

Combining Ineq. (46) and (49), we have

$$H'(u)\cdot\alpha_u + 1/|V|\cdot\sum_{f_{u,i}\in SC_u} c(f_{u,i})\cdot\sum_{v\in V}\beta_v + \mu_u - 1 \ge \sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K} H(u,i,k)\cdot\alpha_u - 1/|V|\cdot\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K} c(f_{u,i})\cdot\sum_{v\in V}\beta_v. \tag{50}$$

Recall that we assume $\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K} H(u,i,k) \ge H'(u)$, by Ineq. (48) and (50), we have

$$\mu_u \ge 1 - \left(1 + \frac{K\cdot H'(u)}{\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K} H(u,i,k)}\right)\cdot\frac{\sum_{f_{u,i}\in SC_u} c(f_{u,i})}{|V|}\cdot\sum_{v\in V}\beta_v. \tag{51}$$

For notation simplicity, we define a constant $\psi_u \ge 0$ for each user request $u \in U$ as follows.

$$\psi_u = \left(1 + \frac{K\cdot H'(u)}{\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K} H(u,i,k)}\right)\cdot\frac{\sum_{f_{u,i}\in SC_u} c(f_{u,i})}{|V|}. \tag{52}$$

We then have

$$\mu_u \ge 1 - \psi_u\cdot\textstyle\sum_{v\in V}\beta_v. \tag{53}$$

We claim that given dual variables $\mu_u \ge 0$ and $\beta_v \ge 0$ subject to Ineq. (53), there always exist feasible dual variables α_u, $\lambda_{u,i,k}$ and $\sigma_{u,i}$ to deliver a feasible solution to problem **P5**. This claim will later be shown in Lemma 6. Therefore, we only focus on μ_u and β_v to devise a feasible solution to problem **P5** by Ineq. (53).

The online algorithm proceeds by simultaneously updating variables in primal and dual problems. Dual variables μ_u and β_v are set as 0 s initially. Upon the arrival of request u, we need to set the dual variables subject to Ineq. (53) through an admission control policy for the admission of each request u as follows. If $1 - \psi_u\cdot\sum_{v\in V}\beta_v \le 0$, the request u is rejected. Otherwise, it is admitted with updating the dual variable μ_u as follows.

$$\mu_u = 1 - \psi_u \cdot \sum_{v \in V} \beta_v. \tag{54}$$

We then obtain the VNF instance set $\mathbb{F}_u$ for user request u to meet its reliability requirement, by constructing an instance of problem **P2** for the single request u and adopting `Algorithm` 1 to solve it. We sort the VNF instances in $\mathbb{F}_u$ in non-increasing order based on their computing resource consumption, and provide the deployment details of VNF instances from $\mathbb{F}_u$ to cloudlets as follows.

Algorithm 3. An Online Algorithm for the Dynamic Service Admission Maximization Problem (Problem **P3**).

Input: The MEC network $G = (V \cup \{v_0\}, E)$, the sequence U of incoming requests with reliability requirements arriving one by one without future knowledge, and digital twins running in the remote cloud to provide reliability prediction of VNF instances.

Output: An online scheduling of incoming requests with the aim of maximizing request admissions.

$\mu_u \leftarrow 0, \forall u \in U;\ \beta_v \leftarrow 0, \forall v \in V;$
while a request u arrives **do**
  if $1 - \psi_u \cdot \sum_{v \in V} \beta_v > 0$ **then**
    Admit request u, and update μ_u by Eq. (54);
    Construct an instance of problem **P2** for the single request u, and obtain the VNF instance set $\mathbb{F}_u$ for problem **P2** by `Algorithm` 1;
    Sort the VNF instances in $\mathbb{F}_u$ in non-increasing order by computing resource demands;
    $\mathcal{F}_{u,v} \leftarrow \emptyset, \forall v \in V;$
    for each VNF instance $f \in \mathbb{F}_u$ in its sorted order **do**
      Identify the cloudlet $v' \in V$ with the minimum $\eta(\beta_{v'}, \mathcal{F}_{u,v'})$, and upda te $\mathcal{F}_{u,v'} \leftarrow \mathcal{F}_{u,v'} \cup \{f\};$
    end for ;
    for each cloudlet $v \in V$ **do**
      Deploy VNF instances in $\mathcal{F}_{u,v}$ to cloudlet v, and update β_v by Eq. (56);
    end for;
  else
    Reject request u;
  end if;
end while;

Suppose $\mathcal{F}_{u,v}$ is the set of VNF instances deployed in cloudlet v for request u, with $\cup_{f \in \mathcal{F}_{u,v}} = \mathbb{F}_u$, and $\mathcal{F}_{u,v} = \emptyset$ initially, $\forall v \in V$. Note that function $\eta(\beta_v, \mathcal{F}_{u,v})$ is used for the updating of the dual variable β_v when deploying the VNF instances in $\mathcal{F}_{u,v}$ to cloudlet v, which will guide the deployment of VNF instances to cloudlets.

$$\eta(\beta_v, \mathcal{F}_{u,v}) = \beta_v \cdot \left(1 + \frac{\psi_u \cdot \sum_{f \in \mathcal{F}_{u,v}} c(f)}{cap_v \cdot (K+1) \cdot \sum_{f_{u,i} \in SC_u} c(f_{u,i})}\right) + \frac{\psi_u \cdot \sum_{f \in \mathcal{F}_{u,v}} c(f)}{cap_v \cdot (K+1) \cdot \sum_{f_{u,i} \in SC_u} c(f_{u,i})}. \tag{55}$$

For each VNF instance $f \in \mathbb{F}_u$ in sorted order, we identify a cloudlet $v' \in V$ with the minimum $\eta(\beta_{v'}, \mathcal{F}_{u,v'})$, and update $\mathcal{F}_{u,v'} \leftarrow \mathcal{F}_{u,v'} \cup \{f\}$. This procedure continues until all VNF instances in $\mathbb{F}_u$ are deployed. Having obtained $\{\mathcal{F}_{u,v} \mid \forall v \in V\}$ for request u, we update the dual variable β_v of each cloudlet v as follows.

$$\beta_v = \beta_v \cdot \left(1 + \frac{\psi_u \cdot \sum_{f \in \mathcal{F}_{u,v}} c(f)}{cap_v \cdot (K+1) \cdot \sum_{f_{u,i} \in SC_u} c(f_{u,i})}\right) + \frac{\psi_u \cdot \sum_{f \in \mathcal{F}_{u,v}} c(f)}{cap_v \cdot (K+1) \cdot \sum_{f_{u,i} \in SC_u} c(f_{u,i})}. \tag{56}$$

The details of the online algorithm for the dynamic service admission maximization problem (problem **P3**) are given in `Algorithm` 3.

5.3 Algorithm Analysis

Lemma 5. *The ILP solution (29) delivers an optimal solution to the offline version of the dynamic service admission maximization problem.*

The proof of Lemma 5 is similar to Lemma 2, omitted.

Lemma 6. *(1) Having obtained dual variables $\mu_u \geq 0$ and $\beta_v \geq 0$ subject to Ineq. (53), there always exist feasible dual variables α_u, $\lambda_{u,i,k}$ and $\sigma_{u,i}$ to deliver a feasible solution for problem **P5**; and (2) `Algorithm` 3 does deliver a feasible solution for problem **P5**.*

Proof. (1) We determine the feasible dual variables $\mu_u \geq 0$ and $\beta_v \geq 0$ subject to Ineq. (53). Given feasible μ_u and β_v, there always exists a feasible dual variable $\alpha_u \geq 0$ by Ineq. (48). If Ineq. (53) holds, with Ineq. (48), it exists feasible dual variable $\lambda_{u,i,k} \geq 0$ meeting Ineq. (46) and (49) with the given μ_u, β_v and α_u, where Ineq. (46) is derived from Constraint (43). Similarly, with Ineq. (46), it exists feasible dual variable $\sigma_{u,i}$ subject to Constraints (41) and (42) with given μ_u, β_v, α_u and $\lambda_{u,i,k}$. Therefore, given $\mu_u \geq 0$ and $\beta_v \geq 0$ subject to Ineq. (53), there always exist feasible α_u, $\lambda_{u,i,k}$ and $\sigma_{u,i}$ to deliver a feasible solution for problem **P5**.

(2) When updating μ_u and β_v, Ineq. (53) still holds. From the update function (54) of μ_u, Ineq. (53) holds with an incoming request u. Due to the non-decreasing nature of the update function (56) of β_v, then $1 - \psi_u \cdot \sum_{v \in V} \beta_v$ is also non-increasing with its updating. Since $\mu_u \geq 0$ and $\beta_v \geq 0$ subject to Ineq. (53), the feasible α_u, $\lambda_{u,i,k}$ and $\sigma_{u,i}$ can be obtained, delivering a feasible solution for problem **P5**. □

Recall that ψ_u is a constant defined in (52). Let ψ_{max} and ψ_{min} be the maximum and minimum values of ψ_u with $\psi_{max} = \max_{u \in U}\{\psi_u\}$ and $\psi_{min} = \min_{u \in U}\{\psi_u\}$, respectively. Recall that c_{max} and c_{min} are the maximum and minimum amounts of demanded computing resources by any VNF instance. $|SC|_{max}$ and $|SC|_{min}$ denote the maximum and minimum SFC lengths, respectively.

We define $\hat{x}_{u,v}$ as a binary decision variable. $\hat{x}_{u,v} = 1$ means that at least one VNF instance for request u in cloudlet v is deployed; otherwise, $\hat{x}_{u,v} = 0$. We have $\hat{x}_{u,v} = 0\ \forall v \in V$, if user request u is rejected.

Lemma 7. *When updating the dual variable β_v by `Algorithm` 3, the following inequalities hold.*

$$\beta_v \geq \left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)cap_{max}|SC|_{max}c_{max}}\right)^{\sum_{u\in U}\hat{x}_{u,v}}-1; \quad (57)$$

$$\beta_v < \frac{1}{\psi_{min}}\cdot\left(1+\frac{\psi_{max}}{cap_{min}}\right)+\frac{\psi_{max}}{cap_{min}}. \quad (58)$$

Proof. We first show that Ineq. (57) holds by induction. The right-hand side of Ineq. (57) is 0 before the first request arrives. The induction hypothesis holds initially because $\beta_v = 0$ initially. Let $\beta_v(start)$ and $\beta_v(end)$ indicate β_v before and after the arrival of request u'. We show that Ineq. (57) holds by induction, no matter whether the value of β_v is updated or not.

Case (i). β_v has no update. This is incurred by the rejection of request u', or the request u' is admitted with no VNF instance for u' deployed in cloudlet v with $\hat{x}_{u,'v}=0$. Because β_v is not updated, i.e., $\beta_v(end)=\beta_v(start)$, then

$$\begin{aligned}
&\beta_v(end) = \beta_v(start)\\
\geq &\left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)cap_{max}|SC|_{max}c_{max}}\right)^{\sum_{u\in U\setminus\{u'\}}\hat{x}_{u,v}}-1\\
\geq &\left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)cap_{max}|SC|_{max}c_{max}}\right)^{\sum_{u\in U\setminus\{u'\}}\hat{x}_{u,v}}\cdot\\
&\left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)cap_{max}|SC|_{max}c_{max}}\right)^{\hat{x}_{u'v}}-1, \text{ by } \hat{x}_{u,'v}=0\\
= &\left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)cap_{max}|SC|_{max}c_{max}}\right)^{\sum_{u\in U}\hat{x}_{u,v}}-1. \quad (59)
\end{aligned}$$

Case (ii). β_v is updated. This is incurred by the admission of request u' with at least one VNF instance for u' deployed in cloudlet v with $\hat{x}_{u,'v}=1$. By the update function (56),

$$\begin{aligned}
\beta_v(end) = \; &\beta_v(start)\cdot\left(1+\frac{\psi_{u'}\cdot\sum_{f\in\mathcal{F}_{u,'v}}c(f)}{cap_v\cdot(K+1)\cdot\sum_{f_{u,'i}\in SC_{u'}}c(f_{u,'i})}\right)\\
&+\frac{\psi_{u'}\cdot\sum_{f\in\mathcal{F}_{u,'v}}c(f)}{cap_v\cdot(K+1)\cdot\sum_{f_{u,'i}\in SC_{u'}}c(f_{u,'i})}\\
\geq \; &\beta_v(start)\cdot\left(1+\frac{\psi_{min}\cdot c_{min}}{cap_{max}\cdot(K+1)\cdot|SC|_{max}\cdot c_{max}}\right)\\
&+\frac{\psi_{min}\cdot c_{min}}{cap_{max}\cdot(K+1)\cdot|SC|_{max}\cdot c_{max}}\\
\geq &\left(\left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)cap_{max}|SC|_{max}c_{max}}\right)^{\sum_{u\in U\setminus\{u'\}}\hat{x}_{u,v}}-1\right)\\
&\cdot\left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)cap_{max}|SC|_{max}c_{max}}\right)\\
&+\frac{\psi_{min}\cdot c_{min}}{(K+1)cap_{max}|SC|_{max}c_{max}}, \quad \text{by Ineq. (57)}\\
\geq &\left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)cap_{max}|SC|_{max}c_{max}}\right)^{\sum_{u\in U\setminus\{u'\}}\hat{x}_{u,v}}\cdot\\
&\left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)cap_{max}|SC|_{max}c_{max}}\right)^{\hat{x}_{u,'v}}-1, \text{by } \hat{x}_{u,'v}=1\\
= &\left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)cap_{max}|SC|_{max}c_{max}}\right)^{\sum_{u\in U}\hat{x}_{u,v}}-1. \quad (60)
\end{aligned}$$

We then show that Ineq. (58) holds. Recall that β_v is 0 initially. We only admit a new request u by placing its VNF instances in cloudlets with $1-\psi_u\cdot\sum_{v\in V}\beta_v > 0$, and updating the value of β_v of each chosen cloudlet $v\in V$. Assuming that cloudlet v is chosen, we then have

$$1-\psi_u\cdot\sum_{v\in V}\beta_v > 0 \Rightarrow \sum_{v\in V}\beta_v < \frac{1}{\psi_u} \Rightarrow \beta_v < \frac{1}{\psi_{min}}.$$

For each cloudlet v hosting at least a VNF instance of request u, if $\beta_v \geq \frac{1}{\psi_{min}}$, β_v will not be updated; otherwise, β_v can be updated by the update function (56).

$$\begin{aligned}
\beta_v = &\beta_v\cdot\left(1+\frac{\psi_u\cdot\sum_{f\in\mathcal{F}_{u,v}}c(f)}{cap_v\cdot(K+1)\cdot\sum_{f_{u,i}\in SC_u}c(f_{u,i})}\right)+\\
&\frac{\psi_u\cdot\sum_{f\in\mathcal{F}_{u,v}}c(f)}{cap_v\cdot(K+1)\cdot\sum_{f_{u,i}\in SC_u}c(f_{u,i})}\\
< &\frac{1}{\psi_{min}}\cdot\left(1+\frac{\psi_{max}}{cap_{min}}\right)+\frac{\psi_{max}}{cap_{min}}, \quad (61)
\end{aligned}$$

where Ineq. (61) holds because $\mathbb{F}_u$ consists of at most $(K+1)$ VNF instances for each VNF of request u, and $\mathcal{F}_{u,v}\subseteq\mathbb{F}_u$, i.e., $\sum_{f\in\mathcal{F}_{u,v}}c(f)\leq(K+1)\cdot\sum_{f_{u,i}\in SC_u}c(f_{u,i})$. □

Lemma 8. *The resource violation on each cloudlet is upper bounded by Λ with* `Algorithm` *3 for problem **P3**, where*

$$\Lambda = \frac{(K+1)\cdot|SC|_{max}\cdot c_{max}}{cap_{min}}\cdot\frac{\ln(\frac{1}{\psi_{min}}\cdot(1+\frac{\psi_{max}}{cap_{min}})+\frac{\psi_{max}}{cap_{min}}+1)}{\ln(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)\cdot cap_{max}\cdot|SC|_{max}\cdot c_{max}})}-1.$$

Proof. From Lemma 7, we have

$$\begin{aligned}
&\left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)\cdot cap_{max}\cdot|SC|_{max}\cdot c_{max}}\right)^{\sum_{u\in U}\hat{x}_{u,v}}-1\\
<&\frac{1}{\psi_{min}}\cdot\left(1+\frac{\psi_{max}}{cap_{min}}\right)+\frac{\psi_{max}}{cap_{min}}. \quad (62)
\end{aligned}$$

We then have

$$\sum_{u\in U}\hat{x}_{u,v} < \frac{\ln\left(\frac{1}{\psi_{min}}\cdot\left(1+\frac{\psi_{max}}{cap_{min}}\right)+\frac{\psi_{max}}{cap_{min}}+1\right)}{\ln\left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)\cdot cap_{max}\cdot|SC|_{max}\cdot c_{max}}\right)}. \quad (63)$$

The amount of utilized computing resource in cloudlet v is calculated as follows.

$$\begin{aligned}
&\sum_{u\in U}\sum_{f_{u,i}\in SC_u}\sum_{k=0}^{K}c(f_{u,i})\cdot x_{u,i,k,v}\\
\leq&(K+1)\cdot|SC|_{max}\cdot c_{max}\cdot\sum_{u\in U}\hat{x}_{u,v}\\
\leq&(K+1)\cdot|SC|_{max}\cdot c_{max}\cdot\\
&\frac{\ln(\frac{1}{\psi_{min}}\cdot\left(1+\frac{\psi_{max}}{cap_{min}}\right)+\frac{\psi_{max}}{cap_{min}}+1)}{\ln\left(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)\cdot cap_{max}\cdot|SC|_{max}\cdot c_{max}}\right)}, \text{by Ineq. (63)}.
\end{aligned}$$

By constraint (31), the resource violation on a cloudlet is no more than Λ, which is defined in this lemma. □

Lemma 9. *Let $\mathcal{A}_3$ and $\mathcal{A}_5$ be the values of the solutions by* `Algorithm` *3, for problems* **P3** *and* **P5**, *respectively. We have*

$$(1 + \psi_{max}) \cdot \mathcal{A}_3 \geq \mathcal{A}_5, \tag{64}$$

where $\psi_{max} = \max_{u \in U}\{\psi_u\}$, and ψ_u is defined in (52).

Proof. Initially the claim holds due to $\mathcal{A}_3 = \mathcal{A}_5 = 0$. We then show $(1 + \psi_{max}) \cdot \Delta\mathcal{A}_3 \geq \Delta\mathcal{A}_5$ after the arrival of request u, where $\Delta\mathcal{A}_3$ and $\Delta\mathcal{A}_5$ are the differences of solution values of problems **P3** and **P5** before and after the arrival of u.

Suppose that request u is rejected, $\Delta\mathcal{A}_3 = \Delta\mathcal{A}_5 = 0$, and $(1 + \psi_{max}) \cdot \Delta\mathcal{A}_3 \geq \Delta\mathcal{A}_5$. Otherwise ($u$ is admitted), $\Delta\mathcal{A}_3 = 1$ and $\Delta\mathcal{A}_5 = \sum_{v \in V} cap_v \cdot \Delta\beta_v + \mu_u$ by (40), where $\Delta\beta_v$ is the value difference before and after the update of β_v. By update functions (56) and (54) of β_v and μ_u, we have

$$\begin{aligned}\Delta\mathcal{A}_5 = &\sum_{v \in V} cap_v \cdot \Bigg(\frac{\psi_u \cdot \sum_{f \in \mathcal{F}_{u,v}} c(f)}{cap_v \cdot (K+1) \cdot \sum_{f_{u,i} \in SC_u} c(f_{u,i})} \cdot \beta_v + \\ &\frac{\psi_u \cdot \sum_{f \in \mathcal{F}_{u,v}} c(f)}{cap_v \cdot (K+1) \cdot \sum_{f_{u,i} \in SC_u} c(f_{u,i})} \Bigg) + \mu_u \\ = &\psi_u \cdot \frac{\sum_{v \in V} \sum_{f \in \mathcal{F}_{u,v}} c(f) \cdot \beta_v}{(K+1) \cdot \sum_{f_{u,i} \in SC_u} c(f_{u,i})} + \\ &\psi_u \cdot \frac{\sum_{v \in V} \sum_{f \in \mathcal{F}_{u,v}} c(f)}{(K+1) \cdot \sum_{f_{u,i} \in SC_u} c(f_{u,i})} + 1 - \psi_u \cdot \sum_{v \in V} \beta_v \\ \leq &1 + \psi_u \leq (1 + \psi_{max}) \cdot \Delta\mathcal{A}_3,\end{aligned} \tag{65}$$

where Ineq. (65) holds because $\sum_{v \in V} \sum_{f \in \mathcal{F}_{u,v}} c(f) \leq (K+1) \cdot \sum_{f_{u,i} \in SC_u} c(f_{u,i})$, i.e., at most K backups are deployed for each primary VNF instance, then we have $\psi_u \cdot \frac{\sum_{v \in V} \sum_{f \in \mathcal{F}_{u,v}} c(f)}{(K+1) \cdot \sum_{f_{u,i} \in SC_u} c(f_{u,i})} \leq \psi_u$. Also, we have $\sum_{f \in \mathcal{F}_{u,v}} c(f) \leq (K+1) \cdot \sum_{f_{u,i} \in SC_u} c(f_{u,i})$, therefore, $\psi_u \cdot \frac{\sum_{v \in V} \sum_{f \in \mathcal{F}_{u,v}} c(f) \cdot \beta_v}{(K+1) \cdot \sum_{f_{u,i} \in SC_u} c(f_{u,i})} \leq \psi_u \cdot \sum_{v \in V} \beta_v$. □

Theorem 4. *Given an MEC network $G = (V \cup \{v_0\}, E)$ and a sequence U of requests with SFC and reliability requirements arriving one by one without the knowledge of future arrivals, there is an online algorithm,* `Algorithm` *3, with the competitive ratio of $(1 + \psi_{max})$ for the dynamic service admission maximization problem, at the expense of computing capacity violation on each cloudlet no greater than Λ, where $\psi_{max} = \max_{u \in U}\{\psi_u\}$ with the constant ψ_u defined in (52), and $\Lambda > 0$ is a constant given in Lemma 8. The algorithm takes $O(|SC|_{max} \cdot K \cdot (\log_2(|SC|_{max} \cdot K) + |V|))$ time to admit each request, where $|SC|_{max}$ is the maximum length of any SFC, K is the maximum number of backups deployed for a primary VNF instance, and V is the cloudlet set.*

Proof. Denote by OPT_3, OPT_4 and OPT_5 the values of the optimal solutions of problems **P3**, **P4**, and **P5**, respectively. Let $\mathcal{A}_3$ and $\mathcal{A}_5$ be the values of the solutions by `Algorithm` 3 for **P3** and **P5**, respectively. Then, $(1 + \psi_{max}) \cdot \mathcal{A}_3 \geq \mathcal{A}_5$ by Lemma 9. Because $\mathcal{A}_5$ is the value of a feasible solution to the minimization problem **P5** due to Lemma 6, we have $\mathcal{A}_5 \geq OPT_5$. Furthermore, problem **P5** is the dual problem of problem **P4**, then $OPT_5 \geq OPT_4$ due to the weak duality. Since problem **P4** is the LP relaxation of the maximization problem **P3**, we have $OPT_4 \geq OPT_3$. Thus, $\mathcal{A}_5 \geq OPT_3$. Then,

$$\mathcal{A}_3 \geq \mathcal{A}_5/1 + \psi_{max} \geq OPT_3/1 + \psi_{max}. \tag{66}$$

The resource violation on any cloudlet is no more than Λ by Lemma 8.

We analyze the time complexity of `Algorithm` 3 via examining the admission of each request u as follows. It takes $O(|SC|_{max} \cdot K \cdot \log_2(|SC|_{max} \cdot K))$ time to obtain the VNF instance set $\mathbb{F}_u$ for request u by invoking `Algorithm` 1 as shown in Lemma 3. Then, it takes $O(|SC|_{max} \cdot K \cdot \log_2(|SC|_{max} \cdot K)))$ time to sort the VNF instances in $\mathbb{F}_u$. It takes $O(|SC|_{max} \cdot K \cdot |V|)$ time to identify set $\mathcal{F}_{u,v}$ for each cloudlet v. Thus, `Algorithm` 3 takes $O(|SC|_{max} \cdot K \cdot (\log_2(|SC|_{max} \cdot K) + |V|))$ time. □

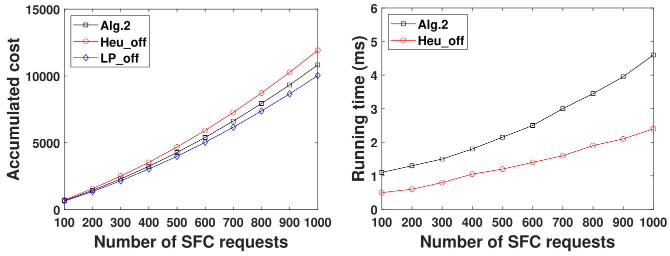
6 Performance Evaluation

In this section, we evaluated the performance of the proposed algorithms for digital twin-assisted SFC-enabled reliable service provisioning in MEC networks. We also investigated the impact of parameters on the performance of the proposed algorithms.

6.1 Experimental Environment Settings

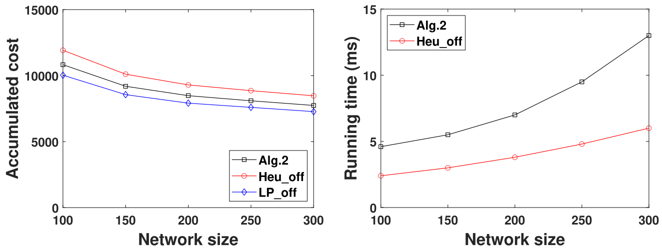
We generate each MEC network instance by adopting the tool GT-ITM [3], where each MEC network consists of 100 APs with each AP co-located with a cloudlet. Each cloudlet possesses a computing capacity ranging from 4,000 MHz to 14,000 MHz [31], while its unit computing resource cost ranges from \$0.01 to \$0.03 per MHz [31]. We further assume that the network service provider offers 20 different types of VNFs. The computing resource demand of a VNF instance is set between 20 MHz and 100 MHz, while the reliability of each VNF for each request over time is randomly drawn from 0.8 to 0.9 [34]. There are 30 different SFCs, and the length of each SFC is set between 2 and 6 [34]. Each request is associated with a random SFC requirement, and its SFC reliability requirement ranges from 0.9 to 0.99 [34]. Each primary VNF instance can possess at most three backups, i.e., K is set at 3. The value in each figure is the result by averaging the results based on 30 different MEC instances with the same network size. Simulations are performed by a desktop with a 3.60 GHz Intel i7 octa-core CPU and 16 GB RAM. Unless otherwise specified, the above-mentioned parameters are adopted by default.

To evaluate `Algorithm` 2, referred to as `Alg.2`, for the service cost minimization problem, a heuristic algorithm `Heu_off` is proposed as follows. For each request, algorithm `Heu_off` identifies a VNF backup of a network function in SFC of the request with the maximum reliability argumentation of the request iteratively until the reliability requirement of the request is met, and all primary and identified backup VNF instances are deployed to cloudlets with the



(a) Accumulated cost (b) Running time

Fig. 1. Performance of different algorithms for the service cost minimization problem.



(a) Accumulated cost (b) Running time

Fig. 2. Impact of the network size on different algorithms for the service cost minimization problem.

lowest unit computing resource cost and sufficient residual computing resource one by one. Notice that the lower bound on the optimal solution of the problem is the value of the LP relaxation of the ILP formulation (11) of the problem, referred to as `LP_off`.

To evaluate `Algorithm` 3, referred to as `Alg.3`, for the dynamic service admission maximization problem, a heuristic `Heu_on` is given as follows. For each arrived request, algorithm `Heu_on` tries to admit the request by identifying the backup for it with the largest reliability argumentation iteratively until meeting its reliability requirement, and all identified primary and backup VNF instances are deployed one by one to the cloudlets that possess sufficient residual computing resource. Notice that an upper bound on the optimal solution of the problem is the LP relaxation of its ILP solution (29), referred to as `LP_on`.

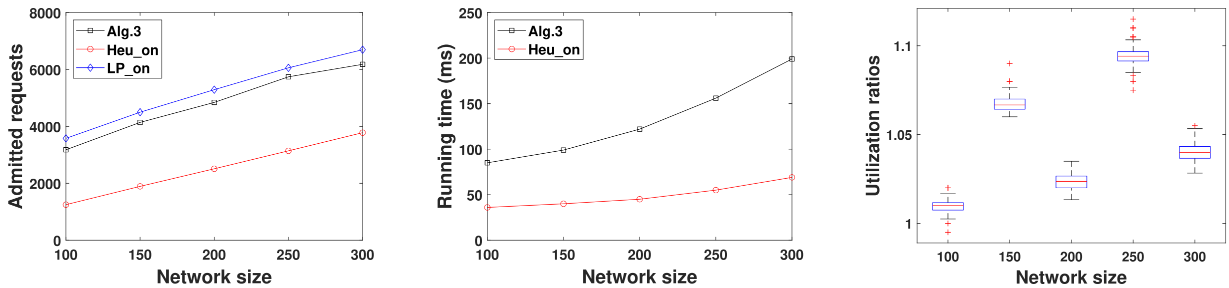
6.2 Algorithm Performance for the Service Cost Minimization Problem

We first evaluated the performance of algorithm `Alg.2` against algorithms `Heu_off` and `LP_off` by varying the number of requests from 100 to 1,000 while fixing network size at 100. It can be seen from Fig. 1a that when there are 1,000 requests, the accumulated service cost by `Alg.2` is 10.2% lower than that by `Heu_off`, and only 7.9% higher than that by `LP_off`. By Fig. 1b, `Alg.2` takes more running time than that of `Heu_off` due to the construction of the VNF instance set $\mathbb{F}$ for all requests. Fig. 1 demonstrates that `Alg.2` outperforms `Heu_off`. The justification is that `Alg.2` identifies the number of backup VNF instances to minimize the computing resource consumption for request admissions. Furthermore, `Alg.2` deploys VNF instances to the cloudlets with a cost-efficient manner.

We then investigated the impact of network size on the performance of different algorithms, by varying the network size from 100 to 300 while fixing the number of user requests at 1,000. It can be seen from Fig. 2a that when network size reaches 300, the accumulated service cost by `Alg.2` is 6.4% higher than that by `LP_off`, while the accumulated service cost of `Heu_off` is 16.5% more than that of `LP_off`. Furthermore, when the network size is 300, the accumulated service cost by `Alg.2` is 71.5% of that by itself when the network size is 100. The rationale is a larger network size brings more cheap computing resource, therefore, `Alg.2` can identify cloudlets for VNF instance deployments of requests with a relatively less accumulative service cost. It also can be seen from Fig. 2b that all mentioned algorithms take more running times with larger network size, because more cloudlets are considered for VNF instance deployments for user service requests.

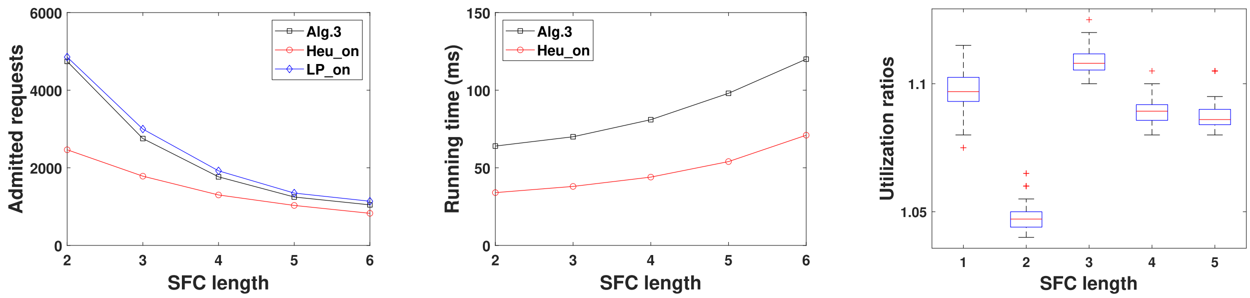
6.3 Algorithm Performance for the Dynamic Service Admission Maximization Problem

We first evaluated the performance of `Alg.3` against algorithms `Heu_on` and `LP_on` for the dynamic service admission maximization problem, by varying the network size from 100 to 300 while fixing the number of requests at 10,000. Fig. 3 demonstrates the evaluation results, where the utilization ratio of a cloudlet v is the ratio of the amount of its consumed computing resource to its computing capacity. Fig. 3a shows that algorithm `Alg.3` outperforms algorithm `Heu_on` by 63.6%, and the performance of `Alg.3` is 92.3% of algorithm `LP_on`, when the network size is set at 300. It can be seen from Fig. 3b that `Alg.3` takes more running time than that of `Heu_on`, due to the fact that `Alg.3` constructs the VNF instance set $\mathbb{F}_u$ for each incoming request u. Fig. 3c depicts the cloudlet utilization ratio by algorithm `Alg.3`, with the resource violation on each cloudlet no greater than 11.5% of its capacity. Recall that we have given an upper bound on



(a) The number of admitted requests (b) Running time (c) Cloudlet utilization ratios by `Alg.3`

Fig. 3. Performance of different algorithms for the dynamic service admission maximization problem.



(a) The number of admitted requests (b) Running time (c) Cloudlet utilization ratios by Alg.3

Fig. 4. Impact of the SFC length on different algorithms for the dynamic service admission maximization problem.

the resource violation of each cloudlet by Algorithm 3 for in Lemma 8, which is very conservative. It can be seen that the empirical results in Fig. 3c indicate that the actual resource violation is much less than the analytical one. Fig. 3 demonstrates that algorithm Alg.3 has a promising performance, compared with algorithms Heu_on and LP_on, respectively.

We then studied the impact of the SFC length on the performance of different algorithms by varying the SFC length from 2 to 6 while the network size is fixed at 100, considering 10,000 incoming requests. Fig. 4 plots the performance curves of different algorithms. It can be seen from Fig. 4a that when SFC length is 6, the performance of Alg.3 is 26.4% higher than that of algorithm Heu_on, while the performance of Alg.3 is 91.8% of that by LP_on. Also, Fig. 4a illustrates the performance of algorithm Alg.3 when the SFC length is 6 is 22.1% of itself when the SFC length is 2. This is due to that a shorter SFC length leads to less computing resource consumption of admitted requests for their VNF instance deployments to meet their reliability requirements. Therefore, more requests can be admitted with shorter SFC lengths. Fig. 4c shows that the resource violation of each cloudlet by Alg.3 is no greater than 12.7% of its capacity. It can be seen from Fig. 4b that a longer SFC length leads to more running time for all comparison algorithms, because more potential backup VNF instances for each request need to be taken into consideration.

7 Conclusion

In this article, we investigated two novel digital twin-assisted, SFC-enabled reliable service provisioning problems in MEC, through exploring reliability dynamics in VNF instance placements, and showed the NP-hardness of both problems. By leveraging the digital twin technique that the reliability of a VNF instance can be dynamically predicted for each user request, we first proposed an exact solution to the service cost minimization problem through providing an ILP solution. We also devised an approximation algorithm with a constant approximation ratio for it when the problem size is large. We then considered the dynamic service provisioning by formulating a dynamic service admission maximization problem, for which we formulated an ILP solution for its offline version, followed by devising a performance-guaranteed online algorithm. We finally evaluated the performance of the proposed algorithms through simulations. The simulation results demonstrate that the proposed algorithms outperform their corresponding benchmarks, improving the algorithm performance by no less than 10.2% in comparison with those of their corresponding benchmarks.

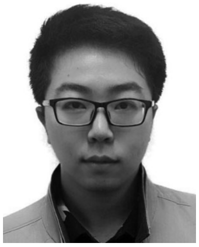
Acknowledgments

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

References

[1] A. Alleg, T. Ahmed, M. Mosbah, and R. Boutaba, “Joint diversity and redundancy for resilient service chain provisioning,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1490–1504, Jul. 2020.
[2] M. X. Goemans and D. P. Williamson, “The primal-dual method for approximation algorithms and its application to network design problems,” in *Book Chapter of Approximation Algorithms for NP-Hard Problems*. Boston, MA, USA: PWS, 1997, pp. 144–191.
[3] GT-ITM, 2019. [Online]. Available: http://www.cc.gatech.edu/projects/gtitm/
[4] B. He and K. Bai, “Digital twin-based sustainable intelligent manufacturing: A review,” *Adv. Manuf.*, vol. 9, no. 1, pp. 1–21, 2021.
[5] M. Huang, W. Liang, X. Shen, Y. Ma, and H. Kan, “Reliability-aware virtualized network function services provisioning in mobile edge computing,” *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2699–2713, Nov. 2020.
[6] G. Ishigaki, S. Devic, R. Gour, and J. P. Jue, “DeepPR: Progressive recovery for interdependent VNFs with deep reinforcement learning,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2386–2399, Oct. 2020.
[7] M. Karimzadeh-Farshbafan, V. Shah-Mansouri, and D. Niyato, “A dynamic reliability-aware service placement for network function virtualization (NFV),” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 318–333, Feb. 2020.
[8] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, “Digital-twin-enabled 6G: Vision, architectural trends, and future directions,” *IEEE Commun. Mag.*, vol. 60, no. 1, pp. 74–80, Jan. 2022.
[9] X. Lai, J. Xia, L. Fan, T. Q. Duong, and A. Nallanathan, “Outdated access point selection for mobile edge computing with cochannel interference,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7445–7455, Jul. 2022.
[10] B. Li, Y. Liu, L. Tan, H. Pan, and Y. Zhang, “Digital twin assisted task offloading for aerial edge computing and networks,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10 863–10 877, Oct. 2022.
[11] J. Li, S. Guo, W. Liang, Q. Chen, Z. Xu, and W. Xu, “SFC-enabled reliable service provisioning in mobile edge computing via digital twins,” in *Proc. IEEE Int. Conf. Mobile Ad-Hoc Smart Syst.*, 2022.
[12] 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.*, 2019, pp. 732–741.

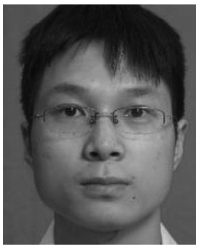
[13] J. Li, W. Liang, M. Huang, and X. Jia, "Reliability-aware network service provisioning in mobile edge-cloud networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 7, pp. 1545–1558, Jul. 2020.
[14] J. Li et al., "Budget-aware user satisfaction maximization on service provisioning in mobile edge computing," *IEEE Trans. Mobile Comput.*, early access, Sep. 09, 2022, doi: 10.1109/TMC.2022.3205427.
[15] J. Li, W. Liang, and Y. Ma, "Robust service provisioning with service function chain requirements in mobile edge computing," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 2138–2153, Jun. 2021.
[16] J. Li et al., "Maximizing user service satisfaction for delay-sensitive IoT applications in edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 5, pp. 1199–1212, May 2022.
[17] J. Li, W. Liang, W. Xu, Z. Xu, Y. Li, and X. Jia, "Service home identification of multiple-source IoT applications in edge computing," *IEEE Trans. Services Comput.*, early access, May 24, 2022, doi: 10.1109/TSC.2022.3176576.
[18] W. Liang, Y. Ma, W. Xu, X. Jia, and S. Chau, "Reliability augmentation of requests with service function chain requirements in mobile edge-cloud networks," in *Proc. 49th Int. Conf. Parallel Process.*, 2020, Art. no. 74.
[19] W. Liang, Y. Ma, W. Xu, Z. Xu, X. Jia, and W. Zhou, "Request reliability augmentation with service function chain requirements in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4541–4554, Dec. 2022.
[20] S. Lin, W. Liang, and J. Li, "Reliability-aware service function chain provisioning in mobile edge-cloud networks," in *Proc. IEEE 29th Int. Conf. Comput. Commun. Netw.*, 2020, pp. 1–9.
[21] T. Liu, L. Tang, W. Wang, Q. Chen, and X. Zeng, "Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1427–1444, Jan. 2022.
[22] J. Martín-Peréz, F. Malandrino, C. F. Chiasserini, and C. J. Bernardos, "OKpi: All-KPI network slicing through efficient resource allocation," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 804–813.
[23] J. Martins et al., "ClickOS and the art of network function virtualization," in *Proc. USENIX Symp. Netw. Syst. Des. Implementation*, 2014, pp. 459–473.
[24] S. Mi, Y. Feng, H. Zheng, Y. Wang, Y. Gao, and J. Tan, "Prediction maintenance integrated decision-making approach supported by digital twin-driven cooperative awareness and interconnection framework," *J. Manuf. Syst.*, vol. 58, pp. 329–345, 2021.
[25] T. Öncan, "A survey of the generalized assignment problem and its applications," *Inf. Syst. Oper. Res.*, vol. 45, no. 3, pp. 123–142, 2007.
[26] Y. Qiu, J. Liang, V. C. M. Leung, X. Wu, and X. Deng, "Online reliability-enhanced virtual network services provisioning in fault-prone mobile edge cloud," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 7299–7313, Sep. 2022.
[27] X. Shang, Y. Huang, Z. Liu, and Y. Yang, "Reducing the service function chain backup cost over the edge and cloud by a self-adapting scheme," *IEEE Trans. Mobile Comput.*, vol. 21, no. 8, pp. 2994–3008, Aug. 2022.
[28] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12 240–12 251, Oct. 2020.
[29] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with Big Data," *Int. J. Adv. Manuf. Technol.*, vol. 94, no. 9, pp. 3563–3576, 2018.
[30] C. Wang, Q. Hu, D. Yu, and X. Cheng, "Proactive deployment of chain-based VNF backup at the edge using online bandit learning," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst.*, 2021, pp. 740–750.
[31] 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.
[32] L. Yang, J. Jia, H. Lin, and J. Cao, "Reliable dynamic service chain scheduling in 5G networks," *IEEE Trans. Mobile Comput.*, early access, Mar. 07, 2022, doi: 10.1109/TMC.2022.3157312.
[33] S. Yang, F. Li, S. Trajanovski, X. Chen, Y. Wang, and X. Fu, "Delay-aware virtual network function placement and routing in edge clouds," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 445–459, Feb. 2021.
[34] J. Zhang, Z. Wang, C. Peng, L. Zhang, T. Huang, and Y. Liu, "RABA: Resource-aware backup allocation for a chain of virtual network functions," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1918–1926.



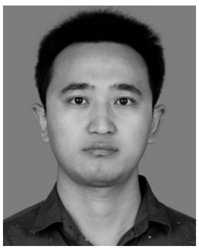
Jing Li received the BSc (first class honours) and PhD degrees from the Australian National University, in 2018 and 2022, respectively. He is currently a postdoctoral fellow with the Hong Kong Polytechnic University. His research interests include edge computing, Internet of Things, digital twin, network function virtualization, and combinatorial optimization.

Song Guo (Fellow, IEEE) is a full professor with the Department of Computing, Hong Kong Polytechnic University. He also holds a Changjiang chair professorship awarded by the Ministry of Education of China. His research interests are mainly in the areas of Big Data, edge AI, mobile computing, and distributed systems. With many impactful papers published in top venues in these areas, he has been recognized as a Highly Cited researcher (Web of Science) and received more than 12 best paper awards from IEEE/ACM conferences, journals and technical committees. He is the editor-in-chief of the *IEEE Open Journal of the Computer Society*. He has served on IEEE Communications Society Board of Governors, IEEE Computer Society Fellow Evaluation Committee, and editorial board of a number of prestigious international journals like *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Cloud Computing*, *IEEE Internet of Things Journal*, etc. He has also served as chair of organizing and technical committees of many international conferences. He is an ACM distinguished member.

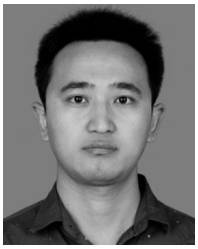
Weifa Liang (Senior Member, IEEE) received the BSc degree in computer science from Wuhan University, China, in 1984, the ME degree in computer science from the University of Science and Technology of China, in 1989, and the PhD degree in computer science from the Australian National University, in 1998. He is a full professor with the Department of Computer Science, City University of Hong Kong. Prior to that, he was a full professor with the Australian National University. His research interests include design and analysis of energy efficient routing protocols for wireless ad hoc and sensor networks, Mobile Edge Computing (MEC), Network Function Virtualization (NFV), Internet of Things and digital twins, design and analysis of parallel and distributed algorithms, approximation algorithms, and graph theory. He currently serves as an editor of the *IEEE Transactions on Communications*.



Quan Chen (Member, IEEE) received the BS, master's, and PhD degrees from the School of Computer Science and Technology, Harbin Institute of Technology, China. He is currently an associate professor with the School of Computers, Guangdong University of Technology. He once worked as a postdoctoral research fellow with the Department of Computer Science, Georgia State University. His research interests include routing and scheduling algorithms in wireless networks and sensor networks.



Zichuan Xu (Member, IEEE) received the BSc and ME degrees in computer science from the Dalian University of Technology, China, in 2008 and 2011, respectively, and the PhD degree in computer science from the Australian National University, in 2016. From 2016 to 2017, he was a research associate with the Department of Electronic and Electrical Engineering, University College London, U.K. He is currently a full professor and PhD advisor with the School of Software, Dalian University of Technology. His research interests include mobile edge computing, serverless computing, network function virtualization, algorithmic game theory, and optimization problems.



Wenzheng Xu (Member, IEEE) received the BSc, ME, and PhD degrees in computer science from Sun Yat-Sen University, Guangzhou, China, in 2008, 2010, and 2015, respectively. He currently is an associate professor with Sichuan University, China. Also, he was a visitor at both the Australian National University, Australia and the Chinese University of Hong Kong, Hong Kong. His research interests include wireless ad hoc and sensor networks, mobile computing, approximation algorithms, combinatorial optimization, online social networks, and graph theory.

Albert Y. Zomaya (Fellow, IEEE) is Peter Nicol Russell chair professor of computer science and director of the Centre for Distributed and High-Performance Computing, University of Sydney. He is currently the editor in chief of the *ACM Computing Surveys* and served in the past as editor in chief of *IEEE Transactions on Computers* (2010-2014) and *IEEE Transactions on Sustainable Computing* (2016-2020). He is a decorated scholar with numerous accolades including fellowship of the IEEE, the American Association for the Advancement of Science, and the Institution of Engineering and Technology. Also, he is a fellow of the Australian Academy of Science, Royal Society of New South Wales, Foreign member of Academia Europaea, and member of the European Academy of Sciences and Arts. Some of his recent awards include the New South Wales Premier's Prize of Excellence in Engineering and Information and Communications Technology (2019), the Research Innovation Award, IEEE Technical Committee on Cloud Computing (2021), the Technical Achievement and Recognition Award, IEEE Communications Society's IoT, Ad Hoc and Sensor Networks Technical Committee (2022). He is a Clarivate 2022 Highly Cited researcher, and his research interests lie in parallel and distributed computing, networking, and complex systems.